Programming Manual

DAQ3120

Benchtop Data Acquisition System

bkprecision.com

BK PRECISION

Contents

About this Manual

This manual describes how to use the Standard Commands for Programming Instruments (SCPI) to communicate with the DAQ3120 Series.

1.1 Intended Audience

This document is designed for instrument programmers tasked with creating SCPI-based programs for the DAQ3120 Series.

1.2 Related Documents

Refer to the following documents for more information:

- DAQ3120 Series User's Manual. This manual describes the operation of the DAQ3120 Series.
- Standard Commands for Programming Instruments (SCPI), Volume1-4, Version 1990.0 May 1999, SCPI Consortium, 2515 Camino del Rio South, Suite 340, San Diego, Ca 92108.
- IEEE Std 488.2-1992, The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017-2394, USA (ISBN 1-55937-238-9)

Syntax Convention

2.1 Introduction

SCPI (Standard Commands for Programmable Instruments) serves as a universal programming language designed for electronic test and measurement instruments. It is grounded in the IEEE 488.1 and IEEE 488.2 standards. The DAQ3120 series aligns with the SCPI language and incorporates the IEEE 488.2 STD status structure.

The commands can be issued over VISA or socket using TCP port 5025.

2.1.1 Types of SCPI Messages

In order to program an DAQ3120 instrument, it is necessary to create a program message. This message comprises one or more appropriately formatted SCPI commands transmitted from the controller to the DAQ3120 instrument. The program message, which can be sent at any time, requests the instrument to execute a specific action or provide data or status information. These requests are also referred to as queries.

Upon receiving a query, the DAQ3120 instrument responds by sending a response message back to the controller. This response message contains data formatted in a specific SCPI format.

The following documents provide more information about SCPI programming:

- Standard Commands for Programming Instruments (SCPI), Volume1-4, Version 1990.0 May 1999, SCPI Consortium, 2515 Camino del Rio South, Suite 340, San Diego, Ca 92108.
- IEEE Std 488.2-1992, The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017-2394, USA (ISBN 1-55937-238-9)

2.2 Types of SCPI Commands

Two types of SCPI commands are available: common commands, described below, and device-specific subsystem commands.

2.2.1 Common Commands

Common SCPI commands, as defined by IEEE 488.2, are responsible for controlling and managing generic system functions like reset, self-test, configuration storage, and device identification. Typically, common commands start with an asterisk (*), have a length of four to five characters, and may involve one or more parameters. The command keyword is separated from the initial parameter by a space. Multiple commands can be separated using a semicolon (;), as demonstrated below:

*RST; *CLS; *ESE 32; *OPC?

Refer to **Table 2.1** for a summary of these common SCPI commands applicable to programming the DAQ3120 series. For a detailed description of these commands, consult **Chapter 5**.

Command	Description
*CLS	Clears all Event Registers summarized in the status byte.
*ESE?	Returns an <NR1>, representing the value of the Standard Event Status Enable Register. Reading the value of the register will result in its clearance.
*ESR?	Returns an <NR1>, representing the value of the Standard Event Status Register. Reading the value of the register will result in its clearance.
*IDN?	Returns the unique identification string of the instrument.
*OPC?	Returns a "+1", when all pending selected device operations have been finished.
*RST	The Reset command performs a device reset. The Reset command is the third level of reset in a three-level reset strategy.
*SRE?	The Service Request Enable query allows the programmer to determine the current contents of the Service Request Enable Register.
*STB?	Returns the current binary value of the Status Byte Register.

Table 2.1 Common SCPI Commands

2.3 Syntax of Program Messages

A program message consist of one or more properly formatted SCPI commands, a parameter (if necessary, and a terminator sent from the controller to the DAQ3120 instrument to request some action or to query the instrument for a response.

Figure 2.3 shows the syntax of a program message:

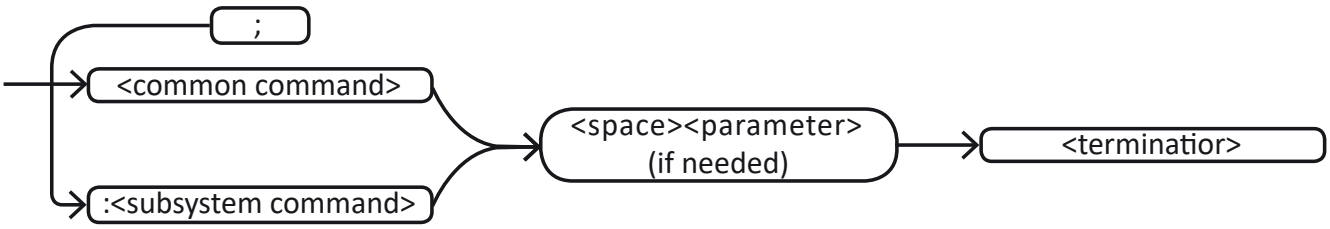

Figure 2.1 Syntax of Program Messages

A semicolon (;) is used to link commands from different groups.

The termination character marks the end of a command line. The following messages conform to the IEEE 488.2 standard: LF, CR, CR+LF, LF+CR.

The DAQ3120 uses the linefeed (LF) terminator.

SCPI Data Types

SCPI defines various data types for use in program messages and response messages.

The MPS series uses the following subset of SCPI data types:

This section summarizes these data types. Refer to the SCPI standards document for more information about these data types.

3.1 Character Data Types

If a command parameter takes data type, a specific number of settings are allowed for the parameter.

Example In the command DISPlay:PAGE the user can specify one of the following character data types:

{ MEAS | MANUMEM | MANUCOMM | PROGMEM1 | PROGMEM2 | PROGMEM3 | PROGCOMM | SYSTENV | SYSTCOM | SYSTTOOL | INTF | EXTF}

Character data types have the following characteristics:

- Can be expressed in either the short or long form, while response messages return them exclusively in the short form.
- Are case insensitive in program messages but in response messages are standardized to uppercase.
- Must have a specific length.

3.2 <NR1> Value Data Type

The data type <NR1> is utilized to indicate zero, positive, and negative integer values, including optional signs.

The following values are examples of the <NR1> data types:

0 100 -10

3.3 <NR2> Value Data Type

The data type <NR2> is utilized to indicate zero, positive, and positive and negative decimal values, including optional signs and decimal points.

The difference between <NR1> and <NR2> is the explicit decimal point.

The following values are examples of the <NR2> data types:

200.50 100.0 0.0

NOTICE 0 is a special case and redundant decimal points are ignored.

3.4 <NR3> Value Data Type

The data type <NR3> is utilized to indicate a floating point with exponent.

The following values are examples of the <NR3> data types:

3e-1 1e+3 4.5e-1

NOTICE 0 is a special case and redundant decimal points are ignored.

3.5 <NRf> Value Data Type

The <NRf> data type is employed to define floating-point values. These values encompass digits with an implied decimal point, an explicit decimal point, or an explicit decimal point along with an exponent.

The following values are examples of the <NRf> data types:

200 15.000e-3 0.015

3.6 Boolean Data Type

A Boolean data type for a parameter and response represents a single binary condition that is either True or False. Boolean values are defined as follows:

- **0 or OFF** : Indicates that the condition is False.
- **1 or ON** : Indicates that the condition is True.

NOTICE The characters OFF and ON are not case sensitive.

3.7 MIN Data Type

For commands, this sets the setting to its minimum value. This parameter can be used as a substitute for any numerical parameter where applicable.

For queries, it returns the lowest permissible value for the specified setting.

3.8 MAX Data Type

For commands, this sets the setting to its maximum value. This parameter can be used as a substitute for any numerical parameter where applicable.

For queries, it returns the highest permissible value for the specified setting.

3.9 DEF Data Type

For commands, this sets the setting to its default value. This parameter can be used as a substitute for any numerical parameter where applicable.

For queries, it returns the default value permitted for the specified setting.

About Commands & Queries

This section lists and describes the remote control commands and queries recognized by the instrument. All commands and queries can be executed in either local or remote state.

The description, command syntax, query syntax, example and respond can be found in a section. The commands are given in both long and short form. All examples are shown in short form. Queries perform actions such as obtaining information, and are recognized by the question mark (?) following the header.

4.1 How They are Listed

The commands are listed by subsystem and alphabetical order according to their short form.

4.2 How They are Described

In the descriptions themselves, a brief explanation of the function performed is given. This is followed by a presentation of the formal syntax, with the header given in Upper-and-Lower-Case characters and the short form derived from it in ALL UPPER-CASE characters. Where applicable, the syntax of the query is given with the format of its response.

4.3 When can they be used?

The commands and queries listed here can be used for the HVL seires.

4.4 Command Notation

The following notation is used in the commands:

< > Angular brackets enclose words that are used as placeholders, of which there are two types: the header path and the data parameter of a command.

:= 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 enclose a list of choices, one of which one must be made.

[ ] Square brackets enclose optional items.

… An ellipsis indicates that the items both to its left and right may be repeated a number of times.

Common SCPI Commands

IEEE standard defines the common commands used for querying the basic inSyntaxion of the instrument or executing basic operations. These commands usually start with "*" and the length of the keywords of the command is usually 3 characters.

5.1 *CLS

Description This command clears all status data structures in a device. For a device which minimally complies with SCPI, these registers are:

SESR OPERation Status	**(IEEE 488.2)**
Register QUEStionable	**(SCPI)**
Status Register Error/Event	**(SCPI)**
Queue	**(SCPI)**

Execution of *CLS shall also clear any additional status data structures implemented in the device. The corresponding enable registers are unaffected.

***CLS** forces the device into OCIS and OQIS without setting the **No Operation Pending** flag TRUE and without setting the OPC bit of the SESR TRUE and without placing a “1” into the Output Queue.

For example, suppose a device implements **INITiate[:IMMediate]** as an overlapped command. Assuming that the trigger model is programmed so that it will eventually return to the IDLE state, and that INITiate[:IMMediate] takes longer to execute than *OPC, sending these commands to this device:

INITiate;*OPC

results in initiating the trigger model and, after some time, setting the OPC bit in the SESR. However, sending these commands:

INITiate;*OPC;*CLS

still initiates the trigger model. Since the operation is still pending when the device executes ***CLS**, the device does not set the OPC bit until it executes another ***OPC** command.

Example *CLS

5.2 *ESE

Description The Standard Event Status Enable command sets the Standard Event Status Enable Register bits

Syntax **Command** *ESE?

5.3 *ESR?

Description Query the Standard Event Status Register. Once a bit is set, it remains set until cleared by a ***CLS** (clear status) command or queried by this command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

Syntax **Query** *ESR?

Example **Query** *ESR? Returns the standard event register

Related *CLS, *ESE

5.4 *IDN?

Description The *IDN? query causes the instrument to identify itself. The response comprises manufacturer, model, serial number, software version and firmware version.

Syntax **Query** *IDN?

Response *IDN, <device id>,<model>,<serial number>, <software version>, <hardware version>.

<device id>:=“BK” is used to identify instrument.

<model>:= A model identifier less than 14 characters will contain the model number.

<serial number>:= Number that uniquely identifies the instrument.

<firmware version>:= Firmware revision number.

<hardware version>:= Hardware revision number.

Example *IDN?

Returns: B&K Precision,MPS1102,XXXXXXXXX,0.90-1.00

5.5 *OPC

Description The Operation Complete command causes the device to generate the operation complete message in the Standard Event Status Register when all pending selected device operations have been finished.

Syntax **Command** *OPC

5.6 *OPC?

Description The Operation Complete query places an ASCII character “1” into the device’s Output Queue when all pending selected device operations have been finished.

Syntax **Query** *OPC?

Example **Query** *OPC? Query the operation complete register.

5.7 *PSC

Description The Power-On Status Clear command controls the automatic power-on clearing of the Service Request Enable Register, the Standard Event Status Enable Register, the Parallel Poll Enable Register, and device-specific event enable registers. This command may also affect the clearing of other status registers.

Syntax **Command** *PSC

5.8 *PSC?

Description The Power-On Status Clear query allows the programmer to query the device’s power-on-status-clear flag. A returned value of zero indicates that the Standard Event Status Enable Register, Service Request Enable Register, and the Parallel Poll Enable Register will retain their status when power is restored to the MPS. A returned value of one indicates that the registers listed above will be cleared when power is restored to the device.

Syntax **Query** *PSC?

Example **Query** *PSC?

5.9 *RCL

Description The *RCL command restores the current settings of a device from a copy stored in local memory.

Syntax **Command** *RCL <NR1>

Parameters <**NR1**> := {0 to 9}

Example **Command** *RCL 1 **Query** *RCL?

5.10 *SAV

Description The *SAV command stores the current settings of the device in local memory.

Syntax **Command** *SAV <NR1>

Parameters <**NR1**> := {0 to 9}

Example **Command** *SAV 2 **Query** *SAV?

5.11 *RST

Description The Reset command performs a device reset. The Reset command is the third level of reset in a three-level reset strategy.

Syntax **Query** *RST?

Example **Query** *RST?

5.12 *SRE

Description The Service Request Enable command sets the Service Request Enable Register bits.

Syntax **Command** *SRE

Example **Query** *SRE

5.13 *SRE?

Description The Service Request Enable query allows the programmer to determine the current contents of the Service Request Enable Register.

Syntax **Query** *SRE?

Example **Query** *SRE?

5.14 *STB?

Description The Read Status Byte query allows the programmer to read the status byte and Master Summary Status bit.

Syntax **Query** *STB?

Example **Query** *STB?

5.15 *TRG

Description The *TRG command generates an immediate trigger when the trigger source is set to BUS

Syntax **Command** *TRG

Example **Command** *TRG

5.16 *TST?

Description The self-test query causes an internal self-test and places a response into the Output Queue indicating whether or not the device completed the self-test without any detected errors.

Syntax **Query** *TST?

Example **Query** *TST?

5.17 *WAI

Description The Wait-to-Continue command prevents the device from executing any further commands or queries until the nooperation-pending flag is TRUE.

Syntax **Command** *WAI

Example **Command** *WAI

Other Commands

6.1 ABORt

Description Cancels an ongoing measurement from a scan, returning the instrument to the trigger idle state.

- If a scan is active when the command is issued, the scan will terminate immediately and cannot be resumed.
- Initiating a new scan will clear all readings stored in the reading memory.

Syntax **Command** ABORt

Parameters none

Example **Command** ABORt

6.2 FETCh?

Description Waits for measurements to finish and transfers all available data to the instrument's output buffer.

- The query does not clear measurements from the reading memory, allowing you to send the query multiple times to retrieve the same data.
- Reading memory can store up to 100,000 readings, each automatically time-stamped.
- If memory overflows, the oldest readings are overwritten by new ones, ensuring the most recent data is preserved. No error is triggered, but the Reading Memory Overflow bit (bit 12) is set in the Questionable Data Register condition register.
- Starting a new scan clears all readings, including alarm data, from the previous measurement in the reading memory. As a result, the reading memory always contains data from the most recent scan.

The returned value is in the **<NRf>** data type.

Syntax **Query** FETCh?

Parameters none

Example **Query** FETCh?

6.3 INITiate[:IMMediate]

Description The command transitions the triggering system from **idle** to **wait-for-trigger** and clears the previous measurements from the reading memory. Measurements start when the specified trigger conditions are met after receiving the command.

Storing measurements in the reading memory with **INITiate** is faster than sending measurements directly to the instrument's output buffer using **READ?**, provided **FETCh?** is not sent until the operation is complete.

- The command is an overlapped command, meaning you can send other commands that do not interfere with the measurements while it is executing.
- To retrieve measurements from the reading memory, use **FETCh?**. Use DATA:REMove? or **R?** to read and erase all or a portion of the stored measurements.
- Once a scan is initiated, attempting to change measurement parameters (**CONFigure** and SENSe commands) or trigger configurations (TRIGger commands) will result in an error.
- Use the **ABORt** command to return the instrument to the idle state.

Syntax **Command** INITiate[:IMMediate]

Parameters none

Example **Command** INIT

6.4 INSTrument:DMM <boolean>

Description The command enables/disables the internal DMM.

The query returns the insternal DMM's state. The returned value is in the **<boolean>** data type.

NOTICE Changing the state of the internal DMM triggers the instrument to perform a Factory Reset (*RST command).

Syntax **Command** INSTrument:DMM **<boolean>**

Query INSTrument:DMM?

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

Mode	Description
0	OFF
1	ON

Table 6.1 Internal DMM State

Example **Command** INST:DMM ON

Query INST:DMM?

6.5 R? <NR1>

Description Reads and erases measurements from the reading memory up to the specified limit, beginning with the oldest measurement first.

- The **R?** and DATA:REMove? queries can be used during a long series of readings to periodically remove readings from memory that might otherwise cause the reading memory to overflow. **R?** does not wait for all readings to complete; it sends the readings that are complete at the time the instrument receives the command.
- Use **READ?** or **FETCh?** if you want the instrument to wait until all readings are complete before sending them.
- No error is generated if the reading memory contains fewer readings than requested. In this case, all available readings in memory are read and deleted.

The query returns **#2** indicating that the next two digits specify the length of the returned memory string in characters. In the example below , the two digits following **#2** are **79** meaning the rest of the string is 79 characters long.

For example;
#279-1.12876922E-04,-1.13123674E-04,-1.16846231E-04,-1.13485152E-04,-1.13667621E-04

Syntax **Query** R? <NR1>

Parameters <**NR1**> := { 1 to 100,000 }

NOTICE If the stop value is not specify all measurements are read and erased.

Example **Query** R? 100

6.6 READ?

Description Transitions the triggering system from **idle** to **wait-for-trigger**. Scanning begins when the specified trigger conditions are met after receiving the **READ?** command. The readings are then stored in the reading memory and sent to the instrument's output buffer.

The query does not return the unit or count number of the reading. Sending **READ?** is equivalent to sending **INITiate:IMMediate** followed immediately by **FETCh?**.

Syntax **Query** READ?

Parameters none

Example **Query** READ?

6.7 TIME:SYNC:SERVer

Description Sets or retrieves the server source for time synchronization.

Syntax **Command** TIME:SYNC:SERVer <character>

Query TIME:SYNC:SERVer?

Parameters <**character**> := { max length = 24 characters}

Example **Command** TIME:SYNC:SERV us.pool.ntp.org
Query TIME:SYNC:SERV?

6.8 UNIT:TEMPerature <character>[,(@<NR1>)]

Description Specifies the units for temperature measurements as °C (Celsius), °F (Fahrenheit), or K (Kelvin).

- If the corresponding channels are not configured for temperature measurements before sending this command, the instrument will return an error message.
- The **CONFigure** and MEASure? commands automatically select °C.

Syntax

Command	UNIT:TEMPerature <character>
Query	UNIT:TEMPerature?

Parameters

<**character**> := { C | F | K }
<**NR1**> := { 1 to 420 }

Example

Command	UNIT:TEMP F,(@1,2,3)
Query	UNIT:TEMP?

Calculate Subsystem

The CALCulate subsystem performs post-acquisition data processing, operating on data acquired by the SENSe subsystem. While SENSe handles data collection, CALCulate transforms the data for output to a display or bus. When a measurement is triggered, the SENSe subsystem collects data, which is processed by CALCulate as configured. CALCulate can also be reconfigured to derive new results from the same data without reacquisition.

7.1 CALCulate:AVERage:ALL? @<NR1>

Description The query returns all the statistic calculation values of the specified channel. The values are returned in the **<NRf>**, in the following order: average, standard deviation, minimum, maximum, count)

Syntax **Query** CALCulate:AVERage:ALL? @**<NR1>**

Parameters <**NR1**> := { 1 to 420 }

Example **Query** CALC:AVER:ALL? @101

7.2 CALCulate:AVERage:{<character>}? @<NR1>

Description The query returns the average, maximum, minimum, peak-to-peak or standard deviation recorded values. The values are returned in the **<NRf>**.

Syntax **Query** CALCulate:AVERage:{**<character>**}? @**<NR1>**

Parameters <**character**> := { AVERage | MAXimum | MINimum | PTPeak | SDEViation }

<**NR1**> := { 1 to 420 }

NOTICE If the **(@<NR1>** parameter is omitted, the query returns the count for all channels in the currently defined scan list.

Example **Query** CALC:AVER:MAX? @100,322,420

7.3 CALCulate:AVERage:CLEar @<NR1>

Description Clears all statistical calculation values for the selected channels, including average, count, maximum, minimum, peak-to-peak, and standard deviation.

Syntax **Syntax**
qquad CALCulate:AVERage:CLEar @**<NR1>**

Parameters <**NR1**> := { 1 to 420 }

Example **Query** CALC:AVER:CLEA @100,322,420

7.4 CALCulate:AVERage:COUNt? @<NR1>

Description Returns the total number of recorded counts on each of the selected channels during the scan.

Syntax **Query**
quad
qquad CALCulate:AVERage:COUNt @**<NR1>**

Parameters <**NR1**> := { 1 to 420 }

Example **Query** CALC:AVER:COUN @100,322,420

7.5 CALCulate:AVERage:{<character>}:TIME? @<NR1>

Description Returns the time that the maximum or minimum reading was taken on the selected channels during the scan (in full time and date format).

Syntax **Query** CALCulate:AVERage:{**<character>**}:TIME? @**<NR1>**

Parameters <**character**> := { MAXimum| MINimum }

<**NR1**> := { 1 to 420 }

NOTICE If the **(@<NR1>** parameter is omitted, the query returns the average time for all channels in the currently defined scan list.

Example **Query** CALC:AVER:MAX:TIME? @100,322,420

7.6 CALCulate:LIMit:{<character>} <NRf>,@<NR1>

Description The instrument has four alarms which you can configure to alert you when a reading exceeds specified limits during a scan.

Syntax **Syntax** CALCulate:LIMit:{**<character>**} {**<NRf>**},@**<NR1>**

Query CALCulate:LIMit:{**<character>**}? @**<NR1>**

Parameters <**character**> := { LOWer | UPPer }

<**NRf**> := { -1.2E+09 to +1.2E+09 | MIN | MAX | DEF}

<**NR1**> := { 1 to 420 }

NOTICE The lower limit value must always be less than or equal to the upper limit.

Example **command** CALC:LIM:LOW 2@100,322,420

Query CALC:LIM:LOW? @100,322,420

7.7 CALCulate:LIMit:{<character>}:STATe <boolean>,@<NR1>

Description Enables/Disables the lower and upper alarm limits on the specified channels during a scan.

Syntax **Syntax** CALCulate:LIMit:{**<character>**}:STATe **<boolean>**,@**<NR1>**

Query CALCulate:LIMit:{**<character>**}:STATe? @**<NR1>**

Parameters <**character**> := { LOWer | UPPer }

<**boolean**> := { 0 | 1 | OFF | ON }

Mode	Description
0	OFF
1	ON

Table 7.1 Calculated Limit State

<**NR1**> := { 1 to 420 }

NOTICE The lower limit value must always be less than or equal to the upper limit.

Example **command** CALC:LIM:STAT ON,@100,322,420

Query CALC:LIM:STAT? @100,322,420

7.8 CALCulate:MATH

Description A computed channel performs mathematical operations on readings from measurement channels or other computed channels.

The data is returned in the <string> format.

Syntax **Syntax** CALCulate:MATH {<string>}

Query CALCulate:MATH? @<**NR1**>

Parameters

Computation type	Mathematical operation	(<expression>)
Basic math	Add	(@ch1+@ch2)
	Subtract	(@ ch1-@ ch2)
	Multiply	(@ch1*@ch2)
	Divide	(@ch1/@ch2)
	Power	(power(@ch1,2))
	Square root	(sqrt(@ch1))
	Reciprocal	(1/(@ch1))
Polynomial	Fifth order	(poly(@ch1, <n5>, <n4>, <n3>, <n2>, <n1>, <n0>)) where n = value of variable in each order
Statistics	Min	(min(@<ch_list>))
	Max	(max(@<ch_list>))
	Sum	(sum(@<ch_list>))
	Average	(avg(@<ch_list>))
	Standard deviation	(sdev(@<ch_list>))

Table 7.2 Functions

<**NR1**> := { 401 to 420 }

NOTICE The returned value is only supported on computed channels (channels 401 through 420).

Example **command** CALC:MATH (sqrt(@201)),(@402)

Query CALC:MATH? (@402)

7.9 CALCulate:SCALe:DB:REFerence <NRf>,@<NR1>

Description Sets the reference value for the dB function.

Syntax

Command CALCulate:SCALe:DB:REFerence {<NRf>},@<NR1>}

Query CALCulate:SCALe:DB:REFerence? @<NR1>

Parameters <NRf> := { (-2.0E+02 to +2.0E+02 | MIN | MAX | DEF}

<NR1> := { 1 to 420}

Example

Command CALC:SCAL:DB:REF 3,@2

Query CALC:SCAL:DB:REF?

7.10 CALCulate:SCALe:DBM:REFerence <NR1>,@<NR1>

Description Sets the reference value for the dBm function.

Syntax

Command CALCulate:SCALe:DBM:REFerence {<NR1>},@<NR1>}

Query CALCulate:SCALe:DBM:REFerence? @<NR1>

Parameters <NRf> := { (2, 4, 8, 16, 50, 75, 93, 110, 124, 125, 135, 150, 250, 300, 500, 600, 800, 900, 1000, 1200, 8000); DEF: 600}

<NR1> := { 1 to 420}

Example

Command CALC:SCAL:DBM:REF 2,(@2)

Query CALC:SCAL:DBM:REF? @2

7.11 CALCulate:SCALe:DECimal:POINt <character>,(@<NR1>

Description Under the Math function, the display of measured values changes based on either the fixed range setting (Range) or the auto range setting (Auto).

Syntax

Command CALCulate:SCALe:DECimal:POINt <character>,@<NR1>

Query CALCulate:SCALe:DECimal:POINt? @<NR1>

Parameters <character> := { AUTO | RANGe }

Example

Command CALC:SCAL:DEC:POIN AUTO,@1

Query CALC:SCAL:DEC:POIN? @1

7.12 CALCulate:SCALe:FUNCtion <character>,(@<NR1>

Description Sets or returns the advanced function.

NOTICE dB scaling function is only available when the measurement function on the specified channels sets to DCV orACV

Syntax
Command CALCulate:SCALe:FUNCtion **<character>**,[@**<NR1>**]
Query CALCulate:SCALe:FUNCtion? [@**<NR1>**]

Parameters <**character**> := { OFF | DB | DBM | SCALe | INV | PCT }

Example
Command CALC:SCAL:FUNC DB,@1
Query CALC:SCAL:FUNC? @1

7.13 CALCulate:SCALe:GAIN <NRf>,(@<NR1>

Description Sets or returns the scale factor M for math measurement.

Syntax
Command CALCulate:SCALe:GAIN **<NRf>**,[@**<NR1>**]
Query CALCulate:SCALe:GAIN? [@**<NR1>**]

Parameters <**NRf**> := { (-1.2E+09 to +1.2E+09 | MIN | MAX | DEF}

<**NR1**> := { 1 to 420}

Example
Command CALC:SCAL:GAIN 10,@1
Query CALC:SCAL:GAIN? @1

7.14 CALCulate:SCALe:OFFSet <NRf>,(@<NR1>

Description Sets or returns the offset factor B for math measurement.

Syntax
Command CALCulate:SCALe:OFFSet **<NRf>**,[@**<NR1>**]
Query CALCulate:SCALe:OFFSet? [@**<NR1>**]

Parameters <**NRf**> := { (-1.2E+09 to +1.2E+09 | MIN | MAX | DEF}

<**NR1**> := { 1 to 420}

Example **Command** CALC:SCAL:OFFS 10,@1
Query CALC:SCAL:OFFS? @1

7.15 CALCulate:SCALe:OFFSet:NULL <NRf>,(@<NR1>

Description Performs an instant null measurement on the specified channels and stores it as the offset (B) for future measurements.

Syntax **Command** CALCulate:SCALe:OFFSet:NULL @<**NR1**>

Parameters <**NR1**> := { 1 to 420}

Example **Command** CALC:SCAL:OFFS:NULL @100

7.16 CALCulate:SCALe:PERCent <NRf>,(@<NR1>

Description Sets or returns the reference value for the PCT function.

Syntax **Command** CALCulate:SCALe:PERCent <**NRf**>,@<**NR1**>
Query CALCulate:SCALe:PERCent? @<**NR1**>

Parameters <**NRf**> := { (-1.2E+09 to +1.2E+09 | MIN | MAX | DEF}
<**NR1**> := { 1 to 420}

Example **Command** CALC:SCAL:PERC 0.1,@100
Query CALC:SCAL:PERC? @100

7.17 CALCulate:SCALe:REFerence <NRf>,(@<NR1>

Description Sets or returns the reference value for the PCT function.

Syntax **Command** CALCulate:SCALe:REFerence <**NRf**>,@<**NR1**>
Query CALCulate:SCALe:REFerence? @<**NR1**>

Parameters <**NRf**> := { (-1.2E+09 to +1.2E+09 | MIN | MAX | DEF}
<**NR1**> := { 1 to 420}

Example **Command** CALC:SCAL:REF 1,@100
Query CALC:SCAL:REF? @100

7.18 CALCulate:SCALe:REFerence:AUTO <boolean>,(@<NR1>

Description Enables/Disables the automatic reference selection for the scaling functions.

Syntax **Command** CALCulate:SCALe:REFerence:AUTO **<boolean>**,@**<NR1>**

Query CALCulate:SCALe:REFerence:AUTO? @**<NR1>**

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

Mode	Description
0	OFF
1	ON

Table 7.3 Reference Auto State

NOTICE

ON: the first measurement made is used as the reference for all subsequent measurements, and automatic reference selection is disabled.

OFF: CALCulate:SCALe:DB:REFerence specifies the reference for DB scaling CALCulate:SCALe:REFerence specifies the reference for PCT scaling

Example **Command** CALC:SCAL:REF:AUTO ON,@100

Query CALC:SCAL:AUTO? @100

7.19 CALCulate:SCALe:REFerence:IMMediate

Description Makes an immediate reference measurement on PCT (%) and dB scaling functions and save the reference value for subsequent measurements.

NOTICE

Performs the regernce measurement on both PCT and dB scaling functions simultaneously.

Syntax **Command** CALCulate:SCALe:REFerence:IMMediate @**<NR1>**

Parameters none

Example **Command** CALC:SCAL:REF:IMMediate @100

7.20 CALCulate:SCALe:UNIT<boolean>,(@<NR1>

Description Enables or disables the scaling function.

Syntax **Command** CALCulate:SCALe:UNIT <**boolean**>,@<**NR1**>

Query CALCulate:SCALe:UNIT? @<**NR1**>

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

Mode	Description
0	OFF
1	ON

Table 7.4 Calculate Scale State

Example **Command** CALC:SCAL ON,@100
Query CALC:SCAL? @100

7.21 CALCulate:SCALe:UNIT <string>,(@<NR1>

Description Specifies the custom unit up to three characters (for example: RPM, PSI, or °C) for scaled measurements.

Syntax **Command** CALCulate:SCALe:UNIT <string>,@<**NR1**>

Query CALCulate:SCALe:UNIT? @<**NR1**>

Parameters <string> := { max length = 3 characters }

<**NR1**> := { 1 to 420}

Example **Command** CALC:SCAL:UNIT PSI,@100
Query CALC:SCAL:UNIT? @100

7.22 CALCulate:SCALe:UNIT:STATe <boolean>,(@<NR1>

Description Enables or disables displaying the unit string with measurements on the front panel when the scaling function is enabled.

Syntax **Command** CALCulate:SCALe:UNIT:STATe <string>,@<**NR1**>

Query CALCulate:SCALe:UNIT:STATe? @<**NR1**>

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

Mode	Description
0	OFF
1	ON

Table 7.5 Unit State

<**NR1**> := { 1 to 420}

Example **Command** CALC:SCAL:UNIT:STATe ON,@100
Query CALC:SCAL:UNIT:STATe? @100

Configure Subsystem

The Configure Subsystem sets up measurement parameters. This subsystem allows users to define the type of measurement and its associated settings without initiating an actual measurement.

8.1 CONFigure?

Description Returns the present configuration (function, range, and resolution) on the specified channels with a seriews of quoted strings.

Syntax **Query** CONFigure? @<**NR1**>

Parameters none

Example **Query** CONF? @1

8.2 CONFigure:CAPacitance <character>,@<NR1>

Description Configures the channels for capacitance measurements.

Syntax **command** CONFigure:CAPacitance <**character**>,@<**NR1**>

Parameters <**character**> := { 1nF|10nF|100nF|1µF| 10µF |100µF | MIN | MAX | DEF}

Example **command** CONF:CAP @100,200

8.3 CONFigure:CURRent{AC|DC}<character>,@<NR1>

Description Configures the channels for AC/DC current measurements.

NOTICE Autoranging (AUTO/DEFault), will generate an error if you specify a resolution because the instrument cannot accurately resolve the integration time (especially if the input continuously changes). If your application requires autoranging, specify DEFault for the resolution or omit the resolution altogether.

Syntax **command** CONFigure:CURRent:{AC|DC}? <**character**>,@<**NR1**>

Parameters <**character**> :=

AC: {100µA | 1mA | 10mA | 100mA | 2A; DEF: AUTO}

DC: {1µA | 10µA | 100µA | 1mA | 10mA | 100mA | 2A; DEF: AUTO}

Example **command**
CONF:CURR:AC 10e-2,(@100)

CONF:CURR:DC 10e-3,DEF,(@101)

8.4 CONFigure:DIODe @<CH List>

Description Configures the channels for diode measurements.

Syntax **command** CONFigure:DIODe? @<**CH List**>

Parameters <channel> := <**NR1**> := { 1 to 420}

Example **command** CONF:DIOD @100,200

8.5 CONFigure:FREQuency|PERiod <range>@<CH List>

Description Configures the channels for frequency and period measurements.

Syntax **command** CONFigure:FREQuency|PERiod <**range**>

Parameters <**range**> := **Frequency**: {3Hz to 300kHz; DEF: 20Hz }
Period: {3.33µs to 333.33ms; DEF: 50ms }

<**character**> := { 1 to 420}

Example **command**
CONF:FREQ 1e3,(@100)

CONF:PER AUTO,DEF,(@101)

8.6 CONFigure:{RESistance|FRESistance} <range>@<CH List>

Description Configures the channels for 2-Wire and 4-Wire resistance measurements.

NOTICE

Autoranging (AUTO or DEFault), will generate an error if you specify a <resolution> because the instrument cannot accurately resolve the integration time (especially if the input continuously changes). If your application requires autoranging, specify DEFault for the <resolution> or omit the <resolution> altogether.

Syntax **command** CONFigure:RESistance|FRESistance **<range>**

Parameters <**range**> := { 100Ω | 1kΩ | 10kΩ | 100kΩ | 1MΩ | 10MΩ | 100MΩ | 1GΩ); DEF:1kΩ }
{AUTO | MIN | MAX | DEF}

<**character**> := { 1 to 420}

Example **command**
CONF:RES 1e3,(@100)

CONF:RES AUTO,(@101)

8.7 CONFigures:STRain:{DIRect | FDIRect}

Description Configures the channels for direct 2-Wire and 4-Wire strain gage measurements.

Syntax **command** CONFigures:STRain:{DIRect | FDIRect} <gage_ohms><gage_factor><range><resolu

Parameters <**gage_ohms**> := { 80 to 1100Ω; DEF:120Ω }
<**gage_factor**> := { 0.5 to 5; DEF:5 }
<**range**> := { 100Ω | 1kΩ | 10kΩ | 100kΩ | 1MΩ | 10MΩ | 100MΩ | 1GΩ); DEF:1kΩ }
{AUTO | MIN | MAX | DEF}

<**character**> := { 1 to 420}

Example **command**
CONF:STR:DIR 100,1,(@100)

CONF:STR:FDIR 100,1,(@101)

8.8 CONFigure:STRain:{FULL | HALF}:BENDing

Description Configures the channels for full and half bending bridge strain gage measurements.

Syntax **command** CONFigure:STRain:{FULL | HALF}:BENDing <gage_factor>,<range>,<resolution>,(@

Parameters <**gage_factor**> := { 0.5 to 5; DEF:5 }
<**range**> := { 100 mV | 1 V | 10 V | 100 V | 600 V; DEF:AUTO }

<**ch_list**> := { 1 to 420}

Example **command**
CONF:STR:FULL:BEND 100,1,(@100)

CONF:STR:HALF:BEND 100,1,(@101)

8.9 CONFigure:STRain:{FULL | HALF}:POISson

Description Configures the channels for full and half poisson bridge strain gage measurements.

Syntax **command** CONFigure:STRain:{FULL | HALF}:POISson <gage_factor>,<poisson_ratio>,<range>

Parameters <**gage_factor**> := { 0.5 to 5; DEF:5 }
<**range**> := { 100 mV | 1 V | 10 V | 100 V | 600 V; DEF:AUTO }

<**ch_list**> := { 1 to 420}

Example **command**
CONF:STR:FULL:POIS (@100)

CONF:STR:HALF:POIS (@101)

8.10 CONFigure:STRain:{FULL | HALF}:BENDing:POISson

Description Configures the channels for full bending poisson bridge strain gage measurements.

Syntax **command** CONFigure:STRain:{FULL | HALF}:POISson <gage_factor>,<poisson_ratio>,<range>

Parameters <**gage_factor**> := { 0.5 to 5; DEF:5 }

<**poisson_ratio**> := { -0.9999 to 0.5; DEF: 0.3 }
<**range**> := { 100 mV | 1 V | 10 V | 100 V | 600 V; DEF:AUTO }

<**ch_list**> := { 1 to 420}

Example **command**
CONF:STR:FULL:BEND:POIS 0.5,0.1,(@101)

CONF:STR:HALF:BEND:POIS 0.5,0.1,(@101)

8.11 CONFigure:STRain:QUARter

Description Configures the channels for quarter bridge strain gage measurements.

Syntax **command** CONFigure:STRain:QUARter <gage_factor>,<range>,<resolution>,(@<ch_list>)

Parameters <**gage_factor**> := { 0.5 to 5; DEF:5 }

<**range**> := { 100 mV | 1 V | 10 V | 100 V | 600 V; DEF:AUTO }

<**ch_list**> := { 1 to 420}

Example **command**
CONF:STR:QUAR 1,(@101)

8.12 CONFigure:TEMPerature

Description Configures the channels for temperature measurements.

Syntax **command** CONFigure:TEMPerature <probe_type>,<type>,<resolution>,(@<ch_list>)

Parameters <**probe type**> := { TCouple | RTD | FRTD | THERmistor | FTHermistor }

<**type**> := **TCouple:** {B | E | J | K | N | R | S | T | USER ; DEF: JRTD}
FRTD : {PT100 | D100 | F100 | PT385 | PT3916 | USER ; DEF: PT100}
THERmistor / FTHermistor :{ 2.2kΩ| 5kΩ| 10kΩ| USER; DEF: 5kΩ}

<**ch_list**> := { 1 to 420}

Example **command**
CONF:TEMP TC,K,(@101)

8.13 CONFigure[:VOLTage]:{AC | DC}

Description Configures the channels for AC and DC voltage measurements

NOTICE

Autoranging (AUTO or DEFault), will generate an error if you specify a <resolution> because the instrument cannot accurately resolve the integration time (especially if the input continuously changes). If your application requires autoranging, specify DEFault for the <resolution> or omit the <resolution> altogether.

Syntax **command** CONFigure[:VOLTage]:AC | DC <range>,<resolution>,(@<ch_list>)

Parameters <**range**> :=
AC: (100mV | 1V | 10V | 100V | 400V); DEF:AUTO
DC: (100mV | 1V | 10V | 100V | 600V); DEF:AUTO

Example **command**
CONF:VOLT:DC 1,MAX,(@101)
CONF:VOLT:AC 10e-2,(@201)

Data Subsystem

The **DATA** subsystem in **SCPI** handles the transfer and formatting of binary or ASCII data between an instrument and a controller. It enables efficient data retrieval, storage, and management.

9.1 DATA:LAST?

Description Returns the most recent reading or readings taken on the selected channel during the scan.

NOTICE If no data is available for the specified channel, an error will be generated.

Syntax **Query** DATA:LAST? [<num_rdgs>,](@<ch_list>)

Parameters <**num_rdgs**> := { 1 to 1000 }

<**ch_list**> := { 1 to 420}

Example **query** DATA:LAST? 1,(@101)

9.2 DATA:POINts?

Description Returns the total number of readings currently saved in reading memory from a scan. Data is returned in the **<NR1>** format

NOTICE You can store up to 100,000 measurements in the reading memory.

Syntax **Query** DATA POINts?

Example **query** DATA:POIN?

9.3 DATA:POINts:EVENt:THReshold

Description Sets or returns the threshold for event number of measurement.

NOTICE When measurement numbers reach the set threshold, the Bit9 within the Operater Event Register (STATus:OPERation:EVENt) will be set as 1.
Once the Memory Threshold bit (bit 9 in the Standard Operation Event register) is set, it remains set until cleared by STATus:OPERation:EVENt? or *CLS.

Syntax **Command** DATA:POINts:EVENt:THReshold <num_rdgs>

Query DATA:POINts:EVENt:THReshold?

Parameters <**num_rdgs**> := { 1 to 100,000 }

Example **command** DATA:POIN:EVEN:THR 20

query DATA:POIN:EVEN:THR?

9.4 DATA:REMove?

Description Reads and erases up to `<num_rdgs>` measurements from memory, starting with the oldest.

Use **DATA:POINts?** to check available readings.

NOTICE

If **<num_rdgs>** exceeds available readings, an error occurs unless WAIT is specified.

DATA:REMove? and R? can free memory to prevent overflow; R? returns completed readings without waiting.

Syntax **Query** DATA:REMove? <num_rdgs>,[WAIT]

If memory overflows, old readings are overwritten without error, but bit 12 in the Questionable Data Register is set.

Parameters <**num_rdgs**> := { 1 to 100,000 }

Example **query** DATA:REM? 4

Digital Interface Subsystem

The Digital Interface Subsystem provides commands for controlling and configuring digital input and output channels. It enables users to set logic levels, read digital states, and configure communication protocols for external device interaction.

10.1 DIGital:INTerface:MODE

Description Sets or returns the application mode of digital I/O (**Remote Control Only**).

Syntax **command** DIGital:INTerface:MODE <type>

Query DIGital:INTerface:MODE?

Parameters <**type**> := { COPM | 4094 | IO}

Example **command** DIG:INT:MODE IO

query DIG:INT:MODE?

10.2 DIGital:INTerface:DATA:OUTPut

Description When the 4094 mode (serial to parallel) is selected for digital I/O, make use of this command to set output status.

Syntax **command** DIGital:INTerface:DATA:OUTPut <data>,<strobe_pulse>

Parameters <**data**> := { 0 to 250}

<**data**> := { 0 | 1}

Example **command** DIG:INT:DATA:OUPT 10,1

10.3 DIGital:INTerface:DATA:SETup

Description When the IO mode is selected for digital I/O, make use of this command to set output status.

NOTICE Sets DIO1 to low, DIO2 to high, DIO3 to low, DIO4 to high

Syntax **command** DIGital:INTerface:DATA:SETup <boolean>,<boolean>,<boolean>,<boolean>

Parameters <**boolean**> := { 0 | 1}

Example **command** DIG:INT:DATA:SET 0,1,0,1

Display Subsystem

The Digital Display subsystem controls the instrument's display settings, including enabling or disabling the screen, adjusting brightness, and managing displayed information.

11.1 DISPlay

Description Controls the instrument's front panel display state.

NOTICE When OFF, the screen goes black except for the timestamp, and all keys except "Local" are disabled.

Syntax	**command**	DISPlay <boolean>
	query	DISPlay?
Example	**command**	DISP ON
	query	DISP?

11.2 DISPlay:TEXT

Description Displays a text on the instrument's front panel display.

NOTICE Messages are limited to 40 characters, and will be displayed even if the screen is OFF.

Syntax **command** DISPlay:TEXT "<message>"`

query DISPlay:TEXT?

Parameters <message> := { "40 character string"}

Example **command** DISP:TEXT "Example"

11.3 DISPlay:TEXT:CLEar

Description Clears the text message fromt he display.

NOTICE Restores normal display mode if ON; clears the message while keeping the display OFF.

Syntax **command** DISPlay:TEXT:CLEar

Parameters None

Example **command** DISP:TEXT:CLE

Format Subsystem

The Format Subsystem configures the data format for instrument communication, including numerical representation, data encoding, and response formatting. It ensures compatibility with various data processing systems by allowing users to specify output precision, separators, and transmission formats for efficient data exchange.

12.1 FORMat:READing:ALARm

Description Enables (On) or disables (Off) the inclusion of alarm information in the reading format.

Syntax **command** FORMat:READing:ALARm <boolean>

query FORMat:READing:ALARm?

Parameters <**boolean**> := { 0 | 1 | OFF | ON}

Example **command** FORM:READ:ALAR ON

query FORM:READ:ALAR?

12.2 FORMat:READing:CHANnel

Description Enables (On) or disables (Off) the inclusion of channel number information in the reading format.

Syntax **command** FORMat:READing:CHANnel <boolean>

query FORMat:READing:CHANnel?

Parameters <**boolean**> := { 0 | 1 | OFF | ON}

Example **command** FORM:READ:CHAN ON

query FORM:READ:CHAN?

12.3 FORMat:READing:TIME

Description Enables (On) or disables (Off) the inclusion of time stamp information in the reading format.

Syntax

command FORMat:READing:TIME <boolean>

query FORMat:READing:TIME?

Parameters <**boolean**> := { 0 | 1 | OFF | ON}

Example

command FORM:READ:TIME ON

query FORM:READ:TIME?

12.4 FORMat:READing:TIME:TYPE

Description Selects the time format (absolute or relative) for time stamp returned when FORMat:READing:TIME is enabled.

Syntax

command FORMat:READing:TIME:TYPE <character>

query FORMat:READing:TIME:TYPE?

Parameters <**character**> := { ABSolute | RELative}

Example

command FORM:READ:TIME:TYPE ABS
command FORM:READ:TIME:TYPE REL

Relative format - shows the time since the start of the scan.
Ex:+1.12379111E-03 VDC,00000000.659,101,2

- Reading with units(1.124mV)
- Elapsed time(659ms)
- Channel number
- Alarm limit threshold crossed (0 = No alarm, 1 = LO, 2 = HI)

Absolute format - shows the time of the day with the date.
Ex:+1.12379111E-03 VDC,2021,01,28,00,43,39.218,101,0

- Reading with units(1.124mV)
- Date(January 28, 2021)
- Time of day(0:43:39.218 AM)
- Channel number
- Alarm limit threshold crossed (0 = No alarm, 1 = LO, 2 = HI)

12.5 FORMat:READing:UNIT

Description Enables (On) or disables (Off) the inclusion of measurement units (VAC, VDC, OHM, etc.) in the reading format.

Syntax

command	FORMat:READing:UNIT <boolean>
query	FORMat:READing:UNIT?

Parameters <**boolean**> := { 0 | 1 | OFF | ON}

Example

command	FORM:READ:UNIT ON
query	FORM:READ:UNIT?

12.6 HCOPy:SDUMp:DATA?

Description Executes TFT LCD screenshot action. Returns the front panel display image ("screen shot"). Returns a count of data streaming by the image file format of BMP.

Syntax

query	HCOPy:SDUMp:DATA?

Example

query	HCOP:SDUM:DATA?

Measure Subsystem

The Measure Subsystem manages measurement functions on the instrument, allowing users to configure and retrieve measurement data. It includes querys to initiate, control, and query various measurement types.

13.1 MEASure:CAPacitance?

Description Configures the channels for capacitance measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:CAPacitance? <range>,<resolution>,(@<ch_list>)

Parameters <**range**> := { 1nF | 10nF | 100nF | 1µF | 10µF | 100µF; DEF: AUTO}

Example **query** MEAS:CAP? DEF,(@101)

13.2 MEASure:CURRent:{AC | DC}?

Description Configures the channels for AC and DC current measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

NOTICE Autoranging (AUTO or DEFault), will generate an error if you specify a <resolution> because the instrument cannot accurately resolve the integration time (especially if the input continuously changes). If your application requires autoranging, specify DEFault for the <resolution> or omit the <resolution> altogether.

Syntax **query** MEASure:CURRent:{AC |DC }? <range>,<resolution>,(@<ch_list>)

Parameters <**range**> :=
AC: {100µA | 1mA | 10mA | 100mA | 2A; DEF: AUTO }
DC: {1µA | 10µA | 100µA | 1mA | 10mA | 100mA | 2A; DEF: AUTO }

Example **query** MEAS:CURR:AC? 10e-2,(@121,122)

13.3 MEASure:DIODe?

Description Configures the channels for Diode current measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

NOTICE The range and resolution for diode test are fixed at 1 VDC, with a 1 mA current source output.

Syntax **query** MEASure:DIODe? (@<ch_list>)

Parameters None

Example **query** MEAS:DIOD? (@101)

13.4 MEASure:{FREQuency | PERiod}?

Description Configures the channels for frequency and period measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:FREQuency|PERiod? <range>,<resolution>,(@<ch_list>)

Parameters <**range**> :=
AC: { 3Hz to 300kHz; DEF: 20Hz }
DC: { 3.33µs to 333.33ms; DEF: 50ms }

Example **query** MEAS:FREQ? MIN,(@101)

13.5 MEASure:{RESistance | FRESistance}?

Description Configures the channels for 2-Wire and 4-Wire resistance measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

NOTICE Autoranging (AUTO or DEFault), will generate an error if you specify a <resolution> because the instrument cannot accurately resolve the integration time (especially if the input continuously changes). If your application requires autoranging, specify DEFault for the <resolution> or omit the <resolution> altogether.

Syntax **query** MEASure:{RESistance|FRESistance}? <range>,<resolution>,(@<ch_list>)

Parameters <**range**> := {100Ω | 1kΩ | 10kΩ | 100kΩ | 1MΩ | 10MΩ | 100MΩ | 1GΩ; DEF:AUTO }

Example **query** MEAS:RES? 100,(@101)

13.6 MEASure:STRain:{DIRect | FDIRect}?

Description Configures the channels for direct 2-Wire and 4-Wire strain gage measurements and immediately sweeps through the specified channels one time (independent of the

present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:STRain:{DIRect | FDIRect}? <gage_ohms>,<gage_factor>,<range>,<resol

Parameters <**gage_ohms**> := { 80 to 1100Ω; DEF: 120Ω }

<**gage_ohms**> := { 0.5 to 5; DEF: 2 }

<**range**> := {100Ω | 1kΩ | 10kΩ | 100kΩ | 1MΩ | 10MΩ | 100MΩ | 1GΩ; DEF:1kΩ }

Example **query** MEAS:STR:DIR? 100,1,(@101)

13.7 MEASure:STRain:{FULL | HALF}:BENDing?

Description Configures the channels for full and half bending bridge strain gage measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:STRain:FULL|HALF:BENDing? <gage_factor>,<range>,<resolution>,(@<c

Parameters <**gage_factor**> := { 0.5 to 5; DEF: 2 }

<**range**> := { 100mV | 1V | 10V | 100V | 600V; DEF: AUTO }

Example **query** MEAS:STR:FULL:BEND? 1,0.1,(@101)

13.8 MEASure:STRain:{FULL | HALF}:POISson?

Description Configures the channels for full and half poisson bridge strain gage measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:STRain:{FULL|HALF}:POISson? <gage_factor>,<poisson_ratio>,<range>,

Parameters <**gage_factor**> := { 0.5 to 5; DEF: 2 }

<**poisson_ratio**> := { -0.9999 to 0.5; DEF: 0.3 }

<**range**> := { 100mV | 1V | 10V | 100V | 600V; DEF: AUTO }

Example **query** MEAS:STR:FULL:POIS? (@101)

13.9 MEASure:STRain:FULL:BENDing:POISson?

Description Configures the channels for full bending poisson bridge strain gage measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:STRain:FULL:BENDing:POISson? <gage_factor>,<poisson_ratio>,<range

Parameters <**gage_factor**> := { 0.5 to 5; DEF: 2 }

<**poisson_ratio**> := { -0.9999 to 0.5; DEF: 0.3 }

<**range**> := { 100mV | 1V | 10V | 100V | 600V; DEF: AUTO }

Example **query** MEAS:STR:FULL:POIS? (@101)

13.10 MEASure:STRain:QUARter?

Description Configures the channels for quarter bridge strain gage measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:STRain:QUARter? <gage_factor>,<range>,<resolution>,(@<ch_list>)

Syntax **query** MEASure:STRain:FULL:BENDing:POISson? <gage_factor>,<poisson_ratio>,<range

Parameters <**gage_factor**> := { 0.5 to 5; DEF: 2 }

<**poisson_ratio**> := { -0.9999 to 0.5; DEF: 0.3 }

<**range**> := { 100mV | 1V | 10V | 100V | 600V; DEF: AUTO }

Example **query** MEAS:STR:QUAR? 1,(@101)

13.11 MEASure:TEMPerature?

Description Configures the channels for temperature measurements and immediately sweeps through the specified channels one time (independent of the present scan list). The results are sent directly to reading memory and the instrument's output buffer.

Syntax **query** MEASure:TEMPerature? <probe_type>,<type>,<resolution>,(@<ch_list>)

Parameters <**probe type**> := { TCouple | FRTD | RTD | FTHermistor | THERmistor }

<type> :=
TCouple: { B | E | J | K | N | R | S | T | USER) ; DEF: J }
RTD / FRTD: { PT100 | D100 | F100 | PT385 | PT3916 | USER) ; DEF: PT100 }
THERmistor / FTHermistor: { 2.2kΩ| 5kΩ| 10kΩ| USER; DEF: 5kΩ }

<range> := { 100mV | 1V | 10V | 100V | 600V; DEF: AUTO }

Example **query** MEAS:TEMP? TC,K,(@101)

13.12 MEASure[:VOLTage]:{AC | DC}?

Description Configures the channels for AC and DC voltage measurements.

NOTICE

Autoranging (AUTO or DEFault), will generate an error if you specify a <resolution> because the instrument cannot accurately resolve the integration time (especially if the input continuously changes). If your application requires autoranging, specify DEFault for the <resolution> or omit the <resolution> altogether

Syntax **query** MEASure[:VOLTage]:AC|DC? <range>,<resolution>,(@<ch_list>)

Parameters <range> :=
AC: {100mV | 1V | 10V | 100V | 400V; DEF:AUTO }
DC: {100mV | 1V | 10V | 100V | 600V; DEF:AUTO }

Example **query** MEAS:VOLT:AC? 100,(@101)

Mmemory Subsystem

14.1 MMEMory:FORMat:READing:CHEAder

Description Specifies the content of each column header to be either the channel number (NUMber) or the channel's user-defined label (LABel).

NOTICE If the value of the column header is set to LABel using the ROUTe:CHANnel:LABel command, any channel without a user-defined label will display its factory-default channel label instead on its column header.

Syntax **command** MMEMory:FORMat:READing:CHEAder {NUMber | LABel}

query MMEMory:FORMat:READing:CHEAder?

Parameters <**format**> := { NUMber | LABel }

Example **query** MMEM:FORM:READ:CHEA LAB

14.2 MMEMory:FORMat:READing:CSEParator

Description Specifies the character to use for separating the information on each row. Syntax: MMEMory:FORMat:READing:CSEParator

Syntax **command** MMEMory:FORMat:READing:CSEParator <character>

query MMEMory:FORMat:READing:CSEParator?

Parameters <**separator**> := {COMMa | SEMicolon | TAB}

Example **query** MMEM:FORM:READ:CSEP COMM

14.3 MMEMory:FORMat:READing:RLIMit

Description Specifies the row limit (maximum number of rows for sweep data) that will be written to each data logging file by the count set by MMEMory:FORMat:READing:RLIMit:COUNt command.

Syntax **command** MMEMory:FORMat:READing:RLIMit <state>

query MMEMory:FORMat:READing:RLIMit?

Parameters <format> := { 0 | 1 | OFF | ON }

Example **query** MMEM:FORM:READ:RLIM ON

14.4 MMEMory:FORMat:READing:RLIMit:COUNt

Description Sets the row limits count when MMEMory:FORMat:READing:RLIMit ON is set.

Syntax **command** MMEMory:FORMat:READing:RLIMit:COUNt <number>

query MMEMory:FORMat:READing:RLIMit:COUNt? [MIN|MAX|DEF]

Parameters <**count**> := { (65536 | 1048576); DEF: 65536 }

Example **query** MMEM:FORM:READ:RLIM:COUN 10000

14.5 MMEMory:LOG[:ENABle]

Description Enables (On) or disables (Off) logging of the scanned memory readings to a USB drive connected to the front panel USB host port.

Syntax **command** MMEMory:LOG[:ENABle] <state>

query MMEMory:LOG[:ENABle]?

Parameters <**state**> := { 0 | 1 | OFF | ON }

Example **query** MMEM:LOG ON

Output Subsystem

The output subsystem controls an instrument's output, allowing users to enable or disable output, set voltage and current levels, and configure protection features. It supports multiple channels and provides state querying for precise, automated control in test environments.

15.1 OUTPut:ALARm:CLEar:ALL

Description Clears the state of all four alarm output lines.

NOTICE

You can manually clear the output lines at any time (even during a scan) and the alarm data in reading memory is not cleared. However, data is cleared when you initiate a new scan.

Syntax **command** OUTPut:ALARm:CLEar:ALL

Parameters None

Example **command** OUTP:ALAR:CLE

15.2 OUTPut:ALARm{1 | 2 | 3 | 4}:CLEar

Description Clears the state of specified alarm output lines.

NOTICE

You can manually clear the output lines at any time (even during a scan) and the alarm data in reading memory is not cleared. However, data is cleared when you initiate a new scan.

Parameters None

Example **command** OUTP:ALAR3:CLE

15.3 OUTPut:ALARm1 | 2 | 3 | 4:SOURce

Description Assigns one of four alarm numbers to report any alarm conditions on the specified multiplexer or digital channels. On the digital modules, you can configure the instrument to generate an alarm when a specific bit pattern or bit pattern change is detected on a digital input channel or when a specific count is reached on a totalizer channel.

NOTICE

The "#2" means that the next 2 digits indicate how many characters are in the returned memory string.In the above example, the 2 digits are the "18" after the "#2". Therefore, the remaining of the string is 18 digits long. An empty scan list (with no channels selected) will return "#13(@)".

Syntax **command** OUTPut:ALARm1 | 2 | 3 | 4:SOURce (@<ch_list>)

query OUTPut:ALARm1 | 2 | 3 | 4:SOURce? (@<ch_list>)

Parameters <**ch_list**> := { 1 to 420}

Example **command** OUTP:ALAR3:SOUR (@101:104)

15.4 OUTPut:ALARm:MODE

Description Clears the state of specified alarm output lines.

NOTICE

Latch Mode: The alarm output is asserted when a channel's reading crosses a limit, and remains asserted until you clear it manually, start a new scan, or cycle power.

Track Mode: The alarm output is asserted when a channel's reading crosses a limit, and remains asserted only while subsequent readings remain outside the limit. When a reading returns within the limits, the output is automatically cleared.

Syntax **command** OUTPut:ALARm:MODE <mode>

query OUTPut:ALARm:MODE?

Parameters <**mode**> := { LATCh | TRACk }

Example **command** OUTP:ALAR:MODE LATC

15.5 OUTPut:ALARm:SLOPe

Description Configures the level for all four alarm output lines that indicates an alarm, either falling edge (NEG - 0 V), or rising edge (POS - 3.3 V).

Syntax **command** OUTPut:ALARm:SLOPe <alarm_slope>

query OUTPut:ALARm:SLOPe?

Parameters <**alarm_slope**> := { POS | NEG }

Example **command** OUTP:ALAR:SLOP POS

15.6 OUTPut:TRIGger:SLOPe

Description Specifies the rising edge (POS) or falling edge (NEG) as the Channel Closed signal on the rear panel Digital I/O connector. The signal operates differently during internal or external scan.

Syntax **command** OUTPut:TRIGger:SLOPe <trigger_slope>

query OUTPut:TRIGger:SLOPe?

Parameters <**trigger_slope**> := { POS | NEG }

Example **command** OUTP:TRIG:SLOP POS

NOTICE

For internal scans (INSTrument:DMM ON command), it is generated at the END of a sweep, not the beginning of a sweep.
For external scans (INSTrument:DMM OFF command), it is generated when each channel is closed, and can be used to trigger the measurement on the external DMM.

Route Subsystem

The Route Subsystem manages signal routing within an instrument, controlling how signals are directed between inputs, outputs, and internal paths. It enables configuration of switching matrices, signal pathways, and connections for flexible test setups.

16.1 ROUTe:CHANnel:ADVance:SOURce

Description Selects the source of signal that advances to the next channel in the scan list when scanning with an external DMM (internal DMM disabled). When the channel advance signal is received, the instrument opens the currently selected channel and closes the next channel in the scan list. The instrument will accept a software command (BUS), continuous scan trigger (IMMediate), or external TTL-compatible (EXTernal) trigger pulse.

Syntax **command** ROUTe:CHANnel:ADVance:SOURce <source>

query ROUTe:CHANnel:ADVance:SOURce?

Parameters <**source**> := {BUS | IMMediate | EXTernal}

Example **query** ROUT:CHAN:ADV:SOUR IMM

16.2 ROUTe:CHANnel:DELay

Description Adds a delay between channels in the scan list (useful for high-impedance or high-capacitance circuits). The delay is inserted between the relay closure and the actual measurement on each channel, in addition to any delay that will implicitly occur due to relay settling time. The programmed channel delay overrides the default channel delay that the instrument automatically adds to each channel.

Syntax **command** ROUTe:CHANnel:DELay <seconds>,(@<ch_list>)

query ROUTe:CHANnel:DELay? (@<ch_list>)

Parameters <**source**> := { 0 to 60 }

Example **query** ROUT:CHAN:DEL 2

16.3 ROUTe:CHANnel:DELay:AUTO

Description Enables (On) or disables (Off) an automatic channel delay on the specified channels. If enabled, the instrument determines the delay based on function, range, integration time, and AC filter setting.

Syntax **command** ROUTe:CHANnel:DELay:AUTO <boolean>,(@<ch_list>)

query ROUTe:CHANnel:DELay:AUTO? (@<ch_list>)

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<ch_list> := { 1 to 420}

Example **query** ROUT:CHAN:DEL:AUTO ON

16.4 ROUTe:CHANnel:FWIRe

Description Configures the specified channels for 4-wire external scanning. When enabled, channel n is paired with channel n+10 (DAQ-900 or DAQ-901) or n+4 (DAQ-909) to provide source and sense connections.

NOTICE

When specifying the scan list using ROUTe:SCAN, only specify the lower channel number (n) for paired channels; the upper channel number (n+10 or n+4) is not allowed in the scan list.

Syntax **command** ROUTe:CHANnel:FWIRe <boolean>,(@<ch_list>)

query ROUTe:CHANnel:FWIRe? (@<ch_list>)

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **query** ROUT:CHAN:FWIRe ON,(@101,102)

16.5 ROUTe:CHANnel:LABel

Description Assigns a user-defined label to the specified channels.

NOTICE

When shipped from the factory, each channel is assigned a unique factory-default label (cannot be overwritten).
Specifying a null string ("") disables the user-defined message.

Syntax **command** ROUTe:CHANnel:LABel "<label>",(@<ch_list>)

query ROUTe:CHANnel:LABel? [<tag>,](@<ch_list>)

Parameters <label> := { max length = 30 characters }

<**tag**> := { USER | FACtory }

USER := {Read the user-defined label on the specified channel.}

FACTory := {Read the factory-default label on the specified channel.}

<**ch_list**> := { 1 to 420}

Example **command** ROUT:CHAN:LAB "test",(@101,103)

query ROUT:CHAN:LAB? USER,(@101,103)

16.6 ROUTe:CHANnel:LABel:CLEar:MODule

Description Clears all user-defined labels on all channels in the specified slot, or on all modules installed in the DAQ3120, and restores the factory-default labels.

NOTICE

This command does not clear the factory-default channel labels. The factory-default labels are always preserved.
The instrument keeps a record of what module types are installed in each slot. If a different module type is detected in a specific slot at power on, all user-defined channel labels for that slot are discarded. If an empty slot is detected at power-on, any previously-defined labels for that slot are preserved and will be restored if the same module type is installed later; however, if a module of a different type is installed in that slot, the previously-defined labels will be discarded.

Syntax **command** ROUTe:CHANnel:LABel:CLEar:MODule <slot>

Parameters <**lot**> := { 1 to 3 | ALL }

Example **command** ROUT:CHAN:LAB:CLE:MOD 1

16.7 ROUTe:CLOSe

Description Closes the specified channels on a multiplexer or switch module. On the multiplexer modules, if any channel on the module is defined to be part of the scan list, attempting to send this command will result in an error.

NOTICE

For the matrix module (DM-304), the channel number represents the intersection of the desired row and column. For example, channel 312 represents the intersection of row 1 and column 2 on the module in slot 3 (assumes two-wire mode).

Syntax **command** ROUTe:CLOSe (@<ch_list>)

query ROUTe:CLOSe? (@<ch_list>)

Parameters <**ch_list**> := { 1 to 420}

Example **command** ROUT:CLOS (@101,102)

query ROUT:CLOS? (@101,102)

16.8 ROUTe:CLOSe:EXCLusive

Description Opens all channels on a multiplexer or switch module and then closes the specified channels. On the multiplexer modules, if any channel on the module is defined to be part of the scan list, attempting to send this command will result in an error.

NOTICE This command opens all channels first, and then closes the channels in the <ch_list>, one at a time. Before it closes each channel, it opens all previous channels.

Syntax **command** ROUTe:CLOSe:EXCLusive (@<ch_list>)

Parameters <**ch_list**> := { 1 to 420}

Example **command** ROUT:CLOS:EXCL (@102)

16.9 ROUTe:DONE?

Description Returns the status of all relay operations on modules that not involved in the scan and returns a 1 when finished (even during a scan).

Syntax **query** ROUTe:CLOSe:EXCLusive (@<ch_list>)

Return Parameters <**boolean**> := { : 0 | 1, (0 = Unfinished, 1 = finished) }

Example **command** ROUT:CLOS:EXCL (@102)

16.10 ROUTe:MONitor

Description Selects the channel to be displayed on the front panel. Only one channel can be monitored at a time.

Syntax **command** ROUTe:MONitor (@<channel>)

query ROUTe:MONitor?

Parameters <**channel**> := { A single channel (1 to 402) }

Example **command** ROUT:MON (@101)

query ROUT: MON?

Returns: #16(@101)

NOTICE

The "#1" means that the next 1 digits indicate how many characters are in the returned memory string. In the above example, the 1 digits are the "6" after the "#1". Therefore, the remaining of the string is 6 digits long.

16.11 ROUTe:MONitor:DATA?

Description Reads the monitor data from the selected channel. It returns the reading only; the units, time, channel, and alarm information are not returned (the FORMat:READing commands do not apply to monitor readings).

Syntax **query** ROUTe:MONitor:DATA?

Parameters None

Example **query** ROUT:MON:DATA?

NOTICE

If the Monitor mode is not currently enabled, this query returns 9.91E37 (not a number).
Readings acquired during a Monitor are not stored in reading memory but they are displayed on the front panel; however, all readings from a scan in progress at the same time are stored in reading memory.

16.12 ROUTe:MONitor:DATA:FULL?

Description Reads the monitor data from the selected channel. It returns all the reading with the units, time, channel, and alarm information (all the FORMat:READing enabled commands apply to this monitor readings).

Syntax **query** ROUTe:MONitor:DATA:FULL?

Parameters None

Example **query** ROUT:MON:DATA:FULL?

NOTICE

If the Monitor mode is not currently enabled, this query returns 9.91E37 (not a number).
Readings acquired during a Monitor are not stored in reading memory but they are displayed on the front panel; however, all readings from a scan in progress at the same time are stored in reading memory.

16.13 ROUTe:MONitor:STATe

Description Enables (On) or disables (Off) the Monitor mode. The Monitor mode is equivalent to making continuous measurements on a single channel with an infinite scan count. Only one channel can be monitored at a time but you can change the channel being monitored at any time.

Syntax **command** ROUTe:MONitor:STATe <boolean>

query ROUTe:MONitor:STATe?

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

Example **command** ROUT:MON:STAT ON

16.14 ROUTe:MONitor:VIEW

Description Selects how measurement data is displayed (numeric, trend chart, histogram, and bar meter format) in monitoring mode.

Syntax **command** ROUTe:MONitor:VIEW <display>

query ROUTe:MONitor:VIEW?

Parameters <**display**> := { NUMeric | TCHart | HISTogram | METer }

Example **command** ROUT:MON:VIEW NUM

16.15 ROUTe:OPEN

Description Opens the specified channels on a multiplexer or switch module. On the multiplexer modules, if any channel on the module is defined to be part of the scan list, attempting to send this command will result in an error.

Syntax **command** ROUTe:OPEN (@<ch_list>)

query ROUTe:OPEN? (@<ch_list>)

Parameters <**ch_list**> := { 1 to 420}

Example **command** ROUT:OPEN (@101,102)

qurey ROUT:OPEN? (@101,102))

NOTICE

For the matrix module (DM-304), the channel number represents the intersection of the desired row and column. For example, channel 312 represents the intersection of row 1 and column 2 on the module in slot 3 (assumes two-wire mode).

16.16 ROUTe:SCAN

Description Selects the channels to be included in the scan list. This command is used in conjunction with the CONFigure commands to set up an automated scan. The specified channels supersede any channels previously defined to be part of the scan list. To start the scan, use the INITiate or READ? command.

Syntax **command** ROUTe:SCAN (@<ch_list>)

query ROUTe:SCAN?

Parameters <**ch_list**> := { 1 to 420}

Example **command** ROUT:SCAN (@101,102)

qurey ROUT:SCAN?

Returns: #210(@101,102)

NOTICE

The "#2" means that the next 2 digits indicate how many characters are in the returned memory string. In the above example, the 2 digits are the "10" after the "#2". Therefore, the remaining of the string is 10 digits long.
To remove all channels from the present scan list, issue the command ROUT:SCAN (@).
An empty scan list (with no channels selected) will return "#13(@)".

16.17 ROUTe:SCAN:SIZE?

Description Returns the number of channels in the scan list as defined by the ROUTe:SCAN command.

Syntax **query** ROUTe:SCAN:SIZE?

Example **qurey** ROUT:SCAN:SIZE?

NOTICE The present scan list is stored in non-volatile memory and will be retained when power is turned off.

Sense Subsystem

The SCPI Sense Subsystem is responsible for configuring measurement parameters in test and measurement instruments. It controls settings such as range, resolution, and integration time to ensure accurate signal acquisition. Commands within this subsystem allow users to define limits, apply filters, and retrieve measured values, making it essential for precision testing applications.

17.1 [SENSe:]FUNCtion[:ON]

Description Selects the measurement function on the selected channels (all function-related measurement attributes areretained).

Syntax **command** [SENSe:]FUNCtion[:ON] "<function>"[,(@<ch_list>)]

query [SENSe:]FUNCtion[:ON]? [(@<ch_list>)]

Parameters <function> := { "CAP" | "CURR:AC" | "CURR[:DC]" | "DIOD" | "FREQ" | "PER" | "FRES" | "RES" | "STR:DIR" | "STR:FDIR" | "STR:QUAR" | "STR:HALF:BEND" | "STR:HALF:POIS" | "STR:FULL:BEND" | "STR:FULL:BEND:POIS" | "STR:FULL:POIS" | "TEMP[:TC]" | "TEMP:FRTD" | "TEMP:RTD" | "TEMP:FTH" | "TEMP:THER" | "VOLT:AC" | "VOLT[:DC]" }

<**ch_list**> := { 1 to 420}

Example **command** FUNC "RES"

17.2 [SENSe:]AVERage:COUNt

Description Sets or returns the digital filter count.

Syntax **command** [SENSe:]AVERage:COUNt {<count>|MIN|MAX}[,(@<ch_list>)]

query [SENSe:]AVERage:COUNt? [{(@<ch_list>)|MIN|MAX}]

Parameters <**count**> := { 2 to 10 }

<**ch_list**> := { 1 to 420}

Example **command** AVER:COUN MIN

query AVER:COUN?

17.3 [SENSe:]AVERage:STATe

Description Enable(On) or disable(Off) the digital filter function state.

NOTICE If NPLC >= 7.2k/s, the filter function will be disabled.

Syntax **command** [SENSe:]AVERage:STATe {OFF | ON}[,(@<ch_list>)]

query [SENSe:]AVERage:STATe? [(@<ch_list>)]

Parameters 0 | 1 | OFF | ON <**boolean**> := { 0 | 1, (0 = OFF, 1 = ON) }

<**ch_list**> := { 1 to 420}

Example **command** AVER:STAT ON

17.4 [SENSe:]AVERage:WINDow

Description Sets or returns a digital filter window value.

Syntax **command** [SENSe:]AVERage:WINDow {<percent> | MIN | MAX }[,(@<ch_list>)]

query [SENSe:]AVERage:WINDow? [{(@<ch_list>) | MIN | MAX}]

Parameters <**percent**> := { 0.01 | 0.1 | 1 | 10 | NONE }

<**ch_list**> := { 1 to 420}

Example **command** AVER:WIND 0.1

17.5 [SENSe:]AVERage:WINDow:METHod

Description Sets or returns a digital filter window method type.

Syntax **command** [SENSe:]AVERage:WINDow:METHod <type>[,(@<ch_list>)]

query [SENSe:]AVERage:WINDow:METHod? [(@<ch_list>)]

Parameters <**type**> := { MEASure | RANGe }

<**ch_list**> := { 1 to 420}

Example **command** AVER:WIND:METH MEAS

17.6 [SENSe:]CAPacitance:RANGe

Description Selects a fixed range for capacitance measurements.

Syntax **command** [SENSe:]CAPacitance:RANGe {<range> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]CAPacitance:RANGe? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**range**> := { 1nF | 10nF | 100nF | 1µF | 10µF | 100µF; DEF:AUTO }

<ch_list> := { 1 to 420}

Example **command** CAP:RANG 1e-6

query CAP:RANG?

17.7 [SENSe:]CAPacitance:RANGe:AUTO

Description Enables or disables autoranging for capacitance measurements.

Syntax **command** [SENSe:]CAPacitance:RANGe:AUTO {OFF | ON }[,(@<ch_list>)]

query [SENSe:]CAPacitance:RANGe:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** CONF:CAP (@101) **query** CAP:RANG:AUTO ON

NOTICE

Autorange thresholds:
Down range at: < 10% of range
Up range at: > 120% of range

17.8 [SENSe:]CURRent:AC:BANDwidth

Description Sets or returns the ac filter bandwith for AC current measurements.

Syntax **command** [SENSe:]CURRent:AC:BANDwidth {<freq> | MIN | MAX | DEF}[,(@<ch_list>)] }

query [SENSe:]CURRent:AC:BANDwidth? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**frequency**> := { 3 | 20 | 200Hz; DEF: 20Hz }

<**ch_list**> := { 1 to 420}

Example **command** CURR:AC:BAND 3

query CURR:AC:BAND?

17.9 [SENSe:]CURRent:{AC | DC}:RANGe

Description Selects a fixed range for AC and DC current measurements.

Syntax **command** [SENSe:]CURRent:{AC | DC}:RANGe {<range> | MIN | MAX | DEF }[,(@<ch_list>)]

query [SENSe:]CURRent:{ AC | DC }:RANGe? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**range**> :=
AC: { 100µA | 1mA | 10mA | 100mA | 2A; DEF:AUTO }
DC: { 1µA | 10µA | 100µA | 1mA | 10mA | 100mA | 2A; DEF:AUTO }

<**ch_list**> := { 1 to 420}

Example **command** CURR:AC:RANG 0.1

query CURR:AC:RANG?

17.10 [SENSe:]CURRent:{AC | DC}:RANGe:AUTO

Description Enables or disables autoranging for AC and DC current measurements.

Syntax **command** [SENSe:]CURRent:{ AC | DC }:RANGe:AUTO {OFF | ON }[,(@<ch_list>)]

query [SENSe:]CURRent:{ AC | DC }:RANGe:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** CONF:CURR:AC (@101)

query CURR:AC:RANG:AUTO ON

NOTICE

Autorange thresholds:
Down range at: < 10% of range
Up range at: > 120% of range

17.11 [SENSe:]CURRent:AC | DC:RANGe:LOW

Description Selects a limit minimum current at autoranging for AC and DC current measurements.

Syntax **command** [SENSe:]CURRent:{AC | DC}:RANGe:LOW {<range> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]CURRent:{AC | DC}:RANGe:LOW? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <range>: AC: (100µA | 1mA | 10mA | 100mA), DEF: 100uA DC: (1µA | 10µA | 100µA | 1mA | 10mA | 100mA) , DEF: 1uA Return parameter: <NRf>

Example **command** CONF:CURR:AC (@121) CURR:AC:RANG:LOW 0.01 CURR:AC:RANG:LOW?

17.12 [SENSe:]CURRent[:DC]:APERture

Description Enables the aperture mode and sets the integration time in seconds (called aperture time) for DC current measurements.

Syntax **command** [SENSe:]CURRent[:DC]:APERture {<seconds> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]CURRent[:DC]:APERture? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**seconds**> := { 20µs to 1s; DEF: 100ms

<**ch_list**> := { 1 to 420}

Example **command** CURR:APER 0.1

query CURR:APER?

17.13 [SENSe:]CURRent[:DC]:APERture:ENABle

Description Enables the setting of integration time in seconds (called aperture time) for DC current measurements. If aperture time mode is disabled , the integration time is set in PLC (power-line cycles).

Syntax **command** [SENSe:]CURRent[:DC]:APERture:ENABle {OFF | ON}[,(@<ch_list>)]

query [SENSe:]CURRent[:DC]:APERture:ENABle? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** CURR:APER:ENAB ON

query CURR:APER:ENAB?

17.14 [SENSe:]CURRent[:DC]:NPLCycles

Description Sets or returns the integration time in number of power line cycles (PLCs) for DC current measurements. Where one PLC is equal to 16.6 milliseconds.

Syntax **command** [SENSe:]CURRent[:DC]:NPLCycles {<PLCs> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]CURRent[:DC]:NPLCycles? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**PLCs**> := { 0.0016 | 0.0032 | 0.0042 | 0.0083 | 0.0125 | 0.025 | 0.05 | 0.15 | 0.6 | 1 | 3 | 12; DEF: 1 PLC }

<**ch_list**> := { 1 to 420}

Example **command** CURR:NPLC 1

query CURR:NPLC?

17.15 [SENSe:]CURRent[:DC]:ZERO:AUTO

Description Enables or disables the autozero mode for DC current measurements.

Syntax **command** [SENSe:]CURRent[:DC]:ZERO:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]CURRent[:DC]:ZERO:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** CURR:ZERO:AUTO ON

query CURR:ZERO:AUTO?

17.16 [SENSe:]DIODe:ZERO:AUTO

Description Enables or disables the autozero mode for diode measurements.

Syntax **command** [SENSe:]DIODe:ZERO:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]DIODe:ZERO:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** DIOD:ZERO:AUTO ON

command DIOD:ZERO:AUTO?

17.17 [SENSe:]{FREQuency | PERiod}:APERture

Description Sets or returns the aperture time (gate time) for the frequency and period measurements.

Syntax **command** [SENSe:]{FREQuency | PERiod}:APERture {<seconds> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]{FREQuency | PERiod}:APERture? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**seconds**> := { 0.001 | 0.01 | 0.1 | 1s); DEF: 0.1s }

<**ch_list**> := { 1 to 420}

Example **command** FREQ:APER 0.1

query FREQ:APER?

17.18 [SENSe:]{FREQuency | PERiod}:RANGe:LOWer

Description

Description Sets or returns the ac filter bandwith of frequency and period measurements.

Syntax **command** [SENSe:]{FREQuency | PERiod}:RANGe:LOWer {<freqeuncy> | MIN | MAX | DEF}[,(@<ch_list>)

query [SENSe:]{FREQuency | PERiod}:RANGe:LOWer? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**frequency**> := { 3 | 20 | 200Hz; DEF: 20Hz }

<**ch_list**> := { 1 to 420}

Example **command** FREQ:RANG:LOW 3

query FREQ:RANG:LOW?

17.19 [SENSe:]{FREQuency | PERiod}:TIMeout:AUTO

Description Sets or returns the timeout time for frequency and period measurements.

Syntax **command** [SENSe:]{FREQuency | PERiod}:TIMeout:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]{FREQuency | PERiod}:TIMeout:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

NOTICE

The return parameter indicates; 0 | 1, (0:timeout time = 1 second, 1:timeout time is different in according with ac filter bandwith)

Example **command** PER:TIM:AUTO ON

query PER:TIM:AUTO?

17.20 [SENSe:]{FREQuency | PERiod}:VOLTage:RANGe

Description Selects a fixed voltage range for frequency and period measurements.

Syntax **command** [SENSe:]{FREQuency | PERiod}:VOLTage:RANGe {<range> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]{FREQuency | PERiod}:VOLTage:RANGe? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <range> (100mV | 1V | 10V | 100V | 400V); DEF: 10V Return parameter: <NRf>

Example **command** FREQ:VOLT:RANG 0.1

query FREQ:VOLT:RANG?

17.21 [SENSe:]{FREQuency | PERiod}:VOLTage:RANGe:AUTO

Description Enables or disables voltage autoranging for frequency and period measurements.

NOTICE

Autorange thresholds:
Down range at: < 10% of range
Up range at: > 120% of range

Syntax **command** [SENSe:]{FREQuency | PERiod}:VOLTage:RANGe:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]{FREQuency | PERiod}:VOLTage:RANGe:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<ch_list> := { 1 to 420}

Example **command** FREQ:VOLT:RANG:AUTO ON

query FREQ:VOLT:RANG:AUTO?

17.22 [SENSe:]{RESistance | FRESistance}:APERture

Description Enables the aperture mode and sets the integration time in seconds (called aperture time) for 2-wire and 4-wire resistance measurements.

Syntax **command** [SENSe:]{RESistance | FRESistance}:APERture {<seconds> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:APERture? [(@<ch_list>) | MIN | MAX | DEF]

Parameters <seconds> := { 20µs to 1s; DEF: 100ms }

<ch_list> := { 1 to 420}

Example **command** RES:APER 0.1

query RES:APER?

17.23 [SENSe:]{RESistance | FRESistance}:APERture:ENABle

Description Enables the setting of integration time in seconds (called aperture time) for 2-wire and 4-wire resistance measurements. If aperture time mode is disabled , the integration time is set in PLC (power-line cycles).

Syntax **command** [SENSe:]{RESistance | FRESistance}:APERture:ENABle {OFF | ON}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:APERture:ENABle? [(@<ch_list>)]

Parameters <boolean> := { 0 | 1 | OFF | ON }

<ch_list> := { 1 to 420}

Example **command** RES:APER:ENAB ON

query RES:APER:ENAB?

17.24 [SENSe:]{RESistance | FRESistance}:NPLCycles

Description Sets or returns the integration time in number of power line cycles (PLCs) for 2-wire and 4-wire resistance measurements. Where one PLC is equal to 16.6 milliseconds.

Syntax **command** [SENSe:]{RESistance | FRESistance}:NPLCycles {<PLCs> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:NPLCycles? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**PLCs**> := { 0.0016 | 0.0032 | 0.0042 | 0.0083 | 0.0125 | 0.025 | 0.05 | 0.15 | 0.6 | 1 | 3 | 12; DEF: 1 PLC }

<**ch_list**> := { 1 to 420}

Example **command** RES:NPLC 1

query RES:NPLC?

17.25 [SENSe:]{RESistance | FRESistance}:OCOMpensated

Description Enables or disables offset compensation for 2-wire and 4-wire resistance measurements.

NOTICE Applies only to resistance measurements on the 100 Ω through 100 kΩ ranges.

Syntax **command** [SENSe:]{RESistance | FRESistance}:OCOMpensated {OFF | ON}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:OCOMpensated? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** RES:OCOM ON

query RES:OCOM?

17.26 [SENSe:]{RESistance | FRESistance}:POWer:LIMit[:STATe]

Description Enables or disables low-power for 2-wire and 4-wire resistance measurements.

NOTICE

Low-power resistance measurements apply to the 100 Ω through 100 kΩ ranges only. The 1 MΩ through 1 GΩ ranges source the same current regardless of the low-power setting.

Syntax **command** [SENSe:]{RESistance | FRESistance}:POWer:LIMit[:STATe] {OFF | ON}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:POWer:LIMit[:STATe]? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** RES:POW:LIM ON

query RES:POW:LIM?

17.27 [SENSe:]{RESistance | FRESistance}:RANGe

Description Selects a fixed range for 2-wire and 4-wire resistance measurements.

Syntax **command** [SENSe:]{RESistance | FRESistance}:RANGe {<range> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:RANGe? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**range**> := { 100Ω | 1kΩ | 10kΩ | 100kΩ | 1MΩ | 10MΩ | 100MΩ | 1GΩ; DEF:1kΩ }

<**ch_list**> := { 1 to 420}

Example **command** FRES:RANG 10e3

query FRES:RANG?

17.28 [SENSe:]{RESistance | FRESistance}:RANGe:AUTO

Description Enables or disables autoranging for 2-wire and 4-wire resistance measurements.

NOTICE

Autorange thresholds:
Down range at: < 10% of range
Up range at: > 120% of range

Syntax **command** [SENSe:]{RESistance | FRESistance}:RANGe:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:RANGe:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** FRES:RANG:AUTO ON

query FRES:RANG:AUTO?

17.29 [SENSe:]{RESistance | FRESistance}:ZERO:AUTO

Description Enables or disables the autozero mode for 2-wire and 4-wire resistance measurements.

Syntax **command** [SENSe:]{RESistance | FRESistance}:ZERO:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]{RESistance | FRESistance}:ZERO:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** FRES:ZERO:AUTO ON

query FRES:ZERO:AUTO?

17.30 [SENSe:]STRain:APERture

Description Enables the aperture mode and sets the integration time in seconds (called aperture time) for strain measurements.

Syntax **command** [SENSe:]STRain:APERture {<seconds> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRain:APERture? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**seconds**> := { 20µs 1s; DEF: 100ms }

<**ch_list**> := { 1 to 420}

Example **command** STR:APER 0.1

query STR:APER?

17.31 [SENSe:]STRain:APERture:ENABle

Description Enables the setting of integration time in seconds (called aperture time) for strain measurements. If aperture time mode is disabled , the integration time is set in PLC (power-line cycles).

Syntax

command [SENSe:]STRain:APERture:ENABle {OFF | ON}[,(@<ch_list>)]

query [SENSe:]STRain:APERture:ENABle? [(@<ch_list>)]

Parameters

<**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example

command STR:APER:ENAB ON

query STR:APER:ENAB?

17.32 [SENSe:]STRain:EXCitation

Description Specifies the excitation voltage applied to the bridge by an external DC voltage source. This value will be used to convert strain bridge measurements on the specified channel.

NOTICE The external DC voltage reference channel must be the next lowest channel than the subsequent strain channel.

Syntax

command [SENSe:]STRain:EXCitation {<voltage> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRain:EXCitation? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters

<**voltage**> := { 1 to 12V; DEF: 5V }

<**ch_list**> := { 1 to 420}

Example

command STR:EXC 3

querry STR:EXC?

17.33 [SENSe:]STRain:EXCitation:TYPE

Description Strain bridge conversions require the value of the external bridge excitation voltage. For this voltage, you can dedicate a multiplexer channel to measure the excitation voltage, or can specify a known fixed voltage value.

Syntax **command** [SENSe:]STRain:EXCitation:TYPE {EXTernal | FIXed}[,(@<ch_list>)]

query [SENSe:]STRain:EXCitation:TYPE? [(@<ch_list>)]

Parameters <> := { EXTernal | FIXed }
FIXed := the excitation voltage specified by SENSe:STRain:EXCitation will be used for the strain conversion.
EXTernal := the next-lowest channel configured for DCV measurements with reference mode enabled (see SENSe:VOLTage:DC:REFerence command) will be used as the excitation voltage reference in the strain conversion.

<**ch_list**> := { 1 to 420}

Example **command** STR:EXC:TYPE FIX

query STR:EXC 3

17.34 [SENSe:]STRain:GFACtor

Description Specifies the gage factor to be used to convert direct strain and strain bridge readings on the specified channel. Gage factor is defined as the ratio of the fractional change in resistance to the fractional change in length (strain) along the axis of the edge.

NOTICE Gage factor is a dimensionless quantity. The larger the value, the more sensitive strain gage.

Syntax **command** [SENSe:]STRain:GFACtor {<gage_factor> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRain:GFACtor? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**gage_factor**> := { 0.5 to 5; DEF: 2 }

<**ch_list**> := { 1 to 420}

Example **command** STR:GFAC 1

query STR:GFAC?

17.35 [SENSe:]STRain:NPLCycles

Description Sets or returns the integration time in number of power line cycles (PLCs) strain measurements. Where one PLC is equal to 16.6 milliseconds.

Syntax **command** [SENSe:]STRain:NPLCycles {<PLCs> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRain:NPLCycles? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**PLCs**> := { 0.0016 | 0.0032 | 0.0042 | 0.0083 | 0.0125 | 0.025 | 0.05 | 0.15 | 0.6 | 1 | 3 | 12; DEF: 1 PLC }

<**ch_list**> := { 1 to 420}

Example **command** STR:NPLC 1

query STR:NPLC?

17.36 [SENSe:]STRain:OCOMpensated

Description Enables or disables offset compensation for strain measurements.

NOTICE Applies only to resistance measurements on the 100 Ω through 100 kΩ ranges.

Syntax **command** [SENSe:]STRain:OCOMpensated {OFF | ON}[,(@<ch_list>)]

query [SENSe:]STRain:OCOMpensated? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** STR:OCOM ON

query STR:OCOM?

17.37 [SENSe:]STRain:POISson

Description This command sets the poisson ratio to be used to convert strain bridge readings on the specified channels. Poisson ratio is defined as the negative ratio of the strain the transverse direction to the strain the longitudinal direction.

Syntax **command** [SENSe:]STRain:POISson {<poisson_ratio> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRainPOISson? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**poisson_ratio**> := { -0.9999 to 0.5; DEF: 0.3 }

<**ch_list**> := { 1 to 420}

Example **command** STR:POIS 1

query STR:POIS?

17.38 [SENSe:]STRain:RESistance

Description This command specifies the gage ohm value to be used to convert direct strain measurements on the specified channel.

Syntax **command** [SENSe:]STRain:RESistance {<gage_ohm> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRain:RESistance? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**gage_ohm**> := { 80 1100Ω; DEF: 120Ω }

<**ch_list**> := { 1 to 420}

Example **command** STR:RES 100

query STR:RES?

17.39 [SENSe:]STRain:UNSTrained

Description This command specifies the unstrained bridge offset (can be either voltage or resistance) that will be subtracted from the strain bridge measurements before the strain conversion is performed strain bridge measurements.

Syntax **command** [SENSe:]STRain:UNSTrained {<offset> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRain:UNSTrained? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**offset**> := { -90 to 90; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** STR:UNST 10

query STR:UNST?

17.40 [SENSe:]STRain:UNSTrained:IMMediate

Description This command immediately measures and stores the bridge offset voltages on the specified channel.

Syntax **command** [SENSe:]STRain:UNSTrained:IMMediate [(@<ch_list>)]

Parameters <**ch_list**> := { 1 to 420}

Example **command** STR:UNST:IMM

query STR:UNST:IMM?

17.41 [SENSe:]STRain:VOLTage:RANGe

Description Selects a fixed range for strain measurements.

Syntax **command** [SENSe:]STRain:VOLTage:RANGe {<range> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]STRain:VOLTage:RANGe? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**range**> := { 100mV | 1V | 10V | 100V | 600V; DEF: 100mV }

<**ch_list**> := { 1 to 420}

Example **command** STR:VOLT:RANG 10

17.42 [SENSe:]STRain:VOLTage:RANGe:AUTO

Description Enables or disables autoranging for strain measurements.

NOTICE

Autorange thresholds:
Down range at: < 10% of range
Up range at: > 120% of range

Syntax **command** [SENSe:]STRain:VOLTage:RANGe:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]STRain:VOLTage:RANGe:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** STR:VOLT:RANG:AUTO ON

query STR:VOLT:RANG:AUTO?

17.43 [SENSe:]STRain:ZERO:AUTO

Description Enables or disables the autozero mode for strain measurements.

Syntax **command** [SENSe:]STRain:ZERO:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]STRain:ZERO:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** STR:ZERO:AUTO ON

command STR:ZERO:AUTO?

17.44 [SENSe:]TEMPerature:APERture

Description Enables the aperture mode and sets the integration time in seconds (called aperture time) for temperature measurements.

Syntax **command** [SENSe:]TEMPerature:APERture {<seconds> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:APERture? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**seconds**> := { 20µs 1s; DEF: 100ms }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:APER 0.5

querry TEMP:APER?

17.45 [SENSe:]TEMPerature:APERture:ENABle

Description Enables the setting of integration time in seconds (called aperture time) for temperature measurements. If aperture time mode is disabled , the integration time is set in PLC (power-line cycles).

Syntax **command** [SENSe:]TEMPerature:APERture:ENABle {OFF | ON}[,(@<ch_list>)]

query [SENSe:]TEMPerature:APERture:ENABle? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:APER:ENAB ON

query TEMP:APER:ENAB?

17.46 [SENSe:]TEMPerature:NPLCycles

Description Sets or returns the integration time in number of power line cycles (PLCs) temperature measurements. Where one PLC is equal to 16.6 milliseconds.

Syntax **command** [SENSe:]TEMPerature:NPLCycles {<PLCs> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:NPLCycles? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**PLCs**> := { 0.0016 | 0.0032 | 0.0042 | 0.0083 | 0.0125 | 0.025 | 0.05 | 0.15 | 0.6 | 1 | 3 | 12; DEF: 1 PLC }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:NPLC 3

query TEMP:NPLC?

17.47 [SENSe:]TEMPerature:RJUNction?

Description Returns the internal reference junction temperature on the specified channels in degrees Celsius, regardless of the temperature units currently selected. This is useful only for an internal reference source.

Syntax **query** [SENSe:]TEMPerature:RJUNction? [(@<ch_list>)]

Parameters Nonen

Example **query** TEMP:RJUN?

17.48 [SENSe:]TEMPerature:RJUNction:SIMulated:AUTO:OFFSet

Description Sets or returns junction reference temperature adjust value of thermocouple measurement which internal temperature is selected.

Syntax **command** [SENSe:]TEMPerature:RJUNction:SIMulated:AUTO:OFFSet {<temperature> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:RJUNction:SIMulated:AUTO:OFFSet?[{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**temperature**> := { -20.00 to 20.00; DEF:0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:RJUN:SIM:AUTO:OFFS 10

query TEMP:RJUN:SIM:AUTO:OFFS?

17.49 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:TYPE

Description Selects the 2-wire and 4-wire RTD sensor type.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:TYPE <sensor_type>[,(@<ch_list

query [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:TYPE? [(@<ch_list>)]

Parameters <**sensor_type**> := { PT100 | D100 | F100 | PT385 | PT3916 | USER }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:TYPE PT100

query TEMP:TRAN:RTD:TYPE PT?

17.50 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:ALPHa

Description Sets or returns the 2-wire and 4-wire RTD alpha coefficient.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:ALPHa {<coefficient> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:ALPHa? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**coefficient**> := { 0.0 to 9.999999; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:USER:ALPH 0.00385

query TEMP:TRAN:RTD:USER:ALPH?

17.51 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:BETA

Description Sets or returns the 2-wire and 4-wire RTD beta coefficient.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:BETA {<coefficient> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:BETA? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**Coeffiecient**> := { 0.0 to 9.999999; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:USER:BETA 0.10863

query TEMP:TRAN:RTD:USER:BETA 0.10863

17.52 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:DELTa

Description Sets or returns the 2-wire and 4-wire RTD delta coefficient.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:DELTa {<coefficient> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:USER:DELTa? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**coeffiecient**> := { 0.0 to 9.999999; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:USER:DELT 1.4999

querry TEMP:TRAN:RTD:USER:DELT?

17.53 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:OCOMpensated

Description Enables or disables offset compensation for temperature measurements.

NOTICE

This command applies only to 2-wire and 4-wire RTD measurements on the 100 Ω, 1 kΩ, and 10 kΩ ranges. Once enabled, offset compensation is applied to both 2-wire and 4-wire RTD measurements on the specified channels. Applies only to resistance measurements on the 100 Ω through 100 kΩ ranges.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:OCOMpensated {OFF|ON}[,(@<ch_list>)] **query** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:OCOMpensated? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<ch_list> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:OCOM ON

query TEMP:TRAN:RTD:OCOM ON

17.54 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:POWer:LIMit[:STATe]

Description Enables or disables low-power for 2-wire and 4-wire RTD measurements.

NOTICE Low-power resistance measurements apply to the 100 Ω through 100 kΩ ranges only. The 1 MΩ through 1 GΩ ranges source the same current regardless of the low-power setting

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:POWer:LIMit[:STATe] {OFF | ON}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:POWer:LIMit[:STATe]? [(@<ch_list>)]

Parameters <boolean> := { 0 | 1 | OFF | ON }

<ch_list> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:POW:LIM ON

query TEMP:TRAN:RTD:POW:LIM ON

17.55 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:REFerence

Description Enables (On) or disables (Off) the specified 2-wire and 4-wire RTD channels to be used as the reference channel for subsequent thermocouple measurements that specify an external reference source.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:REFerence {OFF | ON}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:REFerence? [(@<ch_list>)]

Parameters <boolean> := { 0 | 1 | OFF | ON }

<ch_list> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:REF ON

query TEMP:TRAN:RTD:REF?

17.56 [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:RESistance[:REFerence]

Description Selects the nominal resistance (R0) for 2-wire and 4-wire RTD measurements. R0 is the nominal resistance of an RTD at 0 °C.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:RESistance[:REFerence] {<resistance> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{RTD | FRTD}:RESistance[:REFerence]? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**resistance**> := { 100 to 1000Ω)±20%; DEF: 100Ω }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:RTD:RES 1000

query TEMP:TRAN:RTD:RES?

17.57 [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:POWer:LIM

Description Enables or disables low-power for 2-wire and 4-wire thermistor measurements.

NOTICE Low-power resistance measurements apply to the 100 Ω through 100 kΩ ranges only. The 1 MΩ through 1 GΩ ranges source the same current regardless of the low-power setting.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:POWer:LIMit[:STAT {OFF | ON}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:POWer:LIMit[:STAT [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:THER:POW:LIM ON

query TEMP:TRAN:THER:POW:LIM?

17.58 [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:REFerence

Description Enables (On) or disables (Off) the specified 2-wire and 4-wire thermistor channels to be used as the reference channel for subsequent thermocouple measurements that specify an external reference source.

Syntax

command [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:REFerence {OFF | ON}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:REFerence? [(@<ch_list>)]

Parameters

<**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example

command TEMP:TRAN:THER:REF ON

query TEMP:TRAN:THER:REF?

17.59 [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:TYPE

Description Sets or returns the 2-wire and 4-wire thermistor sensor type.

Syntax

command [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:TYPE {<sensor_type> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:TYPE? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters

<**sensor_type**> := { 2.2kΩ | 5kΩ | 10kΩ | USER; DEF: 5kΩ }

<**ch_list**> := { 1 to 420}

Example

command TEMP:TRAN:THER:TYPE 2200

query TEMP:TRAN:THER:TYPE?

17.60 [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:AVAL

Description Sets or returns the 2-wire and 4-wire thermistor a coefficient.

Syntax

command [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:AVALue {<coefficient> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:AVALue? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**coefficient**> := { 0.0 to 9.9999; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:FTH :USER:AVAL 0.002154

query TEMP:TRAN:FTH :USER:AVAL?

17.61 [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:BVAl

Description Sets or returns the 2-wire and 4-wire thermistor b coefficient.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:BVALue {<coefficient> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:BVALue? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**coefficient**> := { 0.0 to 9.9999; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:FTH :USER:BVAL 0.003425

query TEMP:TRAN:FTH :USER:BVAL?

17.62 [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:CVAl

Description Sets or returns the 2-wire and 4-wire thermistor c coefficient.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:CVALue {<coefficient> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:{THERmistor | FTHermistor}:USER:CVALue? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**coeffiecient**> := { 0.0 to 9.9999; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:FTH:USER:CVAL 0.006993

query TEMP:TRAN:FTH:USER:CVAL?

17.63 [SENSe:]TEMPerature:TRANsducer:TCouple:CHECk

Description Enables or disables the thermocouple check feature to verify that your thermocouples are properly connected for measurements. When enabled, the instrument measures the resistance after each thermocouple measurement to ensure a proper connection. If an open connection is detected (greater than 5 kΩ on the 10 kΩ range), the instrument reports an overload condition.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:TCouple:CHECk {OFF | ON}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:TCouple:CHECk? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:TC:CHEC ON

query TEMP:TRAN:TC:CHEC?

17.64 [SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction

Description Sets the fixed reference junction temperature in degrees Celsius (°C) for thermocouple measurements on the specified channels.

NOTICE For this command, you must always specify the temperature in degrees Celsius regardless of the temperature units currently selected (see UNIT:TEMPerature command).

Syntax **command** [SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction {<temperature> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**temperature**> := { -20 +80; DEF: 0 }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:TC:RJUN 25

query TEMP:TRAN:TC:RJUN?

17.65 [SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE

Description Selects the reference junction source for thermocouple measurements on the specified channels.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE <reference>[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE? [(@<ch_list>)]

Parameters <**reference**> := { INTernal | EXTeranl | FIXed }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:TC:RJUN:TYPE INT

query TEMP:TRAN:TC:RJUN:TYPE?

17.66 [SENSe:]TEMPerature:TRANsducer:TCouple:TYPE

Description Sets or returns the thermocouple sensor type.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:TCouple:TYPE <sensor_type>[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:TCouple:TYPE? [(@<ch_list>)]

Parameters <**sensor_type**> := { J | K | N | R | S | T | B | E }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:TC:TYPE J

query TEMP:TRAN:TC:TYPE?

17.67 [SENSe:]TEMPerature:TRANsducer:TYPE

Description Selects the transducer probe type to use for temperature measurements.

Syntax **command** [SENSe:]TEMPerature:TRANsducer:TYPE <probe_type>[,(@<ch_list>)]

query [SENSe:]TEMPerature:TRANsducer:TYPE? [(@<ch_list>)]

Parameters <> := { TCouple | RTD | FRTD | THERmistor | FTHermistor }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:TRAN:TYPE TC

query TEMP:TRAN:TYPE?

17.68 [SENSe:]TEMPerature:ZERO:AUTO

Description Enables or disables the autozero mode for temperature measurements.

Syntax **command** [SENSe:]TEMPerature:ZERO:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]TEMPerature:ZERO:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** TEMP:ZERO:AUTO ON

query TEMP:ZERO:AUTO?

17.69 [SENSe:]VOLTage:AC:BANDwidth

Description Sets or returns the bandwidth for AC voltage measurements.

Syntax **command** [SENSe:]VOLTage:AC:BANDwidth {<freq> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]VOLTage:AC:BANDwidth? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**freq**> := { 3 | 20 | 200Hz ; DEF: 20Hz }

<**ch_list**> := { 1 to 420}

Example **command** VOLT:AC:BAND 20

query VOLT:AC:BAND?

17.70 [SENSe:]VOLTage:{AC | DC}:RANGe

Description Selects a fixed range for AC and DC voltage measurements.

Syntax **command** [SENSe:]VOLTage:{AC | DC}:RANGe {<range> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]VOLTage:{AC | DC}:RANGe? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**range**> := **AC** := { 100mV | 1V | 10V | 100V | 400V; DEF: AUTO }

DC := { 100mV | 1V | 10V | 100V | 600V; DEF: AUTO }

<**ch_list**> := { 1 to 420}

Example **command** VOLT:AC:RANG 100

query VOLT:AC:RANG?

17.71 [SENSe:]VOLTage:{AC | DC}:RANGe:AUTO

Description Enables or disables autoranging for AC and DC voltage measurements.

NOTICE

Autorange thresholds:
Down range at: < 10% of range

Up range at: > 120% of range

Syntax **command** [SENSe:]VOLTage:{AC | DC}:RANGe:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]VOLTage:{AC | DC}:RANGe:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** VOLT:DC:RANG:AUTO ON

query VOLT:DC:RANG:AUTO?

17.72 [SENSe:]VOLTage[:DC]:APERture

Description Enables the aperture mode and sets the integration time in seconds (called aperture time) for DC voltage measurements.

Syntax **command** [SENSe:]VOLTage[:DC]:APERture {<seconds> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]VOLTage[:DC]:APERture? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <> := { 20µs to 1s; DEF: 100ms }

<**ch_list**> := { 1 to 420}

Example **command** VOLT:APER 0.1

query VOLT:APER ?

17.73 [SENSe:]VOLTage[:DC]:APERture:ENABle

Description Enables the setting of integration time in seconds (called aperture time) for DC voltage measurements. If aperture time mode is disabled , the integration time is set in PLC (power-line cycles).

Syntax

command [SENSe:]VOLTage[:DC]:APERture:ENABle {OFF | ON}[,(@<ch_list>)]

query [SENSe:]VOLTage[:DC]:APERture:ENABle? [(@<ch_list>)]

Parameters

<**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example

command VOLT:APER:ENAB ON

query VOLT:APER:ENAB?

17.74 [SENSe:]VOLTage[:DC]:IMPedance:AUTO

Description Enables or disables automatic input impedance mode for DC voltage measurements.

Syntax

command [SENSe:]VOLTage[:DC]:IMPedance:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]VOLTage[:DC]:IMPedance:AUTO? [(@<ch_list>)]

Parameters

<**boolean**> := { 0 | 1 | OFF | ON }

OFF := {The input impedance for DC voltage measurements is fixed at 10 MΩ for all ranges to minimize noise pickup.}

ON := {The input impedance for DC voltage measurements varies by range. It is set to "HI-Z" (>10 GΩ) for the 100 mV, 1 V, and 10 V ranges to reduce the effects of measurement loading errors on these lower ranges. The 100 V and 300 V ranges remain at a 10 MΩ input impedance.}

<**ch_list**> := { 1 to 420}

Example

command VOLT:DC:IMP:AUTO ON

command VOLT:DC:IMP:AUTO ON

17.75 [SENSe:]VOLTage[:DC]:NPLCycles

Description Sets or returns the integration time in number of power line cycles (PLCs) DC voltage measurements. Where one PLC is equal to 16.6 milliseconds.

Syntax **command** [SENSe:]VOLTage[:DC]:NPLCycles {<PLCs> | MIN | MAX | DEF}[,(@<ch_list>)]

query [SENSe:]VOLTage[:DC]:NPLCycles? [{(@<ch_list>) | MIN | MAX | DEF}]

Parameters <**PLCs**> := { 0.0016 | 0.0032 | 0.0042 | 0.0083 | 0.0125 | 0.025 | 0.05 | 0.15 | 0.6 | 1 | 3 | 12; DEF: 1 PLC }

<**ch_list**> := { 1 to 420}

Example **command** VOLT:NPLC 1

query VOLT:NPLC?

17.76 [SENSe:]VOLTage[:DC]:REFerence

Description Enables (On) or disables (Off) the specified DC voltage channels to be used as the reference channel for subsequent strain bridge measurements that specify an external excitation reference voltage source (see **[SENSe:]STRain:EXCitation:TYPE** command).

NOTICE The external DC voltage reference channel must be one channel lower than the subsequent strain channel.

Syntax **command** [SENSe:]VOLTage[:DC]:REFerence {OFF | ON}[,(@<ch_list>)]

query [SENSe:]VOLTage[:DC]:REFerence? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** VOLT:REF ON

query VOLT:REF?

17.77 [SENSe:]VOLTage[:DC]:ZERO:AUTO

Description Enables or disables the autozero mode for DC voltage measurements.

Syntax **command** [SENSe:]VOLTage[:DC]:ZERO:AUTO {OFF | ON}[,(@<ch_list>)]

query [SENSe:]VOLTage[:DC]:ZERO:AUTO? [(@<ch_list>)]

Parameters <**boolean**> := { 0 | 1 | OFF | ON }

<**ch_list**> := { 1 to 420}

Example **command** VOLT:ZERO:AUTO ON

query VOLT:ZERO:AUTO?

Status Subsystem

The status subsystem is used for monitoring and reporting the state of an instrument. It includes status registers, event queues, and enable masks that help track errors, warnings, and operational states.

18.1 STATus:ALARm:CONDition?

Description Returns the total number of the Alarm Condition register.

NOTICE

A condition register continuously monitors the state of the instrument. Condition register bits are updated in real time; they are neither latched nor buffered.

This register is read-only; bits are not cleared when read.

Syntax **query** STATus:ALARm:CONDition?

Parameters None

Example **query** STAT:ALAR:COND?

18.2 STATus:ALARm:ENABle

Description Sets or returns bits in the Alarm Enable register.

NOTICE

selected bits are then reported to the Status Byte. An enable register defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an enable register

Syntax **command** STATus:ALARm:ENABle <enable>

query STATus:ALARm:ENABle?

Parameters <**enable**> := { 0 to 32767 }

<**ch_list**> := { 1 to 420}

Example **command** STAT:ALAR:ENAB 7

18.3 STATus:ALARm[:EVENt]?

Description Returns the total number of the Alarm Event register.

NOTICE

An event register is a read-only register that latches events from the condition register. While an event bit is set, subsequent events corresponding to that bit are ignored.
Once a bit is set, it remains set until cleared by reading the event register or by sending *CLS (clear status).This register is read-only; bits are not cleared when read.

Syntax **query** STATus:ALARm[:EVENt]?

Parameters None

Example **command** STAT:ALAR:EVEN?

18.4 STATus:OPERation:CONDition?

Description Returns the total number of the Operation Condition register.

NOTICE

A condition register continuously monitors the state of the instrument. Condition register bits are updated in real time; they are neither latched nor buffered.
This register is read-only; bits are not cleared when read.

Syntax **query** STATus:OPERation:CONDition?

Parameters None

Example **command** STAT:OPER:COND?

18.5 STATus:OPERation:ENABle

Description Sets or returns bits in the Operation Enable register.

NOTICE

The selected bits are then reported to the Status Byte. An enable register defines which bits in the event register will be reported to the Status Byte register group.
You can write to or read from an enable register.

Syntax **command** STATus:OPERation:ENABle <enable>

query STATus:OPERation:ENABle?

Parameters <**enable**> := { 0 to 32767 }

<**ch_list**> := { 1 to 420}

Example **command** STAT:OPER:ENAB 10

query STAT:OPER:ENAB?

18.6 STATus:OPERation[:EVENt]?

Description Returns the total number of the Operation Event register.

NOTICE

An event register is a read-only register that latches events from the condition register. While an event bit is set, subsequent events corresponding to that bit are ignored.
Once a bit is set, it remains set until cleared by reading the event register or by sending *CLS (clear status).

Syntax **query** STATus:OPERation[:EVENt]?

Parameters None

Example **query** STAT:OPER:EVEN?

18.7 STATus:PRESet

Description Clears all enable register bits in Alarm Register, Standard Operation Register, and Questionable Data Register.

Syntax **command** STATus:PRESet

Parameters None

Example **command** STAT:PRES

18.8 STATus:QUEStionable:CONDition?

Description Returns the total number of the Questionable Condition register.

NOTICE

A condition register continuously monitors the state of the instrument. Condition register bits are updated in real time; they are neither latched nor buffered.

Syntax **query** STATus:QUEStionable:CONDition?

Parameters None

Example **command** STAT:QUES:COND?

18.9 STATus:QUEStionable:ENABle

Description Sets or returns bits in the Ouestionable Enable register.

NOTICE

The selected bits are then reported to the Status Byte. An enable register defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an enable register.

The selected bits are then reported to the Status Byte. An enable register defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an enable register.

A **STATus:PRESet** clears all bits in the enable register.

The *PSC command controls whether the enable register is cleared at power on.

Syntax **command** STATus:QUEStionable:ENABle <enable>

query STATus:QUEStionable:ENABle?

Parameters <**enable**> := { 0 to 32767 }

<**ch_list**> := { 1 to 420}

Example **command** STAT:QUES:ENAB 4099

18.10 STATus:QUEStionable[:EVENt]?

Description Returns the total number of the Ouestionable Event register. Return parameter: <NR1>

NOTICE

An event register is a read-only register that latches events from the condition register. While an event bit is set, subsequent events corresponding to that bit are ignored.

Once a bit is set, it remains set until cleared by reading the event register or by sending *CLS (clear status).

Syntax **query** STATus:QUEStionable[:EVENt]?

Parameters None

Example **query** STAT:QUES:EVEN?

Version: March 12, 2025