UNI-T®

Instruments.uni-trend.com

SCPI Programming Manual
UT8806E 6 1/2 Desktop Multimeter

Warranty and Statement

Copyright

Brand Information

UNI-T is the registered trademark of Uni-Trend Technology (China) Co., Ltd.

Software Version

00.00.01

Software upgrades may change or add features to the product, please subscribe the UNI-T website for the latest version of the manual or contact UNI-T to upgrade your software.

Statement

- UNI-T products are protected by patents (including obtained and pending) in China and other countries and regions.
- UNI-T reserves the right to change specifications and prices.
- The information provided in this manual supersedes all previous publications.
- The information provided in this manual is subject to change without notice.
- UNI-T shall not be liable for any errors that may be contained in this manual. For any incidental or consequential damages arising out of the use or the information and deductive functions provided in this manual.
- No part of this manual shall be photocopied, reproduced or adapted without the prior written permission of UNI-T.

Product Certification

UNI-T has certified that the product conforms to China national product standard and industry product standard as well as ISO9001:2008 standard and ISO14001:2004 standard. UNI-T will go further to certificate product to meet the standard of other member of the international standards organization.

SCPI Introduction

SCPI（Standard Commands for Programmable Instruments）is a standardized instrument programming language that builds on existing standards IEEE 488.1 and IEEE 488.2 and follows the floating point rules of IEEE 754 standard, ISO 646 message exchange 7-bit encoding notation (equivalent to ASCII programming) and many other standards.

This section introduces the format, symbols, parameters, and abbreviations of the SCPI command.

Command Format

SCPI commands are tree hierarchical, including multiple subsystems, each subsystem consists of a root keyword and one or several hierarchical keywords. **Command lines usually start with a colon ":"; keywords are separated by a colon ":" and optional parameter settings follow the keywords. Command keywords are separated from the first parameter by a space. Command strings must end with a <New Line> (<NL>) character. The command line is followed by a question mark "?" usually indicates a query for this function.**

Symbol Description

The following four symbols are not part of the SCPI command and are not sent with the command, but are often used to aid in describing the parameters in the command.

- Braces { }
  The braces usually contain several optional parameters, one of which must be selected when sending the command.
  Such as the command, :DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE}

- Vertical Bar |
  The vertical bar is used to separate several optional parameters, one of which must be selected when sending the command.
  Such as the command, :DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE}

- Square Brackets [ ]
  The contents in the square brackets (command keywords) can be omitted. If a parameter is omitted, the instrument sets the parameter to its default value.
  For example, for the command :MEASure:NDUTy? [<source>], [<source>] indicates the current channel.

- Angle Brackets < >
  The parameters in angle brackets must be replaced with a valid value.
  For example, send the DISPlay:GRID:BRIGhtness <count> command as DISPlay:GRID:BRIGhtness 30.

Parameter Description

The parameter in this manual can divide into five types: Boolean, Integer, Real, Discrete and ASCII string.

- Boolean
  The parameter can set to "ON"(1) or "OFF" (0). Such as, :SYSTem:LOCK {{1 | ON} | {0 | OFF}}.

- Integer
  Unless otherwise specified, the parameter can take any integer value within the effective range.
  Note: Do not set the parameter to decimal format, otherwise it may occur error.
  Such as, < count > in the command: DISPlay:GRID:BRIGhtness <count> can take integer from 0-100.

- Real
  Unless otherwise specified, the parameter can take any integer value within the effective range.
  Such as, for CH1, <offset> in the command CHANnel1:OFFSet <offset> is take integer value.

- Discrete
  The parameter can only take some specified numbers or characters.
  Such as, the parameter in the command :DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE} can only be FULL, GRID, CROSS, NONE.

- ASCII String
  In actuality, the string parameter can contain all ASCII string sets. The string must begin and end with paired quotes; it can use single or double quotation marks. The quotation and delimiter can also be part of a string by typing it twice and not adding any characters.
  Such as, set IP: SYST:COMM:LAN:IPAD "192.168.1.10"

Shorthand Rule

All command can identify capital and small letter, if command need enter shorthand, it should be all capital letter.

Data Return

Data return is divided into single data and batch data. The single data return is the corresponding parameter type, in which the real return type is presents by the scientific notation method. The part before e retains three figure behind the decimal point, and the e part retains three figure; the batch return must be obey IEEE 488.2# string data format, '#'+ the length of character bits [fixed to one character] + ASCII valid value+ valid data+ end string ['\n']. Such as, #3123xxxxxxxxxxxxxxxxxxx\n represents 123 bytes of valid batch data return format, where '3' means that "123" occupies 3 character bits.

Note: When the returned data is invalid, it is indicated by *.

SCPI Command

IEEE488.2 Common Command

*CLS

- **Command format:**

  *CLS

- **Functional description:**

  Clear the status command.

  This command is used to delete all status registers and error queues in all register groups.

*ESE

- **Command format:**

  *ESE <enable_value>

  *ESE?

- **Functional description:**

  To activate the enable register of standard event. This command is used to enable the standard event corresponding to the register bit.

- **Return format:**

  Query returns the value of enable register.

- **For example:**

  *ESE 1 Enable the standard event corresponding to register bit0.

  *ESE? Query returns 1, enable the standard event corresponding to register bit0.

*ESR?

- **Command format:**

  *ESR?

- **Functional description:**

  To query the event status register of the standard event register set.

- **Return format:**

  Query returns the value in the standard event status register.

- **For example:**

*ESR?	Query returns 1, the current value of the standard event status register is 1.

*IDN?

- Command format:

  *IDN?

- Functional description:

  To query the manufacture name, model, product serial number and software version.

- Return format:

  Manufacture name, model, product serial number and software version.

- For example:

  UNI-T UT8805A, UT1A13460051200, V0.01.0000

*RST

- Command format:

  *RST

- Functional description:

  To restore to factory settings and clear the entire error message, send and receive queue buffers.

*OPC

- Command format:

  *OPC

  *OPC?

- Functional description:

  To force the current instruction that has been executed completion flag position 1.

- Return format:

  Query returns whether the execution operation of the currently sent instruction is completed or not, 1 indicates completion and 0 indicates incompletion.

- For example:

*OPC	Flag the completed instruction as position 1.
*OPC?	Query returns 1, it indicates that the current instruction has been completed, otherwise not completed.

*SRE

- **Command format:**

  *SRE <enable_value>

  *SRE?

- **Functional description:**

  To enable service request command. This command is used to set the value for service request register, and the range is 0~255.

- **Return format:**

  Query returns the value of service request register.

- **For example:**

*SRE 1	Set the value of service request register to 1.
*SRE?	Query returns 1, it indicates the value of current request enable register is 1.

*STB?

- **Command format:**

  *STB?

- **Functional description:**

  To read the value of status byte register, and the value of the status byte register will be cleared after reading.

- **Return format:**

  The current value of status byte register.

- **For example:**

  Omissible.

*TST?

- **Command format:**

  *TST?

- **Functional description:**

  To enable the instrument's self-inspection and return the result.

- **Return format:**

  The result of instrument's self-inspection.

- **For example:**

*TST?	+0 (pass), +1 (one or more tests failed).

*WAI

- **Command format:**

  *WAI

- **Functional description:**

  Wait for the operation to complete.

*PSC

- **Command format:**

  *PSC {0 | 1}

  *PSC?

- **Functional description:**

  To set up to allow (1) or disallow (0) clearing of certain enable registers on power-up, involving the Suspicious Data Register, Standard Operation Register, Status Byte Condition Register, and Standard Event Enable Register.

- **Return format:**

  Query returns the configuration of the power-up clear.

- **For example:**

*PSC 0	Disable the power-up clearing of affected registers
*PSC?	Query returns 0, it indicates that that power-up clearing of affected registers is currently disabled

*TRG

- **Command format:**

  *TRG

- **Functional description:**

  This command is used to software trigger measurement.

- **Return format:**

  Nothing.

- **For example:**

*TRG	Software trigger measurement

CONFigure Subsystem

It is used to configure the measurement range of voltage, current, capacitance, resistance, diode, frequency, period, temperature and obtain the current measurement range of the instrument.

Query Command of Measuring Configuration

:CONFigure

- **Command format:**

  CONFigure?

- **Functional description:**

  Query the current measurement scale and range (temperature probe scale).

- **Return format:**

  Query returns the name of measuring scale (such as CURR:AC) and range.

- **For example:**

  CONFigure? Query returns: VOLT:DC +2.00000000E+01

Measuring Configuration Command

:CONFigure:VOLTage:DC

- **Command format:**

  CONFigure:VOLTage:DC [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for DC voltage measurement.

  The range and resolution can be set by the input parameters, and the unit is V. The range can be set to {200 mV|2 V|20 V|200 V|1000 V |AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger	POS

		polarity	
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		
Input impedance	AUTO		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:VOLT:DC 200,0.001 Set all measurement parameters and trigger parameters to their default values, set the manual range to 200V and resolution to 1mV for DC voltage measurement.

:CONFigure:CURRent:DC

- **Command format:**

  CONFigure:CURRent:DC [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for DC current measurement.

  The range and resolution can be set by the input parameters, and the unit is A. The range can be set to {200uA|2mA|20mA|200mA|2A|10A|AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:CURR:DC Set all measurement parameters and trigger parameters to their default values, that is automatic range.

  Set it for DC current measurement.

CONFigure:CURR:DC 2,0.01 Set all measurement parameters and trigger parameters to their default values, set the manual range to 2A and resolution to 10MA for DC voltage measurement.

:CONFigure:VOLTage:AC

- **Command format:**

  CONFigure:VOLTage:AC [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for AC voltage measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is V. The range can be set to {200 mV|2 V|20 V|200 V|750V |AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:VOLT:AC 20 Set all measurement parameters and trigger parameters to their default values, Set the manual range to 20V for AC voltage measurement.

:CONFigure:CURRent:AC

- **Command format:**

  CONFigure:CURRent:AC [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for AC current measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is A. The range can be set to {2mA|20mA|200mA|2A|10A|AUTO} and the default is AUTO.

Measurement Parameter	Trigger Parameter

Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:CURR:AC 0.02 Set all measurement parameters and trigger parameters to their default values, Set the manual range to 20mA for AC current measurement.

:CONFigure:RESistance

- **Command format:**

  CONFigure:RESistance [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for 2-wire resistance measurement. The range and resolution can be set by the input parameters, and the unit is Ω. The range can be set to {200Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|10 MΩ|100 MΩ|AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

Nothing.

- For example:

CONFigure:RES 200,1 Set all measurement parameters and trigger parameters to their default values, Set the manual range to 200Ω and the resolution to 1Ω for 2-wire resistance measurement.

:CONFigure:FRESistance

- Command format:

CONFigure:FRESistance [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- Functional description:

Set all measurement parameters and trigger parameters to their default values for 4-wire resistance measurement. The range and resolution can be set by the input parameters, and the unit is Ω. The range can be set to {200Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- Return format:

Nothing.

- For example:

CONFigure:FRES 2000,1 Set all measurement parameters and trigger parameters to their default values, Set the manual range to 2kΩ and the resolution to 1Ω for 4-wire resistance measurement.

:CONFigure:CAPacitance

- Command format:

CONFigure:CAPacitance [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- Functional description:

Set all measurement parameters and trigger parameters to their default values for capacitance measurement. The range and resolution can be set by the input parameters (actually, it can be omitted),

and the unit is F. The range can be set to {2nF|20nF|200nF|2uF |20uF| 200uF|2mF |AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:CAP 0.00002 Set all measurement parameters and trigger parameters to their default values, Set the manual range to 20uF for capacitance measurement.

:CONFigure:TEMPerature

- **Command format:**

  CONFigure:TEMPerature [{<probe_type>|DEF}[,{<type>|DEF}[,1[,{<resolution>|MIN|MAX|DEF}]]]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for temperature measurement. The probe setting, probe type and resolution can be set by the input parameters. The probe can select to { RTD|FRTD|THERmistor|FTHermistor|TCouple }, and the default is RTD. Only TCouple probe can select the type to { J|K|E|T|N|R|S|B }. Other probe types are available by default.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative	OFF	Trigger delay	AUTO

value function			
Dual display mode	OFF		
Nominal resistance	100Ω		
Compensation temperature	0		
Fixed reference junction temperature	0		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:TEMP TCouple,J Use TCouple probe,J type thermocouple for temperature measurement

:CONFigure:CONTinuity

- **Command format:**

  CONFigure:CONTinuity

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for continuity measurement.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		
Threshold	30Ω		
Continuity beeper	ON		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:CONT Set all measurement parameters and trigger parameters to their default values for continuity measurement.

:CONFigure:DIODe

- **Command format:**

  CONFigure:DIODe

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for diode measurement.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:DIOD Set all measurement parameters and trigger parameters to their default values for diode measurement.

:CONFigure:FREQuency

- **Command format:**

  CONFigure:FREQuency [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for frequency measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is V. The range can be set to {200 mV|2 V|20 V|200 V|750V|AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1

Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:FREQ 20 Set all measurement parameters and trigger parameters to their default values, Set the manual range to 20V for frequency measurement.

:CONFigure:PERiod

- **Command format:**

  CONFigure:PERiod [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for period measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is V. The range can be set to {200 mV|2 V|20 V|200 V|750V|AUTO} and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  CONFigure:PER 20 Set all measurement parameters and trigger parameters to their default values, Set the manual range to 20V for period measurement.

MEASure Subsystem

It is used to configure the measurement range of voltage, current, capacitance, resistance, diode, frequency, period, temperature and use a new range to measure them and return the measured results immediately.

Trigger Immediately Measuring Command

:MEASure:VOLTage:DC

- **Command format:**

  MEASure:VOLTage:DC? [{<range>|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for DC voltage measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters, and the units are V. The range can be set to {200 mV|2 V|20 V|200 V|1000 V |AUTO} and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		
Input impedance	AUTO		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:VOLT:DC? 200,0.001 Set the range to 200V and the resolution to 1mV in DC voltage measurement and return the measured result "+2.43186951E-02".

:MEASure:CURRent:DC

- **Command format:**

  MEASure:CURRent:DC? [{<range>|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for DC current measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters, and the unit is A. The range can be set to {200uA|2mA|20mA|200mA|2A|10A|AUTO}, and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:CURR:DC? 2,0.1 Set the range to 2A and the resolution to 100mA in DC current measurement and return the measured result "+2.43186951E-02".

:MEASure:VOLTage:AC

- **Command format:**

  MEASure:VOLTage:AC? [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for AC voltage measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is V. The range can be set to {200 mV|2 V|20 V|200 V|750V |AUTO}, and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger	POS

		polarity	
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:VOLT:AC? 20 Set the range to 20V in AC voltage measurement and return the measured result "+2.43186951E-02".

:MEASure:CURRent:AC

- **Command format:**

  MEASure:CURRent:AC? [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for AC current measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is A. The range can be set to {2mA|20mA|200mA|2A|10A|AUTO}, and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:CURR:AC? 0.02 Set the range to 20mA in AC current mode and return the measured result "+0.85491724E-04".

:MEASure:RESistance

- **Command format:**

MEASure:RESistance? [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for 2-wire resistance measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters, and the unit is Ω. The range can be set to {200Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|10 MΩ|100 MΩ|AUTO}, and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:RES? 200,1 Set the range to 200Ω and the resolution to 1Ω in 2-wire resistance measurement and return the measured result "+2.43186951E-02".

:MEASure:FRESistance

- **Command format:**

  MEASure:FRESistance? [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for 4-wire resistance measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters, and the unit is Ω. The range can be set to {200Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|AUTO}, and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1

Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Nothing.

- **For example:**

  MEASure:FRES? 2000,1 Set the range to 2 kΩand the resolution to 1Ω in 4-wire resistance measurement and return the measured result "+2.43186951E-02".

:MEASure:CAPacitance

- **Command format:**

  MEASure:CAPacitance? [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for capacitance measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is F. The range can be set to {2nF|20nF|200nF|2uF |20uF| 200uF|2mF |AUTO}, and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:CAP? 0.00002 Set the range to 20uF in capacitance measurement and return the measured result "+0.14034846E-07".

:MEASure:TEMPerature

- **Command format:**

  MEASure:TEMPerature? [{<probe_type>|DEF}[,{<type>|DEF}[,1[,{<resolution>|MIN|MAX|DEF}]]]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for temperature measurement and immediately trigger the measurement. The probe setting, probe type and resolution can be set by the input parameters. The probe can select to { RTD|FRTD|THERmistor|FTHermistor|TCouple }, and the default is RTD. Only TCouple probe can select to { J|K|E|T|N|R|S|B }. Other probe types are available by default. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		
nominal resistance	100Ω		
Compensation temperature	0		
Fixed reference junction temperature	0		

- **Return format:**

  Measured result, the unit is C/F/K. It set by the command UNIT:TEMPerature.

- **For example:**

  MEASure:TEMP? TCouple,J Use TCouple probe of J type in temperature mode and return the measured result "-2.00000000E+02".

:MEASure:CONTinuity

- **Command format:**

  MEASure:CONTinuity?

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for continuity measurement and immediately trigger the measurement.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		
Threshold	30Ω		
Continuity beeper	ON		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:CONT? Generate the continuity measurement for one time and return the measured result "+9.84739065E+02".

:MEASure:DIODe

- **Command format:**

  MEASure:DIODe?

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for diode measurement and immediately trigger the measurement. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:DIOD? Generate the diode measurement for one time and return the measured result "+9.84733701E-01".

:MEASure:FREQuency

- **Command format:**

  MEASure:FREQuency? [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for frequency measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters (actually, it can be omitted). The range can be set to {200 mV|2 V|20 V|200 V|750V|AUTO}, and the default is AUTO.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:FREQ? 20 Set the range to 20V in frequency mode and return the measured result "+2.43186951E-02".

:MEASure:PERiod

- **Command format:**

  MEASure:PERiod? [{<range>|AUTO|MIN|MAX|DEF}[,{<resolution>|MIN|MAX|DEF}]]

- **Functional description:**

  Set all measurement parameters and trigger parameters to their default values for period measurement and immediately trigger the measurement. The range and resolution can be set by the input parameters (actually, it can be omitted), and the unit is V. The range can be set to {200 mV|2 V|20 V|200 V|750V|AUTO},

and the default is AUTO. The measured result will be returned directly.

Measurement Parameter		Trigger Parameter	
Parameter	Default	Parameter	Default
Statistical counting	OFF	Measuring trigger sampling count	1
Limits measurement	OFF	Measuring trigger time	1
Calibration function	OFF	External trigger polarity	POS
[All scales] relative value function	OFF	Trigger delay	AUTO
Dual display mode	OFF		

- **Return format:**

  Measured results.

- **For example:**

  MEASure:PER? 20 Set the range to 20V in period mode and return the measured result "+2.43186951E-02".

SENSe Subsystem

It is used to configure the measurement range of voltage, current, capacitance, resistance, diode, frequency, period, temperature and switch the measuring range.

Configuration Command of Measurement Mode

:[SENSe:]FUNCtion[:ON]

- **Command format:**

  [SENSe:]FUNCtion[:ON]"<function>"

  [SENSe:]FUNCtion[:ON]?

- **Functional description:**

  Select the measurement mode, it can set to CONTinuity/CURRent:AC/CURRent:DC/DIODe/FREQuency/FRESistance /PERiod/RESistance/TEMPerature/VOLTage:AC/VOLTage:DC/CAPacitance.

- **Return format:**

  Query returns the currently measurement mode.

- For example:

FUNC "VOLT:AC"	Select the AC voltage measurement
FUNC?	Query returns: VOLT: AC

Range Setting Command

:[SENSe:]VOLTage:DC:RANGe

- Command format:

  [SENSe:]VOLTage:DC:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]VOLTage:DC:RANGe?

- Functional description:

  Select the measuring range by manual. The range can be set to {200 mV|2 V|20 V|200 V|1000 V|AUTO}, and the default is AUTO. The unit of parameter reference is V.

- Return format:

  Query returns the currently measuring range and the unit is V.

- For example:

VOLT:DC:RANGe 20	Set the manual range to 20V.
VOLT:DC:RANGe?	Query returns: +2.00000000E+01

:[SENSe:]CURRent:DC:RANGe

- Command format:

  [SENSe:]CURRent:DC:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]CURRent:DC:RANGe?

- Functional description:

  Select the measuring range by manual. The range can be set to {200uA|2mA|20mA|200mA|2A|10A|AUTO} and the default is AUTO. The unit of parameter reference is A.

- Return format:

  Query returns the currently measured range and the unit is A.

- For example:

CURRent:DC:RANGe 0.02	Set the manual range to 20mA.
CURRent:DC:RANGe?	Query returns: +0.20000000E-01

:[SENSe:]VOLTage:AC:RANGe

- Command format:

  [SENSe:]VOLTage:AC:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]VOLTage:AC:RANGe?

- Functional description:

  Select the measuring range by manual. The range can be set to {200 mV|2 V|20 V|200 V|750 V|AUTO}, and the

default is AUTO.The unit of parameter reference is V.

- Return format:

  Query returns the currently measured range and the unit is V.

- For example:

VOLT:AC:RANGe 20	Set the manual range to 20V.
VOLT:AC:RANGe?	Query returns: +2.00000000E+01

:[SENSe:]CURRent:AC:RANGe

- Command format:

  [SENSe:]CURRent:AC:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]CURRent:AC:RANGe?

- Functional description:

  Select the measuring range by manual. The range can be set to {2mA|20mA|200mA|2A|10A|AUTO}, and the default is AUTO. The unit of parameter reference is A.

- Return format:

  Query returns the currently measured range and the unit is A.

- For example:

CURRent:AC:RANGe 0.02	Set the manual range to 20mA.
CURRent:AC:RANGe?	Query returns: +0.20000000E-01

:[SENSe:]RESistance:RANGe

- Command format:

  [SENSe:]RES:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]RES:RANGe?

- Functional description:

  Select the measuring range by manual. The range can be set to {200Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|10 MΩ|100 MΩ|AUTO}, and the default is AUTO.The unit of parameter reference is Ω.

- Return format:

  Query returns the currently measured range and the unit isΩ.

- For example:

RES:RANGe 2000	Set the manual range to 2 kΩ.
RES:RANGe?	Query returns: +2.00000000E+03

:[SENSe:]FRESistance:RANGe

- Command format:

  [SENSe:]FRES:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]FRES:RANGe?

- Functional description:

  Select the measuring range by manual. The range can be set to {200Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|AUTO}, and the default is AUTO.The unit of parameter reference is Ω.

- **Return format:**

  Query returns the currently measured range and the unit is Ω.

- **For example:**

  FRES:RANGe 2000 Set the manual range to 2 kΩ.

  FRES:RANGe? Query returns: +2.00000000E+03

:[SENSe:]CAPacitance:RANGe

- **Command format:**

  [SENSe:]CAPacitance:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]CAPacitance:RANGe?

- **Functional description:**

  Select the measuring range by manual.

  The range can be set to {2nF|20nF|200nF|2uF|20uF|200uF|2mF|AUTO}, and the default is AUTO. The unit of parameter reference is F.

- **Return format:**

  Query returns the currently measuring range and the unit is F.

- **For example:**

  CAPacitance:RANGe 2E-6 Set the manual range to 2uF.

  CAPacitance:RANGe? Query returns: +2.00000000E-06

:[SENSe:]FREQuency:VOLTage:RANGe

- **Command format:**

  [SENSe:]FREQ:VOLTage:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]FREQ:VOLTage:RANGe?

- **Functional description:**

  Select the measuring range by manual. The range can be set to {200 mV|2 V|20 V|200 V|750V|AUTO}, and the default is AUTO. The unit of parameter reference is V.

- **Return format:**

  Query returns the currently measured range and the unit is V.

- **For example:**

  FREQ:VOLT:RANG 20 Set the manual range to 20V.

  FREQ:VOLT:RANG? Query returns: +2.00000000E+01

:[SENSe:]PERiod:VOLTage:RANGe

- **Command format:**

  [SENSe:]PERiod:VOLTage:RANGe {<range>|MIN|MAX|DEF}

  [SENSe:]PERiod:VOLTage:RANGe?

- **Functional description:**

  Select the measuring range by manual. The range can be set to {200 mV|2 V|20 V|200 V|750V|AUTO}, and the default is AUTO.The unit of parameter reference is V.

- Return format:

  Query returns the currently measured range and the unit is V.

- For example:

  PER:VOLT:RANGe 20 Set the manual range to 20V.

  PER:VOLT:RANGe? Query returns: +2.00000000E+01

Switch Command of Automatic Range

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

- Command format:

  [SENSe:]VOLTage:DC:RANGe:AUTO {OFF|ON}

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

- Functional description:

  Disable or enable the automatic range.

- Return format:

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- For example:

  VOLT:DC:RANGe:AUTO ON Turn on the automatic range.

  VOLT:DC:RANGe:AUTO? Query returns: 1

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

- Command format:

  [SENSe:]CURRent:DC:RANGe:AUTO {OFF|ON}

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

- Functional description:

  Disable or enable the automatic range.

- Return format:

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- For example:

  CURRent:DC:RANGe:AUTO ON Turn on the automatic range.

  CURRent:DC:RANGe:AUTO? Query returns: 1

:[SENSe:]VOLTage:AC:RANGe:AUTO

- Command format:

  [SENSe:]VOLTage:AC:RANGe:AUTO {OFF|ON}

  [SENSe:]VOLTage:AC:RANGe:AUTO?

- Functional description:

  Disable or enable the automatic range.

- **Return format:**

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- **For example:**

  VOLT:AC:RANGe:AUTO ON Turn on the automatic range.

  VOLT:AC:RANGe:AUTO? Query returns: 1

:[SENSe:]CURRent:AC:RANGe:AUTO

- **Command format:**

  [SENSe:]CURRent:AC:RANGe:AUTO {OFF|ON}

  [SENSe:]CURRent:AC:RANGe:AUTO?

- **Functional description:**

  Disable or enable the automatic range.

- **Return format:**

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- **For example:**

  CURRent:AC:RANGe:AUTO ON Turn on the automatic range.

  CURRent:AC:RANGe:AUTO? Query returns: 1

:[SENSe:]RESistance:RANGe:AUTO

- **Command format:**

  [SENSe:]RESistance:RANGe:AUTO {OFF|ON}

  [SENSe:]RESistance:RANGe:AUTO?

- **Functional description:**

  Disable or enable the automatic range.

- **Return format:**

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- **For example:**

  RES:RANGe:AUTO ON Turn on the automatic range.

  RES:RANGe:AUTO? Query returns: 1

:[SENSe:]FRESistance:RANGe:AUTO

- **Command format:**

  [SENSe:]FRESistance:RANGe:AUTO {OFF|ON}

  [SENSe:]FRESistance:RANGe:AUTO?

- **Functional description:**

  Disable or enable the automatic range.

- **Return format:**

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- **For example:**

  FRES:RANGe:AUTO ON Turn on the automatic range.

FRES:RANGe:AUTO? Query returns: 1

:[SENSe:]CAPacitance:RANGe:AUTO

- **Command format:**

  [SENSe:]CAPacitance:RANGe:AUTO {OFF|ON}

  [SENSe:]CAPacitance:RANGe:AUTO?

- **Functional description:**

  Disable or enable the automatic range.

- **Return format:**

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- **For example:**

  CAPacitance:RANGe:AUTO ON Turn on the automatic range.

  CAPacitance:RANGe:AUTO? Query returns: 1

:[SENSe:]FREQuency:VOLTage:RANGe:AUTO

- **Command format:**

  [SENSe:]FREQuency:VOLTage:RANGe:AUTO {OFF|ON}

  [SENSe:]FREQuency:VOLTage:RANGe:AUTO?

- **Functional description:**

  Disable or enable the automatic range.

- **Return format:**

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- **For example:**

  FREQ:VOLT:RANGe:AUTO ON Turn on the automatic range.

  FREQ:VOLT:RANGe:AUTO? Query returns: 1

:[SENSe:]PERiod:VOLTage:RANGe:AUTO

- **Command format:**

  [SENSe:]PERiod:VOLTage:RANGe:AUTO {OFF|ON}

  [SENSe:]PERiod:VOLTage:RANGe:AUTO?

- **Functional description:**

  Disable or enable the automatic range.

- **Return format:**

  Query returns the state of automatic range, 0 (OFF) or 1 (ON).

- **For example:**

  PER:VOLT:RANGe:AUTO ON Turn on the automatic range.

  PER:VOLT:RANGe:AUTO? Query returns: 1

Resolution Setting Command

:[SENSe:]VOLTage:DC:RESolution

- Command format:

  [SENSe:]VOLTage:DC:RESolution {<resolution>|MINimum|MAXimum}

  [SENSe:]VOLTage:DC:RESolution? [MINimum|MAXimum]

- Functional description:

  Set the resolution of the current range for DC voltage mode, the maximum value (the worst resolution), which is equivalent to 0.006 PLC; the minimum value (the optimal resolution), which is equivalent to 100 PLC; and the default value is equivalent to 10 PLC. In order to realize the line noise suppression, the instrument selects the appropriate integral time to execute the measurement according to the set resolution.

Accuracy	Resolution
0.03 ppm	0.03ppm * range(V)
0.1 ppm	0.1ppm * range(V)
0.3 ppm	0.3ppm * range(V)
0.7 ppm	0.7ppm * range(V)
1.5 ppm	1.5ppm * range(V)
3 ppm	3ppm * range(V)
6 ppm	6ppm * range(V)

- Return format:

  Query returns the accuracy supported by the current range measurement resolution.

- For example:

  VOLT:DC:RES 3E-5 Set the resolution to 3E-5.

  VOLT:DC:RES? Query returns: +1.00000000E-07

:[SENSe:]CURRent:DC:RESolution

- Command format:

  [SENSe:]CURRent:DC:RESolution {<resolution>|MINimum|MAXimum}

  [SENSe:]CURRent:DC:RESolution? [MINimum|MAXimum]

- Functional description:

  Set the resolution of the current range for DC current mode, the maximum value (the worst resolution), which is equivalent to 0.006 PLC; the minimum value (the optimal resolution), which is equivalent to 100 PLC; and the default value is equivalent to 10 PLC. In order to realize the line noise suppression, the instrument selects the appropriate integral time to execute the measurement according to the set resolution.

Accuracy	Resolution
0.03 ppm	0.03ppm * range(A)
0.1 ppm	0.1ppm * range(A)
0.3 ppm	0.3ppm * range(A)
0.7 ppm	0.7ppm * range(A)
1.5 ppm	1.5ppm * range(A)
3 ppm	3ppm * range(A)
6 ppm	6ppm * range(A)

- Return format:

  Query returns the accuracy supported by the current range measurement resolution.

- For example:

  CURR:DC:RES 3E-5 Set the resolution to 3E-5

  CURR:DC:RES? Query returns: +1.00000000E-07

:[SENSe:]VOLTage:AC:RESolution

- Command format:

  [SENSe:]VOLTage:AC:RESolution {<resolution>|MINimum|MAXimum}

  [SENSe:]VOLTage:AC:RESolution? [MINimum|MAXimum]

- Functional description:

  Set the resolution of current range in AC voltage mode (actually, it will be ignored).

- Return format:

  Query returns the accuracy supported by the current range measurement resolution.

- For example:

  VOLT:AC:RES 3E-5 Set the resolution to 3E-5

  VOLT:AC:RES? Query returns: +1.00000000E-07

:[SENSe:]CURRent:AC:RESolution

- Command format:

  [SENSe:]CURRent:AC:RESolution {<resolution>|MINimum|MAXimum}

  [SENSe:]CURRent:AC:RESolution? [MINimum|MAXimum]

- Functional description:

  Set the resolution of current range in AC current mode (actually, it will be ignored).

- Return format:

  Query returns the accuracy supported by the current range measurement resolution.

- For example:

  CURR:AC:RES 3E-5 Set the resolution to 3E-5

  CURR:AC:RES? Query returns: +1.00000000E-07

:[SENSe:]RESistance:RESolution

- **Command format:**

  [SENSe:]RESistance:RESolution {<resolution>|MINimum|MAXimum}

  [SENSe:]RESistance:RESolution? [MINimum|MAXimum]

- **Functional description:**

  Set the resolution of the current range for the 2-wire resistance mode, the maximum value (the worst resolution), which is equivalent to 0.006 PLC; the minimum value (the optimal resolution), which is equivalent to 100 PLC; and the default value is equivalent to 10 PLC. In order to realize the line noise suppression, the instrument selects the appropriate integral time to execute the measurement according to the set resolution.

Accuracy	Resolution
0.03 ppm	0.03ppm * range(A)
0.1 ppm	0.1ppm * range(A)
0.3 ppm	0.3ppm * range(A)
0.7 ppm	0.7ppm * range(A)
1.5 ppm	1.5ppm * range(A)
3 ppm	3ppm * range(A)
6 ppm	6ppm * range(A)

- **Return format:**

  Query returns the accuracy supported by the current range measurement resolution.

- **For example:**

  RES:RES 3E-5 Set the resolution to 3E-5

  RES:RES? Query returns: +1.00000000E-07

:[SENSe:]FRESistance:RESolution

- **Command format:**

  [SENSe:]FRESistance:RESolution {<resolution>|MINimum|MAXimum}

  [SENSe:]FRESistance:RESolution? [MINimum|MAXimum]

- **Functional description:**

  Set the resolution of the current range for the 4-wire resistance mode, the maximum value (the worst resolution), which is equivalent to 0.006 PLC; the minimum value (the optimal resolution), which is equivalent to 100 PLC; and the default value is equivalent to 10 PLC. In order to realize the line noise suppression, the instrument selects the appropriate integral time to execute the measurement according to the set resolution.

Accuracy	Resolution
0.03 ppm	0.03ppm * range(A)
0.1 ppm	0.1ppm * range(A)

0.3 ppm	0.3ppm * range(A)
0.7 ppm	0.7ppm * range(A)
1.5 ppm	1.5ppm * range(A)
3 ppm	3ppm * range(A)
6 ppm	6ppm * range(A)

- **Return format:**

  Query returns the accuracy supported by the current range measurement resolution.

- **For example:**

  RES:RES 3E-5 Set the resolution to 3E-5

  RES:RES? Query returns: 3E-6

Integral Time Setting Command

:[SENSe:]VOLTage:DC:NPLC

- **Command format:**

  [SENSe:]VOLTage:DC:NPLC {<PLCs>|MINimum|MAXimum}

  [SENSe:]VOLTage:DC:NPLC?

- **Functional description:**

  Set the integral time for the DC voltage mode, it can set to {0.006, 0.02, 0.06, 0.2, 1, 10, 100}.

- **Return format:**

  Query returns the integral time.

- **For example:**

  VOLT:DC:NPLC 0.2 Set the integral time to 0.2 PLC

  VOLT:DC:NPLC? Query returns: +0.20000000E+00

:[SENSe:]CURRent:DC:NPLC

- **Command format:**

  [SENSe:]CURRent:DC:NPLC {<PLCs>|MINimum|MAXimum}

  [SENSe:]CURRent:DC:NPLC?

- **Functional description:**

  Set the sampling rate for the DC voltage mode, it can set to {0.006, 0.02, 0.06, 0.2, 1, 10, 100}.

- **Return format:**

  Query returns the integral time.

- **For example:**

  CURRent:DC:NPLC 0.2 Set the integral time to 0.2 PLC

  CURRent:DC:NPLC? Query returns: +0.20000000E+00

:[SENSe:]RESistance:NPLC

- Command format:

  [SENSe:]RESistance:NPLC {<PLCs>|MINimum|MAXimum}

  [SENSe:]RESistance:NPLC?

- Functional description:

  Set the sampling rate for the 2-wire resistance measurement, it can set to {0.006, 0.02, 0.06, 0.2, 1, 10, 100}.

- Return format:

  Query returns the integral time.

- For example:

  RES:NPLC 0.2 Set the integral time to 0.2 PLC

  RES:NPLC? Query returns: +0.20000000E+00

:[SENSe:]FRESistance:NPLC

- Command format:

  [SENSe:]FRESistance:NPLC {<PLCs>|MINimum|MAXimum}

  [SENSe:]FRESistance:NPLC?

- Functional description:

  Set the sampling rate for the 4-wire resistance measurement, it can set to {0.006, 0.02, 0.06, 0.2, 1, 10, 100}.

- Return format:

  Query returns the integral time.

- For example:

  FRES:NPLC 0.2 Set the integral time to 0.2 PLC

  FRES:NPLC? Query returns: +0.20000000E+00

Switch Command of Relative Value

:[SENSe:]VOLTage:DC:NULL[:STATe]

- Command format:

  [SENSe:]VOLTage:DC:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]VOLTage:DC:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the DC voltage measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  VOLT:DC:NULL:STAT ON Enable the relative value function of the DC voltage measurement mode.

  VOLT:DC:NULL:STAT? Query returns: 1

:[SENSe:]CURRent:DC:NULL[:STATe]

- Command format:

  [SENSe:]CURRent:DC:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]CURRent:DC:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the DC current measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  CURR:DC:NULL:STAT ON Enable the relative value function of the DC current measurement mode.

  CURR:DC:NULL:STAT? Query returns: 1

:[SENSe:]VOLTage:AC:NULL[:STATe]

- Command format:

  [SENSe:]VOLTage:AC:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]VOLTage:AC:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the AC voltage measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  VOLT:AC:NULL:STAT ON Enable the relative value function of the AC voltage measurement mode.

  VOLT:AC:NULL:STAT? Query returns: 1

:[SENSe:]CURRent:AC:NULL[:STATe]

- Command format:

  [SENSe:]CURRent:AC:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]CURRent:AC:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the AC current measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  CURR:AC:NULL:STAT ON Enable the relative value function of the AC current measurement mode.

  CURR:AC:NULL:STAT? Query returns: 1

:[SENSe:]RESistance:NULL[:STATe]

- Command format:

[SENSe:]RESistance:NULL[:STATe]{ON|1|OFF|0}

[SENSe:]RESistance:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the 2-wire resistance measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  RES:NULL:STAT ON Enable the relative value function of the 2-wire resistance measurement mode.

  RES:NULL:STAT? Query returns: 1

:[SENSe:]FRESistance:NULL[:STATe]

- Command format:

  [SENSe:]FRESistance:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]FRESistance:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the 4-wire resistance measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  FRES:NULL:STAT ON Enable the relative value function of the 4-wire resistance measurement mode.

  FRES:NULL:STAT? Query returns: 1

:[SENSe:]CAPacitance:NULL[:STATe]

- Command format:

  [SENSe:]CAPacitance:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]CAPacitance:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the capacitance measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  CAP:NULL:STAT ON Enable the relative value function of the capacitance measurement mode.

  CAP:NULL:STAT? Query returns: 1

:[SENSe:]TEMPerature:NULL[:STATe]

- Command format:

  [SENSe:]TEMPerature:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]TEMPerature:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the temperature measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 01 (ON).

- For example:

  TEMP:NULL:STAT ON Enable the relative value function of the temperature measurement mode.

  TEMP:NULL:STAT? Query returns: 1

:[SENSe:]FREQuency:NULL[:STATe]

- Command format:

  [SENSe:]FREQuency:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]FREQuency:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the frequency measurement mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  FREQ:NULL:STAT ON Enable the relative value function of the frequency measurement mode.

  FREQ:NULL:STAT? Query returns: 1

:[SENSe:]PERiod:NULL[:STATe]

- Command format:

  [SENSe:]PERiod:NULL[:STATe]{ON|1|OFF|0}

  [SENSe:]PERiod:NULL[:STATe]?

- Functional description:

  Enable or disable the relative value function of the period mode. After the relative value function is enabled, the final measured value = the actual measured value – relative value.

- Return format:

  Query returns the state of relative value function, 0 (OFF) or 1 (ON).

- For example:

  PER:NULL:STAT ON Enable the relative value function of the period mode.

  PER:NULL:STAT? Query returns: 1

Relative Value Setting Command

:[SENSe:]VOLTage:DC:NULL:VALue

- Command format:

  [SENSe:]VOLTage:DC:NULL:VALue {<value>|MIN|MAX|DEF}

[SENSe:]VOLTage:DC:NULL:VALue?

- Functional description:

  Set the relative value for the DC voltage mode. The unit is V, and the range is -1200V ~ +1200 V.

- Return format:

  Query returns the relative value and the unit is V.

- For example:

VOLT:DC:NULL:VAL 1	Set the relative value to 1V
VOLT:DC:NULL:VAL?	Query returns: +1.00000000E+00

:[SENSe:]CURRent:DC:NULL:VALue

- Command format:

  [SENSe:]CURRent:DC:NULL:VALue {<value>|MIN|MAX|DEF}

  [SENSe:]CURRent:DC:NULL:VALue?

- Functional description:

  Set the relative value for the DC current mode. The range is -12A ~ 12A. The unit of parameter reference is A.

- Return format:

  Query returns the relative value and the unit is A.

- For example:

CURR:DC:NULL:VAL 0.1	Set the relative value to 100mA
CURR:DC:NULL:VAL?	Query returns: +1.00000000E-01

:[SENSe:]VOLTage:AC:NULL:VALue

- Command format:

  [SENSe:]VOLTage:AC:NULL:VALue {<value>|MIN|MAX|DEF}

  [SENSe:]VOLTage:AC:NULL:VALue?

- Functional description:

  Set the relative value for the AC voltage mode. The unit is V, and the range is -1200V ~ +1200 V.

- Return format:

  Query returns the relative value and the unit is V.

- For example:

VOLT:AC:NULL:VAL 0.1	Set the relative value to 100mV
VOLT:AC:NULL:VAL?	Query returns: +1.00000000E-01

:[SENSe:]CURRent:AC:NULL:VALue

- Command format:

  [SENSe:]CURRent:AC:NULL:VALue {<value>|MIN|MAX|DEF}

  [SENSe:]CURRent:AC:NULL:VALue?

- Functional description:

  Set the relative value for the AC current mode. The range is -12A ~ 12A. The unit of parameter reference is A.

- Return format:

  Query returns the relative value and the unit is A.

- For example:

CURR:AC:NULL:VAL 0.1	Set the relative value to 100mA
CURR:AC:NULL:VAL?	Query returns: +1.00000000E-01

:[SENSe:]RESistance:NULL:VALue

- Command format:

[SENSe:]RESistance:NULL:VALue {<value>|MIN|MAX|DEF}

[SENSe:]RESistance:NULL:VALue?

- Functional description:

Set the relative value for the 2-wire resistance mode. The unit is Ω, and the range is 0~120mΩ.

- Return format:

Query returns the relative value and the unit is Ω.

- For example:

RES:NULL:VAL 0.1	Set the relative value to 100mΩ
RES:NULL:VAL?	Query returns: +1.00000000E-01

:[SENSe:]FRESistance:NULL:VALue

- Command format:

[SENSe:]FRESistance:NULL:VALue {<value>|MIN|MAX|DEF}

[SENSe:]FRESistance:NULL:VALue?

- Functional description:

Set the relative value for the 4-wire resistance mode. The unit is Ω, and the range is 0~2.4mΩ.

- Return format:

Query returns the relative value and the unit is Ω.

- For example:

FRES:NULL:VAL 0.1	Set the relative value to 100mΩ
FRES:NULL:VAL?	Query returns: +1.00000000E-01

:[SENSe:]CAPacitance:NULL:VALue

- Command format:

[SENSe:]CAPacitance:NULL:VALue {<value>|MIN|MAX|DEF}

[SENSe:]CAPacitance:NULL:VALue?

- Functional description:

Set the relative value for the capacitance mode. The unit is F, and the range is 0~2.4mF.

- Return format:

Query returns the relative value and the unit is F.

- For example:

CAP:NULL:VAL 1E-6	Set the relative value to 1uF
CAP:NULL:VAL?	Query returns: +1.00000000E-06

:[SENSe:]TEMPerature:NULL:VALue

- Command format:

[SENSe:]TEMPerature:NULL:VALue {<value>|MIN|MAX|DEF}

[SENSe:]TEMPerature:NULL:VALue?

- **Functional description:**

  Set the relative value for the temperature measurement mode. The unit is °C/ °F/K (specified by the command UNIT:TEMPerature command, the default is °C). The rang is -2000 ~ +2000.

- **Return format:**

  Query returns the relative value. The unit is specified by the command UNIT:TEMPerature.

- **For example:**

UNIT:TEMP F	The temperature unit is °F.
TEMP:NULL:VAL 25	Set the relative value to 25 °F
TEMP:NULL:VAL?	Query returns: +2.50000000E+01

:[SENSe:]FREQuency:NULL:VALue

- **Command format:**

  [SENSe:]FREQuency:NULL:VALue {<value>|MIN|MAX|DEF}

  [SENSe:]FREQuency:NULL:VALue?

- **Functional description:**

  Set the relative value for the frequency mode. The unit is Hz, and the rang is 0 Hz ~ 2MHz.

- **Return format:**

  Query returns the relative value and the unit is Hz.

- **For example:**

FREQ:NULL:VAL 1	Set the relative value to 1Hz
FREQ:NULL:VAL?	Query returns: +1.00000000E+00

:[SENSe:]PERiod:NULL:VALue

- **Command format:**

  [SENSe:]PERiod:NULL:VALue {<value>|MIN|MAX|DEF}

  [SENSe:]PERiod:NULL:VALue?

- **Functional description:**

  Set the relative value for the period mode. The unit is s, and the rang is 0~2s.

- **Return format:**

  Query returns the relative value and the unit is second (s).

- **For example:**

PER:NULL:VAL 0.5	Set the relative value to 0.5s
PER:NULL:VAL?	Query returns: +0.50000000E+00

Automatic Switch of Relative Value

:[SENSe:]VOLTage:DC:NULL:VALue:AUTO

- Command format:

  [SENSe:]VOLTage:DC:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]VOLTage:DC:NULL:VALue:AUTO?

- Functional description:

  Enable or disable the automatic relative value of the DC voltage mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- Return format:

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- For example:

  VOLT:DC:NULL:VAL:AUTO ON Enable the automatic relative value.

  VOLT:DC:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]CURRent:DC:NULL:VALue:AUTO

- Command format:

  [SENSe:]CURRent:DC:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]CURRent:DC:NULL:VALue:AUTO?

- Functional description:

  Enable or disable the automatic relative value of the DC current mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- Return format:

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- For example:

  CURRent:DC:NULL:VAL:AUTO ON Enable the automatic relative value.

  CURRent:DC:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]VOLTage:AC:NULL:VALue:AUTO

- Command format:

  [SENSe:]VOLTage:AC:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]VOLTage:AC:NULL:VALue:AUTO?

- Functional description:

  Enable or disable the automatic relative value of the AC voltage mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- Return format:

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- For example:

  VOLT:AC:NULL:VAL:AUTO ON Enable the automatic relative value.

  VOLT:AC:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]CURRent:AC:NULL:VALue:AUTO

- Command format:

  [SENSe:]CURRent:AC:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]CURRent:AC:NULL:VALue:AUTO?

- Functional description:

  Enable or disable the automatic relative value of the AC current mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- Return format:

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- For example:

  CURRent:AC:NULL:VAL:AUTO ON Enable the automatic relative value.

  CURRent:AC:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]RESistance:NULL:VALue:AUTO

- Command format:

  [SENSe:]RESistance:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]RESistance:NULL:VALue:AUTO?

- Functional description:

  Enable or disable the automatic relative value of the 2-wire resistance mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- Return format:

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- For example:

  RES:NULL:VAL:AUTO ON Enable the automatic relative value.

  RES:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]FRESistance:NULL:VALue:AUTO

- Command format:

  [SENSe:]FRESistance:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]FRESistance:NULL:VALue:AUTO?

- **Functional description:**

  Enable or disable the automatic relative value of the 4-wire resistance mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- **Return format:**

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- **For example:**

  FRES:NULL:VAL:AUTO ON Enable the automatic relative value.

  FRES:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]CAPacitance:NULL:VALue:AUTO

- **Command format:**

  [SENSe:]CAPacitance:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]CAPacitance:NULL:VALue:AUTO?

- **Functional description:**

  Enable or disable the automatic relative value of the capacitance mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- **Return format:**

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- **For example:**

  CAPacitance:NULL:VAL:AUTO ON Enable the automatic relative value.

  CAPacitance:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]TEMPerature:NULL:VALue:AUTO

- **Command format:**

  [SENSe:]TEMPerature:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]TEMPerature:NULL:VALue:AUTO?

- **Functional description:**

  Enable or disable the automatic relative value of the temperature measurement mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- **Return format:**

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- **For example:**

  TEMP:NULL:VAL:AUTO ON Enable the automatic relative value.

  TEMP:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]FREQuency:NULL:VALue:AUTO

- Command format:

  [SENSe:]FREQuency:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]FREQuency:NULL:VALue:AUTO?

- Functional description:

  Enable or disable the automatic relative value of the frequency mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- Return format:

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- For example:

  FREQ:NULL:VAL:AUTO ON Enable the automatic relative value.

  FREQ:NULL:VAL:AUTO? Query returns: 1

:[SENSe:]PERiod:NULL:VALue:AUTO

- Command format:

  [SENSe:]PERiod:NULL:VALue:AUTO {ON|1|OFF|0}

  [SENSe:]PERiod:NULL:VALue:AUTO?

- Functional description:

  Enable or disable the automatic relative value of the period mode. After the automatic relative value is enabled, then enable the relative value function, the instrument will automatically record the first measured value as the relative value.

- Return format:

  Query returns the state of automatic relative value, 0 (OFF) or 1 (ON).

- For example:

  PER:NULL:VAL:AUTO ON Enable the automatic relative value.

  PER:NULL:VAL:AUTO? Query returns: 1

Temperature Measurement Setting Command

:[SENSe:]TEMPerature:TRANsducer:TYPE

- Command format:

  [SENSe:]TEMPerature:TRANsducer:TYPE {FRTD|RTD|FTHermistor|THERmistor|TCouple}

  [SENSe:]TEMPerature:TRANsducer:TYPE?

- Functional description:

  Select the probe for the temperature measurement, the probe can set to 2-wire RTD, 4-wire RTD, 2-wire thermistor, 4-wire thermistor and thermocouple (J/K/E/T/N/R/S/B).

- Return format:

Query returns the probe type of the temperature measurement.

- For example:

TEMP:TRAN:TYPE RTD	Select the 2-wire RTD for the temperature measurement
TEMP:TRAN:TYPE?	Query returns: RTD

:[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction

- Command format:

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction {<temperature>|MIN|MAX|DEF}

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction?

- Functional description:

Set the fixed reference junction temperature (the unit is Celsius degree) for the thermocouple temperature measurement. The range is -20 °C ~ +80 °C and the default is 0°C.

- Return format:

Query returns the fixed reference junction temperature value.

- For example:

TEMP:TRAN:TC:RJUN 20.0	Set the fixed reference junction temperature to 20.0°C in the thermocouple temperature measurement.
TEMP:TRAN:TC:RJUN?	Query returns: +2.00000000E+01

:[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:OFFSet:ADJust

- Command format:

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:OFFSet:ADJust {<temperature>|MIN|MAX|DEF}

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:OFFSet:ADJust?

- Functional description:

Sets the internal reference temperature (the unit is Celsius degree) for the thermocouple temperature measurement, in order to correct the error between the internal temperature measurement connected before the DMM and the actual temperature of the measuring terminal. The range is -20 °C ~ +20 °C and the default is 0°C.

- Return format:

Query returns the compensation temperature value.

- For example:

TEMP:TRAN:TC:RJUN:OFFS:ADJ -5	Set the internal reference temperature to -5 °C
TEMP:TRAN:TC:RJUN:OFFS:ADJ?	Query returns: -5.00000000E+00

:[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE

- Command format:

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE {INTernal|FIXed}

[SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE?

- Functional description:

Set the source for the reference junction temperature, it can use the internal measurement value or a known

fixed junction temperature. The reference junction temperature must be specified when perform the thermocouple temperature measurement.

- Return format:

  Query returns the source of the reference junction temperature, FIXed (fixed) or INTernal (internal).

- For example:

TEMP:TRAN:TC:RJUN:TYPE FIX	The source the reference junction temperature is FIXed.
TEMP:TRAN:TC:RJUN:TYPE?	Query returns: FIX

:[SENSe:]TEMPerature:TRANsducer:TCouple:TYPE

- Command format:

  [SENSe:]TEMPerature:TRANsducer:TCouple:TYPE {J|K|E|T|N|R|S|B}

  [SENSe:]TEMPerature:TRANsducer:TCouple:RJUNction:TYPE?

- Functional description:

  Set the thermocouple type for the thermocouple temperature measurement, it can set to J|K|E|T|N|R|S|B.

- Return format:

  Query returns the thermocouple type of the thermocouple temperature measurement

- For example:

TEMP:TRAN:TC:TYPE K	Select K-type for the thermocouple temperature measurement
TEMP:TRAN:TC:TYPE?	Query returns: K

:[SENSe:]TEMPerature:TRANsducer:RTD:RESistance

- Command format:

  [SENSe:]TEMPerature:TRANsducer:RTD:RESistance[:REFerence]{<reference>|MIN|MAX|DEF}

  [SENSe:]TEMPerature:TRANsducer:RTD:RESistance[:REFerence]? [{MIN|MAX|DEF}]

- Functional description:

  Set the RTD nominal resistance for 2-wire RTD, the range is 50Ω ~ 2100 Ω and the default is 100Ω. The nominal resistance (R0) needs to be configured according to the actual measurement.

- Return format:

  Query returns RTD nominal resistance.

- For example:

TEMP:TRAN:RTD:RES 50	Set RTD nominal resistance R0 = 50Ω
TEMP:TRAN:RTD:RES?	Query returns: +5.00000000E+01

:[SENSe:]TEMPerature:TRANsducer:FRTD:RESistance

- Command format:

  [SENSe:]TEMPerature:TRANsducer:FRTD:RESistance[:REFerence]{<reference>|MIN|MAX|DEF}

  [SENSe:]TEMPerature:TRANsducer:FRTD:RESistance[:REFerence]? [{MIN|MAX|DEF}]

- Functional description:

  Set the RTD nominal resistance for 4-wire RTD and the unit is Ω. The range is 50Ω~2100 Ω and the default is 100Ω. The nominal resistance (R0) needs to be configured according to the actual measurement.

- Return format:

  Query returns RTD nominal resistance, the unit is Ω.

- For example:

TEMP:TRAN:FRTD:RES 50	Set RTD nominal resistance R0 = 50Ω
TEMP:TRAN:FRTD:RES?	Query returns: +5.00000000E+01

Bandwidth of Filter Setting Command

:[SENSe:]VOLTage:AC:BANDwidth

- Command format:

  [SENSe:]VOLTage:AC:BANDwidth {3|20|200|MINimum|MAXimum}

  [SENSe:]VOLTage:AC:BANDwidth? [MINimum|MAXimum]?

- Functional description:

  Set the bandwidth of the filter for the AC voltage measurement and the unit is Hz. It can set to {3Hz\20Hz\200Hz}.

- Return format:

  Query returns the bandwidth of the filter.

- For example:

VOLT:AC:BAND 20	Set the bandwidth of the filter to 20Hz
VOLT:AC:BAND?	Query returns: 20

:[SENSe:]CURRent:AC:BANDwidth

- Command format:

  [SENSe:]CURRent:AC:BANDwidth {3|20|200|MINimum|MAXimum}

  [SENSe:]CURRent:AC:BANDwidth? [MINimum|MAXimum]?

- Functional description:

  Set the bandwidth of the filter for the AC current measurement and the unit is Hz. It can set to {3Hz\20Hz\200Hz}.

- Return format:

  Query returns the bandwidth of the filter.

- For example:

VOLT:AC:BAND 20	Set the bandwidth of the filter to 20Hz
VOLT:AC:BAND?	Query returns: 20

:[SENSe:]FREQuency:RANGe:LOWer

- Command format:

  [SENSe:]FREQuency:RANGe:LOWer {3|20|200|MINimum|MAXimum}

  [SENSe:]FREQuency:RANGe:LOWer? [MINimum|MAXimum]?

- Functional description:

  Set the bandwidth of the filter for the frequency measurement and the unit is Hz. It can set to

{3Hz\20Hz\200Hz}.

- **Return format:**

  Query returns the bandwidth of the filter.

- **For example:**

FREQ:RANG:LOW 20	Set the bandwidth of the filter to 20Hz
FREQ:RANG:LOW?	Query returns: 20

:[SENSe:]PERiod:RANGe:LOWer

- **Command format:**

  [SENSe:]PERiod:RANGe:LOWer {3|20|200|MINimum|MAXimum}

  [SENSe:]PERiod:RANGe:LOWer? [MINimum|MAXimum]?

- **Functional description:**

  Set the bandwidth of the filter for the period measurement and the unit is Hz. It can set to {3Hz\20Hz\200Hz}.

- **Return format:**

  Query returns the bandwidth of the filter.

- **For example:**

PER:RANG:LOW 20	Set the bandwidth of the filter to 20Hz
PER:RANG:LOW?	Query returns: 20

Automatic Zeroing Setting Command

:[SENSe:]VOLTage[:DC]:ZERO:AUTO

- **Command format:**

  [SENSe:]VOLTage[:DC]:ZERO:AUTO {OFF|ONCE|ON}

  [SENSe:]VOLTage[:DC]:ZERO:AUTO?

- **Functional description:**

  Set the state of automatic zeroing function, it can set to {OFF|ONCE|ON}.

State of Automatic Zeroing Function	Function
OFF	The instrument takes the last measured zeroed value and subtracts it from each measured value.
ON (default)	The instrument measure the offset after each measurement.
ONCE	The instrument acquires a measured zeroed value and sets automatic zeroing function to OFF. The acquired zero measurement is used for all subsequent measurements until the function, range or integration time is changed again.

- **Return format:**

  Query returns the state of automatic zeroing function.

- For example:

  VOLT:ZERO:AUTO ON Enable the automatic zeroing function.

  VOLT:ZERO:AUTO? Query returns: 1

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

- Command format:

  [SENSe:]CURRent[:DC]:ZERO:AUTO {OFF|ONCE|ON}

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

- Functional description:

  Set the state of automatic zeroing function, it can set to {OFF|ONCE|ON}.

State of Automatic Zeroing Function	Function
OFF	The instrument takes the last measured zeroed value and subtracts it from each measured value.
ON (default)	The instrument measure the offset after each measurement.
ONCE	The instrument acquires a measured zeroed value and sets automatic zeroing function to OFF. The acquired zero measurement is used for all subsequent measurements until the function, range or integration time is changed again.

- Return format:

  Query returns the state of automatic zeroing function.

- For example:

  CURR:ZERO:AUTO ON Enable the automatic zeroing function.

  CURR:ZERO:AUTO? Query returns: 1

:[SENSe:]RESistance[:DC]:ZERO:AUTO

- Command format:

  [SENSe:]RESistance[:DC]:ZERO:AUTO {OFF|ONCE|ON}

  [SENSe:]RESistance[:DC]:ZERO:AUTO?

- Functional description:

  Set the state of automatic zeroing function, it can set to {OFF|ONCE|ON}.

State of Automatic Zeroing Function	Function
OFF	The instrument takes the last measured zeroed value and subtracts it from each measured value.
ON (default)	The instrument measure the offset after each measurement.
ONCE	The instrument acquires a measured zeroed value and sets automatic zeroing function to OFF. The acquired zero measurement is used for all subsequent measurements until the function, range or integration time is changed again.

- **Return format:**

  Query returns the state of automatic zeroing function.

- **For example:**

  RES:ZERO:AUTO ON ·Enable the automatic zeroing function.

  RES:ZERO:AUTO? Query returns: 1

:[SENSe:]FRESistance[:DC]:ZERO:AUTO

- **Command format:**

  [SENSe:]FRESistance[:DC]:ZERO:AUTO {OFF|ONCE|ON}

  [SENSe:]FRESistance[:DC]:ZERO:AUTO?

- **Functional description:**

  Set the state of automatic zeroing function, it can set to {OFF|ONCE|ON}.

State of Automatic Zeroing Function	Function
OFF	The instrument takes the last measured zeroed value and subtracts it from each measured value.
ON (default)	The instrument measure the offset after each measurement.
ONCE	The instrument acquires a measured zeroed value and sets automatic zeroing function to OFF. The acquired zero measurement is used for all subsequent measurements until the function, range or integration time is changed again.

- **Return format:**

  Query returns the state of automatic zeroing function.

- **For example:**

  FRES:ZERO:AUTO ON Enable the automatic zeroing function.

  FRES:ZERO:AUTO? Query returns: 1

Measuring Interval Setting Command

:[SENSe:]FREQuency:APERture

- **Command format:**

  [SENSe:]FREQuency:APERture {0.01|0.1|1|MINimum|MAXimum}

  [SENSe:]FREQuency:APERture? [MINimum|MAXimum]?

- **Functional description:**

  Set the interval time (gate time) for the frequency measurement and the unit is s. It can set to {1ms\10ms\100ms\1s}.

- **Return format:**

  Query returns the interval time.

- For example:

FREQ:APER 0.1	Set the interval time to 100ms
FREQ:APER?	Query returns: +1.00000000E-03

:[SENSe:]PERiod:APERture

- Command format:

[SENSe:]PERiod:APERture {0.01|0.1|1|MINimum|MAXimum}

[SENSe:]PERiod:APERture? [MINimum|MAXimum]?

- Functional description:

Set the interval time (gate time) for the period measurement and the unit is s. It can set to {1ms\10ms\100ms\1s}.

- Return format:

Query returns the interval time.

- For example:

FREQ:APER 0.1	Set the interval time to 100ms
FREQ:APER?	Query returns: +1.00000000E-03

Current Terminal Setting Command

:[SENSe:]CURRent:DC:TERMinals

- Command format:

[SENSe:]CURRent:DC:TERMinals {Small|Big}

[SENSe:]CURRent:DC:TERMinals?

- Functional description:

Configure the DC current measurement to measure power at the 200mA (Small) or 10A (Big) terminals.

- Return format:

Query returns the terminal type.

- For example:

CURRent:DC:TERM Big	Configure the current measurement to measure power at 10A terminals.
CURRent:DC:TERM?	Query returns: Big

:[SENSe:]CURRent:AC:TERMinals

- Command format:

[SENSe:]CURRent:AC:TERMinals {Small|Big}

[SENSe:]CURRent:AC:TERMinals?

- Functional description:

Configure the AC current measurement to measure power at the 200Ma or 10A terminals.

- Return format:

Query returns the terminal type.

- For example:

CURRent:AC:TERM Big	Configure the current measurement to measure power at 10A terminals.
CURRent:AC:TERM?	Query returns: Big

Relevance Command of Specified Scale

:[SENSe:]VOLTage:DC:IMPedance:AUTO

- Command format:

  [SENSe:]VOLTage:DC:IMPedance:AUTO {ON|1|OFF|0}

  [SENSe:]VOLTage:DC:IMPedance:AUTO?

- Functional description:

  Disable or enable the automatic input impedance for the DC voltage measurement.

- Return format:

  The state of the automatic input impedance, 0 (OFF) or 1 (ON).

- For example:

VOLT:DC:IMPedance:AUTO ON	Turn on the automatic input impedance
VOLT:DC:IMPedance:AUTO?	Query returns: 1

:[SENSe:]CONTinuity:THReshold:VALue

- Command format:

  [SENSe:]CONTinuity:THReshold:VALue {<value>|MIN|MAX|DEF}

  [SENSe:]CONTinuity:THReshold:VALue? [{MIN|MAX|DEF}]

- Functional description:

  Set the threshold resistance for the continuity measurement. The range is 0~2000 Ω, and the unit of the reference parameter is Ω.

- Return format:

  Query returns the threshold resistance value and the unit is Ω.

- For example:

CONT:THR:VAL 2000	Set the threshold resistance value to 2000Ω
CONT:THR:VAL?	Query returns: +2.00000000E+03

:CONTinuity:BEEPer:STATe

- Command format:

  CONTinuity:BEEPer:STATe {ON|1|OFF|0}

  CONTinuity:BEEPer:STATe?

- Functional description:

  Turn on or off the beeper during the continuity measurement.

- Return format:

  Query returns the beeper state during the continuity measurement.

- **For example:**

  CONT:BEEP:STAT ON Turn on the beeper during the continuity measurement.

  CONT:BEEP:STAT? Query returns: 1

DATA Subsystem

This command is used to manage the data cached of the instrument, including getting the instrument cached data, returning the specified data, deleting the data, and querying the amount of data cached.

Acquire Cached Data of Instrument

:FETCh?

- **Command format:**

  FETCh?

- **Functional description:**

  Wait for the recent measurement to complete and return all available measured data.

- **Return format:**

  Available measured data.

- **For example:**

  FETC? Return the measured data: -5.75122019E-04, -5.77518360E-04

:READ?

- **Command format:**

  READ?

- **Functional description:**

  Start to measure a new group of measurement, read the measured data of the instrument and erases the measured results.

- **Return format:**

  The currently cached measured data.

- **For example:**

  READ? Read return: -1.23006735E-03

Cached Data Management Command

:DATA:POINts

- **Command format:**

  DATA:POINts?

- **Functional description:**

  Query the total number of measurements cached by the instrument, up to 1,000 measurements.

- **Return format:**

  The total number of measurements (decimalism).

- **For example:**

  DATA:POIN? Query returns: 100

:DATA:LAST

- **Command format:**

  DATA:LAST?

- **Functional description:**

  Return the recent measured value.

- **Return format:**

  The unit of the measured value.

- **For example:**

  DATA:LAST? Query returns: -4.79221344E-04 VDC

:DATA:REMove

- **Command format:**

  DATA:REMove? <num_readings>

- **Functional description:**

  Read and erases the measured result of <num_readings> from the instrument data cache. If the measured result is less than n <num_readings>, then the measured results are returned as many as are actually cached.

- **Return format:**

  The measured results of 1~10000 that separated by comma.

- **For example:**

  DATA:REM? 5 Query returns:

  -4.55379486E-04,-4.55975533E-04,-4.56273556E-04,
  -4.53591347E-04,-4.55379486E-04

TRIGger Subsystem

This command is used to configure the parameter for measurement, sampling and start the trigger measurement instantly.

Instantly Trigger Setting Command

:INITiate[:IMMediate]

- **Command format:**

  INITiate[:IMMediate]

- **Functional description:**

  Set the state of trigger system to "wait for trigger".

- **Return format:**

  Nothing.

- **For example:**

  INITiate Set the trigger system to "wait for trigger"

Measurement Trigger Setting Command

:SAMPle:COUNt

- **Command format:**

  SAMPle:COUNt {<count>|MIN|MAX|DEF}

  SAMPle:COUNt?

- **Functional description:**

  To specifie the sampling count to be collected by the instrument in one single trigger, it can set to1(default) ~100,000.

- **Return format:**

  Query returns the collected sampling count of single measurement.

- **For example:**

  SAMP:COUN 10 Collect 10 samples per measurement

  SAMPle:COUNt? Query returns 10

:TRIGger:COUNt

- **Command format:**

  TRIGger:COUNt {<count>|MIN|MAX|DEF}

  TRIGger:COUNt? [{MIN|MAX|DEF}]

- **Functional description:**

  Set the trigger to be accepted by the instrument in one measurement.

- **Return format:**

  Query returns the trigger count.

- **For example:**

  TRIG:COUN 10 Set one measurement to accept 10 measurement triggers

  TRIG:COUN? Query returns: 10

:TRIGger:DELay

- **Command format:**

  TRIGger:DELay {<seconds>|MIN|MAX|DEF}

  TRIGger:DELay? [{MIN|MAX|DEF}]

- **Functional description:**

  Set the delay between the trigger signal and the first measurement. The unit is s and the range is 10us ~10s.

- **Return format:**

  Query returns the delay measurement, and the unit is s.

- **For example:**

  TRIG:DEL 2 Set the delay measurement to 2s

  TRIG:DEL? Query returns: +2.00000000E+00

:TRIGger:DELay:AUTO

- **Command format:**

  TRIGger:DELay:AUTO {ON|1|OFF|0}

  TRIGger:DELay:AUTO?

- **Functional description:**

  Disable or enable the automatic trigger delay function.

- **Return format:**

  Query returns 0 (OFF) or 1 (ON).

- **For example:**

  TRIG:DEL:AUTO ON Enable the automatic trigger delay function

  TRIG:DEL:AUTO? Query returns: 1

:TRIGger:SLOPe

- **Command format:**

  TRIGger:SLOPe {POSitive|NEGative}

  TRIGger:SLOPe?

- **Functional description:**

  When using the signal input from the rear panel Ext Trig BNC connector as the trigger source, use this command to select the polarity of trigger signal, rising edge (POS) or falling edge (NEG) trigger.

- **Return format:**

  Query returns the polarity of trigger signal, rising edge (POS) or falling edge (NEG) trigger.

- **For example:**

  TRIG:SLOP POS Select rising edge (POS)

  TRIG:SLOP? Query returns: POS

:TRIGger:SOURce

- Command format:

  TRIGger:SOURce {IMMediate|EXTernal|BUS}

  TRIGger:SOURce?

- Functional description:

  Select the trigger source for the measurement as described below.

IMMediate	The trigger signal will always exit. When set the instrument to "wait for trigger", it will send the trigger signal immediately.
BUS	Once the multimeter is in the state of "wait for trigger", *TRG will trigger the instrument via the remote interface.
EXTernal	The Ext Trig input on the rear panel of the instrument is used as the trigger signal, and the specified number of measurements will be collected for each trigger (SAMPle:COUNt). If the instrument receives an external trigger before it is ready, the instrument will buffer a trigger.

- Return format:

  Query returns the current trigger source.

- For example:

TRIG:SOURce EXT	Select the external trigger source
TRIG:SOURce?	Query returns: EXT

CALCulate Subsystem

This command is used to configure the parameter for data operation, including limit search, histogram operation, statistical operation and calibration operation.

Mathematical Operation Setting Command

:CALCulate:CLEar

- Command format:

  CALCulate:CLEar[:IMMediate]

- Functional description:

  To clear all limit conditions, histogram data, statistics and measured results.

- Return format:

  Nothing.

- For example:

  Omissible.

:CALCulate:RELATive[:STATe]

- Command format:

CALCulate:RELATive[:STATe]{ON|1|OFF|0}

CALCulate:RELATive[:STATe]?

- **Functional description:**

  Turn on or off the relative value function for the current measurement mode.

- **Return format:**

  Query returns the relative value state of the current measurement mode.

- **For example:**

CALCulate:RELATive ON	Turn on the relative value function of the current measurement mode
CALCulate:RELATive?	Query returns: 1

:CALCulate:RELATive:DATA

- **Command format:**

  CALCulate:RELATive:DATA {<value>|MIN|MAX|DEF}

  CALCulate:RELATive:DATA?

- **Functional description:**

  Set the relative value for the current measurement mode.

Measurement Mode	Range of Relative Value
VOLTage:DC/AC	-1200V~1200V
CURRent:DC/AC	-12A~12A
RESistance	0~120MΩ
FRESistance	0~2.4MΩ
CAPacitance	0~2.4mF
TEMPerature	-2000~+2000[°C/ °F/K]
CONTinuity	0~2KΩ
DIODe	0~4V
FREQuency	0~2MHZ
PERiod	0~2S

- **Return format:**

  Query returns the relative value of the current measurement mode.

- **For example:**

CALCulate:RELAT:DATA 10	Set the relative value of the current measurement mode
CALCulate:RELAT:DATA?	Query returns: +1.00000000E+01

Limits Measurement Setting Command

:CALCulate:LIMit:CLEar

- **Command format:**

  CALCulate:LIMit:CLEar[:IMMediate]

- **Functional description:**

  To clear the limit test result, the front panel indication of an exceeded limit, and bits 11("Lower Limit Fail") and 12("Upper Limit Fail") of the condition register in the event register group of the suspicious data register.

- **Return format:**

  Nothing.

- **For example:**

  CALCulate:LIMit:CLEar Clear the measured result of the limits measurement

:CALCulate:LIMit:LOWer

- **Command format:**

  CALCulate:LIMit:LOWer[:DATA]{<value>|MIN|MAX|DEF}

  CALCulate:LIMit:LOWer[:DATA]?[{MIN|MAX|DEF}]

- **Functional description:**

  Set the lower limit for the limits mode, the unit(V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode, and the range is as shown in the table. A command error occurs if the set lower limit value is higher than the set upper limit value.

Measurement Mode	Range of Lower Limit
VOLTage:DC/AC	-1200V~1200V
CURRent:DC/AC	-12A~12A
RESistance	0~120MΩ
FRESistance	0~2.4MΩ
CAPacitance	0~2.4mF
TEMPerature	-2000~+2000[°C/ °F/K]
CONTinuity	0~2KΩ
DIODe	0~4V
FREQuency	0~2MHZ
PERiod	0~2S

- **Return format:**

  Query returns the lower limit of the limits measurement. The unit(V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the

measurement mode.

- For example:

CALCulate:LIMit:LOWer -1000 Set the lower limit of the limits measurement to -1000

CALCulate:LIMit:LOWer? Query returns: -1.00000000E+03

:CALCulate:LIMit:UPPer

- Command format:

CALCulate:LIMit:UPPer[:DATA]{<value>|MIN|MAX|DEF}

CALCulate:LIMit:UPPer[:DATA]?[{MIN|MAX|DEF}]

- Functional description:

Set the upper limit for the limits mode, the unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode, and the range is as shown in the table. A command error occurs if the set upper limit value is lower than the set lower limit value.

Measurement Mode	Range of Lower Limit
VOLTage:DC/AC	-1200V~1200V
CURRent:DC/AC	-12A~12A
RESistance	0~120MΩ
FRESistance	0~2.4MΩ
CAPacitance	0~2.4mF
TEMPerature	-2000~+2000[°C/ °F/K]
CONTinuity	0~2KΩ
DIODe	0~4V
FREQuency	0~2MHZ
PERiod	0~2S

- Return format:

Query returns the upper limit of the limits measurement. The unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- For example:

CALCulate:LIMit:UPPer 1000 Set the upper limit of the limits measurement to 1000

CALCulate:LIMit:UPPer? Query returns: +1.00000000E+03

:CALCulate:LIMit:STATe

- Command format:

CALCulate:LIMit:STATe {ON|1|OFF|0}

CALCulate:LIMit:STATe?

- Functional description:

Enable or disable the limits measurement.

- Return format:

  Query returns the enable state of the limits measurement.

- For example:

  CALCulate:LIMit:STATe ON Enable the limits measurement.

  CALCulate:LIMit:STATe? Query returns: ON

:LIMit:BEEPer:STATe

- Command format:

  LIMit:BEEPer:STATe {ON|1|OFF|0}

  LIMit:BEEPer:STATe?

- Functional description:

  Turn on or off the beeper when it over the limit in the limits measurement.

- Return format:

  Query returns the beeper state in the limits measurement.

- For example:

  LIMit:BEEPer:STATe ON Enable the limits measurement.

  LIMit:BEEPer:STATe? Query returns: ON

Histogram Operation Setting Command

:CALCulate:TRANsform:HISTogram:ALL

- Command format:

  CALCulate:TRANsform:HISTogram:ALL?

- Functional description:

  Query the upper range, lower range, number of measurements of the histogram, and bar data collected since the last time the histogram data was cleared.

- Return format:

  The upper range, the lower range, number of measurements, bar data (the number of bar, the number of measurements that below the lower limit, the number of measurements that corresponding to each bar, the number of measurements that above the upper limit).

- For example:

  CALCulate:TRANsform:HISTogram:ALL? Query returns:

  -1.85337982E+08,+1.60497704E+07,39,12,0,1,0,3,

  3,4,4,7,8, 5,4,0

:CALCulate:TRANsform:HISTogram:DATA

- Command format:

  CALCulate:TRANsform:HISTogram:DATA?

- Functional description:

  Query the bar data collected by the histogram.

- Return format:

  The number of bar, the number of measurements that below the lower limit, the number of measurements that corresponding to each bar and the number of measurements that above the upper limit.

- For example:

  CALCulate:TRANsform:HISTogram:DATA? Query returns:

  12,0,1,0,3, 3,4,4,7,8, 5,4,0

:CALCulate:TRANsform:HISTogram:COUNt

- Command format:

  CALCulate:TRANsform:HISTogram:COUNt?

- Functional description:

  Query the number of measurements collected since the last time the histogram data was cleared.

- Return format:

  The number of decimal measurement results.

- For example:

  CALCulate:TRANsform:HISTogram:COUN? Query returns: +96

:CALCulate:TRANsform:HISTogram:POINts

- Command format:

  CALCulate:TRANsform:HISTogram:POINts {<value>|MIN|MAX|DEF}

  CALCulate:TRANsform:HISTogram:POINts?

- Functional description:

  Set the number of bars {10/20/40/100/200/400} between the lower and upper range values of the histogram. Two additional bars are always present: one for measurements below the lower range limit and one for measurements above the upper range limit. Set the number of bars only when the automatic selection of lower and upper range values is disabled.

- Return format:

  The bar number in decimalism.

- For example:

  CALCulate:TRANsform:HISTogram:POINts 100 Set the bar number for the histogram

  CALCulate:TRANsform:HISTogram:POINts? Query returns: +100

:CALCulate:TRANsform:HISTogram:RANGe:AUTO

- Command format:

  CALCulate:TRANsform:HISTogram:RANGe:AUTO {ON|1|OFF|0}

  CALCulate:TRANsform:HISTogram:RANGe:AUTO?

- Functional description:

  Enable or disable the automatic lower limit and automatic upper limit for the histogram, which is automatic scaling function.

- **Return format:**

  0 (OFF) or 1 (ON).

- **For example:**

  CALCulate:TRANsform:HISTogram:RANGe:AUTO ON Turn on the automatic lower limit and automatic upper limit for the histogram

  CALCulate:TRANsform:HISTogram:RANGe:AUTO? Query returns: 1, turn on the automatic lower limit and automatic upper limit

:CALCulate:TRANsform:HISTogram:RANGe:LOWer

- **Command format:**

  CALCulate:TRANsform:HISTogram:RANGe:LOWer {<value>|MIN|MAX|DEF}

  CALCulate:TRANsform:HISTogram:RANGe:LOWer?

- **Functional description:**

  Set the lower limit for the histogram, the unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode, and the range is as shown in the table. A command error occurs if the set lower limit value is higher than the set upper limit value. The lower limit setting will disable the automatic lower limit and automatic upper limit.

Measurement Mode	Range of Lower Limit
VOLTage:DC/AC	-1200V~1200V
CURRent:DC/AC	-12A~12A
RESistance	0~120MΩ
FRESistance	0~2.4MΩ
CAPacitance	0~2.4mF
TEMPerature	-2000~+2000 [°C/ °F/K]
CONTinuity	0~2KΩ
DIODe	0~4V
FREQuency	0~2MHZ
PERiod	0~2S

- **Return format:**

  The lower range and the unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- **For example:**

  CALCulate:TRANsform:HISTogram:RANGe:LOWer -1000 Set the lower limit to -1000

  CALCulate:TRANsform:HISTogram:RANGe:LOWer? Query returns: -1.00000000E+03

:CALCulate:TRANsform:HISTogram:RANGe:UPPer

- **Command format:**

  CALCulate:TRANsform:HISTogram:RANGe:Upper {<value>|MIN|MAX|DEF}

  CALCulate:TRANsform:HISTogram:RANGe:Upper?

- **Functional description:**

Set the upper limit for the histogram, the unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode, and the range is as shown in the table. A command error occurs if the set upper limit value is lower than the set lower limit value. The upper limit setting will disable the automatic lower limit and automatic upper limit.

Measurement Mode	Range of Lower Limit
VOLTage:DC/AC	-1200V~1200V
CURRent:DC/AC	-12A~12A
RESistance	0~120MΩ
FRESistance	0~2.4MΩ
CAPacitance	0~2.4mF
TEMPerature	-2000~+2000 [°C/ °F/K]
CONTinuity	0~2KΩ
DIODe	0~4V
FREQuency	0~2MHZ
PERiod	0~2S

- **Return format:**

  The upper range and the unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- **For example:**

  CALCulate:TRANsform:HISTogram:RANGe:Upper 1000 Set the upper limit to 1000

  CALCulate:TRANsform:HISTogram:RANGe:Upper? Query returns: +1.00000000E+03

:CALCulate:TRANsform:HISTogram:STATe

- **Command format:**

  CALCulate:TRANsform:HISTogram:[:STATe] {ON|1|OFF|0}

  CALCulate:TRANsform:HISTogram:[:STATe]?

- **Functional description:**

  Enable or disable the histogram operation.

- **Return format:**

  0 (OFF) or 1 (ON).

- **For example:**

  CALCulate:TRANsform:HISTogram:STATe ON Enable the histogram operation.

  CALCulate:TRANsform:HISTogram:STATe? Query returns: 1

:CALCulate:TRANsform:HISTogram:CLEar[:IMMediate]

- **Command format:**

  CALCulate:TRANsform:HISTogram:CLEar[:IMMediate]

- **Functional description:**

  To clear histogram statistics; resets bar counts, upper and lower limits if automatic scaling is turned on.

- **Return format:**

  Nothing.

- For example:

  CALCulate:TRANsform:HISTogram:CLEar Clear histogram statistics

Calibration Operation Setting Command

:CALCulate:SCALe:DB:REFerence

- Command format:

  CALCulate:SCALe:DB:REFerence {<reference>|MIN|MAX|DEF}

  CALCulate:SCALe:DB:REFerence? [{MIN|MAX|DEF}]

- Functional description:

  Set the reference value for DB calibration, and the range is -200.0 dBm ~ +200.0 dBm.

- Return format:

  Query returns the reference value of DB calibration.

- For example:

  CALCulate:SCALe:DB:REFerence -10.0 Set the reference value of DB calibration to -10.0dBm

  CALCulate:SCALe:DB:REFerence? Query returns: -1.00000000E+01

:CALCulate:SCALe:DBM:REFerence

- Command format:

  CALCulate:SCALe:DBM:REFerence {<reference>|MIN|MAX|DEF}

  CALCulate:SCALe:DBM:REFerence? [{MIN|MAX|DEF}]

- Functional description:

  Set the resistance reference for the voltage measurement when enable the dB/dBm function. The unit is Ω. It can set to 50, 75, 93, 110, 124, 125, 135, 150, 250, 300, 500, 600, 800, 900, 1000, 1200, or 8000 Ω. This reference affects both dBm and dBm calibrations.

- Return format:

  Query returns the resistance reference value when using the calibration, and the unit is Ω.

- For example:

  CALCulate:SCALe:DBM:REF 300 Use the resistance reference of 300 Ω to enable the calibration

  CALCulate:SCALe:DBM:REF? Query returns: +3.00000000E+02

:CALCulate:SCALe:FUNCtion

- Command format:

  CALCulate:SCALe:FUNCtion {DB|DBM}

  CALCulate:SCALe:FUNCtion?

- Functional description:

  Select the calibration function used for the calibration execution operation; DB and DBM calibration is only available for DC voltage and AC voltage measurements.

- Return format:

Query returns Calibration function, DB or DBM

- For example:

CALCulate:SCALe:FUNC DBM Select the DBM calibration function

CALCulate:SCALe:FUNC? Query returns: DBM

:CALCulate:SCALe:REFerence:AUTO

- Command format:

CALCulate:SCALe:REFerence:AUTO {ON|1|OFF|0}

CALCulate:SCALe:REFerence:AUTO?

- Functional description:

Enable or disable the automatic reference value for dB calibration. When it enabled, the first measurement will be used as a reference for all subsequent measurements.

- Return format:

Query returns the state of the automatic reference value in dB calibration, 0 (OFF) or 1 (ON).

- For example:

CALCulate:SCALe:REF:AUTO ON Enable the automatic reference value for dB calibration.

CALCulate:SCALe:REF:AUTO? Query returns: 1

:CALCulate:SCALe[:STATe]

- Command format:

CALCulate:SCALe[:STATe]{ON|1|OFF|0}

CALCulate:SCALe[:STATe]?

- Functional description:

Enable or disable the calibration function.

- Return format:

Query returns the state of the calibration function, 0 (OFF) or 1 (ON).

- For example:

CALC:SCAL:STAT ON Enable the calibration function

CALC:SCAL:STAT? Query returns: 1

Statistical Operation Setting Command

:CALCulate:AVERage[:STATe]

- Command format:

CALCulate:AVERage[:STATe]{ON|1|OFF|0}

CALCulate:AVERage[:STATe]?

- Functional description:

Enable or disable the statistical operation.

- Return format:

Query returns the state of the statistical operation, 0 (OFF) or 1 (ON).

- For example:

  CALC:AVER:STAT ON Enable the statistical operation

  CALC:AVER:STAT? Query returns: 1

:CALCulate:AVERage:ALL

- Command format:

  CALCulate:AVERage:ALL?

- Functional description:

  Query returns the result of the statistical operation, including arithmetic mean (average value), standard deviation, maximum and minimum values.

- Return format:

  Arithmetic mean,standard deviation,maximum value,minimum value, all in the form of +1.00000000E+01.

- For example:

  CALCulate:AVERage:ALL? Query returns:

  +1.00520000E+03,+1.00512000E+03,+1.00527000E+03,

  +4.13500000E-01

:CALCulate:AVERage:MINimum

- Command format:

  CALCulate:AVERage:MINimum?

- Functional description:

  Query returns the minimum value of the statistical operation result. The unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- Return format:

  Query returns the minimum value of the statistical operation result (in the format of+1.00000000E+01).

- For example:

  CALCulate:AVERage:MIN? Query returns: +1.00520000E+03

:CALCulate:AVERage:MAXimum

- Command format:

  CALCulate:AVERage:MAXimum?

- Functional description:

  Query returns the maximum value of the statistical operation result. The unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- Return format:

  Query returns the maximum value of the statistical operation result (in the format of +1.00000000E+01).

- For example:

  CALCulate:AVERage:MAX? Query returns: +1.00520000E+03

:CALCulate:AVERage:AVERage

- Command format:

  CALCulate:AVERage:AVERage?

- Functional description:

  Query returns the average value of the statistical operation result. The unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- Return format:

  Query returns the average value of the statistical operation result (in the format of +1.00000000E+01).

- For example:

  CALCulate:AVERage:AVER? Query returns: +1.00520000E+03

:CALCulate:AVERage:COUNt

- Command format:

  CALCulate:AVERage:COUNt?

- Functional description:

  Query returns the count value of the statistical operation result.

- Return format:

  Query returns the count value of the statistical operation result (decimalism).

- For example:

  CALCulate:AVERage:COUN? Query returns: +129

:CALCulate:AVERage:PTPeak

- Command format:

  CALCulate:AVERage:PTPeak?

- Functional description:

  Query returns the peak-to-peak value of the statistical operation result. The unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- Return format:

  Query returns the peak-to-peak value of the statistical operation result (in the format of +1.00000000E+01).

- For example:

  CALCulate:AVERage:PTP? Query returns: +1.00520000E+03

:CALCulate:AVERage:SDEViation

- Command format:

  CALCulate:AVERage:SDEViation?

- Functional description:

  Query returns the standard deviation of the statistical operation result. The unit (V/A/Ω/F/Hz/S/[°C/ °F/K]) depends on the measurement mode.

- Return format:

  Query returns the standard deviation of the statistical operation result (in the format of +1.00000000E+01).

- For example:

  CALCulate:AVERage:SDEV? Query returns: +1.00520000E+03

:CALCulate:AVERage:CLEar[:IMMediate]

- Command format:

  CALCulate:AVERage:CLEar[:IMMediate]

- **Functional description:**

  To clear the statistical information for statistical operations, including minimum, maximum, average, peak-to-peak, count, and standard deviation.

- **Return format:**

  Nothing.

- **For example:**

  CALCulate:AVERage:CLE Clear the statistical information

OUTPut Subsystem

This command is used to control the VMC signal output of the multimeter.

VMC Signal Output Setting Command

:OUTPut:TRIGger:SLOPe

- **Command format:**

  OUTPut:TRIGger:SLOPe {POSitive|NEGative}

  OUTPut:TRIGger:SLOPe?

- **Functional description:**

  Set the polarity of the voltmeter complete output signal at the VM Comp BNC connector on the rear panel.

- **Return format:**

  Query returns the polarity of VMC output pulse signal.

- **For example:**

  OUTPut:TRIGger:SLOPe POS VMC output the positive pulse signal

  OUTPut:TRIGger:SLOPe? Query returns POS

DISPlay Subsystem

This command is used to control the state of display screen and the user-defined information.

:DISPlay[:STATe]

- **Command format:**

  DISPlay[:STATe]{OFF|ON}

  DISPlay[:STATe]?

- **Functional description:**

  Disable or enable the front display screen.

- **Return format:**

  Query returns the state of the front display screen.

- **For example:**

DISP OFF	Turn off the display
DISP?	Query returns 0

:DISPlay:TEXT[:DATA]

- **Command format:**

  DISPlay:TEXT[:DATA]<quoted string>

  DISPlay:TEXT[:DATA]?

- **Functional description:**

  Set the text message displayed on the screen on the front panel.

- **Return format:**

  Query returns the text message displayed on the screen.

- **For example:**

DISP:TEXT "Test in progress..."	Set the text message: "Test in progress..."
DISP:TEXT?	Query returns: "Test in progress..."

:DISPlay:TEXT:CLEar

- **Command format:**

  DISPlay:TEXT:CLEar

- **Functional description:**

  To clear the text message displayed on the screen on the front panel.

- **Return format:**

  Nothing.

- **For example:**

DISP:TEXT:CLEar	Clear the text message

SYSTem Subsystem

This command is used to exit remote mode, configure system parameter, and obtain system information, such as error information.

:SYSTem:ERRor

- **Command format:**
  SYSTem:ERRor[:NEXT]?
- **Functional description:**
  Read and clear an error from the error queue.
- **Return format:**
  Return error message: <error code>,<error string>.
- **For example:**
  SYST:ERR? Return: -113,"Undefined header"

:SYSTem:BEEPer:STATe

- **Command format:**
  SYSTem:BEEPer:STATe {ON|1|OFF|0}
  SYSTem:BEEPer:STATe?
- **Functional description:**
  Disable or enable the beeper function during continuity, diode, probe hold measurements, or when the current panel or remote interface generates an error.
- **Return format:**
  Return the state of the beeper, 0 (OFF) or 1 (ON).
- **For example:**
  SYST:BEEP:STATe ON Turn on the beeper function
  SYST:BEEP:STATe? Return: 1

*SYSTem:LOCal

- **Command format:**
  SYSTem:LOCal
- **Functional description:**
  The button on the front panel will unlock when returning to local mode from remote mode.
- **Return format:**

Nothing.

*SYSTem:REMote

- **Command format:**

  SYSTem:REMote

- **Functional description:**

  Enter the remote mode, the button on the front panel will not lock.

- **Return format:**

  Nothing.

*SYSTem:RWLock

- **Command format:**

  SYSTem:RWLock

- **Functional description:**

  Enter the remote mode and the button on the front panel are locked. It cannot be operated.

- **Return format:**

  Nothing.

:UNIT:TEMPerature {C|F|K}

- **Command format:**

  UNIT:TEMPerature {C|F|K}

  UNIT:TEMPerature?

- **Functional description:**

  Set the unit for the temperature measurement (°C, °F or Kelvin).

- **Return format:**

  Query returns the unit of the temperature measurement

- **For example:**

  UNIT:TEMP F Set the unit of the temperature measurement to °F

  UNIT:TEMP? Query returns: F

Explanation of Programming

This chapter is to describe troubleshooting in process of programming. If you meet any of the following problems, please handle them according to the related instructions.

Programming Preparation

The programming preparation is only applicable for using Visual Studio and Lab VIEW development tools to programming under Windows operating system.

Firstly, the user need to confirm that whether NI -VISA library is installed (it can be download from the website https://www.ni.com/en-ca/support/downloads/drivers/download.ni-visa.html).

In this manual, the default installment path is C:\Program Files\IVI Foundation\VISA.

Build communication with PC via USB or LAN interface of the instrument, use USB data line to connect USB DEVICE port on the rear panel of the instrument with USB port of PC, or use LAN data line to connect LAN port on the rear panel of the instrument with LAN port of PC.

VISA Programming Example

There are some example in this section. Through these examples, the user can know how to use VISA, and it can combined with the command of programming manual to realize the control of the instrument. With these examples, the user can develop more applications.

VC++ Example

- Environment: Window system, visual studio.
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps

1. Open the visual studio software to create a new VC++ win32 console project.
2. Set the project environment that can adjust NI-VISA library, which are static library and dynamic library.

a) Static library

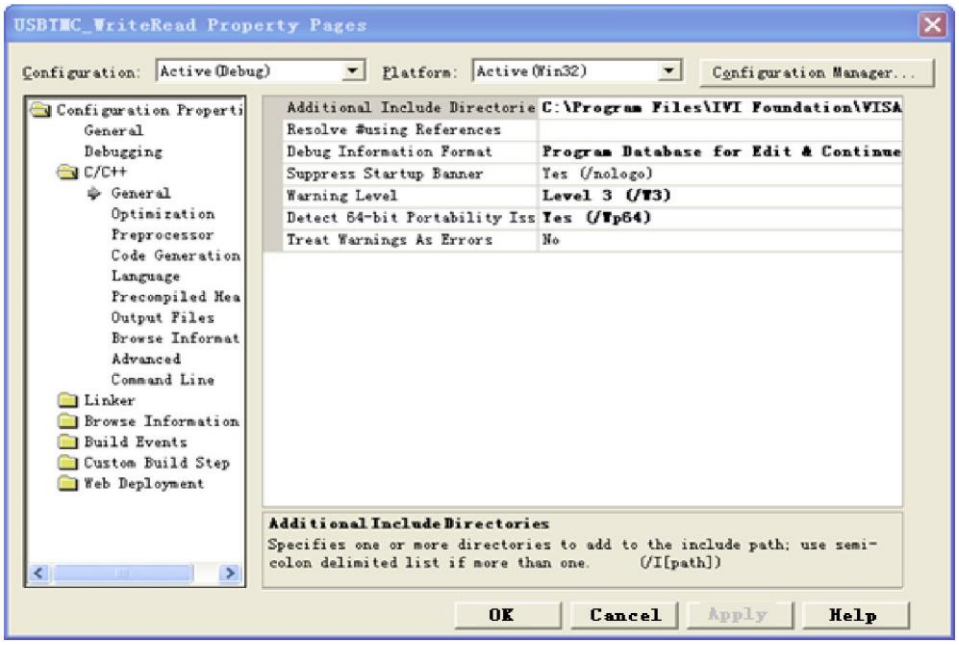
Find file visa.h, visatype.h and visa32.lib in NI-VISA installment path and copy them to the root path of VC++ project and add it to the project. Add two lines of code into file projectname.cpp as follows.

```
#include "visa.h"
```

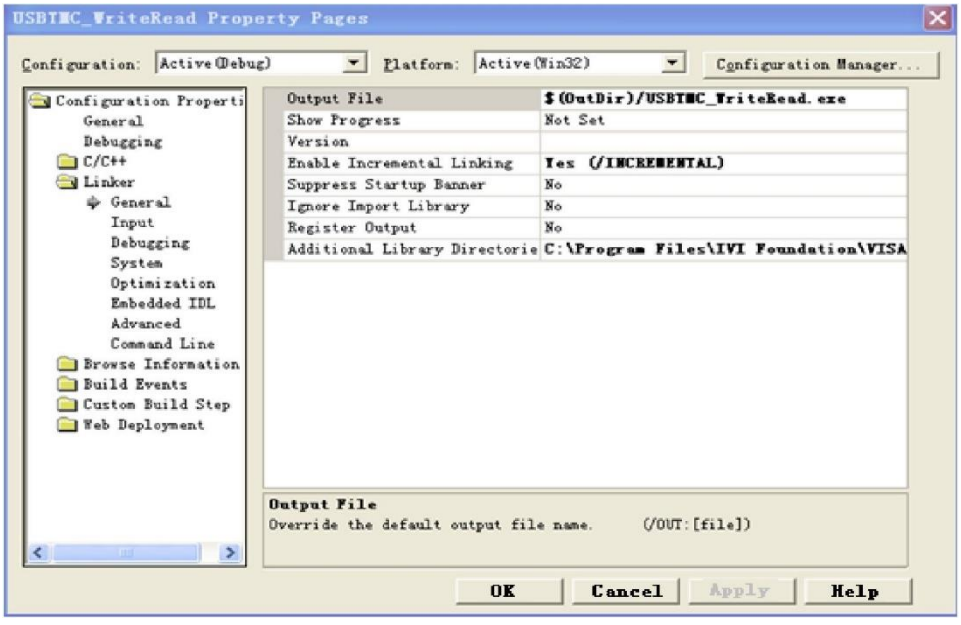
#pragma comment(lib,"visa32.lib")

b) Dynamic library

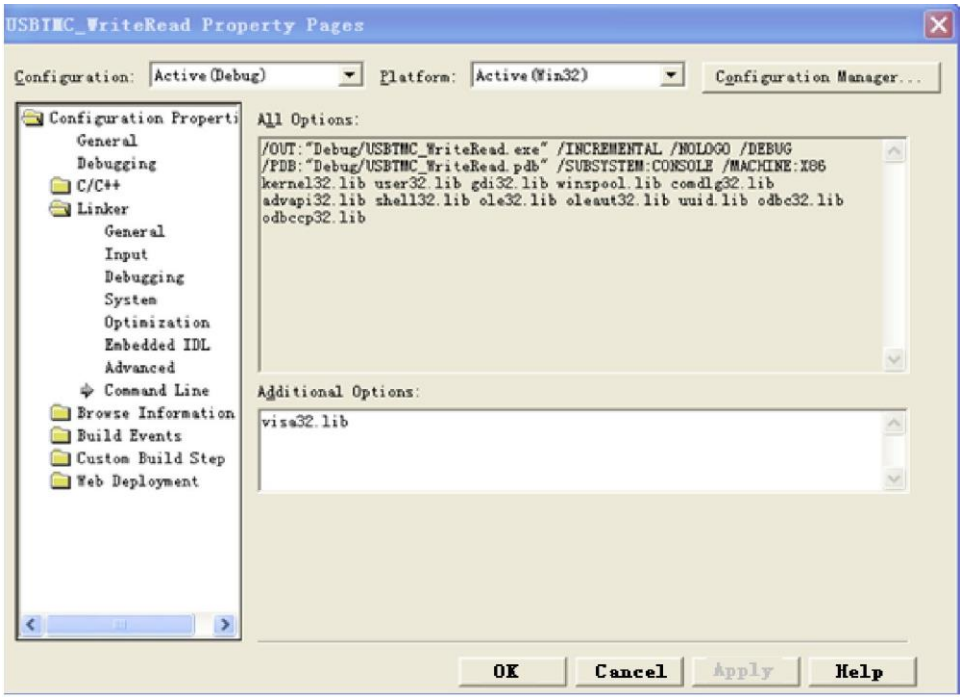
Press “project>>properties”, select"c/c++---General" in attribute dialog on the leftside, set the value of "Additional Include Directories" as the installment path of NI-VIS (such as C:\ProgramFiles\IVI Foundation\VISA\WinNT\include), as shown in the following figure.



Select "Linker-General" in attribute dialog on the left side, set the value of "Additional Library Directories" as the installment path of NI-VIS (such as C:\Program Files\IVI Foundation\VISA\WinNT\include), as shown in the following figure.



Select "Linker-Command Line "in attribute dialog on the left side, set the value of "Additional" as visa32.lib, as shown in the following figure.



Add the file visa.h in projectname.cpp file.

#include <visa.h>

1. Source code

a) USBTMC Example

```
int usbtmc_test()
{ /** This code demonstrates sending synchronous read & write commands
    * to an USB Test & Measurement Class (USBTMC) instrument using NI-VISA
    * The example writes the "*IDN?\n" string to all the USBTMC
    * devices connected to the system and attempts to read back
    * results using the write and read functions.
    * Open Resource Manager
    * Open VISA Session to an Instrument
    * Write the Identification Query Using viPrintf
    * Try to Read a Response With viScanf
    * Close the VISA Session*/
    ViSession defaultRM;
    ViSession instr;
    ViUInt32 numInstrs;
    ViFindList findList;
    ViStatus status;
    char instrResourceString[VI_FIND_BUFLEN];
    unsigned char buffer[100];
```

```
int i;
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
        printf("Could not open a session to the VISA Resource Manager!\n");
        return status;
}
/*Find all the USB TMC VISA resources in our system and store the number of resources in the system in
numInstrs.*/
status = viFindRsrc(defaultRM, "USB?*INSTR", &findList, &numInstrs, instrResourceString);
if (status<VI_SUCCESS)
{
        printf("An error occurred while finding resources. \nPress Enter to continue.");
        fflush(stdin);
        getchar();
        viClose(defaultRM);
        return status;
}
/** Now we will open VISA sessions to all USB TMC instruments.
*       We must use the handle from viOpenDefaultRM and we must
*       also use a string that indicates which instrument to open. This
*       is called the instrument descriptor. The format for this string
*       can be found in the function panel by right clicking on the
*       descriptor parameter. After opening a session to the
*       device, we will get a handle to the instrument which we
*       will use in later VISA functions. The AccessMode and Timeout
*       parameters in this function are reserved for future
*       functionality. These two parameters are given the value VI_NULL. */
for (i = 0; i < int(numInstrs); i++)
{
        if (i > 0)
        {
                viFindNext(findList, instrResourceString);
        }
        status = viOpen(defaultRM, instrResourceString, VI_NULL, VI_NULL, &instr);
        if (status < VI_SUCCESS)
        {
                printf("Cannot open a session to the device %d. \n", i + 1);
                continue;
        }
        /** At this point we now have a session open to the USB TMC instrument.
        *We will now use the viPrintf function to send the device the string "*IDN?\n",
        *asking for the device's identification. */
        char * cmmand = "*IDN?\n";
```

```
            status = viPrintf(instr, cmmand);
            if (status < VI_SUCCESS)
            {
                printf("Error writing to the device %d. \n", i + 1);
                status = viClose(instr);
                continue;
            }
            /** Now we will attempt to read back a response from the device to
            *the identification query that was sent. We will use the viScanf
            *function to acquire the data.
            *After the data has been read the response is displayed. */
            status = viScanf(instr, "%t", buffer);
            if (status < VI_SUCCESS)
            {
                printf("Error reading a response from the device %d. \n", i + 1);
            }
            else
            {
                printf("\nDevice %d: %s\n", i + 1, buffer);
            }
            status = viClose(instr);
        }
        /*Now we will close the session to the instrument using viClose. This operation frees all
        system resources.*/
        status = viClose(defaultRM);
        printf("Press Enter to exit.");
        fflush(stdin);
        getchar();
        return 0;
}

int _tmain(int argc, _TCHAR* argv[ ])
{
        usbtmc_test();
        return 0;
}
```

b) TCP/IP Example

```
int tcp_ip_test(char *pIP)
{
        char outputBuffer[ VI_FIND_BUFLEN ];
        ViSession defaultRM, instr;
        ViStatus status;
        /* First we will need to open the default resource manager. */
```

```
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        printf("Could not open a session to the VISA Resource Manager!\n");
    }
    /* Now we will open a session via TCP/IP device */
    char head[256] = "TCPIP0::";
    char tail[] = "::inst0::INSTR";
    strcat(head, pIP);
    strcat(head, tail);
    status = viOpen(defaultRM, head, VI_LOAD_CONFIG, VI_NULL, &instr);
    if (status < VI_SUCCESS)
    {
        printf("An error occurred opening the session\n");
        viClose(defaultRM);
    }
    status = viPrintf(instr, "*idn?\n");
    status = viScanf(instr, "%t", outputBuffer);
    if (status < VI_SUCCESS)
    {
        printf("viRead failed with error code: %x \n", status);
        viClose(defaultRM);
    }
    else
    {
        printf("\nMesseage read from device: %*s\n", 0, outputBuffer);
    }
    status = viClose(instr);
    status = viClose(defaultRM);
    printf("Press Enter to exit.");
    fflush(stdin);
    getchar();
    return 0;
}
int _tmain(int argc, _TCHAR* argv[])
{
    printf("Please input IP address:");
    char ip[256];
    fflush(stdin);
    gets(ip);
    tcp_ip_test(ip);
    return 0;
}
```

C# Example

- Environment: Window system, visual studio.
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps

1. Open the visual studio software to create a new C# console project.
2. Add C# quote Ivi.Visa.dll and NationalInstruments.Visa.dll of VISA.
3. Source code

a) USBTMC Example

```
class Program
{
    void usbtmc_test()
    {
        using (var rmSession = new ResourceManager())
        {
            var resources = rmSession.Find("USB?*INSTR");
            foreach (string s in resources)
            {
                try
                {
                    var mbSession = (MessageBasedSession)rmSession.Open(s);
                    mbSession.RawIO.Write("*IDN?\n");
                    System.Console.WriteLine(mbSession.RawIO.ReadString());
                }
                catch (Exception ex)
                {
                    System.Console.WriteLine(ex.Message);
                }
            }
        }

        void Main(string[ ] args)
        {
            usbtmc_test();
        }
}
```

b) TCP/IP Example

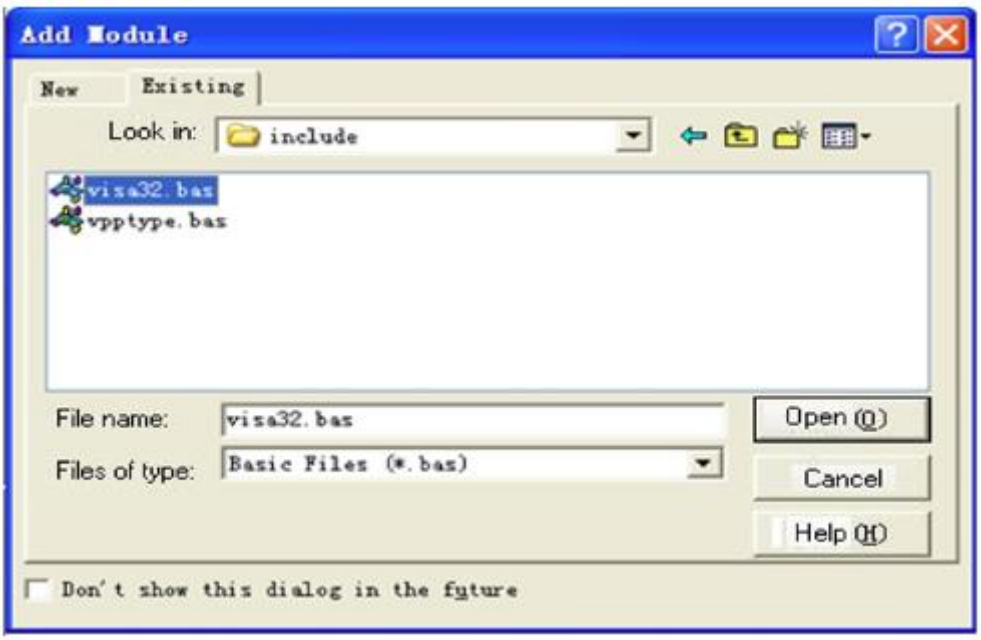
```
class Program
```

```
{
    void tcp_ip_test(string ip)
    {
        using (var rmSession = new ResourceManager())
        {
            try
            {
                var resource = string.Format("TCPIP0::{0}::inst0::INSTR", ip);
                var mbSession = (MessageBasedSession)rmSession.Open(resource);
                mbSession.RawIO.Write("*IDN?\n");
                System.Console.WriteLine(mbSession.RawIO.ReadString());
            }
            catch (Exception ex)
            {
                System.Console.WriteLine(ex.Message);
            }
        }
    }

    void Main(string[ ] args)
    {
        tcp_ip_test("192.168.20.11");
    }
}
```

VB Example

- Environment: Window system, Microsoft visual basic 6.0.
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps

1. Open the visual basic software and create a new standard application program project.
2. Set the project environment that can adjust NI-VISA library, press Existing tab of Project>>Add Existing Item, find the visa32.bas file in the "include" folder under the NI-VISA installation path and add it, as shown in the following figure.



3. Source code

a) USBTMC Example

```
PrivateFunction usbtmc_test() AsLong
' This code demonstrates sending synchronous read & write commands
' to an USB Test & Measurement Class (USBTMC) instrument using NI-VISA
' The example writes the "*IDN?\n" string to all the USBTMC
' devices connected to the system and attempts to read back
' results using the write and read functions.
' The general flow of the code is
' Open Resource Manager
' Open VISA Session to an Instrument
' Write the Identification Query Using viWrite
' Try to Read a Response With viRead
' Close the VISA Session

Const MAX_CNT = 200
Dim defaultRM AsLong
Dim instrsesn AsLong
Dim numInstrs AsLong
Dim findList AsLong
Dim retCount AsLong
Dim status AsLong
Dim instrResourceString AsString *VI_FIND_BUFLEN
Dim Buffer AsString * MAX_CNT
Dim i AsInteger
```

```
' First we must call viOpenDefaultRM to get the manager
' handle. We will store this handle in defaultRM.
status = viOpenDefaultRM(defaultRM)
If(status < VI_SUCCESS) Then
    resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
    usbtmc_test = status
ExitFunction
EndIf

' Find all the USB TMC VISA resources in our system and store the
' number of resources in the system in numInstrs.
status = viFindRsrc(defaultRM, "USB?*INSTR", findList, numInstrs, instrResourceString)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "An error occurred while finding resources."
    viClose(defaultRM)
    usbtmc_test = status
ExitFunction
EndIf

' Now we will open VISA sessions to all USB TMC instruments.
' We must use the handle from viOpenDefaultRM and we must
' also use a string that indicates which instrument to open.  This
' is called the instrument descriptor.  The format for this string
' can be found in the function panel by right clicking on the
' descriptor parameter. After opening a session to the
' device, we will get a handle to the instrument which we
' will use in later VISA functions.  The AccessMode and Timeout
' parameters in this function are reserved for future
' functionality.  These two parameters are given the value VI_NULL.
For i = 0 To numInstrs
If (i > 0) Then
    status = viFindNext(findList, instrResourceString)
EndIf
    status = viOpen(defaultRM, instrResourceString, VI_NULL, VI_NULL, instrsesn)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Cannot open a session to the device " + CStr(i + 1)
GoTo NextFind
EndIf

' At this point we now have a session open to the USB TMC instrument.
' We will now use the viWrite function to send the device the string "*IDN?",
```

```
' asking for the device's identification.
status = viWrite(instrsesn, "*IDN?", 5, retCount)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error writing to the device."
    status = viClose(instrsesn)
GoTo NextFind
EndIf

' Now we will attempt to read back a response from the device to
' the identification query that was sent.  We will use the viRead
' function to acquire the data.
' After the data has been read the response is displayed.
status = viRead(instrsesn, Buffer, MAX_CNT, retCount)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error reading a response from the device." + CStr(i + 1)
Else
    resultTxt.Text = "Read from device: " + CStr(i + 1) + " " + Buffer
EndIf
    status = viClose(instrsesn)
Next i

' Now we will close the session to the instrument using
' viClose. This operation frees all system resources.
status = viClose(defaultRM)
usbtmc_test = 0
EndFunction
```

b) TCP/IP Example

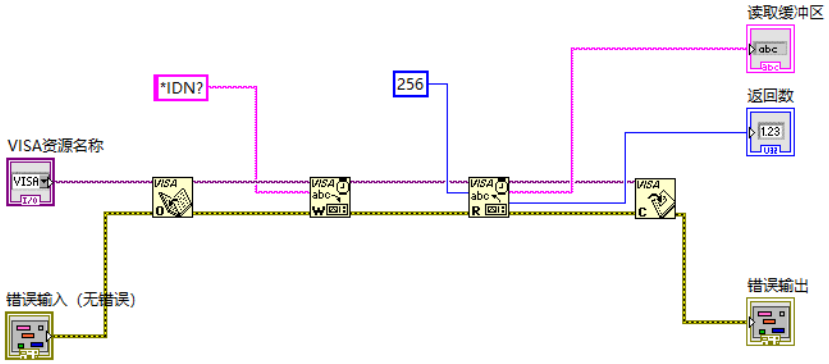
```
PrivateFunction tcp_ip_test(ByVal ip AsString) AsLong
Dim outputBuffer AsString * VI_FIND_BUFLEN
Dim defaultRM AsLong
Dim instrsesn AsLong
Dim status AsLong
Dim count AsLong

' First we will need to open the default resource manager.
status = viOpenDefaultRM(defaultRM)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
    tcp_ip_test = status
ExitFunction
EndIf

' Now we will open a session via TCP/IP device
```

```
status = viOpen(defaultRM, "TCPIP0::" + ip + "::inst0::INSTR", VI_LOAD_CONFIG, VI_NULL, instrsesn)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "An error occurred opening the session"
    viClose(defaultRM)
    tcp_ip_test = status
ExitFunction
EndIf
status = viWrite(instrsesn, "*IDN?", 5, count)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error writing to the device."
EndIf
    status = viRead(instrsesn, outputBuffer, VI_FIND_BUFLEN, count)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error reading a response from the device." + CStr(i + 1)
Else
    resultTxt.Text = "read from device:" + outputBuffer
EndIf
    status = viClose(instrsesn)
    status = viClose(defaultRM)
    tcp_ip_test = 0
EndFunction
```

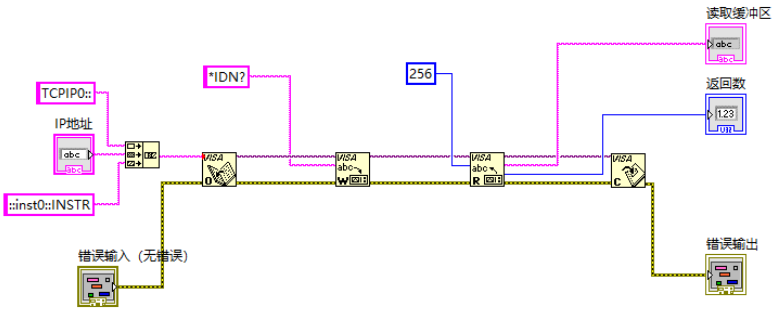
LabVIEW Example

- Environment: Window system, Lab VIEW.
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps

1. Open the Lab VIEW software to create a VI file.
2. Add the control, click the front panel to select and add the VISA source name, error input, error output, and part of the indicator from the control column.
3. Open the diagram, click the VISA source name, and then select and add these functions VISA Write, VISA Read, VISA Open and VISA Close on the VISA menu.
4. The VI opens a VISA session for a USBTMC device and writes the *IDN? command to the device and reads back the response value. When all communication is complete, the VI closes the VISA session as shown in the following figure.



5. Communication with the device via TCP/IP is similar with USBTMC, it need to set VISA write and read function to synchronous I/O, set Lab VIEW to asynchronous IO by default. Right click on the node and select "Synchronous I/O Mode>>Synchronous" from shortcut menu to enable synchronous writing or reading of data, as shown in the following figure.



MATLAB Example

- Environment: Window system, MATLAB.
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps

1. Open the MATLAB software, click File>>New>>Script on Matlab interface to create an empty M file.
2. Source code

a) USBTMC Example

```
function usbtmc_test()
% This code demonstrates sending synchronous read & write commands
% to an USB Test & Measurement Class (USBTMC) instrument using
% NI-VISA
```

```
%Create a VISA-USB object connected to a USB instrument
vu = visa('ni','USB0::0x5345::0x1234::SN20220718::INSTR');

%Open the VISA object created
fopen(vu);

%Send the string "*IDN?",asking for the device's identification.
fprintf(vu,'*IDN?');

%Request the data

outputbuffer = fscanf(vu);
disp(outputbuffer);

%Close the VISA object
fclose(vu);
delete(vu);
clear vu;

end
```

b) TCP/IP Example

```
function tcp_ip_test()
% This code demonstrates sending synchronous read & write commands
% to an TCP/IP instrument using NI-VISA
%Create a VISA-TCPIP object connected to an instrument

%configured with IP address.
vt = visa('ni',['TCPIP0::','192.168.20.11','::inst0::INSTR']);

%Open the VISA object created

fopen(vt);

%Send the string "*IDN?",asking for the device's identification.
fprintf(vt,'*IDN?');

%Request the data
outputbuffer = fscanf(vt);
```

```
disp(outputbuffer);

%Close the VISA object
fclose(vt);
delete(vt);
clear vt;

end
```

Python Example

- Environment: Window system, Python3.8, PyVISA 1.11.0.
- Description: Access the instrument via USBTMC and TCP/IP, and send "*IDN?" command on NI-VISA to query the device information.
- Steps

1. Install the Python at first, then turn on the Python to create an empty file test.py.
2. Use the command pip install PyVISA to install PyVISA. If it cannot be installed, please refer to this link (https://pyvisa.readthedocs.io/en/latest/)
3. Source code

a) USBTMC Example

```
import pyvisa
rm = pyvisa.ResourceManager()
rm.list_resources()
my_instrument = rm.open_resource('USB0::0x5345::0x1234::SN20220718::INSTR')
print(my_instrument.query('*IDN?'))
```

b) TCP/TP Example

```
import pyvisa
rm = pyvisa.ResourceManager()
rm.list_resources()
my_instrument = rm.open_resource('TCPIP0::192.168.20.11::inst0::INSTR')
print(my_instrument.query('*IDN?'))
```

Programming Application Example

Bus Trigger Measurement

A complete flow of a bus-triggered DC voltage measurement, include the following commands.

```
CONF:VOLT:DC 200V        # Switch to the DC voltage measurement and set the manual range to 200V
TRIG:SOUR BUS            # Set the trigger source to bus trigger
TRIG:COUN 5              # Set the trigger count to 5
SAMP:COUN 10             # Set the sampling count to 10
INIT                     # Set the trigger state to wait to trigger
*TRG                     # Send the trigger signal
FETCh?                   # Read the measured number, it returns 5*10
```

Acquire Measured Statistical Result

```
CONF:FREQ                # Switch to the frequency mode, and the default is AUTO
SAMP:COUN 100            # Sampling the measured value of 100
CALC:AVER:STAT ON        # Turn on the statistical function
INIT                     # Enable Sampling instantly
CALC:AVER:ALL?           # Acquire the sampling result
```

SENSe Command to Configure the Parameter for Measurement

```
FUNC "CURRent:DC"        # Select the DC current mode
CURRent:DC:RANGe 0.02    # Set the manual range to 20mA
CURR:AC:NULL:VALue 0.001 # Relative value of 1mA
CURR:AC:NULL:STAT ON     # Turn on the relative value
SAMP:COUN 2              # Sampling 2 measured value
READ?                    # Read the measured value
FUNC "TEMP"              # Select the temperature mode
TEMP:TRAN:TYPE RTD       # Select the 2-wire RTD probe
TEMP:TRAN:FRTD:RES 100   # Configure the nominal resistance (R0) value to 100Ω
SAMP:COUN 2              # Sampling 2 measured value
READ?                    # Read the measured value
```

Appendix 1: IEEE 488.2 Binary Data Format

DATA is data flow, other is ASCII character, as shown in the following figure <#812345678 + DATA + \n>.

Start Mark (1Byte)	Bit Width (1Byte)	Total Length of Data (Bit Width Byte)	DATA (n Byte)	End Mark (1Byte)
#	x	xxxxxxxx	.....	\n