

UT8805A五位半台式万用表

SCPI编程手册

REV 00
2023. 06

UNI-T[®]

保证和声明

版权

2022 优利德中国科技有限公司

商标信息

UNI-T 是优利德中国科技有限公司的注册商标。

文档编号

软件版本

00.00.01

软件升级可能更改或增加产品功能，请关注 **UNI-T** 网站获取最新版本手册或联系 **UNI-T** 升级软件。

声明

- 本产品受中国及其它国家和地区的专利（包括已取得的和正在申请的专利）保护。
- 本公司保留改变规格及价格的权利。
- 本手册提供的信息取代以往出版的所有资料。
- 本手册提供的信息如有变更，恕不另行通知。
- 对于本手册可能包含的错误，或因手册所提供的信息及演绎的功能以及因使用本手册而导致的任何偶然或继发的损失，**UNI-T** 概不负责。
- 未经 **UNI-T** 事先书面许可，不得影印、复制或改编本手册的任何部分。
-

产品认证

UNI-T 认证本产品符合中国国家产品标准和行业产品标准及 ISO9001: 2008 标准和 ISO14001: 2004 标准，并进一步认证本产品符合其它国际标准组织成员的相关标准。

SCPI 指令简介

SCPI (Standard Commands for Programmable Instruments, 即可编程仪器标准命令集) 是一种建立在现有标准 IEEE 488.1 和 IEEE 488.2 基础上, 并遵循了 IEEE754 标准中浮点运算规则、ISO646 信息交换 7 位编码符号 (相当于 ASCII 编程) 等多种标准的标准化仪器编程语言。本节简介 SCPI 命令的格式、符号、参数和缩写规则。

指令格式

SCPI 命令为树状层次结构, 包括多个子系统, 每个子系统由一个根关键字和一个或数个层次关键字构成。命令行通常以冒号“:”开始; 关键字之间用冒号“:”分隔, 关键字后面跟随可选的参数设置。命令关键字和第一个参数之间以空格分开。命令字符串必须以一个 <换行> (<NL>) 字符结尾。命令行后面添加问号“?”通常表示对此功能进行查询。

符号说明

下面四种符号不是 SCPI 命令中的内容, 不随命令发送, 但是通常用于辅助说明命令中的参数。

- **大括号 { }**
大括号中通常包含多个可选参数, 发送命令时必须选择其中一个参数。如: DISPLAY:GRID:MODE { FULL | GRID | CROSS | NONE}命令。
- **竖线 |**
竖线用于分隔多个参数选项, 发送命令时必须选择其中一个参数。
如: DISPLAY:GRID:MODE { FULL | GRID | CROSS | NONE}命令。
- **方括号 []**
方括号中的内容 (命令关键字) 是可省略的。如果省略参数, 仪器将该参数设置为默认值。例如: 对于: MEASure:NDUTy? [<source>]命令, [<source>]表示当前通道。
- **三角括号 < >**
三角括号中的参数必须用一个有效值来替换。例如: 以 DISPLAY:GRID:BRIGhtness 30 的形式发送 DISPLAY:GRID:BRIGhtness <count>命令。

参数说明

本手册介绍的命令中所含的参数可以分为以下 5 种类型: 布尔型、整型、实型、离散型、ASCII 字符串。

- **布尔型**

参数取值为“ON”（1）或“OFF”（0）。例如：`:SYSTem:LOCK {{1 | ON} | {0 | OFF}}`。

- **整型**

除非另有说明，参数在有效值范围内可以取任意整数值。注意：此时，请不要设置参数为小数格式，否则将出现异常。例如：`:DISPlay:GRID:BRIGhtness <count>`命令中的参数`< count >`可取 0 到 100 范围内的任一整数。

- **实型**

除非另有说明，参数在有效值范围内可以取任意值。

例如：对于 CH1，`CHANnel1:OFFSet <offset>`命令中的参数`<offset>`的取值为实型。

- **离散型**

参数只能取指定的几个数值或字符。例如：`:DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE}`命令的参数只能为 FULL、GRID、CROSS、NONE。

- **ASCII 字符串**

字符串参数实际上可包含所有 ASCII 字符集。字符串必须以配对的引号开始和结尾；可以用单引号或双引号。引号分隔符也可以作为字符串的一部分，只需键入两次并且不在中间添加任何字符，例如设置 IP：`SYST:COMM:LAN:IPAD "192.168.1.10"`。

简写规则

所有命令对大小写都能识别，可以全部采用大写或小写。如果要缩写，必须输完命令格式中的所有大写字母。

数据返回

数据返回分为单个数据和批量数据返回，单个数据返回相对应的参数类型，其中实型返回用科学计数法表示，e 前部分小数点后面保留三位数据，e 部分保留三位数据；批量数据返回必须符合 IEEE 488.2 #格式的字符串数据，其格式：`'#+ 长度所占的字符位数[固定为一个字符] + 有效数据长度的 ASCII 值 + 有效数据 + 结束符['\n']`，例如`#3123xxxxxxxxxxxxxxxxxxx\n`表示的具有 123 个字节有效批量数据返回格式，其中‘3’表示“123”占 3 个字符位。

注意：当返回数据为无效数据均用*表示。

无。

■ **举例：**

*RCL 1

从内部闪存文件系统根文件夹中的 STATE_1.sta a 文件调用仪器状态

***SAV**

■ **命令格式：**

*SAV {0 | 1 | 2 | 3 | 4}

■ **功能描述：**

用于将当前仪器状态存储到内部闪存文件系统根文件夹中的状态文件 STATE_<n>.sta 中，其中 <n> 为指定的数字。

■ **返回格式：**

无。

■ **举例：**

*SAV 1

仪器状态存储到内部闪存文件系统根文件夹中的状态文件 STATE_1.sta 中

***TRG**

■ **命令格式：**

*TRG

■ **功能描述：**

用于软件触发测量。

■ **返回格式：**

无。

■ **举例：**

CONF:VOLT:DC

SAMP:COUN 5

TRIG:SOUR BUS

INIT

仪器置于“等待触发”状态

*TRG

触发测量

测量控制相关命令

用于控制万用表测量终止、测量启动、测量等待、获取测量结果、信号输出等操作。

:ABORt

- **命令格式:**

ABORt

- **功能描述:**

终止正在进行的测量，将仪器返回到“触发空闲”状态。

- **返回格式:**

无。

- **举例:**

TRIG:SOUR IMM

设置触发源为立即触发

TRIG:COUN 10

设置触发次数为 10 次

INIT

将触发状态设置为“等待触发”

ABOR

中断该次测量

:FETCh?

- **命令格式:**

FETCh?

- **功能描述:**

等待最近一次测量完成，并将所有可用的测量数据返回。

- **返回格式:**

可用的测量数据。

- **举例:**

TRIG:SOUR IMM

设置触发源为立即触发

TRIG:COUN 10

设置触发次数为 10 次

INIT

将触发状态设置为“等待触发”

FETC?

返回测量数据:

-5.75122019E-04, -5.77518360E-04

:INITiate[:IMMediate]

- **命令格式:**

INITiate[:IMMediate]

- **功能描述:**

将触发系统状态置为“等待触发”状态。

■ **返回格式:**

无。

■ **举例:**

TRIG:SOUR BUS	设置触发源为总线触发
TRIG:COUN 10	设置触发次数为 10 次
INITiate	将触发状态设置为“等待触发”
*TRG	发送触发信号
FETC?	返回测量数据: -5.75122019E-04, -5.77518360E-04

:OUTPut:TRIGger:SLOPe

■ **命令格式:**

OUTPut:TRIGger:SLOPe {POSitive|NEGative}

OUTPut:TRIGger:SLOPe?

■ **功能描述:**

设置在后面板 VM Comp BNC 连接器上 voltmeter complete 输出信号的极性。

■ **返回格式:**

查询返回 VMC 输出脉冲信号的极性。

■ **举例:**

OUTPut:TRIGger:SLOPe POS	VMC 输出正脉冲信号
OUTPut:TRIGger:SLOPe?	查询返回 POS

:READ?

■ **命令格式:**

READ?

■ **功能描述:**

读取仪器缓存的测量结果，并擦除测量结果。

■ **返回格式:**

当前缓存的测量数据。

■ **举例:**

TRIG:COUN 4	设置触发 4 次测量
SAMP:COUN 1	每次测量采集 1 个数据
READ?	读取返回: -1.23006735E-03,-1.30991641E-03,-1.32756530E-03,-1.32002814E-03

:SAMPLE:COUNt

- **命令格式:**

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

SAMPlE:COUNt?

- **功能描述:**

指定仪器单次触发所采集的样本数量，设置值 1（默认）~100,000。

- **返回格式:**

查询返回单次测量所采集的样本数。

- **举例:**

TRIG:COUN 1 设置触发 1 次测量

SAMP:COUN 10 每次测量采集 10 个样本数据

SAMPlE:COUNt? 查询返回 10

:UNIT:TEMPerature {C|F|K}

- **命令格式:**

UNIT:TEMPerature {C|F|K}

UNIT:TEMPerature?

- **功能描述:**

设置温度测量时使用的单位（° C、° F 或 Kelvin）。

- **返回格式:**

查询返回温度测量值使用的单位。

- **举例:**

UNIT:TEMP F 设置温度测量值使用的单位为 ° F

UNIT:TEMP? 查询返回: F

*UNREMOTE

- **命令格式:**

*UNREMOTE

- **功能描述:**

从远程模式返回本地模式，解锁前面板按键。

- **返回格式:**

无。

:CALCulate:LIMit: UPPer

■ 命令格式:

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

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

■ 功能描述:

设置限值模式的上限值，范围+1.0E-15 至 1.0E+15

■ 返回格式:

查询返回限值模式的上限值。

■ 举例:

CALCulate:LIMit:LOWer +1.0E+15 设置限值模式的下限值 +1.0E+15

CALCulate:LIMit:LOWer? 查询返回: +1.0E+15

:CALCulate:LIMit:STATe

■ 命令格式:

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

CALCulate:LIMit: STATe?

■ 功能描述:

启用或禁用限值测试

■ 返回格式:

查询返回限值测试使能状态。

■ 举例:

CALCulate:LIMit: STATe ON 启动限值测试

CALCulate:LIMit:LOWer? 查询返回: ON

:CALCulate:TRANSform:HISTogram:ALL

■ 命令格式:

CALCulate:TRANSform:HISTogram:ALL?

■ 功能描述:

查询直方图上限量程、下限量程值、测量次数以及自从上次清除直方图数据以来所收集的柱体数据。

■ 返回格式:

上限量程值，下限量程值，测量次数，柱体数据（柱形数、低于下限的测量次数、各柱形对应的测量次数、高于上限的测量次数）。

■ 举例:

CALCulate:TRANSform:HISTogram:ALL? 查询返回:
-1.85337982E+08,+1.60497704E+07,39,12,0,1,0,3,
3,4,4,7,8, 5,4,0

:CALCulate:TRANSform:HISTogram:DATA

■ 命令格式:

CALCulate:TRANSform:HISTogram: DATA?

■ 功能描述:

查询直方图收集到的柱体数据。

■ 返回格式:

柱形数、低于下限的测量次数、各柱形对应的测量次数、高于上限的测量次数。

■ 举例:

CALCulate:TRANSform:HISTogram: DATA?

查询返回:

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

:CALCulate:TRANSform:HISTogram:COUNT

■ 命令格式:

CALCulate:TRANSform:HISTogram: COUNT?

■ 功能描述:

查询自从上次清除直方图数据以来收集到的测量结果数。

■ 返回格式:

十进制测量结果数。

■ 举例:

CALCulate:TRANSform:HISTogram: COUN?

查询返回: +96

:CALCulate:TRANSform:HISTogram:POINTS

■ 命令格式:

CALCulate:TRANSform:HISTogram: POINTs {<value>|MIN|MAX|DEF}?

CALCulate:TRANSform:HISTogram: POINTs?

■ 功能描述:

设置直方图下限量程值与上限量程值之间的柱形数{10/20/40/100/200/400}。始终存在两个附加柱形: 一个用于低于下限量程的测量, 一个用于高于上限量程的测量。

■ 返回格式:

十进制柱形数。

■ 举例:

CALCulate:TRANSform:HISTogram: POINT 100

设置直方图柱形数

CALCulate:TRANSform:HISTogram: POINT?

查询返回: +100

:CALCulate:TRANSform:HISTogram:RANGe:AUTO

■ 命令格式:

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

CALCulate:TRANSform:HISTogram:RANGe:AUTO?

■ **功能描述:**

启用或禁用自动选择直方图的下限和上限量程值。

■ **返回格式:**

0 (OFF) 或 1 (ON)。

■ **举例:**

CALCulate:TRANSform:HISTogram: RANGe:AUTO ON 启用自动选择直方图的下限和上限量程值

CALCulate:TRANSform:HISTogram: RANGe:AUTO? 查询返回: 1, 启用自选上下限值

:CALCulate:TRANSform:HISTogram:RANGe:LOWer

■ **命令格式:**

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

CALCulate:TRANSform:HISTogram:RANGe:LOWer?

■ **功能描述:**

设置直方图的下限量程值, 同时会禁用自动选择直方图的下限和上限量程值功能。

■ **返回格式:**

下限量程值, -1.0E+15 至-1.0E-15。

■ **举例:**

CALCulate:TRANSform:HISTogram: RANGe: LOWer +1.00000000E+06 设置下限量程值

CALCulate:TRANSform:HISTogram: RANGe: LOWer? 查询返回:
+1.00000000E+06

:CALCulate:TRANSform:HISTogram:RANGe:UPPer

■ **命令格式:**

CALCulate:TRANSform:HISTogram:RANGe:UPper {<value>|MIN|MAX|DEF}

CALCulate:TRANSform:HISTogram:RANGe: UPper?

■ **功能描述:**

设置直方图的上限量程值, 同时会禁用自动选择直方图的下限和上限量程值功能。

■ **返回格式:**

上限量程值, +1.0E-15 至 1.0E+15。

■ **举例:**

CALCulate:TRANSform:HISTogram: RANGe: LOWer +1.00000000E+06 设置上限量程值

CALCulate:TRANSform:HISTogram: RANGe: LOWer? 查询返回:
+1.00000000E+06

:CALCulate:TRANSform:HISTogram:STATe

■ 命令格式:

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

CALCulate:TRANSform:HISTogram[:STATe]?

■ 功能描述:

启用或禁用直方图计算。

■ 返回格式:

0 (OFF) 或 1 (ON)。

■ 举例:

CALCulate:TRANSform:HISTogram:STATe ON 启用直方图计算

CALCulate:TRANSform:HISTogram:STATe? 查询返回: 1

:CALCulate:SCALE:DB:REFerence

■ 命令格式:

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

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

■ 功能描述:

设置 DB 标定的参考值, 范围-200.0 dB 至 +200.0 dB。

■ 返回格式:

查询返回 DB 标定的参考值。

■ 举例:

CALCulate:SCALE:DB:REFerence -10.0 设置 DB 标定的参考值 -10.0dB

CALCulate:SCALE:DB:REFerence? 查询返回: -10.0

:CALCulate:SCALE:DBM:REFerence

■ 命令格式:

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

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

■ 功能描述:

设置电压测量启动 dBm 定标时使用的电阻参考值, 可选 50、75、93、110、124、125、135、150、250、300、500、600、800、900、1000、1200 或 8000 Ω。此参考值影响 dBm 和 dB 标定。

■ 返回格式:

查询返回 dBm 定标时使用的电阻参考值。

■ 举例:

CALCulate:SCALE:DBM:REF 300 用 300 Ω 参考电阻启动 dBm 标定

CALCulate:SCALE:DBM:REF? 查询返回: 300

:CALCulate:SCALE:FUNCTION

■ 命令格式:

CALCulate:SCALE:FUNCTION {DB|DBM}

CALCulate:SCALE:FUNCTION?

■ 功能描述:

选择标定执行运算所使用的标定函数。

■ 返回格式:

查询返回标定函数，DB 或 DBM。

■ 举例:

CALCulate:SCALE: FUNC DBM 选择 DBM 标定函数

CALCulate:SCALE: FUNC? 查询返回: DBM

:CALCulate:SCALE:REFERENCE:AUTO

■ 命令格式:

CALCulate:SCALE:REFERENCE:AUTO {ON|1|OFF|0}

CALCulate:SCALE:REFERENCE:AUTO?

■ 功能描述:

启用或禁用 dB 标定函数自动选择参考值的功能。启用时，第一个测量值将用作所有后续测量的参考。

■ 返回格式:

查询返回 dB 标定函数自动选择参考值的功能状态，0 (OFF) 或 1 (ON)。

■ 举例:

CALCulate:SCALE:REF:AUTO ON dB 标定函数自动选择参考值的功能开启

CALCulate:SCALE:REF:AUTO? 查询返回: 1

:CALCulate:SCALE[:STATe]

■ 命令格式:

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

CALCulate:SCALE[:STATe]?

■ 功能描述:

启用或禁用标定函数。

■ 返回格式:

查询返回标定函数的功能状态，0 (OFF) 或 1 (ON)。

■ 举例:

CALC:SCAL:STAT ON 启用标定函数

CALC:SCAL:STAT? 查询返回: 1

:CALCulate:AVERage[:STATe]

- **命令格式:**

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

CALCulate:AVERage[:STATe]?

- **功能描述:**

启用或禁用统计运算。

- **返回格式:**

查询返回统计运算的功能状态, 0 (OFF) 或 1 (ON)。

- **举例:**

CALC: AVER:STAT ON

启用统计运算

CALC: AVER:STAT?

查询返回: 1

:CALCulate:AVERage:ALL

- **命令格式:**

CALCulate:AVERage:ALL?

- **功能描述:**

查询返回统计运算的结果, 包括算术平均值(平均值)、标准差、最大值和最小值。

- **返回格式:**

算术平均值,标准差,最大值,最小值, 均为+1.00000000E+01 形式。

- **举例:**

CALCulate:AVERage:ALL?

查询返回:

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

E+03,+4.13500000E-01

:CALCulate:AVERage:MINimum

- **命令格式:**

CALCulate:AVERage:MINimum?

- **功能描述:**

查询返回统计运算结果之最小值。

- **返回格式:**

查询返回统计运算结果之最小值 (+1.00000000E+01 形式)。

- **举例:**

CALCulate:AVERage: MIN?

查询返回: +1.00520000E+03

:CALCulate:AVERage:MAXimum

- **命令格式:**

CALCulate:AVERage:MAXimum?

■ **功能描述:**

查询返回统计运算结果之最大值。

■ **返回格式:**

查询返回统计运算结果之最大值 (+1.00000000E+01 形式)。

■ **举例:**

CALCulate:AVERage:MAX?

查询返回: +1.00520000E+03

:CALCulate:AVERage:AVERage

■ **命令格式:**

CALCulate:AVERage:AVERage?

■ **功能描述:**

查询返回统计运算结果之平均值。

■ **返回格式:**

查询返回统计运算结果之平均值 (+1.00000000E+01 形式)。

■ **举例:**

CALCulate:AVERage:AVER?

查询返回: +1.00520000E+03

:CALCulate:AVERage:COUNT

■ **命令格式:**

CALCulate:AVERage:COUNT?

■ **功能描述:**

查询返回统计运算结果之计数值。

■ **返回格式:**

查询返回统计运算结果之计数值 (十进制)。

■ **举例:**

CALCulate:AVERage:COUN?

查询返回: +129

:CALCulate:AVERage:PTPeak

■ **命令格式:**

CALCulate:AVERage:PTPeak?

■ **功能描述:**

查询返回统计运算结果之峰-峰值。

■ **返回格式:**

查询返回统计运算结果之峰-峰值 (+1.00000000E+01 形式)。

■ **举例:**

CALCulate:AVERage:PTP?

查询返回: +1.00520000E+03

:CALCulate:AVERage:SDEViation

- **命令格式:**

CALCulate:AVERage:SDEViation?

- **功能描述:**

查询返回统计运算结果之标准差值。

- **返回格式:**

查询返回统计运算结果之标准差值（+1.00000000E+01 形式）。

- **举例:**

CALCulate:AVERage:SDE?

查询返回: +1.00520000E+03

:CALCulate:AVERage:CLEar[:IMMEDIATE]

- **命令格式:**

CALCulate:AVERage:CLEar[:IMMEDIATE]

- **功能描述:**

清除统计运算的统计信息，包括最小值、最大值、平均数、峰-峰、计数和标准偏差。

- **返回格式:**

无）。

- **举例:**

CALCulate:AVERage:CLE

清除统计信息

CONFigure 命令

用于配置电压、电流、电容、电阻、二极管、频率、周期、温度等挡位的测量量程，获取仪器当前的测量量程。

:CONFigure

- **命令格式:**

CONFigure?

- **功能描述:**

查询当前测量挡位和量程。

- **返回格式:**

查询返回: 测量挡位名称(如 CURR:AC),量程。

- **举例:**

CONFigure?

查询返回: VOLT:DC,+2.00000000E+01

:CONFigure:CONTinuity

- **命令格式:**

CONFigure:CONTinuity

- **功能描述:**

将通断模式下的所有测量参数和触发参数设置为其默认值。

- **返回格式:**

无。

- **举例:**

CONFigure:CONT

将通断模式下的所有测量参数和触发参数设置为其默认值

:CONFigure:DIODE

- **命令格式:**

CONFigure:DIODE

- **功能描述:**

将二极管模式下的所有测量参数和触发参数设置为其默认值。

- **返回格式:**

无。

- **举例:**

CONFigure:DIOD

将二极管模式下的所有测量参数和触发参数设置为其默认值

:CONFigure:CURRent:DC

■ 命令格式:

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

■ 功能描述:

将直流电流测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200uA|2mA|20mA|200mA|2A|10A|AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure:CURR:DC	将直流电流模式下的所有测量参数和触发参数设置为其默认值
CONFigure:CURR:DC 2A	设置直流电流模式下量程为 2A

:CONFigure:CURRent:AC

■ 命令格式:

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

■ 功能描述:

将交流电流测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{20mA|200mA|2A|10A|AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure:CURR:AC	将直流电流模式下的所有测量参数和触发参数设置为其默认值
CONFigure:CURR:AC 20mA	设置交流电流模式下量程为 20mA

:CONFigure:FREQuency

■ 命令格式:

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

■ 功能描述:

将频率测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200 mV|2 V|20 V|200 V|750V|AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure:FREQ	将频率模式下的所有测量参数和触发参数设置为其默认值
CONFigure:FREQ 20V	设置频率模式下量程为 20V

:CONFigure:PERiod

■ 命令格式:

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

■ 功能描述:

将周期测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选 {200 mV|2 V|20 V|200 V|750V|AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure: PER	将周期模式下的所有测量参数和触发参数设置为其默认值
CONFigure: PER 20V	设置周期模式下量程为 20V

:CONFigure:RESistance

■ 命令格式:

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

■ 功能描述:

将二线电阻测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选 {200 Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|10 MΩ|100 MΩ|AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure: RES	将二线电阻模式下的所有测量参数和触发参数设置为其默认值
CONFigure: RES 200	设置二线电阻模式下量程为 200 Ω

:CONFigure:FRESistance

■ 命令格式:

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

■ 功能描述:

将四线电阻测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选 {200 Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure: FRES	将四线电阻模式下的所有测量参数和触发参数设置为其默认值
CONFigure: FRES 2000	设置四线电阻模式下量程为 2 kΩ

:CONFigure:TEMPerature

■ 命令格式:

CONFigure:TEMPerature [{FRTD|RTD|FTHermistor|THERmistor|TCouple|DEFault} [, {<type>|DEFault}]]

■ 功能描述:

将温度测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置探头及热电偶类型，默认探头为 RTD；只有 TCouple 探头可选择热电偶类型{ J|K|E|T|N|R|S|B }，其它探头类型默认即可。

■ 返回格式:

无。

■ 举例:

CONFigure:TEMP TCouple J 设置温度模式下使用 TCouple 探头,J 类型。

:CONFigure:VOLTage:DC

■ 命令格式:

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

■ 功能描述:

将直流电压测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200 mV|2 V|20 V|200 V|1000 V |AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure:VOLT:DC 将直流电压测量模式下所有测量参数和触发参数设置为其默认值。

CONFigure:VOLT:DC 200V 设置直流电压测量模式下量程为 200V

:CONFigure:VOLTage:AC

■ 命令格式:

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

■ 功能描述:

将交流电压测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200 mV|2 V|20 V|200 V|750V |AUTO}，默认值 AUTO。

■ 返回格式:

无。

■ 举例:

CONFigure:VOLT:AC 将交流电压测量模式下所有测量参数和触发参数设置为其默认值。

CONFigure:VOLT:AC 20V 设置交流电压测量模式下量程为 20V

:CONFigure:CAPacitance

- **命令格式:**

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

- **功能描述:**

将电容测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选 {2nF|20nF|200nF|2uF |20uF| 200uF|10000uF |AUTO}，默认值 AUTO。

- **返回格式:**

无。

- **举例:**

CONFigure:CAP

将电容测量模式下所有测量参数和触发参数设置为其默认值。

CONFigure:CAP 20uF

设置电容测量模式下量程为 20V

DATA 命令

用于管理仪器缓存的数据，包括返回指定数据、删除数据、查询数据缓存量等。

:DATA:POINTs

- **命令格式：**

DATA:POINTs?

- **功能描述：**

查询仪器缓存的测量总数，最多可缓存 1,000 个测量值。

- **返回格式：**

缓存的测量总数（十进制）。

- **举例：**

DATA:POINT?

查询返回：100

:DATA:LAST

- **命令格式：**

DATA: LAST?

- **功能描述：**

返回最近一次的测量值。

- **返回格式：**

测量值 单位。

- **举例：**

DATA: LAST?

查询返回：-4.79221344E-04 VDC

:DATA:REMOve

- **命令格式：**

DATA: REMove? <num_readings> [,WAIT]

- **功能描述：**

从仪器数据缓存中读取并擦除<num_readings>测量结果。如果测量结果少于<num_readings>，但指定了 WAIT 参数，则等待有<num_readings>测量结果再执行；否则，将返回一个错误。

- **返回格式：**

1~10000 个逗号分隔的测量结果，或错误码。

- **举例：**

DATA: REM? 5

查询返回：

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

:MEASure:CURRent:AC

- **命令格式:**

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

- **功能描述:**

将交流电流测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{20mA|200mA|2A|10A|AUTO}，默认值 AUTO，测量结果将直接返回。

- **返回格式:**

测量结果。

- **举例:**

MEASure:CURR:AC? 20mA 设置交流电流模式下量程为 20mA，并返回测量结果: +2.43186951E-02

:MEASure:FREQuency

- **命令格式:**

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

- **功能描述:**

将频率测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200 mV|2 V|20 V|200 V|750V|AUTO}，默认值 AUTO。

- **返回格式:**

测量结果。

- **举例:**

MEASure:FREQ? 20V 设置频率模式下量程为 20V，并返回测量结果: +2.43186951E-02

:MEASure:PERiod

- **命令格式:**

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

- **功能描述:**

将周期测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200 mV|2 V|20 V|200 V|750V|AUTO}，默认值 AUTO，测量结果将直接返回。

- **返回格式:**

测量结果。

- **举例:**

MEASure: PER? 20V 设置周期模式下量程为 20V，并返回测量结果: +2.43186951E-02

:MEASure:RESistance

- **命令格式:**

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

■ **功能描述:**

将二线电阻测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200 Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|10 MΩ|100 MΩ|AUTO}，默认值 AUTO，测量结果将直接返回。

■ **返回格式:**

测量结果。

■ **举例:**

MEASure: RES? 200 设置二线电阻模式下量程为 200 Ω，并返回测量结果: +2.43186951E-02

:MEASure:FRESistance

■ **命令格式:**

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

■ **功能描述:**

将四线电阻测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可选{200 Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|AUTO}，默认值 AUTO，测量结果将直接返回。

■ **返回格式:**

无。

■ **举例:**

MEASure: FRES? 2000 设置四线电阻模式下量程为 2 kΩ，并返回测量结果: +2.43186951E-02

:MEASure:TEMPerature

■ **命令格式:**

MEASure:TEMPerature? [{FRTD|RTD|FTHERmistor|THERmistor|TCouple|DEFault} [, {<type>|DEFault}]

■ **功能描述:**

将温度测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置探头及类型。默认探头为 RTD；只有 TCouple 探头可选择类型{ J|K|E|T|N|R|S|B }，其它探头类型默认即可，测量结果将直接返回。

■ **返回格式:**

测量结果。

■ **举例:**

MEASure:TEMP? TCouple J 设置温度模式下使用 TCouple 探头,J 类型,
并返回测量结果: -2.00000000E+02

:MEASure:VOLTage:DC

■ **命令格式:**

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

■ **功能描述:**

将直流电压测量模式下的所有测量参数和触发参数设置为其默认值，或可通过传入参数设置量程，量程可

选{200 mV|2 V|20 V|200 V|1000 V |AUTO}, 默认值 AUTO, 测量结果将直接返回。

■ **返回格式:**

测量结果。

■ **举例:**

MEASure:VOLT:DC? 200V 设置直流电压测量模式下量程为 200V, 并返回测量结果:
+2.43186951E-02

:MEASure:VOLTage:AC

■ **命令格式:**

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

■ **功能描述:**

将交流电压测量模式下的所有测量参数和触发参数设置为其默认值, 或可通过传入参数设置量程, 量程可选{200 mV|2 V|20 V|200 V|750V |AUTO}, 默认值 AUTO, 测量结果将直接返回。

■ **返回格式:**

测量结果。

■ **举例:**

MEASure:VOLT:AC? 20V 设置交流电压测量模式下量程为 20V, 并返回测量结果:
+2.43186951E-02

:MEASure:CAPacitance

■ **命令格式:**

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

■ **功能描述:**

将电容测量模式下的所有测量参数和触发参数设置为其默认值, 或可通过传入参数设置量程, 量程可选{2nF|20nF|200nF|2uF |20uF| 200uF|10000uF |AUTO}, 默认值 AUTO, 测量结果将直接返回。

■ **返回格式:**

测量结果。

■ **举例:**

MEASure:CAP? 20uF 设置电容测量模式下量程为 20V, 并返回测量结果: +2.43186951E-02

SENSe 命令

配置电压、电流、电容、电阻、二极管、频率、周期、温度等挡位的测量参数，切换测量挡位等。

:[SENSe:]FUNction[:ON]

- **命令格式:**

```
[SENSe:]FUNction[:ON] "<function>"  
[SENSe:]FUNction[:ON]?
```

- **功能描述:**

选择测量模式，可选 CONTInuity/CURRent:AC/CURRent:DC/DIODE/FREQuency/FRESistance /PERiod/RESistance/TEMPerature/VOLTage:AC/VOLTage:DC/CAPacitance。

- **返回格式:**

查询返回仪器当前测量模式。

- **举例:**

```
FUNC "VOLT:AC"           选择交流电压测量模式  
FUNC?                   查询返回: VOLT:AC
```

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

- **命令格式:**

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

- **功能描述:**

启用或禁用直流电流模式的归零功能（即相对值功能）。

- **返回格式:**

查询返回归零功能状态，0(关闭)或 1(开启)。

- **举例:**

```
CURR:DC:NULL:STAT ON      开启直流电流模式的测量归零功能  
CURR:DC:NULL:STAT?       查询返回: 1
```

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

- **命令格式:**

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

- **功能描述:**

设置直流电流模式的归零值（即相对值），范围-12A 至 12A；开启归零功能后，最终的测量值=实测值-归零值。

- **返回格式:**

查询返回零值。

■ **举例：**

CURR:DC:NULL: VAL 100mA	设置归零值为 100mA
CURR:DC:NULL: VAL?	查询返回： 100

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

■ **命令格式：**

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

■ **功能描述：**

启用或禁用直流电流模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ **返回格式：**

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ **举例：**

CURRent:DC:NULL:VAL:AUTO ON	启用自动归零值选择
CURRent:DC:NULL:VAL:AUTO?	查询返回： 1

:[SENSe:]CURRent:DC:RANGe

■ **命令格式：**

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

■ **功能描述：**

手动选择测量量程，量程可选{200uA|2mA|20mA|200mA|2A|10A }，默认值 AUTO

■ **返回格式：**

查询返回当前测量量程。

■ **举例：**

CURRent:DC:RANGe 20	设置手动量程 20mA
CURRent:DC:RANGe?	查询返回： 20

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

■ **命令格式：**

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

■ **功能描述：**

禁用或启用自动调整量程。

■ **返回格式：**

查询返回自动调整量程状态，0(关闭)或1(开启)。

■ **举例：**

CURRent:DC:RANGe:AUTO ON

开启自动调整量程

CURRent:DC:RANGe:AUTO?

查询返回：1

:[SENSe:]CURRent:DC:TERMinals

■ **命令格式：**

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

[SENSe:]CURRent:DC:TERMinals?

■ **功能描述：**

将 DC 电流测量配置为测量 200mA 或 10A 端子上的电源。

■ **返回格式：**

查询返回端子类型。

■ **举例：**

CURRent:DC:TERM Big

电流测量配置为 10A 端子上的电源

CURRent:DC:TERM?

查询返回：Big

:[SENSe:]CURRent:DC:NPLC

■ **命令格式：**

[SENSe:]CURRent:DC:NPLC {Slow|Medium|Fast}

[SENSe:]CURRent:DC:NPLC?

■ **功能描述：**

设置直流电流模式的采样速率，可选 Slow/Medium/Fast。

■ **返回格式：**

查询返回采样速率。

■ **举例：**

CURRent:DC:NPLC Medium

设置中速采样

CURRent:DC:NPLC?

查询返回：Medium

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

■ **命令格式：**

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

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

■ **功能描述：**

启用或禁用交流电流模式的归零功能（即相对值功能）。

■ **返回格式：**

查询返回归零功能状态，0(关闭)或 1(开启)。

■ **举例：**

CURR:AC:NULL:STAT ON	开启交流电流模式的测量归零功能
CURR:AC:NULL:STAT?	查询返回： 1

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

■ **命令格式：**

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

■ **功能描述：**

设置交流电流模式的归零值（即相对值），范围-12 至 12A；开启归零功能后，最终的测量值=实测值-归零值。

■ **返回格式：**

查询返回归零值。

■ **举例：**

CURR:AC:NULL: VAL 100mA	设置归零值为 100mA
CURR:AC:NULL: VAL?	查询返回： 100

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

■ **命令格式：**

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

■ **功能描述：**

启用或禁用交流电流模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ **返回格式：**

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ **举例：**

CURRent:AC:NULL:VAL:AUTO ON	启用自动归零值选择
CURRent:AC:NULL:VAL:AUTO?	查询返回： 1

:[SENSe:]CURRent:AC:RANGe

■ **命令格式：**

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

■ **功能描述：**

手动选择测量量程，量程可选{20mA|200mA|2A|10A }，默认值 AUTO

- **返回格式:**

查询返回当前测量量程。

- **举例:**

CURRent:AC:RANGe 20

设置手动量程 20mA

CURRent:AC:RANGe?

查询返回: 20

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

- **命令格式:**

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

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

- **功能描述:**

禁用或启用自动调整量程。

- **返回格式:**

查询返回自动调整量程状态, 0(关闭)或 1(开启)。

- **举例:**

CURRent:AC:RANGe:AUTO ON

开启自动调整量程

CURRent:AC:RANGe:AUTO?

查询返回: 1

:[SENSe:]CURRent:AC:TERMinals

- **命令格式:**

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

[SENSe:]CURRent:AC:TERMinals?

- **功能描述:**

将 AC 电流测量配置为测量 200mA 或 10A 端子上的电源。

- **返回格式:**

查询返回端子类型。

- **举例:**

CURRent:AC:TERM Big

电流测量配置为 10A 端子上的电源

CURRent:AC:TERM?

查询返回: Big

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

- **命令格式:**

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

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

- **功能描述:**

启用或禁用频率模式的归零功能 (即相对值功能)。

■ **返回格式:**

查询返回归零功能状态, 0(关闭)或 1(开启)。

■ **举例:**

FREQ:NULL:STAT ON 开启频率模式的测量归零功能
FREQ:NULL:STAT? 查询返回: 1

:[SENSe:]FREQuency:NULL:VALue

■ **命令格式:**

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

■ **功能描述:**

设置频率模式的归零值 (即相对值), -1.2E6 至 +1.2E6; 开启归零功能后, 最终的测量值=实测值-归零值。

■ **返回格式:**

查询返回归零值。

■ **举例:**

FREQ:NULL: VAL 1kHz 设置归零值为 1kHz
FREQ:NULL: VAL? 查询返回: +1.00000000E-02

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

■ **命令格式:**

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

■ **功能描述:**

启用或禁用周期模式的自动归零值选择, 开启后第一个测量值记录为归零值。

■ **返回格式:**

查询返回自动归零值选择状态, 0(关闭)或 1(开启)。

■ **举例:**

FREQ:NULL:VAL:AUTO ON 启用自动归零值选择
FREQ:NULL:VAL:AUTO? 查询返回: 1

:[SENSe:]FREQuency:VOLTage:RANGe

■ **命令格式:**

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

■ **功能描述:**

手动选择测量量程，量程可选{200 mV|2 V|20 V|200 V|750V}，默认值 AUTO

■ **返回格式：**

查询返回当前测量量程。

■ **举例：**

FREQ:VOLT:RANG 20	设置手动量程 20V
FREQ:VOLT:RANG?	查询返回： 20

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

■ **命令格式：**

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

■ **功能描述：**

禁用或启用自动调整量程。

■ **返回格式：**

查询返回自动调整量程状态，0(关闭)或 1(开启)。

■ **举例：**

FREQ:VOLT:RANGe:AUTO ON	开启自动调整量程
FREQ:VOLT:RANGe:AUTO?	查询返回： 1

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

■ **命令格式：**

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

■ **功能描述：**

启用或禁用周期模式的归零功能（即相对值功能）。

■ **返回格式：**

查询返回归零功能状态，0(关闭)或 1(开启)。

■ **举例：**

PER:NULL:STAT ON	开启周期模式的测量归零功能
PER:NULL:STAT?	查询返回： 1

:[SENSe:]PERiod:NULL:VALue

■ **命令格式：**

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

■ **功能描述：**

设置周期模式的归零值（即相对值），-1.2 至 +1.2 秒；开启归零功能后，最终的测量值=实测值-归零值。

■ **返回格式：**

查询返回归零值。

■ **举例：**

PER:NULL: VAL 0.5s	设置归零值为 0.5s
PER:NULL: VAL?	查询返回： 0.5

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

■ **命令格式：**

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

■ **功能描述：**

启用或禁用周期模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ **返回格式：**

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ **举例：**

PER:NULL:VAL:AUTO ON	启用自动归零值选择
PER:NULL:VAL:AUTO?	查询返回： 1

:[SENSe:]PERiod:VOLTage:RANGe

■ **命令格式：**

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

■ **功能描述：**

手动选择测量量程，量程可选{200 mV|2 V|20 V|200 V|750V }，默认值 AUTO

■ **返回格式：**

查询返回当前测量量程。

■ **举例：**

PER:VOLT:RANGe 20	设置手动量程 20V
PER:VOLT:RANGe?	查询返回： 20

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

■ **命令格式：**

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

■ **功能描述：**

禁用或启用自动调整量程。

■ **返回格式:**

查询返回自动调整量程状态, 0(关闭)或 1(开启)。

■ **举例:**

PER:VOLT:RANGe:AUTO ON	开启自动调整量程
PER:VOLT:RANGe:AUTO?	查询返回: 1

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

■ **命令格式:**

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

■ **功能描述:**

启用或禁用二线电阻模式的归零功能 (即相对值功能)。

■ **返回格式:**

查询返回归零功能状态, 0(关闭)或 1(开启)。

■ **举例:**

RES:NULL:STAT ON	开启周期模式的测量归零功能
RES:NULL:STAT?	查询返回: 1

:[SENSe:]RESistance:NULL:VALue

■ **命令格式:**

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

■ **功能描述:**

设置二线电阻模式的归零值 (即相对值), 范围-120 M Ω 至+120 M Ω ; 开启归零功能后, 最终的测量值 = 实测值 - 归零值。

■ **返回格式:**

查询返回归零值。

■ **举例:**

RES:NULL: VAL 0.1	设置归零值为 100m Ω
RES:NULL: VAL?	查询返回: 0.1

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

■ **命令格式:**

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

■ **功能描述:**

启用或禁用二线电阻模式的自动归零值选择, 开启后第一个测量值记录为归零值。

■ **返回格式:**

查询返回自动归零值选择状态, 0(关闭)或 1(开启)。

■ **举例:**

RES:NULL:VAL:AUTO ON	启用自动归零值选择
RES:NULL:VAL:AUTO?	查询返回: 1

:[SENSe:]RESistance:RANGe

■ **命令格式:**

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

■ **功能描述:**

手动选择测量量程, 量程可选{200 Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ|10 MΩ|100 MΩ}, 默认值 2 kΩ。

■ **返回格式:**

查询返回当前测量量程。

■ **举例:**

RES:RANGe 2000	设置手动量程 2 kΩ
RES:RANGe?	查询返回: 2000

:[SENSe:]RESistance:RANGe:AUTO

■ **命令格式:**

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

■ **功能描述:**

禁用或启用自动调整量程。

■ **返回格式:**

查询返回自动调整量程状态, 0(关闭)或 1(开启)。

■ **举例:**

RES:RANGe:AUTO ON	开启自动调整量程
RES:RANGe:AUTO?	查询返回: 1

:[SENSe:]RESistance:NPLC

■ **命令格式:**

[SENSe:]RESistance: NPLC {Slow|Medium|Fast}

[SENSe:]RESistance: NPLC?

- **功能描述：**
设置二线电阻模式的采样速率，可选 Slow/Medium/Fast。
- **返回格式：**
查询返回采样速率。
- **举例：**
RES:NPLC Medium 设置中速采样
RES:NPLC? 查询返回： Medium

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

- **命令格式：**
[SENSe:]FRESistance:NULL[:STATe] {ON|1|OFF|0}
[SENSe:]FRESistance:NULL[:STATe]?
- **功能描述：**
启用或禁用四线电阻模式的归零功能（即相对值功能）。
- **返回格式：**
查询返回归零功能状态，0(关闭)或 1(开启)。
- **举例：**
FRES:NULL:STAT ON 开启周期模式的测量归零功能
FRES:NULL:STAT? 查询返回： 1

:[SENSe:]FRESistance:NULL:VALue

- **命令格式：**
[SENSe:]FRESistance:NULL:VALue {<value>|MIN|MAX|DEF}
[SENSe:]FRESistance:NULL:VALue?
- **功能描述：**
设置四线电阻模式的归零值（即相对值），范围-120 M Ω 至+120 M Ω ；开启归零功能后，最终的测量值 =实测值-归零值。
- **返回格式：**
查询返回归零值。
- **举例：**
FRES:NULL: VAL 0.1 设置归零值为 100m Ω
FRES:NULL: VAL? 查询返回： 0.1

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

- **命令格式：**

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

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

■ **功能描述：**

启用或禁用四线电阻模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ **返回格式：**

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ **举例：**

FRES:NULL:VAL:AUTO ON 启用自动归零值选择

FRES:NULL:VAL:AUTO? 查询返回： 1

: [SENSe:]FRESistance:RANGe

■ **命令格式：**

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

[SENSe:]FRES:RANGe?

■ **功能描述：**

手动选择测量量程，量程可选{200 Ω|2 kΩ|20 kΩ|200 kΩ|2 MΩ}，默认值 2 kΩ。

■ **返回格式：**

查询返回当前测量量程。

■ **举例：**

FRES:RANGe 2000 设置手动量程 2 kΩ

FRES:RANGe? 查询返回： 2000

: [SENSe:]FRESistance:RANGe:AUTO

■ **命令格式：**

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

[SENSe:]FRESistance:RANGe:AUTO?

■ **功能描述：**

禁用或启用自动调整量程。

■ **返回格式：**

查询返回自动调整量程状态，0(关闭)或 1(开启)。

■ **举例：**

FRES:RANGe:AUTO ON 开启自动调整量程

FRES:RANGe:AUTO? 查询返回： 1

:[SENSe:]FRESistance:NPLC

■ 命令格式:

[SENSe:]FRESistance: NPLC {Slow|Medium|Fast}

[SENSe:]FRESistance: NPLC?

■ 功能描述:

设置四线电阻模式的采样速率，可选 Slow/Medium/Fast。

■ 返回格式:

查询返回采样速率。

■ 举例:

FRES:NPLC Medium

设置中速采样

FRES:NPLC?

查询返回: Medium

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

■ 命令格式:

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

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

■ 功能描述:

启用或禁用直流电压模式的归零功能（即相对值功能）。

■ 返回格式:

查询返回归零功能状态，0(关闭)或 1(开启)。

■ 举例:

VOLT:DC:NULL:STAT ON

开启直流电压模式的测量归零功能

VOLT:DC:NULL:STAT?

查询返回: 1

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

■ 命令格式:

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

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

■ 功能描述:

设置直流电压模式的归零值（即相对值），范围-1200 至+1,200 V；开启归零功能后，最终的测量值=实测值-归零值。

■ 返回格式:

查询返回归零值。

■ 举例:

VOLT:DC:NULL: VAL 100mV

设置归零值为 100mV

VOLT:DC:NULL: VAL?

查询返回: 100

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

■ 命令格式:

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

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

■ 功能描述:

启用或禁用直流电压模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ 返回格式:

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ 举例:

VOLT:DC:NULL:VAL:AUTO ON 启用自动归零值选择

VOLT:DC:NULL:VAL:AUTO? 查询返回: 1

:[SENSe:]VOLTage:DC:RANGe

■ 命令格式:

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

[SENSe:]VOLTage:DC:RANGe?

■ 功能描述:

手动选择测量量程，量程可选{200 mV|2 V|20 V|200 V|1000 V}，默认值 1000V。

■ 返回格式:

查询返回当前测量量程。

■ 举例:

VOLT:DC:RANGe 20V 设置手动量程 20V

VOLT:DC:RANGe? 查询返回: 20

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

■ 命令格式:

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

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

■ 功能描述:

禁用或启用自动调整量程。

■ 返回格式:

查询返回自动调整量程状态，0(关闭)或 1(开启)。

■ 举例:

VOLT:DC:RANGe:AUTO ON 开启自动调整量程

VOLT:DC:RANGe:AUTO? 查询返回: 1

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

■ **命令格式:**

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

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

■ **功能描述:**

针对 DC 电压测量禁用或启用自动输入阻抗模式。

■ **返回格式:**

自动输入阻抗模式状态, 0 (OFF) 或 1 (ON)。

■ **举例:**

VOLT:DC:IMPedance:AUTO ON 启用自动输入阻抗模式

VOLT:DC:IMPedance:AUTO? 查询返回: 1

:[SENSe:]VOLTage:DC:NPLC

■ **命令格式:**

[SENSe:]VOLTage:DC:NPLC {Slow|Medium|Fast}

[SENSe:]VOLTage:DC:NPLC?

■ **功能描述:**

设置直流电压模式的采样速率, 可选 Slow/Medium/Fast。

■ **返回格式:**

查询返回采样速率。

■ **举例:**

VOLT:DC:NPLC Medium 设置中速采样

VOLT:DC:NPLC? 查询返回: Medium

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

■ **命令格式:**

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

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

■ **功能描述:**

启用或禁用交流电压模式的归零功能 (即相对值功能)。

■ **返回格式:**

查询返回归零功能状态, 0(关闭)或 1(开启)。

■ **举例:**

VOLT:AC:NULL:STAT ON 开启交流电压模式的测量归零功能

VOLT:AC:NULL:STAT? 查询返回: 1

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

■ 命令格式:

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

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

■ 功能描述:

设置交流电压模式的归零值（即相对值），范围范围-1200 至+1,200 V；开启归零功能后，最终的测量值=实测值-归零值。

■ 返回格式:

查询返回归零值。

■ 举例:

VOLT:AC:NULL: VAL 100mV 设置归零值为 100mV

VOLT:AC:NULL: VAL? 查询返回: 100

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

■ 命令格式:

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

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

■ 功能描述:

启用或禁用交流电压模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ 返回格式:

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ 举例:

VOLT:AC:NULL:VAL:AUTO ON 启用自动归零值选择

VOLT:AC:NULL:VAL:AUTO? 查询返回: 1

:[SENSe:]VOLTage:AC:RANGe

■ 命令格式:

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

[SENSe:]VOLTage:AC:RANGe?

■ 功能描述:

手动选择测量量程，量程可选{200 mV|2 V|20 V|200 V|750 V}，默认值 750V。

■ 返回格式:

查询返回当前测量量程。

■ 举例:

VOLT:AC:RANGe 20 设置手动量程 20V

VOLT:AC:RANGe? 查询返回: 20

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

■ 命令格式:

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

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

■ 功能描述:

禁用或启用自动调整量程。

■ 返回格式:

查询返回自动调整量程状态，0(关闭)或 1(开启)。

■ 举例:

VOLT:AC:RANGe:AUTO ON

开启自动调整量程

VOLT:AC:RANGe:AUTO?

查询返回: 1

[SENSe:]CAPacitance:NULL[:STATe]

■ 命令格式:

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

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

■ 功能描述:

启用或禁用电容模式的归零功能（即相对值功能）。

■ 返回格式:

查询返回归零功能状态，0(关闭)或 1(开启)。

■ 举例:

VOLT:AC:NULL:STAT ON

开启电容模式的测量归零功能

VOLT:AC:NULL:STAT?

查询返回: 1

:[SENSe:]CAPacitance:NULL:VALue

■ 命令格式:

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

[SENSe:]CAPacitance:NULL:VALue?

■ 功能描述:

设置电容模式的归零值（即相对值），范围范围-12 至+12 mF；开启归零功能后，最终的测量值=实测值-归零值。

■ 返回格式:

查询返回归零值。

■ 举例:

VOLT:AC:NULL: VAL 100nF

设置归零值为 100nF

VOLT:AC:NULL: VAL?

查询返回: 100

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

■ 命令格式:

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

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

■ 功能描述:

启用或禁用电容模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ 返回格式:

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ 举例:

CAPacitance:NULL:VAL:AUTO ON

启用自动归零值选择

CAPacitance:NULL:VAL:AUTO?

查询返回: 1

:[SENSe:]CAPacitance:RANGe

■ 命令格式:

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

[SENSe:]CAPacitance:RANGe?

■ 功能描述:

手动选择测量量程，量程可选{2nF|20nF|200nF|2uF|20uF|200uF|2mF}，默认值 2uF。

■ 返回格式:

查询返回当前测量量程。

■ 举例:

CAPacitance:RANGe 2E-6

设置手动量程 2uF

CAPacitance:RANGe?

查询返回: +2.00000000E-06

:[SENSe:]CAPacitance:RANGe:AUTO

■ 命令格式:

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

[SENSe:]CAPacitance:RANGe:AUTO?

■ 功能描述:

禁用或启用自动调整量程。

■ 返回格式:

查询返回自动调整量程状态，0(关闭)或 1(开启)。

■ 举例:

CAPacitance:RANGe:AUTO ON

开启自动调整量程

CAPacitance:RANGe:AUTO?

查询返回: 1

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

■ **命令格式:**

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

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

■ **功能描述:**

启用或禁用温度测量模式的归零功能（即相对值功能）。

■ **返回格式:**

查询返回归零功能状态，0(关闭)或.0

1(开启)。

■ **举例:**

TEMP:NULL:STAT ON

开启温度测量模式的测量归零功能

TEMP:NULL:STAT?

查询返回: 1

:[SENSe:]TEMPerature:NULL:VALue

■ **命令格式:**

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

[SENSe:]TEMPerature:NULL:VALue?

■ **功能描述:**

设置温度测量模式的归零值（即相对值），范围-1.0E15 至+1.0E15；开启归零功能后，最终的测量值=实测值-归零值。

■ **返回格式:**

查询返回归零值。

■ **举例:**

TEMP:NULL:VAL 25

设置归零值为 25°

TEMP:NULL:VAL?

查询返回: +2.50000000E+01

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

■ **命令格式:**

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

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

■ **功能描述:**

启用或禁用温度测量模式的自动归零值选择，开启后第一个测量值记录为归零值。

■ **返回格式:**

查询返回自动归零值选择状态，0(关闭)或 1(开启)。

■ **举例:**

TEMP:NULL:VAL:AUTO ON

启用自动归零值选择

TEMP:NULL:VAL:AUTO?

查询返回： 1

:[SENSe:]TEMPerature:TRANsducer:TYPE

■ 命令格式：

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

[SENSe:]TEMPerature:TRANsducer:TYPE?

■ 功能描述：

选择温度测量使用的探头，支持的探头包括二线 RTD、四线 RTD、二线热敏电阻、四线热敏电阻、(J/K/E/T/N/R/S/B) 类热电偶。

■ 返回格式：

查询返回温度测量使用的探头类型。

■ 举例：

TEMP:TRAN:TYPE: RTD

选择温度测量使用二线 RTD 探头

TEMP:TRAN:TYPE?

查询返回： RTD

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

■ 命令格式：

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

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

■ 功能描述：

设置热电偶测量温度时的固定参考结温度（单位为摄氏度），范围-20 °C 至 +80 °C。

■ 返回格式：

查询返回固定参考结温度值。

■ 举例：

TEMP:TRAN:TC:RJUN 20.0

热电偶测量温度时的固定参考结温度为 20.0° C

TEMP:TRAN:TC:TYPE?

查询返回： +2.00000000E+01

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

■ 命令格式：

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

{<temperature>|MIN|MAX|DEF}

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

■ 功能描述：

设置热电偶测量温度时的补偿温度（单位为摄氏度），以便修正 DMM 前连接的内部温度测量值和测量端子实际温度之间的误差，范围-20 °C 至 +20 °C。

■ 返回格式：

查询返回补偿温度值。

■ **举例：**

TEMP:TRAN:TC:RJUN:OFFS:ADJ -5

设置补偿温度-5° C

TEMP:TRAN:TC:RJUN:OFFS:ADJ?

查询返回：-5.00000000E+00

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

■ **命令格式：**

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

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

■ **功能描述：**

设置参考结温度的来源，可以使用内部测量值或者已知的固定结温度；进行热电偶测量时必须指定参考结温度。

■ **返回格式：**

查询返回参考结温度的来源，FIXed（固定）或 INTernal（内部）。

■ **举例：**

TEMP:TRAN:TC:RJUN:TYPE FIX

参考结温度来自固定结温度

TEMP:TRAN:TC:RJUN 20.0

设置固定参考结温度为 20.0° C

TEMP:TRAN:TC:RJUN:TYPE?

查询返回：FIX

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

■ **命令格式：**

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

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

■ **功能描述：**

设置热电偶测量温度时使用的电偶类型，可选 J|K|E|T|N|R|S|B。

■ **返回格式：**

查询返回热电偶测量温度时使用的电偶类型。

■ **举例：**

TEMP:TRAN:TC:YPE K

使用 K 型热电偶测量温度

TEMP:TRAN:TC:TYPE?

查询返回：K

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

■ **命令格式：**

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

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

■ **功能描述：**

设置 2 线 RTD 测量时的 RTD 标称电阻, 范围 50Ω 至 2000 Ω; 需要根据实际测量来配置标称电阻(R0)。

■ **返回格式:**

查询返回 RTD 标称电阻。

■ **举例:**

TEMP:TRAN:RTD:RES 20	设置 RTD 标称电阻 R0 = 20Ω
TEMP:TRAN:RTD:RES?	查询返回: +2.00000000E+01

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

■ **命令格式:**

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

■ **功能描述:**

设置 4 线 RTD 测量时的 RTD 标称电阻, 范围 50Ω 至 2000 Ω; 需要根据实际测量来配置标称电阻(R0)。

■ **返回格式:**

查询返回 RTD 标称电阻。

■ **举例:**

TEMP:TRAN:RTD:FRES 20	设置 RTD 标称电阻 R0 = 20Ω
TEMP:TRAN:FRTD:RES?	查询返回: +2.00000000E+01

:[SENSe:]CONTinuity:THReshold:VALue

■ **命令格式:**

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

■ **功能描述:**

设置联通性测试时的门限电阻, 范围 0~2000 Ω。

■ **返回格式:**

查询返回门限电阻值。

■ **举例:**

CONT:THR:VAL 2000	设置门限电阻值 2000Ω
CONT:THR:VAL?	查询返回: +2.00000000E+03

TRIGger 命令

配置测量、采样相关的参数。

:TRIGger:COUNT

- **命令格式:**

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

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

- **功能描述:**

设置仪器在一次测量中所接受的触发数。

- **返回格式:**

查询返回触发数。

- **举例:**

CONF:VOLT:DC	配置直流电压测量
TRIG:COUN 10	设置一次测量接受 10 测触发
TRIG:COUN?	查询返回: 10
SAMP:COUN 5	设置触发测试后采集 5 个测量值
READ?	读取返回 50 个测量值

:TRIGger:DELAy

- **命令格式:**

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

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

- **功能描述:**

设置触发信号和第一次测量之间的延迟, 范围 1 至 3600 秒。

- **返回格式:**

查询返回测量延迟。

- **举例:**

TRIG:DEL 2	设置测量延迟 2 秒
TRIG:DEL?	查询返回: +2.00000000E+00

:TRIGger:DELAy:AUTO

- **命令格式:**

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

TRIGger: DELAy:AUTO?

- **功能描述:**

禁用或启用自动触发延迟功能。

- **返回格式:**

查询返回 0(关闭)或 1(开启)。

■ **举例：**

TRIG:DEL:AUTO ON	启用自动触发延迟功能
TRIG:DEL:AUTO?	查询返回： 1

:TRIGger:SLOPe

■ **命令格式：**

TRIGger:SLOPe {POSitive|NEGative}
TRIGger:SLOPe?

■ **功能描述：**

当使用后面板 Ext Trig BNC 连接器输入的信号作为触发源时，使用该命令选择触发信号极性，上升沿 (POS) 或下降沿(NEG)触发。

■ **返回格式：**

查询返回触发信号极性，POS (上升沿)或 NEG (下降沿)。

■ **举例：**

TRIG:SLOP POS	选择上升沿触发
TRIG:SLOP?	查询返回： POS

:TRIGger:SOURce

■ **命令格式：**

TRIGger:SOURce {IMMediate|EXTernal|BUS}
TRIGger:SOURce?

■ **功能描述：**

为测量选择触发源，说明如下：

IMMediate	触发信号一直存在。将仪器置于“等待触发”状态时，将立刻发出触发信号。
BUS	一旦数字万用表处于“等待触发”状态，*TRG 就会通过远程接口触发仪器。
EXTernal	仪器后面板 Ext Trig 输入的作为触发信号，每触发一次将采集指定的测量次数 (SAMPle:COUNt)。如果仪器在准备好以前接收了一个外部触发，则仪器将缓冲一个触发。

■ **返回格式：**

查询返回当前选定的触发源。

■ **举例：**

TRIG:SOURce EXT	选择外部触发源
TRIG:SOURce?	查询返回： EXT

SYSTem 命令

配置系统级参数参数、获取如错误等系统级信息。

:SYSTem:ERRor

- **命令格式:**

SYSTem:ERRor[:NEXT]?

- **功能描述:**

读取并清除错误队列中的一个错误。

- **返回格式:**

返回错误码级错误信息: <错误代码>,<错误字符串>。

- **举例:**

SYST:MERR?	发送错误命令
SYST:ERR?	返回: -113,"Undefined header"

:SYSTem:BEEPer:STATe

- **命令格式:**

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

SYSTem:BEEPer:STATe?

- **功能描述:**

在连续性、二级管或者探头保持测量过程中或者当前面板或远程接口产生错误时, 禁用或启用蜂鸣器发声。

- **返回格式:**

返回蜂鸣状态, 0(关闭)或 1(开启)。

- **举例:**

SYST:BEEP:STATe ON	开启蜂鸣
SYST:BEEP:STATe?	返回: 1

编程说明

描述在编程操作过程中可能出现的一些问题及解决方法。当您遇到如下这些问题时，请按照相应的说明进行处理。

编程准备

编程准备工作仅适用于在 Windows 操作系统下使用 Visual Studio 和 LabVIEW 开发工具进行编程。

首先确认您的电脑上是否已经安装 NI 的 VISA 库（可到 <https://www.ni.com/en-ca/support/downloads/drivers/download.ni-visa.html> 下载），本文中默认安装路径为 C:\Program Files\IVI Foundation\VISA。

通过仪器设备的 USB 或 LAN 接口与 PC 建立通信，请使用 USB 数据线将仪器设备后面板的 USB DEVICE 接口与 PC 的 USB 接口相连，或者使用 LAN 数据线将仪器设备后面板的 LAN 口与 PC 的 LAN 接口相连。

VISA 编程示例

本节给出了一些编程示例。通过这些例子，你可以了解如何使用 VISA，并结合编程手册的命令实现对仪器设备的控制。通过下面的例子，你可以开发更多应用。

VC++ 示例

- 环境：Window 系统, Visual Studio。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"*IDN?"命令来查询设备信息。
- 步骤：
 1. 打开 Visual Studio 软件，新建一个 VC++ win32 console project。
 2. 设置调用 NI-VISA 库的项目环境，分别为静态库和动态库。
 - a) 静态库：

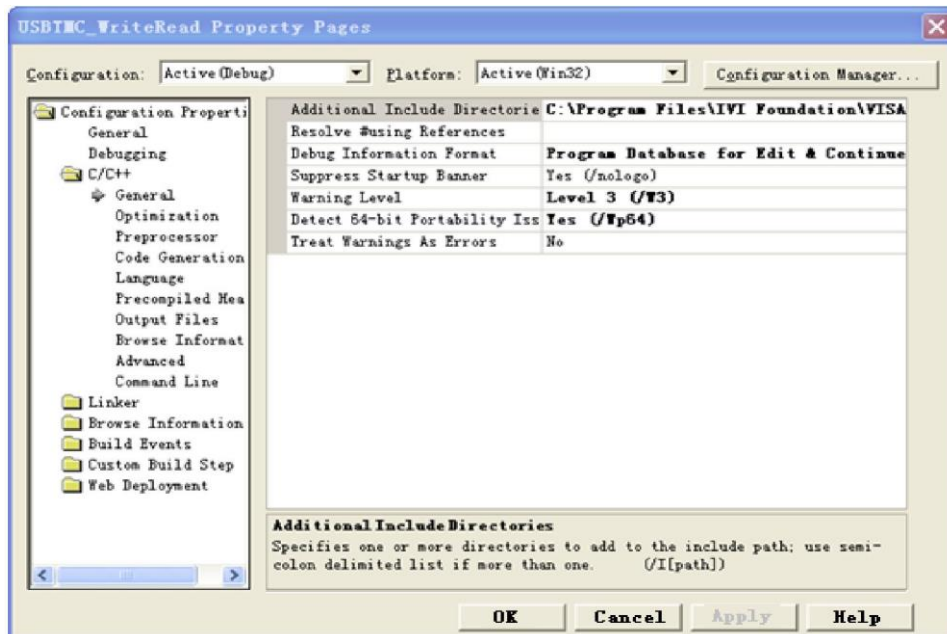
在 NI-VISA 安装路径找:visa.h、visatype.h、visa32.lib 文件，将它们复制到 VC++项目的根路径下并添加到项目中。在 projectname.cpp 文件上添加下列两行代码：

```
#include "visa.h"  
#pragma comment(lib,"visa32.lib")
```

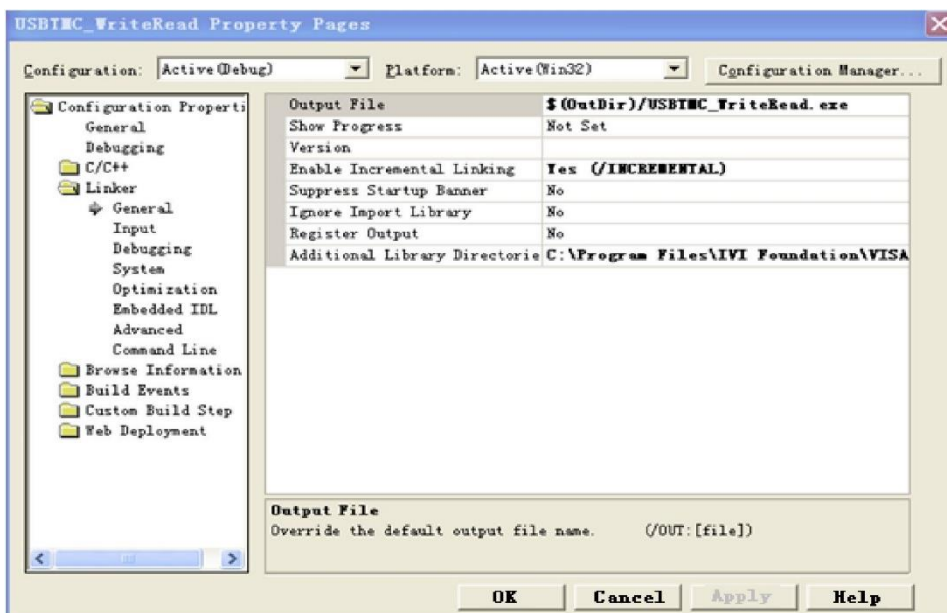
- b) 动态库：

点击"project>>properties"，在属性对话框左侧选择"c/c++---General"中，将 "Additional

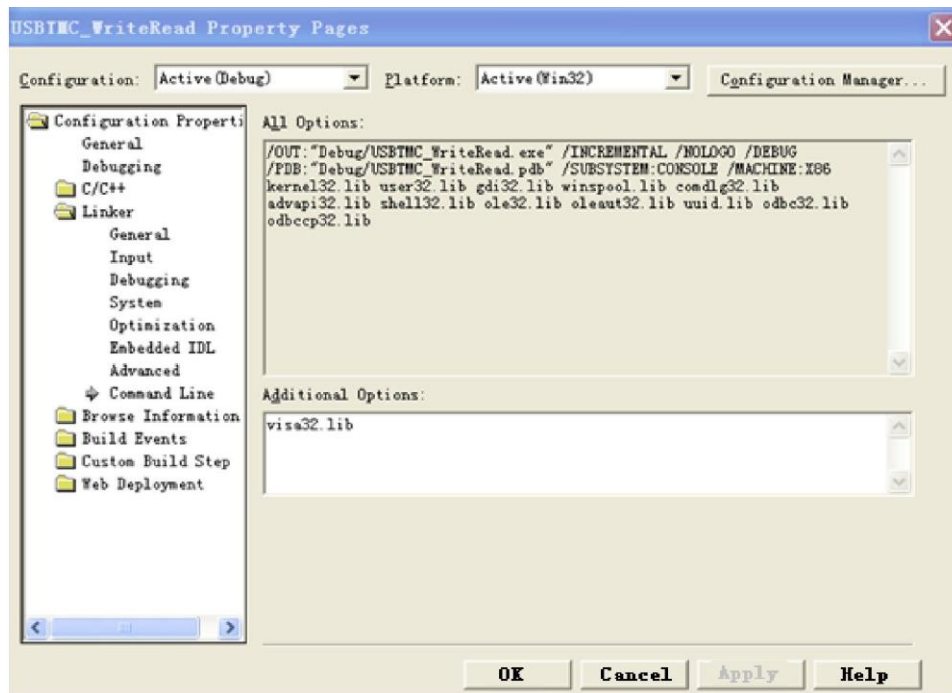
Include Directories"项的值设置为 NI-VISA 的安装路径, (例如: C:\ProgramFiles\IVI Foundation\VISA\WinNT\include),如下图所示:



在属性对话框左侧选择"Linker-General",并将"Additional Library Directories"项的值设置为 NI-VISA 的安装路径, (例如: C:\Program Files\IVI Foundation\VISA\WinNT\include),如下图所示:



在属性对话框左侧选择"Linker-Command Line",将"Additional"项的值设置为 visa32.lib,如下图所示:



在 projectname.cpp 文件上添加 visa.h 文件:

```
#include <visa.h>
```

1. 源码:

a) USBTMC 示例

```
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 示例

```

int tcp_ip_test(char *pIP)
{
    char outputBuffer[VI_FIND_BUFLLEN];
    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, "%dn?\n");
status = viScanf(instr, "%t", outputBuffer);
if (status < VI_SUCCESS)
{
    printf("viRead failed with error code: %x \n", status);
    viClose(defaultRM);
}
else
{
    printf("\nMessage 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#示例

- 环境：Window 系统, Visual Studio。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"*IDN?"命令来查询设备信息。
- 步骤：
 1. 打开 Visual Studio 软件，新建一个 C# console project。
 2. 添加 VISA 的 C#引用 Ivi.Visa.dll 和 NationalInstruments.Visa.dll。
 3. 源码：
 - a) USBTMC 示例

```
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 示例

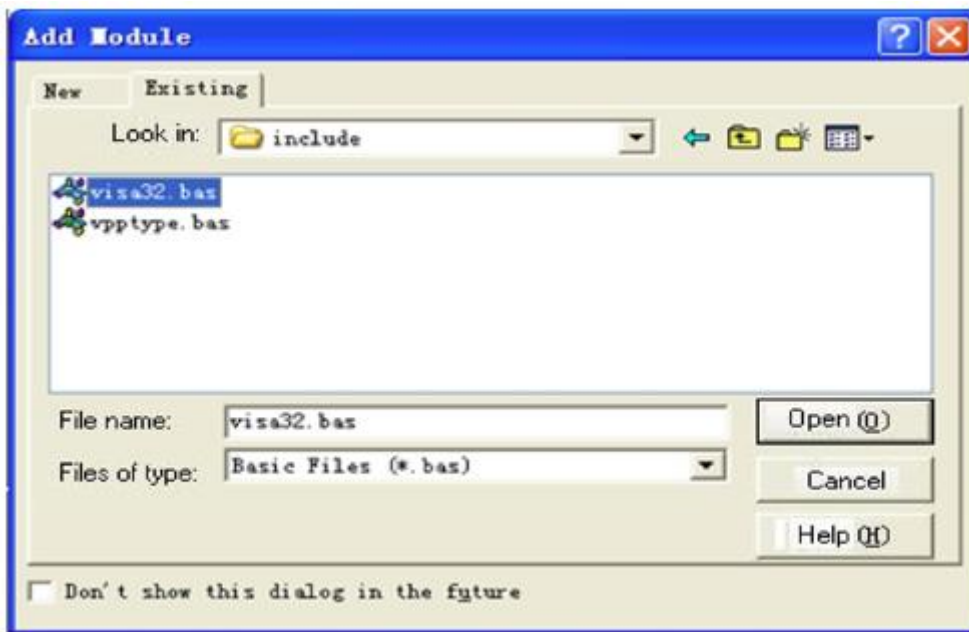
```
class Program
```

```
{
    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 示例

- 环境：Window 系统, Microsoft Visual Basic 6.0。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"*IDN?"命令来查询设备信息。
- 步骤：
 1. 打开 Visual Basic 软件，并新建一个标准的应用程序项目。
 2. 设置调用 NI-VISA 库项目环境：点击 Existing tab of Project>>Add Existing Item，在 NI-VISA 安装路径下的"include"文件夹中查找 visa32.bas 文件并添加该文件。如下图所示：



3. 源码：

a) USBTMC 示例

```
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 示例

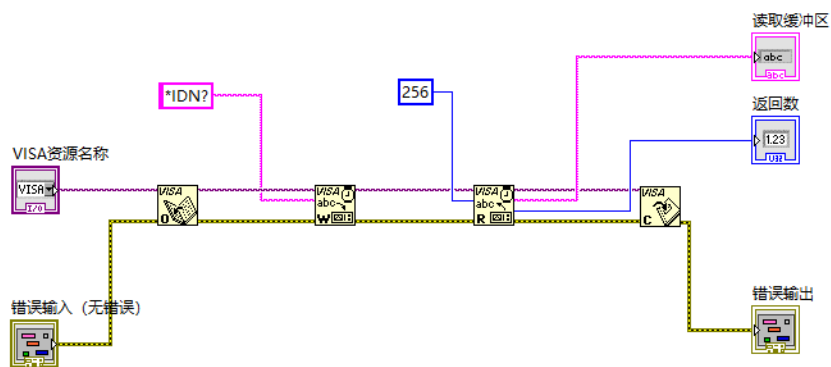
```
PrivateFunction tcp_ip_test(ByVal ip AsString) AsLong
Dim outputBuffer AsString * VI_FIND_BUFLen
Dim defaultRM AsLong
Dim instrsesn AsLong
Dim status AsLong
Dim count AsLong

' 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

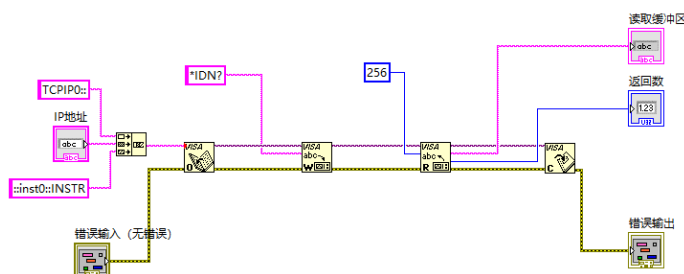
' Now we will open a session via TCP/IP device
status = viOpen(defaultRM, "TCPIP0::" + ip + "::inst0::INSTR", VI_LOAD_CONFIG, VI_NULL, instrsesn)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "An error occurred opening the session"
    viClose(defaultRM)
    tcp_ip_test = status
ExitFunction
EndIf
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 示例

- 环境：Window 系统, LabVIEW。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"*IDN?"命令来查询设备信息。
- 步骤：
 1. 打开 LabVIEW 软件，并创建一个 VI 文件。
 2. 添加控件，右击前面板界面，从控制列中选择并添加 VISA 资源名、错误输入、错误输出以及部分的指示符。
 3. 打开框图界面，右击 VISA 资源名称，并在弹出菜单的 VISA 面板中选择和添加下列功能：VISA Write、VISA Read、VISA Open 和 VISA Close。
 4. VI 打开了一个 USBTMC 设备的 VISA 会话，并向设备写*IDN?命令并回读的响应值。当所有通信完成时，VI 将关闭 VISA 会话，如下图所示：



5. 通过 TCP/IP 与设备通信类似于 USBTMC,但是你需要将 VISA 写函数和 VISA 读函数设置为同步 I/O,LabVIEW 默认设置为异步 IO。右键单击节点，然后从快捷菜单中选择,"Synchronous I/O Mode>>Synchronous"以实现同步写入或读取数据，如下图所示：



MATLAB 示例

- 环境：Window 系统, MATLAB。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"*IDN?"命令来查询设备信息。
- 步骤：
 1. 打开 MATLAB 软件, 点击在 Matlab 界面的 File>>New>>Script 创建一个空的 M 文件。
 2. 源码：
 - a) USBTMC 示例

```
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 示例

```
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 示例

- 环境：Window 系统, Python3.8, PyVISA 1.11.0。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"*IDN?"命令来查询设备信息。
- 步骤：
 1. 首先安装 python，然后打开 Python 脚本编译软件，创建一个空的 test.py 文件。
 2. 使用 pip install PyVISA 指令安装 PyVISA，如无法安装，请参考此链接使用说明 (<https://pyvisa.readthedocs.io/en/latest/>)

3. 源码：

a) USBTMC 示例

```
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/IP 示例

```
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?'))
```

编程应用实例

总线触发测量

一个完整的总线触发直流电压测量的流程，包含的命令：

```
CONF:VOLT:DC 200V      #切换到直流电压测量模式，手动量程 200V。
TRIG:SOUR BUS          #设置触发源为总线触发
TRIG:COUN 5            #设置触发次数为 5 次
SAMP:COUN 10           #设置采样次数为 10 次
INIT                   #将触发状态设置为等待触发
*TRG                   #发送触发信号
FETCh?                 #读取测量数量，返回 5*10
```

测量统计结果获取

```
CONF:FREQ              #切换到频率模式，使用默认 AUTO 量程
SAMP:COUN 100          #采样 100 个测量值
CALC:AVER:STAT ON     #开启统计功能
INIT                   #即时采样启动
CALC:AVER:ALL?        #获取采样结果
```

SENSe 命令配置参数进行测量

```
FUNC "CURRent:DC"     #选择直流电流模式
CURRent:DC:RANGe 20   #设置手动量程 20mA
CURR:AC:NULL:VALue 1mA #归零值（即相对值）1mA
CURR:AC:NULL:STAT ON #开启归零值处理
SAMP:COUN 2           #采样 2 个测量值
READ?                 #读取测量值
FUNC "TEMP"           #选择温度模式
TEMP:TRAN:TYPE RTD    #选择二线 RTD 探头
TEMP:TRAN:FRTD:RES 100 #配置标称电阻(R0)值 100Ω
SAMP:COUN 2           #采样 2 个测量值
READ?                 #读取测量值
```


附录 1：IEEE 488.2 二进制数据格式

DATA 是数据流,其他为 ASCII 字符,如下图所示: <#12345678 + DATA + \n>

开始符 (1Byte)	长度位宽 (1Byte)	数据总长度 (位宽 Byte)	DATA (n Byte)	结束符 (1Byte)
#	x	x x x x x x x x	\n