SMM3000X Series
Source Measure Unit

Programming Guide

EN01A

SIGLENT TECHNOLOGIES CO., LTD.

Contents

1 Programming Overview

This chapter introduces how to build communication between the instrument and the PC. It also introduces how to configure a system for remote instrument control.

Users can remotely control the instrument through USB and LAN interfaces, in combination with National Instruments NI-VISA and programming languages. Through the LAN interface, users can communicate using VXI-11, Sockets and Telnet protocols, depending on the capabilities of the specific instrument.

1.1 Establishing Communications

1.1.1 Install NI-VISA

USB control requires the National Instruments NI-VISA Library for communications. We also recommend using it for LAN communications for its ease of use, but sockets, telnet, and VXI-11 can also be implemented via LAN connections.

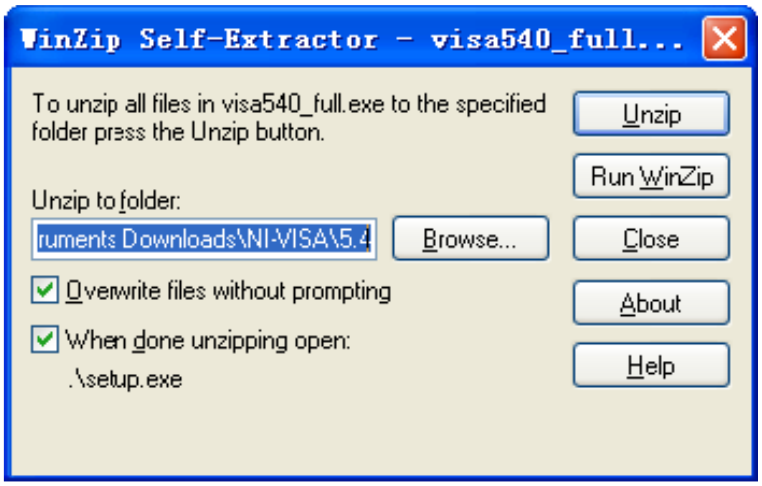
Currently, NI-VISA is packaged in two versions: A full version and a Run-Time Engine version. The full version includes the NI device drivers and a tool named NI MAX which is a user interface to control and test remotely connected devices. The Run-Time Engine is recommended, as it is a much smaller download than the full version and includes the necessary tools for basic communication to instruments.

For example, you can get the NI-VISA 5.4 full version from http://www.ni.com/download/ni-visa-5.4/4230/en/.

You also can download NI-VISA Run-Time Engine 5.4 to your PC and install it as the default selection. Its installation process is similar to the full version.

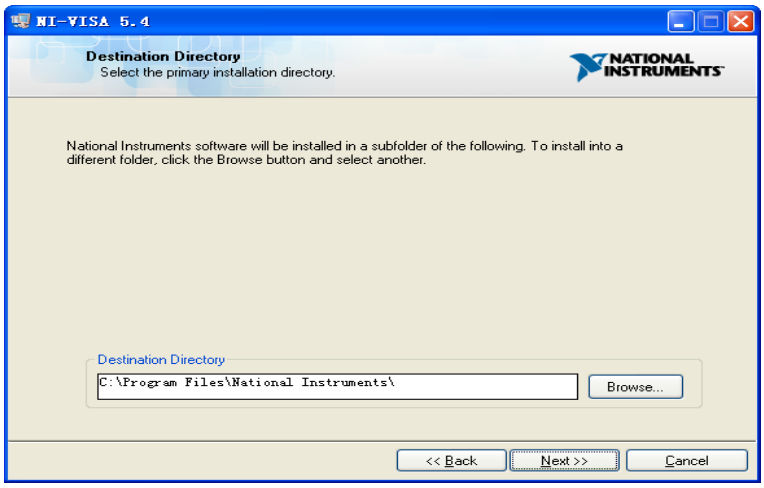
After you downloaded the file, follow these steps to install NI-VISA (The full version of NI-VISA 5.4 is used in this example. Newer versions are likely and should be compatible with SIGLENT instrumentation. Download the latest version available for the operating system being used by the controlling computer):

a. Double click the visa540_full.exe, the dialog will be similar to that shown below:

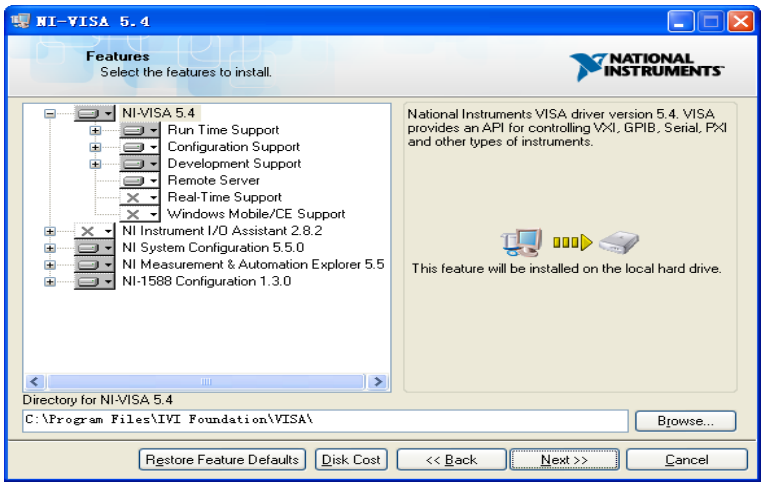


b. Click Unzip, the installation process will automatically launch after unzipping files. If your computer needs to install .NET Framework 4, it may auto start.

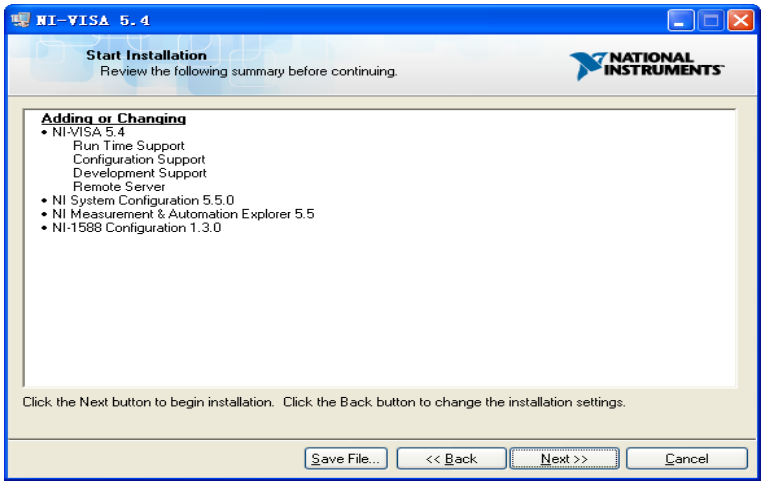
c. The NI-VISA installing dialog is shown above. Click Next to start the installation process.



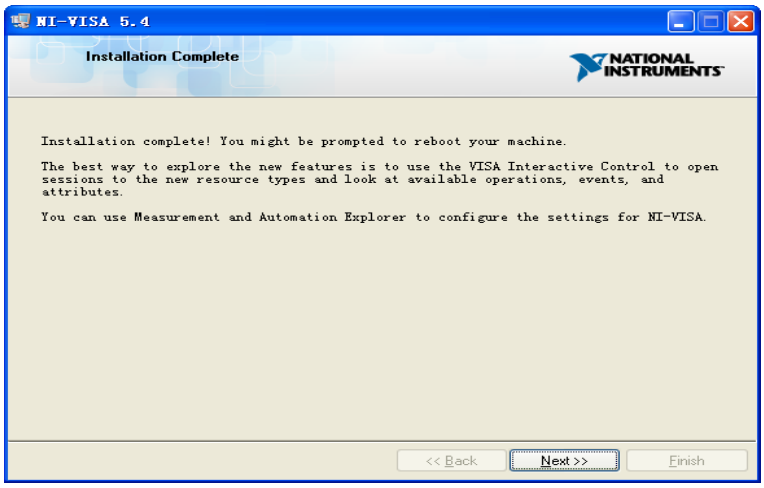
d. Set the install path. The default path is “C:\Program Files\National Instruments\”, you can change it. Click Next, dialog shown as above.



e. Click Next twice, in the License Agreement dialog, select the “I accept the above 2 License Agreement(s).”, and click Next, dialog shown as below:



f. Click Next to begin the installation.

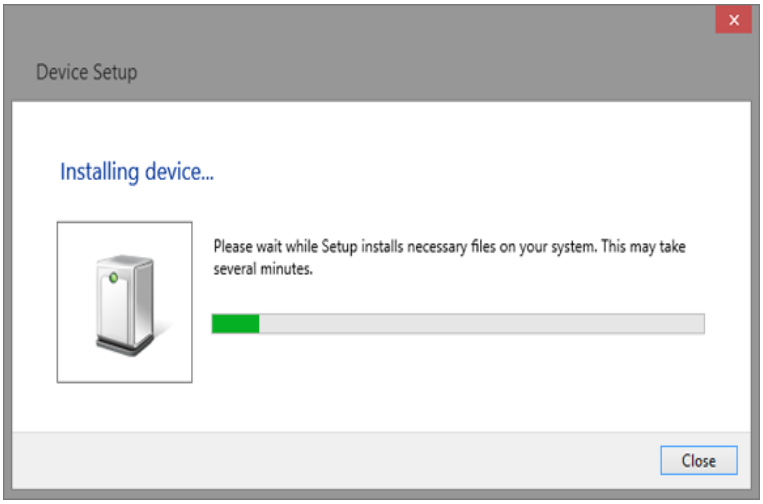


g. Now the installation is complete. Reboot your PC.

1.1.2 Connect the Instrument

Depending on the specific model, your source/measure unit may be able to communicate with a PC through the USB or LAN interface.

Connect the instrument and the USB Host interface of the PC using a USB cable. Assuming your PC is already turned on, turn on your source/measure unit, and then the PC will display the “Device Setup” screen as it automatically installs the device driver as shown below.



Wait for the installation to complete and then proceed to the next step.

1.2 Remote Control

1.2.1 User-defined Programming

Users can use SCPI commands via a computer to program and control the source/measure unit. For details, refer to the introductions below.

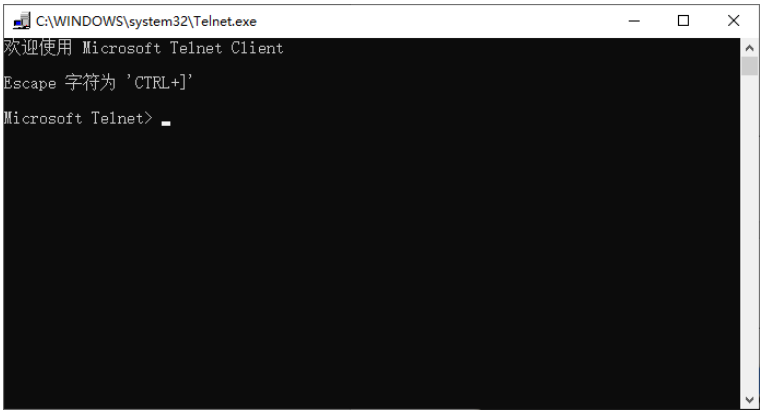
1.2.2 Send SCPI Commands via NI-MAX

NI-Measurement and Automation eXplorer (NI-MAX) is a program created and maintained by National Instruments. It provides a basic remote control interface for VXI, LAN, USB, GPIB, and Serial communications. It is a utility that enables you to send commands one-at-a-time and also retrieve data from connected devices. It is a great tool for troubleshooting and testing command sequences. The source/measure unit can be controlled remotely by sending SCPI commands via NI-MAX.

1.2.3 Using SCPI with Telnet

Telnet provides a means of communicating with the source/measure unit over a LAN connection. The Telnet protocol sends SCPI commands to the source/measure unit from a PC and is similar to communicating with the source/measure unit over USB. It sends and receives information interactively: one command at a time. Windows operating systems use a command prompt style interface for the Telnet client. The steps are as follows:

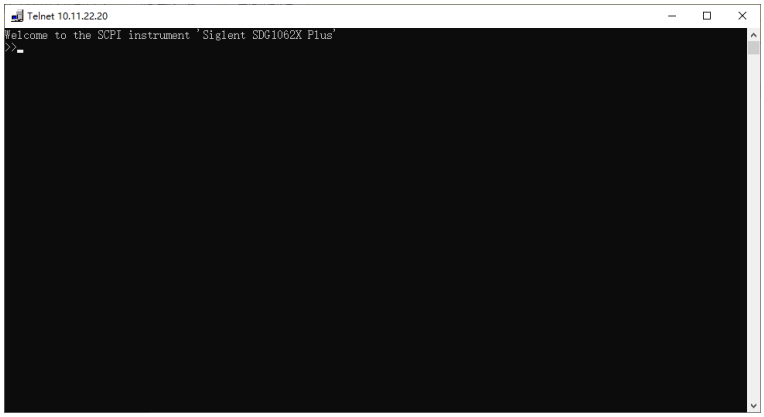
1. On your PC, click Start > All Programs > Accessories > Command Prompt.
2. At the command prompt, type in telnet.
3. Press the Enter key. The Telnet display screen will be displayed.



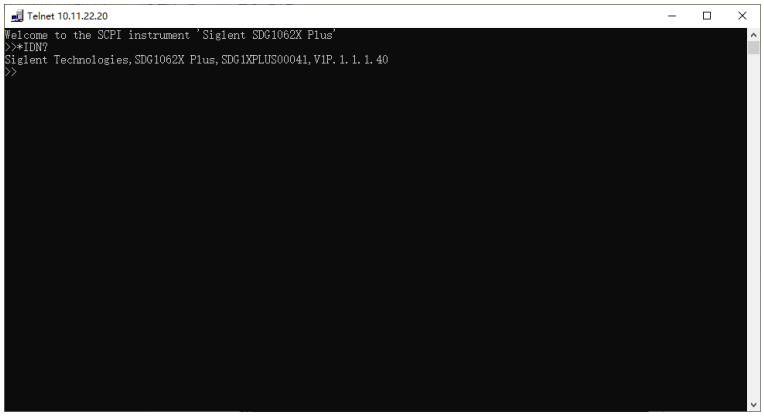
4. At the Telnet command line, type:

   open XXX.XXX.XXX.XXX 5024

Where XXX.XXX.XXX.XXX is the instrument's IP address and 5024 is the port. You should see a response similar to the following:



5. At the SCPI> prompt, input the SCPI commands such as **IDN?* to return the company name, model number, serial number, and firmware version number.



6. To exit the SCPI> session, press the Ctrl+] keys simultaneously.
7. Type *quit* at the prompt or close the Telnet window to close the connection to the instrument and exit Telnet.

1.2.4 Using SCPI with Sockets

Socket API can be used to control the SMM3000X series via LAN without installing any other libraries. This can reduce the complexity of programming.

SOCKET ADDRESS	IP address+port number
IP ADDRESS	SMM IP address
PORT NUMBER	5025

2 Introduction to the SCPI Language

2.1 Command and Query Structure

Commands consist of set commands and query commands (usually called commands and queries). Commands modify source/measure unit settings or tell the source/measure unit to perform a specific action. Queries cause the source/measure unit to return data and status information. Not all commands have both a set and a query form. Some commands have set only and some have query only.

Commands usually start with a colon [:]. A keyword is separated by a colon (:) followed by optional parameter settings. A question mark (?) is added after the command line to indicate that this function is queried. The command keyword is separated from the first parameter by spaces.

2.2 Description

A brief description of the functions performed is given in the description. This is followed by a representation of the formal syntax of the command, where the command header uses case characters and abbreviated forms derived from capitalized characters. Where applicable, the syntax of the query is given in the format of its response.

2.3 Usage

The commands and queries listed in this manual can be used on SMM3000X series instruments.

2.4 Syntax Notation

The following notations are used in the commands:

< > (Angle Brackets)

Angle brackets enclose words that are used as placeholders, of which there are two types: the header path and the data parameter of a command. Parameters are distinguished by enclosing the type name in angle brackets.

:= (Defined As)

A colon followed by an equals sign separates a placeholder from the description of the type and range of values that may be used in a command instead of the placeholder.

{ } (Braces)

Braces or curly brackets are used to enclose one or more parameters that may be included zero or more times. The vertical bar (|) can be read as “or” and is used to separate alternative parameter options.

[ ] (Square Brackets)

Square brackets are used to enclose a keyword that is optional when programming the command; that is, the instrument shall process the command to have the same effect whether the option node is omitted by the programmer or not.

… (Ellipsis)

An ellipsis (trailing dots) indicates that the preceding element may be repeated one or more times.

2.5 Numeric Suffix

Some of the SCPI commands listed in this manual will have the numeric suffix [c], [i], [j], [m] or [n], which usually appear with the command header, and are used to differentiate channels, internal of pins, nodes, limit tests and GPIO,the parameter settings will be automatically set to the integer 1 when the numeric suffix is ignored.

For example, “SOURce:VOLTage 3” is equivalent to “SOURce1:VOLTage 3”, which is to set the voltage 3V for channel 1. The definitions of each digital suffix are as follows:

[c]:used to indicate the selected channel 1 or 2. for dual channel models of the source/measure unit, an integer 1 or 2 can be selected.

[i]: used to indicate the selected internal bus 1 or 2. optional integer 1 or 2.

[j]: used to indicate the selected node 1 or 2. optional integer 1 or 2.

[m]: index 1-12 for the selected limit test. optional integer 1-12.

[n]: used to indicate the selected GPIO pin number 1-12.

2.6 Data Output Format

Data format:

When returning the results of measurement and calculation data, you can return the data in the specified format by selecting the data format (format) by :FORMat [:DATA]

The data format that can be set is as follows:

ASCII, IEEE 754 single precision floating point, IEEE 754 double precision floating point.

When invalid data is returned, the value +9.910000E+37 is returned in ASCII format and NaN is returned in floating point format.

When the returned data is positive infinity, the value +9.900000E+37 is returned in ASCII format and +infinity is returned in floating point format.

When the returned data is a negative infinity, the value -9.900000E+37 is returned in ASCII format and -infinity is returned in floating point format.

Floating point format data can be returned with the specified byte order low-to-high (default) or high-to-low, refer to ":FORMat:BORDer".

Status data (Status):

When returning the results of measurement and calculation data, you can return the status data (Status) by selecting the data format (format), refer to ":FORMat:ELEMents:SENSe" and ":FORMat:ELEMents:CALCulate" , and the status data is returned as a decimal number, which is the weighted sum of the bits of the binary number.

For example, if the source mode is set to current source and is in overvoltage state, Bit0 and Bit3 are set to 1 and value 9 is returned.

The following table lists the states defined by each bit:

Bit	Decimal Number	Description	Defination
0	1	source mode	0 for voltage source, and 1 for current source
1	2	compliance state0	00 means not in compliance state, 01, 10, 11 all means in compliance state (limit value reached)
2	4	compliance state1	
3	8	overvoltage state	1 indicates an overvoltage condition
4	16	overcurrent state	1 indicates an overcurrent condition
5	32	overtemperature state	1 indicates an over-temperature condition
6-12		unused	return to any
13	8192	measurement overrange	1 indicates that the measurement is out of range
14	16384	offset compensation	1 indicates that offset compensation is on
15		unused	return to any
16-20	65536-2031616	composite limit test result	The specific bit definitions for the test results for each of the limit test of NO.1-12 are shown in the following table

The bits of the limit test result for the composite limit test for Sorting mode are defined as follows:

Bit (MSB Bit20-LSB Bit16)	Defination
0 0 0 0 1	NO.1 limit test passed/compliance check failed
0 0 0 1 0	NO.2 limit test passed/compliance check failed
0 0 0 1 1	NO.3 limit test passed/compliance check failed
0 0 1 0 0	NO.4 limit test passed/compliance check failed
0 0 1 0 1	NO.5 limit test passed/compliance check failed
0 0 1 1 0	NO.6 limit test passed/compliance check failed
0 0 1 1 1	NO.7 limit test passed/compliance check failed
0 1 0 0 0	NO.8 limit test passed/compliance check failed
0 1 0 0 1	NO.9 limit test passed/compliance check failed
0 1 0 1 0	NO.10 limit test passed/compliance check failed
0 1 0 1 1	NO.11 limit test passed/compliance check failed
0 1 1 0 0	NO.12 limit test passed/compliance check failed
1 1 1 1 1	All limit tests failed and compliance checks passed

The bits of the limit test result for the composite limit test for Grading mode are defined as follows:

Bit (MSB Bit20-LSB Bit16)	Defination
0 0 0 0 0	All limit tests passed and compliance checks passed
1 0 0 0 1	NO.1 limit test failed due to exceeding the limit
0 0 0 0 1	NO.1 limit test failed due to below lower limit/compliance check failed
1 0 0 1 0	NO.2 limit test failed due to exceeding the limit
0 0 0 1 0	NO.2 limit test failed due to below lower limit/compliance check failed
1 0 0 1 1	NO.3 limit test failed due to exceeding the limit
0 0 0 1 1	NO.3 limit test failed due to below lower limit/compliance check failed
1 0 1 0 0	NO.4 limit test failed due to exceeding the limit
0 0 1 0 0	NO.4 limit test failed due to below lower limit/compliance check failed
1 0 1 0 1	NO.5 limit test failed due to exceeding the limit
0 0 1 0 1	NO.5 limit test failed due to below lower limit/compliance check failed
1 0 1 1 0	NO.6 limit test failed due to exceeding the limit

0 0 1 1 0	NO.6 limit test failed due to below lower limit/compliance check failed
1 0 1 1 1	NO.7 limit test failed due to exceeding the limit
0 0 1 1 1	NO.7 limit test failed due to below lower limit/compliance check failed
1 1 0 0 0	NO.8 limit test failed due to exceeding the limit
0 1 0 0 0	NO.8 limit test failed due to below lower limit/compliance check failed
1 1 0 0 1	NO.9 limit test failed due to exceeding the limit
0 1 0 0 1	NO.9 limit test failed due to below lower limit/compliance check failed
1 1 0 1 0	NO.10 limit test failed due to exceeding the limit
0 1 0 1 0	NO.10 limit test failed due to below lower limit/compliance check failed
1 1 0 1 1	NO.11 limit test failed due to exceeding the limit
0 1 0 1 1	NO.11 limit test failed due to below lower limit/compliance check failed
1 1 1 0 0	NO.12 limit test failed due to exceeding the limit
0 1 1 0 0	NO.12 limit test failed due to below lower limit/compliance check failed

3 Common Commands

3.1 *IDN?

Description	Gets device information string (return content includes: vendor, device model, device serial number, software version number)
Command	*IDN?
Parameters	None
Response Format	Siglent Technologies,model,serial,revision <newline> *model*, device model *serial:* serial number; *revision:* software version number. The response data format is AARD.
Example	*IDN?

3.2 *CAL?

Description	The command executes self-calibration and returns the execution result. Disconnect the measurement terminals before starting self-calibration.
Command	*CAL?
Parameters	None
Response Format	*result* <newline><^END> *result* return 0 or 1 , indicating the self-calibration result. 0:Passed; 1:Failed. The response parameter type is NR1.
Example	*CAL?

3.3 *CLS

Description	Zeroes all event registers and clears the error list
Command	*CLS
Parameters	None
Response Format	None
Example	*CLS

3.4 *ESE

Description	Sets the enable value of the standard event status register
Command	*ESE enable_number *ESE?
Parameters	*enable_number* Decimal value that is the sum of the binary weighted values of the desired bit, hex, octal, or binary value. The parameter data type is NR1 or NDN.
Response Format	*enable_number*<newline> *enable_number* is the sum of the binary weighted values of the enable register bits. The return format can be selected with the command ":format:SREGister". The response data type is NR1 or NDN.
Example	*ESE 16 *ESE?

3.5 *ESR?

Description	Queries and clears the event value of the standard event status registers
Command	*ESR?
Parameters	None
Response Format	*register*<newline> *register* is the binary weighted sum of all bits set in the register. The response data type is NR1.
Example	*ESR?

3.6 *OPC

Description	The OPC bit (bit0) is set to 1 in the standard event status register after all operations are completed
Command	*OPC *OPC?
Parameters	None
Response Format	1 <newline> The query returns 1 if the instrument has completed all operations that sent

	before. The response data type is NR1.
Example	*OPC *OPC?

3.7 *RCL

Description	This command restores the instrument to previously state stored in one of the storage locations by using the command *SAV.
Command	*RCL memory
Parameters	*memory* storage locations 0 to 9. The parameter type is NR1.
Response Format	None
Example	*RCL 0

3.8 *RST

Description	Resets the instrument
Command	*RST
Parameters	None
Response Format	None
Example	*RST

3.9 *SAV

Description	Stores the current state of the instrument to a specified location in non-volatile memory.
Command	*SAV *memory*
Parameters	*memory* storage locations 0 to 9. The parameter type is NR1.
Response Format	None
Example	*SAV 1

3.10 *SRE

Description	Sets the enable value of the status byte register
Command	*SRE *value* *SRE?
Parameters	*value* decimal value corresponding to the binary weighted sum of the bits in the register The data type is NRf.
Response Format	*value* <newline> *value* is the binary weighted sum of all bits set in the register. The response data type is NR1.
Example	*SRE 24 *SRE?

3.11 *STB?

Description	Queries the event value of the status byte register
Command	*STB?
Parameters	None
Response Format	None
Example	*STB?

3.12 *TRG

Description	The command generates a trigger signal when the trigger system selects “Bus” as its source. It is as same as the Group Execution Trigger (GET) command when triggered.
Command	*TRG
Parameters	None
Response Format	None
Example	*TRG

3.13 *TST?

Description	Queries instrument self-test results
Command	*TST?
Parameters	None
Response Format	None
Example	*TST?

3.14 *WAI

Description	Waits for all outstanding operations to complete and then executes the other commands
Command	*WAI
Parameters	None
Response Format	None
Example	*WAI

4 Subsystem Commands

4.1 CALCulate Subsystem

4.1.1 :CALCulate:CLIMits:CLEar:AUTO

Description	Enables or disables the automatic clearing of composite limit tests.
Command	:CALCulate[c]:CLIMits:CLEar:AUTO *mode* :CALCulate[c]:CLIMits:CLEar:AUTO?
Parameters	*mode* 0|OFF|1|ON (default). The parameter type is boolean. Enable the function, and command ":INITiate" will automatically clear the composite limit test results and ports (GPIOs) Disable the function, and manual clearing must be executed before starting next composite limit test. Refer to ":INITiate[:IMMediate]<:ACQuire|:TRANsient|[:ALL]>". Use the command ":CALCulate:CLIMits:CLEar[:IMMediate]" to clear immediately if needed.
Response Format	*mode*<newline> *mode* returns 0 or 1, indicating that the auto clear function is off or on, respectively. The response data type is NR1.
Example	:CALC:CLIM:CLE:AUTO 1 :CALC2:CLIM:CLE:AUTO?

4.1.2 :CALCulate:CLIMits:CLEar:AUTO:DELay

Description	Sets the delay time for automatically clearing the composite limit test Refer to ":CALCulate:CLIMits:CLEar:AUTO" Command. The delay time is defined as the time between the automatic clearing and the measurement operation.
Command	:CALCulate[c]:CLIMits:CLEar:AUTO:DELay *time* :CALCulate[c]:CLIMits:CLEar:AUTO:DELay? [*time*]
Parameters	*time* value (+1E-5 to 60 seconds)|MINimum|MAXimum|DEFault (Default is +1E-4). The parameter type is NRf+. If the specified value is less than MIN or more than MAX, time is automatically set to MIN or MAX.

Response Format	*time* <newline> *time* returns the delay time set. If a parameter (DEFault | MINimum | MAXimum) is specified, time returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:CALC:CLIM:CLE:AUTO:DEL 1E-3 :CALC2:CLIM:CLE:AUTO:DEL?

4.1.3 :CALCulate:CLIMits:CLEar[:IMMediate]

Description	Immediately clears composite limit test results and GPIO ports.
Command	:CALCulate[*c*]:CLIMits:CLEar[:IMMediate]
Parameters	None
Response Format	None
Example	:CALC:CLIM:CLE:IMM :CALC2:CLIM:CLE

4.1.4 :CALCulate:CLIMits:<FAIL|PASS>:DIGital[:DATA]

Description	Defines the (FAIL/PASS) bit pattern used to indicate the result of a composite limit test.
Command	:CALCulate[c]:CLIMits:<FAIL|PASS>:DIGital[:DATA] bit_*pattern* :CALCulate[c]:CLIMits:<FAIL|PASS>:DIGital[:DATA]? <FAIL|PASS>:PASS is used to specify the pass pattern, and FAIL is used to specify the fail pattern
Parameters	*bit_pattern* 0 (Default is decimal) to 63. The parameters data type is NR1 or NDN. The data format should match with the setting by command ":FORMat:DIGital".
Response Format	*bit_pattern* <newline> *bit_pattern* The data format should match with the setting by command ":FORMat:DIGital". The parameters data type is NR1 or NDN.
Example	:CALC:CLIM:FAIL:DIG:DATA 63 :CALC2:CLIM:PASS:DIG?

4.1.5 :CALCulate:CLIMits:MODE

Description	Sets the mode of operation for the composite limit test to Grading or Sorting. The limit test includes one of limit value test and compliance check, and it is to judge whether they pass or fail.
Command	:CALCulate[c]:CLIMits:MODE *mode* :CALCulate[c]:CLIMits:MODE?
Parameters	*mode* SORT (Sorting)|GRAD (Default is Grading). The parameter type is CPD. *mode* = GRAD Performs up to 12 test limit tests and stops when a limit test failed. See the flowchart in user manual. *mode* = SORT Performs up to 12 test limit tests and stops when a limit test passed. See the flowchart in user manual.
Response Format	*mode* <newline> *mode* returns GRAD or SORT. The response data type is CPD.
Example	:CALC:CLIM:MODE SORT :CALC2:CLIM:MODE?

4.1.6 :CALCulate:CLIMits:STATe

Description	Enables or disables a composite limit test
Command	:CALCulate[c]:CLIMits:STATe *mode* :CALCulate[c]:CLIMits:STATe?
Parameters	*mode* 1|ON (default)|0|OFF. The parameter type is boolean.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that composite limit test function is on or off, respectively. The response data type is NR1 or NDN.
Example	:CALC:CLIM:STAT 1 :CALC2:CLIM:STAT?

4.1.7 :CALCulate:CLIMits:STATe:ANY?

Description	Checks if the current composite limit test contains a limit test/compliance check.
Command	:CALCulate[c]:CLIMits:STATe:ANY?
Parameters	None
Response Format	*status* <newline> *status* returns 0 or 1, indicating that there is no limit test or one limit test exists at least, respectively. The response data type is NR1.
Example	:CALC2:CLIM:STAT:ANY?

4.1.8 :CALCulate:CLIMits:UPDate

Description	Enables or disables immediate results output or update only when in Grading mode. See "Update = IMMEDIATE?" step in the Flowchart of the Composite Limit Test under the Grading Mode in user manual. When enabled, the result output/update will be performed immediately if a limit test/compliance check fails or all limit tests pass. The pass/fail bit pattern of the test result to be returned is define by :CALCulate:CLIMits:<FAIL|PASS>:DIGital[:DATA]. Pass pattern will be output only if all tests pass.
Command	:CALCulate[c]:CLIMits:UPDate *result* :CALCulate[c]:CLIMits:UPDate?
Parameters	*result* END|IMMediate (default). The parameter type is CPD. *result* = IMM to enable the immediate output function *result* = END to disable the immediate output function
Response Format	result <newline> result returns IMM or END. The response data type is CRD.
Example	:CALC:CLIM:UPD END :CALC2:CLIM:UPD?

4.1.9 :CALCulate:DATA?

Description	Returns limit test result. Specify the type of element of result by :FORMat:ELEMents:CALCulate. Data of limit test result can be expressed by the following equations: *limit test data = input data - null offset* *input data*: set by :CALCulate:FEED *null offset*: set by :CALCulate:OFFSet or :CALCulate:OFFSet:ACQuire If offset function is off by :CALCulate:OFFSet:STATe, null offset = 0.
Command	:CALCulate[c]:DATA? [*offset*[, *size*]]
Parameters	*offset* n|CURRent|STARt (default). Indicates the location of the first data to be received. The location will change when the command is executed. The parameter type is NR1 or CPD. *offset* = n indicates to start receiving from the (n+1)th data. n is an integer from 0 to the buffer size. *offset* = CURR indicates that the current data position is the starting position. *offset* = STAR indicates to start at the top of the data buffer, which is as same as offset =0. *size* chooses the number of data to be received from 1 to buffer size. The parameter type is NR1. If size is not entered, all data of limit test result from offset will be returned.
Response Format	*data* <newline> The response data type is NR3. Refer to "Data Output Format".
Example	:CALC2:DATA? 0,10

4.1.10 :CALCulate:DATA:LATest?

Description	Returns the latest limit test result. Specify the type of element of result by :FORMat:ELEMents:CALCulate. Data of limit test result can be expressed by the following equations: *limit test data = input data - null offset* *input data*: set by :CALCulate:FEED *null offset*: set by :CALCulate:OFFSet or :CALCulate:OFFSet:ACQuire If offset function is off by :CALCulate:OFFSet:STATe, null offset = 0.
Command	:CALCulate[c]:DATA:LATest?

Parameters	None
Response Format	data <newline> The response data type is NR3. Refer to "Data Output Format".
Example	:CALC2:DATA:LAT?

4.1.11 :CALCulate:DIGital:BIT

Description	Set the GPIO pins for composite limit test result output. The pass/fail bit pattern of the test result to be returned is define by :CALCulate:CLIMits:<FAIL|PASS>:DIGital[:DATA]. Pass pattern will be output only if all tests pass.
Command	:CALCulate[c]:DIGital:BIT *pin* :CALCulate[c]:DIGital:BIT?
Parameters	*pin* EXT*n*|NONE (default). The parameter type is CPD. *pin* = EXT*n* is on the output port of the Digital I/O D-sub connector on the rear panel, from 1 to 6. *pin* = NONE no port used for output. The GPIO ports entered should be separated by a comma and continuous number from small to large, like EXTn,EXTn+1,.... and EXTn. For example: *pin* = EXT6 to set GPIO pin6 for BIT0(LSB) of limit test result output *pin* = EXT2,EXT3,EXT4 to set GPIO pin2,3,4 for limit test result output . EXT2, EXT3, and EXT4 are assigned to BIT0 (LSB), BIT1, and BIT2, respectively. *pin* = EXT2,EXT3,EXT6 or EXT4,EXT3,EXT2 is invalid, because they are not continuous number from small to large.
Response Format	*pin* <newline> *pin* returns NONE or a string separated by a comma. The response data type is CRD.
Example	:CALC:DIG:BIT EXT2,EXT3,EXT4 :CALC2:DIG:BIT?

4.1.12 :CALCulate:DIGital:<BUSY|EOT|SOT>

Description	Sets GPIO pin for BUSY, EOT (End of test) or SOT (Start of test) signal output.

Command	:CALCulate[c]:DIGital:<BUSY|EOT|SOT> *pin* :CALCulate[c]:DIGital:<BUSY|EOT|SOT>? <BUSY|EOT|SOT>: BUSY indicates that limit test is busy in working, EOT indicates that the limit test starts, SOT indicates that the limit test ends.
Parameters	*pin* EXTn|NONE (default). The parameter type is CPD. EXTn is on the output port of the Digital I/O D-sub connector on the rear panel, n= 1 to 6 is selectable for BUSY and EOT which are output ports, and n=7 to 12 is selectable for SOT which are input ports. *pin* = EXT2 set GPIO pin2 for BUSY/EOT. *pin* = NONE no port used for output.
Response Format	*pin* <newline> *pin* returns the NONE or a string EXTn. The response data type is CRD.
Example	:CALC:DIG:EOT EXT2 :CALC2:DIG:SOT?

4.1.13 :CALCulate:FEED

Description	Specifies the type of test data for composite limit test. The limit test includes one of limit value test and compliance check, and it is to judge whether they pass or fail. limit test result can be returned by :CALCulate:DATA? or :CALCulate:DATA:LATest?.
Command	:CALCulate[c]:FEED *type* :CALCulate[c]:FEED?
Parameters	type type of limit test data. MATH|RESistance|CURRent|VOLTage (default).The parameters type is CPD. type = VOLT indicates voltage measurement data. type = CURR indicates current measurement data. type = RES indicates resistance measurement data calculated by the equation below: Resistance = Vmeas/Imeas where Vmeas indicates voltage measurement data, and Imeas indicates current measurement data. type = MATH indicates math result calculated by the math expression. The math expression can be chosen by :CALCulate:MATH[:EXPRession]:NAME or defined by :CALCulate:MATH[:EXPRession][:DEFine].

Response Format	*type* <newline> *type* returns MATH,RES,CURR or VOLT. The response data type is CRD.
Example	:CALC:FEED MATH :CALC2:FEED?

4.1.14 :CALCulate:LIMit:COMPliance:DIGital[:DATA]

Description	Defines the fail bit pattern to indicate that the status of NO.m limit test is failure.
Command	:CALCulate[c]:LIMit[m]:COMPliance:DIGital[:DATA] *bit_pattern* :CALCulate[c]:LIMit[m]:COMPliance:DIGital[:DATA]?
Parameters	*bit_pattern* 0 (Default decimal) to 63. The parameter type is NR1 or NDN.
Response Format	*bit_pattern* <newline> *bit_pattern* returns the fail bit pattern. The data format should match with the setting by command ":FORMat:DIGital". The response data type is NR1 or NDN.
Example	:CALC:LIM:COMP:DIG:DATA 63 :CALC2:LIM12:COMP:DIG?

4.1.15 :CALCulate:LIMit:COMPliance:FAIL

Description	Sets the pass/fail judgement of NO.m compliance check.
Command	:CALCulate[c]:LIMit[m]:COMPliance:FAIL *criteria* :CALCulate[c]:LIMit[m]:COMPliance:FAIL?
Parameters	*criteria* OUT|IN (default). The parameter type is CPD. *criteria* = IN indicates that the compliance check fails if the channel is in compliance state. *criteria* = OUT indicates that the compliance check fails if the channel is not in compliance state.
Response Format	*criteria* <newline> *criteria* returns IN or OUT. The response data type is CRD.

Example	:CALC:LIM:COMP:FAIL OUT :CALC2:LIM12:COMP:FAIL?

4.1.16 :CALCulate:LIMit:FAIL?

Description	Returns the NO.m limit test result.
Command	:CALCulate[c]:LIMit[*m*]:FAIL?
Parameters	None
Response Format	*result* <newline> *result* returns 0 or 1. The response data type is NR1. 0 means test passed while 1 means test failed.
Example	:CALC2:LIM12:FAIL?

4.1.17 :CALCulate:LIMit:FUNCtion

Description	Sets the test type of NO.m limit test to limit value test/compliance check.
Command	:CALCulate[c]:LIMit[m]:FUNCtion type :CALCulate[c]:LIMit[m]:FUNCtion?
Parameters	*type* COMPliance|LIMit (default). The parameter type is CPD. *type* = COMP to set NO.m limit test to compliance check to check if the channel is in compliance state when testing. *type* = LIM to set NO.m limit test to limit value test to check if the test data is between the set upper limit value and lower limit value.
Response Format	*type* <newline> *type* returns COMP or LIM. The response data type is CRD.
Example	:CALC:LIM:FUNC COMP :CALC2:LIM12:FUNC?

4.1.18 :CALCulate:LIMit:<LOWer|UPPer>

Description	Sets the upper/lower limit value of the NO.m limit value test.
Command	:CALCulate[c]:LIMit[m]:<LOWer|UPPer> *limit*

	:CALCulate[c]:LIMit[m]:<LOWer|UPPer>? [*limit*] <LOWer|UPPer>: means lower limit value | upper limit value
Parameters	*limit* value (-9.999999E+20 to +9.999999E+20)|MINimum| MAXimum| DEFault (upper limit value is +1 and lower limit value is -1). The parameter type is NRf+.
Response Format	*limit* <newline> *limit* returns the upper/lower limit value of NO.m limit value test. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:CALC:LIM:LOW -2.5 :CALC2:LIM12:UPP?

4.1.19 :CALCulate:LIMit:<LOWer|UPPer>:DIGital[:DATA]

Description	Defines the fail bit pattern of exceeding the lower limit or upper limit of NO.m limit test only used in Grading mode.
Command	:CALCulate[c]:LIMit[m]:<LOWer|UPPer>:DIGital[:DATA] *bit_pattern* :CALCulate[c]:LIMit[m]:<LOWer|UPPer>:DIGital[:DATA]? <LOWer|UPPer>: LOWer means exceeding the lower limit, and UPPer means exceeding the upper limit.
Parameters	*bit_pattern* 0 (Default decimal) to 63. The data format should match with the setting by command ":FORMat:DIGital". The parameter type is NR1 or NDN.
Response Format	*bit_pattern* <newline> *bit_pattern* returns the set fail bit pattern. The response data type is NR1or NDN.
Example	:CALC:LIM:LOW:DIG:DATA 63 :CALC2:LIM12:UPP:DIG?

4.1.20 :CALCulate:LIMit:PASS:DIGital[:DATA]

Description	Defines the pass bit pattern of NO.m limit value test only used in Sorting mode.
Command	:CALCulate[c]:LIMit[m]:PASS:DIGital[:DATA] *bit_pattern*

	:CALCulate[c]:LIMit[m]:PASS:DIGital[:DATA]?
Parameters	*bit_pattern* 0 (Default decimal) to 63. The parameter type is NR1 or NDN.
Response Format	*bit_pattern* <newline> *bit_pattern* returns the set pass bit pattern. The data format should match with the setting by command ":FORMat:DIGital". The response data type is NR1 or NDN.
Example	:CALC:LIM:PASS:DIG:DATA 32 :CALC2:LIM12:PASS:DIG?

4.1.21 CALCulate:LIMit:STATe

Description	Enables or disables the NO.m limit test.
Command	:CALCulate[c]:LIMit[*m*]:STATe *mode* :CALCulate[c]:LIMit[*m*]:STATe?
Parameters	*mode* 1|ON|0|OFF (default). The parameter type is boolean. *mode* = 1 or ON enable the NO.m limit test. *mode* = 0 or OFF disable the NO.m limit test.
Response Format	*mode* <newline> *mode*is0 or 1, indicating that the NO.m limit test is off or on, respectively. The response data type is NR1.
Example	:CALC:LIM:STAT 1 :CALC2:LIM12:STAT?

4.1.22 :CALCulate:MATH:DATA?

Description	Returns math result calculated by math expression. Specify the type of element of result by :FORMat:ELEMents:CALCulate. Math expresson is defined by :CALCulate:MATH[:EXPRession]:NAME and :CALCulate:MATH[:EXPRession][:DEFine].
Command	:CALCulate[*c*]:MATH:DATA? [*offset*[, *size*]]
Parameters	*offset* n|CURRent|STARt (default). Indicates the location of the first data to be received. The location will change when the command is executed. The parameter type is NR1 or CPD.

	offset = n indicates to start receiving from the (n+1)th data. n is an integer from 0 to the buffer size. *offset* = CURR indicates that the current data position is the starting position. *offset* = STAR indicates to start at the top of the data buffer, which is as same as offset =0. *size* chooses the number of data to be received from 1 to buffer size. The parameter type is NR1. If size is not entered, all data of limit test result from offset will be returned.
Response Format	*data* <newline> The response data type is NR3. Refer to "Data Output Format".
Example	:CALC2:MATH:DATA? 0,10
Remarks	If the math expression contains more than one measurement, multiple measurements are required to get the result. For example: OFFCOMPOHM = (VOLT[c][1]-VOLT[c][0]) / (CURR[c][1]-CURR[c][0]) Two measurements must be performed to obtain the result of OFFCOMPOHM above.

4.1.23 :CALCulate:MATH:DATA:LATest?

Description	Returns the latest math result calculated by math expression. Specify the type of element of result by :FORMat:ELEMents:CALCulate. Math expresson is defined by :CALCulate:MATH[:EXPRession]:NAME and :CALCulate:MATH[:EXPRession][:DEFine].
Command	:CALCulate[c]:MATH:DATA:LATest?
Parameters	None
Response Format	*data* <newline> The response data type is NR3. Refer to "Data Output Format".
Example	:CALC2:MATH:DATA:LAT?

4.1.24 :CALCulate:MATH[:EXPRession]:CATalog?

Description	Returns a list including all default and user- defined math expressions.
Command	:CALCulate[c]:MATH[:EXPRession]:CATalog?
Parameters	None
Response Format	*catalog* <newline> *catalog* returns all default and user- defined math expressions. The response data type is AARD. Default math expressions contain POWER,OFFCOMPOHM,VOLTCOEF, and VARALPHA catalog will return “POWER”,“OFFCOMPOHM”,“VOLTCOEF”,“VARALPHA” and names of user- defined math expressions.
Example	:CALC:MATH:EXPR:CAT?

4.1.25 :CALCulate:MATH[:EXPRession][:DEFine]

Description	Defines a math expression, that is user-defined For valid resources for this expression and default math expressions, refer to “Resources Used in Expressions”. Before executing this command, you must create a name of the new math expression by :CALCulate:MATH[:EXPRession]:NAME
Command	:CALCulate[c]:MATH[:EXPRession][:DEFine] *definition* :CALCulate[c]:MATH[:EXPRession][:DEFine]?
Parameters	*definition* math expression defined has up to 256 ASCII characters. The parameter type is SPD. The math expresson that is user-defined should be surrounded with brackets. For example, definition = ((CURR[1]-CURR[0])*(RES[1]-RES[0])). Up to 32 math expressions can be defined, including default math expressions.
Response Format	*definition* <newline> *definition* returns the definition of the currently selected math expression. For example, definition returns “((CURR[1]-CURR[0])*(RES[1]-RES[0]))”. The response data type is Expr.
Example	:CALC:MATH:EXPR:NAME “Expression_for_ch1” :CALC:MATH:EXPR:DEF “((CURR[1]-CURR[0])*(RES[1]-RES[0]))” :CALC:MATH?

4.1.25.1 Resources Used in Expressions

Reserved variable: measurement data listed in Table 4-1, including scalar variable and vector variable

Scalar variable: measurenment data only with channel number

Vector (array) variable: measurement data for sweep with channel number and serial number

Math operators: math operators including operators, basic functions and numberic value:

Operators: +,-,*, /, ^, refer to Table 4-2.

Basic functions: ln,log,sin,cos,tan,exp

Numeric value:

Decimal (0 to 4294967294, 4294967295 indicates -1);

Binary (32 bit, 0 to 0b11111111111111111111111111111111);

Hexadecimal (0 to 0xFFFFFFFF).

Note: When using functions log and ln, the specified value can be negative, and if so the negative value will be converted to absolute value to be calculated in functions. For example, log (-10) = log (10) = 1.

Reserved variable

Table 4-1

Scalar	Vector	Description
SOUR[c]	SOUR[c][]	Source output setting data
VOLT[c]	VOLT[c][]	Voltage measurement data
CURR[c]	CURR[c][]	Current measurement data
RES[c]	RES[c][]	Resistance measurement data
TIME[c]	TIME[c][]	Timestamp data

Math operators

Table 4-2

Priority	Operator	Description
High : :	( )	Brackets
	+ , -	Unary addition and unary subtraction operators

: : : Low	^	Exponentiation operator
	* , /	Multiplication and division operators
	+ , -	Addition and subtraction operators

Default math expressions: The math expressions will not be cleared or changed when executing power-on or reset operations, including formulas below:

- POWER
- OFFCOMPOHM
- VARALPHA
- VOLTCOEF

POWER use the formula below to calculate power:

POWER = VOLT[c] * CURR[c]

OFFCOMPOHM use the formula below to calculate offset compensation resistance which can effectively reduce measurement errors in low resistance measurement:

OFFCOMPOHM = (VOLT[c][1]-VOLT[c][0]) / (CURR[c][1]-CURR[c][0])

where VOLT[c][0] and CURR[c][0] are the measurenment data with current output level, and VOLT[c][1] and CURR[c][1] are the measurement data with different current output levels or zero output.

VARALPHA use the formula below to calculate varistor alpha:

VARALPHA = log(CURR[c][1] / CURR[c][0]) / log(VOLT[c][1] / VOLT[c][0])

where CURR[c][0] and VOLT[c][0] are measurement data at one point on the nonlinear I-V characteristic curve of the varistor, and CURR[c][1] and VOLT[c][1] are data at another point.

VOLTCOE use the formula below to calculate voltage coefficient which is the ratio of the fractional change in resistance as the resistance changes with voltage:

VOLTCOEF = (RES[c][1]-RES[c][0]) / (RES[c][1] * (VOLT[c][1]-VOLT[c][0])) * 100 %

where RES[c][0] and VOLT[c][0] are measurement data at the first point, and RES[c][1] and VOLT[c][1] are measurement data at the second point.

4.1.26 :CALCulate:MATH[:EXPRession]:DELete:ALL

Description	Deletes all user-defined math expressions which do not contain default math expressions.
Command	:CALCulate[c]:MATH[:EXPRession]:DELete:ALL
Parameters	None
Response Format	None
Example	:CALC2:MATH:EXPR:DEL:ALL

4.1.27 :CALCulate:MATH[:EXPRession]:DELete[:SELected]

Description	Deletes a user-defined math expression and default math expressions can not be chosen.
Command	:CALCulate[c]:MATH[:EXPRession]:DELete[:SELected] name
Parameters	None
Response Format	name The name of the math expression to be deleted is consist of up to 32 ASCII characters. The parameter type is SPD.
Example	:CALC2:MATH:EXPR:DEL:SEL “TempExpression1”

4.1.28 :CALCulate:MATH[:EXPRession]:NAME

Description	Selects the math expression to be used for the calculation. If the math expression does not exist, the command will create a new one, and you can add a new user-defined math expression by :CALCulate:MATH[:EXPRession][:DEFine]. If the math expression exists, you can also edit the expression by :CALCulate:MATH[:EXPRession][:DEFine]. For valid resources for this expression and default math expressions, refer to “Resources Used in Expressions”.
Command	:CALCulate[c]:MATH[:EXPRession]:NAME *name* :CALCulate[c]:MATH[:EXPRession]:NAME?
Parameters	*name* The name of the math expression to be deleted is consist of up to 32 ASCII characters without any control characters, space characters, single and double quotes, and commas.

	The parameter type is SPD.
Response Format	*name* <newline> *name* returns the chosen math expression. For example, name returns "Expression_for_ch2". The response data type is SRD.
Example	:CALC2:MATH:EXPR:NAME "Expression_for_ch2" :CALC2:MATH:NAME?

4.1.29 :CALCulate:MATH:STATe

Description	Enables or disables the math function.
Command	:CALCulate[*c*]:MATH:STATe *mode* :CALCulate[*c*]:MATH:STATe?
Parameters	*mode* 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 1 or ON to enable math function. *mode* = 0 or OFF to disable math function.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the math functon is off or on, respectively. The response data type is NR1.
Example	:CALC:MATH:STAT 1 :CALC2:MATH:STAT?

4.1.30 :CALCulate:MATH:UNITs

Description	Defines the name of the unit of a math expression.
Command	:CALCulate[c]:MATH:UNITs *unit* :CALCulate[c]:MATH:UNITs?
Parameters	*unit* Up to 32 ASCII characters.The parameter type is SPD.
Response Format	*unit* <newline> *unit* returns the name of the unit of the math expression. The response data type is SRD.
Example	:CALC:MATH:UNIT "amps" :CALC2:MATH:UNIT?

4.1.31 :CALCulate:OFFSet

Description	Sets the offset value of limit test. Enable the offset function by :CALCulate:OFFSet:STATe before executing the command.
Command	:CALCulate[c]:OFFSet *offset* :CALCulate[c]:OFFSet? [*offset*]
Parameters	*offset* value (-9.999999E+20 to +9.999999E+20)|MINimum|MAXimum|DEFault (Default is 0.0). The parameter type is NRf+.
Response Format	*offset* <newline> *offset* returns the set offset value. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:CALC:OFFS 0.5 :CALC2:OFFS?

4.1.32 :CALCulate:OFFSet:ACQuire

Description	Automatically sets the offset value of limit test. The offset value will be the latest data read by :CALCulate:DATA:LATest? or :SENSe:DATA:LATest? If available value does not exist, offset value is set to 0.0.
Command	:CALCulate[c]:OFFSet:ACQuire
Parameters	None
Response Format	None
Example	:CALC:OFFS:ACQ :CALC2:OFFS:ACQ

4.1.33 :CALCulate:OFFSet:STATe

Description	Enables or disables the offset value function. The offset value can be set by :CALCulate:OFFSet or :CALCulate:OFFSet:ACQuire.
Command	:CALCulate[c]:OFFSet:STATe *mode*

	:CALCulate[c]:OFFSet:STATe?
Parameters	*mode* 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 1or ON to enable offset function. *mode* = 0 or OFF to disable offset function.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that offset function is off or on, respectively. The response data type is NR1.
Example	:CALC:OFFS:STAT 1 :CALC2:OFFS:STAT?

4.2 DISPlay Subsystem

4.2.1 :DISPlay:DIGits

Description	Sets the display resolution of the measurement readings displayed on the front panel
Command	:DISPlay:DIGits *digits* :DISPlay:DIGits? [MINimum|MAXimum|DEFault]
Parameters	*digits* resolution. value (4 to 7)|MINimum|MAXimum|DEFault (Default is 7). The parameter type is NRf+. *digits* = 4 selects 3½ digit display resolution. *digits* = 5 selects 4½ digit display resolution. *digits* = 6 selects 5½ digit display resolution. *digits* = 7 selects 6½ digit display resolution.
Response Format	*digits* <newline> *digits* returns the set display resolution. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR1.
Example	:DISP:DIG 4 :DISP:DIG? MAX

4.3 FETCh Subsystem

4.3.1 :FETCh:ARRay?

Description	Returns the saved measurement data . The data types are set by :FORMat:ELEMents:SENSe including voltage measurement data, current measurement data, resistance measurement data, time data, status data, or source output setting data. The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:FETCh:ARRay? [*chanlist*]
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the data set by :FORMat:ELEMents:SENSe The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr1, ch1sour1,ch2curr1,ch2sour1,* *ch1curr2, ch1sour2,ch2curr2,ch2sour2,……* *ch1curr5,ch1sour5,ch2curr5,ch2sour5,* *ch1curr6,ch1sour6,*+9.910000E+37,+9.910000E+37,…… *ch1curr10,ch1sour10,*+9.910000E+37,+9.910000E+37 The example contains current mesurement data (ch1currN) and source output setting data (ch1sourN) of 10 steps sweep measurement of channel 1, as well as current mesurement data (ch2currN) and source output setting data (ch2sourN) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FORM:ELEM:SENS CURR,SOUR :FETC:ARR? (@1,2)

4.3.2 :FETCh:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

Description	Returns the saved measurement data that is specified by the parameter, including current measurement data, resistance measurement data, source output setting data, status data, timestamp data, or voltage measurement data. The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:FETCh:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>? [*chanlist*] CURRent specifies the current measurement data; RESistance specifies the resistance measurement data; SOURce specifies the source output setting data; STATus specifies the status data; TIME specifies the timestamp data; VOLTage specifies the voltage measurement data.
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr1,ch2curr1,ch1curr2,ch2curr2,.....* *ch1curr5,ch2curr5,ch1curr6,*+9.910000E+37*, ...* *ch1curr10,*+9.910000E+37 The example contains current mesurement data (ch1currN) of 10 steps sweep measurement of channel 1, as well as current mesurement data (ch2currN) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FETC:ARR:CURR? (@2,1)

4.3.3 :FETCh[:SCALar]?

Description	Returns the latest saved measurement data . The data types are set by :FORMat:ELEMents:SENSe including voltage measurement data, current measurement data, resistance measurement data, time data, status data, or source output setting data. The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:FETCh[:SCALar]? [*chanlist*]
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the latest data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr10,ch1sour10,ch2curr5,ch2sour5* The example contains the latest current mesurement data (ch1curr10) and the latest source output setting data (ch1sour10) of 10 steps sweep measurement of channel 1, as well as the latest current mesurement data (ch2curr5) and the latest source output setting data (ch2sour5) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FORM:ELEM:SENS CURR,SOUR :FETC? (@1,2)

4.3.4 :FETCh[:SCALar]:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

Description	Returns the latest saved measurement data that is specified by the parameter, including current measurement data, resistance measurement data, source output setting data, status data, timestamp data, or voltage

	measurement data. The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:FETCh[:SCALar]:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>? [*chanlist*] CURRent specifies the current measurement data; RESistance specifies the resistance measurement data; SOURce specifies the source output setting data; STATus specifies the status data; TIME specifies the timestamp data; VOLTage specifies the voltage measurement data.
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the latest data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr10,ch2curr5* The example contains the latest current mesurement data (ch1curr10) of 10 steps sweep measurement of channel 1, as well as the latest current mesurement data (ch2curr5) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FETC:CURR? (@2,1)

4.4 FORMat Subsystem

4.4.1 :FORMat:BORDer

Description	This command sets the byte order of the binary output data and is valid when the data output format is set to IEEE-754 binary format by the command :FORMat[:DATA].
Command	:FORMat:BORDer *byte_order* :FORMat:BORDer?
Parameters	*byte_order* NORMal (default)|SWAPped. The parameter type is CPD. *byte_order* = NORMal sets the normal byte order. For IEEE-754 single-precision format, the send order is from byte1 to byte4. For IEEE-754 double-precision format, the send order is from byte1 to byte8. *byte_order* = SWAPped sets the reverse byte order. For IEEE-754 single-precision format, the order of sending is from byte4 to byte1. For IEEE-754 double-precision format, the order of sending is from byte8 to byte1.
Response Format	*byte_order*<newline> *byte_order* returnsNORM or SWAP. The response data type is CRD.
Example	:FORM:BORD SWAP :FORM:BORD?

4.4.2 :FORMat[:DATA]

Description	Sets the data output format. Refer to "Data Output Format".
Command	:FORMat [:DATA] *format* :FORMat [:DATA]?
Parameters	*format* Data output format. The parameter type is CPD. ASCii|REAL,32|REAL,64.REAL,64 only applies to the default language mode set by :SYST:LANG "DEF". *format* = ASCii Specifies the ASCII format (default). *format* = REAL,32 Specifies IEEE-754 single precision format. 4 bytes of data. *format* = REAL,64 Specifies IEEE-754 double precision format. 8 bytes of data.
Response Format	*format*<newline> *format* returnsASC,REAL,32,orREAL,64. The response data type is CRD.

Example	:FORM REAL,32 :FORM?

4.4.3 :FORMat:DIGital

Description	Sets the format of bit pattern of calculate subsystem below: :CALCulate:CLIMits:<FAIL|PASS>:DIGital[:DATA] :CALCulate:LIMit:COMPliance:DIGital[:DATA] :CALCulate:LIMit:<LOWer|UPPer>:DIGital[:DATA] :CALCulate:LIMit:PASS:DIGital[:DATA]
Command	:FORMat:DIGital *format* :FORMat:DIGital?
Parameters	*format* reponse format. ASCii (Default is decimal)|BINary|OCTal|HEXadecimal. The parameter type is CPD.
Response Format	*format*<newline> *format* returns ASC, BIN, OCTorHEX. The response data type is CRD.
Example	:FORM:DIG BIN :FORM:DIG?

4.4.4 :FORMat:ELEMents:CALCulate

Description	Specifies the elements contained in the data returned as a result of the calculation by :CALCulate:DATA?, :CALCulate:DATA:LATest?, :CALCulate:MATH:DATA?, :CALCulate:MATH:DATA:LATest? or :TRACe:DATA?. For data stored in the trace buffer, the command is valid for the math result data or limit test data specified by :TRACe:FEED MATH|LIM. The order of elements is fixed: calc, time, status. For example, if timestamp data (time) is not specified, data returned contain calculation data (calc) and status data (status). If returning data before this command is executed, the data contains only the calculation data
Command	:FORMat:ELEMents:CALCulate *type*{,*type*} :FORMat:ELEMents:CALCulate?
Parameters	*type* element contained in the data..

	CALC (calculation data, default)|TIME|STATus. The parameter type is CPD. CALC selects the calculate data; TIME selects the timestamp data; STAT selects the status data;
Response Format	*type*{,type} <newline> *type* returns CALC,TIME or STAT The response data type is CRD.
Example	:FORM:ELEM:CALC CALC,TIME,STAT :FORM:ELEM:CALC?

4.4.5 :FORMat:ELEMents:SENSe

Description	Specifies the elements contained in the data returned as a result of the measurement data by :FETCh?, :READ?, :MEASure? or :TRACe:DATA? . For data stored in the trace buffer, the command is valid for the measurement data specified by :TRACe:FEED SENSe. The order of elements is fixed: voltage, current, resistance, time, status, source For example, if VOLTage and RESistance are not specified, data returned contain current, time, status and source data. If returning data before this command is executed, the data contains all elements.
Command	:FORMat:ELEMents:SENSe type{,type} :FORMat:ELEMents:SENSe?
Parameters	*type* data elements contained in the data. VOLTage|CURRent|RESistance|TIME|STATus|SOURce. The parameter type is CPD. VOLT selects the voltage measurement data. CURR selects the current measurement data. RES selects the resistance measurement data. TIME selects the timestamp data. STAT selects status data. SOUR selects the source output setting data.
Response Format	*type*{,*type*} <newline> *type* returns VOLT,CURR,RES,TIME,STAT, or SOUR. The response data type is CRD.
Example	:FORM:ELEM:SENS SOUR,CURR,VOLT,RES,TIME,STAT

	:FORM:ELEM:SENS?

4.4.6 :FORMat:SREGister

Description	Sets the response format of the status byte register.
Command	:FORMat:SREGister format :FORMat:SREGister?
Parameters	*format* response format. ASCii (Default is decimal)| BINary|OCTal|HEXadecimal. The parameter type is CPD.
Response Format	*format* <newline> *format* returns ASC, BIN, OCT or HEX. The response data type is CRD.
Example	:FORM:SREG BIN :FORM:SREG?

4.5 HCOPy Subsystem

4.5.1 :HCOPy:SDUMp:DATA?

Description	Returns the data of screen image on the front panel. The image data format is specified by :HCOPy:SDUMp:FORMat .
Command	:HCOPy:SDUMp:DATA?
Parameters	None
Response Format	The return is an arbitrary block of binary data of definite length.
Example	:HCOP:SDUM:DATA?

4.5.2 :HCOPy:SDUMp:FORMat

Description	Sets the image data format used in data of the front panel screen image. Return the image data by :HCOPy:SDUMp:DATA? .
Command	:HCOPy:SDUMp:FORMat format :HCOPy:SDUMp:FORMat?

Parameters	*format* image data format. JPG (default)|BMP|PNG|WMF. The parameter type is CPD.
Response Format	*format* <newline> *format* returns JPG,BMP,PNG, or WMF. The response data type is CRD.
Example	:HCOP:SDUM:FORM BMP :HCOP:SDUM:FORM?

4.6 MEASure Subsystem

4.6.1 :MEASure?

Description	Performs one measurement and returns the measurement data set by :FORMat:ELEMents:SENSe. Sense function should be on by :SENSe:FUNCtion[:ON] before executing this command.
Command	:MEASure? [chanlist]
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr,ch1sour,ch2curr,ch2sour* The example contains the current mesurement data (ch1curr) and source data (ch1sour) of channel 1, as well as the current mesurement data (ch2curr) and source data (ch2sour) of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no

	measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:SENS:FUNC "CURR" :FORM:ELEM:SENS CURR,SOUR :MEAS? (@1,2)

4.6.2 :MEASure:<CURRent|RESistance|VOLTage>?

Description	Performs one measurement and returns the measurement data of specified type. Sense function should be on by :SENSe:FUNCtion[:ON] before executing this command.
Command	:MEASure:<CURRent[:DC]|RESistance|VOLTage[:DC]>? [chanlist] For <CURRent[:DC]|RESistance|VOLTage[:DC]>, select CURRent[:DC] for current measurement, RESistance for resistance measurement, or VOLTage[:DC] for voltage measurement.
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr ,ch2curr* The example contains the current mesurement data (ch1curr) of channel 1, as well as the current mesurement data (ch2curr) of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:SENS:FUNC CURR :MEAS:CURR? (@2,1)

4.7 MMEMory Subsystem

4.7.1 :MMEMory:CATalog?

Description	Returns the amount of memory used and availablity. In addition, a list of files and folders in the currently specified directory is returned.
Command	:MMEMory:CATalog? [directory]
Parameters	directory directory name, <path>|USB:\\<path>. \\ (Double backslash) can be used as a path separator. The parameter length is up to 255 ASCII characters. The parameter type is SPD. If directory is not set, this function will be applied to the current directory. If directory = <path>, returns the amount of memory used and availablity of <current directory>\\<path>. If directory = USB:\\<path>, returns the amount of memory used and availablity of USB:\\<path>, where USB:\\ is the root directory of the USB memory connected to the instrument. If the specified directory does not exist or is set to hidden, an error occurs.
Response Format	*used,free*{,item}<newline> *used* returns the size of the used space in bytes.The response data type is NR1. *free* returns the size of the free space in bytes.The response data type is NR1. *item* returns the information of the file or directory.The response data type is SRD. For files, item returns a string name,type,size indicating the file name, file type and file size, where type returns "ASC", "BIN", "STAT" and "MACR", indicating that the file extension "csv", "dat", "sta", and "mac ", respectively. For a directory, item returns the string of name, type and size, where name is the name of the directory and type, size always returns "FOLD,0".
Example	:MMEM:CAT? “USB:\\device1\\iv_test\\result” :MMEM:CAT? “device1\\iv_test\\result”

4.7.2 :MMEMory:CDIRectory

Description	Switches the current directory to the specified directory.
Command	:MMEMory:CDIRectory directory :MMEMory:CDIRectory?

Parameters	directory directory name, <path>|USB:\\<path>. \\ (Double backslash) can be used as a path separator. The parameter length is up to 255 ASCII characters. The parameter type is SPD. If directory is not set, this function will be applied to the current directory. If directory = <path>, returns the amount of memory used and availablity of <current directory>\\<path>. If directory = USB:\\<path>, returns the amount of memory used and availablity of USB:\\<path>, where USB:\\ is the root directory of the USB memory connected to the instrument. If the specified directory does not exist or is set to hidden, an error occurs.
Response Format	*directory* <newline> *directory* returns the full path to the current directory..The response data type is SRD.
Example	:MMEM:CDIR “USB:\\device1\\iv_test\\result” :MMEM:CDIR?

4.7.3 :MMEMory:COPY

Description	Generates a copy of the specified existing file and saves it in the current directory.
Command	:MMEMory:COPY source,destination
Parameters	*source* Name of the source file, also the name of the original file. *destination* Name of file or directory to be copied, <path>|USB:\\<path>. \\ (Double backslash) can be used as a path separator. The parameter length is up to 255 ASCII characters. The parameter type is SPD. If destination is a filename, the copy file will be created in the current directory. If destination = <path>, the source files will be copied in <current directory>\\<path>. If destination = USB:\\<path>, the source files will be copied in USB:\\<path>, where USB:\\ is the root directory of the USB memory connected to the instrument. If the specified directory does not exist or is set to hidden, an error occurs.
Response Format	None
Example	:MMEM:COPY “original.dat”,"original_copy.dat”

	:MMEM:COPY “original.dat”,“USB:\\device1\\iv_test\\result”

4.7.4 :MMEMory:DELete

Description	Delete the specified file in the current directory.
Command	:MMEMory:DELete *file_name*
Parameters	*file_name* Name of the file to be deleted The parameter length is up to 255 ASCII characters. The parameter type is SPD. If the specified file does not exist, an error will occur.
Response Format	None
Example	:MMEM:DEL “original_copy.dat”

4.7.5 :MMEMory:LOAD:MACRo

Description	Loads a macro from a specified file in the current directory.
Command	:MMEMory:LOAD:MACRo *macro,file_name*
Parameters	*macro* Name of the macro. *file_name* Name of the file containing the macro. The file extension must be mac. The parameter length is up to 255 ASCII characters. The parameter type is SPD.
Response Format	None
Example	:MMEM:LOAD:MACR “abc”,“MacroData1.mac”

4.7.6 :MMEMory:LOAD:STATe

Description	Load the instrument settings from the specified file in the current directory.
Command	:MMEMory:LOAD:STATe file_name
Parameters	*file_name* Name of the file containing the instrument settings. The file extension must be sta. The parameter length is up to 255 ASCII characters. The parameter type is SPD.

Response Format	None
Example	:MMEM:LOAD:STAT “SetupData1.sta”

4.7.7 :MMEMory:MDIRectory

Description	Create a new directory.
Command	:MMEMory:MDIRectory directory
Parameters	directory directory name, <path>|USB:\\<path>. \\ (Double backslash) can be used as a path separator. The parameter length is up to 255 ASCII characters. The parameter type is SPD. If directory = <path>, this command creates a directory of <current directory> \\ <path>. If directory = USB:\\<path>, this command creates a directory of USB:\\<path>, where USB:\\ is the root directory of the USB memory connected to the instrument.
Response Format	None
Example	:MMEM:MDIR “USB:\\device1\\iv_test\\setup”

4.7.8 :MMEMory:MOVE

Description	Move or rename the specified file in the current directory.
Command	:MMEMory:MOVE source,destination
Parameters	*source* Name of the source file, also the name of the original file. *destination* Name of file or directory to be moved or renamed, <path>|USB:\\<path>. \\ (Double backslash) can be used as a path separator. The parameter length is up to 255 ASCII characters. The parameter type is SPD. If destination is a filename, the source file will be renamed to the new filename in the current directory. If destination = <path>, the source file will be moved to <current directory \\ <path>. If destination = USB:\\<path>, the source file will be moved to USB:\\<path>, where USB:\\ is the root directory of the USB memory connected to the instrument.

	If the source file does not exist or the target file already exists, an error will occur.
Response Format	None
Example	:MMEM:MOVE “original.dat”,“new.dat” :MMEM:MOVE “original.dat”,“USB:\\device1\\iv_test\\result”

4.7.9 :MMEMory:RDIRectory

Description	Deletes the specified directory that is empty.
Command	:MMEMory:RDIRectory directory
Parameters	directory directory name, <path>|USB:\\<path>. \\ (Double backslash) can be used as a path separator. The parameter length is up to 255 ASCII characters. The parameter type is SPD. If directory = <path>, this command deletes the directory of <current directory>\\<path>. If directory = USB:\\<path>, this command deletes the USB:\\<path> directory, where USB:\\ is the root directory of the USB memory connected to the instrument. If the specified directory is not empty, an error will occur.
Response Format	None
Example	:MMEM:RDIR “USB:\\device1\\iv_test\\setup”

4.7.10 :MMEMory:STORe:DATA<:LIMit|:MATH|:SENSe|[:ALL]>

Description	Specifies the data that may contain limit test data, math result data and measurement data of specified channel, and saves them to the specified file in the current directory.
Command	:MMEMory:STORe:DATA<:LIMit|:MATH|:SENSe|[:ALL]> *file_name[,chanlist]* :LIMit indicates limit test data; :MATH indicates math expression calculation result data; :SENSe indicates measurement data; [:ALL] indicates data of above all kinds.
Parameters	*file_name* Filename used to save the specified data. Maximum of 255 ASCII characters. The parameter type is SPD. To classify files by :MMEMory:CATalog? , it is better to set the file extension to dat.

	chanlist specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	None
Example	:MMEM:STOR:DATA “AllData1.dat”

4.7.11 :MMEMory:STORe:MACRo

Description	Saves the macro to the specified file in the current directory.
Command	:MMEMory:STORe:MACRo macro,file_name
Parameters	*macro* Name of the macro *file_name* Name of the file to save the macro. Maximum of 255 ASCII characters. The parameter type is SPD. To classify files by :MMEMory:CATalog? , it is better to set the file extension to mac.
Response Format	None
Example	:MMEM:STOR:MACR “abc”,“MacroData1.mac”

4.7.12 :MMEMory:STORe:STATe

Description	Saves the configure settings to the specified file in the current directory.
Command	:MMEMory:STORe:STATe file_name
Parameters	*file_name* Name of the file to save the configure settings. Maximum of 255 ASCII characters. The parameter type is SPD. To classify files by :MMEMory:CATalog? , it is better to set the file extension to sta.
Response Format	None
Example	:MMEM:STOR:STAT “SetupData1.sta”

4.7.13 :MMEMory:STORe:TRACe

Description	Saves the data of the specified channel in trace buffer to the specified file

	in the current directory. Refer to :TRACe:FEED for data type.
Command	:MMEMory:STORe:TRACe file_name[,chanlist]
Parameters	*file_name* Name of the file to save the data in trace buffer. Maximum of 255 ASCII characters. The parameter type is SPD. To classify files by :MMEMory:CATalog? , it is better to set the file extension to tra. *chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	None
Example	:MMEM:STOR:TRAC “AllTraceData1.dat”

4.8 OUTPut Subsystem

4.8.1 :OUTPut:FILTer:AUTO

Description	Enables or disables the automatic output filter function. Output filter function can be on by :OUTPut:FILTer[:LPASs][:STATe]
Command	:OUTPut[c]:FILTer:AUTO mode :OUTPut[c]:FILTer:AUTO?
Parameters	*mode* 1|ON (default)|0|OFF. The parameter type is boolean. *mode* = 0 or OFF disables the automatic output filter function. *mode* = 1 or ON enables the automatic output filter function. If this function is enabled, the instrument will automatically set the cutoff frequency of the output filter At this point, the following command settings will be ignored: :OUTPut:FILTer[:LPASs]:FREQuency :OUTPut:FILTer[:LPASs]:TCONstant
Response Format	*mode*<newline> *mode* is 0 or 1, indicating that the automatic output filter function is off or on, respectively. The response data type is NR1.
Example	:OUTP:FILT:AUTO 1

	:OUTP2:FILT:AUTO?

4.8.2 :OUTPut:FILTer[:LPASs]:FREQuency

Description	Sets the cutoff frequency of the output filter. If you have enabled the automatic output filter function by :OUTPut:FILTer:AUTO, the set value will not work.
Command	:OUTPut[c]:FILTer[:LPASs]:FREQuency frequency :OUTPut[c]:FILTer[:LPASs]:FREQuency? [frequency]
Parameters	*frequency* value (31.830 Hz to +31.831 kHz)|MINimum|MAXimum|DEFault. Default is MAXimum. The parameter type is NRf+. If the value you specify is less than the minimum or greater than the maximum, it will be automatically set to the minimum or maximum. The formula of *frequency* and *time_constant* is below: $frequency = 1/(2 \times \pi \times time\_constant)$ Therefore, to both *frequency* and *time_constant*, change one and the other changes synchronously. *time_constant* can be set by :OUTPut:FILTer[:LPASs]:TCONstant.
Response Format	*frequency* <newline> *frequency* returns the currently set cutoff frequency of output filter. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:OUTP:FILT:FREQ 1E4 :OUTP2:FILT:LPAS:FREQ?

4.8.3 :OUTPut:FILTer[:LPASs][:STATe]

Description	Enables or disables output filter function.
Command	:OUTPut[c]:FILTer[:LPASs][:STATe] mode :OUTPut[c]:FILTer[:LPASs][:STATe]?
Parameters	*mode* 0|OFF|1|ON (default). The parameter type is boolean. *mode* = 1 or ON enables the output filter function. *mode* = 0 or OFF disables the output filter function.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the output filter function is off or on,

	respectively. The response data type is NR1.
Example	:OUTP:FILT 0 :OUTP2:FILT:LPAS:STAT?

4.8.4 :OUTPut:FILTer[:LPASs]:TCONstant

Description	Sets the time constant of the output filter. If you have enabled the automatic output filter function by :OUTPut:FILTer:AUTO, the set value will not work.
Command	:OUTPut[c]:FILTer[:LPASs]:TCONstant time_constant :OUTPut[c]:FILTer[:LPASs]:TCONstant? [time_constant]
Parameters	*time_constant* *value* (5 µs to 5 ms)|MINimum|MAXimum| DEFault. The parameter type is NRf+. If the value you specify is less than the minimum or greater than the maximum, it will be automatically set to the minimum or maximum. The formula of *frequency* and *time_constant* is below: *time_constant* = 1/(2 × π × *frequency*) Therefore, to both *frequency* and *time_constant*, change one and the other changes synchronously. *frequency* can be set by :OUTPut:FILTer[:LPASs]:FREQuency.
Response Format	*time_constant* <newline> *time_constant* returns the currently set time constant of output filter. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:OUTP:FILT:TCON 5E-3 :OUTP2:FILT:LPAS:TCON?

4.8.5 :OUTPut:HCAPacitance[:STATe]

Description	Enables or disables high capacitance mode which is effective to test the devices with high capacitance.
Command	:OUTPut[c]:HCAPacitance[:STATe] mode :OUTPut[c]:HCAPacitance[:STATe]?
Parameters	*mode* 1|ON|0|OFF (default). The parameter type is boolean. *mode* = 1 or ON enables high capacitance mode.

	mode = 0 or OFF disables high capacitance mode.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the high capacitance mode is off or on., respectively The response data type is NR1.
Example	:OUTP:HCAP 1 :OUTP2:HCAP:STAT?

4.8.6 :OUTPut:LOW

Description	Selects the state of the low terminal, only useful when the source output is off by :OUTPut[:STATe]
Command	:OUTPut[c]:LOW low_state :OUTPut[c]:LOW?
Parameters	*low_state* FLOat|GROund (default).The parameter type is CPD. *low_state* = FLOat sets the floating ground state. *low_state* = GROund sets ground mode with the low terminal connected to ground.
Response Format	*low_state* <newline> *low_state* is FLO or GRO. The response data type is CRD.
Example	:OUTP:LOW FLO :OUTP2:LOW?

4.8.7 :OUTPut:OFF:AUTO

Description	Enables or disables the AutoOutput-OFF function.
Command	:OUTPut[c]:OFF:AUTO mode :OUTPut[c]:OFF:AUTO?
Parameters	*mode* 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 0 or OFF disables the AutoOutput-OFF function. *mode* = 1 or ON enables the AutoOutput-OFF function. If the function is enabled, source output will be off once the trigger state is changed to IDLE. Refer to :IDLE<:ACQuire|:TRANsient|[:ALL]>?.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that AutoOutput-OFF function is off or on,

	respectively. The response data type is NR1.
Example	:OUTP:OFF:AUTO 1 :OUTP2:OFF:AUTO?

4.8.8 :OUTPut:OFF:MODE

Description	Selects the source mode after the output is turned off.
Command	:OUTPut[c]:OFF:MODE mode :OUTPut[c]:OFF:MODE?
Parameters	mode ZERO|HIZ|NORMal (default). The parameter type is CPD. mode = NORMal: Disconnect the output relay when the output is turned off. When the output is turned off, it is in voltage source mode with a voltage setting value of 0V and a current limit of 100uA mode = HIZ: Disconnect the output relay when the output is turned off. When the output is turned off, the voltage setting value is ≤40 V. The current setting value is ≤100 mA. mode = ZERO: The output relay is not disconnected after the output is turned off. When the output is turned off, it is in voltage source mode with a voltage setting value of 0V and a current limit of 100uA **NOTE:** If the source output is turned off because of overvoltage, overcurrent, overtemperature or interlock-open, the voltage setting value is set to 0V and the source output button is turned off. In this situation, the output off mode will not function.
Response Format	mode <newline> mode returns NORM, HIZ, or ZERO, and indicates the source mode after the output is turned off. The response data type is CRD.
Example	:OUTP:OFF:MODE HIZ :OUTP2:OFF:MODE?

4.8.9 :OUTPut:ON:AUTO

Description	Enables or disables AutoOutput-ON function.
Command	:OUTPut[c]:ON:AUTO mode :OUTPut[c]:ON:AUTO?
Parameters	*mode* 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 0 or OFF disables AutoOutput-ON function. *mode* = 1 or ON enables AutoOutput-ON function. If enabled, source output will be on once the trigger system is initiated by :INITiate or :READ.
Response Format	mode <newline> mode is 0 or 1, indicating that the AutoOutput-ON function is off or on, respectively. The response data type is NR1.
Example	:OUTP:ON:AUTO 0 :OUTP2:ON:AUTO?

4.8.10 :OUTPut:PROTection[:STATe]

Description	Enables or disables overvoltage/overcurrent protection. If the function is on and a compliance state is triggered, the source output will be off immediately with a voltage value set to 0V.
Command	:OUTPut[c]:PROTection[:STATe] mode :OUTPut[c]:PROTection[:STATe]?
Parameters	*mode* 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 0 or OFF disables overvoltage/overcurrent protection. *mode* = 1 or ON enables overvoltage/overcurrent protection.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the overvoltage/overcurrent protection is off or on, respectively. The response data type is NR1.
Example	:OUTP:PROT 1 :OUTP2:PROT:STAT?

4.8.11 :OUTPut:RECall

Description	Restores to the configure settings of all channels saved by :OUTPut:SAVE.
Command	:OUTPut[c]:RECall index
Parameters	*index* 0|1.The parameter type is NR1. *index* = 0 recalls configure settings 0. *index* = 1 recalls configure settings 1.
Response Format	None
Example	:OUTP:REC 1

4.8.12 :OUTPut:SAVE

Description	Saves the current configure settings to number 0 or 1, which can be selected to recall by :OUTPut:RECall.
Command	:OUTPut[c]:SAVE index
Parameters	*index* 0|1.The parameter type is NR1. *index* = 0 saves the current configure settings to number 0. *index* = 1 saves the current configure settings to number 1.
Response Format	None
Example	:OUTP:SAVE 1

4.8.13 :OUTPut[:STATe]

Description	Turns on or off the source output.
Command	:OUTPut[c][:STATe] mode :OUTPut[c][:STATe]?
Parameters	*mode* 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 1 or ON turns on the source output. *mode* = 0 or OFF turns off the source output.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the source output is off or on, respectively. The response data type is NR1.
Example	:OUTP 1 :OUTP2:STAT?

4.9 READ Subsystem

4.9.1 :READ:ARRay?

Description	Returns the measurement data by executing :INITiate and :FETCh:ARRay? . The data types are set by :FORMat:ELEMents:SENSe including voltage measurement data, current measurement data, resistance measurement data, time data, status data, or source output setting data. The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:READ:ARRay? [chanlist]
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the data set by :FORMat:ELEMents:SENSe The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr1, ch1sour1,ch2curr1,ch2sour1,* *ch1curr2, ch1sour2,ch2curr2,ch2sour2,……* *ch1curr5,ch1sour5,ch2curr5,ch2sour5,* *ch1curr6,ch1sour6,*+9.910000E+37,+9.910000E+37,…… *ch1curr10,ch1sour10,*+9.910000E+37,+9.910000E+37 The example contains current mesurement data (ch1currN) and source output setting data (ch1sourN) of 10 steps sweep measurement of channel 1, as well as current mesurement data (ch2currN) and source output setting data (ch2sourN) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FORM:ELEM:SENS CURR,SOUR :READ:ARR? (@1,2)

4.9.2 :READ:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

Description	Returns the measurement data that is specified by the parameter, including current measurement data, resistance measurement data, source output setting data, status data, timestamp data, or voltage measurement data. The command works by executing :INITiate and :FETCh:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?. The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:READ:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>? [chanlist] CURRent specifies the current measurement data; RESistance specifies the resistance measurement data; SOURce specifies the source output setting data; STATus specifies the status data; TIME specifies the timestamp data; VOLTage specifies the voltage measurement data.
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr1,ch2curr1,ch1curr2,ch2curr2,.....* *ch1curr5,ch2curr5,ch1curr6,*+9.910000E+37*, ...* *ch1curr10,*+9.910000E+37 The example contains current mesurement data (ch1currN) of 10 steps sweep measurement of channel 1, as well as current mesurement data (ch2currN) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:READ:ARR:CURR? (@2,1)

4.9.3 :READ[:SCALar]?

Description	Returns the latest saved measurement data by executing :INITiate and :FETCh[:SCALar]? . The data types are set by :FORMat:ELEMents:SENSe including voltage measurement data, current measurement data, resistance measurement data, time data, status data, or source output setting data. The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:READ[:SCALar]? [chanlist]
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the latest data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr10,ch1sour10,ch2curr5,ch2sour5* The example contains the latest current mesurement data (ch1curr10) and the latest source output setting data (ch1sour10) of 10 steps sweep measurement of channel 1, as well as the latest current mesurement data (ch2curr5) and the latest source output setting data (ch2sour5) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FORM:ELEM:SENS CURR,SOUR :READ? (@1,2)

4.9.4 :READ[:SCALar]:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

Description	Returns the latest saved measurement data that is specified by the parameter, including current measurement data, resistance measurement

	data, source output setting data, status data, timestamp data, or voltage measurement data. The command works by executing :INITiate and :FETCh[:SCALar]:< CURRent|RESistance|SOURce|STATus|TIME|VOLTage>? . The saved data will keep until :INITiate, :MEASure or :READ is executed.
Command	:READ[:SCALar]:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>? [chanlist] CURRent specifies the current measurement data; RESistance specifies the resistance measurement data; SOURce specifies the source output setting data; STATus specifies the status data; TIME specifies the timestamp data; VOLTage specifies the voltage measurement data.
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline> *response* returns the latest data specified by CURRent,RESistance,SOURce,STATus,TIME,or VOLTage. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. If both channel 1 and channel 2 are selected by chanlist, channel 1 data and channel 2 data will be returned in order. The example is as follows: *ch1curr10,ch2curr5* The example contains the latest current mesurement data (ch1curr10) of 10 steps sweep measurement of channel 1, as well as the latest current mesurement data (ch2curr5) of 5 steps sweep measurement of channel 2. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:READ:CURR? (@2,1)

4.10 SENSe Subsystem

4.10.1 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture

Description	Sets the integration time for single-point measurements.
Command	:SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture *time* :SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture? [*time*] For <CURRent[:DC]|RESistance|VOLTage[:DC]>, it is not important to specify CURRent[:DC],RESistance or VOLTage [:DC] because the time value is available to all items no matter which sense mode is set.
Parameters	*time* value (+8E-6 to +2 seconds)|MINimum|MAXimum|DEFault (Default 0.1 PLC, = 0.1/power line frequency). The parameter type is NRf+. If the value you specify is less than the minimum or greater than the maximum, it will be automatically set to the minimum or maximum. The formula of integration time is below: *time* = nplc / power line frequency The NPLC value is set by :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles. Therefore, the commands to set NPLC and integration time interrelate, and the latest command decides both values.
Response Format	*time* <newline> *time* returns the currently set integration time. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3, .
Example	:SENS:CURR:APER 2E-3 :SENS2:CURR:DC:APER?

4.10.2 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture:AUTO

Description	Enables or disables the automatic integration time setting function.
Command	:SENSe[*c*]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture:AUTO *mode* :SENSe[*c*]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture:AUTO? For <CURRent[:DC]|RESistance|VOLTage[:DC]>, it is not important to specify CURRent[:DC],RESistance or VOLTage [:DC] because the setting is available to all items no matter which sense mode is set.

Parameters	*mode* 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 0 or OFF disables the automatic integration time setting function. *mode* = 1 or ON enables the automatic integration time setting function. If enabled, the instrument will automatically set the appropriate integration time (NPLC value) for the measurement range. The command is as same as :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles:AUTO.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the automatic integration time setting function is off or on, respectively. The response data type is NR1.
Example	:SENS:CURR:APER:AUTO 0 :SENS2:CURR:DC:APER:AUTO?

4.10.3 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles

Description	Sets the NPLC value, which means the integration time for single point measurement.
Command	:SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles *nplc* :SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles? [*nplc*] For <CURRent[:DC]|RESistance|VOLTage[:DC]>, it is not important to specify CURRent[:DC],RESistance or VOLTage [:DC] because the setting is available to all items no matter which sense mode is set.
Parameters	*nplc* value (50Hz power line frequency: +4E-4 to +100s, 60Hz power line frequency: +4.8E-4 to +120s)|MINimum|MAXimum|DEFault (Default is 0.1 PLC). The parameter type is NRf+. If the value you specify is less than the minimum or greater than the maximum, it will be automatically set to the minimum or maximum. The NPLC value and integration time set by :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture can be converted to each other. The formula of *nplc* and integration time is below: *nplc* = integration time × power line frequency
Response Format	*nplc* <newline> *nplc* returns the currently set NPLC value. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.

	The response data type is NR3.
Example	:SENS:CURR:NPLC 0.2 :SENS2:CURR:DC:NPLC?

4.10.4 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles:AUTO

Description	Enables or disables the automatic NPLC function.
Command	:SENSe[*c*]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles:AUTO *mode* :SENSe[*c*]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles:AUTO? For <CURRent[:DC]|RESistance|VOLTage[:DC]>, it is not important to specify CURRent[:DC],RESistance or VOLTage [:DC] because the setting is available to all items no matter which sense mode is set.
Parameters	*mode* 1|ON (default)|0|OFF. The parameter type is boolean. *mode* = 0 or OFF disables the automatic NPLC function. *mode* = 1 or ON enables the automatic NPLC function. If enabled, . The command is as same as :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture:AUTO.
Response Format	*mode*<newline> *mode* is 0 or 1, indicating that the automatic NPLC function is off or on, respectively. The response data type is NR1.
Example	:SENS:CURR:NPLC:AUTO 0 :SENS2:CURR:DC:NPLC:AUTO?

4.10.5 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel][:BOTH]

Description	Sets the limit value of voltage/current of the specified channel.
Command	:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel][:BOTH] *compliance* :SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel][:BOTH]? [*compliance*] For <CURRent[:DC]|VOLTage[:DC]>, CURRent[:DC] indicates current limit value when in voltage mode; VOLTage[:DC] indicates voltage limit value when in current mode.
Parameters	*compliance* value (Range in Datasheet) |MINimum| MAXimum|

	DEFault (Default is 100μA or 2 V).The parameter type is NRf+. The valid range is from the minimum measurement value to the maximum measurement value of the channel.
Response Format	*compliance* <newline> *compliance* returns the limit value. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:SENS:CURR:PROT 1E-3 :SENS2:CURR:DC:PROT:LEV?

4.10.6 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:NEGative

Description	Sets the negative limit value of voltage/current of the specified channel.
Command	:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:NEGative *compliance* :SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:NEGative? [*compliance*] For <CURRent[:DC]|VOLTage[:DC]>, CURRent[:DC] indicates current limit value when in voltage mode; VOLTage[:DC] indicates voltage limit value when in current mode.
Parameters	*compliance* value (Range in Datasheet) |MINimum| MAXimum| DEFault (Default is -100μA or -2 V).The parameter type is NRf+. The valid range is from the minimum measurement value to the maximum measurement value of the channel.
Response Format	*compliance* <newline> *compliance* returns the negative limit value. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:SENS:CURR:PROT:NEG -1E-3 :SENS2:CURR:DC:PROT:NEG?

4.10.7 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:POSitive

Description	Sets the positive limit value of voltage/current of the specified channel.
Command	:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:POSitive

	compliance :SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:POSitive? [*compliance*] For <CURRent[:DC]|VOLTage[:DC]>, CURRent[:DC] indicates current limit value when in voltage mode; VOLTage[:DC] indicates voltage limit value when in current mode.
Parameters	*compliance* value (Range in Datasheet) |MINimum| MAXimum| DEFault (Default is +100μA or +2 V). The parameter type is NRf+. The valid range is from the minimum measurement value to the maximum measurement value of the channel.
Response Format	*compliance* <newline> *compliance* returns the positive limit value. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:SENS:CURR:PROT:POS 1E-3 :SENS2:CURR:DC:PROT:POS?

4.10.8 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection:TRIPped?

Description	Returns the compliance state of the specified channel, indicating whether the source output reaches the limit value.
Command	:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection:TRIPped? For <CURRent[:DC]|VOLTage[:DC]>, CURRent[:DC] indicates current limit value when in voltage mode; VOLTage[:DC] indicates voltage limit value when in current mode.
Parameters	None
Response Format	*status* <newline> *status* is 1 or 0, indicating whether the compliance state of the specified channel is in or out. The response data type is NR1.
Example	:SENS:CURR:PROT:TRIP?

4.10.9 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO

Description	Enables or disables the automatic range setting function of the measurement of the specified channel.

Command	:SENSe[*d*]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO *mode* :SENSe[*d*]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO? For <CURRent[:DC]|RESistance|VOLTage[:DC]>, CURRent[:DC] indicates current measurement; RESistance indicates resistance measurement; VOLTage[:DC] indicates voltage measurement.
Parameters	*mode* 0|OFF|1|ON (default). The parameter type is boolean. *mode* = 0 or OFF disables the automatic range setting function. If disabled, the range of the measurement will be set by :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe[:UPPer] mode = 1 or ON enables the automatic range setting function. If enabled, the range of the measurement will be set to provide the best display resolution automatically.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating the automatic range setting function is off or on.The response data type is NR1.
Example	:SENS:CURR:RANG:AUTO 0 :SENS2:CURR:DC:RANG:AUTO?

4.10.10 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO:LLIMit

Description	Sets the lower limit value of the automatic range setting, and sets the minimum range of measurement for the value to provide the best display resolution. The minimum range value should be less than or equal to the maximum range value.
Command	:SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO:LLIMit *range* :SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO:LLIMit? [*range*] For <CURRent[:DC]|RESistance|VOLTage[:DC]>, CURRent[:DC] indicates current measurement; RESistance indicates resistance measurement; VOLTage[:DC] indicates voltage measurement.
Parameters	*range* value|MINimum|MAXimum|DEFault.The parameter type is NRf+. *value* is used for the range of voltage/current/resistance measurement, effectively when the resistance measurement is on by :SENSe[c]:FUNCtion[:ON] and set to AUTO by :SENSe[c]:RESistance:MODE.

Response Format	*range* <newline> *range* returns the lower limit value of the automatic range setting. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:SENS:CURR:RANG:AUTO:LLIM 1E-6 :SENS2:CURR:DC:RANG:AUTO:LLIM?

4.10.11 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:MODE

Description	Selects the operation mode for automatic range setting of measurement. The command is effective only when the automatic range setting is on by :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO.
Command	:SENSe[*d*]:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:MODE *mode* :SENSe[*d*]:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:MODE? For <CURRent[:DC]|VOLTage[:DC]>, CURRent[:DC] indicates current measurement; VOLTage[:DC] indicates voltage measurement.
Parameters	*mode* NORMal (default)|RESolution|SPEed.The parameter type is CPD. *mode* = NORMal enables auto setting operation and range up switching operation. *mode* = RESolution enables auto setting operation and range down switching operation. *mode* = SPEed enables auto setting operation, range up switching operation and range down switching operation. Auto setting operation Automatic range setting of measurement to provide the best display resolution. Range up switching operation If measurement data ≥ up value, range of measurement will go up. *up value* = *range* × *rate* / 100 Range down switching operation If measurement data ≤ down value, range of measurement will go down. *down value* = *range* × *rate* / 1000 Value of rate is set by :SENSe:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:THReshold.

Response Format	*mode* <newline> *mode* returns NORM, RES, or SPE, indicating the operation mode of automatic range setting. The response data type is CRD.
Example	:SENS:CURR:RANG:AUTO:MODE SPE :SENS2:CURR:DC:RANG:AUTO:MODE?

4.10.12 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:THReshold

Description	Sets the threshold for range up or down switching operation of operation mode of automatic range setting of measurement.
Command	:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:THReshold *rate* :SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:THReshold? [*rate*] For <CURRent[:DC]|VOLTage[:DC]>, CURRent[:DC] indicates current measurement; VOLTage[:DC] indicates voltage measurement.
Parameters	*rate* value|MINimum|MAXimum|DEFault (Default is 90 %). The parameter type is NRf+. Valid value is from 11 % to 100 %.
Response Format	*rate* <newline> *rate* returns the threshold for range switching operation. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:SENS:CURR:RANG:AUTO:THR 60 :SENS2:CURR:DC:RANG:AUTO:THR?

4.10.13 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO:ULIMit

Description	Sets the upper limit value of the automatic range setting, and sets the maximum range of measurement for the value to provide the best display resolution. Setting of resistance measurement is effective when the resistance measurement is on by :SENSe[c]:FUNCtion[:ON] and set to AUTO by :SENSe[c]:RESistance:MODE. For current measurement and voltage measurement, the maximum range of mesurement is automatically set.
Command	:SENSe[*c*]:RESistance:RANGe:AUTO:ULIMit *range* :SENSe[*c*]:RESistance:RANGe:AUTO:ULIMit? [*range*]

	:SENSe[*c*]:<CURRent|VOLTage>[:DC]:RANGe:AUTO:ULIMit? For <CURRent[:DC]|VOLTage[:DC]>, CURRent[:DC] indicates current measurement; VOLTage[:DC] indicates voltage measurement.
Parameters	*range* value |MINimum|MAXimum| DEFault(Default is 200 MΩ). The parameter type is NRf+.
Response Format	*range* <newline> *range* returns the upper limit value of the automatic range setting. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:SENS:RES:RANG:AUTO:ULIM 1E6 :SENS2:CURR:DC:RANG:AUTO:ULIM?

4.10.14 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe[:UPPer]

Description	Sets the expected measurement value and automatically selects the measurement range to provide the best display resolution.
Command	:SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:UPPer *range* :SENSe[c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:UPPer? [*range*] For <CURRent[:DC]|RESistance|VOLTage[:DC]>, CURRent[:DC] indicates current measurement; RESistance indicates resistance measurement; VOLTage[:DC] indicates voltage measurement.
Parameters	*range* value|UP|DOWN|MINimum|MAXimum|DEFault. The parameter type is NRf+. value is used for the range of voltage/current/resistance measurement, effectively when the resistance measurement is on by :SENSe[c]:FUNCtion[:ON] and set to AUTO by :SENSe[c]:RESistance:MODE. *range* = UP sets the next higger measurement range. *range* = DOWN sets the previous lower measurement range.
Response Format	*range* <newline> *range* returns the selected measurement range. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:SENS:CURR:RANG:UPP 1

	:SENS2:CURR:DC:RANG:UPP?

4.10.15 :SENSe:DATA?

Description	Returns the measurement data of the specified channel. The data types are set by :FORMat:ELEMents:SENSe including voltage measurement data, current measurement data, resistance measurement data, time data, status data, or source output setting data. The data will keep until :INITiate, :MEASure or :READ is executed.
Command	:SENSe[c]:DATA? [*offset*[, *size*]]
Parameters	*offset* n|CURRent|STARt (default). Indicates the location of the first data to be received. The location will change when the command is executed. The parameter type is NR1 or CPD. *offset* = n indicates to start receiving from the (n+1)th data. n is an integer from 0 to the buffer size. *offset* = CURR indicates that the current data position is the starting position. *offset* = STAR indicates to start at the top of the data buffer, which is as same as offset =0. *size* chooses the number of data to be received from 1 to buffer size. The parameter type is NR1. If size is not entered, all data of limit test result from offset will be returned.
Response Format	*response* <newline> *response* returns the data specified by :FORMat:ELEMents:SENSe. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma.The example is as follows: *ch1curr1,ch1sour1,ch1curr2,ch1sour2, ... ch1curr10,ch1sour10* The example contains current mesurement data (ch1currN) and source output setting data (ch1sourN) of 10 steps sweep measurement of channel 1. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FORM:ELEM:SENS CURR,SOUR :SENS:DATA?

4.10.16 :SENSe:DATA:LATest?

Description	Returns the latest measurement data of the specified channel. The data types are set by :FORMat:ELEMents:SENSe including voltage measurement data, current measurement data, resistance measurement data, time data, status data, or source output setting data. The data will keep until :INITiate, :MEASure or :READ is executed.
Command	:SENSe[c]:DATA:LATest?
Parameters	None
Response Format	*response* <newline> *response* returns the latest measurement data specified by :FORMat:ELEMents:SENSe. The response data type is NR3. Refer to "Data Output Format". *response* uses ASCII data output format, and each data is separated by comma. The example is as follows: *ch1curr10,ch1sour10* The example contains the latest current mesurement data (ch1curr10) and the latest source output setting data (ch1sour10) of 10 steps sweep measurement of channel 1. If the sense function is not on by :SENSe:FUNCtion[:ON] or no measurement data exist, *response* will return +9.910000E+37 (ASCII) or NaN (IEEE-754).
Example	:FORM:ELEM:SENS CURR,SOUR :SENS:DATA:LAT?

4.10.17 :SENSe:FUNCtion:OFF

Description	Disables the specified measurement functions of specified channel.
Command	:SENSe[c]:FUNCtion:OFF function[, *function*[, *function*]] :SENSe[c]:FUNCtion:OFF?
Parameters	*function* "CURRent[:DC]"|"VOLTage[:DC]"|"RESistance" (default). The parameter type is SPD. *function* = "CURRent[:DC]" disables current measurement. *function* = "VOLTage[:DC]" disables voltage measurement. *function* = "RESistance" disables resistance measurement.
Response Format	*function*[, function[, function]]<newline> *function* returns "CURR", "VOLT" or "RES", indicating the measurement functions that are disabled.

	If all functions are enabled, query returns a null string “” . The response data type is SRD.
Example	:SENS:FUNC:OFF “RES”,“VOLT” :SENS2:FUNC:OFF?

4.10.18 :SENSe:FUNCtion:OFF:ALL

Description	Disables all measurement functions, including current, voltage and resistance of specified channel.
Command	:SENSe[c]:FUNCtion:OFF:ALL
Parameters	None
Response Format	None
Example	:SENS:FUNC:OFF:ALL

4.10.19 :SENSe:FUNCtion:OFF:COUNt?

Description	Queries the number of disabled measurement functions of specified channel.
Command	:SENSe[c]:FUNCtion:OFF:COUNt?
Parameters	None
Response Format	None
Example	:SENS:FUNC:OFF:COUN?

4.10.20 :SENSe:FUNCtion[:ON]

Description	Enables the specified measurement functions of specified channel.
Command	:SENSe[c]:FUNCtion[:ON] function[, *function*[, *function*]] :SENSe[c]:FUNCtion[:ON]?
Parameters	*function* “CURRent[:DC]”|“VOLTage[:DC]”|“RESistance” Default is “VOLT”,“CURR”. The parameter type is SPD. *function* = “CURRent[:DC]” enables current measurement. *function* = “VOLTage[:DC]” enables voltage measurement. *function* = “RESistance” enables resistance measurement; After enabling the resistance measurement function, set the measurement

	mode by :SENSe:RESistance:MODE.
Response Format	*function*[, function[, function]]<newline> *function* returns "CURR", "VOLT" or "RES", indicating the measurement functions that are enabled. If no function is enabled, query returns a null string "" . The response data type is SRD.
Example	:SENS:FUNC "RES","VOLT" :SENS2:FUNC:ON?

4.10.21 :SENSe:FUNCtion[:ON]:ALL

Description	Enables all measurement functions, including current, voltage and resistance of specified channel. After enabling the resistance measurement function, set the measurement mode by :SENSe:RESistance:MODE.
Command	:SENSe[c]:FUNCtion[:ON]:ALL
Parameters	None
Response Format	None
Example	:SENS:FUNC:ALL

4.10.22 :SENSe:FUNCtion[:ON]:COUNt?

Description	Queries the number of enabled measurement functions of specified channel.
Command	:SENSe[c]:FUNCtion[:ON]:COUNt?
Parameters	None
Response Format	None
Example	:SENS:FUNC:COUN?

4.10.23 :SENSe:FUNCtion:STATe?

Description	Queries whether the specified measurement function of specified channel is enabled or disabled.
Command	:SENSe[c]:FUNCtion:STATe? *function*

Parameters	*function* "CURRent[:DC]"|"VOLTage[:DC]"|"RESistance". The parameter type is SPD. *function* = "CURRent[:DC]" indicates current measurement. *function* = "VOLTage[:DC]" indicates voltage measurement. *function* = "RESistance" indicates resistance measurement;
Response Format	*response* <newline> *response* returns 0 or 1, indicating that the specified measurement function is disabled or enabled, respectively. The response data type is NR1.
Example	:SENS:FUNC:STAT? "CURR"

4.10.24 :SENSe:REMote

Description	Enables or disables the remote sensing function of specified channel. 4-wire connection can be used only when the function is enabled.
Command	:SENSe[c]:REMote *mode* :SENSe[c]:REMote?
Parameters	*mode* 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 0 or OFF disables the remote sensing function. *mode* = 1 or ON enables the remote sensing function.
Response Format	*mode* <newline> *mode*is 0 or 1, indicating that the remote sensing function is off or on. The response data type is NR1.
Example	:SENS:REM 1 :SENS2:REM?

4.10.25 :SENSe:RESistance:MODE

Description	Sets the resistance measurement mode of specified channel.
Command	:SENSe[c]:RESistance:MODE *mode* :SENSe[c]:RESistance:MODE?
Parameters	*mode* MANual (default)|AUTO. The parameter type is CPD. *mode* = MANual, the source and measurement parameters must be set manually, and range of resistance measurement cannot be set. Resistance is calculated by the formula of "voltage/current". *mode* = AUTO, the channel will set to current mode automatically. Range

	of resistance measurement can be set by :SENS:RES:RANG:UPP or :SENS:RES:RANG:AUTO:LLIM and :SENS:RES:RANG:AUTO:ULIM. The following parameters are automatically set: Source mode:CURRent Current source automatic range setting:OFF Source output shape:DC Source output range mode:FIXed Voltage measurement range:2 V range Voltage limit value:2.1 V Voltage measurement automatic range setting:OFF High capacitance mode:OFF
Response Format	*mode*<newline> *mode* returns MAN or AUTO, indicating the resistance measurement mode. The response data type is CRD.
Example	:SENS:RES:MODE MAN :SENS2:RES:MODE?

4.10.26 :SENSe:RESistance:OCOMpensated

Description	Enables or disables resistance measurement offset compensation.
Command	:SENSe[c]:RESistance:OCOMpensated *mode* :SENSe[c]:RESistance:OCOMpensated?
Parameters	*mode* 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 0 or OFF disables resistance measurement offset compensation. *mode* = 1 or ON enables resistance measurement offset compensation.
Response Format	*mode*<newline> *mode* is 0 or 1, indicating that the resistance measurement offset compensation is off or on. The response data type is NR1.
Example	:SENS:RES:OCOM 1 :SENS2:RES:OCOM?

4.10.27 :SENSe:TOUTput:SIGNal

Description	Sets the trigger output ports for the state changes between the trigger layer and acquire action. Multiple trigger output ports can be set.

Command	:SENSe[*c*]:TOUTput:SIGNal *output*{,*output*} :SENSe[*c*]:TOUTput:SIGNal?
Parameters	*output* trigger output port. EXT1 (default)|EXT2|EXT3|EXT4|EXT5| EXT6||INT1|INT2. The parameter type is CPD. *output* = INT1 or INT2 sets internal bus 1 or 2. *output* = EXTn sets GPIO pin.n, which is on the output port of the Digital I/O D-sub connector on the rear panel, from 1 to 6.
Response Format	*response* <newline> *response* returns the set trigger output ports, INT1, INT2 or EXT1 ~ EXT6. The response data type is CRD. Multiple trigger output ports are separated by a comma.
Example	:SENS:TOUT:SIGN EXT3 :SENS2:TOUT:SIGN?

4.10.28 :SENSe:TOUTput[:STATe]

Description	Enables or disables the trigger output on the ports for the state changes between the trigger layer and acquire action. The output ports are set by :SENSe[*c*]:TOUTput:SIGNal.
Command	:SENSe[c]:TOUTput[:STATe] mode :SENSe[c]:TOUTput[:STATe]?
Parameters	*mode* enables or disables trigger output.1|ON|0|OFF (default). The parameter type is boolean *mode* = 1or ON enables the trigger output. *mode* = 0 or OFF disables the trigger output.
Response Format	*response* <newline> *response* returns 1 or 0, indicating the trigger output is on or off. The response data type is NR1.
Example	:SENS:TOUT 1 :SENS2:TOUT:STAT?

4.10.29 :SENSe:WAIT:AUTO

Description	Enables or disables the initial wait time used to calculate the measurement wait time for the specified channel. The initial wait time is set automatically by the instrument and cannot be changed.

	The wait control function should be on by :SENSe:WAIT[:STATe].
Command	:SENSe[c]:WAIT:AUTO *mode* :SENSe[c]:WAIT:AUTO?
Parameters	*mode* 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 1 or ON enables the initial wait time used to calculate. *mode* = 0 or OFF disables the initial wait time used to calculate and set the initial wait time to 0.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the initial wait time is disabled or enabled. The response data type is NR1.
Example	:SENS:WAIT:AUTO 0 :SENS2:WAIT:AUTO?

4.10.30 :SENSe:WAIT:GAIN

Description	Sets the gain used to calculate the measurement wait time for the specified channel. The wait control function should be on by :SENSe:WAIT[:STATe].
Command	:SENSe[c]:WAIT:GAIN gain :SENSe[c]:WAIT:GAIN? [gain]
Parameters	*gain* value (0 to 100)|MINimum|MAXimum|DEFault (Default is1). The parameter type is NRf.
Response Format	*gain* <newline> *gain* returns the set value of gain. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:SENS:WAIT:GAIN 0.5 :SENS2:WAIT:GAIN?

4.10.31 :SENSe:WAIT:OFFSet

Description	Sets the offset used to calculate the measurement wait time for the specified channel. The wait control function should be on by :SENSe:WAIT[:STATe].
Command	:SENSe[c]:WAIT:OFFSet *offset* :SENSe[c]:WAIT:OFFSet? [*offset*]

Parameters	*offset* value (0 to 1 s)|MINimum|MAXimum|DEFault (Default is 0). The parameter type is NRf.
Response Format	*offset* <newline> *offset* returns the set value of offset. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:SENS:WAIT:OFFS 0.5 :SENS2:WAIT:OFFS?

4.10.32 :SENSe:WAIT[:STATe]

Description	Enables or disables the wait time control function of measurement. The wait time is defined as the amount of time that the channel is unable to start a measurement after the start of the DC output or after the trailing edge of the pulse.
Command	:SENSe[c]:WAIT[:STATe] *mode* :SENSe[c]:WAIT[:STATe]?
Parameters	*mode* 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 0 or OFF disables the wait time control function and the initial wait time is set to 0. *mode* = 1 or ON enables the wait time control function. The formula of calculating wait time is as below. • :SENSe:WAIT:AUTO ON|1 condition: wait time = gain × initial wait time + offset • :SENSe:WAIT:AUTO OFF|0 condition: wait time = offset The initial wait time is set automatically by the instrument and cannot be changed. Gain and offset are set by :SENSe:WAIT:GAIN and :SENSe:WAIT:OFFSet, respectively.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the wait time control function of measurement is disabled or enabled. The response data type is NR1.
Example	:SENS:WAIT 0 :SENS2:WAIT:STAT?

4.11 SOURce Subsystem

4.11.1 [:SOURce]:<CURRent|VOLTage>:<CENTer|SPAN>

Description	Sets the value of center or span of current or voltage sweep output.
Command	[:SOURce[c]]:<CURRent|VOLTage>:<CENTer|SPAN> *data* [:SOURce[c]]:<CURRent|VOLTage>:<CENTer|SPAN>? [*data*] <CURRent|VOLTage>: selects CURRent or VOLTage source output; <CENTer|SPAN>: selects CENTer or SPAN value of sweep output.
Parameters	*data* center value or span value. value (Refer to Datasheet)|MINimum|MAXimum|DEFault (0.0). The parameter type is NRf+. *center* and *span* are calculated by the formula below, where *start* and *stop* are set by [:SOURce]:<CURRENT|VOLTagE>:<START|STOP>: *center* = (*start* + *stop*)/2 *span* = *stop* - *start*
Response Format	*data* <newline> *data* returns the set value of center or span. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:CURR:CENT 1E-3 :SOUR2:VOLT:SPAN?

4.11.2 [:SOURce]:<CURRent|VOLTage>[:LEVel][:IMMediate][:AMPLitude]

Description	Sets the value of DC source output of specified channel.
Command	[:SOURce[c]]:<CURRent|VOLTage>[:LEVel][:IMMediate][:AMPLitude] *level* [:SOURce[c]]:<CURRent|VOLTage>[:LEVel][:IMMediate][:AMPLitude]? [*level*] <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*level* voltage or current value of DC source output. value (Refer to Datasheet)|MINimum|MAXimum|DEFault (Default is 0). The parameter type is NRf+.
Response Format	*level* <newline> *level* returns the set value of DC source output. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.

Example	:VOLT 3 :SOUR2:CURR:LEV:IMM:AMPL?

4.11.3 [:SOURce]:<CURRent|VOLTage>[:LEVel]:TRIGgered[:AMPLitude]

Description	Sets the trigger output value of specified channel. When receiving the trigger signal set by :TRIGger<:ACQuire|:TRANsient|[:ALL]>:SOURce[:SIGNal], immediately change the source output value to the value set by this command. Source output after the trigger state changes to IDLE is controlled by [:SOURce]:FUNCtion:TRIGgered:CONTinuous.
Command	[:SOURce[c]]:<CURRent|VOLTage>[:LEVel]:TRIGgered[:AMPLitude] *level* [:SOURce[c]]:<CURRent|VOLTage>[:LEVel]:TRIGgered[:AMPLitude]? [*level*] <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*level* voltage or current value for trigger output. value (Refer to Datasheet) |MINimum|MAXimum|DEFault (Default is 0). The parameter type is NRf+.
Response Format	level <newline> level returns the set value for trigger output. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:VOLT:TRIG 3 :SOUR2:CURR:LEV:TRIG:AMPL?

4.11.4 [:SOURce]:<CURRent|VOLTage>:MODE

Description	Selects the source mode of specified channel: DC, list sweep or sweep.
Command	[:SOURce[c]]:<CURRent|VOLTage>:MODE *mode* [:SOURce[c]]:<CURRent|VOLTage>:MODE? <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*mode* source mode. SWEep|LIST|FIXed (default). The parameter type is CPD. *mode* = FIX selects DC current or voltage source output. *mode* = LIST selects current or voltage list sweep output. *mode* = SWEep selects current or voltage sweep output.
Response Format	*mode* <newline> *mode* returns FIX, LIST, or SWE.The response data type is CRD.

Example	:VOLT:MODE SWE :SOUR2:CURR:MODE?

4.11.5 [:SOURce]:<CURRent|VOLTage>:POINts

Description	Sets the points of sweep output of specified channel.
Command	[:SOURce[c]]:<CURRent|VOLTage>:POINts *points* [:SOURce[c]]:<CURRent|VOLTage>:POINts? [*points*] <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*points* sweep points. value (from 1 to 100,000) |MINimum| MAXimum|DEFault (Default is 1). The parameter type is NRf+. *points* can be calculated by the formula below: *points* = span /step + 1 (step can not be set to 0 in this formula) where step is set by [:SOURce]:<CURRent|VOLTage>:STEP and span is set by [:SOURce]:<CURRent|VOLTage>:<CENTer|SPAN>. step = 0 if points = 1. Change of *points* will make *step* change but not *span*; Change of *step* will make *points* change but not *span*; Change of *span* will make *step* change but not *points*. The calculated value of points of sweep output will be rounded down. *stop* can be calculated by the formula below: *stop* = start + step × (points -1) If the sweep type is LOG, the settings of step and points are ignored, and they will be calculated automatically.
Response Format	*points* <newline> *points* returns the set number of points of sweep output. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR1.
Example	:CURR:POIN 11 :SOUR2:VOLT:POIN?

4.11.6 [:SOURce]:<CURRent|VOLTage>:RANGe

Description	Sets the current or voltage output range of specified channel, effectively when the automatic range setting function is off by [:SOURce[c]]:<CURRent|VOLTage>:RANGe:AUTO.

Command	[:SOURce[c]]:<CURRent|VOLTage>:RANGe range [:SOURce[c]]:<CURRent|VOLTage>:RANGe? <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*range* value (Refer to Datasheet) |MINimum| MAXimum|DEFault. The parameter type is NRf+. value for current or voltage range setting
Response Format	*range* <newline> *range* returns the set value of range.The response data type is NR3.
Example	:CURR:RANG 1E-6 :SOUR2:VOLT:RANG?

4.11.7 [:SOURce]:<CURRent|VOLTage>:RANGe:AUTO

Description	Enables or disables the automatic range setting function of specified channel.
Command	[:SOURce[c]]:<CURRent|VOLTage>:RANGe:AUTO mode [:SOURce[c]]:<CURRent|VOLTage>:RANGe:AUTO? <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*mode* 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 0 or OFF disables the automatic range setting function. If disabled, the range of the source output will be set by [:SOURce[c]]:<CURRent|VOLTage>:RANGe. *mode* = 1 or ON enables the automatic range setting function. If enabled, the range of the source output will be set to provide the best display resolution automatically.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the automatic range setting function is off or on.The response data type is NR1.
Example	:CURR:RANG:AUTO 0 :SOUR2:VOLT:RANG:AUTO?

4.11.8 [:SOURce]:<CURRent|VOLTage>:RANGe:AUTO:LLIMit

Description	Sets the lower limit value of the automatic range setting function of specified channel, which is also the minimum range that can be selected to provide the best display resolution to the source output value.

Command	[:SOURce[c]]:<CURRent|VOLTage>:RANGe:AUTO:LLIMit *range* [:SOURce[c]]:<CURRent|VOLTage>:RANGe:AUTO:LLIMit? [*range*] <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*range* value (Refer to Datasheet) |MINimum| MAXimum|DEFault. The parameter type is NRf+. value for current or voltage range setting.
Response Format	*range* <newline> *range* returns the set lower limit value of the automatic range setting. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:CURR:RANG:AUTO:LLIM 1E-6 :SOUR2:VOLT:RANG:AUTO:LLIM?

4.11.9 [:SOURce]:<CURRent|VOLTage>:RANGe:RPRiority

Description	Selects the output pulse priority mode, including low noise priority and transient speed priority.
Command	[:SOURce[c]]:<CURRent|VOLTage>:RANGe:RPRiority *mode* [:SOURce[c]]:<CURRent|VOLTage>:RANGe:RPRiority? <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*mode* NOISe (default)|TRANsient.The parameter type is CPD. *mode* = NOISe sets the low noise priority mode to output clean pulse with low noise. *mode* = TRANsient sets the transient speed priority mode to output fast pulse with high slope, effectively when the measurement range is 1A, 1.5 A or 3 A in current source mode.
Response Format	*mode* <newline> *mode* returns NOIS or TRAN.The response data type is CRD.
Example	:CURR:RANG:RPR TRAN :SOUR2:VOLT:RANG:RPR?

4.11.10 [:SOURce]:<CURRent|VOLTage>:<STARt|STOP>

Description	Sets the value of start or stop of current or voltage sweep output.
Command	[:SOURce[c]]:<CURRent|VOLTage>:<STARt|STOP> *data* [:SOURce[c]]:<CURRent|VOLTage>:<STARt|STOP>? [*data*]

	<CURRent|VOLTage>: selects CURRent or VOLTage source output. <STARt|STOP>: selects STARt or STOP value to set.
Parameters	*data* STARt or STOP value. value (Refer to Datasheet)|MINimum|MAXimum|DEFault (Default 0.0). The parameter type is NRf+. *start* and *stop* can be calculated by the formulas below: *start* = *center* - *span*/2 *stop* = *center* + *span*/2 where *center* and *span* are set by [:SOURce[c]]:<CURRent|VOLTage>:<CENTer|SPAN>.
Response Format	*data* <newline> *data* returns the set value of start or stop of sweep output. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:VOLT:STOP 10 :SOUR2:CURR:STAR?

4.11.11 [:SOURce]:<CURRent|VOLTage>:STEP

Description	Sets the value of step of current or voltage sweep output.
Command	[:SOURce[c]]:<CURRent|VOLTage>:STEP *step* [:SOURce[c]]:<CURRent|VOLTage>:STEP? [*step*] <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*step* step value. value (Refer to Datasheet)|MINimum|MAXimum|DEFault (Default is 0). The parameter type is NRf+. *step* can be calculated by the formula below: *step* = span/(points - 1) (points can not be 1 in this formula) points set by [:SOURce]:<CURRent|VOLTage>: POINts and span set by [:SOURce]:<CURRent|VOLTage>:<CENTer|SPAN>. *step* = 0 if *points* = 1. Change of *points* will make *step* change but not *span*; Change of *step* will make *points* change but not *span*; Change of *span* will make *step* change but not *points*. The calculated value of points of sweep output will be rounded down. *stop* can be calculated by the formula below:

	stop = start + step × (points -1) If the sweep type is LOG, the settings of step and points are ignored, and they will be calculated automatically. *step* and *span* must have the same polarity or an error occurs.
Response Format	*step* <newline> *step* returns the set value of step. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:VOLT:STEP 0.5 :SOUR2:CURR:STEP?

4.11.12 [:SOURce]:<CURRent\|VOLTage>:TRANsient:SPEed

Description	Selects the transient speed mode, including normal and fast.
Command	[:SOURce[c]]:<CURRent|VOLTage>:TRANsient:SPEed *mode* [:SOURce[c]]:<CURRent|VOLTage>:TRANsient:SPEed? <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*mode* NORMal (default)|FAST.The parameter type is CPD. *mode* = NORMal sets the normal mode for clean output. *mode* = FAST sets the fast mode for a output with high slew. This mode is disabled when high capacitance mode is enabled.
Response Format	*mode* <newline> *mode* returns NORM or FAST. The response data type is CRD.
Example	:CURR:TRAN:SPE FAST :SOUR2:VOLT:TRAN:SPE?

4.11.13 [:SOURce]:DIGital:DATA

Description	Reads and writes GPIO (digital control port) pin data to the pins on the Digital I/O D-sub connector on the rear panel.
Command	[:SOURce]:DIGital:DATA data [:SOURce]:DIGital:DATA?
Parameters	*data* output data. value (0 to 63) (Default is 0). The parameter type is NR1.
Response Format	*data* <newline> *data* returns the data read from the GPIO pins.

	The response data type is NR1 or NDN set by :FORMat:DIGital.
Example	:DIG:DATA 31 :SOUR:DIG:DATA?

4.11.14 [:SOURce]:DIGital:EXTernal:FUNCtion

Description	Sets the input/output function of the GPIO (digital control port) pins.
Command	[:SOURce]:DIGital:EXTernal[n][:FUNCtion] *function* [:SOURce]:DIGital:EXTernal[n][:FUNCtion]?
Parameters	*function* input/output function of pins. DINPut|DIO|TINPut|TOUT. The parameter type is CPD. *function* = DINP indicates digital input. *function* = DIO indicates digital I/O. *function* = TINP indicates trigger input. *function* = TOUT indicates trigger output. **NOTE:** EXT1 to EXT6 are output pins, and EXT7 to EXT12 are input pins.
Response Format	*function* <newline> *function* returns DIO,DINP,TOUT,orTINP.The response data type is CRD.
Example	:DIG:EXT TOUT :SOUR:DIG:EXT2:FUNC?

4.11.15 [:SOURce]:DIGital:EXTernal:POLarity

Description	Sets the polarity of the GPIO (digital control port) pins. The input/output function is set by [:SOURce]:DIGital:EXTernal:FUNCtion.
Command	[:SOURce]:DIGital:EXTernal[n]:POLarity *polarity* [:SOURce]:DIGital:EXTernal[n]:POLarity?
Parameters	*polarity* polarity of the pins. NEG |POS.The parameter type is CPD. *polarity* = POS indicates positive polarity. *polarity* = NEG indicates negative polarity.
Response Format	*polarity* <newline> *polarity* returns POS or NEG.The response data type is CRD.
Example	:DIG:EXT:POL NEG :SOUR:DIG:EXT6:POL?

4.11.16 [:SOURce]:DIGital:EXTernal:TOUTput[:EDGE]:POSition

Description	Selects the timing of trigger signal output for the specified GPIO pin.
Command	[:SOURce]:DIGital:EXTernal[n]:TOUTput[:EDGE]:POSition position [:SOURce]:DIGital:EXTernal[n]:TOUTput[:EDGE]:POSition?
Parameters	*position* timing of trigger signal output. BEFore|AFTer|BOTH (default).The parameter type is CPD. *type* = BEFore indicates to output trigger signal when entering arm layer, trigger layer and device actions (transient or acquire). *type* = AFTer indicates to output trigger signal when exiting arm layer, trigger layer and device actions (transient or acquire). *type* = BOTH indicates to output trigger signal when entering or exiting arm layer, trigger layer and device actions (transient or acquire).
Response Format	*response* <newline> *response* returns timing of trigger signal output, BEF, AFT or BOTH. The response data type is CRD.
Example	:DIG:EXT:TOUT:POS BEF :SOUR:DIG:EXT2:TOUT:POS?

4.11.17 [:SOURce]:DIGital:EXTernal:TOUTput[:EDGE]:WIDTh

Description	Sets the pulse width of the trigger signal output for the specified GPIO pin.
Command	[:SOURce]:DIGital:EXTernal[n]:TOUTput[:EDGE]:WIDTh *width* [:SOURce]:DIGital:EXTernal[n]:TOUTput[:EDGE]:WIDTh? [*width*]
Parameters	*width* pulse width. value (1E-5 to 1E-2s)|MINimum|MAXimum|DEFault (Default 0.1 ms). The parameter type is NRf+.
Response Format	*width* <newline> *width* returns the pulse width set to trigger signal output. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:DIG:EXT:TOUT:WIDT 1E-5 :SOUR:DIG:EXT3:TOUT:WIDT?

4.11.18 [:SOURce]:DIGital:EXTernal:TOUTput:TYPE

Description	Selects the trigger output signal type for the specified GPIO pin.
Command	[:SOURce]:DIGital:EXTernal[n]:TOUTput:TYPE *type* [:SOURce]:DIGital:EXTernal[n]:TOUTput:TYPE?
Parameters	*type* trigger output signal type. EDGE (default)|LEVel. The parameter type is CPD. *type* = EDGE outputs edge signal. *type* = LEVel outputs level signal.
Response Format	*response* <newline> *response* returns the set trigger output signal type, EDGE or LEV. The response data type is CRD.
Example	:DIG:EXT:TOUT:TYPE EDGE :SOUR:DIG:EXT4:TOUT:TYPE?

4.11.19 [:SOURce]:DIGital:INTernal:TOUTput[:EDGE]:POSition

Description	Selects the timing of trigger signal output for internal port 1 or 2.
Command	[:SOURce]:DIGital:INTernal[i]:TOUTput[:EDGE]:POSition position [:SOURce]:DIGital:INTernal[i]:TOUTput[:EDGE]:POSition?
Parameters	*position* timing of trigger signal output. BEFore|AFTer|BOTH (default).The parameter type is CPD. *type* = BEFore indicates to output trigger signal when entering arm layer, trigger layer and device actions (transient or acquire). *type* = AFTer indicates to output trigger signal when exiting arm layer, trigger layer and device actions (transient or acquire). *type* = BOTH indicates to output trigger signal when entering or exiting arm layer, trigger layer and device actions (transient or acquire).
Response Format	*response* <newline> *response* returns timing of trigger signal output.BEF,AFT or BOTH.The response data type is CRD.
Example	:DIG:INT2:TOUT:POS BEF :SOUR:DIG:INT2:TOUT:POS?

4.11.20 [:SOURce]:FUNCtion:MODE

Description	Sets the source output mode to current or voltage of specified channel.
Command	[:SOURce[c]]:FUNCtion:MODE *mode* [:SOURce[c]]:FUNCtion:MODE?
Parameters	*mode* source output mode. CURRent|VOLTage (default).The parameter type is CPD. *mode* = CURR indicates current mode; the voltage limit value is set by :SENS:VOLT:PROT[:LEV]. *mode* = VOLT indicates voltage mode; the current limit value is set by :SENS:CURR:PROT[:LEV] . Refer to :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel][:BOTH].
Response Format	*mode* <newline> *mode* returns CURR or VOLT. The response data type is CRD.
Example	:FUNC:MODE CURR :SOUR2:FUNC:MODE?

4.11.21 [:SOURce]:FUNCtion[:SHAPe]

Description	Sets the source output shape of specified channel.
Command	[:SOURce[c]]:FUNCtion[:SHAPe] *shape* [:SOURce[c]]:FUNCtion[:SHAPe]?
Parameters	*shape* source output shape. PULSe|DC (default). The parameter type is CPD. *shape* = DC indicates direct-current output. *shape* = PULS indicates pulse output.
Response Format	shape <newline> shape returns DC or PULS. The response data type is CRD.
Example	:FUNC PULS :SOUR2:FUNC:SHAP?

4.11.22 [:SOURce]:FUNCtion:TRIGgered:CONTinuous

Description	Enables or disables the continuous trigger output for the specified channel.
Command	[:SOURce[c]]:FUNCtion:TRIGgered:CONTinuous *mode* [:SOURce[c]]:FUNCtion:TRIGgered:CONTinuous?
Parameters	*mode* 0|OFF (default)|1|ON.The parameter type is boolean. *mode* = 1 or ON enables the continuous trigger output. When the trigger system changes from busy to idle, the immediate output value and range of the source become the value and range of the last trigger output. *mode* = 0 or OFF disables the continuous trigger output. When the trigger system changes from busy to idle, the source output value and range are restored to the value specified by the [:SOURce]:<CURRent|VOLTage>[:LEVel][:IMMediate][:AMPLitude] and range set before.
Response Format	mode <newline> mode returns 0 or 1, indicating that the continuous trigger output function is off or on, respectively. The response data type is NR1.
Example	:FUNC:TRIG:CONT 0 :SOUR2:FUNC:TRIG:CONT?

4.11.23 [:SOURce]:LIST:<CURRent|VOLTage>

Description	Sets the current or voltage data for list sweep output of specified channel.
Command	[:SOURce[c]]:LIST:<CURRent|VOLTage> *list* [:SOURce[c]]:LIST:<CURRent|VOLTage>? <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*list* data of a list. Default is 0. The parameter type is NRf. *list* can add data points up to the maximum number of 100,000 (from 1 to 100,000). Each data point must be separated by a comma, for example, list = 0.1,0.2,0.3. For valid values for current or voltage output, refer to Datasheet.
Response Format	*list* <newline> *list* returns the list data; each data point is separated by a comma.The response data type is NR3.
Example	:LIST:VOLT 0.1,0.2,0.3 :SOUR2:LIST:CURR?

4.11.24 [:SOURce]:LIST:<CURRent|VOLTage>:APPend

Description	Adds data to the end of the current or voltage list created by [:SOURce]:LIST:<CURRent|VOLTage>. Data can be added untill the list has up to 100,000 points.
Command	[:SOURce[c]]:LIST:<CURRent|VOLTage>:APPend *append_list* <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	*append_list* List of the output data.The parameter type is NRf+. *append_list* supports multiple data. Each data point must be separated by a comma, for example, append_list = 1.1,1.2,1.3. For valid values for current or voltage output, refer to Datasheet.
Response Format	None
Example	:LIST:VOLT:APP 1.1,1.2,1.3 :SOUR2:LIST:CURR:APP 1E-6,2E-6,3E-6

4.11.25 [:SOURce]:LIST:<CURRent|VOLTage>:POINts?

Description	Queries the number of points of list sweep output. List data can be set by [:SOURce]:LIST:<CURRent|VOLTage> or added by [:SOURce]:LIST:<CURRent|VOLTage>:APPend.
Command	[:SOURce[c]]:LIST:<CURRent|VOLTage>:POINts? <CURRent|VOLTage>: selects CURRent or VOLTage source output.
Parameters	None
Response Format	*number_of_data <newline>* *number_of_data* returns number of points of the list.The response data type is NR1.
Example	:LIST:VOLT:POIN? :SOUR2:LIST:CURR:POIN?

4.11.26 [:SOURce]:LIST:<CURRent|VOLTage>:STARt

Description	Sets the start point for list sweep output.
Command	[:SOURce[c]]:LIST:<CURRent|VOLTage>:STARtstart [:SOURce[c]]:LIST:<CURRent|VOLTage>:STARt? <CURRent|VOLTage>: selects CURRent or VOLTage source output.

Parameters	start index of list data.1 to maximum number of the list (from 1 to 100,000). An error occurs if out of the range. Default is 1. start = 1 indicates to start at the top of the list. The parameter type is NR1.
Response Format	start <newline> start returns the index of the start point of the currently set list. The response data type is NR1.
Example	:LIST:VOLT:STAR 10 :SOUR2:LIST:CURR:STAR?

4.11.27 [:SOURce]:PULSe:DELay

Description	Sets the pulse delay time for the specified channel. The pulse delay time is the time from the start of pulse base output to the start of pulse peak output.
Command	[:SOURce[c]]:PULSe:DELay delay [:SOURce[c]]:PULSe:DELay? [delay]
Parameters	*delay* pulse delay time. value (0.0 to 99999.9s)|MINimum| MAXimum|DEFault (is 0). The parameter type is NRf+.
Response Format	*delay* <newline> *delay* returns the pulse delay time set for the specified channel. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:PULS:DEL 1E-3 :SOUR2:PULS:DEL?

4.11.28 [:SOURce]:PULSe:WIDTh

Description	Sets the pulse width for the specified channel. The pulse width is the time from the start of the pulse peak output (or the start of the pulse level change) to the end of the pulse peak output, which is defined as the time to rise from 10% peak to 90% peak.
Command	[:SOURce[c]]:PULSe:WIDTh width [:SOURce[c]]:PULSe:WIDTh? [width]
Parameters	*width* pulse width. value (5E-5 to 100000 s, resolution is 1E-6s)|MINimum| MAXimum|DEFault (Default is 5E-5).

	The parameter type is NRf+. The minimum pulse width is 50 μs. The minimum pulse period is 100 μs.
Response Format	*width* <newline> *width* returns the set pulse width. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:PULS:WIDT 2E-2 :SOUR2:PULS:WIDT?

4.11.29 [:SOURce]:SWEep:DIRection

Description	Sets the sweep direction of sweep output, UP or DOWN.
Command	[:SOURce[c]]:SWEep:DIRection direction [:SOURce[c]]:SWEep:DIRection?
Parameters	*direction* sweep direction. DOWN|UP (default).The parameter type is CPD. *direction* = UP The sweep direction is from the start value to the stop value, and measurement is performed from the start value to the stop value given by the following formula: *stop* = *start* + *step* × (*points* -1) *direction* = DOWN The sweep direction is from the stop value to the start value, and measurement is performed from the stop value to the start value given by the following formula: *start* = *stop* - *step* × (*points* -1)
Response Format	*direction* <newline> *direction* returns the sweep direction, UP or DOWN. The response data type is CRD.
Example	:SWE:DIR DOWN :SOUR2:SWE:DIR?

4.11.30 [:SOURce]:SWEep:POINts

Description	Sets the points of sweep output of specified channel, effectively both in current and voltage mode.
Command	[:SOURce[c]]:SWEep:POINts points [:SOURce[c]]:SWEep:POINts? MINimum| MAXimum|DEFault

Parameters	*points* sweep points. value (from 1 to 100,000) |MINimum| MAXimum|DEFault (Default is 1). The parameter type is NRf+. *points* can be calculated by the formula below: *points* = span /step + 1 (step can not be set to 0 in this formula) where step is set by [:SOURce]:<CURRent|VOLTage>:STEP and span is set by [:SOURce]:<CURRent|VOLTage>:<CENTer|SPAN>. step = 0 if points = 1. Change of *points* will make *step* change but not *span*; Change of *step* will make *points* change but not *span*; Change of *span* will make *step* change but not *points*. The calculated value of points of sweep output will be rounded down. *stop* can be calculated by the formula below: *stop* = start + step × (points −1) If the sweep type is LOG, the settings of step and points are ignored, and they will be calculated automatically.
Response Format	*points* <newline> *points* returns value assigned to DEF, MIN or MAX. The response data type is NR1.
Example	:SWE:POIN 11 :SOUR2:SWE:POIN? MAX

4.11.31 [:SOURce]:SWEep:RANGing

Description	Sets the range mode of sweep output.
Command	[:SOURce[c]]:SWEep:RANGing mode [:SOURce[c]]:SWEep:RANGing?
Parameters	*mode* range mode. BEST (default)|FIXed|AUTO.The parameter type is CPD. *mode* = BEST, the channel automatically sets the range that covers the entire sweep output level of the linear sweep (SPACing mode = LINear) , or the range that provides the best resolution for each step of the logarithmic sweep (SPACing mode = LOGarithmic). *mode* = AUTO, the channel automatically sets the range that provides the best resolution for each step of the sweep output value. *mode* = FIX, the channel only uses the range set before the start of sweep output and the range remains unchanged during the sweep output.
Response Format	*mode* <newline>

	mode returns the set range mode, BEST, FIX or AUTO. The response data type is CRD.
Example	:SWE:RANG BEST :SOUR2:SWE:RANG?

4.11.32 [:SOURce]:SWEep:SPACing

Description	Sets the spacing mode of sweep output of specified channel.
Command	[:SOURce[*d*]]:SWEep:SPACing mode [:SOURce[*d*]]:SWEep:SPACing?
Parameters	*mode* spacing ratio. LOGarithmic|LINear (default).The parameter type is CPD. *mode* = LIN indicates linear sweep output that output changes by linear step. *mode* = LOG indicates logarithmic sweep output that output changes by logarithmic step. In this mode, the sweep step value is automatically set.
Response Format	*mode*<newline> *mode* returns the spacing mode, LIN or LOG. The response data type is CRD.
Example	:SWE:SPAC LOG :SOUR2:SWE:SPAC?

4.11.33 [:SOURce]:SWEep:STAir

Description	Sets the sweep stair mode of specified channel.
Command	[:SOURce[c]]:SWEep:STAir mode [:SOURce[c]]:SWEep:STAir?
Parameters	*mode* stair mode. SINGle (default)|DOUBle.The parameter type is CPD. *mode* = SINGle sweep output for one time. *mode* = DOUBle sweep output for two times. The first time is from the start to the end and the second time is from the end to the start.
Response Format	*mode*<newline> *mode* returns SING or DOUB, indicating the sweep output mode is SINGle or DOUBle. The response data type is CRD.
Example	:SWE:STA DOUB

	:SOUR2:SWE:STA?

4.11.34 [:SOURce]:TOUTput:SIGNal

Description	Sets the trigger output ports for the state changes between the trigger layer and transient action. Multiple trigger output ports can be set.
Command	[:SOURce[c]]:TOUTput:SIGNal output{,*output*} [:SOURce[c]]:TOUTput:SIGNal?
Parameters	*output* trigger output port. EXT1 (default)|EXT2|EXT3|EXT4|EXT5| EXT6| |INT1|INT2. The parameter type is CPD. *output* = INT1 or INT2 sets internal bus 1 or 2. *output* = EXTn sets GPIO pin.n, which is on the output port of the Digital I/O D-sub connector on the rear panel, from 1 to 6.
Response Format	*response* <newline> *response* returns the trigger output ports, INT1,INT2,or EXT1 to EXT 6 . The response data type is CRD. Multiple ports are separated by a comma.
Example	:TOUT:SIGN EXT3 :SOUR2:TOUT:SIGN?

4.11.35 [:SOURce]:TOUTput[:STATe]

Description	Enables or disables the trigger output on the ports for the state changes between the trigger layer and transient action. The output ports are set by [:SOURce[c]]:TOUTput:SIGNal.
Command	[:SOURce[c]]:TOUTput[:STATe] mode [:SOURce[c]]:TOUTput[:STATe]?
Parameters	*mode* enables or disables trigger output. 1|ON|0|OFF(default). The parameter type is boolean. *mode* = 1 or ON enables the trigger output. *mode* = 0 or OFF disables the trigger output.
Response Format	*response* <newline> *response* returns 1 or 0, indicating the trigger output is on or off. The response data type is NR1.
Example	:TOUT 1 :SOUR2:TOUT:STAT?

4.11.36 [:SOURce]:WAIT:AUTO

Description	Enables or disables the initial wait time used to calculate the source wait time for the specified channel. The initial wait time is set automatically by the instrument and cannot be changed. The wait control function should be on by [:SOURce]:WAIT[:STATe].
Command	[:SOURce[c]]:WAIT:AUTO mode [:SOURce[c]]:WAIT:AUTO?
Parameters	*mode* 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 1 orON enables the initial wait time used to calculate. *mode* = 0 or OFF disables the initial wait time used to calculate and set the initial wait time to 0.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the initial wait time is disabled or enabled. The response data type is NR1.
Example	:WAIT:AUTO 0 :SOUR2:WAIT:AUTO?

4.11.37 [:SOURce]:WAIT:GAIN

Description	Sets the gain used to calculate the source wait time for the specified channel. The wait control function should be on by [:SOURce]:WAIT[:STATe].
Command	[:SOURce[c]]:WAIT:GAIN gain [:SOURce[c]]:WAIT:GAIN? [gain]
Parameters	*gain* value (0 to 100)|MINimum|MAXimum|DEFault (Default is 1). The parameter type is NRf.
Response Format	*gain* <newline> *gain* returns the set value of gain. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:WAIT:GAIN 0.5 :SOUR2:WAIT:GAIN?

4.11.38 [:SOURce]:WAIT:OFFSet

Description	Sets the offset used to calculate the source wait time for the specified channel. The wait control function should be on by [:SOURce]:WAIT[:STATe].
Command	[:SOURce[c]]:WAIT:OFFSet offset [:SOURce[c]]:WAIT:OFFSet? [offset]
Parameters	*offset* value (0 to 1 seconds)|MINimum|MAXimum|DEFault (Default is 0). The parameter type is NRf.
Response Format	*offset* <newline> *offset* returns the set value of offset. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:WAIT:OFFS 0.5 :SOUR2:WAIT:OFFS?

4.11.39 [:SOURce]:WAIT[:STATe]

Description	Enables or disables the wait time control function of source output. The wait time is defined as the amount of time that the channel is unable to change a source output after the start of the DC output or after the trailing edge of the pulse.
Command	[:SOURce[c]]:WAIT[:STATe] mode [:SOURce[c]]:WAIT[:STATe]?
Parameters	*mode* 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 0 or OFF disables the wait time control function and the initial wait time is set to 0. *mode* = 1 or ON enables the wait time control function. [:SOURce]:WAIT:AUTO ON|1 condition: wait time = *gain* × initial wait time + offset [:SOURce]:WAIT:AUTO OFF|0 condition: wait time = *offset* The initial wait time is set automatically by the instrument and cannot be changed. *gain* and *offset* are set by [:SOURce]:WAIT:GAIN and [:SOURce]:WAIT:OFFSet, respectively.
Response Format	*mode* <newline> *mode* is 0 or 1, indicating that the wait time control function of source is

	disabled or enabled. The response data type is NR1.
Example	:WAIT 0 :WAIT?

4.12 STATus Subsystem

4.12.1 :STATus:<MEASurement|OPERation|QUEStionable>:CONDition?

Description	Returns the value of the measurement, operation, or questionable status condition register. For bit definitions, see Table 4-3, Table4-4 and Table 4-5. This command does not change the register settings.
Command	:STATus:<MEASurement|OPERation|QUEStionable>:CONDition? For <MEASurement|OPERation|QUEStionable>: MEASurement: specifies the measurement status condition register; OPERation: specifies the operation status condition register; QUEStionable: specifies the questionable status condition register.
Parameters	None
Response Format	value <newline> returns the value of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:MEAS:COND? :STAT:OPER:COND? :STAT:QUES:COND?

Table 4-3 Description and Definition of Each Bit in the Questionable Status Condition Register

Bit	Decimal	Description	Defination
0	1	Voltage Summary	Channel 1 or/and 2 overvoltage
1	2	Current Summary	Channel 1 or/and 2 overcurrent
2	4	Channel 1 output protection	The output relay of the specified channel is opened by the automatic output off at compliance function.
3	8	Channel 2 output protection	
4	16	Temperature Summary	Channel 1 or/and 2 over-temperature
5 to 7		unused	returns 0

8	256	Calibration	Channel 1 or/and 2 calibration failure
9	512	Self-test	Channel 1 or/and 2 self-test failures
10	1024	Interlock	Interlock terminal open
11	2048	Channel 1 (Trans) trigger event lost	Loss of trigger event at Arm or Trigger layer during transient (Trans) action of channel 1
12	4096	Channel 1 (Acq) trigger event lost	Loss of trigger event at Arm or Trigger layer during Acquisition (Acq) action of channel 1
13	8192	Channel 2 (Trans) trigger event lost	Loss of trigger event at Arm or Trigger layer during transient (Trans) action of channel 2
14	16384	Channel 2 (Acq) trigger event lost	Loss of trigger event at Arm or Trigger layer during Acquisition (Acq) action of channel 2
15		unused	returns 0

Table 4-4 Description and Definition of Each Bit in the Measurement Status Condition Register

Bit	Decimal	Description	Defination
0	1	Channel 1 Limit Test Summary	Failed limit test in channel 1
1	2	Channel 1 measurement readback available	Channel 1 measurement readback is normal
2	4	Channel 1 measurement readback data over range	Channel 1 reading is outside the set measurement range
3	8	Channel 1 trace buffer available	Channel 1 trace buffer has data
4	16	Channel 1 trace buffer full	Channel 1 trace buffer is full
5	32	unused	returns 0
6	64	Channel 2 Limit Test Summary	Failed limit test in channel 2
7	128	Channel 2 measurement readback available	Channel 2 measurement readback is normal
8	256	Channel 2 measurement	Channel 2 reading is outside the set

		readback data over range	measurement range
9	512	Channel 2 trace buffer available	Channel 2 trace buffer has data
10	1024	Channel 2 trace buffer full	Channel 2 trace buffer is full
11 to 15		unused	returns 0

Table 4-5 Description and Definition of Each Bit of the Operation Status Condition Register

Bit	Decimal	Description	Defination
0	1	Calibration/self-test	Calibration/self-test is running
1	2	Channel 1 transient (Trans) action idle	Channel 1 transient (Trans) action is in idle state
2	4	Channel 1 transient (Trans) action waits in the Trigger layer.	Channel 1 transient (Trans) action waits for trigger input signal in the Trigger layer.
3	8	Channel 1 transient (Trans) action waits in the Arm layer.	Channel 1 transient (Trans) action waits for trigger input signal in the Arm layer.
4	16	Channel 1 acquisition (Acq) action idle	Channel 1 acquisition (Acq) action is in idle state
5	32	Channel 1 acquisition (Acq) action waits in the Trigger layer.	Channel 1 acquisition (Acq) action waits for trigger input signal in the Trigger layer.
6	64	Channel 1 acquisition (Acq) action waits in the Arm layer.	Channel 1 acquisition (Acq) action waits for trigger input signal in the Arm layer.
7	128	Channel 2 transient (Trans) action idle	Channel 2 transient (Trans) action is in idle state
8	256	Channel 2 transient (Trans) action waits in the Trigger layer.	Channel 2 transient (Trans) action waits for trigger input signal in the Trigger layer.
9	512	Channel 2 transient (Trans) action waits in the Arm layer.	Channel 2 transient (Trans) action waits for trigger input signal in the Arm layer.
10	1024	Channel 2 acquisition (Acq) action idle	Channel 2 acquisition (Acq) action is in idle state

11	2048	Channel 2 acquisition (Acq) action waits in the Trigger layer.	Channel 2 acquisition (Acq) action waits for trigger input signal in the Trigger layer.
12	4056	Channel 2 acquisition (Acq) action waits in the Arm layer.	Channel 2 acquisition (Acq) action waits for trigger input signal in the Arm layer.
13	8192	Instrument locked	Set to 1 when the instrument is remotely controlled (GPIB, USB, or LAN)
14	16384	Program in progress	Program is running; set to 0 if program memory execution is stopped
15	32768	unused	returns 0

4.12.2 :STATus:<MEASurement|OPERation|QUEStionable>:ENABle

Description	Sets the measurement, operation, or questionable status enable register. The enable register is a mask that allows the true condition in the event register to be reported in the summary bit.
Command	:STATus:<MEASurement|OPERation|QUEStionable>:ENABle mask :STATus:<MEASurement|OPERation|QUEStionable>:ENABle? For <MEASurement|OPERation|QUEStionable>: MEASurement: specifies the measurement status condition register; OPERation: specifies the operation status condition register; QUEStionable: specifies the questionable status condition register.
Parameters	*mask* mask, 0 to65535 (decimal). Default is 0. The parameter type is NR1or NDN. *mask* is the sum of the binary weighted values of the set of bits.
Response Format	*mask* <newline> *mask* returns the value of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:MEAS:ENAB 65535 :STAT:QUES:ENAB?

4.12.3 :STATus:<MEASurement|OPERation|QUEStionable>[:EVENt]?

Description	Returns the value of the Measurement, Operation, or Questionable status

	event register. This command changes the register settings.
Command	:STATus:<MEASurement|OPERation|QUEStionable>[:EVENt]? For <MEASurement|OPERation|QUEStionable>: MEASurement: specifies the measurement status condition register; OPERation: specifies the operation status condition register; QUEStionable: specifies the questionable status condition register.
Parameters	None
Response Format	*value* <newline> *value* returns the value of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:MEAS:EVEN? :STAT:OPER:EVEN? :STAT:QUES:EVEN?

4.12.4 :STATus:<MEASurement|OPERation|QUEStionable>:NTRansition

Description	Sets a negative conversion filter in the Measurement, Operation, or Questionable status register. If a bit of the filter is set, a 1 to 0 transition of its register bits will set the corresponding bit of the event register.
Command	:STATus:<MEASurement|OPERation|QUEStionable>:NTRansition filter :STATus:<MEASurement|OPERation|QUEStionable>:NTRansition? For <MEASurement|OPERation|QUEStionable>: MEASurement: specifies the measurement status condition register; OPERation: specifies the operation status condition register; QUEStionable: specifies the questionable status condition register.
Parameters	*filter* negative conversion filter. 0 to 65535 (decimal), default is 0. The parameter type is NR1 or NDN. filter is the sum of the binary weighted values of the set of bits.
Response Format	*filter* <newline> *filter* returns the value of the negative conversion filter of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:MEAS:NTR 0 :STAT:QUES:NTR?

4.12.5 :STATus:<MEASurement|OPERation|QUEStionable>:PTRansition

Description	Sets a positive conversion filter in the Measurement, Operation, or Questionable status register. If a bit of the filter is set, a 0 to 1 transition of its register bits will set the corresponding bit of the event register.
Command	:STATus:<MEASurement|OPERation|QUEStionable>:PTRansition filter :STATus:<MEASurement|OPERation|QUEStionable>:PTRansition? For <MEASurement|OPERation|QUEStionable>: MEASurement: specifies the measurement status condition register; OPERation: specifies the operation status condition register; QUEStionable: specifies the questionable status condition register.
Parameters	*filter* positive conversion filter. 0 to 65535 (decimal), default is 0. The parameter type is NR1 or NDN. filter is the sum of the binary weighted values of the set of bits.
Response Format	*filter* <newline> *filter* returns the value of the positive conversion filter of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:MEAS:PTR 32767 :STAT:QUES:PTR?

4.12.6 :STATus:PRESet

Description	Sets all defined bits in the Status System PTR register and clear all bits in the NTR and Enable registers. The registers will return to their default state.
Command	:STATus:PRESet
Parameters	None
Response Format	None
Example	:STAT:PRES

4.12.7 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:CONDition?

Description	Returns the value of Questionable status register. Bits 0 and 1 are defined as the status of channel 1 and channel 2, respectively. This command does not change the register settings.

Command	:STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:CONDition? For <CALibration|CURRent|TEMPerature|TEST|VOLTage>; CALibration: specifies the suspicious calibration status enable register; CURRent: specifies the suspicious current status enable register; TEMPerature: specifies the suspicious temperature status enable register; TEST: specifies the suspicious self-test status enable register; VOLTage: specifies the suspicious voltage status enable register.
Parameters	None
Response Format	*value* <newline> returns the value of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:QUES:CAL:COND? :STAT:QUES:CURR:COND? :STAT:QUES:TEMP:COND? :STAT:QUES:TEST:COND? :STAT:QUES:VOLT:COND?

4.12.8 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:ENABle

Description	Sets the questionable calibration, current, temperature, test, and voltagestatus enable registers. The enable register is a mask that allows the truth in the event register to be reported in the summary bit.
Command	:STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:ENABle mask :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:ENABle? For <CALibration|CURRent|TEMPerature|TEST|VOLTage>; CALibration: specifies the suspicious calibration status enable register; CURRent: specifies the suspicious current status enable register; TEMPerature: specifies the suspicious temperature status enable register; TEST: specifies the suspicious self-test status enable register; VOLTage: specifies the suspicious voltage status enable register.
Parameters	*mask* mask, 0 to 65535 (decimal). Default is 0. The parameter type is NR1 or NDN. *mask* is the sum of the binary weighted values of the set of bits.
Response Format	*mask* <newline>

	mask returns the value of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:QUES:CURR:ENAB 65535 :STAT:QUES:TEMP:ENAB?

4.12.9 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>[:EVENt]?

Description	Returns the value of questionable calibration, current, temperature, test, and voltagestatus enable registers. This command changes the register settings.
Command	:STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>[:EVENt]? For <CALibration|CURRent|TEMPerature|TEST|VOLTage>; CALibration: specifies the suspicious calibration status enable register; CURRent: specifies the suspicious current status enable register; TEMPerature: specifies the suspicious temperature status enable register; TEST: specifies the suspicious self-test status enable register; VOLTage: specifies the suspicious voltage status enable register.
Parameters	None
Response Format	*value* <newline> *value* returns the value of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:QUES:CURR:EVEN? :STAT:QUES:VOLT:EVEN? :STAT:QUES:TEMP:EVEN?

4.12.10 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:NTRansition

Description	Sets a negative conversion filter in the calibration, current, temperature, test or voltage status register. If a bit of the filter is set, a 1 to 0 transition of its register bits will set the corresponding bit of the event register.
Command	:STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:NTRansition filter

	:STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:NTRansition? For <CALibration|CURRent|TEMPerature|TEST|VOLTage>; CALibration: specifies the suspicious calibration status enable register; CURRent: specifies the suspicious current status enable register; TEMPerature: specifies the suspicious temperature status enable register; TEST: specifies the suspicious self-test status enable register; VOLTage: specifies the suspicious voltage status enable register.
Parameters	*filter* negative conversion filter. 0 to 65535 (decimal), default is 0. The parameter type is NR1 or NDN. filter is the sum of the binary weighted values of the set of bits.
Response Format	*filter* <newline> *filter* returns the value of the negative conversion filter of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:QUES:CURR:NTR 0 :STAT:QUES:TEMP:NTR?

4.12.11 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:PTRansition

Description	Sets a positive conversion filter in the calibration, current, temperature, test or voltage status register. If a bit of the filter is set, a 0 to 1 transition of its register bits will set the corresponding bit of the event register.
Command	:STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:PTRansition filter :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:PTRansition? For <CALibration|CURRent|TEMPerature|TEST|VOLTage>; CALibration: specifies the suspicious calibration status enable register; CURRent: specifies the suspicious current status enable register; TEMPerature: specifies the suspicious temperature status enable register; TEST: specifies the suspicious self-test status enable register; VOLTage: specifies the suspicious voltage status enable register.
Parameters	*filter* positive conversion filter. 0 to 65535 (decimal), default is 32768. The parameter type is NR1 or NDN. filter is the sum of the binary weighted values of the set of bits.
Response Format	*filter* <newline>

	filter returns the value of the positive conversion filter of the specified register. The response data type is NR1 (decimal) or NDN (binary, octal, or hexadecimal) , set by :FORMat:SREGister.
Example	:STAT:QUES:CURR:PTR 32767 :STAT:QUES:TEMP:PTR?

4.13 SYSTem Subsystem

4.13.1 :SYSTem:BEEPer[:IMMediate]

Description	Controls the buzzer to sound for a specified duration.
Command	:SYSTem:BEEPer[:IMMediate] time
Parameters	time Duration in seconds. 0.05 to 12.75 s.The parameter type is NRf+.
Response Format	None
Example	:SYST:BEEP 0.5

4.13.2 :SYSTem:BEEPer:STATe

Description	Enables or disables the buzzer. *RST reset operation will set this status of this buzzer to be enabled.
Command	:SYSTem:BEEPer:STATe mode :SYSTem:BEEPer:STATe?
Parameters	*mode* enables or disables the buzzer. 0|OFF|1|ON. The parameter type is boolean. *mode* = 1 or ON enables the buzzer. *mode* = 0 or OFF disables the buzzer.
Response Format	*mode* <newline> *mode* returns 0 or 1, indicating that the buzzer is off or on. The response data type is NR1.
Example	:SYST:BEEP:STAT 1 :SYST:BEEP:STAT?

4.13.3 :SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

Description	Sets the GPIB address. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess address :SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?
Parameters	*address* GPIB address, 0 to 30. The parameter type is NR1.
Response Format	*address* <newline> *address* returns GPIB address.The response data type is NR1.
Example	:SYST:COMM:GPIB:ADDR 15 :SYST:COMM:GPIB:ADDR?

4.13.4 :SYSTem:COMMunicate:LAN:ADDRess

Description	Sets the static IP address. The setting is effective after the dynamic IP address is off by :SYSTem:COMMunicate:LAN:DHCP. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:COMMunicate:LAN:ADDRess address :SYSTem:COMMunicate:LAN:ADDRess? [CURRent|STATic]
Parameters	*address* IP address. It must be in A.B.C.D format, with a maximum of 15 characters. A, B, C, and D must be numbers from 0 to 225. The parameter type is SPD.
Response Format	*address* <newline> *address* returns the static IP address. If Parameter = CURRent, *address* returns the currently set IP. If Parameter = STATic, *address* returns the reserved IP for the next power-on. The response data type is SRD.
Example	:SYST:COMM:LAN:ADDR “192.168.100.200” :SYST:COMM:LAN:ADDR?

4.13.5 :SYSTem:COMMunicate:LAN:DHCP

Description	Enables or disables Dynamic Host Configuration Protocol (DHCP). This setting is not changed by power-off or executing the *RST reset command.

	When DHCP is enabled, the instrument will attempt to obtain an IP address from a DHCP server. If the DHCP server finds the instrument, it will assign IP address, subnet mask, and default gateway to the instrument a dynamic. When DHCP is disabled or unavailable, the instrument will use a static IP address, subnet mask, and default gateway during power-on. If the DHCP server does not assign a DHCP LAN address, a static IP address will be used after a timeout of approximately 2 minutes.
Command	:SYSTem:COMMunicate:LAN:DHCP mode :SYSTem:COMMunicate:LAN:DHCP?
Parameters	*mode* DHCP off or on. 0|OFF|1|ON. The parameter type is boolean.
Response Format	*mode* <newline> *mode* returns 0 or 1, indicating that DHCP is off or on. The response data type is NR1.
Example	:SYST:COMM:LAN:DHCP 0 :SYST:COMM:LAN:DHCP?

4.13.6 :SYSTem:COMMunicate:LAN:DNS

Description	Sets the IP address of the DNS server. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:COMMunicate:LAN:DNS[j] address :SYSTem:COMMunicate:LAN:DNS[j]? [CURRent|STATic]
Parameters	Address It must be in A.B.C.D format, with a maximum of 15 characters. A, B, C, and D must be numbers from 0 to 225. The parameter type is SPD.
Response Format	address <newline> address returns the IP address of the DNS server. If Parameter = CURRent, *address* returns the currently set IP. If Parameter = STATic, *address* returns the reserved IP for the next power-on. The response data type is SRD.
Example	:SYST:COMM:LAN:DNS “192.168.0.254” :SYST:COMM:LAN:DNS2?

4.13.7 :SYSTem:COMMunicate:LAN:<GATE|GATeway>

Description	Sets the IP address of the default gateway. The setting is effective after the

	dynamic IP address is off by :SYSTem:COMMunicate:LAN:DHCP. This setting is not changed by power-off or executing the *RST reset command. For <GATE|GATeway>, specifies GATE or GATeway.
Command	:SYSTem:COMMunicate:LAN:<GATE|GATeway> address :SYSTem:COMMunicate:LAN:<GATE|GATeway>? [CURRent|STATic]
Parameters	address It must be in A.B.C.D format, with a maximum of 15 characters. A, B, C, and D must be numbers from 0 to 225. The parameter type is SPD.
Response Format	*address* <newline> *address* returns the IP address of the default gateway. If Parameter = CURRent, *address* returns the currently set IP. If Parameter = STATic, *address* returns the reserved IP for the next power-on. The response data type is SRD.
Example	:SYST:COMM:LAN:GATE "192.168.100.1" :SYST:COMM:LAN:GATE?

4.13.8 :SYSTem:COMMunicate:LAN:<HNAMe|HOSTname>

Description	Sets the host name of the instrument. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:COMMunicate:LAN:<HNAMe|HOSTname> hostname :SYSTem:COMMunicate:LAN:<HNAMe|HOSTname>? [CURRent|STATic]
Parameters	*hostname* host name. Up to 15 characters. The parameter type is SPD.
Response Format	*hostname* <newline> *hostname* returns the host name of the instrument. If Parameter = CURRent, *hostname* returns the currently set IP. If Parameter = STATic, *hostname* returns the reserved IP for the next power-on. The response data type is SRD.
Example	:SYST:COMM:LAN:HNAM "S-SMM3312X-00001" :SYST:COMM:LAN:HOST?

4.13.9 :SYSTem:COMMunicate:LAN:MAC?

Description	Returns the MAC address of the instrument.
Command	:SYSTem:COMMunicate:LAN:MAC?

Parameters	None
Response Format	mac_address <newline> mac_address returns the MAC address of the instrument. The response data type is SRD.
Example	:SYST:COMM:LAN:MAC?

4.13.10 :SYSTem:COMMunicate:LAN:SMASk

Description	Sets the static subnet mask. The setting is effective after the dynamic IP address is off by :SYSTem:COMMunicate:LAN:DHCP. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:COMMunicate:LAN:SMASk subnet_mask :SYSTem:COMMunicate:LAN:SMASk? [CURRent|STATic]
Parameters	*subnet_mask* It must be in A.B.C.D format, with a maximum of 15 characters. A, B, C, and D must be numbers from 0 to 225.The parameter type is SPD.
Response Format	*subnet_mask* <newline> *subnet_mask* returns subnet mask. If Parameter = CURRent, *subnet_mask* returns the currently set IP. If Parameter = STATic, *subnet_mask* returns the reserved IP for the next power-on. The response data type is SRD.
Example	:SYST:COMM:LAN:SMAS “255.255.255.0” :SYST:COMM:LAN:SMAS?

4.13.11 :SYSTem:DATA:QUANtity?

Description	Returns the number of data in the data buffer for the specified channel.
Command	:SYSTem:DATA:QUANtity? [chanlist]
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	*response* <newline>

	response returns the number of data.The response data type is NR1. If both channel 1 and 2 are selected for the channel list, the response will return the number of channel 1 data and the number of channel 2 data separated by commas.
Example	:SYST:DATA:QUAN? (@2)

4.13.12 :SYSTem:DATE

Description	Sets the date of the internal clock. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:DATE year, month, day :SYSTem:DATE?
Parameters	*year* 4-bit integer.The parameter type is NR1. *month* an integer from 1 to 12. The parameter type is NR1. *day* an integer from 1 to 31. The parameter type is NR1.
Response Format	*response* <newline> *response* returns the year, month, and day. Each value is separated by a comma. The response data type is NR1.
Example	:SYST:DATE 2025,1,1

4.13.13 :SYSTem:ERRor:ALL?

Description	Reads and returns all items in the error/event queue and clears the queue.
Command	:SYSTem:ERRor:ALL?
Parameters	None
Response Format	*response* <newline> *response* returns in the format of code and message, indicating the error/event code and message. Multiple responses are listed in FIFO (first in first out) order, separated by commas. The code returned is NR1, the message returned is SRD. If the queue is empty, returns +0,“No error”.
Example	:SYST:ERR:ALL?

4.13.14 :SYSTem:ERRor:CODE:ALL?

Description	Reads all items in the error/event queue, returns all codes, and clears the queue.
Command	:SYSTem:ERRor:CODE:ALL?
Parameters	None
Response Format	*code* <newline> *code* returns the code of error/event queue. Multiple responses are listed in FIFO (first in first out) order, separated by commas.The response data type is NR1. If the queue is empty, returns +0.
Example	:SYST:ERR:CODE:ALL?

4.13.15 :SYSTem:ERRor:CODE[:NEXT]?

Description	Reads and removes the top item in the error/event queue and returns the top code.
Command	:SYSTem:ERRor:CODE[:NEXT]?
Parameters	None
Response Format	*code* <newline> *code* returns the code of error/event queue.The response data type is NR1. If the queue is empty, returns +0.
Example	:SYST:ERR:CODE?

4.13.16 :SYSTem:ERRor:COUNt?

Description	Returns the number of items in the error/event queue.
Command	:SYSTem:ERRor:COUNt?
Parameters	None
Response Format	*response* <newline> *response* returns the number of items. The response data type is NR1. If the queue is empty, returns +0.
Example	:SYST:ERR:COUN?

4.13.17 :SYSTem:ERRor[:NEXT]?

Description	Reads and removes the top item in the error/event queue, and returns the message and code of the top item.
Command	:SYSTem:ERRor[:NEXT]?
Parameters	None
Response Format	*response* <newline> *response* returns in the format of code and message, indicating the error/event code and message. Multiple responses are listed in FIFO (first in first out) order, separated by commas. The code returned is NR1, the message returned is SRD. If the queue is empty, returns +0,"No error".
Example	:SYST:ERR?

4.13.18 :SYSTem:FAN:MODE

Description	Sets the fan control mode. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:FAN:MODE mode :SYSTem:FAN:MODE?
Parameters	*mode* fan control mode, NORMal|RACK.The parameter type is CPD. *mode* = NORM indicates normal mode. *mode* = RACK indicates cabinet mode.
Response Format	*mode* <newline> *mode* returns the set fan control mode, NORM or RACK. The response data type is CRD.
Example	:SYST:FAN:MODE RACK :SYST:FAN:MODE?

4.13.19 :SYSTem:GROup[:DEFine]

Description	Defines the channel grouping. This setting is not changed by power-off or executing the *RST reset command. Channel grouping is used to automatically control the timing of channel outputs in order to keep outputs while other channels perform measurements. Grouped channels start source outputs in order of channel

	number, then simultaneously start measurements and keep outputs until the measurement is completed. If the channel grouping is released, the channels can work independently regardless of the status of the other channels. Conditions for channel grouping: • The trigger source must be the same. • If the trigger source is set to TIMER, the trigger period of the source output trigger (transient action) must be the same. It is not effective if TRIGGER COUNT is set to 1.
Command	:SYSTem:GROup[:DEFine] grouplist :SYSTem:GROup[:DEFine]?
Parameters	*grouplist* Channel grouping. The parameter type is channel list. *grouplist* = (@1,2) groups channel 1 and channel 2 together; *grouplist* = (@1),(@2) releases channel grouping..
Response Format	*grouplist* <newline> *grouplist* returns the setting of channel grouping. The response data type is channel list.
Example	:SYST:GRO (@1,2) :SYST:GRO:DEF?

4.13.20 :SYSTem:GROup:RESet

Description	Releases the channel grouping defined by :SYSTem:GROup[:DEFine].
Command	:SYSTem:GROup:RESet
Parameters	None
Response Format	None
Example	:SYST:GRO:RES

4.13.21 :SYSTem:INTerlock:TRIPped?

Description	Queries the closed or open status of the interlock circuit.
Command	:SYSTem:INTerlock:TRIPped?
Parameters	None
Response Format	mode <newline> mode returns 0 (closed) or 1 (open) , to indicate that the interlock circuit

	is closed or open, respectively. The response data type is NR1.
Example	:SYST:INT:TRIP?

4.13.22 :SYSTem:LANGuage

Description	Selects the control command set for SMM3000X. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:LANGuage *mode* :SYSTem:LANGuage?
Parameters	*mode* control command set for SMM3000X. "DEFault"|"2400". The parameter type is SPD. *mode* = "DEF" selects the default command set that supports all SMM3000X functions. *mode* ="2400" selects the routine command set designed for existing programs that you have created to control existing instruments, such as the Keithley 2400 Standard Series. When this mode is set, the SMM3000X does not support the SCPI commands described in this manual, but only the SCPI commands listed in " Standard Commands Supported by the SMM3000X"
Response Format	*mode*<newline> *mode* returns the currently used command set, DEF or 2400. The response data type is SRD.
Example	:SYST:LANG "2400" :SYST:LANG?

4.13.23 :SYSTem:LFRequency

Description	Selects the power line frequency. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:LFRequency *frequency* :SYSTem:LFRequency?
Parameters	*frequency* power line frequency. 50 (50 Hz, default)|60 (60 Hz).The parameter type is NR1.
Response Format	*frequency*<newline> *frequency* returns the selected power line frequency, 50 or 60.The response data type is NR1.
Example	:SYST:LFR 60

	:SYST:LFR?

4.13.24 :SYSTem:PON

Description	Sets the power-on state. The power-on state can be selected from the default reset and user-defined RCL0, RCL1, RCL2, RCL3 and RCL4, which can be saved by *SAV 0, *SAV 1, *SAV 2, *SAV 3 and *SAV 4 in advance.
Command	:SYSTtem:PON *memory*
Parameters	*memory* power-on state, RST(default)|RCL0|RCL1|RCL2|RCL3|RCL4; The parameter type is CPD.
Response Format	None
Example	:SYST:PON RCL0

4.13.25 :SYSTem:PRESet

Description	Default resets the instrument settings.
Command	:SYSTem:PRESet
Parameters	None
Response Format	None
Example	:SYST:PRESet

4.13.26 :SYSTem:TIME

Description	Sets the time of the internal clock. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:TIME *hour, minute, second* :SYSTem:TIME?
Parameters	*hour* An integer from 0 to 23. The parameter type is NR1. *minute* An integer from 0 to 59. The parameter type is NR1. *second* An integer from 0 to 59. The parameter type is NR1.
Response Format	*response* <newline> *response* returns hour, minute and second. Each value is separated by a comma.

	The response data type is NR1.
Example	:SYST:TIME 23,59,59 :SYST:TIME?

4.13.27 :SYSTem:TIME:TIMer:COUNt?

Description	Returns the current count value of the timer.
Command	:SYSTem:TIME:TIMer:COUNt?
Parameters	None
Response Format	*response* <newline> *response* returns the current count value of the timer. The response data type is NR3.
Example	:SYST:TIME:TIM:COUN?

4.13.28 :SYSTem:TIME:TIMer:COUNt:RESet:AUTO

Description	Enables or disables the automatic reset function of the timer. If this function is enabled, the timer will reset and re count when :INITiate.
Command	:SYSTem:TIME:TIMer:COUNt:RESet:AUTO mode :SYSTem:TIME:TIMer:COUNt:RESet:AUTO?
Parameters	*mode* enables or disables the automatic reset function. 0|OFF|1|ON (default).The parameter type is boolean. *mode* = 1or ON enables the automatic reset function. *mode* = 0 or OFF disables the automatic reset function.
Response Format	*mode* <newline> *mode* returns 0 or 1, indicating that the automatic reset function is off or on, respectively.The response data type is NR1.
Example	:SYST:TIME:TIM:COUN:RES:AUTO 0 :SYST:TIME:TIM:COUN:RES:AUTO?

4.13.29 :SYSTem:TIME:TIMer:COUNt:RESet[:IMMediate]

Description	Immediately resets the timer count.
Command	:SYSTem:TIME:TIMer:COUNt:RESet[:IMMediate]

Parameters	None
Response Format	None
Example	:SYST:TIME:TIM:COUN:RES

4.13.30 :SYSTem:VERSion?

Description	Returns the version of the SCPI standard. This setting is not changed by power-off or executing the *RST reset command.
Command	:SYSTem:VERSion?
Parameters	*response* <newline> *response* returns the version of the SCPI standard. such as 1999.0. The response data type is NR2.
Response Format	None
Example	:SYST:VERS?

4.14 TRACe Subsystem

4.14.1 :TRACe:CLEar

Description	Clears the trace buffer for the specified channel. This command is valid when the trace buffer control mode is set to NEV by the :TRACe:FEED:CONTrol.
Command	:TRACe[c]:CLEar
Parameters	None
Response Format	None
Example	:TRAC:CLE

4.14.2 :TRACe:DATA?

Description	Returns the data in the trace buffer whose data type is specified by :TRACe:FEED.
Command	:TRACe[c]:DATA? [offset[, size]]

Parameters	*offset* n|CURRent|STARt (default). Indicates the location of the first data to be received. The parameter type is NR1 or CPD. *offset* = n indicates to start receiving from the (n+1)th data. n is an integer from 0 to the buffer size. *offset* = CURR indicates that the current data position is the starting position. *offset* = STAR indicates to start at the top of the trace buffer, which is as same as offset =0. *size* chooses the number of data to be received from 1 to buffer size. The parameter type is NR1. If size is not entered, all data in trace buffer from offset will be returned.
Response Format	data <newline> The response data type is NR3. Refer to "Data Output Format".
Example	:TRAC:DATA? 0,10

4.14.3 :TRACe:FEED

Description	Specifies the type of data saved in trace buffer. This command is valid when the trace buffer control mode is set to NEV by the :TRACe:FEED:CONTrol.
Command	:TRACe[c]:FEED *type* :TRACe[c]:FEED?
Parameters	*type* data type. MATH|LIMit|SENSe (default).The parameter type is CPD. *type* = SENS specifies the measurement result data that contains all voltage measurement data, current measurement data, resistance measurement data, time data, status data, or source output setting data specified by :FORMat:ELEMents:SENSe. *type* = LIM specifies the limit test result data that contains all calculation data, time data or status data specified by :FORMat:ELEMents:CALCulate, refer to :CACLulate: DATA? . *type* = MATH specifies the math calculation result data that contains all calculation data, time data or status data specified by :FORMat:ELEMents:CALCulate, refer to :CALCulate:MATH:DATA?.
Response Format	*type* <newline> *type* returns the current type of data, MATH, LIM, or SENS. The response data type is CRD.
Example	:TRAC:FEED MATH

	:TRAC:FEED?

4.14.4 :TRACe:FEED:CONTrol

Description	Selects the trace buffer control mode.
Command	:TRACe[c]:FEED:CONTrol mode :TRACe[c]:FEED:CONTrol?
Parameters	*mode* trace buffer control mode. NEXT|NEVer (default).The parameter type is CPD. *mode* = NEV disables write operations to the trace buffer. In this mode, the commands to set the trace buffer including :TRACe:CLEar, :TRACe:FEED, and :TRACe:POINts are effective. *mode* = NEXT enables write operations to the trace buffer until it is full, and then the trace buffer control mode is set to NEV automatically.
Response Format	*mode* <newline> *mode* returns the trace buffer control mode, NEXT or NEV. The response data type is CRD.
Example	:TRAC:FEED:CONT NEXT :TRAC2:FEED:CONT?

4.14.5 :TRACe:FREE?

Description	Returns the available size and total size of trace buffer.
Command	:TRACe[c]:FREE?
Parameters	None
Response Format	response <newline> response returns available,total. Two value are separated by a comma. The response data type is NR1.
Example	:TRAC2:FREE?

4.14.6 :TRACe:POINts

Description	Sets the trace buffer size. This command is valid when the trace buffer control mode is set to NEV by the :TRACe:FEED:CONTrol.
Command	:TRACe[c]:POINts points

	:TRACe[c]:POINts? [points]
Parameters	*points* trace buffer size. value (1 to 100,000)|MINimum|MAXimum|DEFault (Default is 100,000). The parameter type is NR1.
Response Format	*points* <newline> *points* returns the set trace buffer size. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR1.
Example	:TRAC:POIN 10000 :TRAC2:POIN?

4.14.7 :TRACe:POINts:ACTual?

Description	Returns the actual number of data in trace buffer.
Command	:TRACe[c]:POINts:ACTual?
Parameters	None
Response Format	points <newline> points returns number of data in trace buffer. The response data type is NR1.
Example	:TRAC2:POIN:ACT?

4.14.8 :TRACe:STATistic:DATA?

Description	Returns the result of a statistical operation on data stored in the trace buffer. The statistical operation is set by :TRACe:STATistic:FORMat . Statistical computations of all measurement or calculatioin data except time data and status data in trace buffer will be performed.
Command	:TRACe[c]:STATistic:DATA?
Parameters	response <newline> response returns the result of a statistical operation. The response data type is NR3. Refer to "Data Output Format".
Response Format	None
Example	:TRAC:STAT:DATA?

4.14.9 :TRACe:STATistic:FORMat

Description	Sets the statistical computations to be performed by :TRACe:STATistic:DATA?.
Command	:TRACe[c]:STATistic:FORMat *operation* :TRACe[c]:STATistic:FORMat?
Parameters	*operation* statistical computations. MINimum|MAXimum|SDEViation|PKPK|MEAN (default).The parameter type is CPD. *operation* = MEAN gets the average value. *operation* = SDEV gets the standard deviation. *operation* = PKPK gets the peak-to-peak value. *operation* = MIN gets the minimum value. *operation* = MAX gets the maximum value.
Response Format	*operation* <newline> *operation* returns the currently set statistical computations, MEAN, SDEV, PKPK, MIN or MAX. The response data type is CRD.
Example	:TRAC:STAT:FORM PKPK :TRAC2:STAT:FORM?

4.14.10 :TRACe:TSTamp:FORMat

Description	Selects the rule to return timestamp data in trace buffer.
Command	:TRACe[c]:TSTamp:FORMat *rule* :TRACe[c]:TSTamp:FORMat?
Parameters	*rule* rule to return timestamp data. DELTa|ABSolute (default).The parameter type is CPD. *rule* = ABS sets the returned data to the incremental value of the first timestamped data. *rule* = DELT sets the returned data to the incremental value of the previous timestamped data.
Response Format	*rule* <newline> *rule* returns the currently set rule, DELT or ABS. The response data type is CRD.
Example	:TRAC:TST:FORM DELT :TRAC2:TST:FORM?

4.15 TRIGger Subsystem

4.15.1 :ABORt<:ACQuire|:TRANsient|[:ALL]>

Description	Aborts the specified action for the specified channel, and the trigger state changes to idle.
Command	:ABORt<:ACQuire|:TRANsient|[:ALL]> [chanlist] :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	None
Example	:ABOR:ACQ (@2)

4.15.2 :ARM<:ACQuire|:TRANsient|[:ALL]>[:IMMediate]

Description	Sends an immediate trigger signal to the arm layer of the specified action for the specified channel, and when the specified action is initiated, the trigger signal causes a change from the arm layer to the trigger layer.
Command	:ARM<:ACQuire|:TRANsient|[:ALL]>[:IMMediate] [chanlist] :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	None
Example	:ARM:ACQ (@2)

4.15.3 :ARM<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:BYPass

Description	Enables or disables the bypass function in arm layer.
Command	:ARM[c]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:BYPass *bypass* :ARM[c]<:ACQuire|:TRANsient>[:LAYer]:BYPass? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*bypass* setting of bypass. ONCE|OFF (default).The parameter type is CPD. *bypass* = OFF disables the bypass. *bypass* = ONCE enables the bypass, to skip the trigger signal detector in arm layer in the first trigger cycle.
Response Format	*response* <newline> *response* returns the setting of bypass function, OFF or ONCE. The response data type is CRD.
Example	:ARM:BYP ONCE :ARM2:TRAN:BYP?

4.15.4 :ARM<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:COUNt

Description	Sets the trigger count of specified action in arm layer.
Command	:ARM[c]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:COUNt *arm_count* :ARM[c]<:ACQuire|:TRANsient>[:LAYer]:COUNt? [*arm_count*] :ARM[c][:ALL][:LAYer]:COUNt? *arm_count* :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*arm_count* trigger count in arm layer. value (1 to 100,000 or 2147483647)|INFinity|MINimum|MAXimum| DEFault (Default is 1).The parameter type is NRf+. value=2147483647 indicates infinity. *Arm count* must less than 100,001.
Response Format	*response* <newline> *response* returns the trigger count of specified action in arm layer. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR1. If *Arm count* is set to infinity, returns 2147483647.

Example	:ARM:COUN 10 :ARM2:TRAN:COUN?

4.15.5 :ARM<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:DELay

Description	Sets the trigger delay of specified action in arm layer.
Command	:ARM[*c*]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:DELay delay :ARM[*c*]<:ACQuire|:TRANsient>[:LAYer]:DELay? [delay] :ARM[*c*][:ALL][:LAYer]:DELay? delay :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*delay* trigger delay in second in arm layer. value (0 to 1 00,000)|MINimum|MAXimum|DEFault (0). The parameter type is NRf+.
Response Format	*response* <newline> *response* returns the trigger delay of specified action in arm layer. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:ARM:DEL 0.1 :ARM2:TRAN:DEL?

4.15.6 :ARM<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:SOURce[:SIGNal]

Description	Selects the trigger source for specified action in arm layer.
Command	:ARM[c]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:SOURce[:SIGNal] source :ARM[c]<:ACQuire|:TRANsient>[:LAYer]:SOURce[:SIGNal]? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*source* trigger source in arm layer. AINT (default)|BUS|TIMer|INT1|INT2|EXT7| EXT8|EXT9|EXT10| EXT11|EXT12.The parameter type is CPD. *source* = AINT,(AUTO) Automatically selects the most suitable arm source for the current operating mode by using an internal algorithm. *source* = BUS, selects remote interface trigger commands such as the Group Execution Trigger (GET) and *TRG commands.

	source = TIMer, selects signals generated at regular intervals set by :ARM<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:TIMer. *source* = INT1 or INT2, selects a signal from internal bus 1 or 2, respectively. *source* = EXTn, sets GPIO pin.n, which is on the input port of the Digital I/O D-sub connector on the rear panel, from 7 to 12.
Response Format	*response* <newline> *response* returns the trigger source for specified action in arm layer, AINT, BUS, TIM, INT1, INT2, or EXT7 to EXT12. The response data type is CRD.
Example	:ARM:SOUR AINT :ARM2:TRAN:SOUR?

4.15.7 :ARM<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:TIMer

Description	Sets the time interval of the specified action in arm layer when trigger source is TIMer.
Command	:ARM[c]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:TIMer *interval* :ARM[c]<:ACQuire|:TRANsient>[:LAYer]:TIMer? [*interval*] :ARM[c][:ALL][:LAYer]:TIMer? *interval* :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*interval* time interval in second. value (1E-5 to 1E+5s)|MINimum|MAXimum| DEFault (Default is MINimum).The parameter type is NRf+.
Response Format	*response* <newline> *response* returns the time interval of the specified action in arm layer. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:ARM:TIM 2E-4 :ARM2:TRAN:TIM?

4.15.8 :ARM<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:TOUTput:SIGNal

Description	Sets the trigger output ports for the state changes from the idle state to arm layer. Multiple trigger output ports can be set.
Command	:ARM[c]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:TOUTput:SIGNal *output*{,*output*}

	:ARM[c]<:ACQuire|:TRANsient>[:LAYer]:TOUTput:SIGNal? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*output* trigger output ports. EXT1 (default)|EXT2|EXT3|EXT4| EXT5|EXT6|INT1|INT2. The parameter type is CPD. *output* = INT1 / INT2, selects a signal from internal bus 1 or 2, respectively. *output* = EXTn sets GPIO pin.n, which is on the output port of the Digital I/O D-sub connector on the rear panel, from 1 to 6.
Response Format	*response* <newline> *response* returns the trigger output ports, INT1, INT2 or EXT1 to EXT6. The response data type is CRD. Multiple responses are separated by a comma.
Example	:ARM:TOUT:SIGN EXT1 :ARM2:TRAN:TOUT:SIGN?

4.15.9 :ARM<:ACQuire\|:TRANsient\|[:ALL]>[:LAYer]:TOUTput[:STATe]

Description	Enables or disables the trigger output on the ports for the state changes from the idle state to arm layer. The output ports are set by :ARM[c]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:TOUTput:SIGNal.
Command	:ARM[c]<:ACQuire|:TRANsient|[:ALL]>[:LAYer]:TOUTput[:STATe] *mode* :ARM[c]<:ACQuire|:TRANsient>[:LAYer]:TOUTput[:STATe]? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*mode* trigger output on or off. 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 1 or ON enables the trigger output. *mode* = 0 or OFF disables the trigger output.
Response Format	*response* <newline> *response* returns 1 or 0, indicating that the trigger output is on or off. The response data type is NR1.
Example	:ARM:TOUT 1 :ARM2:TRAN:TOUT:STAT?

4.15.10 :IDLE<:ACQuirel:TRANsient|[:ALL]>?

Description	Checks the status of the specified action for the specified channel and waits for the trigger state to turn to idle.
Command	:IDLE[c]<:ACQuirel:TRANsient|[:ALL]>? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	response <newline> response returns 1 if the trigger state of specified action is idle . The response data type is NR1.
Response Format	None
Example	:IDLE2:ACQ?

4.15.11 :INITiate[:IMMediate]<:ACQuirel:TRANsient|[:ALL]>

Description	Initiates the specified action for the specified channel. The trigger state is changed from idle to start state.
Command	:INITiate[:IMMediate]<:ACQuirel:TRANsient|[:ALL]> [*chanlist*] :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	None
Example	:INIT:ACQ (@2)

4.15.12 :TRIGger<:ACQuirel:TRANsient|[:ALL]>:BYPass

Description	Enables or disables the bypass function in trigger layer.
Command	:TRIGger[c]<:ACQuirel:TRANsient|[:ALL]>:BYPass *bypass* :TRIGger[c]<:ACQuirel:TRANsient>:BYPass?

	:ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*bypass* setting of bypass. ONCE|OFF (default).The parameter type is CPD. *bypass* = OFF disables the bypass. *bypass* = ONCE enables the bypass, to skip the trigger signal detector in trigger layer in the first trigger cycle.
Response Format	*response* <newline> *response* returns the setting of bypass function, OFF or ONCE. The response data type is CRD.
Example	:TRIG:BYP ONCE :TRIG2:TRAN:BYP?

4.15.13 :TRIGger<:ACQuire|:TRANsient|[:ALL]>:COUNt

Description	Sets the trigger count of specified action in trigger layer.
Command	:TRIGger[c]<:ACQuire|:TRANsient|[:ALL]>:COUNt *trigger_count* :TRIGger[c]<:ACQuire|:TRANsient>:COUNt? [*trigger_count*] :TRIGger[c][:ALL]:COUNt? trigger_count :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	trigger_count trigger count. value (1to100000 or 2147483647)|INFinity|MINimum|MAXimum| DEFault (1).The parameter type is NRf+. value=2147483647 indicates infinity. Arm count × Trigger count must less than100,001.
Response Format	response <newline> response returns the trigger count of specified action in trigger layer. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR1.
Example	:TRIG:COUN 10 :TRIG2:TRAN:COUN?

4.15.14 :TRIGger<:ACQuire|:TRANsient|[:ALL]>:DELay

Description	Sets the trigger delay of specified action in trigger layer.
Command	:TRIGger[c]<:ACQuire|:TRANsient|[:ALL]>:DELay *delay* :TRIGger[c]<:ACQuire|:TRANsient>:DELay? [*delay*] :TRIGger[c][:ALL]:DELay? *delay* :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*delay* trigger delay in second. value (0to100000)|MINimum| MAXimum|DEFault (0). The parameter type is NRf+.
Response Format	*response* <newline> *response* returns the trigger delay of specified action in trigger layer. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX.The response data type is NR3.
Example	:TRIG:DEL 0.1 :TRIG2:TRAN:DEL?

4.15.15 :TRIGger<:ACQuire|:TRANsient|[:ALL]>[:IMMediate]

Description	Sends an immediate trigger to the specified action for the specified channel. This trigger signal triggers the specified action when the specified action is in the start state.
Command	:TRIGger<:ACQuire|:TRANsient|[:ALL]>[:IMMediate] [*chanlist*] :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action
Parameters	*chanlist* specifies the data of the selected channel;. The parameter type is channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1). (@1): only channel 1; (@2): only channel 2; (@1,2), (@1:2), (@2,1), and (@2:1): both channel 1 and channel 2. If the parameter is not set, default chanlist = (@1).
Response Format	None
Example	:TRIG:ACQ (@2)

4.15.16 :TRIGger<:ACQuire|:TRANsient|[:ALL]>:SOURce[:SIGNal]

Description	Selects the trigger source for specified action in trigger layer.
Command	:TRIGger[c]<:ACQuire|:TRANsient|[:ALL]>:SOURce[:SIGNal] *source* :TRIGger[c]<:ACQuire|:TRANsient>:SOURce[:SIGNal]? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*source* trigger source in trigger layer. AINT (default)|BUS|TIMer|INT1|INT2|EXT7| EXT8|EXT9|EXT10| EXT11|EXT12.The parameter type is CPD. *source* = AINT,(AUTO) Automatically selects the most suitable trigger source for the current operating mode by using an internal algorithm. *source* = BUS, selects remote interface trigger commands such as the Group Execution Trigger (GET) and *TRG commands. *source* = TIMer, selects signals generated at regular intervals set by :TRIGger <:ACQuire|:TRANsient|[:ALL]>:TIMer. *source* = INT1 or INT2, selects a signal from internal bus 1 or 2, respectively. *source* = EXTn, sets GPIO pin.n, which is on the input port of the Digital I/O D-sub connector on the rear panel, from 7 to 12.
Response Format	*response* <newline> *response* returns the trigger source for specified action in trigger layer, AINT, BUS, TIM, INT1, INT2, or EXT7 to EXT12. The response data type is CRD.
Example	:TRIG:SOUR EXT8 :TRIG2:TRAN:SOUR:SIGN?

4.15.17 :TRIGger<:ACQuire|:TRANsient|[:ALL]>:TIMer

Description	Sets the time interval of the specified action in trigger layer when trigger source is TIMer.
Command	:TRIGger[c]<:ACQuire|:TRANsient|[:ALL]>:TIMer *interval* :TRIGger[c]<:ACQuire|:TRANsient>:TIMer? [*interval*] :TRIGger[c][:ALL]:TIMer? *interval* :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*interval* time interval in second.

	value (1E-5 to 1E+5s)|MINimum|MAXimum| DEFault (Default is MINimum). The parameter type is NRf+.
Response Format	*response* <newline> *response* returns the time interval. If a parameter (DEFault | MINimum | MAXimum) is specified, query returns the value assigned to DEF, MIN, or MAX. The response data type is NR3.
Example	:TRIG:TIM 2E-4 :TRIG2:TRAN:TIM?

4.15.18 :TRIGger<:ACQuire|:TRANsient|[:ALL]>:TOUTput:SIGNal

Description	Sets the trigger output ports for the state changes between the arm layer and trigger layer. Multiple trigger output ports can be set.
Command	:TRIGger[c]<:ACQuire|:TRANsient|[:ALL]>:TOUTput:SIGNal *output*{,*output*} :TRIGger[c]<:ACQuire|:TRANsient>:TOUTput:SIGNal? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*output* trigger output ports. EXT1 (default)|EXT2|EXT3|EXT4|EXT5| EXT6| INT1|INT2.The parameter type is CPD. *output* = INT1 or INT2 sets internal bus 1 or 2. *output* = EXTn sets GPIO pin.n, which is on the output port of the Digital I/O D-sub connector on the rear panel, from 1 to 6.
Response Format	*response* <newline> *response* returns trigger output ports, INT1,INT2 or EXT1 toEXT6. The response data type is CRD. Multiple responses are separated by a comma.
Example	:TRIG:TOUT:SIGN EXT3 :TRIG2:TRAN:TOUT:SIGN?

4.15.19 :TRIGger<:ACQuire|:TRANsient|[:ALL]>:TOUTput[:STATe]

Description	Enables or disables the trigger output on the ports for the state changes between the arm layer and trigger layer. The output ports are set by :TRIGger[c]<:ACQuire|:TRANsient|[:ALL]>:TOUTput:SIGNal.
Command	:TRIGger[c]<:ACQuire|:TRANsient|[:ALL]>:TOUTput[:STATe] *mode*

	:TRIGger[c]<:ACQuire|:TRANsient>:TOUTput[:STATe]? :ACQuire indicates the acquire action; :TRANsient indicates the transient action; :ALL indicates both acquire and transient action;
Parameters	*mode* trigger output on or off. 1|ON|0|OFF (default).The parameter type is boolean. *mode* = 1 or ON enables the trigger output. *mode* = 0 or OFF disables the trigger output.
Response Format	*response* <newline> *response* returns 1 or 0, indicating that the trigger output is on or off. The response data type is NR1.
Example	:TRIG:TOUT 1 :TRIG2:TRAN:TOUT:STAT?

5 Compatibility with Commands of the SMM3000X

5.1 Standard Commands Supported by the SMM3000X

:ABORt

:ARM[:SEQuence[1]][:LAYer[1]]:COUNt <NRf|DEFault|MINimum|MAXimum>

:ARM[:SEQuence[1]][:LAYer[1]]:COUNt?

:ARM[:SEQuence[1]][:LAYer[1]]:SOURce?

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure]:DIRection SOURce|ACCeptor

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure]:DIRection?

:ARM[:SEQuence[1]][:LAYer[1]]:TIMer <NRf>

:ARM[:SEQuence[1]][:LAYer[1]]:TIMer?

:CALCulate[1]:DATA:LATest?

:CALCulate[1]:DATA?

:CALCulate[1]:MATH[:EXPRession]:CATalog?

:CALCulate[1]:MATH[:EXPRession][:DEFine] <form>

:CALCulate[1]:MATH[:EXPRession][:DEFine]?

:CALCulate[1]:MATH[:EXPRession]:DELete:ALL

:CALCulate[1]:MATH[:EXPRession]:DELete[:SELected] <SPD>

:CALCulate[1]:MATH[:EXPRession]:NAME <SPD>

:CALCulate[1]:MATH[:EXPRession]:NAME?

:CALCulate[1]:MATH:UNITs <name>

:CALCulate[1]:MATH:UNITs?

:CALCulate[1]:STATe <Bool>

:CALCulate[1]:STATe?

:CALCulate2:CLIMits:BCONtrol IMMediate|END

:CALCulate2:CLIMits:BCONtrol?

:CALCulate2:CLIMits:CLEar:AUTO <Bool>

:CALCulate2:CLIMits:CLEar:AUTO?

:CALCulate2:CLIMits:CLEar[:IMMediate]

:CALCulate2:CLIMits:FAIL:SOURce2 <NRf|NDN>

:CALCulate2:CLIMits:FAIL:SOURce2?

:CALCulate2:CLIMits:MODE GRADing|SORTing

:CALCulate2:CLIMits:MODE?

:CALCulate2:CLIMits:PASS:SOURce2 <NRf|NDN>

:CALCulate2:CLIMits:PASS:SOURce2?

:CALCulate2:DATA:LATest?

:CALCulate2:DATA?

:CALCulate2:FEED CALCulate[1]|VOLTage|CURRent|RESistance

:CALCulate2:FEED?

:CALCulate2:LIMit[1]:COMPliance:FAIL IN|OUT

:CALCulate2:LIMit[1]:COMPliance:FAIL?

:CALCulate2:LIMit[1]:COMPliance:SOURce2 <NRf|NDN>

:CALCulate2:LIMit[1]:COMPliance:SOURce2?

:CALCulate2:LIMit[1]:FAIL?

:CALCulate2:LIMit[1]:STATe <Bool>

:CALCulate2:LIMit[1]:STATe?

:CALCulate2:LIMit[2|3|5-12]:FAIL?

:CALCulate2:LIMit[2|3|5-12]:LOWer[:DATA] <NRf>

:CALCulate2:LIMit[2|3|5-12]:LOWer[:DATA]? [DEFault|MINimum|MAXimum]

:CALCulate2:LIMit[2|3|5-12]:LOWer:SOURce2 <NRf|NDN>

:CALCulate2:LIMit[2|3|5-12]:LOWer:SOURce2?

:CALCulate2:LIMit[2|3|5-12]:PASS:SOURce2 <NRf|NDN>

:CALCulate2:LIMit[2|3|5-12]:PASS:SOURce2?

:CALCulate2:LIMit[2|3|5-12]:STATe <Bool>

:CALCulate2:LIMit[2|3|5-12]:STATe?

:CALCulate2:LIMit[2|3|5-12]:UPPer[:DATA] <NRf|DEFault|MINimum|MAXimum>

:CALCulate2:LIMit[2|3|5-12]:UPPer[:DATA]? [DEFault|MINimum|MAXimum]

:CALCulate2:LIMit[2|3|5-12]:UPPer:SOURce2 <NRf|NDN>

:CALCulate2:LIMit[2|3|5-12]:UPPer:SOURce2?

:CALCulate2:LIMit4:SOURce2 <NRf>

:CALCulate2:LIMit4:SOURce2?

:CALCulate2:NULL:ACQuire

:CALCulate2:NULL:OFFSet <NRf|DEFault|MINimum|MAXimum>

:CALCulate2:NULL:OFFSet? [DEFault|MINimum|MAXimum]

:CALCulate2:NULL:STATe <Bool>

:CALCulate2:NULL:STATe?

:CALCulate3:DATA?

:CALCulate3:FORMat MEAN|SDEViation|MAXimum|MINimum|PKPK

:CALCulate3:FORMat?

:CONFigure:CURRent[:DC]

:CONFigure:RESistance

:CONFigure:VOLTage[:DC]

:CONFigure?

:DISPlay:CNDisplay

:DISPlay:DIGits <4|5|6|7|DEFault|MINimum|MAXimum>

:DISPlay:DIGits? [DEFault|MINimum|MAXimum]

:DISPlay:ENABle <Bool>

:DISPlay:ENABle?

:DISPlay[:WINDow[1]]:TEXT:DATA <SPD>

:DISPlay[:WINDow[1]]:TEXT:DATA?

:DISPlay[:WINDow[1]]:TEXT:STATe <Bool>

:DISPlay[:WINDow[1]]:TEXT:STATe?

:DISPlay:WINDow2:TEXT:DATA <SPD>

:DISPlay:WINDow2:TEXT:DATA?

:DISPlay:WINDow2:TEXT:STATe <Bool>

:DISPlay:WINDow2:TEXT:STATe?

:FETCh?

:FORMat:BORDer <NORMal|SWAPped>

:FORMat:BORDer?

:FORMat:ELEMents:CALCulate <CALC|TIME|STATus>

:FORMat:ELEMents:CALCulate?

:FORMat:ELEMents[:SENSe[1]] <VOLTage|CURRent|RESistance|TIME|STATus>

:FORMat:ELEMents[:SENSe[1]]?

:FORMat:SOURce2 <ASCii|HEXadecimal|OCTal|BINary>

:FORMat:SOURce2?

:FORMat:SREGister <ASCii|HEXadecimal|OCTal|BINary>

:FORMat:SREGister?

:FORMat[:DATA] <ASCii|REAL|SREal>[,NRf]

:FORMat[:DATA]?

:INITiate[:IMMediate]

:MEASure:CURRent[:DC]?

:MEASure:RESistance?

:MEASure:VOLTage[:DC]?

:MEASure?

:OUTPut[1]:INTerlock:TRIPped?

:OUTPut[1]:SMODe <HIMPedance|NORMal|ZERO|GURAd>

:OUTPut[1]:SMODe?

:OUTPut[1][:STATe] <Bool>

:OUTPut[1][:STATe]?

:READ?

[:SENSe[1]]:CURRent[:DC]:NPLCycles <NRf|DEFault|MINimum|MAXimum>

[:SENSe[1]]:CURRent[:DC]:NPLCycles? [DEFault|MINimum|MAXimum]

[:SENSe[1]]:CURRent[:DC]:PROTection[:LEVel] <NRf|DEFault|MINimum|MAXimum>

[:SENSe[1]]:CURRent[:DC]:PROTection[:LEVel]? DEFault|MINimum|MAXimum

[:SENSe[1]]:CURRent[:DC]:PROTection:TRIPped?

[:SENSe[1]]:CURRent[:DC]:RANGe:AUTO <Bool>

[:SENSe[1]]:CURRent[:DC]:RANGe:AUTO?

[:SENSe[1]]:CURRent[:DC]:RANGe:AUTO:LLIMit <NRf>

[:SENSe[1]]:CURRent[:DC]:RANGe:AUTO:LLIMit?

[:SENSe[1]]:CURRent[:DC]:RANGe:AUTO:ULIMit?

[:SENSe[1]]:CURRent[:DC]:RANGe[:UPPer] <NRf|DEFault|MINimum|MAXimum|UP|DOWN>

[:SENSe[1]]:CURRent[:DC]:RANGe[:UPPer]? [DEFault|MINimum|MAXimum]

[:SENSe[1]]:DATA[:LATest]?

[:SENSe[1]]:FUNCtion:CONCurrent <Bool>

[:SENSe[1]]:FUNCtion:CONCurrent?

[:SENSe[1]]:FUNCtion:OFF <CURRent[:DC]|VOLTage[:DC]|RESistance>,..

[:SENSe[1]]:FUNCtion:OFF:ALL

[:SENSe[1]]:FUNCtion:OFF:COUNt?

[:SENSe[1]]:FUNCtion:OFF?

[:SENSe[1]]:FUNCtion[:ON] <CURRent[:DC]|VOLTage[:DC]|RESistance>,..

[:SENSe[1]]:FUNCtion[:ON]:ALL

[:SENSe[1]]:FUNCtion[:ON]:COUNt?

[:SENSe[1]]:FUNCtion[:ON]?

[:SENSe[1]]:FUNCtion:STATe? <“CURRent[:DC]”|“VOLTage[:DC]”|“RESistance”>

[:SENSe[1]]:RESistance:MODE <MANual|AUTO>

[:SENSe[1]]:RESistance:MODE?

[:SENSe[1]]:RESistance:NPLCycles <NRf|DEFault|MINimum|MAXimum>

[:SENSe[1]]:RESistance:NPLCycles? [DEFault|MINimum|MAXimum]

[:SENSe[1]]:RESistance:OCOMpensated <Bool>

[:SENSe[1]]:RESistance:OCOMpensated?

[:SENSe[1]]:RESistance:RANGe:AUTO <Bool>

[:SENSe[1]]:RESistance:RANGe:AUTO?

[:SENSe[1]]:RESistance:RANGe:AUTO:LLIMit <NRf>

[:SENSe[1]]:RESistance:RANGe:AUTO:LLIMit?

[:SENSe[1]]:RESistance:RANGe:AUTO:ULIMit <NRf>

[:SENSe[1]]:RESistance:RANGe:AUTO:ULIMit?

[:SENSe[1]]:RESistance:RANGe[:UPPer] <NRf|DEFault|MINimum|MAXimum|UP|DOWN>

[:SENSe[1]]:RESistance:RANGe[:UPPer]? [DEFault|MINimum|MAXimum]

[:SENSe[1]]:VOLTage[:DC]:NPLCycles <NRf|DEFault|MINimum|MAXimum>

[:SENSe[1]]:VOLTage[:DC]:NPLCycles? [DEFault|MINimum|MAXimum]

[:SENSe[1]]:VOLTage[:DC]:PROTection[:LEVel] <NRf|DEFault|MINimum|MAXimum>

[:SENSe[1]]:VOLTage[:DC]:PROTection[:LEVel]? <DEFault|MINimum|MAXimum>

[:SENSe[1]]:VOLTage[:DC]:PROTection:TRIPped?

[:SENSe[1]]:VOLTage[:DC]:RANGe:AUTO <Bool>

[:SENSe[1]]:VOLTage[:DC]:RANGe:AUTO?

[:SENSe[1]]:VOLTage[:DC]:RANGe:AUTO:LLIMit <NRf>

[:SENSe[1]]:VOLTage[:DC]:RANGe:AUTO:LLIMit?

[:SENSe[1]]:VOLTage[:DC]:RANGe:AUTO:ULIMit?

[:SENSe[1]]:VOLTage[:DC]:RANGe[:UPPer] <NRf|DEFault|MINimum|MAXimum|UP|DOWN>

[:SENSe[1]]:VOLTage[:DC]:RANGe[:UPPer]? [DEFault|MINimum|MAXimum]

:SOURce[1]:CLEar:AUTO <Bool>

:SOURce[1]:CLEar:AUTO?

:SOURce[1]:CLEar[:IMMediate]

:SOURce[1]:CURRent:CENTer <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:CURRent:CENTer? [DEFault|MINimum|MAXimum]

:SOURce[1]:CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:CURRent[:LEVel][:IMMediate][:AMPLitude]? [DEFault|MINimum|MAXimum]

:SOURce[1]:CURRent[:LEVel]:TRIGgered[:AMPLitude] <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:CURRent[:LEVel]:TRIGgered[:AMPLitude]? [DEFault|MINimum|MAXimum]

:SOURce[1]:CURRent:MODE <FIXed|LIST|SWEep>

:SOURce[1]:CURRent:MODE?

:SOURce[1]:CURRent:RANGe <NRf|DEFault|MINimum|MAXimum|UP|DOWN>

:SOURce[1]:CURRent:RANGe? [DEFault|MINimum|MAXimum]

:SOURce[1]:CURRent:RANGe:AUTO <Bool>

:SOURce[1]:CURRent:RANGe:AUTO?

:SOURce[1]:CURRent:SPAN <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:CURRent:SPAN? [DEFault|MINimum|MAXimum]

:SOURce[1]:CURRent:STARt <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:CURRent:STARt? [DEFault|MINimum|MAXimum]

:SOURce[1]:CURRent:STEP <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:CURRent:STEP? [DEFault|MINimum|MAXimum]

:SOURce[1]:CURRent:STOP <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:CURRent:STOP? [DEFault|MINimum|MAXimum]

:SOURce[1]:DELay <NRf|MINimum|MAXimum|DEFault>

:SOURce[1]:DELay? [MINimum|MAXimum|DEFault]

:SOURce[1]:DELay:AUTO <Bool>

:SOURce[1]:DELay:AUTO?

:SOURce[1]:FUNCtion[:MODE]?

:SOURce[1]:LIST:CURRent <NRf>{,<NRf>}..

:SOURce[1]:LIST:CURRent?

:SOURce[1]:LIST:CURRent:APPend <NRf>{,<NRf>}..

:SOURce[1]:LIST:CURRent:POINts?

:SOURce[1]:LIST:CURRent:STARt <NRf>

:SOURce[1]:LIST:CURRent:STARt?

:SOURce[1]:LIST:VOLTage <NRf>{,<NRf>}..

:SOURce[1]:LIST:VOLTage?

:SOURce[1]:LIST:VOLTage:APPend <NRf>{,<NRf>}..

:SOURce[1]:LIST:VOLTage:POINts?

:SOURce[1]:LIST:VOLTage:STARt <NRf>

:SOURce[1]:LIST:VOLTage:STARt?

:SOURce[1]:SWEep:DIRection <UP|DOWN>

:SOURce[1]:SWEep:DIRection?

:SOURce[1]:SWEep:POINts <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:SWEep:POINts? [DEFault|MINimum|MAXimum]

:SOURce[1]:SWEep:RANGing <BEST|AUTO|FIXed>

:SOURce[1]:SWEep:RANGing?

:SOURce[1]:SWEep:SPACing <LINear|LOGarithmic>

:SOURce[1]:SWEep:SPACing?

:SOURce[1]:VOLTage:CENTer <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage:CENTer? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage[:LEVel][:IMMediate][:AMPLitude] <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage[:LEVel][:IMMediate][:AMPLitude]? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage[:LEVel]:TRIGgered[:AMPLitude] <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage:MODE <FIXed|LIST|SWEep>

:SOURce[1]:VOLTage:MODE?

:SOURce[1]:VOLTage:PROTection[:LEVel] <NRf|NONE|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage:PROTection[:LEVel]? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage:PROTection:TRIPped?

:SOURce[1]:VOLTage:RANGe <NRf|DEFault|MINimum|MAXimum|UP|DOWN>

:SOURce[1]:VOLTage:RANGe? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage:RANGe:AUTO <Bool>

:SOURce[1]:VOLTage:RANGe:AUTO?

:SOURce[1]:VOLTage:SPAN <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage:SPAN? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage:STARt <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage:STARt? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage:STEP <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage:STEP? [DEFault|MINimum|MAXimum]

:SOURce[1]:VOLTage:STOP <NRf|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage:STOP? [DEFault|MINimum|MAXimum]

:SOURce2:BSIZe <3|4>

:SOURce2:BSIZe?

:SOURce2:CLEar:AUTO <Bool>

:SOURce2:CLEar:AUTO?

:SOURce2:CLEar:AUTO:DELay <NRf|DEFault|MINimum|MAXimum>

:SOURce2:CLEar:AUTO:DELay? [DEFault|MINimum|MAXimum]

:SOURce2:CLEar[:IMMEdiate]

:SOURce2:TTL[:LEVel]:ACTual?

:SOURce2:TTL[:LEVel][:DEFault] <NRf|NDN>

:SOURce2:TTL[:LEVel][:DEFault]?

:SOURce2:TTL4:BSTate <Bool>

:SOURce2:TTL4:BSTate?

:SOURce2:TTL4:MODE <EOTest|BUSY>

:SOURce2:TTL4:MODE?

:STATus:MEASurement:CONDition?

:STATus:MEASurement:ENABle <NDN|NRf>

:STATus:MEASurement:ENABle?

:STATus:MEASurement[:EVENt]?

:STATus:OPERation:CONDition?

:STATus:OPERation:ENABle <NDN|NRf>

:STATus:OPERation:ENABle?

:STATus:OPERation[:EVENt]?

:STATus:PRESet

:STATus:QUEStionable:CONDition?

:STATus:QUEStionable:ENABle <NDN|NRf>

:STATus:QUEStionable:ENABle?

:STATus:QUEStionable[:EVENt]?

:STATus:QUEue:CLEar

:STATus:QUEue[:NEXT]?

:SYSTem:BEEPer:STATe <Bool>

:SYSTem:BEEPer:STATe?

:SYSTem:CLEar

:SYSTem:ERRor:ALL?

:SYSTem:ERRor:CODE:ALL?

:SYSTem:ERRor:CODE[:NEXT]?

:SYSTem:ERRor:COUNt?

:SYSTem:ERRor[:NEXT]?

:SYSTem:LFRequency <50|60>

:SYSTem:LFRequency?

:SYSTem:POSetup <RST|PRESet|SAV0|SAV1|SAV2|SAV3|SAV4>

:SYSTem:POSetup?

:SYSTem:PRESet

:SYSTem:RSENse <Bool>

:SYSTem:RSENse?

:SYSTem:TIME?

:SYSTem:TIME:RESet

:SYSTem:TIME:RESet:AUTO <Bool>

:SYSTem:TIME:RESet:AUTO?

:SYSTem:VERSion?

:TRACe:CLEar

:TRACe:DATA?

:TRACe:FEED <SENSe[1]|CALCulate[1]|CALCulate2>

:TRACe:FEED?

:TRACe:FEED:CONTrol <NEXT|NEVer>

:TRACe:FEED:CONTrol?

:TRACe:FREE?

:TRACe:POINts <NR1|MINimum|MAXimum|DEFault>

:TRACe:POINts? [MINimum|MAXimum|DEFault]

:TRACe:POINts:ACTual?

:TRACe:TSTamp:FORMat <ABSolute|DELTa>

:TRACe:TSTamp:FORMat?

:TRIGger[:SEQuence[1]]:COUNt <NRf|DEFault|MINimum|MAXimum>

:TRIGger[:SEQuence[1]]:COUNt? [DEFault|MINimum|MAXimum]

:TRIGger[:SEQuence[1]]:DELay <NRf|DEFault|MINimum|MAXimum>

:TRIGger[:SEQuence[1]]:DELay? [DEFault|MINimum|MAXimum]

:TRIGger[:SEQuence[1]]:SOURce?

:TRIGger[:SEQuence[1]][:TCONfigure]:DIRection <SOURce|ACCeptor >

:TRIGger[:SEQuence[1]][:TCONfigure]:DIRection?

5.2 Standard Commands Partially Supported by SMM3000X

Parameters may be changed:

:SYSTem:BEEPer[:IMMediate] <NRf>

Parameters TLINk and BSTest are not effective. No error is reported:

:ARM[:SEQuence[1]][:LAYer[1]]:SOURce<IMMediate|TLINk|TIMer|MANual|BUS|NSTest|PSTest|BSTest>

:TRIGger[:SEQuence[1]]:SOURce<IMMediate|TLINk|TIMer|MANual|BUS|NSTest|PSTest|BSTest>

Parameters are ignored. Cause no action or response:

:ROUTe:TERMinals <FRONt|REAR>

[:SENSe[1]]:AVERage:COUNt <NRf|DEFault|MINimum|MAXimum>

[:SENSe[1]]:AVERage:TCONtrol <REPeat|MOVing>

:SOURce[1]:PULSe:DELay <NRf>

:SOURce[1]:PULSe:DELay?

:SOURce[1]:PULSe:WIDTh <NRf>

:SOURce[1]:PULSe:WIDTh?

:SYSTem:AZERo:CACHing:REFResh

:SYSTem:AZERo:CACHing:RESet

:SYSTem:AZERo:CACHing[:STATe] <Bool>

:SYSTem:AZERo:STATe <Bool>

:SYSTem:CCHeck:RESistance <NRf>

:SYSTem:GUARd <OHMS|CABLe>

:SYSTem:KEY <NR1>

:SYSTem:LOCal

:SYSTem:MEMOry:INITialize

:SYSTem:MEP:HOLDoff

:SYSTem:RWLock <Bool>

:TRIGger:CLEAr

:TRIGger:SEQuence2:SOURce <name>

:TRIGger:SEQuence2:TOUT <NRf|DEFault|MINimum|MAXimum>

Always returns the specified value:

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:ILINe? (returns 1)

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:OLINe? (returns 2)

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:OUTPut? (returns NONE)

:CALCulate2:CLIMits:FAIL:SMLocation? (returns NEXT)

:CALCulate2:CLIMits:PASS:SMLocation? (returns NEXT)

:DISPlay[:WINDow[1]]:ATTRibutes? (returns 20 zeros)

:DISPlay:WINDow2:ATTRibutes? (returns 32 zeros)

:OUTPut[1]:INTerlock:STATe? (returns 1)

:ROUTe:TERMinals? (returns FRON)

[:SENSe[1]]:AVERage:COUNt? [DEFault|MINimum|MAXimum] (returns 10)

[:SENSe[1]]:AVERage:TCONtrol? (retruns REP)

[:SENSe[1]]:AVERage:STATe? (returns 1)

[:SENSe[1]]:CURRent[:DC]:PROTection:RSYNchronize? (returns 1)

[:SENSe[1]]:VOLTage[:DC]:PROTection:RSYNchronize? (returns 1)

:SOURce[1]:CLEar:AUTO:MODE? (returns TCO)

:SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACtor? (returns 1.0)

:SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACtor:STATe? (returns 0)

:SOURce[1]:FUNCtion:SHAPe? (returns DC)

:SOURce[1]:MEMory:POINts? (returns 1)

:SOURce[1]:MEMory:STARt? (returns 1)

:SOURce[1]:SOAK? (returns 0)

:SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACtor? (returns 1.0)

:SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACtor:STATe? (returns 0)

:STATus:QUEue:DISable? (returns (+0))

:STATus:QUEue:ENABle? (returns (-440:-100,+111:+954))

:SYSTem:AZERo:CACHing:NPLCycles? (returns 0)

:SYSTem:AZERo:CACHing[:STATe]? (returns 0)

:SYSTem:AZERo:STATe? (returns 0)

:SYSTem:CCHeck? (returns 0)

:SYSTem:CCHeck:RESistance? (returns 50)

:SYSTem:GUARd? (returns CABL)

:SYSTem:KEY? (returns 0)

:SYSTem:LFRequency:AUTO? (returns 0)

:SYSTem:MEP[:STATe]? (returns 1)

:SYSTem:MEP:HOLDoff? (returns 0)

:SYSTem:RCMode? (returns SING)

:SYSTem:RWLock? (returns 0)

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:ILINe? (returns 1)

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:INPut? (returns NONE)

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:OLINe? (returns 2)

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:OUTPut? (returns NONE)

:TRIGger:SEQuence2:SOURce? (returns IMM)

:TRIGger:SEQuence2:TOUT? [DEFault|MINimum|MAXimum] (returns 0.0)

5.3 Standard Commands not Supported by SMM3000X

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:ILINe <1|2|3|4>

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:OLINe <1|2|3|4>

:ARM[:SEQuence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:OUTPut <TENTer|TEXit|NONE>

:CALCulate2:CLIMits:FAIL:SMLocation <NR1>|NEXT

:CALCulate2:CLIMits:PASS:SMLocation <NR1>|NEXT

:DISPlay[:WINDow[1]]:DATA?

:DISPlay:WINDow2:DATA?

:OUTPut[1]:INTerlock:STATe <Bool>

[:SENSe[1]]:AVERage:STATe <Bool>

[:SENSe[1]]:CURRent[:DC]:PROTection:RSYNchronize <Bool>

[:SENSe[1]]:VOLTage[:DC]:PROTection:RSYNchronize <Bool>

:SOURce[1]:CLEar:AUTO:MODE <ALWays|TCOunt>

:SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACtor <NRf>

:SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACtor:STATe <Bool>

:SOURce[1]:FUNCtion[:MODE] <VOLTage|CURRent|MEMory>

:SOURce[1]:FUNCtion:SHAPe <DC|PULSe>

:SOURce[1]:MEMory:POINts <NR1>

:SOURce[1]:MEMory:RECall <NR1>

:SOURce[1]:MEMory:SAVE <NR1>

:SOURce[1]:MEMory:STARt <NR1>

:SOURce[1]:SOAK <NRf>

:SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACtor <NRf>

:SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACtor:STATe <Bool>

:STATus:QUEue:DISable (NR1 list)

:STATus:QUEue:ENABle (NR1 list)

:SYSTem:CCHeck ON|OFF

:SYSTem:LFRequency:AUTO <Bool>

:SYSTem:MEP[:STATe]

:SYSTem:RCMode <SINGle|MULTiple>

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:ILINe <1|2|3|4>

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:INPut <SOURce|DELay|SENSe|NONE>

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:OLINe <1|2|3|4>

:TRIGger[:SEQuence[1]][:TCONfigure][:ASYNchronous]:OUTPut<SOURce|DELay|SENSe|NONE>

About SIGLENT

SIGLENT is an international high-tech company, concentrating on R&D, sales, production and services of electronic test & measurement instruments.

SIGLENT first began developing digital oscilloscopes independently in 2002. After more than a decade of continuous development, SIGLENT has extended its product line to include digital oscilloscopes, isolated handheld oscilloscopes, function/arbitrary waveform generators, RF/MW signal generators, spectrum analyzers, vector network analyzers, digital multimeters, DC power supplies, electronic loads and other general purpose test instrumentation. Since its first oscilloscope was launched in 2005, SIGLENT has become the fastest growing manufacturer of digital oscilloscopes. We firmly believe that today SIGLENT is the best value in electronic test & measurement.

Headquarters:

SIGLENT Technologies Co., Ltd
Add: Bldg No.4 & No.5, Antongda Industrial Zone,
3rd Liuxian Road, Bao'an District,
Shenzhen, 518101, China
Tel: + 86 755 3688 7876
Fax: + 86 755 3359 1582
Email: sales@siglent.com
Website: int.siglent.com

North America:

SIGLENT Technologies NA, Inc
Add: 6557 Cochran Rd Solon, Ohio 44139
Tel: 440-398-5800
Toll Free:877-515-5551
Fax: 440-399-1211
Email: support@siglentna.com
Website: www.siglentna.com

Europe:

SIGLENT Technologies Germany GmbH
Add: Staetzlinger Str. 70
86165 Augsburg, Germany
Tel: +49(0)-821-666 0 111 0
Fax: +49(0)-821-666 0 111 22
Email: info-eu@siglent.com
Website: www.siglenteu.com

Malaysia:

SIGLENT Technologies (M) Sdn.Bhd
Add: NO.6 Lorong Jelawat 4
Kawasan Perusahaan Seberang Jaya
13700, Perai Pulau Pinang
Tel: 006-04-3998964
Email: sales@siglent.com
Website: int.siglent.com

Follow us on
Facebook: SiglentTech