

# 1586A

SUPER-DAQ Precision Temperature Scanner

## Remote Programmers Guide

July 2013

© 2013 Fluke Corporation. All rights reserved. Specifications are subject to change without notice.  
All product names are trademarks of their respective companies.

## LIMITED WARRANTY AND LIMITATION OF LIABILITY

Each Fluke product is warranted to be free from defects in material and workmanship under normal use and service. The warranty period is one year and begins on the date of shipment. Parts, product repairs, and services are warranted for 90 days. This warranty extends only to the original buyer or end-user customer of a Fluke authorized reseller, and does not apply to fuses, disposable batteries, or to any product which, in Fluke's opinion, has been misused, altered, neglected, contaminated, or damaged by accident or abnormal conditions of operation or handling. Fluke warrants that software will operate substantially in accordance with its functional specifications for 90 days and that it has been properly recorded on non-defective media. Fluke does not warrant that software will be error free or operate without interruption.

Fluke authorized resellers shall extend this warranty on new and unused products to end-user customers only but have no authority to extend a greater or different warranty on behalf of Fluke. Warranty support is available only if product is purchased through a Fluke authorized sales outlet or Buyer has paid the applicable international price. Fluke reserves the right to invoice Buyer for importation costs of repair/replacement parts when product purchased in one country is submitted for repair in another country.

Fluke's warranty obligation is limited, at Fluke's option, to refund of the purchase price, free of charge repair, or replacement of a defective product which is returned to a Fluke authorized service center within the warranty period.

To obtain warranty service, contact your nearest Fluke authorized service center to obtain return authorization information, then send the product to that service center, with a description of the difficulty, postage and insurance prepaid (FOB Destination). Fluke assumes no risk for damage in transit. Following warranty repair, the product will be returned to Buyer, transportation prepaid (FOB Destination). If Fluke determines that failure was caused by neglect, misuse, contamination, alteration, accident, or abnormal condition of operation or handling, including overvoltage failures caused by use outside the product's specified rating, or normal wear and tear of mechanical components, Fluke will provide an estimate of repair costs and obtain authorization before commencing the work. Following repair, the product will be returned to the Buyer transportation prepaid and the Buyer will be billed for the repair and return transportation charges (FOB Shipping Point).

THIS WARRANTY IS BUYER'S SOLE AND EXCLUSIVE REMEDY AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FLUKE SHALL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OR LOSSES, INCLUDING LOSS OF DATA, ARISING FROM ANY CAUSE OR THEORY.

Since some countries or states do not allow limitation of the term of an implied warranty, or exclusion or limitation of incidental or consequential damages, the limitations and exclusions of this warranty may not apply to every buyer. If any provision of this Warranty is held invalid or unenforceable by a court or other decision-maker of competent jurisdiction, such holding will not affect the validity or enforceability of any other provision.

Fluke Corporation  
P.O. Box 9090  
Everett, WA 98206-9090  
U.S.A.

Fluke Europe B.V.  
P.O. Box 1186  
5602 BD Eindhoven  
The Netherlands

11/99

To register your product online, visit [register.fluke.com](http://register.fluke.com)

# Table of Contents

Title	Page
Introduction.....	1
How to Contact Fluke Calibration .....	1
Remote Operation Commands .....	2
System Status Diagram .....	3
List of Commands.....	4
Remote Programming Examples .....	14
Single Channel, One-Shot .....	14
One-Shot Scan .....	15
Single Channel, Continuous .....	15
Continuous Scanning.....	16
Number of Scan Sweeps.....	17
Full-Speed Sampling .....	18
Scanning with Scan Interval .....	18
External Trigger.....	19
Monitoring.....	20
Alarm Trigger.....	21
Alphabetical List of Serial Commands .....	22



## ***Introduction***

This document defines the remote interface commands for the 1586A SUPER-DAQ Precision Temperature Scanner (the Product or Instrument). These commands may be used by a computer connected through any of the remote interface ports to set settings, read measurement data, and control the operation of the instrument. Command syntax and names follow the IEEE-488.2 and SCPI standards.

## ***How to Contact Fluke Calibration***

To contact Fluke Calibration, call one of the following telephone numbers:

- Technical Support USA: 1-877-355-3225
- Calibration/Repair USA: 1-877-355-3225
- Canada: 1-800-36-FLUKE (1-800-363-5853)
- Europe: +31-40-2675-200
- Japan: +81-3-6714-3114
- Singapore: +65-6799-5566
- China: +86-400-810-3435
- Brazil: +55-11-3759-7600
- Anywhere in the world: +1-425-446-6110

To see product information and download the latest manual supplements, visit Fluke Calibration's website at [www.flukecal.com](http://www.flukecal.com).

To register your product, visit <http://flukecal.com/register-product>.

## **Remote Operation Commands**

The Instrument accepts commands for setting parameters, executing functions or responding with requested data. These commands are in the form of strings of ASCII-encoded characters.

Commands consist of a command header and, if necessary, parameter data. All commands must be terminated with either a carriage return (ASCII 0D hex or 13 decimal) or new line character (ASCII 0A hex or 10 decimal).

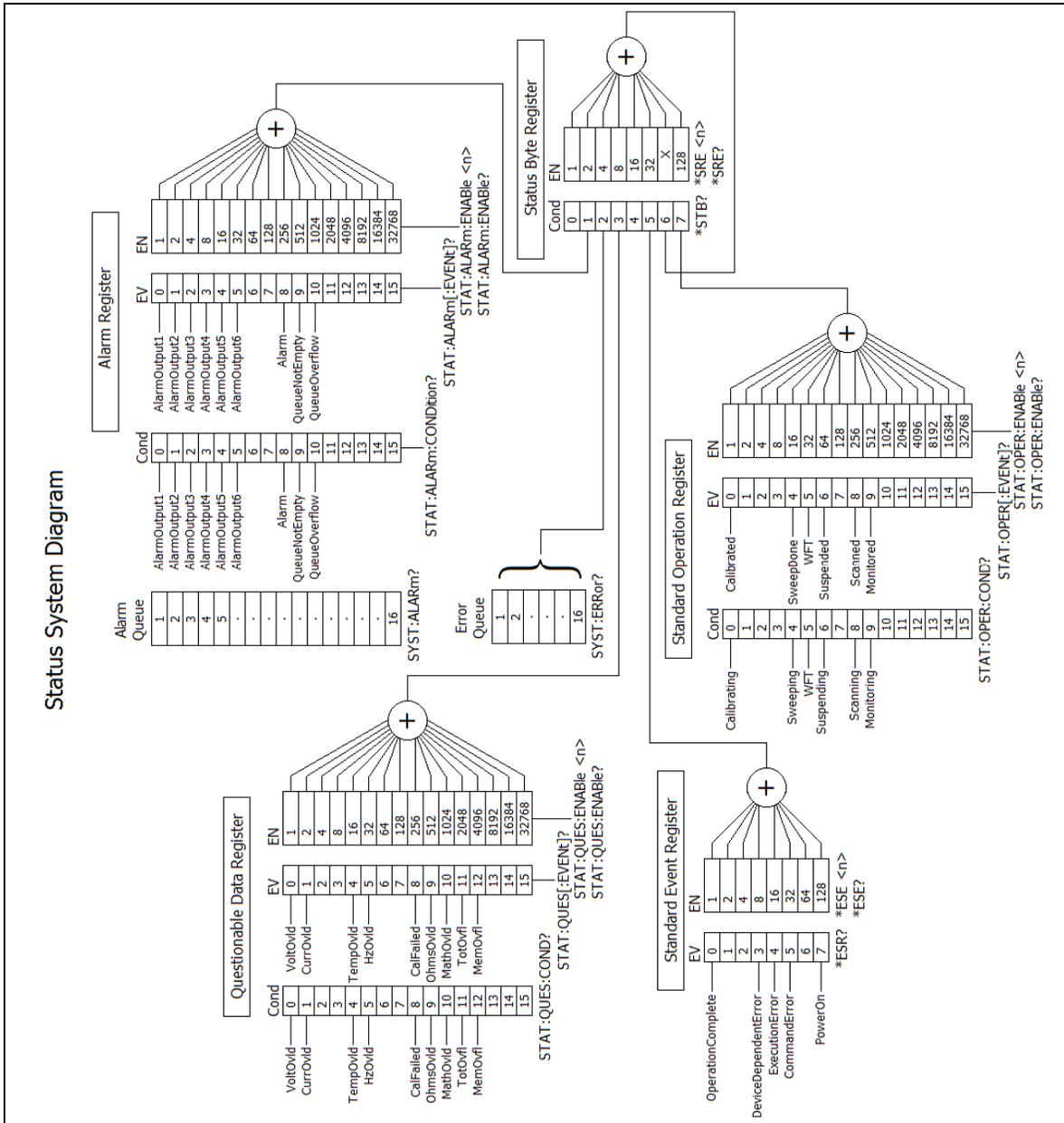
Command headers consist of one or more mnemonics separated by colons (:). Mnemonics may use letter characters, the underscore character (\_), and possibly numeric digits as well. Commands are not case sensitive. Mnemonics often have alternate forms. Most mnemonics have a long form that is more readable and a short form consisting of three or four characters that is more efficient.

Query commands are commands that request data in response. Query commands have a question mark (?) immediately following the command header. Responses to query commands are generated immediately and placed in the output buffer. Responses are then transmitted automatically to the PC.

Some commands require parameter data to specify values for one or more parameters. The command header is separated from the parameter data by a space (ASCII 20 hex or 32 decimal). Multiple parameters are separated by a comma(,).

## System Status Diagram

Figure 1 shows the System Status Diagram.



status.png

Figure 1. System Status Diagram

## List of Commands

Table 1 lists the command set for the Product in alphabetical order.

**Table 1. List of Commands**

<b>*CLS</b>	Clear all status registers and the error queue.
<b>*ESE &lt;n&gt;</b>	Set the Standard Event Status Enable register.
<b>*ESR?</b>	Query and clear the Standard Event Status register.
<b>*IDN?</b>	Query the instrument identification.
<b>*OPC</b>	Set the operation complete bit of the Standard Event Enable Register when all pending operations are completed.
<b>*OPC?</b>	Return "1" when all pending operations are completed.
<b>*OPT?</b>	Query the installed modules.
<b>*PSC &lt;boolean&gt;</b>	Enable or disable clearing of status enable registers at power on.
<b>*RCL &lt;n&gt;</b>	Recall a setup file.
<b>*RST</b>	Reset operating settings to default states.
<b>*SAV &lt;n&gt;</b>	Save a setup file.
<b>*SRE &lt;n&gt;</b>	Set the Service Request Enable register.
<b>*STB?</b>	Query the Status Byte Register.
<b>*TRG</b>	Trigger a single scan sweep.
<b>*TST?</b>	Perform an instrument self-test and return the results.
<b>*WAI</b>	Force the command parser to wait until all pending operations complete.
<b>ABORt</b>	Stop scanning.
<b>CALCulate:AVERage:AVERage? [(@&lt;ch_list&gt;)]</b>	Query the mean of the set of readings of the given channels.
<b>CALCulate:AVERage:CLEar [(@&lt;ch_list&gt;)]</b>	Clear the statistics and restart the sample set for the given channels.
<b>CALCulate:AVERage:CLEar:ALL</b>	Clear the statistics and restart the sample set for all channels.

**Table 1. List of Commands (cont.)**

<b>CALCulate:AVERage:COUNT? [(@&lt;ch_list&gt;)]</b>
Query the number of readings in the statistics.
<b>CALCulate:AVERage:MAXimum? [(@&lt;ch_list&gt;)]</b>
Query the maximum of each of the given channels.
<b>CALCulate:AVERage:MAXimum:TIME? [(@&lt;ch_list&gt;)]</b>
Query the time stamp of the maximum of each of the given channels.
<b>CALCulate:AVERage:MINimum? [(@&lt;ch_list&gt;)]</b>
Query the minimum of each of the given channels.
<b>CALCulate:AVERage:MINimum:TIME? [(@&lt;ch_list&gt;)]</b>
Query the time stamp of the minimum of each of the given channels.
<b>CALCulate:AVERage:PTPeak? [(@&lt;ch_list&gt;)]</b>
Query peak-to-peak statistics.
<b>CALCulate:AVERage:RATE? [(@&lt;ch_list&gt;)]</b>
Query rate of change.
<b>CALCulate:AVERage:RATE:BASE &lt;time base&gt;[,(@&lt;ch_list&gt;)]</b>
Set the rate of change time base.
<b>CALCulate:AVERage:SDEViation? [(@&lt;ch_list&gt;)]</b>
Query standard deviation statistics.
<b>CALCulate:LIMit{1 2} &lt;n&gt;[,(@&lt;ch_list&gt;)]</b>
Set the first alarm limit of the given channels.
<b>CALCulate:LIMit:CLEar [(@&lt;ch_list&gt;)]</b>
Clear the alarms for the given channels.
<b>CALCulate:LIMit:CLEar:ALL</b>
Clear all alarms.
<b>CALCulate:LIMit:FAIL? [(@&lt;ch_list&gt;)]</b>
Query the alarm condition for a channel.
<b>CALCulate:LIMit{1 2}:FEED &lt;n&gt;[,(@&lt;ch_list&gt;)]</b>
Associate a channel alarm to an alarm port bit.
<b>CALCulate:LIMit{1 2}:STATe &lt;type&gt;[,(@&lt;ch_list&gt;)]</b>
Set the alarm type for the given channels.
<b>CALCulate:MATH:COEFFicients &lt;c0&gt;[,&lt;c1&gt;[,&lt;c2&gt;[,&lt;c3&gt;]]][,(@&lt;ch_list&gt;)]</b>
Set coefficients for math functions.
<b>CALCulate:MATH:EXPOnent &lt;exponent&gt;[,(@&lt;ch_list&gt;)]</b>
Set the exponent for math power function.
<b>CALCulate:MATH:FUNCTion &lt;function&gt;[,(@&lt;ch_list&gt;)]</b>
Select the math function for math calculations.
<b>CALCulate:MATH:SOURce:ACHannel (@&lt;A&gt;)[,(@&lt;ch_list&gt;)]</b>
Set the A source channel for math channel calculations.
<b>CALCulate:MATH:SOURce:BCHannel (@&lt;B&gt;)[,(@&lt;ch_list&gt;)]</b>
Set the B source channel for math channel calculations.
<b>CALCulate:MATH:SOURce:LIST (@&lt;source_list&gt;)[,(@&lt;ch_list&gt;)]</b>
Set the source channel list for math channel statistics calculations.
<b>CALCulate:MATH:UNIT "&lt;unit&gt;"[,(@&lt;ch_list&gt;)]</b>
Set the unit of measurement for a math calculations.
<b>CALCulate:SCALE:GAIN &lt;gain&gt;[,(@&lt;ch_list&gt;)]</b>
Set the Mx+B scaling gain (M).
<b>CALCulate:SCALE:OFFSet &lt;offset&gt;[,(@&lt;ch_list&gt;)]</b>
Set the Mx+B scaling offset (B).

Table 1. List of Commands (cont.)

<b>CALCulate:SCALE:STATE</b> <boolean>[,(@<ch_list>)]
Enable or disable Mx+B scaling.
<b>CALCulate:SCALE:UNIT</b> "<unit>"[,(@<ch_list>)]
Set the unit of measurement for Mx+B scaling.
<b>CALibrate:DATE?</b>
Read the instrument calibration date.
<b>CALibrate:MODule:DATE?</b> <slot>
Read the module calibration date.
<b>CONFigure?</b> (@<channel>)
Query the measurement function of a channel.
<b>CONFigure:CURRent[:DC]</b> [<range>,@<ch_list>]
Configure channels to measure dc current.
<b>CONFigure:FRESistance</b> [<range>,@<ch_list>]
Configure channels to measure four-wire resistance.
<b>CONFigure:RESistance</b> [<range>,@<ch_list>]
Configure channels to measure two-wire resistance.
<b>CONFigure:TEMPerature</b> <sensor>,<type>,@<ch_list>]
Configure channels to measure temperature.
<b>CONFigure:VOLTage[:DC]</b> [<range>,@<ch_list>]
Configure channels to measure dc voltage.
<b>DATA:CLEar</b>
Clear scan memory.
<b>DATA[:LAST]?</b> [(@<channel>)]
Get from scan memory the most recent reading of the given channel.
<b>DATA:LOG:AUTO</b> <boolean>
Enable or disable automatic data recording.
<b>DATA:LOG:DESTination</b> <memory>
Select the memory device for data recording.
<b>DATA:LOG[:ENABLE]</b> <boolean>
Enable or disable data recording.
<b>DATA:POINts?</b>
Query the total number of readings stored in scan memory.
<b>DATA:READ?</b>
Get scan sweep readings from scan memory and delete the readings.
<b>DIAGnostic:CHANnel:RELAy:CYCLes?</b> (@<ch_list>)
Query the relay cycle counts of the given channels.
<b>DIAGnostic:CHANnel:RELAy:CYCLes:CLEar</b> (@<ch_list>)
Clear the relay cycle counts of the given channels.
<b>DIAGnostic:DMM:RELAy:CYCLes?</b>
Query the cycle counts of the DMM circuit relays.

**Table 1. List of Commands (cont.)**

<b>DISPlay:BRIGhtness &lt;brightness&gt;</b> Set the display brightness.
<b>DISPlay:DATE:FORMat &lt;date format&gt;</b> Set the display date format.
<b>DISPlay:DECimal:FORMat &lt;decimal format&gt;</b> Set the display decimal format.
<b>DISPlay:LANGUage &lt;language&gt;</b> Set the display language.
<b>DISPlay:SAVer &lt;n&gt;</b> Set the screen saver time.
<b>DISPlay:STATe &lt;boolean&gt;</b> Switch the UI display off or on.
<b>DISPlay:TIME:FORMat &lt;time format&gt;</b> Set the display time format.
<b>FETCh?</b> Get readings from scan memory.
<b>INITiate[:IMMediate]</b> Initiate measuring/scanning.
<b>MEASure:CURRent[:DC]? [&lt;range&gt;,@&lt;ch_list&gt;]</b> Acquire and return one scan of readings of dc current from the given channels.
<b>MEASure:DIGital:DATA:BYTE?</b> Acquire and return a reading of the digital input channel.
<b>MEASure:FRESistance? [&lt;range&gt;,@&lt;ch_list&gt;]</b> Acquire and return one scan of readings of four-wire resistance from the given channels.
<b>MEASure:RESistance? [&lt;range&gt;,@&lt;ch_list&gt;]</b> Acquire and return one scan of readings of two-wire resistance from the given channels.
<b>MEASure:TEMPerature? &lt;sensor&gt;,&lt;type&gt;,@&lt;ch_list&gt;]</b> Acquire and return one scan of readings of temperature from the given channels.
<b>MEASure:TOTalize?</b> Acquire and return a reading of the digital totalize port.
<b>MEASure:VOLTage[:DC]? [&lt;range&gt;,@&lt;ch_list&gt;]</b> Acquire and return one scan of readings of dc voltage from the given channels.
<b>MEMory:LOG:CLEar</b> Erase all data files stored in internal memory.
<b>MEMory:LOG:DELeTe ["&lt;name&gt;"]</b> Delete a data file stored in internal memory.
<b>MEMory:LOG:EXPort ["&lt;name&gt;"]</b> Copy a stored data file to USB memory.
<b>MEMory:LOG:FREE?</b> Query the available space in internal data memory.
<b>MEMory:LOG:NAME? &lt;n&gt;</b> Get a filename of specified data file stored in internal memory.

Table 1. List of Commands (cont.)

<b>MEMory:LOG:NFIles?</b>	Query the number of data files stored in internal memory.
<b>MEMory:LOG:PROPERTIES?</b> ["<name>"]	Query the properties of a data file stored in internal memory.
<b>MEMory:LOG:READ[:DATA]?</b> ["<name>"]	Read a data file stored in internal memory.
<b>MEMory:LOG:READ:CONF?</b> ["<name>"]	Read configuration of a data file stored in internal memory.
<b>MEMory:LOG:SECurity</b>	Enable or disable data file security.
<b>MEMory:LOG:STORE</b>	Store the latest scan data to internal memory.
<b>MEMory:NSTATes?</b>	Query the possible number of setup files.
<b>MEMory:STATE:CLEar</b>	Erase all setup files stored in memory.
<b>MEMory:STATE:DEFine</b> "<name>",<n>	Assign a name to a setup file storage location.
<b>MEMory:STATE:DEFine:NAME?</b> <n>	Query the name associated with a storage location.
<b>MEMory:STATE:DELeTe</b> <n>	Delete a setup file.
<b>MEMory:STATE:VALid?</b> <n>	Query the existence of a setup file.
<b>MMEMemory:LOG:NAME?</b> <n>	Get a filename of specified data file stored in USB memory.
<b>MMEMemory:LOG:NFIles?</b>	Query the number of data files stored in USB memory.
<b>MMEMemory:LOG:READ[:DATA]?</b> ["<name>"]	Read a data file stored in USB memory.
<b>MMEMemory:LOG:READ:CONF?</b> ["<name>"]	Read configuration of a data file stored in USB memory.
<b>MMEMemory:LOG:STORE</b>	Store scan data to USB memory.
<b>MMEMemory:MOUNT?</b>	Query the mount state of the USB memory device.
<b>OUTPut:ALARm?</b>	Query the alarm output.
<b>READ?</b>	Initiate one scan sweep and return the readings.
<b>ROUTE:CHANnel:DELay</b> <delay>,(@<ch_list>)	Set the channel delay.

**Table 1. List of Commands (cont.)**

<b>ROUTe:CHANnel:NAME "&lt;name&gt;",(@&lt;ch_list&gt;)</b>
Set the name associated with a channel.
<b>ROUTe:CHANnel:STATe &lt;boolean&gt;,(@&lt;ch_list&gt;)</b>
Enable or disable channels for scanning.
<b>ROUTe:MONitor (@&lt;channel&gt;)</b>
Select and enable a channel for monitoring during scanning.
<b>ROUTe:MONitor:DATA?</b>
Return the latest reading from the monitored channel.
<b>ROUTe:MONitor:STATe &lt;boolean&gt;</b>
Enable or disable channel monitoring during scanning.
<b>ROUTe:SCAN (@&lt;ch_list&gt;)</b>
Set the channel scan list.
<b>ROUTe:SCAN:RESume &lt;boolean&gt;</b>
Enable or disable automatic resume scan at power on.
<b>[SENSe:]CURRent[:DC]:RANGe &lt;range&gt;[,(@&lt;ch_list&gt;)]</b>
Set the range for dc current.
<b>[SENSe:]CURRent[:DC]:RANGe:AUTO &lt;boolean&gt;[,(@&lt;ch_list&gt;)]</b>
Enable or disable auto-ranging for dc current.
<b>[SENSe:]DIGital:DATA:BYTE?</b>
Read the digital I/O port.
<b>[SENSe:]FREStance:RANGe &lt;range&gt;[,(@&lt;ch_list&gt;)]</b>
Set the four-wire resistance range.
<b>[SENSe:]FREStance:RANGe:AUTO &lt;boolean&gt;[,(@&lt;ch_list&gt;)]</b>
Enable or disable auto-ranging for four-wire resistance.
<b>[SENSe:]FUNctIon "&lt;function&gt;"[,(@&lt;ch_list&gt;)]</b>
Set the measurement function of the given channels.
<b>[SENSe:]RATE &lt;rate&gt;</b>
Set the sample rate.
<b>[SENSe:]RESistance:RANGe &lt;range&gt;[,(@&lt;ch_list&gt;)]</b>
Set the two-wire resistance range.
<b>[SENSe:]RESistance:RANGe:AUTO &lt;boolean&gt;[,(@&lt;ch_list&gt;)]</b>
Enable or disable auto-ranging for two-wire resistance.
<b>[SENSe:]TEMPerature:CALCulate? &lt;n&gt;[,&lt;RJT&gt;],(@&lt;channel&gt;)</b>
Return the temperature calculation for a given channel using the given input values.
<b>[SENSe:]TEMPerature:FRTD:A385:RZERo &lt;n&gt;,(@&lt;channel&gt;)</b>
Set the 0 °C resistance of a four-wire PRT/RTD of type A385.
<b>[SENSe:]TEMPerature:FRTD:A392:RZERo &lt;n&gt;,(@&lt;channel&gt;)</b>
Set the 0 °C resistance of a four-wire PRT/RTD of type A392.
<b>[SENSe:]TEMPerature:FRTD:ABC:COEFFicients &lt;a&gt;,&lt;b&gt;,&lt;c&gt;,(@&lt;channel&gt;)</b>
Set the polynomial coefficients A, B, and C for a four-wire PRT/RTD of type ABC.
<b>[SENSe:]TEMPerature:FRTD:ABC:RZERo &lt;n&gt;,(@&lt;channel&gt;)]</b>
Set the 0 °C resistance of a four-wire PRT/RTD of type ABC.

Table 1. List of Commands (cont.)

[SENSe:]TEMPerature:FRTD:CALCulate:RESistance <boolean>,(@<ch_list>)	Enable or disable four-wire PRT/RTD resistance calculation.
[SENSe:]TEMPerature:FRTD:SPRT:COEFFicients:HIGH <a>[,<b>[,<c>[,<d>]]](@<channel>)	Set the high temperature range polynomial coefficients A, B, C, and D for a four-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:FRTD:SPRT:COEFFicients:LOW <a>[,<b>](@<channel>)	Set the low temperature range polynomial characterization coefficients A4 and B4 for a four-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:FRTD:SPRT:RTPW <n>,(@<channel>)	Set the 0.01 °C resistance of a four-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:FRTD:TYPE <type>,(@<ch_list>)	Select the four-wire PRT/RTD type.
[SENSe:]TEMPerature:FTHerMistor:CALCulate:RESistance <boolean>,(@<ch_list>)	Enable or disable four-wire thermistor resistance calculation.
[SENSe:]TEMPerature:FTHerMistor:RPOL:COEFFicients <c0>,<c1>,<c2> ,<c3>,(@<channel>)	Set the characterization coefficients for the four-wire thermistor type RPOL.
[SENSe:]TEMPerature:FTHerMistor:TYPE <type>,(@<ch_list>)	Set the type and characterization for four-wire thermistor.
[SENSe:]TEMPerature:RJUNction? (@<ch_list>)	Read the internal reference junction temperature.
[SENSe:]TEMPerature:RTD:A385:RZERo <n>,(@<channel>)	Set the 0 °C resistance of a two-wire PRT/RTD of type A385.
[SENSe:]TEMPerature:RTD:A392:RZERo <n>,(@<channel>)	Set the 0 °C resistance of a two-wire PRT/RTD of type A392.
[SENSe:]TEMPerature:RTD:ABC:COEFFicients <a>,<b>,<c>,(@<channel>)	Set the polynomial coefficients A, B, and C for a two-wire PRT/RTD of type ABC.
[SENSe:]TEMPerature:RTD:ABC:RZERo <n>,(@<channel>)]	Set the 0 °C resistance of a two-wire PRT/RTD of type ABC.
[SENSe:]TEMPerature:RTD:CALCulate:RESistance <boolean>,(@<ch_list>)	Enable or disable two-wire PRT/RTD resistance calculation.
[SENSe:]TEMPerature:RTD:SPRT:COEFFicients:HIGH <a>[,<b>[,<c>[,<d>]]],(@<channel>)	Set the high temperature range polynomial coefficients A, B, C, and D for a two-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:RTD:SPRT:COEFFicients:LOW <a>[,<b>],(@<channel>)	Set the low temperature range polynomial characterization coefficients A4 and B4 for a two-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:RTD:SPRT:RTPW <n>,(@<channel>)	Set the 0.01 °C resistance of a two-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:RTD:TYPE <type>,(@<ch_list>)	Select the two-wire PRT/RTD type.
[SENSe:]TEMPerature:TCouple:CALCulate:VOLTagE <boolean>,(@<ch_list>)	Enable or disable thermocouple voltage calculation.
[SENSe:]TEMPerature:TCouple:ODETect <boolean>,(@<ch_list>)	Enable or disable thermocouple open-circuit detection.
[SENSe:]TEMPerature:TCouple:POLY:COEFFicients <c0>[,<c1> [,<c2>[,<c3>[,<c4>[,<c5>[,<c6>]]]]],(@<channel>)	Set the characterization coefficients for a thermocouple of type POLY.
[SENSe:]TEMPerature:TCouple:R:COEFFicients <c0>[,<c1>[,<c2> [,<c3>]]],(@<channel>)	Set the calibration coefficients for a thermocouple of type R.

**Table 1. List of Commands (cont.)**

[SENSe:]TEMPerature:TCouple:RJUNction <n>,(@<ch_list>)	Set the reference junction temperature for the fixed reference junction type.
[SENSe:]TEMPerature:TCouple:RJUNction:TYPE <type>,(@<ch_list>)	Set the reference junction type for a thermocouple.
[SENSe:]TEMPerature:TCouple:S:COEFFicients <c0>[,<c1>[,<c2> [,<c3>]]],(@<channel>)	Set the calibration coefficients for a thermocouple of type S.
[SENSe:]TEMPerature:TCouple:TYPE <type>,(@<ch_list>)	Select the thermocouple type.
[SENSe:]TEMPerature:THERmistor:CALCulate:RESistance <boolean>,(@<ch_list>)	Enable or disable two-wire thermistor resistance calculation.
[SENSe:]TEMPerature:THERmistor:RPOL:COEFFicients <c0>,<c1>,<c2>,<c3>(@<channel>)	Set the characterization coefficients for a two-wire thermistor of type RPOLY.
[SENSe:]TEMPerature:THERmistor:TYPE <type>,(@<ch_list>)	Set the type and characterization for two-wire thermistor.
[SENSe:]TEMPerature:TRANsducer <sensor>,(@<ch_list>)	Select the type of transducer for temperature measurements.
[SENSe:]TEMPerature:TRTD:A385:RZERo <n>,(@<channel>)	Set the 0 °C resistance of a three-wire PRT/RTD of type A385.
[SENSe:]TEMPerature:TRTD:A392:RZERo <n>,(@<channel>)	Set the 0 °C resistance of a three-wire PRT/RTD of type A392.
[SENSe:]TEMPerature:TRTD:ABC:COEFFicients <a>,<b>,<c>,(@<channel>)	Set the polynomial coefficients A, B, and C for a three-wire PRT/RTD of type ABC.
[SENSe:]TEMPerature:TRTD:ABC:RZERo <n>,(@<channel>)	Set the 0 °C resistance of a three-wire PRT/RTD of type ABC.
[SENSe:]TEMPerature:TRTD:CALCulate:RESistance <boolean>,(@<ch_list>)	Enable or disable three-wire PRT/RTD resistance calculation.
[SENSe:]TEMPerature:TRTD:SPRT:COEFFicients:HIGH <a>[,<b>[,<c>[,<d>]]],(@<channel>)	Set the high temperature range polynomial coefficients A, B, C, and D for a three-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:TRTD:SPRT:COEFFicients:LOW <a>[,<b>],(@<channel>)	Set the low temperature range polynomial characterization coefficients A4 and B4 for a three-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:TRTD:SPRT:RTPW <n>,(@<channel>)	Set the 0.01 °C resistance of a three-wire PRT/RTD of type SPRT.
[SENSe:]TEMPerature:TRTD:TYPE <type>,(@<ch_list>)	Select the three-wire PRT/RTD type.
[SENSe:]TOTalize:CLEar	Clear the totalizer count.
[SENSe:]TOTalize:DATA?	Query the totalizer count.
[SENSe:]TOTalize:DEBounce <boolean>	Enable or disable totalizer debounce.
[SENSe:]TOTalize:STATe <boolean>	Enable or disable the totalizer counter.
[SENSe:]TOTalize:TYPE <mode>	Enable or disable automatic reset of the totalizer count.
[SENSe:]VOLTage[:DC]:RANGe <range>,(@<ch_list>)]	Set the range for dc voltage.
[SENSe:]VOLTage[:DC]:RANGe:AUTO <boolean>,(@<ch_list>)]	Enable or disable auto-ranging for dc voltage.
SOURce:DIGital:DATA:BYTE <n>	Set the digital port output byte.
STATus:ALARm:CONDition?	Query the Alarm Status Condition Register.

Table 1. List of Commands (cont.)

<b>STATus:ALARm:ENABLE &lt;n&gt;</b>	Set the Alarm Status Enable Register.
<b>STATus:ALARm[:EVENT]?</b>	Query and clear the Alarm Status Event Register.
<b>STATus:OPERation:CONDition?</b>	Query the Operation Status Condition Register.
<b>STATus:OPERation:ENABLE &lt;n&gt;</b>	Set the Operation Status Enable Register.
<b>STATus:OPERation[:EVENT]?</b>	Query and clear the Operation Status Event Register.
<b>STATus:PRESet</b>	Set status enable registers to disabled states.
<b>STATus:QUEStionable:CONDition?</b>	Query the Questionable Status Condition Register.
<b>STATus:QUEStionable:ENABLE &lt;n&gt;</b>	Set the Questionable Status Enable Register.
<b>STATus:QUEStionable[:EVENT]?</b>	Query and clear the Questionable Status Event Register.
<b>SYSTem:ALARm?</b>	Read and remove the first alarm in the alarm queue.
<b>SYSTem:BEEPer:KEY &lt;boolean&gt;</b>	Enable or disable the key beeper.
<b>SYSTem:COMMunicate:SERial:BAUD &lt;baud&gt;</b>	Set the serial port baud rate for temperature source control.
<b>SYSTem:COMMunicate:TERMinator {CR   LF   CRLF}</b>	Set the response termination character.
<b>SYSTem:CONFigure:MAC?</b>	Query the LAN MAC address.
<b>SYSTem:DATE &lt;year&gt;,&lt;month&gt;,&lt;day&gt;</b>	Set the date of the system clock.
<b>SYSTem:ERRor?</b>	Read and remove the next error in the error queue.
<b>SYSTem:KLOCK &lt;boolean&gt;</b>	Lock or unlock the front panel keys.
<b>SYSTem:LFRequency?</b>	Query the power line frequency.
<b>SYSTem:MODule:CONFigure:SNUM? &lt;slot&gt;</b>	Query the specified module's serial number.
<b>SYSTem:MODule:CONFigure:TYPE? &lt;slot&gt;</b>	Query the specified module's type.
<b>SYSTem:PON:TIME?</b>	Query the time elapsed since power-on.
<b>SYSTem:SECurity:IMMEDIATE</b>	Clear memory for security.
<b>SYSTem:TIME &lt;hour&gt;,&lt;minute&gt;,&lt;second&gt;</b>	Set the time of the system clock.
<b>SYSTem:VERSion?</b>	Query the SCPI version.

Table 1. List of Commands (cont.)

<b>TRIGger:ALARm:CHANnel (@&lt;channel&gt;)</b>
Set trigger channel for alarm trigger.
<b>TRIGger:AUTO:CONTRol &lt;boolean&gt;</b>
Enable or disable source instrument control for the automated test.
<b>TRIGger:AUTO:DWELl &lt;hour&gt;[,&lt;minute&gt;[,&lt;second&gt;[,&lt;point number&gt;]]]</b>
Set the soak time for test points in the automated test.
<b>TRIGger:AUTO:ORDer &lt;order&gt;</b>
Set the channel order during scan sweeps for the automated test.
<b>TRIGger:AUTO:POINts &lt;n&gt;</b>
Set the number of test points in the automated test.
<b>TRIGger:AUTO:REFerence {NONE   (@&lt;channel&gt;)}</b>
Set the primary reference channel for the automated test.
<b>TRIGger:AUTO:SPOint &lt;n&gt;[,&lt;point number&gt;]</b>
Set the set-point value for test points in the automated test.
<b>TRIGger:AUTO:SREFerence {NONE   (@&lt;channel&gt;)}</b>
Set the secondary reference channel for the automated test.
<b>TRIGger:AUTO:STABility &lt;n&gt;[,&lt; point number&gt;]</b>
Set the stability limit for test points in the automated test.
<b>TRIGger:AUTO:TOLerance &lt;n&gt;[,&lt;point number&gt;]</b>
Set the reference tolerance for test points in the automated test.
<b>TRIGger:COUNt {&lt;n&gt;   INFinity}</b>
Set the scan count.
<b>TRIGger:ENABle &lt;boolean&gt;</b>
Enable or suspend triggering and scanning.
<b>TRIGger:SOURce &lt;source&gt;</b>
Set the trigger source.
<b>TRIGger:TIMer {&lt;n&gt;   INFinity}</b>
Set the scan interval time.
<b>UNIT:TEMPerature {C   CEL   F   FAR}</b>
Set the temperature Unit, °C or °F.

## Remote Programming Examples

This section gives examples of sequences of commands for several likely scenarios.

### Single Channel, One-Shot

In this scenario, it is desired to measure one channel on command in a one-shot fashion. This example measures the temperature of a type K thermocouple using internal reference junction compensation. Below are two methods that can be used. In the first method, the channel is configured and read in one operation. The second method configures the channel and reads the channel in separate operations.

Command	Action
MEAS:TEMP? TC, K, (@102)	Configure channel 102 to measure the temperature of a type K thermocouple using internal RJC. Enable channel 102 only then set the trigger source to timer and the trigger count to 1. After configured, acquire the measurement and return the result.
MEAS:TEMP? TC, K, (@102)	Acquire another temperature measurement and return the result.

Command	Action
CONF:TEMP TC, K, (@102)	Configure channel 102 to measure the temperature of a type K thermocouple using internal RJC. Enable channel 102 only then set the trigger source to timer and the trigger count to 1.
READ?	Acquire one measurement then return the reading.
READ?	Acquire another measurement then return the reading.

### **One-Shot Scan**

In this scenario it is desired to measure scan several channels on command in a one-shot fashion. This example measures dc volts on some channels and temperature of a type K thermocouples on other channels.

Command	Action
*RST	Stop scanning and disable all channels and clear the scan list. Set the trigger source to timer and set the trigger count to 1. Set the sample rate to medium. Clear scan data in memory.
FUNC "VOLT:DC", (@101:104)	Configure channels 101 through 104 to measure dc volts.
TEMP:TC:TYPE K, (@105:108)	Configure channels 105 through 108 to measure temperature of a K thermocouple.
ROUT:SCAN (@101:108)	Enable the channels for scanning.
INIT	Initiate a scan.
STAT:OPER?	Query the operation status until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan.

### **Single Channel, Continuous**

In this scenario it is desired to measure one channel continuously for an indefinite length of time.

Command	Action
CONF:TEMP TC, K, (@102)	Configure and enable channel 102 to measure the temperature of a type K thermocouple using internal RJC. Set the scan list to channel 102 only, set the trigger source to timer, set the scan interval to 0. Set the sample rate to medium. Clear scan data in memory.
TRIG:COUN 0	Set the trigger count to infinity.
INIT	Initiate scanning.
FETC?	Return a measurement.
ABOR	Stop scanning.

### Continuous Scanning

In this scenario it is desired to scan several channel at full speed indefinitely. The example measures the temperature of 2.252 k $\Omega$  thermistors on several channels.

Command	Action
*RST	Stop scanning then disable all channels and clear the scan list. Set the trigger source to timer, set the scan interval to 0, and set the sample rate to medium. Clear scan data in memory.
TEMP:THER:TYPE R2K2, (@101,106:108)	Configure channels 101, 106, 107, and 108 to measure thermistor temperature.
ROUT:SCAN (@101,106:108)	Enable the channels for scanning.
TRIG:COUN 0	Set the trigger count to infinity.
INIT	Initiate scanning.
STAT:OPER?	Query the operation status until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
STAT:OPER?	Query the operation status until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
ABOR	Stop scanning.

### Number of Scan Sweeps

In this scenario it is desired to scan several channels a certain number of times. The example measures the temperature of SPRT on one channel and the resistance of four-wire PRTs on other channels. The program waits for all the scans to complete and then reads the data stored in memory.

Command	Action
*RST	Stop scanning and disable all channels. Clear the scan list. Set the trigger source to timer, set the scan interval to 0, and set the sample rate to medium. Clear scan data in memory.
TEMP:FRTD:TYPE SPRT, (@101)	Configure channel 101 to measure temperature of an SPRT.
TEMP:FRTD:SPRT:RTPW 25.60147, (@101)	Set the R(TPW) of the SPRT.
TEMP:FRTD:SPRT:COEF:LOW -2.0551897E-5, 9.8415366E-6, (@101)	Set the A4 and B4 characterization coefficients of the SPRT.
TEMP:FRTD:SPRT:COEF:HIG -7.700559E-5, -5.8801596E-6, (@101)	Set the A8 and B8 characterization coefficients of the SPRT.
TEMP:FRTD:TYPE A392, (@103:110)	Configure channels 103 through 110 to measure PRT sensors.
TEMP:FRTD:CALC:RES ON, (@103:110)	Set channels 103 through 110 to read resistance of the temperature sensors.
ROUT:SCAN (@101,103:110)	Enable the channels for scanning.
TRIG:COUN 20	Set the scan count to 20 scan sweeps.
INIT	Initiate scanning.
STAT:OPER?	Query the operation status until bit 8 (scan completed) is SET.
DATA:READ?	Return the readings of the first scan sweep; delete the readings from data memory.
DATA:READ?	Return the readings of the next scan sweep; delete the readings from data memory.

### Full-Speed Sampling

In this scenario it is desired to measure a single parameter with the fast sample rate. The program waits for all the sample to complete and then reads the data stored in memory.

Command	Action
*RST	Stop scanning and disable all channels then clear the scan list. Set the trigger source to timer, set the scan interval to 0, and set the sample rate to medium.
CONF:VOLT:DC (@101)	Configure and enable channel 101 to measure dc volts and set the scan list to channel 101 only.
RATE FAST	Set the sample rate to fast.
TRIG:COUN 500	Set the scan count to 500 samples.
INIT	Initiate scanning.
STAT:OPER?	Query the operation status until bit 8 (scan completed) is SET.
DATA:READ?	Return the first reading and delete the reading from data memory.

### Scanning with Scan Interval

In this scenario it is desired to scan several channels with a defined time interval between scan sweeps. The example measures the temperature of 10 k $\Omega$  thermistors on several channels.

Command	Action
*RST	Stop scanning and disable all channels then clear the scan list. Set the trigger source to timer and set the sample rate to medium. Clear scan data in memory.
TEMP:THER:TYPE R10K, (@101:104)	Configure channels 101 through 104 to measure thermistor temperature.
ROUT:SCAN (@101:104)	Enable the channels for scanning.
TRIG:TIM 60	Set the scan interval to 1 minute.
TRIG COUN 0	Set the scan count to infinity.
INIT	Initiate scanning.
STAT:OPER?	Query standard operation register until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
STAT:OPER?	Query standard operation register until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
ABOR	Stop scanning.

### External Trigger

In this scenario it is desired to control scanning with an external trigger. The example measures dc volts on several channels and dc current on two dedicated current channels. Each time the external trigger signal is asserted the instrument performs one scan sweep.

Command	Action
*RST	Stop scanning, disable all channels, and clear the scan list. Set the scan interval to 0 and set the sample rate to medium. Clear scan data in memory.
FUNC "VOLT:DC", (@101:105)	Configure channels 101 through 105 to measure dc volts.
FUNC "CURR:DC", (@121:122)	Configure channels 121 and 122 to measure dc current.
ROUT:SCAN (@101:105,121:122)	Enable the channels for scanning.
TRIG:SOUR EXT	Select the external trigger source.
TRIG:TIM INF	Perform one scan sweep for each trigger assertion.
TRIG:COUN 0	Set the trigger count to infinity.
INIT	Initiate scanning.
STAT:OPER?	Query standard operation register until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
STAT:OPER?	Query standard operation register until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
ABOR	Stop scanning.

## Monitoring

In this scenario it is desired to continuously monitor a channel when not performing scan sweeps. The example monitors 4-wire resistance on the front terminals and periodically scans dc volts on other channels as well as 4-wire resistance on the front channel.

Command	Action
*RST	Stop scanning, disable all channels, and clear the scan list. Set the scan interval to 0 and set the sample rate to medium. Clear scan data in memory.
FUNC "FRES", (@1)	Configure the front channel to measure 4-wire resistance.
FUNC "VOLT:DC", (@101:108)	Configure channels 101 through 108 to measure dc volts.
ROUT:SCAN (@1,101:108)	Enable the channels for scanning
TRIG:SOUR TIM	Select the timer trigger.
TRIG:TIM 60	Set the scan interval to 1 minute.
ROUT:MON (@1)	Select the front channel for monitor.
TRIG:COUN 0	Set the scan count to infinity.
ROUT:MON:STAT ON	Enable monitoring.
ROUT:MON:DATA?	Return a reading of the monitored channel.
INIT	Initiate scanning.
STAT:OPER?	Query standard operation register until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
ROUT:MON:DATA?	Return a reading of the monitored channel.
STAT:OPER?	Query standard operation register until bit 4 (scan sweep completed) is SET.
FETC?	Return the readings of the scan sweep.
ROUT:MON:STAT 0	Disable monitoring.
ABOR	Stop scanning.

### Alarm Trigger

In this scenario it is desired to continuously monitor temperature on a channel and trigger scan sweeps of several other channels when the readings of the monitored channel are above a limit. While the alarm is active scan sweeps are performed once every 10 seconds. Data is read later after all the data is collected.

Command	Action
*RST	Stop scanning, disable all channels, and clears the scan list. Set the sample rate to medium, clear scan data in memory, and disables all alarms.
TEMP:TC:TYPE K, (@101)	Configure channel 101 to measure the temperature of a type K thermocouple using internal RJC.
FUNC "VOLT:DC", (@103:106)	Configure and enable channels 103 through 106 to measure dc volts.
FUNC "FRES", (@109)	Configure channel 109 to measure 4-wire resistance.
ROUT:SCAN (@101,103:106,109)	Enable the channels for scanning.
CALC:LIM1:STAT HIGH, (@101)	Enable one alarm for channel 101 as a high limit.
CALC:LIM1 80, (@101)	Set the alarm limit
TRIG:SOUR ALAR	Select the alarm trigger source
TRIG:TIM 10	Set the scan interval to 10 seconds
TRIG:ALAR:CHAN (@101)	Set the Ch101 as the alarm trigger.
TRIG:COUN 10	Set the trigger count to 10 scan sweeps.
INIT	Initiate scanning and monitoring.
DATA:POIN?	Query the number of scan sweeps stored in memory.
ABOR	Stop scanning.
DATA:READ?	Get scan sweep data; delete the readings from memory.
DATA:READ?	Get scan sweep data; delete the readings from memory.

## Alphabetical List of Serial Commands

Each command description provides the structure (long and short format), a description of the command purpose, a command example, an example of what the command returns (as applicable to query commands), and notes specific to the command. The bullet-points below apply to each group of commands:

- Numeric data, specified by the mnemonic, <num>, uses ASCII characters to represent numbers. Numbers may contain a plus or minus ('+' or '-') sign, decimal point ('.'), and exponent ('E' or 'e') with its sign. If a fractional component is received when only an integer is required, the number is rounded to the nearest integer without any resulting error message.
- Unrecognized commands or commands with incorrect syntax or invalid parameters generate error messages in the error queue.
- Upper case letters designate syntax that is required when issuing the command. Lower case letters are optional and may be omitted.
- <> indicates a required parameter.
- [ ] indicates optional parameters.
- ( ) indicates a group of parameters that must be used together.
- '|' indicates alternate parameter values.
- <n> indicates a number is required.
- <channel> indicates a channel number is required. The channel number must be preceded by the '@' symbol and enclosed in parentheses, e.g. "(@101)".
- <ch\_list> indicates a list of channel numbers are required. The channel list can include a range of channels defined using the colon symbol. For example, "101:110" specifies the range of channels from 101 through 110 inclusive. The channel list can also include multiple channels or multiple ranges of channels separated by the comma symbol. The entire channel list must be preceded by the '@' symbol and enclosed in parentheses, e.g. (@1,101:110,201:204).
- <boolean> indicates a Boolean value (0 or 1) is required. The mnemonics OFF and ON are also accepted for 0 and 1, respectively.
- <unit> indicates max unit string is required (maximum 3 letters). The character string must be enclosed in quotation marks.
- <range> indicates a range value is required.
- <resolution> indicates resolution value is required.
- <gain> indicates a Mx+B gain value is required.
- <offset> indicates a Mx+B offset value is required.
- <slot> indicates a slot number is required.
- <name> indicates an character string is required. The character string must be enclosed in quotation marks.
- <year> indicates a four digit number is required.
- <month> indicates a one or two digit number is required.
- <day> indicates a one or two digit number is required.
- <hour> indicates a one or two digit number is required.
- <minute> indicates a one or two digit number is required.
- <second> indicates a one or two digit number is required.
- <baud> indicates a valid baud number is required.

**\*CLS**

**Description:** Clear status registers and the error queue. Status event registers are reset to 0. The registers affected are the Operation Status Event register, Questionable Status Event register, Alarm Status Event register, and the Event Status Register. The \*CLS command does not affect any of the associated condition or enable registers. It may indirectly affect the Status Byte Register. The error queue is also cleared of all logged errors.

**Example:** \*CLS

**Related Commands:**

\*ESR?  
SYSTem:ERRor?  
STATus:OPERation:EVEnt?  
STATus:QUEStionable:EVEnt?  
STATus:ALARm:EVEnt?

**\*ESE <n>**

**\*ESE?**

**Description:** Set the Event Status Enable mask that determines which bits of the Event Status Register are reported in the Event Status Summary bit (bit5) of the Status Byte Register. <n> is the sum of the decimal values of the bits of the Event Status Register that will be reported in the Event Status Summary bit of the Status Byte Register. The Status Byte register is updated to reflect any change in the enable registers associated with the summary bits. Event Status Enable is set to 0 at power-on if the \*PSC setting is 1. \*CLS and \*RST does not affect Event Status Enable.

**Example:** \*ESE?

**Response:** 255

**Set Example:** \*ESE 60

**Related Commands:**

\*ESR?  
\*STB?  
\*PSC

**\*ESR?**

**Description:** Query the Event Status Register and clear the register. The Event Status Register reports various instrument events or changes when they occur. The return value is the sum of the decimal values of the asserted bits of the register. The register bits, their decimal values (in parentheses), and their definitions are as follows:

- Bit 0 (1): Operation Complete
- Bit 1 (2): (Not used)
- Bit 2 (4): (Not used)
- Bit 3 (8): Device Dependent Error
- Bit 4 (16): Execution Error
- Bit 5 (32): Command Error
- Bit 6 (64): (Not used)
- Bit 7 (128): Power On

Bits in the Event Status Register may affect the Event Status Summary bit (bit 5) of the Status Byte Register depending on the bits that are set in the Event Status Enable register. \*CLS and \*RST does not affect Event Status Enable.

**Example:** \*ESR?

**Response:** 32

**Related Commands:**

\*ESE  
 \*STB?  
 \*CLS

**\*IDN?**

**Description:** Query the product identification. The response contains the name of the manufacturer, model number, serial number, firmware version, and OS build date of the product. The product information cannot be changed.

**Example:** \*IDN?

**Response:** FLUKE,1586A,12345678,1.00+1.00+20130618

**\*OPC**

**Description:** Set the Operation Complete bit of the Event Status Register when all pending command operations complete. All commands are sequential, so the Operation Complete bit is always set immediately when this command is received.

**Example:** \*OPC

**Related Commands:**

\*ESR?  
 STATus:OPERation:EVENT?  
 \*OPC  
 \*WAI

**\*OPC?**

**Description:** Return “1” when all pending command operations are complete. All commands are sequential, so this query always returns ‘1’ immediately.

**Example:** \*OPC?

**Response:** 1

**Related Commands:**

\*OPC  
 STATus:OPERation[:EVENT]?

**\*OPT?**

**Description:** Query the instrument module configuration. The response is a comma-separated list of the model numbers of the installed modules and status of input module for slots 1 and 2 respectively.

**Example:** \*OPT?

**Response:** 1586-2586,1,1586-2588,1

**Related Commands:**

\*IDN?

**\*PSC <boolean>**  
**\*PSC?**

**Description:** Enable or disable clearing of status enable registers at power on. <boolean> can be 0 (disabled), OFF, 1 (enabled), or ON. The default is 1.

If the \*PSC setting is on, all status enable registers will be set to 0 at power-on. If the setting is off, the status enable registers will be retained. The registers that are affected are the Event Status Enable register, Service Request Enable register, Operation Status Enable register, Alarm Status Enable register, and the Questionable Status Enable register. \*RST does not affect the \*PSC setting.

**Example:** \*PSC?  
**Response:** 1  
**Set Example:** \*PSC 1

**Related Commands:**

\*ESE  
\*SRE  
STATus:OPERation:ENABle  
STATus:ALARM:ENABle  
STATus:QUESTionable:ENABle

**\*RCL <n>**

**Description:** Recall a setup file. <n> specifies the storage location. The range is 1 to 99. If a setup file does not exist at the given storage location, the command fails and an error is logged, -222, "Data out of range".

*Note*

*\*RST does not affect setup file storage.*

**Example:** \*RCL 10

**Related Commands:**

\*SAV  
MEMory:STATe:CATalog:NUMB?  
MEMory:STATe:VALid?

**\*RST**

**Description:** Sets channel setup and scan settings to default states. The reset command performs the following actions:

- A scan or test in progress is cancelled.
- Scan data in memory is cleared.
- Measurement statistics are cleared.
- All general purpose measurement channels are configured for dc volts. Autoranging current channels are configured to measure dc current.
- Channel name is cleared for all channels.
- Mx+B scaling is disabled for all channels. Scale factor M is set to 1.0 and offset B is set to 0.0. The unit of Mx+B is cleared.
- All channels are disabled.
- Alarms are cleared and disabled for all channels.
- Alarm limits are set to 0.0.
- The alarm port association is set to none (0) for all channels.
- Math channels are configured for polynomial with coefficient  $c_1$  set to 1.0 and other coefficients set to 0.0 and the source channel set to channel 1.
- Sample rate is set to MEDium.
- Trigger source is set to TIMer.
- Trigger is enabled.
- Scan count is set to 1.
- Scan interval is set to 0.
- Monitor is disabled.
- Data recording is disabled, destination is set to internal.
- The temperature unit is set to C.
- The display is enabled.
- All setpoints of automated test are cleared, two reference channels are set to none, scan sequence is set to linear, and heat source control set to off.

Settings and memory not affected by \*RST include data file memory, setup file memory, probe definitions, language, remote interface settings, time and date, password configuration, and instrument calibration. Reset also does not directly affect status registers or the error queue, nor does it affect status enable registers.

**Example:** \*RST

**Related Commands:**

\*CLS

**\*SAV <n>**

**Description:** Save a setup file. <n> specifies the storage location. The range is 1 to 99. Every storage location also has a name associated with it. Use the command 'MEM:STAT:DEF' to assign a name to a setup file storage location. \*RST does not affect setup file storage.

**Example:** \*SAV 10

**Related Commands:**

\*RCL

MEMory:STATe:CATalog:NUMBer?

MEMory:STATe:DEFine

MEMory:STATe:DELeTe

**\*SRE <n>**  
**\*SRE?**

**Description:** Set the Service Request Enable for the Status Byte register. The Service Request Enable determines which bits of the Status Byte register are reported in the Master Summary Status bit of the Status Byte register. <n> is the sum of the decimal values of the bits of the Status Byte register that will be reported in the Master Status Summary bit.

The Status Byte register is updated to reflect any change in the enable registers associated with the summary bits. Service Request Enable is set to 0 at power-on if the \*PSC setting is 1. \*CLS and \*RST does not affect Event Status Enable.

**Example:** \*SRE?  
**Response:** 32  
**Set Example:** \*SRE 32

**Related Commands:**

\*STB?  
\*PSC

**\*STB?**

**Description:** Query the Status Byte register. The Status Byte register reports various instrument conditions. The return value is the sum of the decimal values of the asserted bits of the register. The register bits, their decimal values (in parentheses), and their definitions are as follows:

- Bit 0 (1): (Not used)
- Bit 1 (2): Alarm Status Summary
- Bit 2 (4): Error Queue Summary
- Bit 3 (8): Questionable Status Summary
- Bit 4 (16): (Not used)
- Bit 5 (32): Event Status Summary
- Bit 6 (64): Master Status Summary
- Bit 7 (128): Operation Status Summary

Summary bits are set when any bit in the associated event register is set and the corresponding bit in the associated enable register is also set. The Alarm Status Summary bit is associated with the Alarm Status Event register and the Alarm Status Enable register. The Questionable Status Summary bit is associated with the Questionable Status Event register and the Questionable Status Enable register. The Event Status Summary bit is associated with the Event Status Register and the Event Status Enable register. The Operation Status Summary bit is associated with the Operation Status Event register and the Operation Status Enable register. The Master Status Summary bit is set when any other bit in the Status Byte register is set while the corresponding bit in the Service Request Enable register is set. The Error Queue Summary bit is set if there are one or more errors in the error queue.

Reading the Status Byte register does not clear the register. The register always reports the associated status. \*CLS does not clear the Status Byte register but may indirectly affect it through the associated status registers and queues. \*RST does not affect the Status Byte register.

**Example:** \*STB?  
**Response:** 32

**Related Commands:**

\*ESR?

```
SYSTem:ERRor?  
STATus:OPERation:EVENT?  
STATus:ALARm:EVENT?  
STATus:QUEStionable:EVENT?  
*SRE
```

**\*TRG**

**Description:** Trigger a single scan sweep. This command is accepted only while the BUS trigger source is selected and the scan has been initiated. If any other trigger source is selected or if a scan sweep is already in progress, the command is ignored and an error is logged, -211, "Trigger ignored".

**Example:** \*TRG

**Related Commands:**

```
TRIGger:SOURce  
INITiate:IMMediate
```

**\*TST?**

**Description:** Perform a self-test and return the results. Check whether IG is active, and the files are OK.

**Example:** \*TST

**\*WAI**

**Description:** Wait until all pending command operations complete before executing further commands. All commands are sequential, so this command has no effect. It is provided for IEEE-488.2 compliance.

**Example:** \*WAI

**Related Commands:**

```
STATus:OPERation:EVENT?  
*OPC?
```

**ABORt**

**Description:** Abort a scan sweep in progress. If scanning is already stopped, the command will have no effect. \*RST also stops scanning.

**Example:** ABORt

**Related Commands:**

```
INITiate:IMMediate
```

**CALCulate:AVERage:AVERage? [(@<ch\_list>)]**

**Description:** Return the arithmetic mean for the given channels. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels. The statistics are calculated from all the valid readings of the channels since scanning was initiated or since the statistics were cleared. If there are no readings, the response is 9.910000E+37 and an error is logged (603,"Data not available"). \*RST clears the statistics for all channels.

**Example:**       CALC:AVER:AVER? (@101:105)  
**Response:**     1.000000e+00,1.000000e+00,1.000000e+00,  
                  1.000000e+00,1.000000e+00

**Related Commands:**  
CALCulate:AVERage:SDEviation?  
CALCulate:AVERage:CLEar  
INITiate:IMMEDIATE

**CALCulate:AVERage:CLEar [(@<ch\_list>)]**

**Description:** Clear the statistics for the given channels. (@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels. \*RST clears the statistics for all channels.

**Example:**       CALC:AVER:CLE (@101:105)

**Related Commands:**  
CALCulate:AVERage:AVERage?  
CALCulate:AVERage:CLEar:ALL

**CALCulate:AVERage:CLEar:ALL**

**Description:** Clear the statistics for all channels. \*RST clears the statistics for all channels.

**Example:**       CALC:AVER:CLE:ALL

**Related Commands:**  
CALCulate:AVERage:AVERage?  
CALCulate:AVERage:CLEar

**CALCulate:AVERage:COUNT? [(@<ch\_list>)]**

**Description:** Return the statistics sample size for the given channels. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels. The statistics are calculated from all the valid readings of the channels since scanning was initiated or since the statistics were cleared. \*RST clears the statistics for all channels, setting the count to 0.

**Example:**       CALC:AVER:COUN? (@101:105)  
**Response:**     10,10,10,8,8

**Related Commands:**  
CALCulate:AVERage:AVERage?  
CALCulate:AVERage:CLEar

**CALCulate:AVERage:MAXimum? [(@<ch\_list>)]**

**Description:** Return the maximum readings for the given channels. (@<ch\_list>) is a channel list containing channels, which may be a general purpose measurement channels, current channels, or math channels. The statistics are calculated from all the valid readings of the channels since scanning was initiated or since the statistics were cleared. If there are no readings, the response is 9.910000E+37 and an error is logged (603,"Data not available"). \*RST clears the statistics for all channels.

**Example:** CALC:AVER:MAX? (@101:105)

**Response:** 1.000000e+01,1.000000e+01,1.000000e+01,  
1.000000e+01,1.000000e+01

**Related Commands:**

CALCulate:AVERage:MINimum?

CALCulate:AVERage:AVERage?

CALCulate:AVERage:CLEar

**CALCulate:AVERage:MAXimum:TIME? [(@<ch\_list>)]**

**Description:** The query returns the time that the maximum reading was taken on the specified channels during the scan. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels. If there are no readings, the response is "0000,00,00,00,00,00.000" and an error is logged (603,"Data not available"). \*RST clears the statistics for all channels.

**Example:** CALC:AVER:MAX:TIME? (@101)

**Response:** 2013,05,03,09,17,30,000

**Related Commands:**

CALCulate:AVERage:MINimum:TIME?

**CALCulate:AVERage:MINimum? [(@<ch\_list>)]**

**Description:** Return the minimum readings for the given channels. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels. The statistics are calculated from all the valid readings of the channels since scanning was initiated or since the statistics were cleared. If there are no readings, the response is 9.910000E+37 and an error is logged (603,"Data not available"). \*RST clears the statistics for all channels.

**Example:** CALC:AVER:MIN (@101:105)

**Response:** 1.000000e+00,1.000000e+00,1.000000e+00,  
1.000000e+00,1.000000e+00

**Related Commands:**

CALCulate:AVERage:MAXimum?

CALCulate:AVERage:AVERage?

CALCulate:AVERage:CLEar

*CALCulate:AVERage:MINimum:TIME? [(@<ch\_list>)]*

**Description:** The query returns the time that the minimum reading was taken on the specified channels during the scan. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels. If there are no readings, the response is "0000,00,00,00,00,00.000" and an error is logged (603,"Data not available"). \*RST clears the statistics for all channels.

**Example:** CALC:AVER:MIN:TIME? (@101)

**Response:** 2013,05,03,09,17,30,000

**Related Commands:**

CALCulate:AVERage:MAXimum:TIME?

*CALCulate:AVERage:PTPeak? [(@<ch\_list>)]*

**Description:** Return the difference between the maximum and minimum for the given channels. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels. The statistics are calculated from all the valid readings of the channels since scanning was initiated or since the statistics were cleared. If there are no readings, the response is 9.910000E+37 and an error is logged (603,"Data not available"). \*RST clears the statistics for all channels.

**Example:** CALC:AVER:PTP? (@101:105)

**Response:** 2.000000e+01,2.000000e+01,2.000000e+01,  
2.000000e+01,2.000000e+01

**Related Commands:**

CALCulate:AVERage:SDEviation?

CALCulate:AVERage:AVERage?

CALCulate:AVERage:CLEar

*CALCulate:AVERage:RATE? [(@<ch\_list>)]*

**Description:** The query returns the rate of change of readings were taken on the specified channels during the scan. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels. If the data is not available, the response is 9.910000E+37 and an error is logged (603,"Data not available").

**Example:** CALC:AVER:RATE? (@101)

**Response:** 1.0000000e+00

**Related Commands:**

CALCulate:AVERage:RATE:BASE

CALCulate:AVERage:RATE:BASE?

*CALCulate:AVERage:RATE:BASE <time>[,(@<ch\_list>)]*

*CALCulate:AVERage:RATE:BASE? [(@<ch\_list>)]*

**Description:** Set the rate of change for the given channels. <time> is the key word SECond or MINute, (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels.

**Example:** CALC:AVER:RATE:BASE? (@101:103)

**Response:** SEC,SEC,SEC

**Set Example:** CALC:AVER:RATE:BASE SEC, (@101:103)

**Related Commands:**

CALCulate:AVERage:RATE? [(@<ch\_list>)]

**CALCulate:AVERage:SDEVIation? [(@<ch\_list>)]**

**Description:** Return the standard deviation for the given channels. (@<ch\_list>) is a channel list containing channels, which may be a general purpose measurement channels, current channels, or math channels. The statistics are calculated from all the valid readings of the channels since scanning was initiated or since the statistics were cleared. There must be at least two readings for the standard deviation calculation to be valid. If the data is not available, the response is 9.910000E+37 and an error is logged (603,"Data not available). \*RST clears the statistics for all channels.

**Example:** CALC:AVER:SDEV? (@101:105)  
**Response:** 1.000000e+00,1.000000e+00,1.000000e+00,  
 1.000000e+00,1.000000e+00

**Related Commands:**

CALCulate:AVERage:AVERage?  
 CALCulate:AVERage:PTPeak?  
 CALCulate:AVERage:CLEar

**CALCulate:LIMit{1|2} <n>[,(@<ch\_list>)]****CALCulate:LIMit{1|2}? [,@<ch\_list>]**

**Description:** Set the alarm limit for one of the alarm functions of the given channels. <n> is the alarm limit numeric value in floating-point format. It assumes the same unit of measurement as a channel's reading. The default is 0.0. (@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels. The query command returns the limit for only one channel. \*RST sets both alarm limits to 0.0 for all channels.

**Example:** CALC:LIM1? (@101:105)  
**Response:** 1.000000e+00,1.000000e+00,1.000000e+00,  
 1.000000e+00,1.000000e+00,  
**Set Example:** CALC:LIM1 1, (@101:105)

**Related Commands:**

CALCulate:LIMit1:STATe

**CALCulate:LIMit:CLEar [(@<ch\_list>)]**

**Description:** Clear the alarms for the given channels. (@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels. Associated alarm ports are updated to reflect the change in alarm conditions. The Status Alarm Condition register is also updated. \*RST clears the statistics for all channels.

**Example:** CALC:LIM:CLE (@101:105)

**Related Commands:**

CALCulate:LIMit:CLEar:ALL  
 CALCulate:LIMit:FAIL?  
 OUTPut:ALARm?  
 STATus:ALARm:CONDition?

**CALCulate:LIMit:CLEar:ALL**

**Description:** Clear all alarms. Associated alarm ports are updated to reflect the change in alarm conditions. The Status Alarm Condition register is also updated. \*RST clears the statistics for all channels.

**Example:**      CALC:LIM:CLE:ALL

**Related Commands:**

CALCulate:LIMit:CLEar  
CALCulate:LIMit:FAIL?  
OUTPut:ALARm?  
STATus:ALARm:CONDition?

**CALCulate:LIMit:FAIL? [(@<ch\_list>)]**

**Description:** Query the alarm condition of the given channels. (@<ch\_list>) is a channel list containing channels, which may be general purpose measurement channels, current channels, or math channels.

Possible responses are as follows:

Response	Description
0	Neither alarm tests positive
1	Only the first alarm tests positive
2	Only the second alarm tests positive
3	Both alarms test positive

The alarm condition is non-latching and will become 0 when a new scan changes the alarm test results or the alarms are cleared by command. \*RST clears the statistics for all channels.

**Example:**      CALC:LIM:FAIL? (@101:105)

**Response:**    0,1,2,3,1

**Related Commands:**

CALCulate:LIMit1  
CALCulate:LIMit1:STATe  
CALCulate:LIMit:CLEar  
OUTPut:ALARm?

*CALCulate:LIMit{1|2}:FEED {NONE | <n>[,(@<ch\_list>)]*  
*CALCulate:LIMit{1|2}:FEED? [(@<ch\_list>)]*

**Description:** Associate a channel alarm with an alarm port. <n> is the alarm port number ranging from 1 to 6. A value of NONE disables the association with any alarm port. The default is NONE. [(@<ch\_list>)] specifies the channels, which may be general purpose measurement channels, current channels, or math channels.

The alarm port association determines which alarm port is asserted when the channel alarm is asserted. An alarm port may have more than one channel alarm associated with it, in which case the port will be asserted when any one of the channel alarms are asserted. The query command returns the alarm port association for one channel. \*RST sets the alarm port association to NONE (disabled) for all channels.

**Example:** CALC:LIM1:FEED? (@101:105)  
**Response:** 1,1,1,1,1  
**Set Example:** CALC:LIM1:FEED 1, (@101:105)

**Related Commands:**

OUTPut:ALARm?  
 CALCulate:LIMit1:STATe

*CALCulate:LIMit{1|2}:STATe <state>[,(@<ch\_list>)]*  
*CALCulate:LIMit{1|2}:STATe? [(@<ch\_list>)]*

**Description:** Set the type of alarm function for one of the alarms associated with the given channels. <state> can be OFF, HIGH, or LOW. The default is OFF.

- OFF disables the alarm function for the channel.
- HIGH enables the alarm to trigger when a channel's reading is higher than the alarm limit.
- LOW enables the alarm to trigger when a channel's reading is lower than the alarm limit.

(@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels. The query command returns the state for only one channel. \*RST sets both alarm states for all channels to OFF.

**Example:** CALC:LIM1:STAT? (@101:105)  
**Response:** HIGH,HIGH,HIGH,HIGH,HIGH  
**Set Example:** CALC:LIM1:STAT HIGH, (@101:105)

**Related Commands:**

CALCulate:LIMit1  
 CALCulate:LIMit:FAIL?  
 OUTPut:ALARm?

*CALCulate:MATH:COEFFicients <c0>[,<c1>[,<c2>[,<c3>[,<c4>[,<c5>]]]]],(@<ch\_list>)*  
*CALCulate:MATH:COEFFicients? (@<ch\_list>)*

**Description:** Set the coefficients for the polynomial math function for the given math channels. (@<ch\_list>) specifies the math channels to configure. c0, c1, c2, c3, c4, and c5 are the zeroth through fifth order coefficients respectively. Omitted coefficients are set to 0, which effectively removes their associated terms from the polynomial function. The default values for c0, c1, c2, c3, c4, and c5 are 0, 1, 0, 0, 0, and 0 respectively. The query command returns the coefficients of only one math channel. \*RST sets c1 to 1 and all other coefficients to 0 for all math channels.

**Example:** CALC:MATH:COEF? (@101:102)

**Response:** 1.000000e+00,2.000000e+00,3.000000e+00,  
4.000000e+00,5.000000e+00,1.000000e+00,  
2.000000e+00,3.000000e+00,4.000000e+00,  
5.000000e+00

**Set Example:** CALC:MATH:COEF 1,2,3,4,5, (@101:102)

**Related Commands:**

CALCulate:MATH:FUNction

*CALCulate:MATH:EXPonent <exponent>,@<ch\_list>)*  
*CALCulate:MATH:EXPonent? (@<ch\_list>)*

**Description:** Set the exponent value for use with the power function. <exponent> is the numeric exponent value. The default is 1.0. (@<ch\_list>) specifies the math channels to configure. The query command returns the exponent value of the given math channels. \*RST sets the exponent to 1.0 for all math channels.

**Example:** CALC:MATH:EXP? (@501:502)

**Response:** 2.000000e+00,2.000000e+00

**Set Example:** CALC:MATH:EXP 2, (@501:502)

**Related Commands:**

CALCulate:MATH:FUNction

*CALCulate:MATH:FUNCTION <function>,@<ch\_list>*  
*CALCulate:MATH:FUNCTION? (@<ch\_list>)*

**Description:** Select the math function for the given math channels. <function> specifies the math function. The options are listed below. The default is POLY.

<function>	Formula	Display
POLYnomial	$c_5 \times A^5 + c_4 \times A^4 + c_3 \times A^3 + c_2 \times A^2 + c_1 \times A + c_0$	Polynomial
SROot	$\sqrt{A}$	Square Root
POWer	$A^{<exponent>}$	Power
EXPonential	$e^A$	Exponential
LOGarithm	$\log_{10}A$	Log10
ABSolute	A	A  or Absolute Value
RECiprocal	1 / A	1 / A
ADD	A + B	A + B
SUBTtract	A – B	A – B
MULTiPLY	A × B	A × B
DIVide	A / B	A / B
AVERage	Average of the latest readings of selected channels	Average
MAXimum	Maximum of the latest readings of selected channels	Maximum
MINimum	Minimum of the latest readings of selected channels	Minimum
SUM	Sum of the latest readings of selected channels	Sum

(@<ch\_list>) specifies the math channels to configure. A math channel is only calculated when it is included in the scan list and when a scan sweep is performed. Enabled math channels are calculated in order of channel number. A math channel can use a physical channel or another math channel for source A or B. Feeding one math channel into another allows complicated math expressions to be constructed. Mx+B scaling can also be applied to a math channel calculation and a unit of measurement can be defined. Mx+B scaling is performed after the math function is evaluated.

The query command returns the math function of the given math channels. \*RST sets the math function to POLY for all math channels.

**Example:** CALC:MATH:FUNC? (@501:502)

**Response:** POLY, POLY

**Set Example:** CALC:MATH:FUNC POLY, (@501:502)

**Related Commands:**

CALCulate:MATH:SOURce:ACHannel

CALCulate:MATH:SOURce:LIST

CALCulate:MATH:COEFFicients

CALCulate:MATH:EXPonent

CALCulate:SCALE:STATE

CALCulate:SCALE:UNIT

ROUTE:SCAN

*CALCulate:MATH:SOURce:ACHannel (@<channel\_A>),( @<ch\_list>)*

*CALCulate:MATH:SOURce:BCHannel (@<channel\_B>),( @<ch\_list>)*

*CALCulate:MATH:SOURce:ACHannel? (@<ch\_list>)*

*CALCulate:MATH:SOURce:BCHannel? (@<ch\_list>)*

**Description:** Set the A or B source channel for math calculations for the given math channels. The first parameter, (@<channel\_A>) or (@<channel\_B>), specifies the source channel, A or B respectively, to be used as the argument or arguments for a math function. The default is channel 1. (@<ch\_list>) specifies the math channels to configure.

Many math functions require the selection of a source channel, A, and some require two source channels, A and B. When a math function is calculated it uses the latest valid readings from the specified source channels. If no reading is available or if the latest reading from a source channel is invalid, the calculation result will be over-limit.

The query command returns the source channels of the given math channels. \*RST sets both source channels to channel 1 for all math channels.

**Example:** CALC:MATH:SOUR:ACH? (@501:502)

**Response:** 101,101

**Set Example:** CALC:MATH:SOUR:ACH (@101), (@501:502)

**Related Commands:**

CALCulate:MATH:FUNCTION

*CALCulate:MATH:SOURce:LIST (@<source\_list>),( @<ch\_list>)*

*CALCulate:MATH:SOURce:LIST? (@<ch\_list>)*

**Description:** Select the source channels for statistics calculations for math channels. (@<source\_list>) specifies the source channels, which may be general purpose measurement channels, current channels, or math channels. The maximum number of source channels is 10. The default is channel 1 only. (@<ch\_list>) specifies the math channels to configure.

The source channel list is used for channel statistics calculations, which include average, maximum, minimum, and sum. When a math function is calculated it uses the latest valid reading from each of the specified channels in the source channel list. If a reading is not available or if a reading from a source channel is invalid, the math channel result will be over-limit. The query command returns the source channel list of one math channel.

\*RST sets the source channel list to only channel 1 for all math channels.

**Example:** CALC:MATH:SOUR:LIST? (@501)

**Response:** 101,102,103,104

**Set Example:** CALC:MATH:SOUR:LIST (@101:104), (@501)

**Related Commands:**

CALCulate:MATH:FUNCTION

*CALCulate:SCALe:GAIN <n>,( @<ch\_list>)*

*CALCulate:SCALe:GAIN? (@<ch\_list>)*

**Description:** Set the  $Mx+B$  scaling gain (M) for the given channels. <n> is the gain value as a floating-point number. The default is 1.0. (@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels.

The query command returns the  $Mx+B$  scaling gain for the given channels. \*RST sets the  $Mx+B$  scaling gain to 1.0 for all channels.

**Example:** CALC:SCAL:GAIN? (@101:102)

**Response:** 1.500000e+00, 1.500000e+00

**Set Example:** CALC:SCAL:GAIN 1.5, (@101:102)

**Related Commands:**

CALCulate:SCALE:OFFSet

CALCulate:SCALE:UNIT

CALCulate:SCALE:STATe

*CALCulate:SCALE:OFFSet <n>,@<ch\_list>**CALCulate:SCALE:OFFSet? (@<ch\_list>)*

**Description:** Set the Mx+B scaling offset (B) for the given channels. <n> is the offset value as a floating-point number. The default is 0.0. (@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels. The query command returns the Mx+B scaling offset for the given channels. \*RST sets the Mx+B scaling offset to 0.0 for all channels.

**Example:** CALC:SCAL:OFFS? (@101:102)**Response:** 1.000000e-01,1.000000e-01**Set Example:** CALC:SCAL:OFFS 0.1, (@101:102)**Related Commands:**

CALCulate:SCALE:GAIN

CALCulate:SCALE:UNIT

CALCulate:SCALE:STATe

*CALCulate:SCALE:STATe <boolean>,@<ch\_list>**CALCulate:SCALE:STATe? (@<ch\_list>)*

**Description:** Enable or disable Mx+B scaling for the given channels. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 0. (@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels.

The query command returns the state for the given channels. \*RST disables Mx+B scaling for all channels.

**Example:** CALC:SCAL:STAT? (@101:102)**Response:** 1,1**Set Example:** CALC:SCAL:STAT ON, (@101:102)**Related Commands:**

CALCulate:SCALE:GAIN

CALCulate:SCALE:OFFSet

CALCulate:SCALE:UNIT

*CALCulate:SCALE:UNIT "<unit>",<@<ch\_list>*  
*CALCulate:SCALE:UNIT? (<@<ch\_list>*

**Description:** Set the unit of measurement for Mx+B scaling. "<unit>" specifies the unit. It is a string of from zero to three characters enclosed in quotation marks. The unit will be presented alongside the numeric value of the channel's reading on the display. The default is no unit (an empty string). (<@<ch\_list>) specifies the channels, which may be general purpose measurement channels, current channels, or math channels.

The query command returns the unit of one channel. \*RST clears the Mx+B unit of measurement for all channels.

**Example:** CALC:SCAL:UNIT? (@101:102)  
**Response:** "AAA", "AAA"  
**Set Example:** CALC:SCAL:UNIT "AAA", (@101:102)

**Related Commands:**

CALCulate:SCALE:GAIN  
CALCulate:SCALE:OFFSet  
CALCulate:SCALE:STATe

*CALibration:DATE?*

**Description:** Read the calibration date. The response is <year>,<month>,<day>. \*RST does not affect the calibration date.

**Example:** CAL:DATE?  
**Response:** 2013,1,1

*CALibrate:MODule:DATE? <slot>*

**Description:** Read the calibration date of a module. The response is <year>,<month>,<day>. \*RST does not affect the module calibrate date.

**Example:** CAL:MOD:DATE? 1  
**Response:** 2013,1,1

*CONFigure? [(<@<ch\_list>)]*

**Description:** Query the measurement function of a channel. (<@<ch\_list>) is a channel list containing the channels in the scan list. The response is the short-form name of the function within quotation marks. \*RST configures all physical channels other than special current channels to measure dc volts. Current channels are configured to measure dc current. The return will be "VOLT", "CURR", "FRES", "RES", "TEMP TC", "TEMP FRTD", "TEMP TRTD", "TEMP RTD", "TEMP THER", "TEMP FTH", "DIO", "TOT", or "MATH".

**Example:** CONF? (@101:102)  
**Response:** "VOLT", "VOLT"

**Related Commands:**

CONFigure:VOLTagE:DC  
CONFigure:CURREnt:DC  
CONFigure:RESistance  
CONFigure:TEMPerature  
SENSe:FUNCTion

**CONFigure:CURRent[:DC] [<range>,@<ch\_list>]**

**Description:** Configure channels to measure dc current. The scan list is cleared and then set to include the given channels. <range> is a number that selects the measurement range. The lowest range that covers the given <range> value, in amperes, will be used. If <range> is omitted it is set to automatic. (@<ch\_list>) specifies the channels. Only channel 1 on the front panel or either of the two special current channels in each installed channel module can be configured for current measurements. \*RST configures all physical channels other than special current channels to measure dc volts. Current channels are configured to measure dc current.

**Example:** CONF:CURR 0.1, 1e-5, (@121:122)

**Related Commands:**

CONFigure?  
SENSe:FUNctIon  
SENSe:CURRent:DC:RANGe

**CONFigure:DIGital:DATA:BYTE**

**Description:** Open DIO channel for scan.

**Example:** CONF:DIG:DATA:BYTE

**CONFigure:FRESistance [<range>,@<ch\_list>]**

**Description:** Configure channels to measure four-wire resistance. The scan list is cleared and then set to include the given channels. <range> is a number that selects the measurement range. The lowest range that covers the given <range> value, in ohms, will be used. If <range> is omitted it is set to automatic. Sensing current is also automatic, depending on the range (see specifications). (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two module channels are linked automatically when a module channel is configured for four-wire resistance measurement. \*RST configures all physical channels other than special current channels to measure dc volts. Current channels are configured to measure dc current.

**Example:** CONF:FRES 1000, 1e-3, (@101:102)

**Related Commands:**

CONFigure?  
SENSe:FUNctIon  
SENSe:FRESistance:RANGe

**CONFigure:RESistance [<range>,@<ch\_list>]**

**Description:** Configure channels to measure two-wire resistance. The scan list is cleared and then set to include the given channels. <range> is a number that selects the measurement range. The lowest range that covers the given <range> value, in ohms, will be used. If <range> is omitted it is set to automatic. Sensing current is also automatic, depending on the range (see specifications). (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. \*RST configures all physical channels other than special current channels to measure dc volts. Current channels are configured to measure dc current.

**Example:** CONF:RES 1e+03, 1e-03, (@101:102)

**Related Commands:**

CONFigure?  
SENSe:FUNctIon  
SENSe:RESistance:RANGe

**CONFigure:TEMPerature** <sensor>,<type>,(@<ch\_list>)

**Description:** Configure channels to measure temperature using a temperature sensor. The scan list is cleared and then set to include the given channels. <sensor> selects the sensor type. <type> selects the specific sensor type or characterization. The options for <sensor> and <type> are shown in the table below.

<sensor>	<type>	Description
TCouple	K	K thermocouple
	T	T thermocouple
	R	R thermocouple
	S	S thermocouple
	J	J thermocouple
	N	N thermocouple
	E	E thermocouple
	B	B thermocouple
	C	W5Re/W26Re thermocouple
	D	W3Re/W25Re thermocouple
	G	W/W26Re thermocouple
	L	Fe/Constantin thermocouple
	M	Ni18Mo/Ni0.8Co thermocouple
	U	Cu/Constantin thermocouple
W	W/W26Re thermocouple	
POLY	Polynomial characterized thermocouple	
RTD	A385	Two-wire PRT/RTD, alpha 0.00385
	A392	Two-wire PRT/RTD, alpha 0.00392
	ABC	Two-wire PRT/RTD, polynomial characterization
	SPRT	Two-wire PRT/RTD, ITS-90 characterization
FRTD	A385	Four-wire PRT/RTD, alpha 0.00385
	A392	Four-wire PRT/RTD, alpha 0.00392
	ABC	Four-wire PRT/RTD, polynomial characterization
	SPRT	Four-wire PRT/RTD, ITS-90 characterization
TRTD	A385	Three-wire PRT/RTD, alpha 0.00385
	A392	Three-wire PRT/RTD, alpha 0.00392
	ABC	Three-wire PRT/RTD, polynomial characterization
	SPRT	Three-wire PRT/RTD, ITS-90 characterization
THERmistor	R2K2	Two-wire thermistor, nominal 2252 $\Omega$ at 25 $^{\circ}\text{C}$
	R5K	Two-wire thermistor, nominal 5 k $\Omega$ at 25 $^{\circ}\text{C}$
	R10K	Two-wire thermistor, nominal 10 k $\Omega$ at 25 $^{\circ}\text{C}$
	RPOLY	Two-wire thermistor, R from T polynomial
FTHermistor	R2K2	Four-wire thermistor, nominal 2252 $\Omega$ at 25 $^{\circ}\text{C}$
	R5K	Four-wire thermistor, nominal 5 k $\Omega$ at 25 $^{\circ}\text{C}$
	R10K	Four-wire thermistor, nominal 10 k $\Omega$ at 25 $^{\circ}\text{C}$
	RPOLY	Four-wire thermistor, R from T polynomial

Characterization coefficients can be set, if necessary, using the SENSE:TEMPerature group of commands. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two module channels are linked automatically when a module channel is configured for four-wire or three-wire sensor measurement. \*RST configures all physical channels other than special current channels to measure dc volts. Current

channels are configured to measure dc current.

**Example:** CONF:TEMP TC, K, (@101:102)

**Related Commands:**

CONFigure?  
 SENSE:TEMPERature:TRANsducer  
 SENSE:TEMPERature:TCouple:TYPE  
 SENSE:TEMPERature:RTD:TYPE  
 SENSE:TEMPERature:THERmistor:TYPE

**CONFigure:VOLTage[:DC] [<range>,@<ch\_list>]**

**Description:** Configure channels to measure dc voltage. The scan list is cleared and then set to include the given channels. <range> is a number that selects the measurement range. The lowest range that covers the given <range> value, in volts, will be used. If <range> is omitted it is set to automatic. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. \*RST configures all physical channels other than special current channels to measure dc volts. Current channels are configured to measure dc current.

**Example:** CONF:VOLT 1,1e-05, (@101:102)

**Related Commands:**

CONFigure?  
 SENSE:VOLTage:DC:RANGE

**DATA:CLEAr**

**Description:** Delete all readings in scan memory. \*RST clears all readings from scan memory.

**Example:** DATA:CLE

**Related Commands:**

DATA:READ?

**DATA[:LAST]? [(@<channel>)]**

**Description:** Return the readings of the latest scan sweep or a specific channel. (@<channel>) is a channel list containing only one channel. Readings of the latest complete scan sweep, if available, are returned immediately without waiting for a new scan sweep in progress to complete. If the channel parameter is omitted, readings from the latest scan sweep are returned in a comma separated list of values for all channels in the scan list in order of increasing channel number.

If a channel number is given, the command returns the latest reading of only the given channel, if it is available. An invalid or out-of-range reading of a channel is represented by the value 9.900000e+37 or -9.900000e+37.

Scanning should be initiated and allowed to complete at least one scan sweep before attempting to read scan data. If no data is available, the response is 9.910000E+37 and an error code will be logged (603,"Data not available"). \*RST clears all readings in scan memory.

**Example:** DATA? (@101)

**Response:** 1.000000e-01

**Related Commands:**

INITiate  
 STATus:OPERation?  
 DATA:READ?  
 READ?

*DATA:LOG:AUTO <boolean>*  
*DATA:LOG:AUTO?*

**Description:** Enable or disable automatic data recording. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 0. If automatic data recording is enabled, data recording will automatically start when scanning is initiated. \*RST disables automatic data recording.

**Example:** DATA:LOG:AUTO?  
**Response:** 1  
**Set Example:** DATA:AUTO ON

**Related Commands:**  
DATA:LOG:DESTination  
DATA:LOG:ENABle  
DATA:LOG:RLIMit

*DATA:LOG:DESTination <memory>*  
*DATA:LOG:DESTination?*

**Description:** Select the memory device for data recording. <memory> can be MEMory or MMEMory. The default is MEM. MEM designates internal flash memory. MMEM designates external USB memory (mass memory). \*RST sets the data recording destination to internal flash (MEM).

**Example:** DATA:LOG:DEST?  
**Response:** MMEM  
**Set Example:** DATA:LOG:DEST MMEM

**Related Commands:**  
DATA:LOG:ENABle  
MEMory:LOG:READ?  
MMEMory:MOUNt?  
MMEMory:LOG:READ?

*DATA:LOG[:ENABle] <boolean>*  
*DATA:LOG[:ENABle]?*

**Description:** Enable or disable data recording to internal flash or USB memory. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 0. The memory device to which data is recorded is selected using the DATA:LOG:DESTination command.

If scan is not INITed, an error will be logged (434, "Cannot record while idle"). While data recording is enabled, all channel readings from the Scan or Measure function will be recorded to a data file in memory. The data file name is automatically generated based on the time and date. Monitor readings are not recorded. The amount of data exceeds the maximum number of rows, the data is stored in multiple files. \*RST disables data recording.

**Example:** DATA:LOG?  
**Response:** 1  
**Set Example:** DATA:LOG ON

**Related Commands:**  
DATA:LOG:DESTination  
DATA:LOG:RLIMit  
DATA:LOG:AUTO  
MEMory:LOG:FREE?  
MMEMory:MOUNt?

**DATA:POINTs?**

**Description:** Return the number of scan sweeps stored in memory. \*RST clears all readings in scan memory.

**Example:** DATA:POIN?

**Response:** 10000

**Related Commands:**

DATA:READ?

**DATA:READ?**

**Description:** Return the readings of the earliest scan sweep in scan memory and delete the readings. Readings from the scan sweep are returned in a comma separated list of values for all channels in the scan list. An invalid or out-of-range reading of a channel is represented by the value 9.900000e+37 or -9.900000e+37.

Scanning should be initiated and allowed to complete at least one scan sweep before attempting to read scan data. If no data is available, the response is 9.910000E+37 and an error code will be logged (603,"Data not available"). \*RST clears all readings in scan memory.

**Example:** DATA:READ?

**Response:** 1.000000e+00,1.000000e+00

**Related Commands:**

INITiate

STATus:OPERation?

DATA:POINTs?

DATA:LAST?

READ?

**DIAGnostic:DMM:RELAy:CYCLes?**

**Description:** Query the relay cycle count of DMM. The response is a set of ten numbers separated by commas. \*RST does not affect the relay cycle counts.

**Example:** DIAG:DMM:REL:CYCL?

**Response:** 2,0,2,2,6,2,2,28,2,2

**Related Commands:**

DIAGnostic:DMM:RELAy:CYCLes:CLEar

**DIAGnostic:CHANnel:RELAy:CYCLes? (@<ch\_list>)**

**Description:** Query the relay cycle count of the given channels. The response is the relay counts of the given channels separated by commas. \*RST does not affect the relay cycle counts.

**Example:** DIAG:CHAN:REL:CYCL? (@101:105)

**Response:** 50,100,75,84,70

**Related Commands:**

DIAGnostic:CHANnel:RELAy:CYCLes:CLEar

**DIAGnostic:CHANnel:RELAy:CYCLes:CLEar (@<ch\_list>)**

**Description:** Clear the relay counts of the given channels.

**Example:** DIAG:CHAN:REL:CYCL:CLE (@101:105)

**Related Commands:**

DIAGnostic:CHANnel:RELAy:CYCLes?

*DISPlay:BRIGhtness <brightness>*

*DISPlay:BRIGhtness?*

**Description:** Set display brightness level. <brightness> can be LOW, MEDium or HIGH.

**Example:** DISP:BRIG?

**Response:** MED

**Set Example:** DISP:BRIG HIGH

*DISPlay:DATE:FORMat <date\_format>*

*DISPlay:DATE:FORMat?*

**Description:** Set date display format. <date\_format> can be YMD (yyyy-mm-dd), DMY (dd/mm/yyyy) or MDY (mm-dd-yyyy).

**Example:** DISP:DATE:FORM?

**Response:** YMD

**Set Example:** DISP:DATE:FORM MDY

*DISPlay:DECimal:FORMat <decimal>*

*DISPlay:DECimal:FORMat?*

**Description:** Set the decimal format for display. <decimal> can be COMMA or PERiod.

**Example:** DISP:DEC:FORM?

**Response:** PER

**Set Example:** DISP:DEC:FORM COMM

*DISPlay:LANGUage <language>*

*DISPlay:LANGUage?*

**Description:** Set display language. <language> can be ENGLISH, CHINESE, FRENCH, GERMAN, PORTUTUESE, SPANISH, RUSSIAN, JAPANESE or KOREAN.

**Example:** DISP:LANG?

**Response:** ENGLISH

**Set Example:** DISP:LANG CHINESE

*DISPlay:SAVer <saver>*

*DISPlay:SAVer?*

**Description:** Set the time for screen saver. <saver> can be 0 (Never), 15, 30 or 60 minutes.

**Example:** DISP:SAV?

**Response:** 15

**Set Example:** DISP:SAV 0

*DISPlay:TIME:FORMat <time\_format>*

*DISPlay:TIME:FORMat?*

**Description:** Set time display format. <time\_format> can be 12 or 24.

**Example:** DISP:TIME:FORM?

**Response:** 24

**Set Example:** DISP:TIME:FORM 12

*DISPlay:STATe <boolean>*

*DISPlay:STATe?*

**Description:** Enable or disable the front panel display. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 1. \*RST enables the display.

**Example:** DISP:STAT?

**Response:** 1

**Set Example:** DISP:STAT ON

**Related Commands:**

CONFigure:VOLTage

SENSe:TEMPerature:TRANsducer

ROUte:SCAN

TRIG:SOURce

TRIG:TIMer

INITiate

ABORt

READ?

*FETCh?*

**Description:** Return readings of the latest measurement or scan sweep. Readings from the scan sweep are returned in a comma separated list of values for all channels in the scan list. An invalid or out-of-range reading of a channel is represented by the value 9.900000e+37 or -9.900000e+37.

If a scan sweep is in progress, the response will be held off until the scan sweep completes. If scanning is waiting for a trigger or has completed, and data is available, the latest readings will be returned immediately.

Scanning should be initiated and allowed to complete at least one scan sweep before attempting to read scan data. If no data is available, an error code will be logged as 603, "Data not available" in the error queue. \*RST clears all readings in scan memory.

**Example:** FETC?

**Response:** 1.000000e-01,1.000000e+01

**Related Commands:**

INITiate

STATus:OPERation?

DATA:READ?

READ?

*INITiate[:IMMediate]*

**Description:** Initiate scanning. The timing of scanning will be controlled by the selected trigger source and the trigger timer. Scanning will stop when the scan count is reached or scanning is aborted using the ABORt command. If a scanning is already active, the command is ignored and an error is logged (-213,“Init ignored”). Channels should be previously configured and enabled in the scan list. \*RST stops scanning.

If the trigger source is AUTO, the INITiate:IMMediate command starts an automated test.

**Example:** INIT

**Related Commands:**

CONFigure:VOLTagE  
SENSE:TEMPERature:TRANsducer  
ROUTe:SCAN  
TRIG:SOURce  
TRIG:TIMer  
ABORt  
READ?

*MEASure:CURRent[:DC]? [<range>,@<ch\_list>]*

**Description:** This command is equivalent to CONFigure:CURRent:DC followed by a READ?.

**Example:** MEAS:CURR 1e-04, 1e-09, (@121:122)

**Response:** 1.000000e-07, 1.000000e-06

**Related Commands:**

CONFigure:CURRent:DC  
READ?

*MEASure:DIGital:DATA:BYTE?*

**Description:** Acquire and return a reading of the digital input channel. The response is a decimal value between 0 and 255 which when converted to binary represents the condition of each of the eight bits of the digital I/O port. Each bit of the digital I/O port can be driven low by the data byte that is output to the port or an external device connected to the port.

**Example:** DIG:DATA:BYTE?

**Response:** 170

*MEASure:FRESistance? [<range>,@<ch\_list>]*

**Description:** This command is equivalent to CONFigure:FRESistance followed by a READ?.

**Example:** MEAS:FRES? 1e+03, 1e-1, (@101:102)

**Response:** 9.000000e+02, 8.000000e+02

**Related Commands:**

CONFigure:FRESistance  
READ?

*MEASure:RESistance? [<range>,@<ch\_list>]*

**Description:** This command is equivalent to CONFigure:RESistance followed by a READ?.

**Example:** MEAS:RES? 1e+03, 1e-1, (@101:102)

**Response:** 9.000000e+02, 8.000000e+02

**Related Commands:**

CONFigure:RESistance  
READ?

**MEASure:TEMPerature? <sensor>,<type>,@<ch\_list>**

**Description:** This command is equivalent to CONFigure:TEMPerature followed by a READ?.

**Example:** MEAS:TEMP? TC, K, (@101:102)

**Response:** 1.230000e+02,3.210000e+02

**Related Commands:**

CONFigure:TEMPerature  
READ?

**MEASure:TOTalize?**

**Description:** Acquire and return a reading of the digital totalize port.

**Example:** MEAS:TOT?

**Response:** 1.234500e+04

**MEASure:VOLTage[:DC]? [<range>,@<ch\_list>**

**Description:** This command is equivalent to CONFigure:VOLTage:DC followed by a READ?.

**Example:** MEAS:VOLT? (@101:102)

**Response:** 9.990000e-01, 1.000000e+00

**Related Commands:**

CONFigure:VOLTage:DC  
READ?

**MEMory:LOG:CLEAr**

**Description:** Erase all scan data files in internal memory. \*RST does not affect data file memory.

**Example:** MEM:LOG:CLE

**Related Commands:**

MEMory:LOG:DELeTe  
MEMory:LOG:NFILEs?

**MEMory:LOG:DELeTe ["<name>"]**

**Description:** Delete a data file in internal memory. <name> is the name of a stored data file folder. If the named file does not exist or there are no data files, no action is taken and an error is logged (901,"File not found"). If the <name> parameter is omitted, the latest data file is deleted. \*RST does not affect data file memory.

**Example:** MEM:LOG:DEL " 20130624\_143609333 "

**Related Commands:**

MEMory:LOG:NFILEs?  
MEMory:LOG:CLEAR

**MEMory:LOG:EXPort** [*"<name>"*]

**Description:** Copy a data file stored in internal memory to USB memory. <name> is the name of a stored data file folder. If the named file does not exist or there are no data files, no action is taken and an error is logged, 901,"File not found". If the <name> parameter is omitted, the latest data file is copied to USB memory. If USB memory is not available or is write protected, an error is logged (1603,"USB memory not ready").

**Example:** MEM:LOG:EXP " 20130624\_143609333 "

**Related Commands:**

MEMory:LOG:READ?

**MEMory:LOG:FREE?**

**Description:** Query the internal memory usage. The response gives the number of bytes not used followed by the number of bytes used, separated by a comma. \*RST does not affect data file memory.

**Example:** MEM:LOG:FREE?

**Response:** 19369984,815104

**Related Commands:**

MEMory:LOG:NFILES?

MEMory:LOG:DELeTe

**MEMory:LOG:NAME? <n>**

**Description:** Query name of specified data file folder in internal space.

**Example:** MEM:LOG:NAME? 10

**Response:** 20130520\_160808060

**Related Commands:**

MEM:LOG:NFILES?

**MEMory:LOG:NFILES?**

**Description:** Query the number of data folders stored in internal memory. \*RST does not affect data file memory.

**Example:** MEM:LOG:NFIL?

**Response:** 10

**Related Commands:**

MEMory:LOG:DELeTe

MEMory:LOG:FREE?

**MEMory:LOG:PROPerties?** ["<name>"]

**Description:** Query the properties of a data file stored in internal memory. <name> is the name of a stored data file folder. If the named file does not exist or there are no data files, and an error is logged (901,"File not found"). If the <name> parameter is omitted, the properties of the latest data file are returned.

The comma-separated response gives the size in bytes of the data file, the time and date when the file was created, and the user ID of the user who created the file. \*RST does not affect data file memory.

**Example:** MEM:LOG:PROP?

**Response:** 4589,1999-11-30 02:51:51,GUEST

**Related Commands:**

MEMory:LOG:STORe

MEMory:LOG:READ?

**MEMory:LOG:READ[:DATA]?** ["<name>"]

**Description:** Read a data file stored in internal memory. <name> is the name of a stored data file folder. If the named file does not exist or there are no data files, and an error is logged (901,"File not found"). If the <name> parameter is omitted, the latest data file is read.

The response contains the entire contents of the data file arranged in rows of data separated by semicolons. Each row consists of a series of data elements separated by commas. A data element may be a number or a text string. Text strings are enclosed in quotation marks. \*RST does not affect data file memory.

**Example:** MEM:LOG:READ?

**Response:** Record #,Time,Ch 101 (VDC),Ch 102 (VDC),Ch 103  
(VDC),Ch 104 (VDC),Ch 105 (VDC),Alarm  
Output,Signature 1,1999-11-30  
03:08:57.101,9.900000e+37,9.900000e+37,9.900000e  
+37,9.900000e+37,9.900000e+37,0,YENWY 2,1999-11-  
30 03:08:59.211,9.900000e+37,9.900000e+37,9.9000  
00e+37,9.900000e+37,9.900000e+37,0,BMDLH

**Related Commands:**

DATA:LOG:ENABLe

DATA:LOG:DESTination

MEMory:LOG:STORe

MEMory:LOG:EXPort?

**MEMory:LOG:READ:CONF?** ["<name>"]

**Description:** Read the configuration of data file stored in internal memory. <name> is the name of a stored data file folder. If the named file does not exist, an error is logged (901, "File not found"), If the <name> is omitted, the latest data file is read. \*RST does not affect file memory.

**Example:** MEM:LOG:READ:CONF?

**Response:** The entire contents of the configuration file.

**Related commands:**

MEMory:LOG:READ[:DATA]?

**MEMory:LOG:SECurity <boolean>**  
**MEMory:LOG:SECurity?**

**Description:** Enable or disable data file security. <boolean> can be 0 (disabled), OFF, 1 (enabled), or ON. The default is 0.

The data file security allows a user ID and password to be entered when scanning is started and the user ID recorded in the data file for the purpose of establishing data security and traceability. The data security feature is relevant only when the instrument is operated from the front panel. A user ID is not recorded in the data file when scanning is operated remotely. \*RST disables data file security.

**Example:** MEM:LOG:SEC?  
**Response:** 0  
**Set Example:** MEM:LOG:SEC OFF

**Related Commands:**  
INITiate[:IMMediate]  
DATA:LOG:ENABle  
MEMory:LOG:STORe  
MEMory:LOG:READ?

**MEMory:LOG:STORe**

**Description:** Store the latest scan data to internal memory. If no scan data is available, no action is taken and an error is logged (603 "Data not available"). If internal data memory has inadequate free space, no action is taken and an error is logged (908, "Error writing file").

If the amount of data exceeds the maximum number of rows, the data is stored in multiple files. The name of the file created using this command can be queried using the MEMory:LOG:CATalog? command. \*RST does not affect data file memory.

**Example:** MEM:LOG:STOR

**Related Commands:**  
MEMory:LOG:FREE?  
DATA:LOG:ENABle  
MEMory:LOG:PROPerties?  
MEMory:LOG:READ?

**MEMory:NSTates?**

**Description:** Query the maximum number of stored setup files, which is 99. \*RST does not affect setup file memory.

**Example:** MEM:NST?  
**Response:** 99

**Related Commands:**  
MEMory:STATe:CATalog:NUMBER?  
\*SAV

**MEMory:STATe:CLEar**

**Description:** Erase all setup files stored in memory. \*RST does not affect setup file memory.

**Example:** MEM:STAT:CLE

**Related Commands:**

MEMory:STATe:DELeTe

**MEMory:STATe:DEFine "<name>",<n>****MEMory:STATe:DEFine? "<name>"**

**Description:** Assign a name to a setup file storage location. <name> is the name enclosed in quotation marks. It can consist of letters and digits and certain other characters.

When a setup file is deleted, the name associated with the storage location is reset to the default name. The default name is the word "SETUP" followed by the number of the storage location.

The query command returns the storage location number that has the given name associated with it. If the name is not recognized, an error is logged (901,"File not found"). \*RST does not affect setup file memory.

**Example:** MEM:STAT:DEF? "FILE"

**Response:** 02

**Set Example:** MEM:STAT:DEF "FILE", 02

**Related Commands:**

MEMory:STATe:DEFine:NAME?

MEMory:STATe:CATaLog?

\*SAV

**MEMory:STATe:DEFine:NAME? <n>**

**Description:** Query the file name associated with a setup file storage location. The response is the name enclosed in quotation marks. \*RST does not affect setup file memory.

**Example:** MEM:STAT:DEF:NAME? 02

**Response:** "FILE"

**Related Commands:**

MEMory:STATe:DEFine

MEMory:STATe:VALid?

\*RCL

\*SAV

**MEMory:STATe:DELeTe <n>**

**Description:** Delete a setup file. <n> specifies the storage location. The range is 1 to 99. When a setup file is deleted, the name associated with the storage location is reset to the default name. \*RST does not affect setup file memory.

**Example:** MEM:STAT:DEL 02

**Related Commands:**

MEMory:STATe:VALid?

\*SAV

**MEMory:STATe:VALid? <n>**

**Description:** Query the existence of a setup file. <n> specifies the storage location. The range is 1 to 99. The response is '1' if a setup file is stored in the given storage location, and '0' if the storage location is empty. \*RST does not affect setup file memory.

**Example:** MEM:STAT:VAL? 02

**Response:** 1

**Related Commands:**

\*RCL

MEMory:STATe:DELeTe

**MMEMory:LOG:NAME? <n>**

**Description:** Query the specified data file name in USB memory.

**Example:** MMEM:LOG:NAME? 10

**Response:** 20130520\_160808060

**Related Commands**

MMEM:LOG:NFILEs?

**MMEMory:LOG:NFILEs?**

**Description:** Query the number of data files stored in USB memory.

**Example:** MMEM:LOG:NFIL?

**Response:** 101

**Related Commands:**

MMEMory:LOG:NAME?

**MMEMory:LOG:READ[:DATA]? [“<name>”]**

**Description:** Read a data file stored in USB memory. <name> is the name of a stored data file folder. If the named file does not exist or there are no data files and an error is logged (901, "File not found"). If the <name> parameter is omitted, the latest data file is read.

If USB memory is not available, an error is logged (1603, "USB memory not ready"). The response contains the entire contents of the data file arranged in rows of data separated by semicolons. Each row consists of a series of data elements separated by commas. A data element may be a number or a text string. Text strings are enclosed in quotation marks.

\*RST does not affect USB memory.

**Example:** MMEM:LOG:READ?

**Response:** 1.000000e+00,1.000000e-01,

**Related Commands:**

DATA:LOG:ENABle?

DATA:LOG:DESTination?

MMEMory:LOG:STORE

MMEMory:LOG:CATalog?

**MMEMory:LOG:READ:CONF? [<name>]**

**Definition:** Read the configuration of a data file stored in USB memory. <name> is the name of a stored configuration file. If the named file does not exist or there are no data files an error is logged (901,"File not found"). If the <name> parameter is omitted, the latest data file is read. The response contains the entire contents of the configuration file. \*RST does not affect USB memory.

**Example:** MMEM:LOG:READ:CONF?

**Response:** The entire contents of the configuration file.

**Related Commands:**

MMEMory:LOG:READ[:DATA]?

**MMEMory:LOG:STORe**

**Description:** Store the latest scan data to USB memory. If no scan data is available, no action is taken and an error is logged 603,"Data not available"). If USB memory is not available or is write protected and an error is logged (1603,"USB memory not ready").

If the amount of data exceeds the maximum number of rows, the data is stored in multiple associated files. The name of the file created using this command can be queried using the MMEMory:LOG:CATAlOG? command. \*RST does not affect USB memory.

**Example:** MMEM:LOG:STOR

**Related Commands:**

DATA:LOG:ENABle?

DATA:LOG:RLIMit

MMEMory:LOG:READ?

MMEMory:MOUNt?

**MMEMory:MOUNt?**

**Description:** Query the mount state of the USB memory device. The response is 1 if USB memory is available and 0 if USB memory is not detected or is unreadable. \*RST does not affect USB memory.

**Example:** MMEM:MOUN?

**Response:** 1

**Related Commands:**

DATA:LOG:ENABle?

DATA:LOG:DESTination?

MMEMory:LOG:STORe

MMEMory:LOG:READ?

### **OUTPut:ALARm?**

**Description:** Query the alarm output. The response is a decimal number which when converted to binary indicates the condition of each of the alarm output bits. The value of the alarm output is dependent on the alarm conditions of the channels assigned to each bit of the alarm output. The least significant bit is designated bit 1. \*RST clears all alarms.

**Example:**       OUTP:ALAR?

**Response:**     18

**Related Commands:**

CALCulate:LIMit1:FEED  
CALCulate:LIMit1:STATe  
CALCulate:LIMit:CLEar:ALL  
STATus:ALARm:CONDition?

### **READ?**

**Description:** Set the trigger source to TIMer, set the scan count to 1, initiate a scan, wait for the scan sweep to complete, and return the readings. Readings from the scan sweep are returned in a comma separated list of values for all channels in the scan list. An invalid or out-of-range reading of a channel is represented by the value +9.900000E+37 or -9.900000E+37.

There may be a long delay until the scan sweep completes and the response is returned. Alternate commands may be used instead which perform the same operations individually and avoid delay of the response.

Channels should be configured for measurement before using the READ? command.

**Example:**       READ?

**Response:**     1.000000e+00,1.000000e+00

**Related Commands:**

CONFigure:VOLTage:DC  
CONFigure:CURRent:DC  
CONFigure:RESistance  
CONFigure:TEMPerature  
SENSe:FUNctIon  
INITiate  
FETCh?

### **ROUTE:CHANnel:DElay <time>[,(@<channel\_list>)]**

### **ROUTE:CHANnel:DElay? [(@<channel\_list>)]**

**Description:** Set the Channel Delay between the closing of a channel relay and the sampling of the channel. <time> is the delay time in seconds. The range is 0.0 to 600.0 and the default is 0.0. (@<channel\_list>) specifies the channels, which must be general purpose measurement channels or current channels. If the channel list is omitted, all multi-function and current channels in the scan list are included.

Channel Delay allows extra settling time which can improve the measurement accuracy when source impedance is high or a PRT/RTD sensor requires extra time for self-heating to stabilize. The Channel Delay time is added to any automatic channel delay. The total measurement time for a channel is the minimum measurement time plus the Channel Delay time. \*RST sets Channel Delay for all physical channels to 0.0 (seconds).

**Related commands:**

SENSe:RATE

*ROUTE:CHANnel:NAME "<name>",(@<ch\_list>)*

*ROUTE:CHANnel:NAME? (@<ch\_list>)*

**Description:** Set the name associated with a channel. "<name>" is the name to assign to the channel. It must be a string of from zero to 16 characters enclosed in quotation marks. The default name is empty. (@<ch\_list>) specifies the channels, which may be physical channels or math channels or the totalizer channel. The query command returns the name of the given channels. \*RST sets the names of all channels to empty.

**Example:** ROUT:CHAN:NAME? (@101:102)

**Response:** "AAA", "AAA"

**Set Example:** ROUT:CHAN:NAME "AAA", (@101:102)

**Related Commands:**

SENSE:FUNCTION

ROUTE:CHANnel:STATE

*ROUTE:CHANnel:STATE <boolean>,@<ch\_list>)*

*ROUTE:CHANnel:STATE? (@<ch\_list>)*

**Description:** Enable or disable channels for measurement and scanning. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 0 for all channels. (@<ch\_list>) specifies the channels, which may be physical channels or math channels or the totalizer channel. When channels are enabled they are added to the scan list. When channels are disabled they are removed from the scan list. The query command accepts multiple channels in the parameter channel list and returns the states of these channels. \*RST disables all channels.

**Example:** ROUT:CHAN:STAT (@101:102)

**Response:** 1,1

**Set Example:** ROUT:CHAN:STAT ON, (@101:102)

**Related Commands:**

ROUTE:SCAN

INITiate

*ROUTE:MONitor (@<ch\_list>)*

*ROUTE:MONitor?*

**Description:** Select the monitor channel. (@<ch\_list>) is channel list containing only one channel number. When monitoring is enabled and scanning is active, the monitor channel will be measured and displayed continuously during the time between scan sweeps.

This command does not initiate scanning.

**Example:** ROUT:MON?

**Response:** 101

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

**Related Commands:**

INITiate

ROUTE:MONitor:DATA?

ROUTE:MONitor:STATE

***ROUTe:MONitor:DATA?***

**Description:** Query the latest reading of the monitor channel. At least one measurement must have been acquired from the monitor channel. If no monitor data is available, an error is logged (603,"Data not available"). \*RST disables monitor and clears monitor data.

**Example:**       ROUT:MON:DATA?

**Response:**     1.000000e+00

**Related Commands:**

ROUTe:MONitor

ROUTe:MONitor:STATe

INITiate

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

***ROUTe:MONitor:STATe?***

**Description:** Enable or disable channel monitoring during scanning. <boolean> can be 0 (off), OFF, 1 (on), or ON. When monitoring is enabled and scanning is initiated, a designated channel will be measured and displayed continuously during the time between scan sweeps. \*RST disables monitor.

**Example:**       ROUT:MON:STAT?

**Response:**     1

**Set Example:**  ROUT:MON:STAT ON

**Related Commands:**

ROUTe:MONitor

***ROUTe:SCAN (@<ch\_list>)***

***ROUTe:SCAN?***

**Description:** Enable channels for measurement and scanning. (@<ch\_list>) specifies the channels, which may be physical channels or math channels or the totalizer channel.

All channels not included in (@<ch\_list>) are disabled and removed from the scan list. An empty channel list, '@', disables all channels. The response to the query command has channel ranges expanded to enumerate each channel. Measurement/scanning must be idle, otherwise the command fails and an error is logged (527,"Operation not allowed while busy"). \*RST disables all channels.

**Example:**       ROUT:SCAN?

**Response:**     101,102,103

**Set Example:**  ROUT:SCAN (@101,103:105)

**Related Commands:**

INITiate

READ?

***ROUTe:SCAN:RESume <boolean>***

***ROUTe:SCAN:RESume?***

**Description:** Enable or disable automatic resume scan state at power on. <boolean> can be 0 (off), OFF, 1 (on) or ON. The default is 0. \*RST disables automatic scan resume.

**Example:**       ROUT:SCAN:RES?

**Response:**     0

**Set Example:**  ROUT:SCAN:RES 1

**[SENSe:]CURRent[:DC]:RANGe <range>[,(@<ch\_list>)]**  
**[SENSe:]CURRent[:DC]:RANGe? [(@<ch\_list>)]**

**Description:** Select the dc current range for channels. <range> is a number that selects the measurement range. The lowest range discrete that covers the given <range> value, in amperes, will be used. Resolution is automatic and depends on the range and sample rate. (@<ch\_list>) specifies the channels, which must be current channels.

The channel function must already be set to dc current. Otherwise an error is logged (403,"Conflict with channel configuration"). The query command returns the settings for the given channels. \*RST sets the range to automatic for all channels.

**Example:** CURR:RANG? (@121:122)  
**Response:** 1.000000e-01,1.000000e-01  
**Set Example:** CURR:RANG 1e-01,(@121:122)

**Related Commands:**

CONFigure:CURRent:DC  
 SENSe:FUNCTion  
 SENSe:CURRent:DC:RANGe:AUTO

**[SENSe:]CURRent[:DC]:RANGe:AUTO <boolean>[,(@<ch\_list>)]**  
**[SENSe:]CURRent[:DC]:RANGe:AUTO? [(@<ch\_list>)]**

**Description:** Enable or disable automatic dc current range for channels. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels, which must be current channels. The channel function must already be set to dc current. Otherwise an error is logged (403,"Conflict with channel configuration"). The query command returns the settings for the given channels. \*RST sets automatic range on for all channels.

**Example:** CURR:RANG:AUTO? (@121:122)  
**Response:** 1,1  
**Set Example:** CURR:RANG:AUTO ON,(@121:122)

**Related Commands:**

CONFigure:CURRent:DC  
 SENSe:FUNCTion  
 SENSe:CURRent:DC:RANGe

**[SENSe:]DIGital:DATA:BYTE?**

**Description:** Read the digital I/O port. The response is a decimal value between 0 and 255 which when converted to binary represents the condition of each of the eight bits of the digital I/O port. Each bit of the digital I/O port can be driven low by the data byte that is output to the port or an external device connected to the port.

**Example:** DIG:DATA:BYTE?  
**Response:** 170

**Related Commands:**

SOURce:DIGital:DATA:BYTE

**[SENSe:]FRESistance:RANGe <range>[,(@<ch\_list>)]**  
**[SENSe:]FRESistance:RANGe? [(@<ch\_list>)]**

**Description:** Select the four-wire resistance range for channels. <range> is a number that selects the measurement range. The lowest discrete range that covers the given <range> value, in ohms, will be used. Resolution is automatic and depends on the range and sample rate. Sensing current is also automatic, depending on the range. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for four-wire resistance measurement.

The channel function must already be set to four-wire resistance. Otherwise an error is logged (403, "Conflict with channel configuration"). The query command returns the settings for the given channels. \*RST sets the range to automatic for all channels.

**Example:** FRES:RANG? (@101:102)  
**Response:** 1.000000e+03,1.000000e+03  
**Set Example:** FRES:RANG 1e+3, (@101:102)

**Related Commands:**

CONFigure:FRESistance  
 SENSe:FUNCTion  
 SENSe:FRESistance:RANGe:AUTO

**[SENSe:]FRESistance:RANGe:AUTO <boolean>[,(@<ch\_list>)]**  
**[SENSe:]FRESistance:RANGe:AUTO? [(@<ch\_list>)]**

**Description:** Enable or disable automatic four-wire resistance range for channels. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for four-wire resistance measurement. The channel function must already be set to four-wire resistance. Otherwise an error is logged (403, "Conflict with channel configuration"). The query command returns the settings for the given channels. \*RST sets automatic range on for all channels.

**Example:** FRES:RANG:AUTO? (@101:102)  
**Response:** 1,1  
**Set Example:** FRES:RANG:AUTO ON, (@101:102)

**Related Commands:**

CONFigure:FRESistance  
 SENSe:FUNCTion  
 SENSe:FRESistance:RANGe

**[SENSe:]FUNCTion "<function>"[,(@<ch\_list>)]**  
**[SENSe:]FUNCTion? [(@<ch\_list>)]**

**Description:** Configure the measurement function for channels. "<function>" is the measurement function name. The name must be enclosed in quotation marks. The options for "<function>" include CURRent[:DC], FRESistance, RESistance, TEMPerature, and VOLTage[:DC]. If "<function>" is TEMPerature, the sensor type will be set to thermocouple and the thermocouple type will be set to K.

(@<ch\_list>) specifies the channels. Two module channels are linked automatically when a module channel is configured for four-wire resistance or four-wire or three-wire sensor measurement. Only channel 1 on the front panel or either of the two special current channels in each installed channel module can be configured for current measurements.

The channel scan list is not affected. More specific configuration can be done with additional SENSe commands. The query command returns the settings for the given channels. \*RST configures all physical channels other than special current channels to measure dc volts. Current channels are configured to measure dc current.

**Example:** FUNC? (@101:102)  
**Response:** "VOLT", "VOLT"  
**Set Example:** FUNC "VOLT", (@101:102)

**Related Commands:**

CONFigure:VOLTagE  
 CONFigure:RESistance  
 SENSE:TEMPerature:TRANsducer  
 ROUTe:SCAN

**[SENSe:]RATE <rate>**  
**[SENSe:]RATE?**

**Description:** Select the measurement sample rate. <rate> is the sample rate. Options are MEdium, SLOW, and FAST. The setting applies to all channels. \*RST sets the sample rate to MED.

**Example:** RATE?  
**Response:** MED  
**Set Example:** RATE SLOW

**[SENSe:]RESistance:RANGe <range>[(@<ch\_list>)]**  
**[SENSe:]RESistance:RANGe? [(@<ch\_list>)]**

**Description:** Select the two-wire resistance range for the given channels. <range> is a number that selects the measurement range. The lowest discrete range that covers the given <range> value, in ohms, will be used. Resolution is automatic and depends on the range and sample rate. Sensing current is also automatic, depending on the range (see specifications). (@<ch\_list>) specifies the channels, which must be general purpose measurement channels.

The channel function must already be set to two-wire resistance. Otherwise an error is logged (403,"Conflict with channel configuration"). The query command returns the settings for the given channels. \*RST sets the range to automatic for all channels.

**Example:** RES:RANG? (@101:102)  
**Response:** 1.000000e+03,1.000000e+03  
**Set Example:** RES:RANG 1e+03, (@101:102)

**Related Commands:**

CONFigure:FRESistance  
 SENSE:FUNCTion  
 SENSE:RESistance:RANGe:AUTO

**[SENSe:]RESistance:RANGe:AUTO <boolean>[(@<ch\_list>)]**  
**[SENSe:]RESistance:RANGe:AUTO? [(@<ch\_list>)]**

**Description:** Enable or disable automatic two-wire resistance range for the given channels. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels.

The channel function must already be set to two-wire resistance. Otherwise an error is logged (403,"Conflict with channel configuration"). The query command returns the settings for the given channels. \*RST sets automatic range on for all channels.

**Example:** RES:RANG:AUTO? (@101:102)  
**Response:** 1,1  
**Set Example:** RES:RANG:AUTO ON, (@101:102)

**Related Commands:**

CONFigure:FRESistance  
 SENSE:FUNCTion  
 SENSE:RESistance:RANGe

**[SENSe:]TEMPerature:CALCulate? <n>[,<RJT>],(@<channel>)**

**Description:** Calculate temperature for a given set of input values using the configuration of the given channel. This command can be used to test the channel configuration, the sensor characterization, and the accuracy of the temperature calculation.

<n> is the input value. If the channel is configured for a thermocouple, the value is volts. If the channel is configured for any other type of temperature sensor, the value is resistance in ohms. <RJT> is the reference junction temperature in degrees Celsius or Fahrenheit depending on the configured system temperature unit. If the parameter is omitted, it is assumed to be 0. (@<channel>) specifies the channel, which must be general purpose measurement channel.

The channel function must already be set to temperature. Otherwise an error is logged (403,"Conflict with channel configuration").

**Example:** temp:calc? 1e-3,25, (@101)  
**Response:** 4.944627e+01

**Related Commands:**

CONFigure:TEMPerature  
SENSe:TEMPerature:TRANsducer

**[SENSe:]TEMPerature:FRTD:A385:RZERo <n>,(@<channel>)**

**[SENSe:]TEMPerature:FRTD:A385:RZERo? (@<channel>)**

**Description:** Set the resistance at 0 °C of a four-wire PRT/RTD of characterization type A385. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel >) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to FRTD, and the characterization type to A385. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:FRTD:A385:RZER? (@101)  
**Response:** 1.000000e+02  
**Set Example:** TEMP:FRTD:A385:RZER 100.045, (@101)

**Related Commands:**

SENSe:TEMPerature:FRTD:TYPE

**[SENSe:]TEMPerature:FRTD:A392:RZERo <n>,@<channel>**

**[SENSe:]TEMPerature:FRTD:A392:RZERo? (@<channel>)**

**Description:** Set the resistance at 0 °C of a four-wire PRT/RTD of characterization type A392. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to FRTD, and the characterization type to A392. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:FRTD:A392:RZER? (@101)

**Response:** 1.000000e+02

**Set Example:** TEMP:FRTD:A392:RZER 99.9558, (@101)

**Related Commands:**

SENSe:TEMPerature:FRTD:TYPE

**[SENSe:]TEMPerature:FRTD:ABC:COEFFicients <a>,<b>,<c>,@<channel>**

**[SENSe:]TEMPerature:FRTD:ABC:COEFFicients? (@<channel>)**

**Description:** Set the polynomial characterization coefficients for a four-wire PRT/RTD of characterization type ABC. <a>, <b>, and <c> are the values of the A, B, and C polynomial characterization coefficients of the PRT/RTD. (@<channel>) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to FRTD, and the characterization type to ABC. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the A, B, and C coefficients to 3.9083E-3, -5.775E-7, and -4.183E-12 respectively for all channels.

**Example:** TEMP:FRTD:ABC:COEF? (@101)

**Response:** 3.9083E-3, -5.775E-7, -4.183E-12

**Set Example:** TEMP:FRTD:ABC:COEF 3.9083E-3, -5.775E-7, -4.183E-12, (@101)

**Related Commands:**

SENSe:TEMPerature:FRTD:TYPE

SENSe:TEMPerature:FRTD:ABC:RZERo

**[SENSe:]TEMPerature:FRTD:ABC:RZERo <n>,@<channel>**

**[SENSe:]TEMPerature:FRTD:ABC:RZERo? (@<channel>)**

**Description:** Set the resistance at 0 °C for a four-wire PRT/RTD of characterization type ABC. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to FRTD, and the characterization type to ABC. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:FRTD:ABC:RZER? (@101)

**Response:** 1.000000e+02

**Set Example:** TEMP:FRTD:ABC:RZER 100.0221, (@101)

**Related Commands:**

SENSe:TEMPerature:FRTD:TYPE

SENSe:TEMPerature:FRTD:ABC:COEFFicients

*[SENSe:]TEMPerature:FRTD:CALCulate:RESistance <boolean>,@<ch\_list>*  
*[SENSe:]TEMPerature:FRTD:CALCulate:RESistance? (@<ch\_list>)*

**Description:** Enable or disable resistance calculation of four-wire PRT/RTD for the given channels. When the resistance calculation is on, readings of the PRT/RTD temperature will be resistance in ohms rather than calculated temperature. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature and the sensor type set to FRTD. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST disables resistance calculation for all channels.

**Example:** TEMP:FRTD:CALC:RES? (@101:102)  
**Response:** 1,1  
**Set Example:** TEMP:FRTD:CALC:RES ON, (@101:102)

**Related Commands:**  
CONFigure:TEMPerature  
SENSe:FUNCTION

*[SENSe:]TEMPerature:FRTD:SPRT:COEFFicients:HIGH <a>,<b>,<c>,<d>]]],(@<channel >)*  
*[SENSe:]TEMPerature:FRTD:SPRT:COEFFicients:HIGH? (@<channel>)*

**Description:** Set the polynomial characterization coefficients for a four-wire PRT/RTD of characterization type ABC. <a>, <b>, and <c> are the values of the A, B, and C polynomial characterization coefficients of the PRT/RTD. (@<channel>) specifies the channel, which must be general purpose measurement channel. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to FRTD, and the characterization type to ABC. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:FRTD:SPRT:COEF:HIGH? (@101)  
**Response:** 0.000000e+00,0.000000e+00,0.000000e+00,  
0.000000e+00  
**Set Example:** TEMP:FRTD:SPRT:COEF:HIGH 1.2554E-4,  
-2.504E-5,0.0,0.0,(@101)

**Related Commands:**  
SENSe:TEMPerature:FRTD:TYPE  
SENSe:TEMPerature:FRTD:RZERO

**[SENSe:]TEMPerature:FRTD:SPRT:COEFFicients:LOW <a>[,<b>],(@<channel>)**  
**[SENSe:]TEMPerature:FRTD:SPRT:COEFFicients:LOW? (@<channel>)**

**Description:** Set the deviation polynomial coefficients for the range above 0 °C for a four-wire PRT/RTD of characterization type SPRT. <a>, <b>, <c>, and <d> are the values of the A, B, C, and D deviation polynomial characterization coefficients. Coefficients that are omitted are set to 0. (@<channel>) specifies the channel, which must be general purpose measurement channel. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to FRTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:FRTD:SPRT:COEF:LOW? (@101)  
**Response:** 0.000000e+00,0.000000e+00  
**Set Example:** TEMP:FRTD:SPRT:COEF:LOW -9.554105E-5,  
 2.2208E-6, (@101)

**Related Commands:**

SENSe:TEMPerature:FRTD:TYPE  
 SENSe:TEMPerature:FRTD:SPRT:RTPW  
 SENSe:TEMPerature:FRTD:SPRT:COEFFicients:HIGH

**[SENSe:]TEMPerature:FRTD:SPRT:RTPW <n>,@<channel>)**  
**[SENSe:]TEMPerature:FRTD:SPRT:RTPW? (@<channel>)**

**Description:** Set the characterization resistance at 0.01 °C for a four-wire PRT/RTD of characterization type SPRT. <n> is the resistance, in ohms, of the PRT/RTD at 0.01 °C. (@<channel>) specifies the channel, which must be general purpose measurement channel. Two channels are paired for four-wire sensor measurement. The channel function must already be set to temperature, the sensor type to FRTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the characterization resistance to 100.0 for all channels.

**Example:** TEMPerature:FRTD:SPRT:RTPW? (@101)  
**Response:** 1.000000e+02  
**Set Example:** TEMPerature:FRTD:SPRT:RTPW 25.48803, (@101)

**Related Commands:**

SENSe:TEMPerature:FRTD:TYPE  
 SENSe:TEMPerature:FRTD:COEFFicients:HIGH  
 SENSe:TEMPerature:FRTD:COEFFicients:LOW

**[SENSe:]TEMPerature:FRTD:TYPE <type>,@<ch\_list>)**  
**[SENSe:]TEMPerature:FRTD:TYPE? (@<ch\_list>)**

**Description:** Set the channel function to temperature, set the sensor type to four-wire PRT/RTD, and set the specific PRT/RTD type. <type> selects the specific PRT/RTD type. The options are listed in the following table.

<type>	Description
A385	Alpha 0.00385 standard PRT/RTD (IEC 60751)
A392	Alpha 0.00392 standard PRT/RTD
ABC	Polynomial characterization
SPRT	ITS-90 characterization

(@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for four-wire sensor measurement.

Characterization coefficients associated with the PRT/RTD type are set to default values. The default value for RZERO is 100.0  $\Omega$ . The default value for RTPW is 100.0. The default value for SPRT coefficients are 0. The characterization coefficients can be changed using the SENSE:TEMPERature:FRTD group of commands. The channel scan list is not affected by this command. \*RST sets the PRT/RTD type to A385 for all channels.

**Example:**       TEMPERature:FRTD:TYPE? (@101:102)  
**Response:**       A385,A385  
**Set Example:**   TEMP:FRTD:TYPE SPRT, (@101:102)

**Related Commands:**

SENSE:TEMPERature:FRTD:RZERO  
ROUTE:SCAN  
SENSE:TEMPERature:TRANSDUCER  
CONFIGure:TEMPERature

*[SENSE:]TEMPERature:FTHermistor:CALCulate:RESistance <boolean>,@<ch\_list>  
[SENSE:]TEMPERature:FTHermistor:CALCulate:RESistance? (@<ch\_list>)*

**Description:** Enable or disable resistance calculation of four-wire thermistor for the given channels. When the resistance calculation is on, readings of the thermistor temperature will be resistance in ohms rather than calculated temperature. <boolean> can be 0 (off), OFF, 1 (on), or ON.

(@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for four-wire sensor measurement. The channel function must already be set to temperature and the sensor type set to FTHermistor. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST disables the resistance calculation for all channels.

**Example:**       TEMP:FTH:CALC:RES? (@101:102)  
**Response:**       1,1  
**Set Example:**   TEMP:FTH:CALC:RES ON, (@101:102)

**Related Commands:**

CONFIGure:TEMPERature  
SENSE:FUNCTION

*[SENSe:]TEMPerature:FTHermistor:RPOLy:COEFFicients <c0>,<c1>,<c2>,<c3>,@<channel>*  
*[SENSe:]TEMPerature:FTHermistor:RPOLy:COEFFicients? (@<channel>)*

**Description:** Set the characterization coefficients for a four-wire thermistor of type RPOLY. The characterization defines the thermistor resistance as a function of temperature. <c0>, <c1>, <c2>, and <c3> are the polynomial coefficients of the thermistor characterization. (@<channel>) specifies the channel, which must be general purpose measurement channel. Two channels are paired for four-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to thermistor, and the thermistor type to RPOLY. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:FTH:RPOL:COEF? (@101)  
**Response:** 0.000000e+00,0.000000e+00,0.000000e+00,  
0.000000e+00  
**Set Example:** TEMP:FTH:RPOL:COEF 4.904,3.9951E3,0.0,  
-1.667E5, (@101)

**Related Commands:**

SENSe:TEMPerature:THERmistor:TYPE

*[SENSe:]TEMPerature:FTHermistor:TYPE <type>,@<ch\_list>*  
*[SENSe:]TEMPerature:FTHermistor:TYPE? (@<ch\_list>)*

**Description:** Set the channel function to temperature, set the sensor type to four-wire thermistor, and set the specific thermistor type. <type> selects the specific thermistor type. The options are listed in the following table.

<type>	Description
R2K2	Nominal 2252 $\Omega$ at 25 $^{\circ}\text{C}$
R5K	Nominal 5 k $\Omega$ at 25 $^{\circ}\text{C}$
R10K	Nominal 10 k $\Omega$ at 25 $^{\circ}\text{C}$
RPOLy	Polynomial characterization, resistance versus temperature

(@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for four-wire sensor measurement. The channel scan list is not affected. \*RST sets the thermistor type to R10K for all channels.

**Example:** TEMP:FTH:TYPE? (@101:102)  
**Response:** R2K2,R2K2  
**Set Example:** TEMP:FTH:TYPE R2K2, (@101:102)

**Related Commands:**

SENSe:TEMPerature:FTHermistor:COEFFicients  
ROUTE:SCAN  
SENSe:TEMPerature:TRANsducer  
CONFigure:TEMPerature

**[SENSe:]TEMPerature:RJUNction? (@<ch\_list>)**

**Description:** Query the internal reference junction compensation for the channel. The channel must be configured to measure a thermocouple. The response is a comma separated list of temperatures for each of the given channels. The values are temperatures sensed by the reference junction temperature sensors in degrees Celsius or Fahrenheit depending on the configured system temperature unit. The channel function must already be set to temperature and the sensor type set to thermocouple. Otherwise an error is logged, 403, "Conflict with channel configuration".

**Example:** TEMP:RJUN? (@101:102)  
**Response:** 2.500000e+01,2.500000e+01

**Related Commands:**

SENSe:TEMPerature:TCouple:RJUNction:TYPE  
UNIT:TEMPerature

**[SENSe:]TEMPerature:RTD:A385:RZERo <n>,@<channel>**

**[SENSe:]TEMPerature:RTD:A385:RZERo? (@<channel>)**

**Description:** Set the resistance at 0 °C of a two-wire PRT/RTD of characterization type A385. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to RTD, and the characterization type to A385. Otherwise an error is logged (403, "Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:RTD:A385:RZER? (@101)  
**Response:** 1.000000E+02  
**Set Example:** TEMP:RTD:A385:RZER 100.104, (@101)

**Related Commands:**

SENSe:TEMPerature:RTD:TYPE

**[SENSe:]TEMPerature:RTD:A392:RZERo <n>,@<channel>**

**[SENSe:]TEMPerature:RTD:A392:RZERo? (@<channel>)**

**Description:** Set the resistance at 0 °C of a two-wire PRT/RTD of characterization type A392. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to RTD, and the characterization type to A392. Otherwise an error is logged (403, "Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:RTD:A392:RZER? (@101)  
**Response:** 1.000000E+02  
**Set Example:** TEMP:RTD:A392:RZER 99.9895, (@101)

**Related Commands:**

SENSe:TEMPerature:RTD:TYPE

**[SENSe:]TEMPerature:RTD:ABC:COEFFicients <a>,<b>,<c>,@<channel>**

**[SENSe:]TEMPerature:RTD:ABC:COEFFicients? (@<channel>)**

**Description:** Set the polynomial characterization coefficients for a two-wire PRT/RTD of characterization type ABC. <a>, <b>, and <c> are the values of the A, B, and C polynomial characterization coefficients of the PRT/RTD. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to RTD, and the characterization type to ABC. Otherwise an error is logged (403, "Conflict with channel configuration"). \*RST sets the A, B, and C coefficients to 3.9083E-3, -5.775E-7, and -

4.183E-12 respectively for all channels.

**Example:** TEMP:RTD:ABC:COEF? (@101)  
**Response:** 3.908300E-03,-5.775000E-07,-4.183000E-12.  
**Set Example:** TEMP:RTD:ABC:COEF 3.9084e-3,-5.775e-7,4.183e-12, (@101)

**Related Commands:**

SENSe:TEMPerature:RTD:TYPE  
 SENSe:TEMPerature:RTD:ABC:RZERo

*[SENSe:]TEMPerature:RTD:ABC:RZERo <n>,@<channel>*

*[SENSe:]TEMPerature:RTD:ABC:RZERo? (@<channel>)*

**Description:** Set the resistance at 0 °C for a two-wire PRT/RTD of characterization type ABC. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to RTD, and the characterization type to ABC. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the characterization resistance to 100.0 for all channels.

**Example:** TEMP:RTD:ABC:RZER? (@101)  
**Response:** +1.000000E+02  
**Set Example:** TEMP:RTD:ABC:RZER 100, (@101)

**Related Commands:**

SENSe:TEMPerature:RTD:TYPE  
 SENSe:TEMPerature:RTD:ABC:COEFFicients

*[SENSe:]TEMPerature:RTD:CALCulate:RESistance <boolean>,@<ch\_list>*

*[SENSe:]TEMPerature:RTD:CALCulate:RESistance? (@<ch\_list>)*

**Description:** Enable or disable resistance calculation of two-wire PRT/RTD for the given channels. When the resistance calculation is on, readings of the PRT/RTD temperature will be resistance in ohms rather than calculated temperature. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels to configure. Channels in the list must be physical channels.

The channel function must already be set to temperature and the sensor type set to RTD. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST disables resistance calculation for all channels.

**Example:** TEMP:RTD:CALC:RES? (@101)  
**Response:** 0  
**Set Example:** TEMP:RTD:CAL:RES 1, (@101)

**Related Commands:**

CONFigure:TEMPerature  
 SENSe:FUNction

*[SENSe:]TEMPerature:RTD:SPRT:COEFFicients:HIGH <a>,<b>,<c>,<d>]]],(@<channel>)*  
*[SENSe:]TEMPerature:RTD:SPRT:COEFFicients:HIGH? (@<channel>)*

**Description:** Set the deviation polynomial coefficients for the range above 0 °C for a two-wire PRT/RTD of characterization type SPRT. <a>, <b>, <c>, and <d> are the values of the A, B, C, and D deviation polynomial coefficients. Coefficients that are omitted are set to 0. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to RTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:RTD:SPRT:COEF:HIGH? (@101)  
**Response:** 0.000000E+00,0.000000E+00,0.000000E+00,  
0.000000E+00  
**Set Example:** TEMP:RTD:SPRT:COEF:HIGH 0,0,0,0,(@101)

**Related Commands:**

SENSe:TEMPerature:RTD:TYPE  
SENSe:TEMPerature:RTD:SPRT:RTPW  
SENSe:TEMPerature:RTD:SPRT:COEFFicients:LOW

*[SENSe:]TEMPerature:RTD:SPRT:COEFFicients:LOW <a>,<b>],(@<channel>)*  
*[SENSe:]TEMPerature:RTD:SPRT:COEFFicients:LOW? (@<channel>)*

**Description:** Set the deviation polynomial coefficients for the range below 0 °C for a two-wire PRT/RTD of characterization type SPRT. <a> and <b> are the values of the A4 and B4 deviation polynomial coefficients. Coefficients that are omitted are set to 0. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to RTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:RTD:SPRT:COEF:LOW? (@101)  
**Response:** 0.000000E+00,0.000000E+00  
**Set Example:** TEMP:RTD:SPRT:COEF:LOW 0,0,(@101)

**Related Commands:**

SENSe:TEMPerature:RTD:TYPE  
SENSe:TEMPerature:RTD:SPRT:RTPW  
SENSe:TEMPerature:RTD:SPRT:COEFFicients:HIGH

**[SENSe:]TEMPerature:RTD:SPRT:RTPW <n>,@<channel>**

**[SENSe:]TEMPerature:RTD:SPRT:RTPW? (@<channel>)**

**Description:** Set the characterization resistance at 0.01 °C for a two-wire PRT/RTD of characterization type SPRT. <n> is the resistance, in ohms, of the PRT/RTD at 0.01 °C. (@<channel>) specifies the channels to configure. Channels in the list must be physical channels.

The channel function must already be set to temperature, the sensor type to RTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the characterization resistance to 100.0 for all channels.

**Example:** TEMP:RTD:SPRT:RTPW? (@101)

**Response:** +1.000000E+02

**Set Example:** TEMP:RTD:SPRT:RTPW 100, (@101)

**Related Commands:**

SENSe:TEMPerature:RTD:TYPE

SENSe:TEMPerature:RTD:COEFFicients:HIGH

SENSe:TEMPerature:RTD:COEFFicients:LOW

**[SENSe:]TEMPerature:RTD:TYPE <type>,@<ch\_list>**

**[SENSe:]TEMPerature:RTD:TYPE? (@<ch\_list>)**

**Description:** Set the channel function to temperature, set the sensor type to two-wire PRT/RTD, and set the specific PRT/RTD type. <type> selects the specific PRT/RTD type. The options are listed in the following table.

<type>	Description
A385	Alpha 0.00385 standard PRT/RTD (IEC 60751)
A392	Alpha 0.00392 standard PRT/RTD
ABC	Polynomial characterization
SPRT	ITS-90 characterization

(@<ch\_list>) specifies the channels to configure. Channels in the list must be physical channels.

Characterization coefficients associated with the PRT/RTD type are set to default values. The default value for RZERO is 100.0 Ω. The characterization coefficients can be changed using the SENSe:TEMPerature:RTD group of commands.

The channel scan list is not affected by this command. \*RST sets the PRT/RTD type to A385 for all channels.

**Example:** TEMP:RTD:TYPE? (@101:103)

**Response:** A385,A385,A385

**Set Example:** TEMP:RTD:TYPE A385, (@101:110)

**Related Commands:**

SENSe:TEMPerature:RTD:A385:RZERO

SENSe:TEMPerature:FRTD:TYPE

SENSe:TEMPerature:TRTD:TYPE

ROUTE:SCAN

SENSe:TEMPerature:TRANsducer

CONFigure:TEMPerature

**[SENSe:]TEMPerature:TCouple:CALCulate:VOLTage <boolean>,@<ch\_list>**  
**[SENSe:]TEMPerature:TCouple:CALCulate:VOLTage? (@<ch\_list>)**

**Description:** Enable or disable voltage calculation of thermocouple for the given channels. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels to configure. Channels in the list must be physical channels.

When the voltage calculation is on, readings of the thermocouple temperature will be compensated thermocouple voltage rather than calculated temperature. Reference junction compensation is applied depending on the reference junction setting.

The channel function must already be set to temperature and the sensor type set to TCouple. Otherwise an error is logged (403,"Conflict with channel configuration").  
\*RST disables voltage calculation for all channels.

**Example:** TEMP:TC:CALC:VOLT? (@101:102)

**Response:** 1,1

**Set Example:** TEMP:TC:CALC:VOLT ON, (@101:102)

**Related Commands:**

CONFigure:TEMPerature

SENSe:FUNCTION

SENSe:TEMPerature:TCouple:RJUNCTION:TYPE

**[SENSe:]TEMPerature:TCouple:ODETect <boolean>,@<ch\_list>**  
**[SENSe:]TEMPerature:TCouple:ODETect? (@<ch\_list>)**

**Description:** Enable or disable open-circuit detection for thermocouples. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 0. (@<ch\_list>) specifies the channels to configure. Channels in the list must be multi-function measurement channels.

The query command returns the settings for the given channels in a comma-separated list.  
\*RST disables open-circuit detection for all channels.

**Example:** TEMP:TC:ODET? (@101:102)

**Response:** 1,1

**Set Example:** TEMP:TC:ODET ON, (@101:102)

**Related Commands:**

CONFigure:TEMPerature

SENSe:TEMPerature:TRANsducer

SENSe:TEMPerature:TCouple:TYPE

*[SENSe:]TEMPerature:TCouple:POLY:COEFFicients*  
*<c0>[,<c1>[,<c2>[,<c3>[,<c4>[,<c5>[,<c6>]]]]]]],(@<channel >)*  
*[SENSe:]TEMPerature:TCouple:POLY:COEFFicients? (@<channel>)*

**Description:** Set the characterization coefficients for a polynomial characterized thermocouple. <c0> through <c6> are polynomial characterization coefficients. Coefficients that are omitted are set to 0. The coefficients define a polynomial that calculates the thermocouple millivolt output versus temperature in degrees Celsius or Fahrenheit depending on the configured system temperature unit. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to thermocouple, and the thermocouple type to POLY. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the thermocouple polynomial coefficients to 0.0 for all channels.

**Example:** TEMP:TC:POLY:COEF? (@101)

**Response:** 0.000000e+00,0.000000e+00,0.000000e+00,  
0.000000e+00,0.000000e+00,0.000000e+00

**Set Example:** TEMP:TC:POLY:COEF 0.0,0.0,0.0,0.0,0.0,0.0,(@101)

**Related Commands:**

SENSe:TEMPerature:TCouple:TYPE

*[SENSe:]TEMPerature:TCouple:R:COEFFicients <c0>[,<c1>[,<c2>[,<c3>]]],(@<channel >)*  
*[SENSe:]TEMPerature:TCouple:R:COEFFicients? (@<channel>)*

**Description:** Set the calibration coefficients for a type R thermocouple.

<c0> through <c3> are calibration coefficients when the thermocouple type is R or S. Coefficients that are omitted are set to 0. The coefficients define a polynomial that calculates a correction to the thermocouple millivolt output versus temperature in degrees Celsius or Fahrenheit depending on the configured system temperature unit.

(@<channel>) specifies the channel to configure. It must be a physical channel.

The channel function must already be set to temperature, the sensor type to thermocouple, and the thermocouple type to R. Otherwise an error is logged (403,"Conflict with channel configuration").\*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:TC:R:COEF? (@101)

**Response:** 0.000000e+00,0.000000e+00,0.000000e+00,  
0.000000e+00

**Set Example:** TEMP:TC:R:COEF 0.0,0.0,0.0,0.0,(@101)

**Related Commands:**

SENSe:TEMPerature:TCouple:TYPE

*[SENSe:]TEMPerature:TCouple:RJUNction <n>,@<ch\_list>*  
*[SENSe:]TEMPerature:TCouple:RJUNction? (@<ch\_list>)*

**Description:** Set the reference junction temperature for channels configured to measure thermocouples using fixed reference junction compensation. <n> is the temperature of the reference junction in degrees Celsius or Fahrenheit depending on the configured system temperature unit. (@<ch\_list>) specifies the channels to configure. Channels in the list must be general purpose measurement channels.

The channel function must already be set to temperature and the sensor type set to thermocouple. Otherwise an error is logged (403,"Conflict with channel configuration"). The query command returns the settings for the given channels in a comma-separated list. \*RST sets the reference junction temperature to 0.0 for all channels.

**Example:** TEMP:TC:RJUN? (@101:102)  
**Response:** 2.500000e+01,2.500000e+01  
**Set Example:** TEMP:TC:RJUN 25, (@101:102)

**Related Commands:**

SENSe:TEMPerature:TCouple:RJUNction:TYPE  
SENSe:TEMPerature:RJUNction?

*[SENSe:]TEMPerature:TCouple:RJUNction:TYPE <type>,@<ch\_list>*  
*[SENSe:]TEMPerature:TCouple:RJUNction:TYPE? (@<ch\_list>)*

**Description:** Select the reference junction compensation type for channels configured to measure thermocouples. <type> selects the reference junction type. The options are INTernal, FIXed, and EXTernal. The default is INT. If the selection is FIXed, the reference junction temperature is set to 0 in degrees Celsius or Fahrenheit depending on the configured system temperature unit. The value can be changed using the SENSe:TEMPerature:TCouple:RJUNction command. (@<ch\_list>) specifies the channels to configure. Channels in the list must be multi-function measurement channels.

The channel function must already be set to temperature and the sensor type set to thermocouple. Otherwise an error is logged (403,"Conflict with channel configuration"). The query command returns the settings for the given channels in a comma-separated list. \*RST sets the reference junction type to INT for all channels.

**Example:** TEMP:TC:RJUN:TYPE? (@101:102)  
**Response:** INT,INT  
**Set Example:** TEMP:TC:RJUN:TYPE INT, (@101:102)

**Related Commands:**

SENSe:TEMPerature:TCouple:RJUNction  
CONFigure:TEMPerature  
SENSe:TEMPerature:RJUNction?  
SENSe:TEMPerature:TRANsducer

*[SENSe:]TEMPerature:TCouple:S:COEfficients <c0>[,<c1>[,<c2>[,<c3>]]],(@<channel >)*  
*[SENSe:]TEMPerature:TCouple:S:COEfficients? (@<channel>)*

**Description:** Set the calibration coefficients for a type S thermocouple. <c0> through <c3> are calibration coefficients. Coefficients that are omitted are set to 0. The coefficients define a polynomial that calculates a correction to the standard thermocouple millivolt output versus temperature in degrees Celsius or Fahrenheit depending on the configured system temperature unit. (@<channel>) specifies the channel, which must be a general purpose measurement channel.

The channel function must already be set to temperature, the sensor type to thermocouple, and the thermocouple type to S. Otherwise an error is logged, 403,"Conflict with channel configuration". \*RST sets the thermocouple calibration coefficients to 0.0 for all channels.

**Example:** TEMP:TC:S:COEF? (@101)

**Response:** 0.000000e+00,0.000000e+00,0.000000e+00,  
0.000000e+00

**Set Example:** TEMP:TC:S:COEF 0.0,0.0,0.0,0.0,(@101)

**Related Commands:**

SENSe:TEMPerature:TCouple:TYPE

*[SENSe:]TEMPerature:TCouple:TYPE <type>,(@<ch\_list>)*

*[SENSe:]TEMPerature:TCouple:TYPE? (@<ch\_list>)*

**Description:** Set the channel function to temperature, the sensor type to thermocouple, and set the specific thermocouple type. <type> selects the thermocouple type. The options are listed in the following table.

<type>	Description
K	K thermocouple
T	T thermocouple
R	R thermocouple
S	S thermocouple
J	J thermocouple
N	N thermocouple
E	E thermocouple
B	B thermocouple
C	W5Re/W26Re thermocouple
D	W3Re/W25Re thermocouple
G	W/W26Re thermocouple
L	Fe/Constantin thermocouple
M	Ni18Mo/Ni0.8Co thermocouple
U	Cu/Constantin thermocouple
W	W/W26Re thermocouple
POLY	Polynomial characterized thermocouple

(@<ch\_list>) specifies the channels to configure. Channels in the list must be physical channels.

The reference junction type is set to internal except Ch001 (Ch001 will always be set to fixed CJC). It can be changed using the SENSE:TEMPERature:TCouple:RJUNction:TYPE command. The channel scan list is not affected. \*RST sets the thermocouple type to K for all channels.

**Example:** TEMP:TC:TYPE? (@101:102)  
**Response:** K,K  
**Set Example:** TEMP:TC:TYPE K, (@101:102)

**Related Commands:**

SENSE:TEMPERature:TCouple:RJUNction:TYPE  
ROUTE:SCAN  
SENSE:FUNCTion  
SENSE:TEMPERature:TRANsducer  
CONFigure:TEMPERature

*[SENSE:]TEMPERature:THERmistor:CALCulate:RESistance <boolean>,@<ch\_list>*  
*[SENSE:]TEMPERature:THERmistor:CALCulate:RESistance? (@<ch\_list>*

**Description:** Enable or disable resistance calculation of two-wire thermistor for the given channels. When the resistance calculation is on, readings of the thermistor temperature will be resistance in ohms rather than calculated temperature. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels to configure. Channels in the list must be physical channels.

The channel function must already be set to temperature and the sensor type set to THERmistor. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST disables resistance calculation for all channels.

**Example:** TEMP:THER:CALC:RES? (@101:105)  
**Response:** 0,0,0,0,0,0  
**Set Example:** TEMP:THER:CALC:RES 1, (@101:105)

**Related Commands:**

CONFigure:TEMPERature  
SENSE:FUNCTion

*[SENSE:]TEMPERature:THERmistor:RPOLy:COEFFicients <c0>,<c1>,<c2>,<c3>,@<channel>*  
*[SENSE:]TEMPERature:THERmistor:RPOLyCOEFFicients? (@<channel>*

**Description:** Set the characterization coefficients for a two-wire thermistor of type RPOLY. The characterization defines the thermistor resistance as a function of temperature. <c0>, <c1>, <c2>, and <c3> are the polynomial coefficients of the thermistor characterization. (@<channel>) specifies the channels to configure. Channels in the list must be physical channels.

The channel function must already be set to temperature, the sensor type to thermistor, and the thermistor type to RPOLY. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:THER:RPOL:COEF? (@101)  
**Response:** 0.000000E+00,0.000000E+00,0.000000E+00,  
0.000000E+00  
**Set Example:** TEMP:THER:RPOL:COEF 0,0,0,0, (@101)

**Related Commands:**

SENSE:TEMPERature:THERmistor:TYPE

*[SENSe:]TEMPerature:THERmistor:TYPE <type>,@<ch\_list>*

*[SENSe:]TEMPerature:THERmistor:TYPE? (@<ch\_list>)*

**Description:** Set the channel function to temperature, set the sensor type to two-wire thermistor, and set the specific thermistor type. <type> selects the specific thermistor type. The options are listed in the following table.

<type>	Description
R2K2	Nominal 2252 $\Omega$ at 25 °C
R5K	Nominal 5 k $\Omega$ at 25 °C
R10K	Nominal 10 k $\Omega$ at 25 °C
RPOLy	Polynomial characterization, resistance versus temperature

(@<ch\_list>) specifies the channels to configure. Channels in the list must be physical channels. The channel scan list is not affected. \*RST sets the thermistor type to R10K for all channels.

**Example:** TEMP:THER:TYPE? (@101:105)

**Response:** R2K2,R2K2,R2K2,R2K2,R2K2

**Set Example:** TEMP:THER:TYPE R5K,(@101)

**Related Commands:**

SENSe:TEMPerature:THERmistor:COEfficents

ROUTE:SCAN

SENSe:TEMPerature:TRANsducer

CONFigure:TEMPerature

*[SENSe:]TEMPerature:TRANsducer <sensor>,@<ch\_list>*

*[SENSe:]TEMPerature:TRANsducer? (@<ch\_list>)*

**Description:** Set the channel function to temperature and set the sensor type. <sensor> selects the sensor type. The options are TCouple, FRTD, RTD, TRTD, THERmistor, and FTHERmistor. (@<ch\_list>) specifies the channels to configure. Channels in the list must be physical channels. Two channels are paired for four-wire or three-wire sensor measurement.

The characterization type can subsequently be set using the SENSE:TEMPerature group of commands. The default characterization types are K for TCouple; A385 for FRTD, RTD, and TRTD; and R10K for THERmistor and FTHERmistor. \*RST sets the transducer type to TC for all channels.

**Example:** TEMP:TRAN? (@101:102)

**Response:** TC,TC

**Set Example:** TEMP:TRAN TC, (@101:102)

**Related Commands:**

CONFigure:TEMPerature

SENSe:FUNCTion

SENSe:TEMPerature:TCouple:TYPE

SENSe:TEMPerature:RTD:TYPE

SENSe:TEMPerature:FRTD:TYPE

SENSe:TEMPerature:THERmistor:TYPE

ROUTE:SCAN

**[SENSe:]TEMPerature:TRTD:A385:RZERo <n>,@<channel>**

**[SENSe:]TEMPerature:TRTD:A385:RZERo? (@<channel>)**

**Description:** Set the resistance at 0 °C of a three-wire PRT/RTD of characterization type A385. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to TRTD, and the characterization type to A385. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:TRTD:A385:RZER? (@101)

**Response:** 1.000000E+02

**Set Example:** TEMP:TRTD:A385:RZER 100, (@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:TYPE

**[SENSe:]TEMPerature:TRTD:A392:RZERo <n>,@<channel>**

**[SENSe:]TEMPerature:TRTD:A392:RZERo? (@<channel >)**

**Description:** Set the resistance at 0 °C of a three-wire PRT/RTD of characterization type A392. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to TRTD, and the characterization type to A392. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:TRTD:A392:RZER? (@101)

**Response:** 1.000000E+02

**Set Example:** TEMP:TRTD:A392:RZER 100, (@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:TYPE

**[SENSe:]TEMPerature:TRTD:ABC:COEFFicients <a>,<b>,<c>,@<channel>**

**[SENSe:]TEMPerature:TRTD:ABC:COEFFicients? (@<channel>)**

**Description:** Set the polynomial characterization coefficients for a three-wire PRT/RTD of characterization type ABC. <a>, <b>, and <c> are the values of the A, B, and C polynomial characterization coefficients of the PRT/RTD. (@<channel>) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to TRTD, and the characterization type to ABC. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the A, B, and C coefficients to 3.9083E-3, -5.775E-7, and -4.183E-12 respectively for all channels.

**Example:** TEMP:TRTD:ABC:COEF? (@101)

**Response:** 3.908300E-03, -5.775000E-07, -4.183000E-12

**Set Example:** TEMP:TRTD:ABC:COEF 3.9083e-3, -5.775e-7, -4.183e-12, (@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:TYPE

SENSe:TEMPerature:TRTD:ABC:RZERo

**[SENSe:]TEMPerature:TRTD:ABC:RZERo <n>,@<channel>**

**[SENSe:]TEMPerature:TRTD:ABC:RZERo? (@<channel>)**

**Description:** Set the resistance at 0 °C for a three-wire PRT/RTD of characterization type ABC. <n> is the resistance, in ohms, of the PRT/RTD at 0 °C. (@<channel>) specifies the channel, which must be a general purpose measurement channel. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to TRTD, and the characterization type to ABC. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the 0 °C resistance to 100.0 for all channels.

**Example:** TEMP:TRTD:ABC:RZER? (@101)

**Response:** 1.000000E+02

**Set Example:** TEMP:TRTD:ABC:RZER 100, (@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:TYPE

SENSe:TEMPerature:TRTD:ABC:COEFFicients

**[SENSe:]TEMPerature:TRTD:CALCulate:RESistance <boolean>,@<ch\_list>**

**[SENSe:]TEMPerature:TRTD:CALCulate:RESistance? (@<ch\_list>)**

**Description:** Enable or disable resistance calculation of three-wire PRT/RTD for the given channels. When the resistance calculation is on, readings of the PRT/RTD temperature will be resistance in ohms rather than calculated temperature. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature and the sensor type set to TRTD. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST disables resistance calculation for all channels.

**Example:** TEMP:TRTD:CALC:RES? (@101:105)

**Response:** 0,0,0,0,0

**Set Example:** TEMP:TRTD:CALC:RES 1, (@101:105)

**Related Commands:**

CONFigure:TEMPerature

SENSe:FUNction

*[SENSe:]TEMPerature:TRTD:SPRT:COEFFicients:HIGH <a>[,<b>[,<c>[,<d>]]],(@<channel>)*  
*[SENSe:]TEMPerature:TRTD:SPRT:COEFFicients:HIGH? (@<channel>)*

**Description:** Set the deviation polynomial coefficients for the range above 0 °C for a four-wire PRT/RTD of characterization type SPRT. <a>, <b>, <c>, and <d> are the values of the A, B, C, and D deviation polynomial characterization coefficients. Coefficients that are omitted are set to 0. (@<channel>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to TRTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:TRTD:SPRT:COEF:HIGH? (@101)  
**Response:** 0.000000E+00,0.000000E+00,0.000000E+00,  
0.000000E+00  
**Set Example:** TEMP:TRTD:SPRT:COEF:HIGH 0,0,0,0,(@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:TYPE  
SENSe:TEMPerature:TRTD:RTPW  
SENSe:TEMPerature:TRTD:COEFFicients:LOW

*[SENSe:]TEMPerature:TRTD:SPRT:COEFFicients:LOW <a>[,<b>],(@<channel>)*  
*[SENSe:]TEMPerature:TRTD:SPRT:COEFFicients:LOW? (@<channel>)*

**Description:** Set the deviation polynomial coefficients for the range below 0 °C for a three-wire PRT/RTD of characterization type SPRT. <a> and <b> are the values of the A4 and B4 deviation polynomial characterization coefficients. Coefficients that are omitted are set to 0. (@<channel>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to TRTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the coefficients to 0.0 for all channels.

**Example:** TEMP:TRTD:SPRT:COEF:LOW? (@101)  
**Response:** 0.000000E+00,0.000000E+00  
**Set Example:** TEMP:TRTD:SPRT:COEF:LOW 0,0,(@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:TYPE  
SENSe:TEMPerature:TRTD:RTPW  
SENSe:TEMPerature:TRTD:COEFFicients:HIGH

**[SENSe:]TEMPerature:TRTD:SPRT:RTPW <n>,@<channel>**

**[SENSe:]TEMPerature:TRTD:SPRT:RTPW? (@<channel>)**

**Description:** Set the characterization resistance at 0.01 °C for a three-wire PRT/RTD of characterization type SPRT. <n> is the resistance, in ohms, of the PRT/RTD at 0.01 °C. (@<channel>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for three-wire sensor measurement.

The channel function must already be set to temperature, the sensor type to TRTD, and the characterization type to SPRT. Otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the characterization resistance to 100.0 for all channels.

**Example:** TEMP:TRTD:SPRT:RTPW? (@101)

**Response:** 1.000000E+02

**Set Example:** ES TEMP:TRTD:SPRT:RTPW 25.5, (@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:TYPE

SENSe:TEMPerature:TRTD:COEFFicients:HIGH

SENSe:TEMPerature:TRTD:COEFFicients:LOW

**[SENSe:]TEMPerature:TRTD:TYPE <type>,@<ch\_list>**

**[SENSe:]TEMPerature:TRTD:TYPE? (@<ch\_list>)**

**Description:** Set the channel function to temperature, set the sensor type to three-wire PRT/RTD, and set the specific PRT/RTD type. <type> selects the specific PRT/RTD type. The options are listed in the following table.

<type>	Description
A385	Alpha 0.00385 standard PRT/RTD (IEC 60751)
A392	Alpha 0.00392 standard PRT/RTD
ABC	Polynomial characterization
SPRT	ITS-90 characterization

(@<ch\_list>) specifies the channels, which must be general purpose measurement channels. Two channels are paired for three-wire sensor measurement.

Characterization coefficients associated with the PRT/RTD type are set to default values. The default value for RZERO is 100.0 Ω. The characterization coefficients can be changed using the SENSe:TEMPerature:TRTD group of commands. The channel scan list is not affected. \*RST sets the PRT/RTD type to A385 for all channels.

**Example:** TEMP:TRTD:TYPE? (@101:105)

**Response:** A385,A385,A385,A385,A385

**Set Example:** TEMP:TRTD:TYPE A392, (@101)

**Related Commands:**

SENSe:TEMPerature:TRTD:RZERO

ROUTE:SCAN

SENSe:TEMPerature:TRANsducer

CONFigure:TEMPerature

*[SENSe:]TOTalize:CLEar*

**Description:** Clear the totalize count. \*RST also clears the totalize count.

**Example:** TOT:CLE

**Related Commands:**

TOTalize:DATA?  
TOTalize:STATE

*[SENSe:]TOTalize:DATA?*

**Description:** Read the totalize count. \*RST clears the totalize count.

**Example:** TOT:DATA?

**Response:** 841

**Related Commands:**

TOTalize:CLEar  
TOTalize:STATE

*[SENSe:]TOTalize:DEBounce <boolean>*

*[SENSe:]TOTalize:DEBounce?*

Enable or disable totalize debounce. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 0. \*RST clears and disables the totalize counter.

**Example:** TOT:DEB?

**Response:** 1

**Set Example:** TOT:DEB ON

**Related Commands:**

TOTalize:STATE

*[SENSe:]TOTalize:STATE <boolean>*

*[SENSe:]TOTalize:STATE?*

**Description:** Enable or disable the totalize counter. <boolean> can be 0 (off), OFF, 1 (on), or ON. \*RST clears and disables the totalize counter.

**Example:** TOT:STAT?

**Response:** 1

**Set Example:** TOT:STAT ON

**Related Commands:**

TOTalize:DATA?  
TOTalize:CLEar  
TOTalize:DEBounce

*[SENSe:]TOTalize:TYPE <mode>*

*[SENSe:]TOTalize:TYPE?*

**Description:** Enable or disable automatic reset of the totalize count. <mode> can be READ or RRESet. When <mode> is READ, reading the totalizer count has no effect on the count. When <mode> is RRESet, reading the totalizer count resets the count to 0. \*RST clears and disables the totalize counter.

**Example:** TOT:TYPE?

**Response:** READ

**Set Example:** TOT:TYPE RRES

**Related Commands:**

TOTalize:DATA?  
TOTalize:CLEar

**[SENSe:]VOLTage[:DC]:RANGe <range>[,(@<ch\_list>)]**  
**[SENSe:]VOLTage[:DC]:RANGe? [(@<ch\_list>)]**

**Description:** Select the dc voltage range for the given channels. <range> is a number that selects the measurement range. The lowest discrete range that covers the given <range> value, in volts, will be used. Resolution is automatic and depends on the range and sample rate. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels.

The channel function must already be set to dc voltage, otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets the range to automatic for all channels.

**Example:** VOLT:RANG? (@101:102)  
**Response:** 1.000000e+00,1.000000e+00  
**Set Example:** VOLT:RANG 1, (@101:102)

**Related Commands:**

CONFigure:VOLTage:DC  
 SENSe:FUNCTion  
 SENSe:VOLTage:DC:RANGe:AUTO

**[SENSe:]VOLTage[:DC]:RANGe:AUTO <boolean>[,(@<ch\_list>)]**  
**[SENSe:]VOLTage[:DC]:RANGe:AUTO? [(@<ch\_list>)]**

**Description:** Enable or disable automatic dc voltage range for the given channels. <boolean> can be 0 (off), OFF, 1 (on), or ON. (@<ch\_list>) specifies the channels, which must be general purpose measurement channels.

The channel function must already be set to dc voltage, otherwise an error is logged (403,"Conflict with channel configuration"). \*RST sets automatic range on for all channels.

**Example:** VOLT:RANG:AUTO? (@101:102)  
**Response:** 1,1  
**Set Example:** VOLT:RANG:AUTO ON, (@101:102)

**Related Commands:**

CONFigure:VOLTage:DC  
 SENSe:FUNCTion  
 SENSe:VOLTage:DC:RANGe

**SOURce:DIGital:DATA:BYTE <n>**  
**SOURce:DIGital:DATA:BYTE?**

**Description:** Set the digital I/O port output data byte. <n> is a decimal value between 0 and 255 that is the sum of the decimal values of the bits that are 1.

The query command returns the data byte that is output to the digital I/O port. The actual state of the I/O port may be different from the output byte because the port can also be controlled externally. \*RST sets the I/O port output byte to 255 (all bits high).

**Example:** SOUR:DOG:DATA:BYTE?  
**Response:** 170  
**Set Example:** SOUR:DOG:DATA:BYTE 170

**Related Commands:**

SENSe:DIGital:DATA:BYTE?

**STATus:ALARm:CONDition?**

**Description:** Query the Alarm Status Condition register. The Alarm Status Condition register reflects the states of alarms. The return value is the sum of the decimal values of the asserted bits of the register. The defined bits, their decimal values (in parentheses), and their meanings are as follows:

- Bit 0 (1): Alarm port 1 is asserted.
- Bit 1 (2): Alarm port 2 is asserted.
- Bit 2 (4): Alarm port 3 is asserted.
- Bit 3 (8): Alarm port 4 is asserted.
- Bit 4 (16): Alarm port 5 is asserted.
- Bit 5 (32): Alarm port 6 is asserted.
- Bit 8 (256): One or more channels have an asserted alarm condition.
- Bit 9 (512): The alarm queue contains one or more alarms.
- Bit 10 (1024): The alarm queue is full.

\*CLS clears the alarm queue, so Bit 9 is cleared. \*RST clears all alarms and the Alarm Status Condition register is updated with the alarm-related bits cleared. \*RST does not clear the alarm queue, so Bit 9 and 10 are not cleared.

**Example:** STAT:ALAR:COND?

**Response:** 768

**Related Commands:**

STATus:ALARm:EVENT?

OUTPut:ALARm?

SYSTem:ALARm?

CALCulate:LIMit1:STATe

CALCulate:LIMit:CLEAr:ALL

**STATus:ALARm:ENABLE <n>**

**STATus:ALARm:ENABLE?**

**Description:** Set the Alarm Status Enable register. The Alarm Status Enable register determines which bits of the Alarm Status Event register are reported in the Alarm Status Summary bit of the Status Byte register. <n> is the sum of the decimal values of the bits of the Alarm Status Event register that will be reported in the Status Byte register.

The Status Byte register is updated to reflect any change in the enable registers associated with the summary bits. Alarm Status Enable is set to 0 at power-on if the \*PSC setting is 1. \*CLS does not affect Alarm Status Enable. \*RST does not affect Alarm Status Enable.

**Example:** STAT:ALAR:ENAB?

**Response:** 768

**Set Example:** STAT:ALAR:ENAB 768

**Related Commands:**

STATus:ALARm:EVENT?

\*STB?

\*PSC

**STATus:ALARm[:EVENTi]?**

**Description:** Query and clear the Alarm Status Event register. The return value is the sum of the decimal values of the asserted bits of the register. The defined bits, their decimal values (in parentheses), and their meanings are as follows:

- Bit 0 (1): Alarm port 1 has been asserted.
- Bit 1 (2): Alarm port 2 has been asserted.
- Bit 2 (4): Alarm port 3 has been asserted.
- Bit 3 (8): Alarm port 4 has been asserted.
- Bit 4 (16): Alarm port 5 has been asserted.
- Bit 5 (32): Alarm port 6 has been asserted.
- Bit 8 (256): One or more channels have asserted an alarm condition.
- Bit 9 (512): An alarm message was added in the alarm queue.
- Bit 10 (1024): The alarm queue overflowed.

An asserted bit of the Alarm Status Event register will assert the Alarm Status Summary bit (Bit 1) of the Status Byte register if the corresponding bit of the Alarm Status Enable register is set. \*CLS clears all the bits in the Alarm Status Event register. \*RST does not directly affect the Alarm Status Event register.

**Example:** STAT:ALAR?

**Response:** 768

**Related Commands:**

STATus:ALARm:CONDition?

STATus:ALARm:ENABle

OUTPut:ALARm?

SYSTem:ALARm?

CALCulate:LIMit1:STATe

\*STB?

**STATus:OPERation:CONDition?**

**Description:** Query the Operation Status Condition register. The Operation Status Condition register reflects various states of operation. The return value is the sum of the decimal values of the asserted bits of the register. The defined bits, their decimal values (in parentheses), and their meanings are as follows:

- Bit 0 (1): A calibration operation is in progress.
- Bit 4 (16): A scan sweep or measurement is in progress.
- Bit 5 (32): Scanning is waiting for a trigger.
- Bit 6 (64): Triggering is suspended or an automated test is waiting for operator input.
- Bit 8 (256): Scanning is active or an automated test is in progress.
- Bit 9 (512): Monitor is active.

\*RST resets the measurement system and returns it to the idle state, and the Operation Status Condition register is updated to reflect the new state.

**Example:** STAT:OPER:COND?

**Response:** 272

**Related Commands:**

STATus:OPERation:EVENT?

INITiate:IMMediate

TRIGger:ENABle

**STATus:OPERation:ENABLE <n>**  
**STATus:OPERation:ENABLE?**

**Description:** Set the Operation Status Enable register. The Operation Status Enable register determines which bits of the Operation Status Event register are reported in the Operation Status Summary bit of the Status Byte register. <n> is the sum of the decimal values of the bits of the Operation Status Event register that will be reported in the Status Byte register.

The Status Byte register is updated to reflect any change in the enable registers associated with the summary bits. Operation Status Enable is set to 0 at power-on if the \*PSC setting is 1. \*CLS does not affect Operation Status Enable. \*RST does not affect Operation Status Enable.

**Example:** STAT:OPER:ENAB?  
**Response:** 255  
**Set Example:** STAT:OPER:ENAB 1024

**Related Commands:**

STATus:OPERation:EVENT?  
\*STB?  
\*PSC

**STATus:OPERation[:EVENT]?**

**Description:** Query and clear the Operation Status Event register. The return value is the sum of the decimal values of the asserted bits of the register. The defined bits, their decimal values (in parentheses), and their meanings are as follows:

- Bit 0 (1): A calibration operation completed or was cancelled.
- Bit 4 (16): A scan sweep completed.
- Bit 5 (32): Scanning has entered a waiting-for-trigger state.
- Bit 6 (64): Trigger was suspended or an automated test stopped to await operator input.
- Bit 8 (256): Scanning completed or an automated test completed.
- Bit 9 (512): A Monitor reading completed.

An asserted bit of the Operation Status Event register will assert the Operation Status Summary bit (Bit 7) of the Status Byte register if the corresponding bit of the Operation Status Enable register is set. \*CLS clears all the bits in the Operation Status Event register. \*RST does not directly affect the Operation Status Event register.

**Example:** STAT:OPER?  
**Response:** 16

**Related Commands:**

STATus:OPERation:CONDition?  
STATus:OPERation:ENABLE  
\*STB?  
INITiate:IMMediate  
TRIGger:ENABLE

**STATus:PRESet**

**Description:** Set status enable registers to disabled states. The Alarm Status Enable, Questionable Status Enable, and Operation Status Enable registers are all set to 0. The Status Byte register is updated to reflect any changes in the enable registers associated with the summary bits. The Event Status Enable and Service Request Enable registers are not affected.

**Example:** STAT:PRES

**Related Commands:**

STATus:ALARm:ENABle  
 STATus:OPERation:ENABle  
 STATus:QUEStionable:ENABle  
 \*STB?  
 \*ESE  
 \*SRE

**STATus:QUEStionable:CONDition?**

**Description:** Query the Questionable Status Condition register. The Questionable Status Condition register reports out-of-range or failed measurements. The return value is the sum of the decimal values of the asserted bits of the register. The defined bits, their decimal values (in parentheses), and their meanings are as follows:

- Bit 0 (1): A voltage reading is out-of-range.
- Bit 1 (2): A current reading is out-of-range.
- Bit 4 (16): A temperature reading is out-of-range.
- Bit 8 (256): The present calibration operation failed.
- Bit 9 (512): A resistance reading is out-of-range.
- Bit 10 (1024): A math reading is out-of-range.
- Bit 11 (2048): The totalizer count is out-of-range.
- Bit 12 (4096): Scan memory is full, readings may be lost.

\*RST clears the measurement system and the Questionable Status Condition register is updated with the bits cleared.

**Example:** STAT:QUES:COND?

**Response:** 256

**Related Commands:**

STATus:QUEStionable:EVENT?

*STATus:QUEStionable:ENABle <n>*  
*STATus:QUEStionable:ENABle?*

**Description:** Set the Questionable Status Enable register. The Questionable Status Enable register determines which bits of the Questionable Status Event register are reported in the Questionable Status Summary bit of the Status Byte register. <n> is the sum of the decimal values of the bits of the Questionable Status Event register that will be reported in the Status Byte register.

The Status Byte register is updated to reflect any change in the enable registers associated with the summary bits. Questionable Status Enable is set to 0 at power-on if the \*PSC setting is 1. \*CLS does not affect Questionable Status Enable. \*RST does not affect Questionable Status Enable.

**Example:** STAT:QUES:ENAB?  
**Response:** 256  
**Set Example:** STAT:QUES:ENAB 256

**Related Commands:**  
STATus:QUEStionable:EVENT?  
\*STB?  
\*PSC

*STATus:QUEStionable [:EVENT]?*

**Description:** Query and clear the Questionable Status Event register. The return value is the sum of the decimal values of the asserted bits of the register. The defined bits, their decimal values (in parentheses), and their meanings are as follows:

- Bit 0 (1): A voltage reading was out-of-range.
- Bit 1 (2): A current reading was out-of-range.
- Bit 4 (16): A temperature reading was out-of-range.
- Bit 8 (256): A calibration operation failed.
- Bit 9 (512): A resistance reading was out-of-range.
- Bit 10 (1024): A math reading was out-of-range.
- Bit 11 (2048): The totalizer count overflowed.
- Bit 12 (4096): Scan memory overflowed.

An asserted bit of the Questionable Status Event register will assert the Questionable Status Summary bit (Bit 3) of the Status Byte register if the corresponding bit of the Questionable Status Enable register is set. \*CLS clears all the bits in the Questionable Status Event register. \*RST does not directly affect the Questionable Status Event register.

**Example:** STAT:QUES?  
**Response:** 256

**Related Commands:**  
STATus:QUEStionable:CONDition?  
STATus:QUEStionable:ENABle  
\*STB?

**SYSTem:ALARm?**

**Description:** Read and remove the first alarm in the alarm queue. The response contains the reading causing the alarm, unit of measurement, channel number, date, and time formatted as follows:

<reading>,<unit>,<channel>,<year>,<month>,<day>,<hour>,<minute>,<second>,<type>,<output>

If the alarm queue is empty, the response is “0.000000e+00,,000,0000,00,00,00,00,00,000,0,0”. If the alarm queue is full, no more alarms will be logged until the alarm queue is read. The alarm queue can hold up to 16 alarms.

**Example:** SYST:ALAR?

**Response:** 8.919101e-04,VDC,001,1999,11,30,00,20,43.000,1,0

**Related Commands:**

CALCulate:LIMit1:STATe?

OUTPut:ALARm?

STATus:ALARm:EVENT?

**SYSTem:BEEPer:KEY <boolean>****SYSTem:BEEPer:KEY?**

**Description:** Enable or disable the key beeper. <boolean> can be 0 (disabled), OFF, 1 (enabled), or ON. The default is 1. If the key beeper is enabled, the beeper will beep whenever a front panel key is pressed. \*RST does not affect the beeper setting.

**Example:** SYST:BEEP:KEY?

**Response:** 1

**Set Example:** SYST:BEEP:KEY ON

**SYSTem:COMMunicate:SERial:BAUD <baud>****SYSTem:COMMunicate:SERial:BAUD?**

**Description:** Set the serial port baud rate. <baud> is the baud rate number in bits per second. Allowed values are 1200, 2400, 4800, 9600, and 19200. The default is 2400.

The serial port is used to automatically control the temperature source instrument during an automated test. \*RST does not affect the serial port baud rate setting.

**Example:** SYST:COMM:SER:BAND?

**Response:** 9600

**Set Example:** SYST:COMM:SER:BAND 9600

**Related Commands:**

TRIGger:AUTO:CONTRol

**SYSTem:COMMunicate:TERMinator {CR | LF | CRLF}****SYSTem:COMMunicate:TERMinator?**

**Description:** Set the termination character for response strings. The setting can be CR (ASCII 13 decimal), LF (ASCII 10), or CRLF (both characters). The setting only affects the termination character of responses transmitted from the Instrument. Regardless of the setting, commands are accepted whether they are terminated by a CR character, LF character, or both.

**Example:** SYST:COMM:TERM?

**Response:** CR

**Set Example:** SYST:COMM:TERM LF

**SYSTem:CONFigure:MAC?**

**Description:** Query the LAN MAC address.

**Example:** SYST:CONF:MAC?

**Response:** "00:80:40:01:ff:fb"

**SYSTem:DATE <year>,<month>,<day>**

**SYSTem:DATE?**

**Description:** Set the clock date. <year> is the present four-digit year. <month> is the present month number, from 1 to 12. <day> is the present day within the month, from 1 to 31. \*RST does not affect the clock date.

**Example:** SYST:DATE?

**Response:** 2013,05,07

**Set Example:** SYST:DATE 2013,05,07

**Related Commands:**

SYSTem:TIME

**SYSTem:ERRor?**

**Description:** Read and remove the first error in the error queue. The response is an error number followed by a brief description in quotes, with the two parts separated by a comma. The earliest error logged is returned and removed from the error queue. If the error queue is empty, the response is 0, "No Error". If the error queue contains at least one error, the Error Queue Summary Bit (bit 2) of the Status Byte Register will be set.

If the error queue was overfilled, the most recent error in the error queue will be replaced by the error -350, "Queue overflow". The error queue can hold up to 10 errors. An error is logged in the error queue if a received command cannot be executed for any reason. Errors are categorized as one of the following types: command error, execution error, device-specific error, or query error.

Command errors report syntax errors or unrecognized commands. An error of this type will cause the Command Error Bit (bit 5) of the Standard Event Status Register to be set.

Execution errors report failures to execute recognized commands. The form of the command is not acceptable, parameters are incorrect or invalid, or the command is not valid in the present state of the instrument or its settings. An error of this type will cause the Execution Error Bit (bit 4) of the Standard Event Register to be set.

Device-specific errors report problems indirectly caused by commands or that are due to abnormal operation of the instrument. An error of this type will cause the Device Dependent Error Bit (bit 3) of the Standard Event Status Register to be set.

Query errors report communication bus errors during transmission of a response to a query command. An error of this type will cause the Query Error Bit (bit 2) of the Standard Event Status Register to be set.

Error bits in the Standard Event Status Register affect the Event Summary Bit (bit 5) of the Status Byte Register if the corresponding bits in the Event Status Enable register are set. Possible error responses can be found in the Users Manual.

No response is available for query, or a response was not transmitted before a new command was received. \*CLS clears the error queue. \*RST does not clear the error queue.

**Example:** SYST:ERR?

**Response:** 0, "No error"

**Related Commands:**

\*CLS

\*ESR?

\*STB?

**SYSTem:KLOCK <boolean>****SYSTem:KLOCK?**

**Description:** Lock or unlock the front panel keys. <boolean> can be 0 (unlocked), OFF, 1 (locked), or ON. The default is 0. \*RST unlocks the front panels keys.

**Example:** SYST:KLOC?

**Response:** 0

**Set Example:** SYST:KLOC OFF

**SYSTem:LFRequency?**

**Description:** Query the power line frequency. The response is the power line frequency in Hertz. It is typically 50 Hz or 60 Hz.

**Example:** SYST:LFR?

**Response:** 50

**SYSTem:MODule:CONFigure:SNUM? <slot>**

**Description:** Query the specified module's serial number. <slot> can be 1 or 2.

**Example:** SYST:MOD:CONF:SNUM? 1

**Response:** 12345678

**SYSTem:MODule:CONFigure:TYPE? <slot>**

**Description:** Query the specified module's type, <slot> can be 1 or 2.

**Example:** SYST:MOD:CONF:TYPE? 1

**Response:** "1586-2586"

**SYSTem:PON:TIME?**

**Description:** Query the time elapsed since power-on. The response is the time elapsed in the following format: <days>,<hours>,<minutes>,<seconds>

The time since power-on may be useful to ensure that the instrument has had adequate warm-up time before it is used to make critical measurements.

**Example:** SYST:PON:TIME?

**Response:** 3,19,40,51

**Related Commands:**

\*ESR?

**SYSTem:SECurity:IMMEDIATE**

**Description:** Clear memory for security. The command immediately clears scan memory, restores all setup settings to default values, and erases all set-up files and data files. This may be useful in secure facilities that require instrument memory to be completely erased.

The command does not clear or reset passwords, instrument calibration, instrument identification, or MAC address. The command does not erase a USB memory device that may be attached.

**Example:** SYST:SEC:IMM

**Related Commands:**

\*RST

MEMory:LOG:CLEar

MEMory:STATE:CLEar

**SYSTem:TIME** <hour>,<minute>,<second>

**SYSTem:TIME?**

**Description:** Set the clock time. <hour> is the present hour of the day, from 0 to 23. <minute> is the present minute, from 0 to 59. <second> is the present second, from 0 to 59. \*RST does not affect the clock time.

**Example:** SYST:TIME?

**Response:** 14,00,00

**Set Example:** SYST:TIME 9,33,00

**Related Commands:**

SYSTem:DATE

**SYSTem:VERsion?**

**Description:** Query the SCPI version. The return value is '1999.0'.

**Example:** SYST:VERS?

**Response:** 1999.0

**Related Commands:**

\*IDN?

**TRIGger:ALARm:CHANnel** (@<channel>)

**TRIGger:ALARm:CHANnel?**

**Description:** Set the input channel for the alarm trigger. (@<channel>) is the input channel number. The default is 0 (None). This setting is applicable when ALARm is the trigger source. After scanning is initiated the designated input will be monitored. When the channels readings exceed the channel's alarm limits a scan sweep will be triggered. For the designated channel to trigger a scan an alarm limit for the channel must be set and enabled. \*RST sets the alarm trigger channel to 0 (none).

**Example:** TRIG:ALAR:CHAN?

**Response:** 101

**Set Example:** TRIG:ALAR:CHAN (@101)

**Related Commands:**

TRIGger:SOURce

CALCulate:LIMit1

CALCulate:LIMit1:STATe

*TRIGger:AUTO:CONTRol <boolean>*

*TRIGger:AUTO:CONTRol?*

**Description:** Enable or disable automatic source instrument control for automated test. <boolean> can be 0 (off), OFF, 1 (on), or ON. The default is 0.

When source instrument control is enabled, the Instrument will send commands from the serial port to set the source set-point. This feature is compatible with Fluke calibration temperature sources.

When source instrument control is disabled, the source instrument set-point must be manually set by the operator at each test point. Each time a test point scan sweep completes, the automated test stops to wait for next input. Also, bit 6 of the Operation Status Condition register and bit 6 of the Operation Status Event register are set. When ready, the automated test goes to the next point automatically. \*RST disables source instrument control.

**Example:** TRIG:AUTO:CONT?

**Response:** 1

**Set Example:** TRIG:AUTO:CONT ON

**Related Commands:**

TRIGger:AUTO:SPOint

STATus:OPERation:CONDition?

STATus:OPERation:EVENT?

INITiate:IMMediate

TRIGger:ENABle

*TRIGger:AUTO:DWELI <h>[,<m>[,<s>[,<point number>]]]*

*TRIGger:AUTO:DWELI? [<point number>]*

**Description:** Set the soak time for test points in automated test. <h>, <m>, and <s> are the number of hours, minutes, and seconds. The soak time can range from 00:00:00 to 23:59:59. The default is 00:10:00. <point number> is the ordinal number of the test point. The range is 1 to 20. If the <point number> parameter is omitted, the setting is applied to all test points. The number of points should be selected with the TRIG:AUTO:POIN command prior to setting any test point settings.

The soak timer will start after the automated test advances to the next test point and the readings from the reference channel satisfy the tolerance and stability requirements. Then as soon as the soak timer reaches the soak time it will trigger a scan sweep. An adequate soak time ensures that the system has stabilized before a scan sweep starts. \*RST sets the soak time to 00:10:00.

**Example:** TRIG:AUTO:DWEL? 1

**Response:** 00,10,00

**Set Example:** TRIG:AUTO:DWEL 00,10,00,1

**Related Commands:**

TRIGger:AUTO:POINTs

TRIGger:AUTO:SPOint

*TRIGger:AUTO:ORDER <order>*  
*TRIGger:AUTO:ORDER?*

**Description:** Select the channel order during scan sweeps in automated test.

<order> selects the channel order. The options are LINear, REVersing, and ALternating. The default is LINear.

LINear results in a scan sweep the begins with the primary reference channel, if specified, proceeds in ascending order of channel number, and concludes with the secondary reference channel, if specified.

REVersing results in a scan sweep of two parts. The first part is the same as the LINear scan sweep. The second part is the same in reverse.

ALternating results in a scan sweep in which the primary reference channel, if specified, and the secondary reference channel, if specified, are sampled just prior to every other channel, and the general order of channels is in ascending order by channel number.

With either order, channels may be sampled multiple times depending on the channel sample count setting. \*RST sets the channel order to LINear.

**Example:** TRIG:AUTO:ORD?

**Response:** LIN

**Set Example:** TRIG:AUTO:ORD LIN

**Related Commands:**

TRIGger:AUTO:CHANnel:COUNT

TRIGger:AUTO:REFerence

TRIGger:AUTO:SREFerence

TRIGger:AUTO:POINTs

*TRIGger:AUTO:POINTs <n>*  
*TRIGger:AUTO:POINTs?*

**Description:** Set the number of test points in the automated test. <n> is the number of test points. The range is 1 to 20, and the default is 1. The number of points should be selected prior to setting any specific test point settings. Channels to include in the automated test are selected using the ROUTe:SCAN command.

Scan sweeps are performed at each test point after the automated test advances to the test point and the tolerance and stability requirements of the reference channel are satisfied and the soak time elapses. The automated test can advance to the next test point automatically or manually by the operator, depending on the control setting.

The command INITiate:IMMediate may be used to start an automated test. \*RST sets the number of test points to 1.

**Example:** TRIG:AUTO:POIN?

**Response:** 2

**Set Example:** TRIG:AUTO:POIN 2

**Related Commands:**

TRIGger:AUTO:SPOint

TRIGger:AUTO:CONTRol

ROUTE:SCAN

TRIGger:AUTO:REFerence

TRIGger:AUTO:ORDER

TRIGger:AUTO:TOLerance

TRIGger:AUTO:STABILity

TRIGger:AUTO:DWELL

TRIGger:SOURce

```
TRIGger:COUNT
INITiate:IMMediate
TRIGger:ENABle
```

*TRIGger:AUTO:REFerence {NONE | (@<channel>)}*  
*TRIGger:AUTO:REFerence?*

**Description:** Set the primary reference channel for automated test. (@<channel>) is the primary reference channel number.

When a channel is specified, it will be monitored for tolerance and stability and included measured in the scan sweep in the reference channel position. The reference channel must be previously set up and enabled in the scan list. If the specified channel is not enabled in the scan list, an error is logged (407,"Reference channel not enabled"). \*RST sets the primary reference to 0 (none).

**Example:** TRIG:AUTO:REF?  
**Response:** 101  
**Set Example:** TRIG:AUTO:REF (@101)

**Related Commands:**

```
ROUTe:SCAN
TRIGger:AUTO:SREFerence
TRIGger:AUTO:TOLerance
TRIGger:AUTO:STABIlity
```

*TRIGger:AUTO:SPOint <n>[,<point number>]*  
*TRIGger:AUTO:SPOint? [<point number>]*

**Description:** Set the set-point value for test points in automated test. <n> is the set-point value as a floating-point number. The default is 0.0. <point number> is the ordinal number of the test point. The range is 1 to 20. If the <point number> parameter is omitted, the setting applies to all test points.

The number of points should be selected with the TRIGger:AUTO:POINTs command prior to setting any test point settings. \*RST sets the set-point value to 0.0 for all points.

**Example:** TRIG:AUTO:SPO? 1  
**Response:** 1.000000e+02  
**Set Example:** TRIG:AUTO:SPO 100, 1

**Related Commands:**

```
TRIGger:AUTO:POINTs
TRIGger:AUTO:TOLerance
TRIGger:AUTO:STABIlity
TRIGger:AUTO:DWELL
TRIGger:AUTO:CONTRol
```

*TRIGger:AUTO:SREference {NONE | (@<channel>)}*  
*TRIGger:AUTO:SREference?*

**Description:** Set the secondary reference channel for automated test. (@<channel>) is the secondary reference channel number. The default is no channel, '@'. If no channel is specified, the scan sweep will not include samples of a secondary reference channel. When a channel is specified, it will be included in the scan sweep in the secondary reference position in the channel order. The reference channel must be previously set up and enabled in the scan list. If the specified channel is not enabled in the scan list, an error is logged (407, "Reference channel not enabled"). \*RST sets the secondary reference to 0 (none).

**Example:** TRIG:AUTO:SREF?  
**Response:** 102  
**Set Example:** TRIG:AUTO:SREF (@102)

**Related Commands:**

TRIGger:AUTO:TOLerance  
TRIGger:AUTO:STABility  
TRIGger:AUTO:REFerence

*TRIGger:AUTO:STABility <n>[,<point number>]*  
*TRIGger:AUTO:STABility? [<point number>]*

**Description:** Set the reference stability limit for test points in automated test. <n> is the stability value as a floating-point number. The default is 1.0. <point number> is the ordinal number of the test point. The range is 1 to 20. If the <point number> parameter is omitted, the setting applies to all test points. The number of points should be selected with the TRIGger:AUTO:POINTs command prior to setting any test point settings.

At each test point, the primary reference channel, if specified, is first monitored for tolerance from the set-point value. Once the tolerance is reached, the primary reference channel is then monitored for stability and soak time starts at the same time. A PTP value of the samples is calculated during the soak time. When the PTP value is less than the stability and the soak time completes, a scan sweep is triggered. If a PTP value of the samples is more than the stability during the soak time, the soak time will be reset. A scan sweep is triggered when the soak time completes.

\*RST sets the reference stability limit to 1.0 for all test points.

**Example:** TRIG:AUTO:STAB? 1  
**Response:** 1.000000e+00  
**Set Example:** TRIG:AUTO:STAB 1,1

**Related Commands:**

TRIGger:AUTO:POINTs  
TRIGger:AUTO:SPOINT  
TRIGger:AUTO:CONTRol  
TRIGger:AUTO:TOLerance  
TRIGger:AUTO:DWELL

*TRIGger:AUTO:TOLerance <n>[,<point number>]*

*TRIGger:AUTO:TOLerance? [<point number>]*

**Description:** Set the reference tolerance limit for test points in automated test. <n> is the tolerance value as a floating-point number. The default is 1.0. <point number> is the ordinal number of the test point. The range is 1 to 20. If the <point number> parameter is omitted, the setting applies to all test points.

The number of points should be selected with the TRIGger:AUTO:POINTs command prior to setting any test point settings. At each test point, the primary reference channel, if specified, is first monitored for accuracy relative to the set-point value. The magnitude of the difference between the latest reading and the set-point value is calculated. Once the difference is within the tolerance setting, the primary reference channel is then monitored for stability. A scan sweep is finally triggered when the tolerance and stability are satisfied and the soak time completes. \*RST sets the reference tolerance limit to 1.0 for all test points.

**Example:** TRIG:AUTO:TOL? 1

**Response:** 1.000000e+00

**Set Example:** TRIG:AUTO:TOL 1,1

**Related Commands:**

TRIGger:AUTO:POINTs

TRIGger:AUTO:SPOINT

TRIGger:AUTO:CONTROL

TRIGger:AUTO:STABILITY

TRIGger:AUTO:DWELL

*TRIGger:COUNt {<n> | INFINity}*

*TRIGger:COUNt?*

**Description:** Set the scan count, which is the maximum number of scan sweeps that will occur after scanning is initiated. <n> is the scan count number. The range is 1 to 99999 and 0 or INFINity. The default is 1.

Scanning will start when initiated and will stop immediately after <n> scan sweeps are performed. If the scan count is 0 (INFINity), scanning will continue indefinitely until the ABORT command or \*RST command is issued. If the scan count is 1, there will be only one scan sweep each time scanning is initiated. Once the scan count is reached, subsequent triggers will be ignored.

The timing of the scan sweeps is determined by the trigger source and the trigger timer.

During an automated test, the trigger count is the number of scan sweeps performed at each test point. \*RST sets the trigger count to 1.

**Example:** TRIG:COUN?

**Response:** 1

**Set Example:** TRIG:COUN 10

**Related Commands:**

TRIGger:SOURce

TRIGger:TIMer

INITiate

**TRIGger:ENABle <boolean>**  
**TRIGger:ENABle?**

**Description:** Enable or suspend triggering and scanning. <boolean> can be 0 (trigger suspended), OFF, 1 (trigger enabled), or ON. The default is 1.

Normally the TRIGger:ENABle setting is 1 and triggering and scanning are enabled. Setting the trigger enable to 0 temporarily suspends triggering and scanning or pauses an automated test. Scanning resumes when the TRIGger:ENABle setting is set to 1 again. While triggering is suspended the scan counter is kept at its last count, and it continues on when triggering is enabled again. If the trigger is suspended while a scan sweep is in progress, the scan sweep is cancelled.

While triggering is suspended, bit 6 of the Operation Status Condition register is set. Bit 6 of the Operation Status Event register is set when the trigger enable state changes from enabled to suspended. The trigger enable setting is automatically set to 1 when scanning is started. \*RST enables triggering.

**Example:** TRIG:ENAB?  
**Response:** 1  
**Set Example:** TRIG:ENAB ON

**Related Commands:**

INITiate:IMMediate  
TRIGger:SOURce  
STATus:OPERation:CONDition?  
STATus:OPERation:EVENT?  
TRIGger:AUTO:CONTRol

**TRIGger:SOURce <source>**  
**TRIGger:SOURce?**

**Description:** Select a trigger source for scan sweeps. <source> is the trigger source. Options are TIMer, EXTErnal, ALARm, BUS, MANual, and AUTO. The default is TIMer.

- TIMer controls scan sweeps using only the scan timer. The scan interval between scan sweeps is determined by the TRIGger:TIMer setting.
- EXTErnal enables the external trigger input. A scan sweep will be triggered when the external trigger signal is pulled low. If the external trigger remains asserted, scan sweeps will be performed at the scan interval determined by the TRIGger:TIMer setting. Scan sweeps stop and the scan timer is deactivated and reset when the external trigger is released.
- ALARm enables the alarm trigger. A scan sweep will be triggered when the alarm is active. While the alarm condition is true, scan sweeps will be performed at the scan interval determined by the TRIGger:TIMer setting. The scan timer is deactivated and reset when the alarm condition is no longer true.
- BUS enables the \*TRG command to trigger scan sweeps.
- MANual enables the front panel trigger soft-key.
- AUTO enables automated tests. Scan sweeps will be performed automatically according to the automated test settings (see the TRIGger:AUTO group of commands). An automated test may be started using the INITiate[:IMMediate] command.

\*RST sets the trigger source to TIM.

**Example:** TRIG:SOUR?  
**Response:** TIM  
**Set Example:** TRIG:SOUR BUS

**Related Commands:**

TRIGger:TIMer  
\*TRG  
INITiate:IMMediate  
TRIGger:AUTO:POINTs

*TRIGger:TIMer {<n> | INFinity}*  
*TRIGger:TIMer?*

**Description:** Set the scan interval, which determines the time between scan sweeps while either the TIMer, EXTernal, or ALARm trigger source is used. <n> is the scan interval in seconds. The range is 0 to 359999 and INFinity. The default is 0. \*RST sets the scan interval to 0.

**Example:** TRIG:TIM?  
**Response:** 10  
**Set Example:** TRIG:TIM 10

**Related Commands:**

TRIGger:SOURce  
INITiate:IMMediate

*UNIT:TEMPerature {C | CEL | F | FAR}*  
*UNIT:TEMPerature?*

**Description:** Set the temperature unit to degrees Celsius or degrees Fahrenheit. The default is C. In some regional versions of the product the Fahrenheit option may not be available — the command will be rejected and an error will be logged (502, "Temperature unit locked"). \*RST sets the temperature unit to C.

**Example:** UNIT:TEMP?  
**Response:** C  
**Set Example:** UNIT:TEMP F