

# MSO/UPO2000系列可编程数字示波器

---

## 编程手册

REV 00  
2023. 05

**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

20230530

## 软件版本

01.05.0075

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

- 本公司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。

# SCPI 指令详解

## IEEE488.2 通用命令

### \* IDN?

- **命令格式:**

\*IDN?

- **功能描述:**

用于查询制造商名称、示波器型号、产品序列号和软件版本号。

- **返回格式:**

制造商名称, 示波器型号, 产品序列号, 由点号分隔的软件版本号。

- **举例:**

UNI-T Technologies, MSO2000CS, MSO1000, 00.00.01

### \* RST

- **命令格式:**

\*RST

- **功能描述:**

用于恢复出厂设置并清空所有的错误信息及发送接收队列缓冲。

### \* OPC

- **命令格式:**

\*OPC

\*OPC?

- **功能描述:**

用于强制将当前指令已执行完成标志位置 1。

- **返回格式:**

查询返回当前发送的指令执行操作是否完成, 1 表示完成, 0 表示未完成。

- **举例:**

\*OPC

将指令已完成标志位置 1

\*OPC?

查询返回 1, 表示当前指令已执行完成, 反之未完成

## SYSTEM 命令

用于对示波器进行最基本的操作，主要包括运行控制、全键盘锁定、错误队列和系统设置数据的操作。

### :RUN

- **命令格式:**

:RUN

- **功能描述:**

用于开始示波器波形采样工作，如需停止工作，需要执行:STOP 命令。

### :STOP

- **命令格式:**

:STOP

- **功能描述:**

用于停止示波器波形采样工作，如需恢复工作，需要执行:RUN 命令。

### :AUTO

- **命令格式:**

:AUTO

- **功能描述:**

用于自动设定仪器的控制值，通过自动设置使输入的波形达到最佳显示效果。

### :SYSTEM:LOCK

- **命令格式:**

:SYSTEM:LOCK {{1 | ON} | {0 | OFF}}

:SYSTEM:LOCK?

- **功能描述:**

用于锁定或者解锁全键盘按键。

- **返回格式:**

查询返回全键盘锁定状态，0 表示未锁定，1 表示锁定。

- **举例:**

:SYSTEM:LOCK ON/:SYST:LOCK 1                      全键盘锁定

:SYSTEM:LOCK OFF/:SYST:LOCK 0                     全键盘解锁

:SYSTEM:LOCK?                                         查询返回 1，表示锁定

## :SYSTem:ERRor

### ■ 命令格式:

:SYSTem:ERRor

:SYSTem:ERRor?

### ■ 功能描述:

用于清空错误消息队列。

### ■ 返回格式:

查询返回最后一次消息错误，查询以“<消息编号>,<消息内容>”格式返回错误消息，其中<消息编号>是一个整数，<消息内容>是一个带双引号的 ASCII 字符串。

如-113,"Undefined header; command cannot be found"。

### ■ 举例:

:SYSTem:ERR

清空错误队列

:SYSTem:ERR?

查询返回:

-113,"Undefined header; command cannot be found"

表示未定义指令头

## :SYSTem:SETup

### ■ 命令格式:

:SYSTem:SETup <setup\_data>

:SYSTem:SETup?

### ■ 功能描述:

用于配置系统设置数据，其中<setup\_data>为符合[附录 2: IEEE 488.2 二进制数据格式](#)。

### ■ 返回格式:

查询返回系统设置数据。

## :SYSTem:LANGuage

### ■ 命令格式:

:SYSTem:LANGuage { ENGLISH | SIMPLifiedchinese | TRADitionalchinese }

:SYSTem:LANGuage?

### ■ 功能描述:

用于设置系统语言。

### ■ 返回格式:

查询返回{ ENGLISH | SIMPLifiedchinese | TRADitionalchinese }。

### ■ 举例:

:SYSTem:LANGuage ENGL

设置系统语言为英文

:SYSTem:LANGuage?

查询返回 ENGLISH

## **:SYSTem:RTC**

### ■ **命令格式:**

:SYSTem:RTC <year>,<month>,<day>,<hour>,<minute>,<second>

:SYSTem:RTC?

### ■ **功能描述:**

用于设置系统时间。

### ■ **返回格式:**

查询返回 年, 月, 日, 时, 分, 秒。

### ■ **举例:**

:SYSTem:RTC 2017,7,7,20,8,8          设置系统时间 2017 年 7 月 7 日 20 点 8 分 8 秒

:SYSTem:RTC?                          查询返回 2017,7,7,20,8,8

## **:SYSTem:CAL**

### ■ **命令格式:**

:SYSTem:CAL

### ■ **功能描述:**

用于设置系统自校准, 自校正期间, 不能正常通信。

## **:SYSTem:CLEAr**

### ■ **命令格式:**

:SYSTem:CLEAr

### ■ **功能描述:**

用于清除系统所有的存储波形和设置数据。

## **:SYSTem:CYMOmeter**

### ■ **命令格式:**

:SYSTem:CYMOmeter {1 | ON} | {0 | OFF}

:SYSTem:CYMOmeter?

### ■ **功能描述:**

用于设置打开或关闭频率计。

### ■ **返回格式:**

查询返回频率计状态, 1 表示打开, 0 表示关闭。

### ■ **举例:**

:SYSTem:CYMOmeter ON                  打开频率计

:SYSTem:CYMOmeter?                  查询返回 1



## **:SYSTem:CYMOmeter:FREQuency?**

- **命令格式:**  
:SYSTem:CYMOmeter:FREQuency?
- **功能描述:**  
用于获取频率计测量的频率值。
- **返回格式:**  
查询返回频率计测量的频率值，无效值时返回\*。
- **举例:**  
:SYSTem:CYMOmeter:FREQuency? 查询返回 1.2000E+3

## **:SYSTem:SQUare:SElect**

- **命令格式:**  
:SYSTem:SQUare:SElect { 10Hz | 100Hz | 1KHz | 10KHz }  
:SYSTem:SQUare:SElect?
- **功能描述:**  
用于设置选择方波输出。
- **返回格式:**  
查询返回{ 10Hz | 100Hz | 1KHz | 10KHz }。
- **举例:**  
:SYSTem:SQUare:SElect 10Hz      选择 10Hz 的方波输出  
:SYSTem:SQUare:SElect?          查询返回 10Hz

## **:SYSTem:OUTPut:SElect**

- **命令格式:**  
:SYSTem:OUTPut:SElect { TRIGger | PASS\_FAIL }  
:SYSTem:OUTPut:SElect?
- **功能描述:**  
用于设置输出选择，TRIGger（触发）、PASS\_FAIL（通过&失败）。
- **返回格式:**  
查询返回{ TRIGger | PASS\_FAIL }。
- **举例:**  
:SYSTem:OUTPut:SElect TRIG      输出选择触发  
:SYSTem:OUTPut:SElect?          查询返回 TRIG

## **:SYSTem:MNuDisplay**

### ■ 命令格式:

:SYSTem:MNuDisplay { 5S | 10S | 20S | INFinite}

:SYSTem:MNuDisplay?

### ■ 功能描述:

用于设置菜单的显时间, INFinite 表示菜单一直显示。

### ■ 返回格式:

查询返回{ 5S | 10S | 20S | INFinite }。

### ■ 举例:

:SYSTem:MNuDisplay 5S                    设置菜单显示时基为 5S 之后, 自动收回。

:SYSTem:MNuDisplay?                    查询返回 5S

## **:SYSTem:BRIGhtness**

### ■ 命令格式:

:SYSTem:BRIGhtness <count>

:SYSTem:BRIGhtness?

### ■ 功能描述:

用于设置屏幕亮度, <count>取值为 1~100, 数字越大屏幕越亮。

### ■ 返回格式:

查询返回当前屏幕亮度。

### ■ 举例:

:SYSTem:BRIGhtness 50                    设置屏幕亮度 50

:SYSTem:BRIGhtness?                    查询返回 50

## **:SYSTem:VERsion?**

### ■ 命令格式:

:SYSTem:VERsion?

### ■ 返回格式:

查询返回版本信息, 128 字节的字符串信息。

HW 为硬件版本号, SW 为软件版本号, PD 为生成日期, ICV 为协议版本号。

### ■ 举例:

:SYST:VERS?                                查询返回 HW:1.0;SW:1.0;PD:2014-11-20;ICV:1.4.0

## **:SYSTem:COMMunicate:LAN:APPLy**

### ■ 命令格式:

:SYSTem:COMMunicate:LAN:APPLy

■ **功能描述:**

用于立即生效当前设置的网络参数。

**:SYSTem:COMMunicate:LAN:GATEway**

■ **命令格式:**

:SYSTem:COMMunicate:LAN:GATEway <gateway>

:SYSTem:COMMunicate:LAN:GATEway?

■ **功能描述:**

用于设置默认网关。<gateway>属于 ASCII 字符串参数，格式为 xxx.xxx.xxx.xxx。

■ **返回格式:**

查询返回默认网关。

■ **举例:**

:SYST:COMM:LAN:GATE "192.168.1.1"      设置默认网关 192.168.1.1

:SYST:COMM:LAN:GATE?                      查询返回 192.168.1.1

**:SYSTem:COMMunicate:LAN:SMASK**

■ **命令格式:**

:SYSTem:COMMunicate:LAN:SMASK <submask>

:SYSTem:COMMunicate:LAN:SMASK?

■ **功能描述:**

用于设置子网掩码。<submask>属于 ASCII 字符串参数，格式为 xxx.xxx.xxx.xxx。

■ **返回格式:**

查询返回子网掩码。

■ **举例:**

:SYST:COMM:LAN:SMASK "255.255.255.0"      设置子网掩码 255.255.255.0

:SYST:COMM:LAN:SMASK?                      查询返回 255.255.255.0

**:SYSTem:COMMunicate:LAN:IPADdress**

■ **命令格式:**

:SYSTem:COMMunicate:LAN:IPADdress <ip>

:SYSTem:COMMunicate:LAN:IPADdress?

■ **功能描述:**

用于设置 IP 地址。<ip>属于 ASCII 字符串参数，格式为 xxx.xxx.xxx.xxx。

■ **返回格式:**

查询返回 IP 地址。

■ 举例：

:SYST:COMM:LAN:IPAD "192.168.1.10" 设置 IP 地址 192.168.1.10

:SYST:COMM:LAN:IPAD? 查询返回 192.168.1.10

**:SYSTem:COMMunicate:LAN:DHCP**

■ 命令格式：

:SYSTem:COMMunicate:LAN:DHCP {{1 | ON} | {0 | OFF}}

:SYSTem:COMMunicate:LAN:DHCP?

■ 功能描述：

用于切换（自动 IP）和（手动 IP）配置模式。

■ 返回格式：

查询返回动态配置模式，0 表示（手动 IP），1 表示（自动 IP）。

■ 举例：

:SYST:COMM:LAN:DHCP ON 打开 IP 动态配置

:SYST:COMM:LAN:DHCP? 查询返回 1

**:SYSTem:COMMunicate:LAN:MAC?**

■ 命令格式：

:SYSTem:COMMunicate:LAN:MAC?

■ 返回格式：

查询返回 MAC 物理地址。

■ 举例：

:SYST:COMM:LAN:MAC? 查询返回 00-2A-A0-AA-E0-56

**:SYSTem:AUTO:CHANnel**

■ 命令格式：

:SYSTem:AUTO:CHANnel {AUTO|KEEP}

:SYSTem:AUTO:CHANnel?

■ 功能描述：

用于自动设置时通道属性是否保持当前状态。

AUTO 表示通道按预置参数自动设置；KEEP 表示通道在保持当前状态前提进行自动设置。

■ 返回格式：

查询{AUTO|KEEP}。

■ 举例：

:SYSTem:AUTO:CHANnel KEEP 通道保持当前状态自动设置。

:SYSTem:AUTO:CHANnel? 查询返回 KEEP。

## :SYSTem:AUTO:ACQuire

### ■ 命令格式:

:SYSTem:AUTO:ACQuire {AUTO|KEEP}

:SYSTem:AUTO:ACQuire?

### ■ 功能描述:

用于自动设置时采样属性是否保持当前状态。

AUTO 表示采样按预置参数自动设置；KEEP 表示采样在保持当前状态前提进行自动设置。

### ■ 返回格式:

查询{AUTO|KEEP}。

### ■ 举例:

:SYSTem:AUTO:ACQuire KEEP

采样保持当前状态自动设置。

:SYSTem:AUTO:ACQuire?

查询返回 KEEP。

## :SYSTem:AUTO:TRIGger

### ■ 命令格式:

:SYSTem:AUTO:TRIGger {AUTO|KEEP}

:SYSTem:AUTO:TRIGger?

### ■ 功能描述:

用于自动设置时触发属性是否保持当前状态。

AUTO 表示触发按预置参数自动设置；KEEP 表示触发在保持当前状态前提进行自动设置。

### ■ 返回格式:

查询{AUTO|KEEP}。

### ■ 举例:

:SYSTem:AUTO:TRIGger KEEP

触发保持当前状态自动设置。

:SYSTem:AUTO:TRIGger?

查询返回 KEEP。

## :SYSTem:AUTO:SIGNal

### ■ 命令格式:

:SYSTem:AUTO:SIGNal {AUTO|KEEP}

:SYSTem:AUTO:SIGNal?

### ■ 功能描述:

用于自动设置时输入信号的通道(活动通道)是否保持当前状态。

AUTO 表示活动通道按预置参数自动设置；KEEP 表示活动通道在保持当前状态前提进行自动设置。

### ■ 返回格式:

查询{AUTO|KEEP}。

### ■ 举例:

:SYSTem:AUTO:SIGNal KEEP

活动通道保持当前状态自动设置。

:SYSTem:AUTO:SIGNal?

查询返回 KEEP。

## KEY 命令

用于控制示波器操作面板上的按键和旋钮。

**:KEY:<key>**

### ■ 命令格式:

:KEY:<key>

:KEY:<key>:LOCK { {1 | ON} | {0 | OFF} }

:KEY:<key>:LOCK?

:KEY:<key>:LED?

### ■ 功能描述:

用于设置按键功能及该按键的锁定/解锁。按键<key>的定义和描述，详见附录 1：[按键列表](#)。

### ■ 返回格式:

查询返回按键锁定状态或者具有 LED 按键灯状态。

锁定状态：0 表示未锁定，1 表示锁定；

LED 灯状态：0 表示不亮，1 表示亮，其中 RUN/STOP 按键 0 表示红色，1 表示绿色。

### ■ 举例:

:KEY:AUTO	自动设置示波器的各项控制值
:KEY:AUTO:LOCK ON/OFF	锁定/解锁按键
:KEY:AUTO:LOCK?	查询返回该按键锁定状态，1 表示锁定
:KEY:AUTO:LED?	查询返回 LED 灯状态，0 表示不亮

## CHANnel 命令

用于对每个通道单独进行设置，其中<n>取值为 1/2/3/4 分别表示{CH1/CH2/ CH3/ CH4}。

**:CHANnel<n>:BWLimit**

### ■ 命令格式:

:CHANnel<n>:BWLimit { {1|ON} | {0|OFF} }

:CHANnel<n>:BWLimit?

### ■ 功能描述:

用于设置带宽限制功能为 ON（打开限制带宽至 20MHz，以减少显示的噪音）或 OFF（关闭带宽限制实现满带宽显示）。

<n>: {1|2|3|4}，分别表示{CH1|CH2| CH3|CH4}。

### ■ 返回格式:

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例：**

:CHAN1:BWL ON

打开通道 1 的带宽限制。

:CHAN1:BWL?

查询返回 1，表示已经打开通道 1 的带宽限制。

### **:CHANnel<n>:COUPling**

■ **命令格式：**

:CHANnel<n>:COUPling {DC|AC|GND}

:CHANnel<n>:COUPling?

■ **功能描述：**

用于设置通道的耦合方式。DC（直流）表示可通过输入信号的交流和直流分量；AC（交流）表示阻挡输入信号的直流分量；GND（接地）表示断开输入信号。

<n>: {1|2|3|4}，分别表示{CH1|CH2|CH3|CH4}。

■ **返回格式：**

查询 AC、DC 或 GND。

■ **举例：**

:CHAN1:COUP DC

设置通道 1 的耦合方式为直流。

:CHAN1:COUP?

查询返回 DC。

### **:CHANnel<n>:DISPlay**

■ **命令格式：**

:CHANnel<n>:DISPlay { {1|ON} | {0|OFF} }

:CHANnel<n>:DISPlay?

■ **功能描述：**

用于设置指定通道 ON（打开）或 OFF（关闭）。

<n>: {1|2|3|4}，分别表示{CH1|CH2|CH3|CH4}。

■ **返回格式：**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例：**

:CHAN1:DISP ON

打开通道 1。

:CHAN1:DISP?

查询返回 1，表示已经打开通道 1。

### **:CHANnel<n>:INVert**

■ **命令格式：**

:CHANnel<n>:INVert { {1|ON} | {0|OFF} }

:CHANnel<n>:INVert?











查询返回< scale>值。采用科学计数法，单位为 s/div。

■ **举例：**

:TIM:SCAL 2

设置主时基偏移量为 2s/div。

:TIM:SCAL?

查询返回 2.000e000。

### **:TIMebase:WINDow:SCALe**

■ **命令格式：**

:TIMebase:WINDow:SCALe < scale >

:TIMebase:WINDow:SCALe?

■ **功能描述：**

用于设置 WINDow（缩放时基<Zoomed>）时基档位，即 s/div（秒/格）。

■ **返回格式：**

查询返回< scale>值。采用科学计数法，单位为 s/div。

■ **举例：**

:TIM:WIND:SCAL 2

设置 WINDow 时基偏移量为 2s/div。

:TIM:WIND:SCAL?

查询返回 2.000e000。

### **:TIMebase:INDPendent**

■ **命令格式：**

:TIMebase:INDPendent { {1|ON} | {0|OFF} }

:TIMebase:INDPendent?

■ **功能描述：**

用于打开和关闭时基独立模式。

■ **返回格式：**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例：**

:TIM:INDP ON

打开时基独立模式。

:TIM:INDP?

查询返回 1。

### **:TIMebase:HOLDoff**

■ **命令格式：**

:TIMebase:HOLDoff <time>

:TIMebase:HOLDoff?

■ **功能描述：**

用于设置触发释抑时间，范围 100ns~10s。

■ **返回格式：**





■ **返回格式:**

查询返回{RECTangular|HANNing|HAMMing|BMAN}。

■ **举例:**

:FUNction:SOUR1 CHAN1	将一通道作为源
:FUNC:FFT:WIND HAMM	加汉明窗
:FUNC:FFT:WIND?	查询返回 HAMMing

**:FUNction:FFT:DISPlay**

■ **命令格式:**

:FUNction:FFT:DISPlay {FULL| SPLIt| WATerfall1| WATerfall2| INDEpendent }  
:FUNction:FFT:DISPlay?

■ **功能描述:**

用于设置 FFT 显示模式。

■ **返回格式:**

查询返回{FULL| SPLIt| WATerfall1| WATerfall2| INDEpendent}, FULL 为全屏显示, SPLIt 为分屏显示  
WATerfall1 为瀑布图 1, WATerfall2 为瀑布图 2, INDEpendent 表示独立显示 FFT 频谱。

■ **举例:**

:FUNction:FFT:DISPlay FULL	设置 FFT 全屏显示
:FUNction:FFT:DISPlay?	返回 FULL

**:FUNction:FFT:WATerfall:SLICe**

■ **命令格式:**

:FUNction:FFT:WATerfall:SLICe <value>  
:FUNction:FFT:WATerfall:SLICe ?

■ **功能描述:**

用于选择 FFT 瀑布图的片段, 选择范围 1~200。

■ **返回格式:**

查询返回当前选择的是第几帧片段。

■ **举例:**

:FUNction:FFT:WATerfall:SLICe 100	设置 FFT 瀑布图选择第 100 帧片段
:FUNction:FFT:WATerfall:SLICe ?	查询返回 100

**:FUNction:FFT:POINts**

■ **命令格式:**

:FUNction:FFT:POINts {AUTO|1K|2K|4K|8K|16K|32K|64K|128K|256K|512K|1024K|2048K|4096K}  
:FUNction:FFT:POINts?

■ **功能描述:**

用于设置 FFT 点数。

■ **返回格式:**

查询返回{AUTO|1K|2K|4K|8K|16K|32K|64K|128K|256K|512K|1024K|2048K|4096K}。

■ **举例:**

:FUNCTION:FFT:POINTS 8K                    设置 FFT 点数为 8K

:FUNCTION:FFT:POINTS?                    返回 8K

**:FUNCTION:FFT:VTYPE**

■ **命令格式:**

:FUNCTION:FFT:VTYPE {VRMS|DBRMS}

:FUNCTION:FFT:VTYPE?

■ **功能描述:**

选择 FFT 垂直方向的单位为 dBRMS 或者 VRMS。dBRMS 表示功率均方根，VRMS 表示电压均方根。

■ **返回格式:**

查询返回 VRMS、DBRMS。

■ **举例:**

:FUNCTION:SOUR1 CHAN1                    将一通道作为源

:FUNC:FFT:VTYP VRMS                    设置 FFT 垂直方向的单位电压均方根

:FUNC:FFT:VTYP?                    查询返回 VRMS

**:FUNCTION:FFT:FREQUENCY**

■ **命令格式:**

:FUNCTION:FFT:FREQUENCY?

■ **功能描述:**

获得 FFT 之后频谱波形的中心频率。

■ **返回格式:**

查询返回 FFT 之后频谱波形的中心频率，单位为 Hz。

■ **举例:**

:FUNCTION:FFT:FREQUENCY?                    查询返回 1.000e003

**:FUNCTION:FFT:FREQUENCY:START**

■ **命令格式:**

:FUNCTION:FFT:FREQUENCY:START <freq>

:FUNCTION:FFT:FREQUENCY:START?

■ **功能描述:**



设置 FFT 的起始频率。

■ **返回格式:**

查询返回 FFT 的起始频率, 单位为 Hz。

■ **举例:**

:FUNCTION:FFT:FREQUENCY:START 1KHz      设置起始频率为 1KHz

:FUNC:FFT:FREQ:START?      查询返回 1.000e003

**:FUNCTION:FFT:FREQUENCY:END**

■ **命令格式:**

:FUNCTION:FFT:FREQUENCY:END <freq>

:FUNCTION:FFT:FREQUENCY:END?

■ **功能描述:**

设置 FFT 的终止频率。

■ **返回格式:**

查询返回 FFT 的终止频率, 单位为 Hz。

■ **举例:**

:FUNCTION:FFT:FREQUENCY:END 1KHz      设置终止频率为 1KHz

:FUNC:FFT:FREQ:END?      查询返回 1.000e003

**:FUNCTION:FFT:FREQUENCY:CENTER**

■ **命令格式:**

:FUNCTION:FFT:FREQUENCY:CENTER <freq>

:FUNCTION:FFT:FREQUENCY:CENTER?

■ **功能描述:**

设置 FFT 的中心频率。

■ **返回格式:**

查询返回 FFT 的中心频率, 单位为 Hz。

■ **举例:**

:FUNCTION:FFT:FREQUENCY:CENTER 1KHz      设置中心频率为 1KHz

:FUNC:FFT:FREQ:CENTER?      查询返回 1.000e003

**:FUNCTION:FFT:FREQUENCY:BW**

■ **命令格式:**

:FUNCTION:FFT:FREQUENCY:BW <freq>

:FUNCTION:FFT:FREQUENCY:BW?

■ **功能描述:**

设置 FFT 的频率带宽。

■ **返回格式:**

查询返回 FFT 的频率带宽, 单位为 Hz。

■ **举例:**

:FUNCTION:FFT:FREQUENCY:BW 1KHz                    设置频率带宽为 1KHz  
:FUNC:FFT:FREQ:BW?                                    查询返回 1.000e003

**:FUNCTION:FFT:DETECTION:REALTIME**

■ **命令格式:**

:FUNCTION:FFT:DETECTION:REALTIME {OFF|PPEAK|NPEAK| AVERAGE| SAMPLE}  
:FUNCTION:FFT:DETECTION:REALTIME?

■ **功能描述:**

设置实时频谱的检波模式。

OFF: 关闭检波模式; PPEAK: 取每个抽点区间内的最大值; NPEAK: 取每个抽点区间内的最小值;  
AVERAGE: 取每个抽点区间内的平均值; SAMPLE: 取每个抽点区间内第一个点的值。

■ **返回格式:**

查询返回实时频谱的检波模式。

■ **举例:**

:FUNCTION:FFT:DETECTION:REALTIME PPEAK 设置实时频谱的检波模式为+峰值检波  
:FUNCTION:FFT:DETECTION:REALTIME?                    查询返回 PPEAK

**:FUNCTION:FFT:DETECTION:AVERAGE**

■ **命令格式:**

:FUNCTION:FFT:DETECTION:AVERAGE {OFF|PPEAK|NPEAK|AVERAGE|SAMPLE}  
:FUNCTION:FFT:DETECTION:AVERAGE?

■ **功能描述:**

设置平均值频谱的检波模式。

OFF: 关闭平均值频谱; PPEAK: 取每个抽点区间内的最大值; NPEAK: 取每个抽点区间内的最小值;  
AVERAGE: 取每个抽点区间内的平均值; SAMPLE: 取每个抽点区间内第一个点的值。

■ **返回格式:**

查询返回平均值频谱的检波模式。

■ **举例:**

:FUNCTION:FFT:DETECTION:AVERAGE PPEAK 设置平均值频谱的检波模式为+峰值检波  
:FUNCTION:FFT:DETECTION:AVERAGE?                    查询返回 PPEAK

## :FUNCTION:FFT:DETECTION:AVERAGE:COUNT

### ■ 命令格式:

:FUNCTION:FFT:DETECTION:AVERAGE:COUNT <value>

:FUNCTION:FFT:DETECTION:AVERAGE:COUNT?

### ■ 功能描述:

设置平均频谱的平均次数，范围  $2^n$  次方，其中 n 取值 1~10。

### ■ 返回格式:

查询平均频谱的平均次数。

### ■ 举例:

:FUNCTION:FFT:DETECTION:AVERAGE:COUNT 64 设置平均值频谱平均次数为 64 次

:FUNCTION:FFT:DETECTION:AVERAGE:COUNT? 查询返回 64

## :FUNCTION:FFT:DETECTION:MAXHOLD

### ■ 命令格式:

:FUNCTION:FFT:DETECTION:MAXHOLD {OFF|PPEAK|NPEAK| AVERAGE|SAMPLE}

:FUNCTION:FFT:DETECTION:MAXHOLD?

### ■ 功能描述:

设置最大值保持频谱的检波模式。

OFF: 关闭平均值频谱；PPEAK: 取每个抽点区间内的最大值；NPEAK: 取每个抽点区间内的最小值；

AVERAGE: 取每个抽点区间内的平均值；SAMPLE: 取每个抽点区间内第一个点的值。

### ■ 返回格式:

查询返回最大值保持频谱的检波模式。

### ■ 举例:

:FUNCTION:FFT:DETECTION:MAXHOLD PPEAK

设置最大保持频谱的检波模式为+峰值检波

:FUNCTION:FFT:DETECTION:MAXHOLD?

查询返回 PPEAK

## :FUNCTION:FFT:DETECTION:MINHOLD

### ■ 命令格式:

:FUNCTION:FFT:DETECTION:MINHOLD {OFF|PPEAK|NPEAK| AVERAGE|SAMPLE}

:FUNCTION:FFT:DETECTION:MINHOLD?

### ■ 功能描述:

设置最小值保持频谱的检波模式。

OFF: 关闭平均值频谱；PPEAK: 取每个抽点区间内的最大值；NPEAK: 取每个抽点区间内的最小值；

AVERAGE: 取每个抽点区间内的平均值；SAMPLE: 取每个抽点区间内第一个点的值。

■ **返回格式:**

查询返回最小值保持频谱的检波模式。

■ **举例:**

:FUNCTION:FFT:DETEction:MINHold PPEAK      设置频谱的检波模式为+峰值检波  
:FUNCTION:FFT:DETEction:MINHold?              查询返回 PPEAK

**:FUNCTION:FFT:DETEction:RESet**

■ **命令格式:**

:FUNCTION:FFT:DETEction:RESet

■ **功能描述:**

用于重置平均值、最大保持、最小保持频谱。

■ **举例:**

:FUNCTION:FFT:DETEction:RESet                      重置各频谱

**:FUNCTION:FFT:MARK:TYPE**

■ **命令格式:**

:FUNCTION:FFT:MARK:TYPE {AUTO|THReshold| MANUal}  
:FUNCTION:FFT:MARK:TYPE?

■ **功能描述:**

设置频谱标记类型。

AUTO: 自动频谱标记; THReshold: 门限频谱标记; MANUal: 手动频谱标记。

■ **返回格式:**

查询返回当前选择的频谱标记类型。

■ **举例:**

:FUNCTION:FFT:MARK:TYPE AUTO                      设置频谱标记类型为自动标记  
:FUNCTION:FFT:MARK:TYPE?                          查询返回 AUTO

**:FUNCTION:FFT:MARK:SOURce**

■ **命令格式:**

:FUNCTION:FFT:MARK:SOURce {REALtime|AVERAge|MAXHold|MINHold}  
:FUNCTION:FFT:MARK:SOURce?

■ **功能描述:**

设置频谱标记的标记源, 此指令频谱标记各类型共用指令。

REALtime: 标记实时频谱; AVERAge: 标记均值频谱; MAXHold: 标记最大保持频谱;  
MINHold: 标记最小保持频谱。

■ **返回格式:**

查询返回当前选择的标记源。

■ **举例：**

:FUNCTION:FFT:MARK:SOURce AVERage      设置频谱标记的标记源为均值频谱

:FUNCTION:FFT:MARK:SOURce?      查询返回 AVERage

**:FUNCTION:FFT:MARK:POINts**

■ **命令格式：**

:FUNCTION:FFT:MARK:POINts <value>

:FUNCTION:FFT:MARK:POINts ?

■ **功能描述：**

设置频谱标记的标记点数。

<value>: 标记点数的值，范围 1~50。

■ **返回格式：**

查询返回频谱标记的标记点数。

■ **举例：**

:FUNCTION:FFT:MARK:POINts 20      设置频谱标记的标记点数为 20

:FUNCTION:FFT:MARK:POINts?      查询返回 20

**:FUNCTION:FFT:MARK:EVENT**

■ **命令格式：**

:FUNCTION:FFT:MARK:EVENT {1 | ON} | {0 | OFF}

:FUNCTION:FFT:MARK:EVENT?

■ **功能描述：**

频谱标记打开或关闭标记列表。

■ **返回格式：**

查询标记列表打开状态，1 表示打开，0 表示关闭。

■ **举例：**

:FUNCTION:FFT:MARK:EVENT ON      打开标记列表

:FUNCTION:FFT:MARK:EVENT?      查询返回 1

**:FUNCTION:FFT:MARK:DATA?**

■ **命令格式：**

:FUNCTION:FFT:MARK:DATA?

■ **功能描述：**

用于读取示波器的 FFT 下标记事件表数据。

■ **返回格式：**

查询返回 FFT 下标记事件的数据，返回的数据符合附录 2：IEEE 488.2 二进制数据格式。

■ **举例：**

:FUNCTION:FFT:MARK:DATA?查询返回：

#9000000089FFT,

ID,Freq,Amp,

1,1.000E+003,7.800E-001,

2,2.000E+003,7.900E-001,

3,3.000E+003,7.700E-001,

4,4.000E+003,7.300E-001,

5,5.000E+003,7.400E-001,

FFT 表示 FFT 模式，后面紧跟 CSV 格式的事件表数据，事件表数据的具体格式由不同设备自动适配，数据以逗号分开，会根据列表的行自动换行。

**:FUNCTION:FFT:MARK:THReshold:LEVel**

■ **命令格式：**

:FUNCTION:FFT:MARK:THReshold:LEVel <value>

:FUNCTION:FFT:MARK:THReshold:LEVel?

■ **功能描述：**

设置门限频谱标记的门限电压。

<value>: 门限电压值, 当垂直单位为 Vrms 时, 单位为 V, 范围 1mVrms~100KVrms; 当垂直单位为 dBVrms 时, 单位为 dB, 范围-60dB~100dB; 当单位与垂直单位不匹配时, 设置无效。

■ **返回格式：**

查询返回频谱标记的阈值，以科学计数法表示，单位与:FUNCTION:FFT:VTYPE 相关。

■ **举例：**

:FUNCTION:FFT:MARK:THReshold:LEVel -12.5dB 设置门限频谱标记的门限为-12.5dB

:FUNCTION:FFT:MARK:THReshold:LEVel? 查询返回-1.250e-001

:FUNCTION:FFT:MARK:THReshold:LEVel 0.15V 设置门限频谱标记的门限为 0.15V

:FUNCTION:FFT:MARK:THReshold:LEVel? 查询返回 1.500e-001

**:FUNCTION:FFT:MARK:MANUal:PEAK**

■ **命令格式：**

:FUNCTION:FFT:MARK:MANUal:PEAK

■ **功能描述：**

用于将标记移动到最大峰值处。

■ **举例：**

:FUNCTION:FFT:MARK:MANUal:PEAK 标记移动到最大峰值处

## :FUNCTION:FILTER:TYPE

### ■ 命令格式:

:FUNCTION:FILTER:TYPE {LP|HP|BP|BS}

:FUNCTION:FILTER:TYPE?

### ■ 功能描述:

设置滤波器类型。LP、HP、BP、BS 分别表示低通滤波器，高通滤波器，带通滤波器，带阻滤波器。

### ■ 返回格式:

查询返回 LP、HP、BP、BS。

### ■ 举例:

:FUNCTION:SOUR1 CHAN1	将一通道作为源
:FUNC:FILT:TYPE BP	设置为带通滤波器
:FUNC:FILT:TYPE?	查询返回 BP

## :FUNCTION:FILTER:FREQUENCY:HIGH

### ■ 命令格式:

:FUNCTION:FILTER:FREQUENCY:HIGH <freq>

:FUNCTION:FILTER:FREQUENCY:HIGH?

### ■ 功能描述:

设置滤波器上限截止频率值。适用于高通滤波器、带通滤波器、带阻滤波器。

### ■ 返回格式:

查询返回 1.000e003，单位为 Hz。

### ■ 举例:

:FUNCTION:SOUR1 CHAN1	将一通道作为源
:FUNC:FILT:FREQ:HIGH 1KHz	设置滤波器上限为 1KHz 截止频率
:FUNC:FILT:FREQ:HIGH?	查询返回 1.000e003

## :FUNCTION:FILTER:FREQUENCY:LOW

### ■ 命令格式:

:FUNCTION:FILTER:FREQUENCY:LOW <freq>

:FUNCTION:FILTER:FREQUENCY:LOW?

### ■ 功能描述:

设置滤波器上限截止频率值。适用于低通滤波器、带通滤波器、带阻滤波器。

### ■ 返回格式:

查询返回 6.000e001，单位为 Hz。

### ■ 举例:

:FUNC:SOUR1 CHAN1	将一通道作为源
-------------------	---------

:FUNC:FILT:FREQ:LOW 60Hz

设置滤波器下限为 60Hz 截止频率

:FUNC:FILT:FREQ:LOW?

查询返回 6.000e001

## :FUNCtion:EXPRession

### ■ 命令格式:

:FUNCtion:EXPRession <expression>

### ■ 功能描述:

利用自由组合表达式进行数学计算。

表达式格式可见示波器下 MATH 菜单下 Advance 选项, <expression>属于 ASCII 字符串参数。

### ■ 举例:

:FUNCtion:EXPRession "CH1\*CH2"

表示将通道一和二相乘。

## MEASure 命令

用于示波器最基本的测量操作, 所有参数测量在不需要打开测量就能获取到测量值, 默认获取测量值时自动打开测量并获得值, 通常以科学计数方式返回测量结果。

## :MEASure:ALL

### ■ 命令格式:

:MEASure:ALL {{1 | ON} | {0 | OFF}}

:MEASure:ALL?

### ■ 功能描述:

用于打开或关闭全部测量功能。

### ■ 返回格式:

查询返回是否打开全部测量功能。

### ■ 举例:

:MEASure:ALL ON

打开全部测量功能

:MEASure:ALL?

查询返回 1

## :MEASure:CLEar

### ■ 命令格式:

:MEASure:CLEar

### ■ 功能描述:

用于清除当前测量的参数值。

### ■ 举例:

:MEAS:CLE

清除当前测量的参数值







### :MEASure:PDUTy?

- **命令格式:**

:MEASure:PDUTy? [<source>]

- **功能描述:**

用于测量指定通道波形的正占空比。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

- **返回格式:**

查询返回 5.000e001，单位%。

### :MEASure:NDUTy?

- **命令格式:**

:MEASure:NDUTy? [<source>]

- **功能描述:**

用于测量指定通道波形的负占空比。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

- **返回格式:**

查询返回 5.000e001，单位%。

### :MEASure:PDElay?

- **命令格式:**

:MEASure:PDElay? [<source1>,<source2>]

- **功能描述:**

用于测量<source1>、<source2>相对于上升沿的时间延迟。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4。

- **返回格式:**

查询返回-1.000e-004，单位为 s。

- **举例:**

测量相对于上升沿的时间延迟

:MEASure:PDEL? CHAN1,CHAN2

### :MEASure:NDElay?

- **命令格式:**

:MEASure:NDElay? [<source1>,<source2>]

- **功能描述:**

用于测量<source1>、<source2>相对于下降沿的时间延迟。其中<source>取值为 CHANnel1、CHANnel2、

CHANnel3 或者 CHANnel4。

- **返回格式:**

查询返回-1.000e-004, 单位为 s。

- **举例:**

测量相对于下降沿的时间延迟

:MEASure:NDEL? CHAN1,CHAN2

### **:MEASure:PHASe?**

- **命令格式:**

:MEASure:PHASe? [<source1>,<source2>]

- **功能描述:**

用于定时测量<source1>相对于<source2>超前或者滞后的时间量, 以度表示, 360°为一周期。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4。

- **返回格式:**

查询返回 1.000e001, 单位为度。

- **举例:**

测量<source1>相对于<source2>超前或者滞后的时间量

:MEASure:PHAS? CHAN1,CHAN2

### **:MEASure:VPP?**

- **命令格式:**

:MEASure:VPP? [<source>]

- **功能描述:**

用于测量指定通道波形的峰峰值。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4, 省略表示当前通道。

- **返回格式:**

查询返回 3.120e000, 单位为 V。

### **:MEASure:VMAX?**

- **命令格式:**

:MEASure:VMAX? [<source>]

- **功能描述:**

用于测量指定通道波形的最大值。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4, 省略表示当前通道。

- **返回格式:**

查询返回 2.120e000, 单位为 V。

### :MEASure:VMIN?

- **命令格式:**

:MEASure:VMIN? [<source>]

- **功能描述:**

用于测量指定通道波形的最小值。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

- **返回格式:**

查询返回-2.120e000，单位为 V。

### :MEASure:VAMPlitude?

- **命令格式:**

:MEASure:VAMPlitude? [<source>]

- **功能描述:**

用于测量指定通道波形的幅值。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

- **返回格式:**

查询返回 3.120e000，单位为 V。

### :MEASure:VTOP?

- **命令格式:**

:MEASure:VTOP? [<source>]

- **功能描述:**

用于测量指定通道波形的顶端值。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

- **返回格式:**

查询返回 3.120e000，单位为 V。

### :MEASure:VBASe?

- **命令格式:**

:MEASure:VBASe? [<source>]

- **功能描述:**

用于测量指定通道波形的底端值。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

- **返回格式:**

查询返回-3.120e000，单位为 V。

### :MEASure:VMIDdle?

- **命令格式:**

:MEASure:VMIDdle? [<source>]

- **功能描述:**

用于测量指定通道波形的中间值。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

- **返回格式:**

查询返回 0.120e000，单位为 V。

### :MEASure:VAverage?

- **命令格式:**

:MEASure:VAverage? [<interval>][,<source>]

- **功能描述:**

用于测量指定通道波形的平均值。其中<source>指定通道，取值为 CHANnel1、CHANnel2、CHANnel3、CHANnel4，如果没有指定<source>，则默认当前通道；<interval>指定测量间距，取值为 CYCLE、DISPlay，其中 CYCLE 表示整数循环周期，DISPlay 表示全屏，如果没有指定<interval>，则默认 DISPlay。

- **返回格式:**

查询返回 1.120e000，单位为 V。

### :MEASure:VRMS?

- **命令格式:**

:MEASure:VRMS? [<interval>][,<source>]

- **功能描述:**

用于测量指定通道波形的均方根值。其中<source>指定通道，取值为 CHANnel1、CHANnel2、CHANnel3、CHANnel4，如果没有指定<source>，则默认当前通道；<interval>指定测量间距，取值为 CYCLE、DISPlay，其中 CYCLE 表示整数循环周期，DISPlay 表示全屏，如果没有指定<interval>，则默认 DISPlay。

- **返回格式:**

查询返回 1.230e000，单位为 V。

### :MEASure:ACRMs?

- **命令格式:**

:MEASure:ACRMs? [<source>]

- **功能描述:**

用于测量指定通道波形的 AC 均方根值。其中<source>指定通道，取值为 CHANnel1、CHANnel2、CHANnel3、CHANnel4，如果没有指定<source>，则默认当前通道。

- **返回格式:**

查询返回 1.230e000, 单位为 V。

#### **:MEASure:AREa?**

- **命令格式:**

:MEASure:AREa? [<interval>][,<source>]

- **功能描述:**

用于测量指定通道波形的面积。其中<source>指定通道, 取值为 CHANnel1、CHANnel2、CHANnel3、CHANnel4, 如果没有指定<source>, 则默认当前通道; <interval>指定测量间距, 取值为 CYCLe、DISPlay, 其中 CYCLe 表示整数循环周期, DISPlay 表示全屏, 如果没有指定<interval>, 则默认 DISPlay。

- **返回格式:**

查询返回 3.456e002, 单位为 Vs。

#### **:MEASure:OVERshoot?**

- **命令格式:**

:MEASure:OVERshoot? [<source>]

- **功能描述:**

用于测量指定通道波形的过冲。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4, 省略表示当前通道。

- **返回格式:**

查询返回 1.230e002, 单位为 %。

#### **:MEASure:PREShoot?**

- **命令格式:**

:MEASure:PREShoot? [<source>]

- **功能描述:**

用于测量指定通道波形的预冲。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4, 省略表示当前通道。

- **返回格式:**

查询返回 1.230e-002, 单位为 %。

#### **:MEASure:FREQuency?**

- **命令格式:**

:MEASure:FREQuency? [<source>]

- **功能描述:**

用于测量指定通道波形的频率。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

■ **返回格式:**

查询返回 2.000e003, 单位 Hz。

**:MEASure:RISetime?**

■ **命令格式:**

:MEASure:RISetime? [<source>]

■ **功能描述:**

用于测量指定通道波形的上升时间。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

■ **返回格式:**

查询返回 5.000e-005, 单位 s。

**:MEASure:FALLtime?**

■ **命令格式:**

:MEASure:FALLtime? [<source>]

■ **功能描述:**

用于测量指定通道波形的下降时间。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

■ **返回格式:**

查询返回 5.000e-005, 单位 s。

**:MEASure:PERiod?**

■ **命令格式:**

:MEASure:PERiod? [<source>]

■ **功能描述:**

用于测量指定通道波形的周期。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4，省略表示当前通道。

■ **返回格式:**

查询返回 5.000e-003, 单位 s。

**:MEASure:PWIDth?**

■ **命令格式:**

:MEASure:PWIDth? [<source>]



■ **功能描述:**

用于测量指定通道波形的正脉宽。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4, 省略表示当前通道。

■ **返回格式:**

查询返回 5.000e-003, 单位 s。

**:MEASure:NWIDth?**

■ **命令格式:**

:MEASure:NWIDth? [<source>]

■ **功能描述:**

用于测量指定通道波形的负脉宽。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4, 省略表示当前通道。

■ **返回格式:**

查询返回 5.000e-003, 单位 s。

**:MEASure:PULSes?**

■ **命令格式:**

:MEASure:PULSes? [<source>]

■ **功能描述:**

用于测量指定通道波形的正脉冲个数。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4, 省略表示当前通道。

■ **返回格式:**

查询返回 2.000e+000, 表示 2 个脉冲。

**:MEASure:FRR?**

■ **命令格式:**

:MEASure:FRR? <source1>,<source2>

■ **功能描述:**

用于测量<source1>与<source2>第一个上升沿之间的时间。其中<source>取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4。

■ **返回格式:**

查询返回 5.000e-003, 单位 s。

**:MEASure:FRF?**

■ **命令格式:**

:MEASure:FRF? <source1>,<source2>

■ **功能描述:**

用于测量<source1>第一个上升沿与<source2>第一个下降沿之间的时间。其中<source>取值为 CHANnel1、CHANnel2、 CHANnel3 或者 CHANnel4，省略表示当前通道。

■ **返回格式:**

查询返回 5.000e-003，单位 s。

**:MEASure:FFR?**

■ **命令格式:**

:MEASure:FFR? <source1>,<source2>

■ **功能描述:**

用于测量<source1>第一个下降沿与<source2>第一个上升沿之间的时间。其中<source>取值为 CHANnel1、CHANnel2、 CHANnel3 或者 CHANnel4。

■ **返回格式:**

查询返回 5.000e-003，单位 s。

**:MEASure:FFF?**

■ **命令格式:**

:MEASure:FFF? <source1>,<source2>

■ **功能描述:**

用于测量<source1>与<source2>第一个下降沿之间的时间。其中<source>取值为 CHANnel1、CHANnel2、 CHANnel3 或者 CHANnel4

■ **返回格式:**

查询返回 5.000e-003，单位 s。

**:MEASure:LRR?**

■ **命令格式:**

:MEASure:LRR? <source1>,<source2>

■ **功能描述:**

用于测量<source1>第一个上升沿与<source2>最后一个上升沿之间的时间。其中<source>取值为 CHANnel1、CHANnel2、 CHANnel3 或者 CHANnel4。

■ **返回格式:**

查询返回 5.000e-003，单位 s。

## :MEASure:LRF?

- **命令格式:**

:MEASure:LRF? <source1>,<source2>

- **功能描述:**

用于测量 <source1> 第一个上升沿与 <source2> 最后一个下降沿之间的时间。其中 <source> 取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4。

- **返回格式:**

查询返回 5.000e-003, 单位 s。

## :MEASure:LFR?

- **命令格式:**

:MEASure:LFR? <source1>,<source2>

- **功能描述:**

用于测量 <source1> 第一个下降沿与 <source2> 最后一个上升沿之间的时间。其中 <source> 取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4。

- **返回格式:**

查询返回 5.000e-003, 单位 s。

## :MEASure:LFF?

- **命令格式:**

:MEASure:LFF? <source1>,<source2>

- **功能描述:**

用于测量 <source1> 第一个下降沿与 <source2> 最后一个下降沿之间的时间。其中 <source> 取值为 CHANnel1、CHANnel2、CHANnel3 或者 CHANnel4。

- **返回格式:**

查询返回 5.000e-003, 单位 s。

## TRIGger 命令

用于控制触发器扫描模式和触发器规范，触发决定了示波器何时开始采集数据和显示波形。

## 触发控制

### :TRIGger:MODE

- **命令格式:**



## :TRIGger:LEVel:ASETup

- **命令格式:**

:TRIGger:LEVel:ASETup

- **功能描述:**

用于将触发电平设置于信号幅值的垂直中点处。

- **举例:**

:TRIG:LEVel:ASETup                      将触发电平位置置于中心处

## :TRIGger:STATus?

- **命令格式:**

:TRIGger:STATus?

- **功能描述:**

查询当前的示波器触发运行状态。

- **返回格式:**

查询返回 STOP/ARMED/READY/TRIGED/AUTO/SCAN /RESET / REPLAY/ WAIT。

- **举例:**

:TRIGger:STATus?                      查询返回 AUTO

## :TRIGger:LEVel

- **命令格式:**

:TRIGger:LEVel <level>

:TRIGger:LEVel?

- **功能描述:**

用于设置正常触发的触发电平值。<level>数值必须根据幅格挡、屏幕信息换算后设置。

- **返回格式:**

查询返回<level>的设置值，单位 V。

- **举例:**

:TRIG:LEV 2                              设置触发的触发电平为 2V

:TRIG:LEV?                              查询返回 2.000e000

## :TRIGger:LEVel:LOW

- **命令格式:**

:TRIGger:LEVel:LOW <level>

:TRIGger:LEVel:LOW?

- **功能描述:**



## :TRIGger:COUPling

- **命令格式:**

:TRIGger:COUPling {DC|AC|LF|HF|NOISE}

:TRIGger:COUPling?

- **功能描述:**

用于设置耦合方式，DC（直流）、AC（交流）、LF（低频抑制）、HF（高频抑制）、NOISE（噪声抑制）。只有 VIDEO 不支持。

- **返回格式:**

查询返回耦合方式{DC|AC|LF|HF|NOISE}。

- **举例:**

:TRIGger:COUPling AC                      设置边沿触发为交流

:TRIGger:COUPling?                      查询返回 AC

## 边沿触发

## :TRIGger:EDGE:SLOPe

- **命令格式:**

:TRIGger:EDGE:SLOPe {POSitive|NEGative|ALTerNation}

:TRIGger:EDGE:SLOPe?

- **功能描述:**

用于设置触发的边沿类型，POSitive（上升沿）、NEGative（下降沿）、ALTerNation（上升下降沿）。

- **返回格式:**

查询返回触发的边沿类型{ POSitive | NEGative | ALTerNation }。

- **举例:**

:TRIGger:EDGE:SLOP POS                      设置边沿触发为上升沿

:TRIGger:EDGE:SLOP?                      查询返回 POSitive

## 脉宽触发

## :TRIGger:PULSe:QUALifier

- **命令格式:**

:TRIGger:PULSe:QUALifier {GREaterthan | LESSthan | INRange}

:TRIGger:PULSe:QUALifier?

- **功能描述:**

用于设置脉冲时间设置条件，GREaterthan（大于）、LESSthan（小于）、INRange（之间）。

- **返回格式:**

查询返回{GREaterthan | LESSthan | INRange}。

■ **举例：**

:TRIGger:PULSe:QUALifier GRE            设置脉冲条件为大于  
:TRIGger:PULSe:QUALifier?            查询返回 GREaterthan

**:TRIGger:PULSe:POLarity**

■ **命令格式：**

:TRIGger:PULSe:POLarity {POSitive | NEGative}  
:TRIGger:PULSe:POLarity?

■ **功能描述：**

用于设置脉冲极性，POSitive（正脉宽）、NEGative（负脉宽）。

■ **返回格式：**

查询返回{ POSitive | NEGative }。

■ **举例：**

:TRIGger:PULSe:POL POS            设置脉冲极性为正脉宽  
:TRIGger:PULSe:POL?            查询返回 POSitive

**:TRIGger:PULSe:TIME**

■ **命令格式：**

:TRIGger:PULSe:TIME <time>  
:TRIGger:PULSe:TIME?

■ **功能描述：**

用于设置脉宽触发时间间隔。

■ **返回格式：**

查询返回当前时间间隔，单位 s。

■ **举例：**

:TRIGger:PULSe:TIME 1            设置脉冲宽度为 1s  
:TRIGger:PULSe:TIME?            查询返回 1.000e000

**:TRIGger:PULSe:TIME:UPPer**

■ **命令格式：**

:TRIGger:PULSe:TIME:UPPer <time>  
:TRIGger:PULSe:TIME:UPPer?

■ **功能描述：**

用于设置脉宽触发时间上限。

■ **返回格式：**

查询返回当前时间上限，单位 s。

■ **举例：**

:TRIGger:PULSe:TIME:UPPer 1            设置脉冲触发上限为 1s  
:TRIGger:PULSe:TIME:UPPer?            查询返回 1.000e000



## :TRIGger:PULSe:TIME:LOWer

- **命令格式:**

:TRIGger:PULSe:TIME:LOWer <time>

:TRIGger:PULSe:TIME:LOWer?

- **功能描述:**

用于设置脉宽触发时间下限。

- **返回格式:**

查询返回当前时间下限，单位 s。

- **举例:**

:TRIGger:PULSe:TIME:LOWer 1

设置脉冲触发下限为 1s

:TRIGger:PULSe:TIME:LOWer?

查询返回 1.000e000

## 视频触发

### :TRIGger:VIDeo:MODE

- **命令格式:**

:TRIGger:VIDeo:MODE { ODD | EVEN | LINE | ALINes }

:TRIGger:VIDeo:MODE?

- **功能描述:**

用于设置视频触发同步方式，ODD（奇数）、EVEN（偶数）、LINE（指定行）、ALINes（所有行）。

- **返回格式:**

查询返回{ ODD | EVEN | LINE | ALIN }。

- **举例:**

:TRIGger:VIDeo:MODE ODD

设置视频触发同步模式为奇数场

:TRIGger:VIDeo:MODE?

查询返回 ODD

### :TRIGger:VIDeo:STANdard

- **命令格式:**

:TRIGger:VIDeo:STANdard { NTSC | PAL | SECAM }

:TRIGger:VIDeo:STANdard?

- **功能描述:**

用于设置视频标准。

- **返回格式:**

查询返回{ NTSC | PAL | SECAM }。

- **举例:**

:TRIGger:VIDeo:STANdard NTSC

设置视频标准为 NTSC

:TRIGger:VIDeo:STANdard?

查询返回 NTSC



:TRIGger:SLOPe:SLOPe?

查询返回 POSitive

### :TRIGger:SLOPe:TIME

- **命令格式:**

:TRIGger:SLOPe:TIME <time>

:TRIGger:SLOPe:TIME?

- **功能描述:**

用于设置斜率触发模式的时间。

- **返回格式:**

查询返回当前时间间隔，单位 s。

- **举例:**

:TRIGger:SLOPe:TIME 1                      斜率触发模式时间间隔设置 1s

:TRIGger:SLOPe:TIME?                      查询返回 1.000e000

### :TRIGger:SLOPe:TIME:UPPer

- **命令格式:**

:TRIGger:SLOPe:TIME:UPPer <time>

:TRIGger:SLOPe:TIME:UPPer?

- **功能描述:**

用于设置斜率触发时间上限。

- **返回格式:**

查询返回当前时间上限，单位 s。

- **举例:**

:TRIGger:SLOPe:TIME:UPPer 1                      设置斜率触发上限为 1s

:TRIGger:SLOPe:TIME:UPPer?                      查询返回 1.000e000

### :TRIGger:SLOPe:TIME:LOWer

- **命令格式:**

:TRIGger:SLOPe:TIME:LOWer <time>

:TRIGger:SLOPe:TIME:LOWer?

- **功能描述:**

用于设置斜率触发时间下限。

- **返回格式:**

查询返回当前时间下限，单位 s。

- **举例:**

:TRIGger:SLOPe:TIME:LOWer 1            设置斜率触发下限为 1s  
:TRIGger:SLOPe:TIME:LOWer?            查询返回 1.000e000

### :TRIGger:SLOPe:THReshold

■ **命令格式:**

:TRIGger:SLOPe:THReshold {LOW|HIGH|LH}  
:TRIGger:SLOPe:THReshold?

■ **功能描述:**

用于设置斜率触发模式的阈值模式。

■ **返回格式:**

查询返回{LOW|HIGH|LH}。

■ **举例:**

:TRIGger:SLOPe:THR HIGH            斜率触发阈值为高模式  
:TRIGger:SLOPe:THR?            查询返回 HIGH

### :TRIGger:SLOPe:RATE:LOWer?

■ **命令格式:**

:TRIGger:SLOPe:RATE:LOWer?

■ **功能描述:**

查询当前斜率触发斜率下限值。

■ **返回格式:**

查询返回触发斜率下限值，采用科学计数法。

■ **举例:**

:TRIGger:SLOPe:RATE:LOWer?            查询返回 3.210E+000

### :TRIGger:SLOPe:RATE:UPPer?

■ **命令格式:**

:TRIGger:SLOPe:RATE:UPPer?

■ **功能描述:**

查询当前斜率触发斜率上限值。

■ **返回格式:**

查询返回触发斜率上限值，采用科学计数法。

■ **举例:**

:TRIGger:SLOPe:RATE:UPPer?            查询返回 3.210E+000

## 欠幅触发

### :TRIGger:RUNT:QUALifier

■ **命令格式:**

:TRIGger:RUNT:QUALifier {GREATERthan | LESSthan | INRange | NONE}

:TRIGger:RUNT:QUALifier?

■ **功能描述:**

用于设置矮电平时间设置条件，GREATERthan（大于）、LESSthan（小于）、INRange（范围内）、NONE（任意）。

■ **返回格式:**

查询返回{GREATERthan | LESSthan | INRange | NONE}。

■ **举例:**

:TRIGger:RUNT:QUALifier GRE                    设置斜率条件为大于

:TRIGger:RUNT:QUALifier?                    查询返回 GREATERthan

### :TRIGger:RUNT:POLarity

■ **命令格式:**

:TRIGger:RUNT:POLarity {POSitive | NEGative}

:TRIGger:RUNT:POLarity?

■ **功能描述:**

用于设置矮电平极性，POSitive（正脉宽）、NEGative（负脉宽）。

■ **返回格式:**

查询返回{POSitive | NEGative}。

■ **举例:**

:TRIGger:RUNT:POL POS                    设置脉冲极性为正脉宽

:TRIGger:RUNT:POL?                    查询返回 POSitive

### :TRIGger:RUNT:LEVel

■ **命令格式:**

:TRIGger:RUNT:LEVel {LOW | HIGH | LH }

:TRIGger:RUNT:LEVel?

■ **功能描述:**

用于设置矮电平触发电平模式。

■ **返回格式:**

查询返回{LOW | HIGH | LH }。

■ **举例:**

:TRIGger:RUNT:LEV HIGH                    设置矮电平为高触发电平

:TRIGger:RUNT:LEV?                    查询返回 HIGH

## :TRIGger:RUNT:TIME

### ■ 命令格式:

:TRIGger:RUNT:TIME <time>

:TRIGger:RUNT:TIME?

### ■ 功能描述:

用于设置矮电平触发时间间隔。

### ■ 返回格式:

查询返回当前时间间隔, 单位 s。

### ■ 举例:

:TRIGger:RUNT:TIME 1

设置矮电平触发模式时间间隔 1s

:TRIGger:RUNT:TIME?

查询返回 1.000e000

## :TRIGger:RUNT:TIME:UPPer

### ■ 命令格式:

:TRIGger:RUNT:TIME:UPPer <time>

:TRIGger:RUNT:TIME:UPPer?

### ■ 功能描述:

用于设置矮电平触发时间上限。

### ■ 返回格式:

查询返回当前时间上限, 单位 s。

### ■ 举例:

:TRIGger:RUNT:TIME:UPPer 1

设置矮电平触发上限为 1s

:TRIGger:RUNT:TIME:UPPer?

查询返回 1.000e000

## :TRIGger:RUNT:TIME:LOWer

### ■ 命令格式:

:TRIGger:RUNT:TIME:LOWer <time>

:TRIGger:RUNT:TIME:LOWer?

### ■ 功能描述:

用于设置矮电平触发时间下限。

### ■ 返回格式:

查询返回当前时间下限, 单位 s。

### ■ 举例:

:TRIGger:RUNT:TIME:LOWer 1

设置矮电平触发下限为 1s

:TRIGger:RUNT:TIME:LOWer?

查询返回 1.000e000

## 超幅触发

### :TRIGger:WINDow:SLOPe

#### ■ 命令格式:

:TRIGger:WINDow:SLOPe {POSitive|NEGative|ALTerNation}

:TRIGger:WINDow:SLOPe?

#### ■ 功能描述:

用于设置触发的边沿类型，POSitive（上升沿）、NEGative（下降沿）、ALTerNation（上升下降沿）。

#### ■ 返回格式:

查询返回触发的边沿类型{POSitive|NEGative|ALTerNation}。

#### ■ 举例:

:TRIGger:WINDow:SLOP POS                      设置窗口触发为上升沿

:TRIGger:WINDow:SLOP?                      查询返回 POS

### :TRIGger:WINDow:LEVel

#### ■ 命令格式:

:TRIGger:WINDow:LEVel {LOW | HIGH | LH }

:TRIGger:WINDow:LEVel?

#### ■ 功能描述:

用于设置窗口触发电平模式。

#### ■ 返回格式:

查询返回{LOW | HIGH | LH }。

#### ■ 举例:

:TRIGger:WINDow:LEV HIGH                      设置窗口触发为高触发电平

:TRIGger:WINDow:LEV?                      查询返回 HIGH

### :TRIGger:WINDow:TIME

#### ■ 命令格式:





用于设置延迟触发的焦点源。

■ **返回格式：**

查询返回{CHANnel1 | CHANnel2| CHANnel3| CHANnel4}。

■ **举例：**

:TRIGger:DElay:ARM:SOUR CHAN1	设置通道一焦点源
:TRIGger:DElay:ARM:SOUR?	查询返回 CHANnel1

**:TRIGger:DElay:ARM:SLOPe**

■ **命令格式：**

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

■ **功能描述：**

用于设置触发的边沿类型，POSitive（上升沿）、NEGative（下降沿）。

■ **返回格式：**

查询返回{NEGative | POSitive}。

■ **举例：**

:TRIGger:DElay:ARM:SLOPe NEG	设置触发源的边沿类型为下降沿
:TRIGger:DElay:ARM:SLOPe?	查询返回 NEGative

**:TRIGger:DElay:TRIGger:SOURce**

■ **命令格式：**

:TRIGger:DElay:TRIGger:SOURce {CHANnel1 | CHANnel2| CHANnel3| CHANnel4}  
:TRIGger:DElay:TRIGger:SOURce?

■ **功能描述：**

用于设置延迟触发的触发源。

■ **返回格式：**

查询返回{CHANnel1 | CHANnel2| CHANnel3| CHANnel4}。

■ **举例：**

:TRIGger:DElay:TRIGger:SOUR CHAN1	设置通道一触发源
:TRIGger:DElay:TRIGger:SOUR?	查询返回 CHANnel1

**:TRIGger:DElay:TRIGger:SLOPe**

■ **命令格式：**

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

■ **功能描述：**

用于设置触发的边沿类型，POSitive（上升沿）、NEGative（下降沿）。

■ **返回格式：**

查询返回{NEGative | POSitive}。

■ **举例：**

:TRIGger:DElay:TRIGger:SLOPe NEG                    设置触发源的边沿类型为下降沿  
:TRIGger:DElay:TRIGger:SLOPe?                    查询返回 NEGative

### :TRIGger:DElay:QUALifier

■ **命令格式:**

:TRIGger:DElay:QUALifier { GREaterthan | LESSthan | INRange | OUTRange }  
:TRIGger:DElay:QUALifier?

■ **功能描述:**

用于设置延迟触发时间间隔条件, GREaterthan(大于)、LESSthan(小于)、INRange(范围内)、OUTRange(范围外)。

■ **返回格式:**

查询返回{ GREaterthan | LESSthan | INRange | OUTRange }。

■ **举例:**

:TRIGger:DElay:QUALifier GRE                    设置斜率条件为大于  
:TRIGger:DElay:QUALifier?                    查询返回 GREaterthan

### :TRIGger:DElay:TIME

■ **命令格式:**

:TRIGger:DElay:TIME <time>  
:TRIGger:DElay:TIME?

■ **功能描述:**

用于设置延迟触发时间间隔。

■ **返回格式:**

查询返回当前时间间隔, 单位 s。

■ **举例:**

:TRIGger:DElay:TIME 1                    设置延迟触发时间间隔 1s  
:TRIGger:DElay:TIME?                    查询返回 1.000e000

### :TRIGger:DElay:TIME:UPPer

■ **命令格式:**

:TRIGger:DElay:TIME:UPPer <time>  
:TRIGger:DElay:TIME:UPPer?

■ **功能描述:**

用于设置延迟触发时间上限。

■ **返回格式:**

查询返回当前时间上限, 单位 s。

■ **举例:**

:TRIGger:DElay:TIME:UPPer 1                    设置延迟触发上限为 1s  
:TRIGger:DElay:TIME:UPPer?                    查询返回 1.000e000

## :TRIGger:DELay:TIME:LOWer

### ■ 命令格式:

:TRIGger:DELay:TIME:LOWer <time>

:TRIGger:DELay:TIME:LOWer?

### ■ 功能描述:

用于设置延迟触发时间下限。

### ■ 返回格式:

查询返回当前时间下限，单位 s。

### ■ 举例:

:TRIGger:DELay:TIME:LOWer 1

设置延迟触发下限为 1s

:TRIGger:DELay:TIME:LOWer?

查询返回 1.000e000

## :TRIGger:DELay:SElect

### ■ 命令格式:

:TRIGger:DELay:SElect <SOURce<n>>

:TRIGger:DELay:SElect

### ■ 功能描述:

用于切换选中的源。SOURce<n>代表源，n 取值 1、2。

SOURce1 表示焦点源；SOURce2 表示触发源。

### ■ 返回格式:

查询返回{ SOURce1 | SOURce2 }。

### ■ 举例:

:TRIGger:DELay:SElect SOURce1

设置选中焦点源

:TRIGger:DELay:SElect?

查询返回 SOURce1

## 超时触发

## :TRIGger:TIMEOUT:TIME

### ■ 命令格式:

:TRIGger:TIMEOUT:TIME <time>

:TRIGger:TIMEOUT:TIME?

### ■ 功能描述:

用于设置超时触发时间间隔。

### ■ 返回格式:

查询返回当前时间间隔，单位 s。

### ■ 举例:

:TRIGger:TIMEOUT:TIME 1

设置超时触发模式时间间隔 1s

:TRIGger:TIMEOUT:TIME?

查询返回 1.000e000

## :TRIGger:TIMEOUT:SLOPe

### ■ 命令格式:

:TRIGger:TIMEOUT:SLOPe {POSitive|NEGative|ALTernation}

:TRIGger:TIMEOUT:SLOPe?

### ■ 功能描述:

用于设置触发的边沿类型, POSitive (上升沿)、NEGative (下降沿)、ALTernation (上升下降沿)。

### ■ 返回格式:

查询返回触发的边沿类型{POSitive|NEGative|ALTernation}。

### ■ 举例:

:TRIGger:TIMEOUT:SLOP POS                      设置边沿触发为上升沿

:TRIGger:TIMEOUT:SLOP?                      查询返回 POSitive

## 持续时间触发

## :TRIGger:DURation:PATtern

### ■ 命令格式:

:TRIGger:DURation:PATtern { HIGH | LOW | X }

:TRIGger:DURation:PATtern?

### ■ 功能描述:

用于设置持续触发码型, HIGH (码型值为 1)、LOW (码型值为 0)、X (通道无效)。

### ■ 返回格式:

查询返回{ HIGH | LOW | X }。

### ■ 举例:

:TRIGger:DURation:PATtern HIGH              设置持续触发码型为 1

:TRIGger:DURation:PATtern?              查询返回 HIGH

## :TRIGger:DURation:QUALifier

### ■ 命令格式:

:TRIGger:DURation:QUALifier { GREaterthan | LESSthan | INRange }

:TRIGger:DURation:QUALifier?

### ■ 功能描述:

用于设置延迟触发时间间隔条件, GREaterthan (大于)、LESSthan (小于)、INRange (范围内)。

### ■ 返回格式:

查询返回{ GREaterthan | LESSthan | INRange }。

### ■ 举例:

:TRIGger:DURation:QUALifier GRE                    设置斜率条件为大于  
:TRIGger:DURation:QUALifier?                    查询返回 GREaterthan

### :TRIGger:DURation:TIME:LOWer

#### ■ 命令格式:

:TRIGger:DURation:TIME:LOWer <time>  
:TRIGger:DURation:TIME:LOWer?

#### ■ 功能描述:

用于设置持续时间触发的时间下限，时间间隔条件（大于）时，用于设置时间下限。

#### ■ 返回格式:

查询返回当前时间下限，单位 s。

#### ■ 举例:

:TRIGger:DURation:TIME:LOWer 1                    设置持续时间触发时间下限 1s  
:TRIGger:DURation:TIME:LOWer?                    查询返回 1.000e000

### :TRIGger:DURation:TIME:UPPer

#### ■ 命令格式:

:TRIGger:DURation:TIME:UPPer <time>  
:TRIGger:DURation:TIME:UPPer?

#### ■ 功能描述:

用于设置持续时间触发的时间上限，时间间隔条件（小于）时，用于设置时间上限。

#### ■ 返回格式:

查询返回当前时间上限，单位 s。

#### ■ 举例:

:TRIGger:DURation:TIME:UPPer 1                    设置持续时间触发时间上限 1s  
:TRIGger:DURation:TIME:UPPer?                    查询返回 1.000e000

## 建立保持触发

### :TRIGger:SHOLd:DATA:SOURce

#### ■ 命令格式:

:TRIGger:SHOLd:DATA:SOURce {CHANnel1 | CHANnel2| CHANnel3| CHANnel4}  
:TRIGger:SHOLd:DATA:SOURce?

#### ■ 功能描述:

用于设置建立保持触发数据源。

■ **返回格式:**

查询返回{CHANnel1 | CHANnel2| CHANnel3| CHANnel4}。

■ **举例:**

:TRIGger:SHOLd:DATA:SOUR CHAN1            设置通道一为数据源  
:TRIGger:SHOLd:DATA:SOUR?                查询返回 CHANnel1

**:TRIGger:SHOLd:CLOCK:SOURce**

■ **命令格式:**

:TRIGger:SHOLd:CLOCK:SOURce {CHANnel1 | CHANnel2| CHANnel3| CHANnel4}  
:TRIGger:SHOLd:CLOCK:SOURce?

■ **功能描述:**

用于设置建立保持触发时钟源。

■ **返回格式:**

查询返回{CHANnel1 | CHANnel2| CHANnel3| CHANnel4}。

■ **举例:**

:TRIGger:SHOLd:CLOCK:SOUR CHAN1            设置通道一为时钟源  
:TRIGger:SHOLd:CLOCK:SOUR?                查询返回 CHANnel1

**:TRIGger:SHOLd:SLOPe**

■ **命令格式:**

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

■ **功能描述:**

用于设置建立保持触发边沿类型，POSitive（上升沿）、NEGative（下降沿）。

■ **返回格式:**

查询返回{POSitive|NEGative}。

■ **举例:**

:TRIGger:SHOLd:SLOPe POS                    建立保持触发为上升沿  
:TRIGger:SHOLd:SLOPe?                    查询返回 POSitive

**:TRIGger:SHOLd:PATTern**

■ **命令格式:**

:TRIGger:SHOLd:PATTern { HIGH | LOW }  
:TRIGger:SHOLd:PATTern?

■ **功能描述:**

用于设置建立保持触发码型，HIGH（码型值为 1）、LOW（码型值为 0）。

■ **返回格式：**

查询返回{ HIGH | LOW }。

■ **举例：**

:TRIGger:SHOLd:PATTErn HIGH            设置建立保持触发码型为 1  
:TRIGger:SHOLd:PATTErn?                查询返回 HIGH

**:TRIGger:SHOLd:MODE**

■ **命令格式：**

:TRIGger:SHOLd:MODE     { SETup | HOLD | SH }  
:TRIGger:SHOLd:MODE?

■ **功能描述：**

用于设置触发时间模式。SETup（建立时间）、HOLD（保持时间）、SH（建立保持时间）

■ **返回格式：**

查询返回{ SETup | HOLD | SH }。

■ **举例：**

:TRIGger:SHOLd:MODE HOLD            设置触发时间为保持时间模式  
:TRIGger:SHOLd:MODE?                查询返回 HOLD

**:TRIGger:SHOLd:TIME**

■ **命令格式：**

:TRIGger:SHOLd:TIME     <time>  
:TRIGger:SHOLd:TIME?

■ **功能描述：**

用于设置建立保持触发时间间隔。

■ **返回格式：**

查询返回当前时间间隔，单位 s。

■ **举例：**

:TRIGger:SHOLd:TIME 1                设置建立保持触发模式时间间隔 1s  
:TRIGger:SHOLd:TIME?                查询返回 1.000e000

**:TRIGger:SHOLd:SELEct**

■ **命令格式：**

:TRIGger:SHOLd:SELEct     <SOURce<n>>  
:TRIGger:SHOLd:SELEct

■ **功能描述:**

用于切换选中的源。SOURCE<n>代表源, n 取值 1、2。

SOURCE1 表示数据源; SOURCE2 表示时钟源。

■ **返回格式:**

查询返回{ SOURCE1 | SOURCE2 }。

■ **举例:**

:TRIGGER:SHOLD:SELECT SOURCE1                    设置选中焦点源

:TRIGGER:SHOLD:SELECT?                                查询返回 SOURCE1

## N 边沿触发

### :TRIGGER:NEDGE:SLOPE

■ **命令格式:**

:TRIGGER:NEDGE:SLOPE {POSITIVE|NEGATIVE}

:TRIGGER:NEDGE:SLOPE?

■ **功能描述:**

用于设置触发的边沿类型, POSITIVE (上升沿)、NEGATIVE (下降沿)。

■ **返回格式:**

查询返回触发的边沿类型{POSITIVE|NEGATIVE}。

■ **举例:**

:TRIGGER:NEDGE:SLOPE POS                            设置边沿触发为上升沿

:TRIGGER:NEDGE:SLOPE?                                查询返回 POSITIVE

### :TRIGGER:NEDGE:TIME

■ **命令格式:**

:TRIGGER:NEDGE:TIME <time>

:TRIGGER:NEDGE:TIME?

■ **功能描述:**

用于设置 N 边沿触发时间间隔。

■ **返回格式:**

查询返回当前时间间隔, 单位 s。

■ **举例:**

:TRIGGER:NEDGE:TIME 1                                设置 N 边沿触发模式时间间隔 1s

:TRIGGER:NEDGE:TIME?                                查询返回 1.000e000

### :TRIGGER:NEDGE:VALUE

■ **命令格式:**

:TRIGGER:NEDGE:VALUE <value>

:TRIGGER:NEDGE:VALUE?



- **功能描述:**  
用于设置 N 边沿值, <value>整型值, 范围 0~65535。
- **返回格式:**  
查询返回当前 N 边沿值。
- **举例:**  
:TRIGger:NEDGE:VALue 100                    设置 N 边沿值 100  
:TRIGger:NEDGE:VALue?                        查询返回 100

## 码型触发

### :TRIGger:PATtern:PATtern

- **命令格式:**  
:TRIGger:PATtern:PATtern { HIGH | LOW | X | POSitive | NEGative }  
:TRIGger:PATtern:PATtern?
- **功能描述:**  
用于设置码型触发码型, HIGH (码型值为 1)、LOW (码型值为 0)、X (通道无效)、POSitive (上升)、NEGative (下降)。
- **返回格式:**  
查询返回{ HIGH | LOW | X | POSitive | NEGative }。
- **举例:**  
:TRIGger:PATtern:PATtern HIGH                设置码型触发码型为 1  
:TRIGger:PATtern:PATtern?                    查询返回 HIGH

## CURSor 命令

用于设置光标参数, 对屏幕波形数据进行测量。

### :CURSor:MODE

- **命令格式:**  
:CURSor:MODE { TRACK | INDEpendent }  
:CURSor:MODE?
- **功能描述:**  
用于设置光标模式的光标模式。  
TRACK (跟踪)、INDEpendent (独立)。
- **返回格式:**  
查询返回{ TRACK | INDEpendent }。
- **举例:**  
:CURSor:MODE TRACK                        设置光标为跟踪模式

:CURSor:MODE?                      查询返回 TRACK

## :CURSor:TYPE

### ■ 命令格式:

:CURSor:TYPE {AMPlitude | TIME | SCReen | CLOSe }

:CURSor:TYPE?

### ■ 功能描述:

用于设置光标模式的光标类型。

AMPlitude（幅度）、TIME（时间）、CLOSe（关闭）、SCReen（屏幕）。

### ■ 返回格式:

查询返回{AMPlitude | TIME | SCReen| CLOSe }。

### ■ 举例:

:CURSor:TYPE AMP                      设置光标类型为幅度

:CURSor:TYPE?                          查询返回 AMPlitude

## :CURSor:SOURce

### ■ 命令格式:

:CURSor:SOURce <source>

:CURSor:SOURce?

### ■ 功能描述:

用于设置使用手动光标模式的光标源。

<source>取值{CHANnel<n>|MATH| REFA | REFB | REFC | REFD }, n 取值 1、2、3、4。

### ■ 返回格式:

查询返回{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4| MATH | REFA | REFB | REFC | REFD }。

### ■ 举例:

:CURSor:SOURce CHAN1                      设置通道一为光标源

:CURSor:SOURce?                          查询返回 CHANnel1

## :CURSor:CURAX

### ■ 命令格式:

:CURSor:CURAX <value>

:CURSor:CURAX?

### ■ 功能描述:

用于设置光标线 A 的横向位置。

时基 YT 模式范围从左到右[0,699]，时基 XY 模式从左到右[28,227]。

### ■ 返回格式:

查询返回光标线 A 横向位置。

■ **举例:**

:CURSor:CURAX 50

设置手动光标线 A 横向位置为 50

:CURSor:CURAX?

查询返回 50

**:CURSor:CURAY**

■ **命令格式:**

:CURSor:CURAY <value>

:CURSor:CURAY?

■ **功能描述:**

用于设置光标线 A 纵向位置，范围从下到上[28,227]。

■ **返回格式:**

查询返回光标线 A 纵向位置。

■ **举例:**

:CURSor:CURAY 50

设置手动光标线 A 纵向位置为 50

:CURSor:CURAY?

查询返回 50

**:CURSor:CURBX**

■ **命令格式:**

:CURSor:CURBX <value>

:CURSor:CURBX?

■ **功能描述:**

用于设置光标线 B 的横向位置。

时基 YT 模式范围从左到右[0,699]，时基 XY 模式从左到右[28,227]。

■ **返回格式:**

查询返回光标线 B 横向位置。

■ **举例:**

:CURSor:CURBX 50

设置手动光标线 B 横向位置为 50

:CURSor:CURBX?

查询返回 50

**:CURSor:CURBY**

■ **命令格式:**

:CURSor:CURBY <value>

:CURSor:CURBY?

■ **功能描述:**

用于设置光标线 B 纵向位置，范围从下到上[28,227]。

■ **返回格式:**

查询返回光标线 B 纵向位置。

■ **举例:**





## FILE 命令

用于参考波形和存储功能相关设置。

### :FILE:LOAD

#### ■ 命令格式:

:FILE:LOAD <filename>[,<source>][,<disk>]

#### ■ 功能描述:

用于加载波形到相关参考通道中或者设置数据。

**<filename>**表示文件名称，文件名称必须是字符串类型数据，需带双引号，例如"test.bsv"

- 文件名为\*.dat 表示某个文件的波形数据加载到参考通道里，以示波器自身后缀名称相匹配。
- 文件名为\*.set 表示某个文件的设置数据加载到示波器中，以示波器自身后缀名称相匹配。
- 文件名为\*.bode.csv 表示某个文件的波特图配置数据加载到示波器中，以示波器自身后缀名称相匹配，目前只对 U 盘生效。

**<source >**表示参考通道{REFA | REFB | REFC | REFD}，可选参数，加载波形数据时才有效。

- REFA 表示参考通道 A
- REFB 表示参考通道 B
- REFC 表示参考通道 C
- REFD 表示参考通道 D

**<disk>**表示存储介质{ FLASH | UDISK }，可选参数，忽略表示 FLASH 内部数据。

- FLASH 表示内部数据
- UDISK 表示 U 盘数据

#### ■ 举例:

FILE:LOAD "test.dat",REFA,UDISK	从 U 盘加载 test.dat 波形数据到参考通道 A 中
FILE:LOAD "system-set-up01.set"	从内部介质加载 1 位置配置数据到示波器中
FILE:LOAD "001.bode.csv"	从 U 盘加载波特图数据到示波器中

注意：在示波器无法自定义文件名的时

- 存储内部设置文件名必须是"system-set-up01.set"~ "system-set-up255.set"，最大 255 个文件。
- 存储内部文 bsv 文件文件名必须是 "wave01.bsv"~ " wave255.bsv"，最大 255 个文件。

### :FILE:SAVE

#### ■ 命令格式:

:FILE:SAVE <filename>[,<source>][,<disk>]

#### ■ 功能描述:

用于保存通道波形或者设置数据到文件中。

**<filename>**表示文件名称，文件名称必须是字符串类型数据，需带双引号，例如"test.bsv"

- 文件名为\*.dat 表示以后缀名格式保存某个通道的波形到该文件中，以示波器自身后缀名称相匹配。

- 文件名为\*.set 表示某个设置数据到该文件中，以示波器自身后缀名称相匹配。
- 文件名为\*.bode.csv 表示以后缀名格式保存波特图配置数据到该文件中，以示波器自身后缀名称相匹配，目前只对 U 盘生效。

**<source>表示物理通道{CHANnel1 | CHANnel2| CHANnel3| CHANnel4}，可选参数，保存波形数据时才有效。**

- CHANnel1 表示通道 1
- CHANnel2 表示通道 2
- CHANnel3 表示通道 3
- CHANnel4 表示通道 4

**<disk>表示存储介质{ FLASH | UDISK }，可选参数，忽略表示 FLASH 内部数据。**

- FLASH 表示内部数据
- UDISK 表示 U 盘数据

#### ■ 举例：

FILE:SAVE "test.dat",CHANnel1,UDISK	通道 1 波形数据保存成 U 盘的 test.dat 文件
FILE:SAVE "system-set-up01.set"	示波器配置数据保存成内部介质 1 号位置
FILE:SAVE "wave01.dat",CHANnel1,FLASH	通道 1 波形数据保存到内部介质
FILE:SAVE "wave01.dat",CHANnel1	通道 1 波形数据保存到内部介质
FILE:SAVE "system-set-up01.set",FLASH	示波器配置数据保存成内部介质
FILE:SAVE "system-set-up01.set"	示波器配置数据保存成内部介质 1 号位置
FILE:SAVE "001.bode.csv",UDISK	波特图数据保存到 U 盘的 001.bode.csv 中

注意：在示波器无法自定义文件名的时

- 存储内部设置文件名必须是"system-set-up01.set"~ "system-set-up255.set"，最大 255 个文件。
- 存储内部文 bsv 文件文件名必须是 "wave01.bsv"~ "wave255.bsv"，最大 255 个文件。

## RECORD 命令

用于示波器波形录制功能相关设置。

### :RECORD:ENABLE

#### ■ 命令格式：

```
:RECORD:ENABLE { {1|ON} | {0|OFF} }
:RECORD:ENABLE?
```

#### ■ 功能描述：

用于设置波形录制功能 ON（打开）或 OFF（关闭）。

#### ■ 返回格式：

查询返回 1 或 0，分别代表 ON 或 OFF。

#### ■ 举例：

```
:RECORD:ENABLE ON          打开波形录制功能
```

:RECOrd:ENABle?

查询返回 1，表示已经打开波形录制功能

### :RECOrd:STARt

■ **命令格式:**

:RECOrd:STARt { {1|ON} | {0|OFF} }

:RECOrd:STARt?

■ **功能描述:**

用于设置波形录制 ON（开始）或 OFF（停止）。

■ **返回格式:**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例:**

:RECOrd:STARt ON

开始波形录制

:RECOrd:STARt?

查询返回 1，表示已启动波形录制

### :RECOrd:FAST

■ **命令格式:**

:RECOrd:FAST { {1|ON} | {0|OFF} }

:RECOrd:FAST?

■ **功能描述:**

用于设置快速录制 ON（打开）或 OFF（关闭）。

■ **返回格式:**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例:**

:RECOrd:FAST ON

打开快速录制

:RECOrd:FAST?

查询返回 1，表示已打开快速录制

### :RECOrd:INTerval

■ **命令格式:**

:RECOrd:INTerval <time>

:RECOrd:INTerval?

■ **功能描述:**

用于设置波形录制的时间间隔。

■ **返回格式:**

查询返回录制波形的时间间隔，采用科学计数法，单位 s。

■ **举例:**

:RECOrd:INTerval 200ns

设置录制波形播放延时时间为 200ns



:RECORD:INTERVAL?

查询返回 2.000e-004

### :RECORD:PLAY

■ **命令格式:**

:RECORD:PLAY { {1|ON} | {0|OFF} }

:RECORD:PLAY?

■ **功能描述:**

用于设置录制波形播放 ON（开始）或 OFF（停止）。

■ **返回格式:**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例:**

:RECORD:PLAY ON

开始录制波形播放

:RECORD:PLAY?

查询返回 1，表示已启动录制波形播放

### :RECORD:PLAY:DELAY

■ **命令格式:**

:RECORD:PLAY:DELAY <time>

:RECORD:PLAY:DELAY?

■ **功能描述:**

用于设置录制波形播放延时时间。

■ **返回格式:**

查询返回录制波形播放延时时间，采用科学计数法，单位 s。

■ **举例:**

:RECORD:PLAY:DELAY 20ms

设置录制波形播放延时时间为 20ms

:RECORD:PLAY:DELAY?

查询返回 2.000e-002

### :RECORD:CURRENT

■ **命令格式:**

:RECORD:CURRENT <value>

:RECORD:CURRENT?

■ **功能描述:**

用于设置或查询录制波形播放的当前帧。

■ **返回格式:**

查询返回录制波形播放的当前帧，整型数据。

■ **举例:**

:RECORD:CURRENT 100

设置波形播放的当前帧为 100

:RECOrd:CURRent?

查询返回 100

### :RECOrd:FRAMes

#### ■ 命令格式:

:RECOrd:FRAMes <value>

:RECOrd:FRAMes?

#### ■ 功能描述:

用于设置或查询波形录制帧数。

#### ■ 返回格式:

查询返回波形录制帧数，整型数据。

#### ■ 举例:

:RECOrd:FRAMes 400

设置波形录制帧数为 400

:RECOrd:FRAMes?

查询返回 400

## DVM 命令

用于示波器数字电压表功能相关设置。

### :DVM:ENABle

#### ■ 命令格式:

:DVM:ENABle { {1|ON} | {0|OFF} }

:DVM:ENABle?

#### ■ 功能描述:

用于设置或查询数字电压表功能 ON（打开）或 OFF（关闭）。

#### ■ 返回格式:

查询返回 1 或 0，分别代表 ON 或 OFF。

#### ■ 举例:

:DVM:ENABle ON

打开数字电压表

:DVM:ENABle?

查询返回 1

### :DVM:SOURCe

#### ■ 命令格式:

:DVM:SOURCe <source>

:DVM:SOURCe?

#### ■ 功能描述:

用于设置或查询数字电压表信源。

<source>: CHANnel<n>, 其中 n 取值 1、2。

#### ■ 返回格式:

查询返回{CHANnel1 | CHANnel2 }。

- **举例:**  
:DVM:SOURCe CHANnel                    设置信源为通道一  
:DVM:SOURCe?                            查询返回 CHANnel

### :DVM:MODE

- **命令格式:**  
:DVM:MODE {ACRMs|DC|DCRMs}  
:DVM:MODE?
- **功能描述:**  
用于设置或查询数字电压表模式。
- **返回格式:**  
查询返回{ACRMs|DC|DCRMs}。
- **举例:**  
:DVM:MODE DC                            设置数字电压表模式为 DC  
:DVM:MODE?                              查询返回 DC

### :DVM:CURRent?

- **命令格式:**  
:DVM:CURRent?
- **功能描述:**  
用于查询数字电压表当前所测的电压值。
- **返回格式:**  
查询返回数字电压表当前所测的电压值，标准单位，采用科学计数法，单位由:DVM:MODE 决定。
- **举例:**  
:DVM:CURRent?                            当前所测的电压值为 3.000e-003

## PF 命令

用于示波器通过/失败测试功能相关设置。

### :PF:ENABle

- **命令格式:**  
:PF:ENABle { {1|ON} | {0|OFF} }  
:PF:ENABle?
- **功能描述:**  
用于设置或查询通过/失败测试功能 ON（打开）或 OFF（关闭）。
- **返回格式:**  
查询返回 1 或 0，分别代表 ON 或 OFF。

- **举例：**  
 :PF:ENABLE ON                                    打开通过/失败测试功能  
 :PF:ENABLE?                                        查询返回 1

### **:PF:SOURCe**

- **命令格式：**  
 :PF:SOURCe <source>  
 :PF:SOURCe?
- **功能描述：**  
 用于设置或查询通过/失败测试的测量源。  
 <source>: CHANnel<n>, 其中 n 取值 1、2、3、4。
- **返回格式：**  
 查询返回{CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。
- **举例：**  
 :PF:SOURCe CHANnel1                            设置测量源为通道一  
 :PF:SOURCe?                                        查询返回 CHANnel1

### **:PF:OPERate**

- **命令格式：**  
 :PF:OPERate {RUN|STOP}  
 :PF:OPERate?
- **功能描述：**  
 用于设置运行或停止通过/失败测试。
- **返回格式：**  
 查询返回{RUN|STOP}。
- **举例：**  
 :PF:OPERate RUN                                    运行通过/失败测试  
 :PF:OPERate?                                        查询返回 RUN

### **:PF:OUTPut**

- **命令格式：**  
 :PF:OUTPut {PASS|FAILED}  
 :PF:OUTPut?
- **功能描述：**  
 用于设置或查询通过/失败测试的输出。
- **返回格式：**  
 查询返回{PASS|FAILED}。
- **举例：**



## :PF:TEMPLate:SOURce

### ■ 命令格式:

:PF:TEMPLate:SOURce {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | REF | USB}  
:PF:TEMPLate:SOURce?

### ■ 功能描述:

用于设置或查询通过/失败测试模板源。

可使用{CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}物理通道作为模板源。

如果模板源设置成 REF，可使用:PF:TEMPLate:LOAD 指令从 REF 加载波形文件作为模板源；如果模板源设置成 USB，可使用:PF:TEMPLate:LOAD 指令从 U 盘加载波形文件作为模板源。

### ■ 返回格式:

查询返回{CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | REF | USB}。

### ■ 举例:

:PF:TEMPLate:SOURce CHANnel1	设置模板源为通道一
:PF:TEMPLate:SOURce?	查询返回 CHANnel1

## :PF:TEMPLate:LOAD

### ■ 命令格式:

:PF:TEMPLate:LOAD <filename>

### ■ 功能描述:

用于模板源为 REF 或 USB 时，加载指定波形文件作为模板源。

### ■ 举例:

PF:TEMPLate:LOAD "test.dat"	加载 test.dat 波形文件作为模板源
-----------------------------	-----------------------

## :PF:TEMPLate:X

### ■ 命令格式:

:PF:TEMPLate:X <value>  
:PF:TEMPLate:X?

### ■ 功能描述:

用于设置或查询通过/失败测试模板设置的水平容限。

<value>: 水平容限，其范围 1~100，具体范围根据示波器设备自适应。

### ■ 返回格式:

查询返回模板设置的水平容限，整型数据。

### ■ 举例:

:PF:TEMPLate:X 50	设置模板设置的水平容限为 50
:PF:TEMPLate:X?	查询返回 50

## :PF:TEMPLate:Y

### ■ 命令格式:

:PF:TEMPLate:Y <value>  
:PF:TEMPLate:Y?

■ **功能描述:**

用于设置或查询通过/失败测试模板设置的垂直容限。

<value>: 垂直容限, 其范围 1~100, 具体范围根据示波器设备自适应。

■ **返回格式:**

查询返回模板设置的垂直容限, 整型数据。

■ **举例:**

:PF:TEMPlate:Y 50

设置模板设置的垂直容限为 50

:PF:TEMPlate:Y?

查询返回 50

**:PF:RESult?**

■ **命令格式:**

:PF:RESult?

■ **功能描述:**

用于查询通过/失败测试统计结果。

返回数据格式: <pass>,<failed>,<total>, 其中<pass>表示通过次数, <failed>表示失败的次数, <total>表示总的次数。

■ **返回格式:**

查询返回通过/失败测试统计结果。

■ **举例:**

:PF:RESult?

查询返回 35,42,77

## ACQUIRE 命令

用于设置示波器的采样方式。

**:ACQUIRE:TYPE**

■ **命令格式:**

:ACQUIRE:TYPE {NORMAL | AVERAGE | PEAKdetect | ENVELOPE | HRESolution }

:ACQUIRE:TYPE?

■ **功能描述:**

用于设置示波器的采样获取方式。

NORMAL (正常)、AVERAGE (平均)、PEAKdetect (峰值)、ENVELOPE (包络)、HRESolution (高分辨率)。

■ **返回格式:**

查询返回{NORMAL | AVERAGE | PEAKdetect | ENVELOPE | HRESolution }。

■ **举例:**

:ACQ:TYPE AVER

设置获取方式为平均。

:ACQ:TYPE?

查询返回 AVERAGE。

**:ACQUIRE:AVERages:COUNT**

■ **命令格式:**

:ACquire:AVERages:COUNT <count>

:ACquire:AVERages:COUNT?

■ **功能描述:**

用于设置示波器在平均采样时的平均采样次数。其中<count>以 2 的 N 次幂步进, 在 2 到 8192 间取值,  $1 \leq N \leq 30$ 。

■ **返回格式:**

查询返回当前平均采样次数。

■ **举例:**

:ACQ:AVER:COUN 32

设置平均采样平均次数为 32。

:ACQ:AVER:COUN?

查询返回 32。

## :ACquire:MEMory:DEPTH

■ **命令格式:**

:ACquire:MEMory:DEPTH { AUTO | 700 | 7K | 70K | 700K | 7M | 14M | 28M | 56M }

:ACquire:MEMory:DEPTH?

■ **功能描述:**

用于设置采样存储深度模式, 根据不同机型存储深度自适应设置。

■ **返回格式:**

查询返回{ AUTO | 700 | 7K | 70K | 700K | 7M | 14M | 28M | 56M }。

■ **举例:**

:ACQ:MEM:DEPT AUTO

设置存储深度为自动模式。

:ACQ:MEM:DEPT?

查询返回 AUTO

## DISPlay 命令

用于设置或查询示波器的显示功能或数据。

### :DISPlay:DATA?

■ **命令格式:**

:DISPlay:DATA?

■ **功能描述:**

用于查询示波器当前屏幕的图像数据。

■ **返回格式:**

查询返回 BMP 格式图像数据, 返回的数据符合[附录 2: IEEE 488.2 二进制数据格式](#)。

■ **举例:**

:DISPlay:DATA?

查询返回图像数据

### :DISPlay:FORMat

■ **命令格式:**



:DISPlay:FORMat { VECTors | DOTS }

:DISPlay:FORMat?

■ **功能描述:**

用于设置采样点的显示格式。VECTors（矢量显示）、DOTS（直接显示采样点）

■ **返回格式:**

查询返回{ VECTors | DOTS }。

■ **举例:**

:DISPlay:FORMat VECT                    设置采样点矢量显示

:DISPlay:FORMat                         查询返回 VECTors

**:DISPlay:GRID**

■ **命令格式:**

:DISPlay:GRID {FULL|HALF|CROSS|NONE}

:DISPlay:GRID?

■ **功能描述:**

用于设置显示屏上网格显示模式。

■ **返回格式:**

查询返回{FULL|HALF|CROSS|NONE}。

■ **举例:**

:DISPlay:GRID CROSS                    设置显示十字坐标线，不显示分格

:DISPlay:GRID?                         查询返回 CROSS

**:DISPlay:GRID:BRIGhtness**

■ **命令格式:**

:DISPlay:GRID:BRIGhtness     <count>

:DISPlay:GRID:BRIGhtness?

■ **功能描述:**

用于设置网格亮度，<count>取值为 1~100，数字越大网格越亮。

■ **返回格式:**

查询返回当前网格亮度。

■ **举例:**

:DISPlay:GRID:BRIGhtness 50         设置网格亮度 50

:DISPlay:GRID:BRIGhtness?         查询返回 50

**:DISPlay:GRAD:TIME**

■ **命令格式:**

:DISPlay:GRAD:TIME            {MINimum|50ms|100ms|200ms|500ms|1s|2s|5s|10s|20s|INFinite}

:DISPlay:GRAD:TIME?

■ **功能描述:**

用于设置余辉时间。

■ **返回格式:**

查询返回{MINimum|50ms|100ms|200ms|500ms|1s|2s|5s|10s|20s|INFinite}。

■ **举例:**

:DISPlay:GRAD:TIME 50ms                    设置余辉时间 50ms

:DISPlay:GRAD:TIME?                        查询返回 50ms

**:DISPlay:COLOR**

■ **命令格式:**

:DISPlay:COLOR { {1|ON} | {0|OFF} }

:DISPlay:COLOR?

■ **功能描述:**

用于设置显示色温 ON（打开）或 OFF（关闭）。

■ **返回格式:**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例:**

:DISPlay:COLOR ON                        显示颜色开。

:DISPlay:COLOR?                        查询返回 1，表示已经打开颜色显示。

**:DISPlay:COLOR:INVERT**

■ **命令格式:**

:DISPlay:COLOR:INVERT { {1|ON} | {0|OFF} }

:DISPlay:COLOR:INVERT?

■ **功能描述:**

用于设置反转色温 ON（打开）或 OFF（关闭）。

■ **返回格式:**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例:**

:DISPlay:COLOR:INVERT ON                反转颜色开。

:DISPlay:COLOR:INVERT?                查询返回 1，表示已经打开反转颜色显示。

**:DISPlay:WAVE:BRIGhtness**

■ **命令格式:**

:DISPlay:WAVE:BRIGhtness    <count>

:DISPlay:WAVE:BRIGhtness?

■ **功能描述:**

用于设置波形亮度，<count>取值为 1~100，数字越大波形越亮。

■ **返回格式:**

查询返回当前波形亮度。

■ **举例:**

:DISPlay:WAVE:BRIGhtness 50            设置波形亮度 50

:DISPlay:WAVE:BRIGhtness?      查询返回 50

### :DISPlay:CLEAr

■ **命令格式:**

:DISPlay:CLEAr

■ **功能描述:**

用于清除并刷新示波器屏幕上的波形，如果有参考波形，则清除参考波形并刷新。

### :DISPlay:TYPE

■ **命令格式:**

:DISPlay:TYPE {XY12|XY13|XY14| XY23|XY24|XY34|YT}

:DISPlay:TYPE?

■ **功能描述:**

用于设置时基显示类型为 XY12（X-Y 方式：在水平轴上显示通道 1 幅值，垂直轴上显示通道 2 幅值，XY13|XY14| XY23|XY24|XY34 类型同上）；YT（Y-T 方式：显示垂直电压与水平时间的相对关系）。

■ **返回格式:**

查询返回{XY12|XY13|XY14| XY23|XY24|XY34|YT}。

■ **举例:**

:DISP:TYPE YT

设置时基格式为 YT 方式。

:DISP:TYPE?

查询返回 YT。

## WAVeform 命令

用于读取示波器屏幕中的波形数据及相关参数。

### :WAVeform:MODE

■ **命令格式:**

:WAVeform:MODE {NORMal | RAW}

:WAVeform:MODE?

■ **功能描述:**

NORMal: 读取当前屏幕显示的波形数据，此波形数据点数为固定点数。

RAW: 读取内存中的波形数据，此波形数据点数和存储深度相关。注意：此指令复位起始点和截止点及波形点数，内存中的数据必须在示波器停止状态下才能进行读取，MATH 通道下该指令无效。

■ **返回格式:**

查询返回{NORMal | RAW}。

■ **举例:**

:WAVeform:MODE RAW

设置波形数据的读取模式为内存读取

:WAVeform:MODE?

查询返回 RAW

## :WAVeform:FORMat

### ■ 命令格式:

:WAVeform:FORMat { WORD | BYTE | ASCII }

:WAVeform:FORMat?

### ■ 功能描述:

示波器默认波形数据格式 AD 波形点数据

BYTE: 返回 AD 数据, 一个波形点占一个字节 (即 8 位)。

WORD: 返回 AD 数据, 一个波形点占两个字节 (即 16 位), 低 8 位有效, 高 8 位为 0。

ASCII: 返回波形以科学计数形式返回各波形点的实际电压值, 各电压值之间以逗号分隔且符合附录 2: IEEE 488.2 二进制数据格式。

例如:#412342.00000E+01,2.20000E+01, 2.30000E+01.....\n。

### ■ 返回格式:

查询返回{ WORD | BYTE | ASCII }。

### ■ 举例:

:WAVeform:FORMat BYTE                    波形 AD 数据的返回格式为单字节模式

:WAVeform:FORMat?                        查询返回 BYT

## :WAVeform:STARt

### ■ 命令格式:

:WAVeform:STARt <start>

:WAVeform:STARt?

### ■ 功能描述:

设置或查询波形数据读取的起始位置, <start>整型数据类型。

NORMal: 1 到 1400

RAW: 1 至当前最大的存储深度点数

### ■ 返回格式:

查询返回起始位置。

### ■ 举例:

:WAVeform:STARt 200                    设置波形数据读取的起始点为 200

:WAVeform:STARt?                        查询返回 200

## :WAVeform:STOP

### ■ 命令格式:

:WAVeform:STOP <stop>

:WAVeform:STOP?

### ■ 功能描述:

设置或查询波形数据读取的截止位置，< stop>整型数据类型。

NORMal: < stop>范围 1 到 1400

RAW: < stop>范围 1 至当前最大的存储深度点数

■ **返回格式:**

查询返回截止位置。

■ **举例:**

:WAVeform:STOP 400                      设置波形数据读取的结束点为 400

:WAVeform:STOP?                      查询返回 400

**:WAVeform:SOURce**

■ **命令格式:**

:WAVeform:SOURce {CHANnel<n>|MATH |<Dx>}

:WAVeform:SOURce?

■ **功能描述:**

用于设置当前要查询波形数据的信号源，如果不发送该指令，表示要查询当前通道的波形数据。

<Dx>: {D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。

通道源设为 MATH 通道时，只能读取屏幕显示的波形数据。

通道源设为数字通道时，查询命令:WAVeform:DATA?固定以 BYTE 格式返回波形数据。若读取的是屏幕波形数据，则返回当前所选通道源的信号状态，一个波形点占 2 个字节（16 位）。若读取的是内存波形数据，则返回当前所选通道源所在通道组的信号状态，2 个字节（16 位）表示一组数字通道信号的状态。

■ **返回格式:**

■ 查询返回 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH |<Dx>}。

■ **举例:**

:WAVeform:SOURce CHAN1              设置当前要查询波形数据的信号源为通道一

:WAVeform:SOURce?                      查询返回 CHANnel1

**:WAVeform:POINts**

■ **命令格式:**

:WAVeform:POINts      <points>

:WAVeform:POINts?

■ **功能描述:**

用于设置需要返回的波形点数，默认值为 0。

■ **返回格式:**

查询返回需要返回的波形点数。

■ **举例:**

:WAVeform:POINts 120                      设置需要返回的 120 个波形点数

:WAVeform:POINts?                      查询返回 120

## :WAVeform:DATA?

### ■ 命令格式:

:WAVeform:DATA?

### ■ 功能描述:

用于读取指定数据源中的波形数据。

### ■ 返回格式:

WAVeform:POINts 指定数量的波形数据，波形数据源与:WAVeform:SOURce 相关，数据格式与 WAVeform:FORMat 相关，返回的数据符合附录 2: IEEE 488.2 二进制数据格式。

注意: 通道源设为数字通道时，查询固定以 WORD 格式返回波形数据。若读取的是屏幕波形数据，则返回当前所选通道源的信号状态，一个波形点占 2 个字节（16 位）。若读取的是内存波形数据，则返回当前所选通道源所在通道组的信号状态，2 个字节（16 位）表示一组数字通道信号的状态。

### ■ 举例:

获得指定数据源的波形数据指令顺序执行如下:

#### ◆ 获得屏幕波形数据流程

:WAVeform:SOURce CHAN1	设置当前要查询波形数据的信号源为通道一
:WAVeform:MODE NORMal	设置读取屏幕显示波形数据
:WAVeform:FORMat BYTE	波形数据的返回格式为 AD 单字节模式
:WAVeform:DATA?	获得波形数据

#### ◆ 获得内存波形数据流程，此流程只有在停止状态下有效

:WAVeform:SOURce CHAN1	设置当前要查询波形数据的信号源为通道一
:WAVeform:MODE RAW	设置读取内存波形数据
:WAVeform:FORMat BYTE	波形数据的返回格式为 AD 单字节模式
:WAVeform:POINts 5000	读取内存波形点数为 5000
循环读取内存波形数据	
{	
:WAVeform:DATA?	获得内存中一块的波形数据
:WAVeform:START?	波形数据读取的起始位置，-1 表示已到最后一个点
}	

说明: 分批次读取内存数据时，每次读回的数据只是内存中一块区域的数据，相邻两块间的波形数据连续，每块数据符合附录 2: IEEE 488.2 二进制数据格式

## :WAVeform:PREamble?

### ■ 命令格式:

:WAVeform:PREamble?

### ■ 功能描述:

查询返回当前系统波形设置参数。

■ **返回格式:**

查询返回以逗号“,”隔开。

返回的数据格式: Format,Type,Points,Count,Xinc,Xor,Xref,Yinc,Yor,Yref。

Format: BYTE (0)、WORD (1)、ASCII(2)。

Type: NORMAL(0)、PEAK(1)、AVER(2)、ENVELOPE(3)、HRESOLUTION(4)。

Points: 需要返回的波形数据点数。

Count: 在平均采样下为平均次数, 其它模式下为 1。

Xinc: 波形数据源 X 方向两点之间的时间差。

Xor: 触发点相对时间。

Xref: X 基准。

Yinc: Y 方向单位电压。

Yor: Y 方向相对 YREF 的零点位置。

Yref: Y 方向参考值, 通道零点电平 ADC 值。

■ **举例:**

:WAVEFORM:PREAmble?                    返回 1,0,0,1,8.000e-009,-6.000e-006,0,4.000e-002,0.000e000,100

**:WAVEFORM:XINCReMENT?**

■ **命令格式:**

:WAVEFORM:XINCReMENT?

■ **功能描述:**

用于查询当前选中通道源 X 方向上相邻两点之间的时间间隔。

返回值与当前的数据读取模式相关:

NORMAL 模式下, XINCReMENT=TimeScale/时基挡横向的波形点数(50)。

RAW 模式下, XINCReMENT=1/SampleRate。

■ **返回格式:**

查询返回时基数, 单位 s。

■ **举例:**

:WAV:XINC?                                查询返回 3.000e-003

**:WAVEFORM:XORIGIN?**

■ **命令格式:**

:WAVEFORM:XORIGIN?

■ **功能描述:**

查询当前选中通道源 X 方向上波形数据的起始时间。

返回值与当前的数据读取模式相关, 触发点时间为零, 在触发点之前为负数:









## :SBUS:DATA?

### ■ 命令格式:

:SBUS:DATA?

### ■ 功能描述:

用于读取示波器的解码事件表数据。

### ■ 返回格式:

查询返回解码事件表中的数据，返回的数据符合附录 2: IEEE 488.2 二进制数据格式。

### ■ 举例:

:SBUS:DATA? 查询返回:

#9000000089RS232,

TIME,DATA,CHECK,

-1us,0,0,

-890.5ns,1,0,

-403.4ns,0,0,

9.8ns,1,0,

531.7ns,0,0,

RS232 表示解码类型（还可为 I2C、SPI 或 CAN 等），后面紧跟 CSV 格式的事件表数据，事件表数据的具体格式由不同设备自动适配，数据以逗号分开，会根据解码列表的行自动换行，数据值与设定的进制显示有关。

## :SBUS:VERTical:POSition

### ■ 命令格式:

:SBUS:VERTical:POSition <value>

:SBUS:VERTical:POSition?

### ■ 功能描述:

用于设置示波器的总线垂直位置值。参数类型为整型，步进为 6，范围为[-160,160]，以屏幕中心为零点，上正下负。

### ■ 返回格式:

查询返回垂直位置值。

### ■ 举例:

:SBUS:VERTical:POSition 10

打开总线垂直位置值为 10

:SBUS:VERTical:POSition?

查询返回 10

## :SBUS:TRIGger:SWEep

### ■ 命令格式:

:SBUS:TRIGger:SWEep {AUTO|NORMal|SINGle}

:SBUS:TRIGger:SWEep?

■ **功能描述:**

用于选择总线触发扫描模式。

AUTO (自动): 在没有触发条件情况下, 内部将产生触发信号, 强制触发。

NORMal (普通): 只有满足触发条件时才能触发。

SINGle (单次): 在符合触发条件情况下进行一次触发, 然后停止。

■ **返回格式:**

查询返回触发扫描模式{AUTO|NORMal}。

■ **举例:**

:SBUS:TRIGger:SWEep AUTO

设置总线触发扫描模式为自动

:SBUS:TRIGger:SWEep?

查询返回 AUTO

## RS232

### :SBUS:RS232:BAUDrate

■ **命令格式:**

:SBUS:RS232:BAUDrate <baudrate>

:SBUS:RS232:BAUDrate?

■ **功能描述:**

用于设置示波器的 RS232 总线解码波特率。参数为整型, 范围为 120~5000000。

■ **返回格式:**

查询返回波特率。

■ **举例:**

:SBUS:RS232:BAUDrate 500

设置 RS232 波特率为 500b/s

:SBUS:RS232:BAUDrate?

查询返回 500

### :SBUS:RS232:BITorder

■ **命令格式:**

:SBUS:RS232:BITorder {LSBFirst | MSBFirst}

:SBUS:RS232:BITorder?

■ **功能描述:**

用于设置示波器的 RS232 总线解码字节序。LSBFirst 小端模式、MSBFirst 大端模式。

■ **返回格式:**

查询返回{LSBFirst | MSBFirst}。

■ **举例:**

:SBUS:RS232:BITorder LSBF

设置 RS232 字节序为 LSB

:SBUS:RS232:BITorder?

查询返回 LSBFirst

## :SBUS:RS232:SOURce

### ■ 命令格式:

:SBUS:RS232:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}  
:SBUS:RS232:SOURce?

### ■ 功能描述:

用于设置示波器的 RS232 总线解码源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

### ■ 返回格式:

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

### ■ 举例:

:SBUS:RS232:SOURce CHANnel1	设置通道一为 RS232 总线解码源
:SBUS:RS232:SOURce?	查询返回 CHANnel1

## :SBUS:RS232:POLarity

### ■ 命令格式:

:SBUS:RS232:POLarity { POSitive | NEGative }  
:SBUS:RS232:POLarity?

### ■ 功能描述:

用于设置示波器的 RS232 总线解码极性。POSitive (上升)、NEGative (下降)。

### ■ 返回格式:

查询返回{ POSitive | NEGative }。

### ■ 举例:

:SBUS:RS232:POLarity POSitive	设置 RS232 总线解码极性为上升
:SBUS:RS232:POLarity?	查询返回 POSitive

## :SBUS:RS232:PARity

### ■ 命令格式:

:SBUS:RS232:PARity {EVEN | ODD | NONE}  
:SBUS:RS232:PARity?

### ■ 功能描述:

用于设置示波器的 RS232 总线奇偶性。

### ■ 返回格式:

查询返回{EVEN | ODD | NONE}。

### ■ 举例:

:SBUS:RS232:PARity ODD	设置 RS232 总线奇偶性为偶数
:SBUS:RS232:PARity?	查询返回 6

## :SBUS:RS232:DATA:BIT

### ■ 命令格式:

:SBUS:RS232:DATA:BIT {5 | 6 | 7 | 8}

:SBUS:RS232:DATA:BIT?

■ **功能描述:**

用于设置示波器的 RS232 总线数据位。

■ **返回格式:**

查询返回{5 | 6 | 7 | 8}。

■ **举例:**

:SBUS:RS232:DATA:BIT 6	设置 RS232 数据位为 6
:SBUS:RS232:DATA:BIT?	查询返回 6

**:SBUS:RS232:STOP:BIT**

■ **命令格式:**

:SBUS:RS232:STOP:BIT {1 | 2}

:SBUS:RS232:STOP:BIT?

■ **功能描述:**

用于设置示波器的 RS232 总线停止位。

■ **返回格式:**

查询返回{1 | 2}。

■ **举例:**

:SBUS:RS232:STOP:BIT 6	设置 RS232 停止位为 6
:SBUS:RS232:STOP:BIT?	查询返回 6

**:SBUS:RS232:DATA**

■ **命令格式:**

:SBUS:RS232:DATA <value>

:SBUS:RS232:DATA?

■ **功能描述:**

用于设置示波器的 RS232 总线触发数据。参数为 0 或 1 表示的二进制字符串数据，其范围与:SBUS:RS232:DATA:BIT 指令设置的值相关，为[0~2<sup>databit</sup> - 1]。

■ **返回格式:**

查询返回二进制字符串数据。

■ **举例:**

:SBUS:RS232:DATA "01111111"	设置 RS232 总线数据为 0x7F
:SBUS:RS232:DATA?	查询返回 01111111

**:SBUS:RS232:QUALifier**

■ **命令格式:**

:SBUS:RS232:QUALifier {BEGFrame | ERRFrame | ECCErrror | DATA}

:SBUS:RS232:QUALifier?

■ **功能描述:**

用于设置示波器的 RS232 总线触发条件。

■ **返回格式:**

查询返回{BEGFrame|ERRFrame|ECCError|DATA}。

■ **举例:**

:SBUS:RS232:QUALifier ERRF

设置 RS232 总线条件为错误帧

:SBUS:RS232:QUALifier?

查询返回 ERRFrame

## I2C

### :SBUS:I2C:CLOCK:SOURce

■ **命令格式:**

:SBUS:I2C:CLOCK:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}

:SBUS:I2C:CLOCK:SOURce?

■ **功能描述:**

用于设置示波器的 I2C 总线时钟源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

■ **返回格式:**

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

■ **举例:**

:SBUS:I2C:CLOCK:SOURce CHANnel1

设置通道一为 I2C 总线时钟源

:SBUS:I2C:CLOCK:SOURce?

查询返回 CHANnel1

### :SBUS:I2C:DATA:SOURce

■ **命令格式:**

:SBUS:I2C:DATA:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}

:SBUS:I2C:DATA:SOURce?

■ **功能描述:**

用于设置示波器的 I2C 总线数据源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

■ **返回格式:**

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

■ **举例:**

:SBUS:I2C:DATA:SOURce CHANnel1

设置通道一为 I2C 总线数据源

:SBUS:I2C:DATA:SOURce?

查询返回 CHANnel1

## :SBUS:I2C:ASIZe

### ■ 命令格式:

:SBUS:I2C:ASIZe {7 | 10}

:SBUS:I2C:ASIZe?

### ■ 功能描述:

用于设置示波器的 I2C 总线地址位宽。

### ■ 返回格式:

查询返回{7 | 10}。

### ■ 举例:

:SBUS:I2C:ASIZe 7

设置 I2C 总线地址位宽为 7

:SBUS:I2C:ASIZe?

查询返回 7

## :SBUS:I2C:ADIRection

### ■ 命令格式:

:SBUS:I2C:ADIRection { READ | WRITE }

:SBUS:I2C:ADIRection?

### ■ 功能描述:

用于设置示波器的 I2C 总线地址方向。

### ■ 返回格式:

查询返回{ READ | WRITE }。

### ■ 举例:

:SBUS:I2C:ADIRection READ

设置 I2C 总线地址方向为读

:SBUS:I2C:ADIRection?

查询返回 READ

## :SBUS:I2C:ADDRess

### ■ 命令格式:

:SBUS:I2C:ADDRess <value>

:SBUS:I2C:ADDRess?

### ■ 功能描述:

用于设置示波器的 I2C 总线地址。参数为 0、1 或 X 表示的二进制字符串数据，其 X 表示不确定，其范围与:SBUS:I2C:ASIZe 指令设置的值相关，为 $[0 \sim 2^{\text{addressbit}} - 1]$ 。

### ■ 返回格式:

查询返回二进制字符串数据。

### ■ 举例:

:SBUS:I2C:ADDRess "X00X00X1"

设置 I2C 总线地址为 X00X00X1



:SBUS:I2C:ADDRess?

查询返回 X00X00X1

### :SBUS:I2C:DATA:LEN

#### ■ 命令格式:

:SBUS:I2C:DATA:LEN <length>

:SBUS:I2C:DATA:LEN?

#### ■ 功能描述:

用于设置示波器的 I2C 总线触发数据长度，取值范围 1~5。

#### ■ 返回格式:

查询返回示波器的 I2C 总线触发数据长度，整型数据。

#### ■ 举例:

:SBUS:I2C:DATA:LEN 2

设置 I2C 总线触发数据长度为 2 个字节

:SBUS:I2C:DATA:LEN?

查询返回 2

### :SBUS:I2C:DATA

#### ■ 命令格式:

:SBUS:I2C:DATA <value>

:SBUS:I2C:DATA?

#### ■ 功能描述:

用于设置示波器的 I2C 总线触发数据。参数为 0、1 或 X 表示的二进制字符串数据，其 X 表示不确定，数据位取决于:SBUS:I2C:DATA:LEN 指令设置值\*8bits，其数据范围为 0x0~0xFFFFFFFFFFFFFFFF。

#### ■ 返回格式:

查询返回二进制字符串数据。

#### ■ 举例:

:SBUS:I2C:DATA "X00X00X1"

设置 I2C 总线数据为 X00X00X1

:SBUS:I2C:DATA?

查询返回 X00X00X1

### :SBUS:I2C:QUALifier

#### ■ 命令格式:

:SBUS:I2C:QUALifier {START | REStart | STOP | LOSS | ADDRess | DATA | ADATA}

:SBUS:I2C:QUALifier?

#### ■ 功能描述:

用于设置示波器的 I2C 总线触发条件。

#### ■ 返回格式:

查询返回{START | REStart | STOP | LOSS | ADDRess | DATA | ADATA}。

■ 举例：

:SBUS:I2C:QUALifier STOP                    设置 I2C 总线条件为停止  
:SBUS:I2C:QUALifier?                        查询返回 STOP

## SPI

### :SBUS:SPI:CS:SOURce

■ 命令格式：

:SBUS:SPI:CS:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}  
:SBUS:SPI:CS:SOURce?

■ 功能描述：

用于设置示波器的 SPI 总线片选源。

<Dx>：当前示波器具备 LA 功能，可设置 D0~D15 通道作为总线解码源。

■ 返回格式：

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

■ 举例：

:SBUS:SPI:CS:SOURce CHANnel1                    设置通道一为 SPI 总线片选源  
:SBUS:SPI:CS:SOURce?                            查询返回 CHANnel1

### :SBUS:SPI:CLOCK:SOURce

■ 命令格式：

:SBUS:SPI:CLOCK:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}  
:SBUS:SPI:CLOCK:SOURce?

■ 功能描述：

用于设置示波器的 SPI 总线时钟源。

<Dx>：当前示波器具备 LA 功能，可设置 D0~D15 通道作为总线解码源。

■ 返回格式：

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

■ 举例：

:SBUS:SPI:CLOCK:SOURce CHANnel1                    设置通道一为 SPI 总线时钟源  
:SBUS:SPI:CLOCK:SOURce?                            查询返回 CHANnel1

### :SBUS:SPI:MOSI:SOURce

■ 命令格式：

:SBUS:SPI:MOSI:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>|OFF}

:SBUS:SPI:MOSI:SOURce?

■ **功能描述:**

用于设置示波器的 SPI 总线主机输出从机输入源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

■ **返回格式:**

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>|OFF}。

■ **举例:**

:SBUS:SPI:MOSI:SOURce CHANnel1

设置通道一为 SPI 主机输出从机输入源

:SBUS:SPI:MOSI:SOURce?

查询返回 CHANnel1

**:SBUS:SPI:BITOrder**

■ **命令格式:**

:SBUS:SPI:BITOrder {LSBFirst | MSBFirst}

:SBUS:SPI:BITOrder?

■ **功能描述:**

用于设置示波器的 SPI 总线解码字节序。LSBFirst 小端模式、MSBFirst 大端模式。

■ **返回格式:**

查询返回{LSBFirst | MSBFirst}。

■ **举例:**

:SBUS:SPI:BITOrder LSBF

设置 SPI 字节序为 LSB

:SBUS:SPI:BITOrder?

查询返回 LSBFirst

**:SBUS:SPI:CS:POLarity**

■ **命令格式:**

:SBUS:SPI:CS:POLarity {NEGative | POSitive}

:SBUS:SPI:CS:POLarity?

■ **功能描述:**

用于设置示波器的 SPI 总线片选极性。POSitive (上升)、NEGative (下降)。

■ **返回格式:**

查询返回{NEGative | POSitive}。

■ **举例:**

:SBUS:SPI:CS:POLarity POSitive

设置 SPI 总线片选极性为上升

:SBUS:SPI:CS:POLarity?

查询返回 POSitive

**:SBUS:SPI:CLOCK:POLarity**

■ **命令格式:**

:SBUS:SPI:CLOCK:POLarity {NEGative | POSitive}

:SBUS:SPI:CLOCK:POLarity?

■ **功能描述:**

用于设置示波器的 SPI 总线时钟极性。POSitive (上升)、NEGative (下降)。

■ **返回格式:**

查询返回{NEGative | POSitive}。

■ **举例:**

:SBUS:SPI:CLOCK:POLarity POSitive

设置通 SPI 总线时钟极性为上升

:SBUS:SPI:CLOCK:POLarity?

查询返回 POSitive

### :SBUS:SPI:MOSI:POLarity

■ **命令格式:**

:SBUS:SPI:MOSI:POLarity {NEGative | POSitive}

:SBUS:SPI:MOSI:POLarity?

■ **功能描述:**

用于设置示波器的 SPI 总线主机输出从机输入极性。POSitive (上升)、NEGative (下降)。

■ **返回格式:**

查询返回{NEGative | POSitive}。

■ **举例:**

:SBUS:SPI:MOSI:POLarity POSitive

设置 SPI 主机输出从机输入极性为上升

:SBUS:SPI:MOSI:POLarity?

查询返回 POSitive

### :SBUS:SPI:WIDTh

■ **命令格式:**

:SBUS:SPI:WIDTh <width>

:SBUS:SPI:WIDTh?

■ **功能描述:**

用于设置示波器的 SPI 总线数据位宽。

<width>整型数据, 范围 4~32。

■ **返回格式:**

查询返回 SPI 总线数据位宽。

■ **举例:**

:SBUS:SPI:WIDTh 4

设置 SPI 总线数据位宽为 4

:SBUS:SPI:WIDTh?

查询返回 4

### :SBUS:SPI:FRAMelen

■ **命令格式:**

:SBUS:SPI:FRAMelen <len>

:SBUS:SPI:FRAMelen?

■ **功能描述:**

用于设置示波器的 SPI 总线数据帧长度。

<len>整型数据，范围 1~32，其中数据位宽\*数据帧长度不超过 128 位数据。

■ **返回格式：**

查询返回 SPI 总线数据帧长度，整型数据。

■ **举例：**

:SBUS:SPI:FRAMelen 1	设置 SPI 总线数据帧长度为 1
:SBUS:SPI:FRAMelen?	查询返回 1

**:SBUS:SPI:DATA**

■ **命令格式：**

:SBUS:SPI:DATA <value>  
:SBUS:SPI:DATA?

■ **功能描述：**

用于设置示波器的 SPI 总线数据，参数为 0、1 或 X 表示的二进制字符串数据，其 X 表示不确定，数据位取决于 :SBUS:SPI:WIDTH 和 :SBUS:SPI:FRAMelen 指令设置值的乘积，其数据范围为 0x0~0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF。

■ **返回格式：**

查询返回二进制字符串数据。

■ **举例：**

:SBUS:SPI:DATA "X00X00X1"	设置 SPI 总线数据为 X00X00X1
:SBUS:SPI:DATA?	查询返回 X00X00X1

**:SBUS:SPI:QUALifier**

■ **命令格式：**

:SBUS:SPI:QUALifier {CS|CSDATA|IDLE|IDLEDATA}  
:SBUS:SPI:QUALifier?

■ **功能描述：**

用于设置示波器的 SPI 总线条件。

■ **返回格式：**

查询返回{CS|CSDATA|IDLE|IDLEDATA}。

■ **举例：**

:SBUS:SPI:QUALifier CS	设置 I2C 总线条件为片选
:SBUS:SPI:QUALifier?	查询返回 CS

**:SBUS:SPI:TRIGger:TIMEout**

■ **命令格式：**

:SBUS:SPI:TRIGger:TIMEout <vlaue>  
:SBUS:SPI:TRIGger:TIMEout?

■ **功能描述：**

用于设置示波器的 SPI 总线触发超时时间。参数类型为整型，<vlaue> 等于  $n * 4ns$  且值不超过[100ns,1s) 范围。n 的取值范围 [25,25\*10<sup>8</sup>]。

- **返回格式:**

查询返回触发超时时间值, 采用科学计数法, 单位为 s。

- **举例:**

:SBUS:SPI:TRIGger:TIMEout 100ns            设置 SPI 总线触发超时时间为 100ns  
:SBUS:SPI:TRIGger:TIMEout?                查询返回 1.000e-007

## CAN

用于设置 CAN 总线解码和触发相关参数。

### :SBUS:CAN:SOURce

- **命令格式:**

:SBUS:CAN:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}  
:SBUS:CAN:SOURce?

- **功能描述:**

用于设置示波器的 CAN 总线输入源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

- **返回格式:**

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

- **举例:**

:SBUS:CAN:SOURce CHANnel1                设置通道一为 CAN 总线输入源  
:SBUS:CAN:SOURce?                        查询返回 CHANnel1

### :SBUS:CAN:SIGNal:DEFinition

- **命令格式:**

:SBUS:CAN:SIGNal:DEFinition { CANH | CANL}  
:SBUS:CAN:SIGNal:DEFinition?

- **功能描述:**

用于设置示波器的 CAN 总线信号类型。

- **返回格式:**

查询返回{ CANH | CANL}。

- **举例:**

:SBUS:CAN:SIGNal:DEFinition CANH            设置通道一为 CAN 总线信号 CANH  
:SBUS:CAN:SIGNal:DEFinition?                查询返回 CANH

## **:SBUS:CAN:SIGNal:BAUDrate**

### ■ 命令格式:

:SBUS:CAN:SIGNal:BAUDrate <baudrate>

:SBUS:CAN:SIGNal:BAUDrate?

### ■ 功能描述:

用于设置示波器的 CAN 总线信号波特率, <baudrate>范围为 10000~1000000, 单位 bps。

### ■ 返回格式:

查询返回信号波特率。

### ■ 举例:

:SBUS:CAN:SIGNal:BAUDrate 100000

设置 CAN 总线信号波特率为 100kbps

:SBUS:CAN:SIGNal:BAUDrate?

查询返回 100000

## **:SBUS:CAN:QUALifier**

### ■ 命令格式:

:SBUS:CAN:QUALifier {START|ID|DATA|ACKError|BITFILL|IDDATA|END}

:SBUS:CAN:QUALifier?

### ■ 功能描述:

用于设置示波器的 CAN 总线触发条件

{START|ID|DATA|ACKError|BITFILL|IDDATA|END}, 分别表示起始、标识符、数据帧、丢失确认、位填充、标识符&数据、结束。

### ■ 返回格式:

查询返回{START | ID | DATA | ACKError | BITFILL | IDDATA | END}。

### ■ 举例:

:SBUS:CAN:QUALifier ACK

设置 CAN 总线触发条件为 ACK 错误

:SBUS:CAN:QUALifier?

查询返回 ACKError

## **:SBUS:CAN:FRAME:TYPE**

### ■ 命令格式:

:SBUS:CAN:FRAME:TYPE { DATA | REMote | OVERload | ERRor }

:SBUS:CAN:FRAME:TYPE?

### ■ 功能描述:

用于设置示波器的 CAN 总线触发帧类型。

### ■ 返回格式:

查询返回{ DATA | REMote | OVERload | ERRor }。

### ■ 举例:

:SBUS:CAN:FRAME:TYPE ERRor                    设置 CAN 总线触发帧类型为错误帧  
:SBUS:CAN:FRAME:TYPE?                        查询返回 ERRor

## :SBUS:CAN:ID:MODE

### ■ 命令格式:

:SBUS:CAN:ID:MODE {STANdard | EXTended}

:SBUS:CAN:ID:MODE?

### ■ 功能描述:

用于设置示波器的 CAN 总线 ID 标识符帧格式。

### ■ 返回格式:

查询返回{STANdard | EXTended}。

### ■ 举例:

:SBUS:CAN:ID:MODE STANdard                    设置 CAN 总线 ID 帧格式为标准帧

:SBUS:CAN:ID:MODE?                        查询返回 STANdard

## :SBUS:CAN:ID

### ■ 命令格式:

:SBUS:CAN:ID <string>

:SBUS:CAN:ID?

### ■ 功能描述:

用于设置示波器的 CAN 总线 ID 标识符帧数据，根据:SBUS:CAN:ID:MODE 设置的模式来设置对应的 ID 标识，参数为 0、1 或 X 表示的二进制字符串数据，其中 X 表示不确定。

标准帧范围为 0x0~07FF，数据位占 11bits；扩展帧范围为 0x0~0x1FFFFFFF，数据位占 29bits。

### ■ 返回格式:

查询返回二进制字符串数据。

### ■ 举例:

:SBUS:CAN:ID "X00X00X1"                    设置 CAN 总线 ID 标识符帧数据 X00X00X1

:SBUS:CAN:ID?                            查询返回 X00X00X1

## :SBUS:CAN:ID:DIRection

### ■ 命令格式:

:SBUS:CAN:ID:DIRection { READ | WRITE | ANY }

:SBUS:CAN:ID:DIRection?

### ■ 功能描述:

用于设置示波器的 CAN 总线 ID 标识符方向。



- **返回格式:**  
查询返回{ READ | WRITE | ANY}。
- **举例:**  
:SBUS:CAN:ID:DIRrection READ            设置 CAN 总线 ID 方向为读  
:SBUS:CAN:ID:DIRrection?                查询返回 READ

### :SBUS:CAN:DATA:LEN

- **命令格式:**  
:SBUS:CAN:DATA:LEN <length>  
:SBUS:CAN:DATA:LEN?
- **功能描述:**  
用于设置示波器的 CAN 总线触发数据长度，取值范围 1~8。
- **返回格式:**  
查询返回示波器的 CAN 总线触发数据长度，整型数据。
- **举例:**  
:SBUS:CAN:DATA:LEN 2                    设置 CAN 总线触发数据长度为 2 个字节  
:SBUS:CAN:DATA:LEN?                    查询返回 2

### :SBUS:CAN:DATA

- **命令格式:**  
:SBUS:CAN:DATA <string>  
:SBUS:CAN:DATA?
- **功能描述:**  
用于设置示波器的 CAN 总线 DATA 数据，参数为 0、1 或 X 表示的二进制字符串数据，其中 X 表示不确定，数据位取决于 :SBUS:CANFD:DATA:LEN 指令设置值 \*8bits，其数据范围为 0x0~0xFFFFFFFFFFFFFFFF。
- **返回格式:**  
查询返回二进制字符串数据。
- **举例:**  
:SBUS:CAN:DATA "X00X00X1"            设置 CAN 总线 DATA 为 X00X00X1  
:SBUS:CAN:DATA?                        查询返回 X00X00X1

## CAN-FD

用于设置 CAN-FD 总线解码和触发相关参数。

### :SBUS:CANFD:SOURce

■ **命令格式:**

:SBUS:CANFD:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}

:SBUS:CANFD:SOURce?

■ **功能描述:**

用于设置示波器的 CAN-FD 总线输入源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

■ **返回格式:**

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

■ **举例:**

:SBUS:CANFD:SOURce CHANnel1

设置通道一为 CAN-FD 总线输入源

:SBUS:CANFD:SOURce?

查询返回 CHANnel1

### :SBUS:CANFD:SIGNal:DEFinition

■ **命令格式:**

:SBUS:CANFD:SIGNal:DEFinition { CANH | CANL }

:SBUS:CANFD:SIGNal:DEFinition?

■ **功能描述:**

用于设置示波器的 CAN-FD 总线信号类型。

■ **返回格式:**

查询返回{ CANH | CANL }。

■ **举例:**

:SBUS:CANFD:SIGNal:DEFinition CANH

设置通道一 CAN-FD 总线信号 CANH

:SBUS:CANFD:SIGNal:DEFinition?

查询返回 CANH

### :SBUS:CANFD:BAUDrate

■ **命令格式:**

:SBUS:CANFD:BAUDrate <baudrate>

:SBUS:CANFD:BAUDrate?

■ **功能描述:**

用于设置示波器的 CAN-FD 总线信号波特率, <baudrate>范围为 10000~1000000, 单位 bps。

■ **返回格式:**

查询返回信号波特率。

■ **举例:**

:SBUS:CANFD:BAUDrate 100000

设置 CAN-FD 总线信号波特率为 100kbps

:SBUS:CANFD:BAUDrate?

查询返回 100000

## :SBUS:CANFD:FD:BAUDrate

### ■ 命令格式:

:SBUS:CANFD:FD:BAUDrate <baudrate>

:SBUS:CANFD:FD:BAUDrate?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线信号 FD 波特率, <baudrate>范围为 250000~8000000, 单位 bps。

### ■ 返回格式:

查询返回信号波特率。

### ■ 举例:

:SBUS:CANFD:FD:BAUDrate 250000

设置 CAN-FD 总线 FD 波特率为 250kbps

:SBUS:CANFD:FD:BAUDrate?

查询返回 100000

## :SBUS:CANFD:QUALifier

### ■ 命令格式:

:SBUS:CANFD:QUALifier {START|ID|DATA|ACKError|BITFILL|IDDATA|END}

:SBUS:CANFD:QUALifier?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线触发条件

{START|ID|DATA|ACKError|BITFILL|IDDATA|END}, 分别表示起始、标识符、数据帧、丢失确认、位填充、标识符&数据、结束。

### ■ 返回格式:

查询返回{START | ID | DATA | ACKError | BITFILL | IDDATA | END}。

### ■ 举例:

:SBUS:SPI:QUALifier ACK

设置 CAN-FD 总线触发条件为 ACK 错误

:SBUS:SPI:QUALifier?

查询返回 ACKError

## :SBUS:CANFD:FRAME:TYPE

### ■ 命令格式:

:SBUS:CANFD:FRAME:TYPE { DATA | REMote | OVERload | ERRor }

:SBUS:CANFD:FRAME:TYPE?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线触发帧类型。

### ■ 返回格式:

查询返回{ DATA | REMote | OVERload | ERRor }。

### ■ 举例:

:SBUS:CANFD:FRAME:TYPE ERRor

设置 CAN-FD 总线触发帧类型为错误帧

:SBUS:CANFD:FRAME:TYPE?

查询返回 ERRor

## :SBUS:CANFD:ID:MODE

### ■ 命令格式:

:SBUS:CANFD:ID:MODE {STANdard | EXTended | FD\_STD| FD\_EXT }  
:SBUS:CANFD:ID:MODE?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线 ID 标识符帧格式。

### ■ 返回格式:

查询返回{STANdard | EXTended | FD\_STD| FD\_EXT }。

### ■ 举例:

:SBUS:CANFD:ID:MODE STANdard                    设置 CAN-FD 总线 ID 帧格式为标准帧  
:SBUS:CANFD:ID:MODE?                            查询返回 STANdard

## :SBUS:CANFD:ID

### ■ 命令格式:

:SBUS:CANFD:ID <string>  
:SBUS:CANFD:ID?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线 ID 标识符帧数据，根据:SBUS:CANFD:ID:MODE 设置的模式来设置对应的 ID 标识，参数为 0、1 或 X 表示的二进制字符串数据，其中 X 表示不确定。

标准帧范围为 0x0~0x7FFF，占 11bits；扩展帧范围为 0x0~0x1FFFFFFF，占 29bits；FD 标准帧范围为 0x0~0x7FFF，占 11bits；FD 扩展帧范围为 0x0~0x1FFFFFFF，占 29bits。

### ■ 返回格式:

查询返回二进制字符串数据。

### ■ 举例:

:SBUS:CANFD:ID "X00X00X1"                    设置 CAN-FD 总线 ID 标识符帧数据 X00X00X1  
:SBUS:CANFD:ID?                                查询返回 X00X00X1

## :SBUS:CANFD:DATA:LEN

### ■ 命令格式:

:SBUS:CANFD:DATA:LEN <length>  
:SBUS:CANFD:DATA:LEN?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线触发数据长度，取值范围 1~16。

### ■ 返回格式:

查询返回示波器的 CAN-FD 总线触发数据长度，整型数据。

### ■ 举例:

:SBUS:CANFD:DATA:LEN 2                        设置 CAN-FD 总线触发数据长度 2 个字节  
:SBUS:CANFD:DATA:LEN?                        查询返回 2

## :SBUS:CANFD:DATA

### ■ 命令格式:

:SBUS:CANFD:DATA <string>

:SBUS:CANFD:DATA?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线 DATA 数据，参数为 0、1 或 X 表示的二进制字符串数据，其中 X 表示不确定，数据位取决于 :SBUS:CANFD:DATA:LEN 指令设置值 \* 8bits，其数据范围为 0x0~0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF。

### ■ 返回格式:

查询返回二进制字符串数据。

### ■ 举例:

:SBUS:CANFD:DATA "X00X00X1"            设置 CAN-FD 总线 DATA 为 X00X00X1

:SBUS:CANFD:DATA?                        查询返回 X00X00X1

## :SBUS:CANFD:DATA:OFFSet:CTL

### ■ 命令格式:

:SBUS:CANFD:DATA:OFFSet:CTL { {1|ON} | {0|OFF} }

:SBUS:CANFD:DATA:OFFSet:CTL?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线 DATA 数据字节偏置开关状态。

### ■ 返回格式:

查询返回 1 或 0，分别代表 ON 或 OFF。

### ■ 举例:

:SBUS:CANFD:DATA:OFFSet:CTL ON    打开 CAN-FD 总线 DATA 数据字节偏置

:SBUS:CANFD:DATA:OFFSet:CTL?        查询返回 1

## :SBUS:CANFD:DATA:OFFSet

### ■ 命令格式:

:SBUS:CANFD:DATA:OFFSet <offset>

:SBUS:CANFD:DATA:OFFSet?

### ■ 功能描述:

用于设置示波器的 CAN-FD 总线 DATA 数据字节偏置，使用此指令默认偏置已打开。

<offset>: 字节偏置，整型数据类型，范围 0~63。

### ■ 返回格式:

查询返回字节偏置。

### ■ 举例:

:SBUS:CANFD:DATA:OFFSet 8            设置 CAN-FD 总线 DATA 数据字节偏置为 8

:SBUS:CANFD:DATA:OFFSet?            查询返回 8

## :SBUS:LIN:SOURce

- **命令格式:**

```
:SBUS:LIN:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}
:SBUS:LIN:SOURce?
```

- **功能描述:**

用于设置示波器的 LIN 总线解码源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

- **返回格式:**

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

- **举例:**

```
:SBUS:LIN:SOURce CHANnel1      设置通道一为 LIN 总线解码源
:SBUS:LIN:SOURce?              查询返回 CHANnel1
```

## :SBUS:LIN:POLarity

- **命令格式:**

```
:SBUS:LIN:POLarity {NORMal | INVert}
:SBUS:LIN:POLarity?
```

- **功能描述:**

用于设置示波器的 LIN 总线极性。NORMal (正常, 高=1)、INVert (反转, 高=0)。

- **返回格式:**

查询返回{NORMal | INVert}。

- **举例:**

```
:SBUS:LIN:POLarity NORMal      设置 LIN 总线极性为正常
:SBUS:LIN:POLarity?           查询返回 POSitive
```

## :SBUS:LIN:VERSion

- **命令格式:**

```
:SBUS:LIN:VERSion {VER1|VER2|ANY}
:SBUS:LIN:VERSion?
```

- **功能描述:**

用于设置示波器的 LIN 总线版本。

{VER1|VER2|ANY}: V1.x 版本、V2.x 版本、任意版本。

- **返回格式:**

查询返回{VER1|VER2|ANY}。

- **举例:**

```
:SBUS:LIN:VERSion VER1        设置 LIN 总线版本为 V1.x 版本
:SBUS:LIN:VERSion?           查询返回 POSitive
```

## **:SBUS:LIN:SIGNal:BAUDrate**

### ■ 命令格式:

:SBUS:LIN:SIGNal:BAUDrate <baudrate>

:SBUS:LIN:SIGNal:BAUDrate?

### ■ 功能描述:

用于设置示波器的 LIN 总线信号波特率, <baudrate>范围为 1~100000, 单位 bps。

### ■ 返回格式:

查询返回信号波特率。

### ■ 举例:

:SBUS:LIN:SIGNal:BAUDrate 100000      设置 LIN 总线信号波特率为 100kbps

:SBUS:LIN:SIGNal:BAUDrate?              查询返回 100000

## **:SBUS:LIN:PARity:DISPlay**

### ■ 命令格式:

:SBUS:LIN:PARity:DISPlay { {1|ON} | {0|OFF} }

:SBUS:LIN:PARity:DISPlay?

### ■ 功能描述:

用于设置示波器的 LIN 总线 ID 是否包含奇偶位, ON (是) 或 OFF (否)。

### ■ 返回格式:

查询返回 1 或 0, 分别代表 ON 或 OFF。

### ■ 举例:

:SBUS:LIN:PARity:DISPlay ON              LIN 总线解码 ID 包含奇偶位。

:SBUS:LIN:PARity:DISPlay?              查询返回 1

## **:SBUS:LIN:DATA:LENGth:DISPlay**

### ■ 命令格式:

:SBUS:LIN:DATA:LENGth:DISPlay { {1|ON} | {0|OFF} }

:SBUS:LIN:DATA:LENGth:DISPlay?

### ■ 功能描述:

用于示波器的 LIN 总线是否设置数据长度, ON (是) 或 OFF (否)。

### ■ 返回格式:

查询返回 1 或 0, 分别代表 ON 或 OFF。

### ■ 举例:

:SBUS:LIN:DATA:LENGth:DISPlay ON      LIN 总线解码设置数据长度。

:SBUS:LIN:DATA:LENGth:DISPlay?        查询返回 1

## **:SBUS:LIN:DATA:LENGth**

### ■ 命令格式:

:SBUS:LIN:DATA:LENGth <length>

:SBUS:LIN:DATA:LENGth?

■ **功能描述:**

用于示波器的 LIN 总线设置数据的长度, 使用此指令默认设置数据长度已打开。

<length>: 数据长度, 整型数据类型, 范围 1~8。

■ **返回格式:**

查询返回数据长度。

■ **举例:**

:SBUS:LIN:DATA:LENGth 6                      LIN 总线解码设置数据长度为 6。

:SBUS:LIN:DATA:LENGth?                      查询返回 6

**:SBUS:LIN:QUALifier**

■ **命令格式:**

:SBUS:LIN:QUALifier {SYNC|ID|DATA|IDDATA|WAKE|SLEEP|ERROR}

:SBUS:LIN:QUALifier?

■ **功能描述:**

用于设置示波器的 LIN 总线触发条件。

{SYNC|ID|DATA|IDDATA|WAKE|SLEEP|ERROR}: 同步、标识符、数据、ID 和数据、唤醒帧、睡眠帧、错误。

■ **返回格式:**

查询返回{SYNC|ID|DATA|IDDATA|WAKE|SLEEP|ERROR}。

■ **举例:**

:SBUS:LIN:QUALifier SYNC                      设置 LIN 总线触发条件为同步

:SBUS:LIN:QUALifier?                      查询返回 SYNC'

**:SBUS:LIN:ID**

■ **命令格式:**

:SBUS:LIN:ID <string>

:SBUS:LIN:ID?

■ **功能描述:**

用于设置示波器 LIN 总线标识符数据, 参数为 0、1 或 X 表示的二进制字符串数据, 占用 8bits, 其中 X 表示不确定, 其数据范围为 0x0~0xFF。

■ **返回格式:**

查询返回二进制字符串数据。

■ **举例:**

:SBUS:LIN:ID "X00X00X1"                      设置 LIN 总线标识符数据为 X00X00X1

:SBUS:LIN:ID?                      查询返回 X00X00X1

**:SBUS:LIN:TRIGger:DATA:LENGth**

■ **命令格式:**

:SBUS:LIN:TRIGger:DATA:LENGth <length>



:SBUS:LIN:TRIGger:DATA:LENGth?

■ **功能描述:**

用于设置示波器的 LIN 总线触发数据长度, 取值范围 1~8。

■ **返回格式:**

查询返回示波器的 LIN 总线触发数据长度, 整型数据。

■ **举例:**

:SBUS:LIN:TRIGger:DATA:LENGth 2      设置 LIN 总线触发数据长度 2 个字节

:SBUS:LIN:TRIGger:DATA:LENGth?      查询返回 2

## :SBUS:LIN:DATA

■ **命令格式:**

:SBUS:LIN:DATA <string>

:SBUS:LIN:DATA?

■ **功能描述:**

用于设置示波器的 LIN 总线 DATA 数据, 参数为 0、1 或 X 表示的二进制字符串数据, 其中 X 表示不确定, 长度取决于 :SBUS:LIN:TRIGger:DATA:LENGth 指令设置值 \* 8bits, 其数据范围为 0x0~0xFFFFFFFFFFFFFFFF。

■ **返回格式:**

查询返回二进制字符串数据。

■ **举例:**

:SBUS:LIN:DATA "X00X00X1"      设置 LIN 总线 DATA 为 X00X00X1

:SBUS:LIN:DATA?      查询返回 X00X00X1

## :SBUS:LIN:ERRor:TYPE

■ **命令格式:**

:SBUS:LIN:ERRor:TYPE { SYNC | PARity | SUM}

:SBUS:LIN:ERRor:TYPE?

■ **功能描述:**

用于设置示波器的 LIN 总线触发条件错误类型。

{ SYNC | PARity | SUM}: 同步、ID 奇偶校验、校验和。

■ **返回格式:**

查询返回{ SYNC | PARity | SUM}。

■ **举例:**

:SBUS:LIN:ERRor:TYPE SYNC      设置 LIN 总线触发条件为同步错误

:SBUS:LIN:ERRor:TYPE?      查询返回 SYNC

# FlexRay

## :SBUS:FR:SOURce

### ■ 命令格式:

:SBUS:FR:SOURce {CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}

:SBUS:FR:SOURce?

### ■ 功能描述:

用于设置示波器的 FlexRay 总线解码源。

<Dx>: 当前示波器具备 LA 功能, 可设置 D0~D15 通道作为总线解码源。

### ■ 返回格式:

查询返回{CHANnel1|CHANnel2|CHANnel3|CHANnel4|<Dx>}。

### ■ 举例:

:SBUS:FR:SOURce CHANnel1

设置通道一为 FlexRay 总线解码源

:SBUS:FR:SOURce?

查询返回 CHANnel1

## :SBUS:FR:POLarity

### ■ 命令格式:

:SBUS:FR:POLarity {BP | BM}

:SBUS:FR:POLarity?

### ■ 功能描述:

用于设置示波器的 FlexRay 总线极性, BP (Bdiff 或 BP)。

### ■ 返回格式:

查询返回{BP | BM}。

### ■ 举例:

:SBUS:FR:POLarity BP

设置 FlexRay 总线极性为 BP

:SBUS:FR:POLarity?

查询返回 BP

## :SBUS:FR:CHANnel:TYPE

### ■ 命令格式:

:SBUS:FR:CHANnel:TYPE {A | B}

:SBUS:FR:CHANnel:TYPE?

### ■ 功能描述:

用于设置示波器的 FlexRay 总线通道类型。

### ■ 返回格式:

查询返回{A | B}。

■ **举例：**

:SBUS:FR:CHANnel:TYPE A                    设置 FlexRay 通道类型为 A  
:SBUS:FR:CHANnel:TYPE?                    查询返回 A

**:SBUS:FR:SIGNal:BAUDrate**

■ **命令格式：**

:SBUS:FR:SIGNal:BAUDrate {2.5M|5M|10M}  
:SBUS:FR:SIGNal:BAUDrate?

■ **功能描述：**

用于设置示波器的 FlexRay 总线信号波特率。

■ **返回格式：**

查询返回信号波特率。

■ **举例：**

:SBUS:FR:SIGNal:BAUDrate 2.5M            设置 FlexRay 总线信号波特率为 2.5Mbps  
:SBUS:FR:SIGNal:BAUDrate?            查询返回 2.5M

**:SBUS:FR:QUALifier**

■ **命令格式：**

:SBUS:FR:QUALifier {START|IND|ID|CYCLES|HEAD|DATA|IDDATA|END|ERROR}  
:SBUS:FR:QUALifier?

■ **功能描述：**

用于设置示波器的 FlexRay 总线触发条件。

{START|IND|ID|CYCLES|HEAD|DATA|IDDATA|END|ERROR}：帧起始、指示位(Indicator)、标识位、循环数、标头字段、数据、ID 和数据、帧结尾、错误。

■ **返回格式：**

查询返回{START|IND|ID|CYCLES|HEAD|DATA|IDDATA|END|ERROR}。

■ **举例：**

:SBUS:FR:QUALifier START                    设置 FlexRay 总线触发条件为帧起始  
:SBUS:FR:QUALifier?                    查询返回 START

**:SBUS:FR:INDicator**

■ **命令格式：**

:SBUS:FR:INDicator {NORMal|STATIC|NULL|SYNC|START}  
:SBUS:FR:INDicator?

■ **功能描述：**

用于设置示波器的 FlexRay 总线触发指示位类型。

{NORMal|STATIC|NULL|SYNC|START}: 正常(01XX)、静载荷(11XX)、空(00XX)、同步(XX10)、启动(XX11)。

■ **返回格式:**

查询返回{NORMal|STATIC|NULL|SYNC|START}。

■ **举例:**

:SBUS:FR:INDicator NORMal                    设置 FlexRay 总线触发指示位正常状态  
:SBUS:FR:INDicator?                         查询返回 NORMal

**:SBUS:FR:INDicator:DATA**

■ **命令格式:**

:SBUS:FR:INDicator:DATA <string>  
:SBUS:FR:INDicator:DATA?

■ **功能描述:**

用于设置示波器的 FlexRay 总线触发条件下标头字段指示位数据, 参数为 0、1 或 X 表示的二进制字符串数据, X 表示不确定, 数据位为 5bits, 其中数据范围为 0x0~0x1F。

■ **返回格式:**

查询返回 0x 格式的字符串类型。

■ **举例:**

:SBUS:FR:INDicator:DATA "000100X1"    设置 FlexRay 总线指示位数据为 000100X1  
:SBUS:FR:INDicator:DATA?                查询返回 000100X1

**:SBUS:FR:PAYLoad:SIZE**

■ **命令格式:**

:SBUS:FR:PAYLoad:SIZE <size>  
:SBUS:FR:PAYLoad:SIZE?

■ **功能描述:**

用于设置示波器的 FlexRay 总线触发条件下标头字段静载荷长度, 参数为 0、1 或 X 表示的二进制字符串数据, X 表示不确定, 数据位为 7bits, 其数据范围为 0x0~0x7F。

■ **返回格式:**

查询返回二进制字符串数据。

■ **举例:**

:SBUS:FR:PAYLoad:SIZE "X00X00X1"    设置 FlexRay 总线静载荷长度为 X00X00X1  
:SBUS:FR:PAYLoad:SIZE?                查询返回 X00X00X1

**:SBUS:FR:HEADer:CRC**

■ **命令格式:**

:SBUS:FR:HEADer:CRC <crc>  
:SBUS:FR:HEADer:CRC?

■ **功能描述:**

用于设置示波器的 FlexRay 总线触发条件下标头字段标头 CRC, 参数为 0、1 或 X 表示的二进制字符串数据, X 表示不确定, 数据位为 11bits, 其数据范围为 0x0~0x7FF。

■ **返回格式:**

查询返回二进制字符串数据。

■ **举例:**

:SBUS:FR:HEADer:CRC "X00X00X1"

设置 FlexRay 总线标头 CRC 为 X00X00X1

:SBUS:FR:HEADer:CRC?

查询返回 X00X00X1

**:SBUS:FR:CYCLes**

■ **命令格式:**

:SBUS:FR:CYCLes <string>

:SBUS:FR:CYCLes?

■ **功能描述:**

用于示波器的 FlexRay 总线触发条件循环数下的循环数目，参数为 0、1 或 X 表示的二进制字符串数据，X 表示不确定，数据位为 6bits，其数据范围为 0x0~0x3F。

■ **返回格式:**

查询返回二进制字符串数据。

■ **举例:**

:SBUS:FR:CYCLes "0X00X1"

设置 FlexRay 总线触发循环数为 0X00X1。

:SBUS:FR:CYCLes?

查询返回 0X00X1

**:SBUS:FR:ID**

■ **命令格式:**

:SBUS:FR:ID <string>

:SBUS:FR:ID?

■ **功能描述:**

用于设置示波器 FlexRay 总线标识符数据，参数为 0、1 或 X 表示的二进制字符串数据，X 表示不确定，数据位占 11bits，其数据范围为 0x0~0x7FF。

■ **返回格式:**

查询返回二进制字符串数据。

■ **举例:**

:SBUS:FR:ID "X00X00X1"

设置 FlexRay 总线标识符数据为 X00X00X1

:SBUS:FR:ID?

查询返回 X00X00X1

**:SBUS:FR:DATA:LEN**

■ **命令格式:**

:SBUS:FR:DATA:LEN <length>

:SBUS:FR:DATA:LEN?

■ **功能描述:**

用于设置示波器的 FlexRay 总线触发数据长度，取值范围 1~16。

■ **返回格式:**

查询返回示波器的 FlexRay 总线触发数据长度，整型数据。

- **举例:**  
 :SBUS:FR:DATA:LEN 2                      设置 FlexRay 总线触发数据长度 2 个字节  
 :SBUS:FR:DATA:LEN?                      查询返回 2

### **:SBUS:FR:DATA**

- **命令格式:**  
 :SBUS:FR:DATA <string>  
 :SBUS:FR:DATA?
- **功能描述:**  
 用于设置示波器的 FlexRay 总线触发不确定的数据，参数为 0、1 或 X 表示的二进制字符串数据，X 表示不确定，数据位取决于 :SBUS:FR:DATA:LEN 指令设置值 \*8bits，其数据范围为 0x0~0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF。
- **返回格式:**  
 查询返回二进制字符串数据。
- **举例:**  
 :SBUS:FR:DATA "X00X00X1"              设置 FlexRay 总线数据为 X00X00X1  
 :SBUS:FR:DATA?                          查询返回 X00X00X1

### **:SBUS:FR:DATA:OFFSet:CTL**

- **命令格式:**  
 :SBUS:FR:DATA:OFFSet:CTL { {1|ON} | {0|OFF} }  
 :SBUS:FR:DATA:OFFSet:CTL?
- **功能描述:**  
 用于设置示波器的 FlexRay 总线 DATA 数据字节偏置开关状态。
- **返回格式:**  
 查询返回 1 或 0，分别代表 ON 或 OFF。
- **举例:**  
 :SBUS:FR:DATA:OFFSet:CTL ON            打开 FlexRay 总线 DATA 数据字节偏置  
 :SBUS:FR:DATA:OFFSet:CTL?            查询返回 1

### **:SBUS:FR:DATA:OFFSet**

- **命令格式:**  
 :SBUS:FR:DATA:OFFSet <offset>  
 :SBUS:FR:DATA:OFFSet?
- **功能描述:**  
 用于设置示波器的 FlexRay 总线 DATA 数据字节偏置，使用此指令默认偏置已打开。  
 <offset>: 字节偏置，整型数据类型，范围 0~253。
- **返回格式:**  
 查询返回字节偏置。
- **举例:**

:SBUS:FR:DATA:OFFSet 8                    设置 FlexRay 总线 DATA 数据字节偏置为 8  
:SBUS:FR:DATA:OFFSet?                    查询返回 8

#### **:SBUS:FR:END:TYPE**

■ **命令格式:**

:SBUS:FR:END:TYPE { STATIC | DTS | ALL}

:SBUS:FR:END:TYPE?

■ **功能描述:**

用于设置示波器的 FlexRay 总线触发条件帧结尾的帧类型。

{ STATIC | DTS | ALL}: 静态、动态、全部。

■ **返回格式:**

查询返回{ STATIC | DTS | ALL}。

■ **举例:**

:SBUS:FR:END:TYPE STATIC                    设置 FlexRay 总线触发条件静态段帧结尾

:SBUS:FR:END:TYPE?                    查询返回 STATIC

#### **:SBUS:FR:ERRor:TYPE**

■ **命令格式:**

:SBUS:FR:ERRor:TYPE

{HEADCRC| ENDCRC | NULLSTATIC|NULLDYNAMIC|SYNC|START}

:SBUS:FR:ERRor:TYPE?

■ **功能描述:**

用于设置示波器的 FlexRay 总线触发条件错误类型。

{HEADCRC|ENDCRC|NULLSTATIC|NULLDYNAMIC|SYNC|START}: 标头 CRC、帧尾 CRC、空帧静态、空帧动态、同步帧、启动帧。

■ **返回格式:**

查询返回{HEADCRC| ENDCRC | NULLSTATIC|NULLDYNAMIC|SYNC|START}。

■ **举例:**

:SBUS:FR:ERRor:TYPE SYNC                    设置 FlexRay 总线触发条件错误类型为同步

:SBUS:FR:ERRor:TYPE?                    查询返回 SYNC

## AWG 命令

用于设置示波器内置信号源相关功能。

## 通道指令

用于设置内置信号源通道相关功能。

### AWG:CHANnel<n>:OUTPut

#### ■ 命令格式:

```
AWG:CHANnel<n>:OUTPut {{1 | ON} | {0 | OFF}}
```

```
AWG:CHANnel<n>:OUTPut?
```

#### ■ 功能描述:

设置打开或关闭指定通道的输出。

<n>: 通道号, n 取值 1、2。

#### ■ 返回格式:

查询返回指定通道的输出状态, 0 表示关闭, 1 表示打开。

#### ■ 举例:

```
AWG:CHANnel1:OUTPut ON           设置打开通道 1 输出
```

```
AWG:CHANnel1:OUTPut?             查询返回 1
```

### AWG:CHANnel<n>:LOAD

#### ■ 命令格式:

```
AWG:CHANnel<n>:LOAD <resistance>
```

```
AWG:CHANnel<n>:LOAD?
```

#### ■ 功能描述:

设置指定通道输出负载。

<resistance>表示负载电阻值, 单位为  $\Omega$

<n>: 通道号, n 取值 1、2。

注意: 阻值取值范围为 1~10000,其中 10000 对应于高阻。

#### ■ 返回格式:

查询返回指定通道的负载阻值, 采用科学计数法返回。

#### ■ 举例:

```
AWG:CHANnel1:LOAD 50              设置通道 1 输出负载 50 $\Omega$ 
```

```
AWG:CHANnel1:LOAD?                查询返回 50e+0
```



## AWG:CHANnel<n>:LIMit:ENABle

### ■ 命令格式:

AWG:CHANnel<n>:LIMit:ENABle {{1 | ON} | {0 | OFF}}

AWG:CHANnel<n>:LIMit:ENABle?

### ■ 功能描述:

设置指定通道限幅开关。

<n>: 通道号, n 取值 1、2。

### ■ 返回格式:

查询返回指定通道的限幅状态。

### ■ 举例:

AWG:CHANnel1:LIMit:ENABle ON                    设置打开通道 1 限幅

AWG:CHANnel1:LIMit:ENABle?                    查询返回 1

## AWG:CHANnel<n>:LIMit:LOWer

### ■ 命令格式:

AWG:CHANnel<n>:LIMit:LOWer <voltage>

AWG:CHANnel<n>:LIMit:LOWer?

### ■ 功能描述:

设置指定通道限幅下限值。

<voltage>表示电压, 单位当前通道指定单位。

<n>: 通道号, n 取值 1、2。

### ■ 返回格式:

查询返回指定通道的限幅下限值, 采用科学计数法返回。

### ■ 举例:

AWG:CHANnel1:LIMit:LOWer 2                    设置通道 1 限幅下限 2V

AWG:CHANnel1:LIMit:LOWer?                    查询返回 2e+0

## AWG:CHANnel<n>:LIMit:UPPer

### ■ 命令格式:

AWG:CHANnel<n>:LIMit:UPPer <voltage>

AWG:CHANnel<n>:LIMit:UPPer?

### ■ 功能描述:

设置指定通道限幅上限值。

<voltage>表示电压, 单位当前通道指定单位。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道的限幅上限值, 采用科学计数法返回。

■ **举例:**

AWG:CHANnel1:LIMit:UPPer 2	设置通道 1 限幅上限 2V
AWG:CHANnel1:LIMit:UPPer?	查询返回 2e+0

### AWG:CHANnel<n>:AMPLitude:UNIT

■ **命令格式:**

AWG:CHANnel<n>:AMPLitude:UNIT {VPP | VRMS | DBM}  
AWG:CHANnel<n>:AMPLitude:UNIT?

■ **功能描述:**

设置指定通道输出幅度单位。  
<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道的输出幅度单位。

■ **举例:**

AWG:CHANnel1:AMPLitude:UNIT VPP	设置通道 1 输出幅度单位为 VPP
AWG:CHANnel1:AMPLitude:UNIT?	查询返回 VPP

### AWG:CHANnel<n>:MODE

■ **命令格式:**

AWG:CHANnel<n>:MODE {BASE | MODulate}  
AWG:CHANnel<n>:MODE?

■ **功能描述:**

设置指定通道信号模式, BASE 表示基波模式, MODulate 表示调制模式。  
<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道信号模式。

■ **举例:**

AWG:CHANnel1:MODE BASE	设置通道 1 信号以基波模式输出
AWG:CHANnel1:MODE?	查询返回 BASE

### AWG:CHANnel<n>:BASE:WAVE

■ **命令格式:**

AWG:CHANnel<n>:BASE:WAVE {SINe | SQUare | PULSe | RAMP | ARB | NOISe | DC }

AWG:CHANnel<n>:BASE:WAVE?

■ **功能描述:**

设置指定通道基波类型。分别为正弦波、方波、脉冲波、三角波、任意波、噪声、直流。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道的基波类型。

■ **举例:**

AWG:CHANnel1:BASE:WAVE SINE                    设置通道 1 基本类型为正弦波

AWG:CHANnel1:BASE:WAVE?                        查询返回 SINE

### AWG:CHANnel<n>:BASE:FREQuency

■ **命令格式:**

AWG:CHANnel<n>:BASE:FREQuency <freq>

AWG:CHANnel<n>:BASE:FREQuency?

■ **功能描述:**

设置指定通道输出频率。

<freq>表示频率值, 单位 Hz。(1e-6s ~ 当前波形允许最大频率)

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道的输出频率, 采用科学计数法返回。

■ **举例:**

AWG:CHANnel1:BASE:FREQuency 2000            设置通道 1 输出频率 2KHz

AWG:CHANnel1:BASE:FREQuency?                查询返回 2e+3

### AWG:CHANnel<n>:BASE:AMPLitude

■ **命令格式:**

AWG:CHANnel<n>:BASE:AMPLitude <amp>

AWG:CHANnel<n>:BASE:AMPLitude?

■ **功能描述:**

设置指定通道输出幅度。

<amp>表示电压, 单位当前通道指定单位。1mVpp ~ 当前负载下输出的最大值。

若当前单位为 VPP, 当前负载下最大值=当前负载\*20/(50+当前负载)

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道的输出幅度, 采用科学计数法返回。

■ **举例:**

AWG:CHANnel1:BASE:AMPLitude 2

设置通道 1 输出幅度为 2V

AWG:CHANnel1:BASE:AMPLitude?

查询返回 2e+0

### AWG:CHANnel<n>:BASE:PERiod

#### ■ 命令格式:

AWG:CHANnel<n>:BASE:PERiod <period>

AWG:CHANnel<n>:BASE:PERiod?

#### ■ 功能描述:

设置指定通道输出周期。

<period>表示周期，单位 S。

若为正弦波：范围为（当前允许最大时间 ~ 1e3s）

<n>：通道号，n 取值 1、2。

#### ■ 返回格式:

查询返回指定通道的限幅上限值，采用科学计数法返回。

#### ■ 举例:

AWG:CHANnel1:BASE:PERiod 0.002

设置通道 1 输出周期 2ms

AWG:CHANnel1:BASE:PERiod?

查询返回 2e-3

### AWG:CHANnel<n>:BASE:PHASe

#### ■ 命令格式:

AWG:CHANnel<n>:BASE:PHASe <phase>

AWG:CHANnel<n>:BASE:PHASe?

#### ■ 功能描述:

设置指定通道输出相位。

<phase>表示相位，单位°，范围-360~360。

<n>：通道号，n 取值 1、2。

#### ■ 返回格式:

查询返回指定通道的输出相位。

#### ■ 举例:

AWG:CHANnel1:BASE:PHASe 20

设置通道 1 输出相位为 20°

AWG:CHANnel1:BASE:PHASe?

查询返回 20

### AWG:CHANnel<n>:BASE:OFFSet

#### ■ 命令格式:

AWG:CHANnel<n>:BASE:OFFSet <voltage>

AWG:CHANnel<n>:BASE:OFFSet?

■ **功能描述:**

设置指定通道输出直流偏移。

<voltage>表示电压，单位 V。范围为：0~±当前负载下最大直流。

当前负载下的最大直流= 当前负载\*10/(50+当前负载) - 当前交流最小值/2;

交流最小值为 2mVpp,直流模式取 0;

<n>: 通道号，n 取值 1、2。

■ **返回格式:**

查询返回指定通道的输出直流偏移，采用科学计数法返回。

■ **举例:**

AWG:CHANnel1:BASE:OFFSet 2

设置通道 1 输出直流偏移为 2V

AWG:CHANnel1:BASE:OFFSet?

查询返回 2e+0

### AWG:CHANnel<n>:BASE:HIGh

■ **命令格式:**

AWG:CHANnel<n>:BASE:HIGh <voltage>

AWG:CHANnel<n>:BASE:HIGh?

■ **功能描述:**

设置指定通道信号输出高值。

<voltage>表示电压，单位当前通道指定单位。

<n>: 通道号，n 取值 1、2。

■ **返回格式:**

查询返回指定通道信号输出高值，采用科学计数法返回。

■ **举例:**

AWG:CHANnel1:BASE:HIGh 2

设置通道 1 信号输出高值为 2V

AWG:CHANnel1:BASE:HIGh?

查询返回 2e+0

### AWG:CHANnel<n>:BASE:LOW

■ **命令格式:**

AWG:CHANnel<n>:BASE:LOW <voltage>

AWG:CHANnel<n>:BASE:LOW?

■ **功能描述:**

设置指定通道信号输出低值。

<voltage>表示电压，单位当前通道指定单位。

<n>: 通道号，n 取值 1、2。

■ **返回格式:**

查询返回指定通道信号输出低值, 采用科学计数法返回。

■ **举例:**

AWG:CHANnel1:BASE:LOW 2

设置通道 1 信号输出低值为 2V

AWG:CHANnel1:BASE:LOW?

查询返回 2e+0

### AWG:CHANnel<n>:BASE:DUTY

■ **命令格式:**

AWG:CHANnel<n>:BASE:DUTY <duty>

AWG:CHANnel<n>:BASE:DUTY?

■ **功能描述:**

设置指定通道信号输出占空比。

<duty>表示占空比, 单位%, 范围 0~100。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道信号输出占空比。

■ **举例:**

AWG:CHANnel1:BASE:DUTY 20

设置通道 1 信号输出占空比为 20%

AWG:CHANnel1:BASE:DUTY?

查询返回 20

### AWG:CHANnel<n>:RAMP:SYMMetry

■ **命令格式:**

AWG:CHANnel<n>:RAMP:SYMMetry <symmetry>

AWG:CHANnel<n>:RAMP:SYMMetry?

■ **功能描述:**

设置指定通道斜波信号输出对称度。

< symmetry >表示对称度, 单位%, 范围 0~100。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道斜波信号输出对称度。

■ **举例:**

AWG:CHANnel1:RAMP:SYMMetry 20

设置通道 1 斜波信号对称度为 20%

AWG:CHANnel1:RAMP:SYMMetry?

查询返回 20

## AWG:CHANnel<n>:PULSe:RISe

### ■ 命令格式:

AWG:CHANnel<n>:PULSe:RISe <width>

AWG:CHANnel<n>:PULSe:RISe?

### ■ 功能描述:

设置指定通道信号脉冲波上升沿脉宽。

<width>表示脉宽，单位 S。

<n>: 通道号，n 取值 1、2。

### ■ 返回格式:

查询返回指定通道信号脉冲波上升沿脉宽，采用科学计数法返回。

### ■ 举例:

AWG:CHANnel1:PULSe:RISe 0.002                      设置通道 1 信号上升沿脉宽为 2ms

AWG:CHANnel1:PULSe:RISe?                      查询返回 2e-3

## AWG:CHANnel<n>:PULSe:FALL

### ■ 命令格式:

AWG:CHANnel<n>:PULSe:FALL <width>

AWG:CHANnel<n>:PULSe:FALL?

### ■ 功能描述:

设置指定通道信号脉冲波下降沿脉宽。

<width>表示脉宽，单位 S。

<n>: 通道号，n 取值 1、2。

### ■ 返回格式:

查询返回指定通道信号脉冲波下降沿脉宽，采用科学计数法返回。

### ■ 举例:

AWG:CHANnel1:PULSe:FALL 0.002                      设置通道 1 信号下降沿脉宽为 2ms

AWG:CHANnel1:PULSe:FALL?                      查询返回 2e-3

## AWG:CHANnel<n>:MODulate

### ■ 命令格式:

AWG:CHANnel<n>:MODulate { AM | FM }

AWG:CHANnel<n>:MODulate?

### ■ 功能描述:

设置指定通道调制类型，分别为 AM（调幅）、FM（调频）。

<n>: 通道号，n 取值 1、2。

■ **返回格式:**

查询返回指定通道调制类型。

■ **举例:**

AWG:CHANnel1:MODulate AM

设置通道 1 信号调制类型为调幅

AWG:CHANnel1:MODulate?

查询返回 AM

### AWG:CHANnel<n>:MODulate:WAVE

■ **命令格式:**

AWG:CHANnel<n>:MODulate:WAVE {SINE|SQUare|UPRamp|DNRamp|ARB|NOISe}

AWG:CHANnel<n>:MODulate:WAVE?

■ **功能描述:**

设置指定通道信号调制信号载波类型，分别为正弦波、方波、上三角、下三角、任意波、噪声。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道信号调制信号载波类型。

■ **举例:**

AWG:CHANnel1:MODulate:WAVE SINE

设置通道 1 信号调制载波类型为正弦波

AWG:CHANnel1:MODulate:WAVE?

查询返回 SINE

### AWG:CHANnel<n>:MODulate:FREQuency

■ **命令格式:**

AWG:CHANnel<n>:MODulate:FREQuency <freq>

AWG:CHANnel<n>:MODulate:FREQuency?

■ **功能描述:**

设置指定通道信号调制频率。

<freq>表示频率，单位 Hz。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道信号调制频率，返回采样科学计数法表示。

■ **举例:**

AWG:CHANnel1:MODulate:FREQuency 2000

设置通道 1 信号调制频率 2KHz

AWG:CHANnel1:MODulate:FREQuency?

查询返回 2e+3

### AWG:CHANnel<n>:AM:MODulate:DEPTh

■ **命令格式:**

AWG:CHANnel<n>:AM:MODulate:DEPTh <depth>

AWG:CHANnel<n>:AM:MODulate:DEPTh?

■ **功能描述:**

设置指定通道调幅 AM 调制深度。

<depth>表示调制深度，单位%。0% ~ 100%，其中 AM 调制深度为 0% ~ 120%



<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道调制深度。

■ **举例:**

AWG:CHANnel1:AM:MODulate:DEPTh 50	设置通道一调制深度为 50%
AWG:CHANnel1:AM:MODulate:DEPTh?	查询返回 50

### AWG:CHANnel<n>:FM:FREQuency:DEV

■ **命令格式:**

AWG:CHANnel<n>:FM:FREQuency:DEV <freq>  
AWG:CHANnel<n>:FM:FREQuency:DEV?

■ **功能描述:**

设置指定通道调频 FM 时的频率偏差。  
<freq>表示频率偏移, 单位 Hz。0Hz ~ 当前基波频率  
<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道频率偏移, 采样科学计数法返回数据。

■ **举例:**

AWG:CHANnel<n>:FM:FREQuency:DEV 2000	设置通道一频率偏移 2KHz
AWG:CHANnel<n>:FM:FREQuency:DEV?	查询返回 2e+3

### AWG:CHANnel<n>:ARB:SOURce

■ **命令格式:**

AWG:CHANnel<n>:ARB:SOURce { INTernal|EXTernal }  
AWG:CHANnel<n>:ARB:SOURce?

■ **功能描述:**

设置指定通道任意波源, 分别内部、外部两种。  
<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道任意波源。

■ **举例:**

AWG:CHANnel1:ARB:SOURce INTernal	设置通道一任意波源为内部
AWG:CHANnel1:ARB:SOURce?	查询返回 INTernal

### AWG:CHANnel<n>:ARB:INDex

■ **命令格式:**

AWG:CHANnel<n>:ARB:INDex <index>  
AWG:CHANnel<n>:ARB:INDex?

■ **功能描述:**

设置指定通道加载信号源存储的任意波序号。

<index >表示任意波序号。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道任意波序号。

■ **举例:**

AWG:CHANnel1:ARB:IND 2

设置通道 1 加载信号源存储的第二种任意波

AWG:CHANnel1:ARB:IND?

查询返回 2

## AWG:CHANnel<n>:ARB

■ **命令格式:**

AWG:CHANnel<n>:ARB <filename>

AWG:CHANnel<n>:ARB?

■ **功能描述:**

设置指定通道加载存储的任意波文件名称。

<n>: 通道号, n 取值 1、2。

<filename>表示文件名称, 文件名称必须是字符串类型数据, 需带双引号, 例如"test.bsv".

■ **返回格式:**

查询返回指定通道任意波文件名称。

■ **举例:**

AWG:CHANnel1:ARB "test.bsv"

设置通道 1 加载信号源存储的第二种任意波

AWG:CHANnel1:ARB?

查询返回 test.bsv

## AWG:CHANnel<n>:MODulate:ARB:SOURce

■ **命令格式:**

AWG:CHANnel<n>:ARB:SOURce { INTernal|EXTernal }

AWG:CHANnel<n>:ARB:SOURce?

■ **功能描述:**

设置指定通道调制任意波源, 分别内部、外部两种。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道调制任意波源。

■ **举例:**

AWG:CHANnel1:ARB:SOURce INTernal

设置通道一任意波源为内部

AWG:CHANnel1:ARB:SOURce?

查询返回 INTernal

## AWG:CHANnel<n>:MODulate:ARB:INDex

■ **命令格式:**

AWG:CHANnel<n>:ARB:INDex <index >

AWG:CHANnel<n>:ARB:INDex?

■ **功能描述:**

设置指定通道加载信号源存储的调制任意波序号。

<index >表示调制任意波序号。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道调制任意波序号。

■ **举例:**

AWG:CHANnel1:ARB:IND 2

设置通道 1 加载信号源存储的第二种任意波

AWG:CHANnel1:ARB:IND?

查询返回 2

## AWG:CHANnel<n>:MODulate:ARB

■ **命令格式:**

AWG:CHANnel<n>:ARB <filename>

AWG:CHANnel<n>:ARB?

■ **功能描述:**

设置指定通道加载存储的调制任意波文件名称。

<n>: 通道号, n 取值 1、2。

<filename>表示文件名称, 文件名称必须是字符串类型数据, 需带双引号, 例如"test.bsv".

■ **返回格式:**

查询返回指定通道调制任意波文件名称。

■ **举例:**

AWG:CHANnel1:ARB "test.bsv"

设置通道 1 加载信号源存储的第二种任意波

AWG:CHANnel1:ARB?

查询返回 test.bsv

## AWG:CHANnel<n>:ARB:PLAYmode

■ **命令格式:**

AWG:CHANnel<n>:ARB:PLAYmode {{1 | ON} | {0 | OFF}}

AWG:CHANnel<n>:ARB:PLAYmode?

■ **功能描述:**

设置指定通道任意波播放模式开关。

ON 表示以原始数据输出; OFF 表示可抽点输出。

<n>: 通道号, n 取值 1、2。

■ **返回格式:**

查询返回指定通道的任意波播放模式开关状态, 0 表示关闭, 1 表示打开。

■ **举例:**

AWG:CHANnel1:ARB:PLAYmode ON

设置通道一任意波以原始数据播放

AWG:CHANnel1:ARB:PLAYmode?

查询返回 1

## WARB 指令

用于写任意波形文件指令，包括基本任意波形和调制任意波形写配置。

### AWG:WARB<n>:MODulate

- **命令格式:**

AWG:WARB<n>:MODulate <arb file>

- **功能描述:**

用于写调制任意波形，波形数据最大 8k 个点，先发送该指令，然后发送任意波形文件数据到信号源。  
<arb file>表示任意波形文件，此指令只支持.bsv 格式数据。

- **举例:**

AWG:WARB1:MODulate                    写通道一调制任意波形文件

### AWG:WARB<n>:CARRier

- **命令格式:**

AWG:WARB<n>:CARRier <arb file>

- **功能描述:**

用于写基波任意波形，波形数据最大 8k 个点，先发送该指令，然后发送任意波形文件数据到信号源。  
<arb file>表示任意波形文件，此指令只支持.bsv 格式数据。

- **举例:**

AWG:WARB1:CARRier                    写通道一基波任意波形文件

## LA 命令

命令用于对数字通道进行相关操作。

### :LA:STATe

- **命令格式:**

:LA:STATe { {1|ON} | {0|OFF} }

:LA:STATe?

- **功能描述:**

打开或关闭 LA 功能，或查询 LA 功能的狀態。

- **返回格式:**

查詢返回 1 或 0，分別代表 ON 或 OFF。

- **举例:**

:LA:STATe ON                    打开 LA 功能  
:LA:STATe?                    查询返回 1

#### **:LA:ACTive**

■ **命令格式:**

:LA:ACTive {<Dx> | CHANnel1 | CHANnel2 | OFF}  
:LA:ACTive?

■ **功能描述:**

设置或查询当前的活动通道或通道组。

<Dx>: {D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。

■ **返回格式:**

查询返回当前活动通道。

■ **举例:**

:LA:ACTive D7                    设置当前活动通道为 D3  
:LA:ACTive?                    查询返回 D3

#### **:LA:AUTOsort**

■ **命令格式:**

:LA:AUTOsort { {1|ON} | {0|OFF} }  
:LA:AUTOsort?

■ **功能描述:**

设置屏幕中已打开通道波形的自动排列方式。

OFF: 屏幕中的波形从上至下依次为 D0 至 D15;

ON: 屏幕中的波形从上至下依次为 D15 至 D0。

■ **返回格式:**

查询返回 1 或 0。

■ **举例:**

:LA:AUTOsort ON                    屏幕中的波形从上至下依次为 D15 至 D0  
:LA:AUTOsort?                    查询返回 1

#### **:LA:DIGital<n>:DISPlay**

■ **命令格式:**

:LA:DIGital<n>:DISPlay { {1|ON} | {0|OFF} }  
:LA:DIGital<n>:DISPlay?

■ **功能描述:**

打开或关闭指定的数字通道，或查询指定数字通道的状态。

<n>: 整型数字 1~16, 表示{D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。  
16 个数字通道。

■ **返回格式:**

查询返回 1 或 0。

■ **举例:**

:LA:DIgital4:DISPlay ON            打开 D3  
:LA:DIgital4:DISPlay?            查询返回 1

**:LA:DIgital<n>:POSition**

■ **命令格式:**

:LA:DIgital<n>:POSition <position>  
:LA:DIgital<n>:POSition?

■ **功能描述:**

设置或查询指定数字通道波形在屏幕中的显示位置, 仅当当前已打开指定数字通道时, 设置命令有效。

<n>: 整型数字 1~16, 表示{D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。

<position>: 整型。

波形显示方式为小: 0 至 31

波形显示方式为中: 0 至 15

波形显示方式为大: 0 至 7

■ **返回格式:**

查询返回 0 至 31、0 至 15 或 0 至 7 之间的整数。

■ **举例:**

:LA:DIgital3:POSition 6            设置 D4 的显示位置为 6  
:LA:DIgital3:POSition?            查询返回 6

**:LA:DIgital<n>:LABel**

■ **命令格式:**

:LA:DIgital<n>:LABel <label>  
:LA:DIgital<n>:LABel?

■ **功能描述:**

设置或查询指定数字通道的标签。

<n>: 整型数字 1~16, 表示{D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。

<label>: ASCII 字符串, 包含英文字母和数字, 也可包含部分符号。

■ **返回格式:**

查询返回以 ASCII 字符串形式返回指定数字通道的标签。

■ **举例:**

:LA:DIgital3:LABel "ACK"                    设置 D4 的标签为 ACK  
:LA:DIgital3:LABel?                        查询返回 ACK

### **:LA:POD<n>:DISPlay**

■ **命令格式:**

:LA:POD<n>:DISPlay { {1|ON} | {0|OFF} }  
:LA:POD<n>:DISPlay?

■ **功能描述:**

打开或关闭指定的默认通道组，或查询指定默认通道组的状态。

<n>: 整型数字 1~2, 1 表示 (D0 至 D7) 和 2 表示 (D8 至 D15)。

■ **返回格式:**

查询返回 1 或 0。

■ **举例:**

:LA:POD1:DISPlay ON                    打开 POD1 (D0 至 D7)  
:LA:POD1:DISPlay?                        查询返回 1

### **:LA:POD<n>:THReshold**

■ **命令格式:**

:LA:POD<n>:THReshold {<threshold>|TTL|ECL|PECL|CLDS|0V|5.0V|3.3V|2.5V|1.8V}  
:LA:POD<n>:THReshold?

■ **功能描述:**

设置或查询指定默认通道组的阈值，默认单位为 V。

<n>: 整型数字 1~2, 1 表示 (D0 至 D7) 和 2 表示 (D8 至 D15)。

<threshold>: 实型, -20.0V 至+20.0V。

■ **返回格式:**

查询返回以科学计数形式返回指定通道组当前的阈值。

■ **举例:**

:LA:POD1:THReshold 1.4                    设置 POD1 (D0 至 D7) 的阈值为 1.4V  
:LA:POD1:THReshold?                        查询返回 1.400000E0

### **:LA:SIze**

■ **命令格式:**

:LA:SIze {SMAL|LARGe|MEDIum}  
:LA:SIze?

■ **功能描述:**

设置或查询已打开通道的波形在屏幕中显示的大小。

- **返回格式：**

查询返回 SMALl、LARGe 或 MEDium。

- **举例：**

:LA:SIZE SMALl

设置波形显示方式为小

:LA:SIZE?

查询返回 SMALl

## :LA:TCALibrate

- **命令格式：**

:LA:TCALibrate <tcAl>

:LA:TCALibrate?

- **功能描述：**

设置或查询数字通道的延时校正时间，默认单位为 s。

<tcAl>：实型，-100ns 至 100ns。

使用示波器进行实际测量时，探头电缆的传输延迟可能带来较大的误差（零点偏移）。零点偏移定义为波形与阈值电平线的交点相对于触发位置的偏移量。用户可以通过设定一个延迟时间以校正对应通道的零点偏移。

- **返回格式：**

查询以科学计数形式返回延时校正时间。

- **举例：**

:LA:TCALibrate 20ns

设置延时校正时间为 20ns

:LA:TCALibrate?

查询返回 2.000000E-8

## :LA:DELete

- **命令格式：**

:LA:DELete {GROup1|GROup2|GROup3|GROup4}

:LA:DELete?

- **功能描述：**

取消 16 个数字通道中任一通道的分组设置，或者取消 GROup1-GROup4 中任一通道组的通道设置，该命令仅可对已进行分组设置的数字通道或自定义通道组执行取消分组设置操作。

- **返回格式：**

查询返回{GROup1|GROup2|GROup3|GROup4}。

- **举例：**

:LA:DELete GROup1

取消 GROup1 通道组的通道设置

:LA:DELete?

查询返回 GROup1



## :LA:GROup<n>:DISPlay

### ■ 命令格式:

:LA:GROup<n>:DISPlay { {1|ON} | {0|OFF} }

:LA:GROup<n>:DISPlay?

### ■ 功能描述:

打开或关闭指定的数字通道组。

<n>: 整型数字 1~4, 分别表示 GROup1、GROup2、GROup3、GROup4。

### ■ 返回格式:

查询返回 1 或 0。

### ■ 举例:

:LA:GROup1:DISPlay ON                      打开数字通道组 1

:LA:GROup1:DISPlay?                      查询返回 1

## :LA:GROup<n>:APPend

### ■ 命令格式:

:LA:GROup<n>:APPend <digital0>[, digital1~digital15]

### ■ 功能描述:

为指定的自定义分组添加通道。

<n>: 整型数字 1~4, 分别表示 GROup1、GROup2、GROup3、GROup4。

<digital>: 表示 {D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。

### ■ 举例:

:LA:GROup1:APPend D0,D3                      为分组 1 添加通道 D0 和 D3

## LA 总线

命令用于对 LA 总线数据和并行解码的相关设置, 仪器需要具备此功能指令才能生效。

## :LA:BUS<n>:DISPlay

### ■ 命令格式:

:LA:BUS<n>:DISPlay { {1|ON} | {0|OFF} }

:LA:BUS<n>:DISPlay?

### ■ 功能描述:

用于设置示波器的 LA 总线状态 ON (打开) 或 OFF (关闭)。

<n>: 整型数据 {1|2}, 分别表示总线 1~2。

### ■ 返回格式:

查询返回 1 或 0, 分别代表 ON 或 OFF。

■ **举例：**

:LA:BUS1:DISPlay ON                                    打开 LA 总线 1 解码状态，显示解码波形  
:LA:BUS1:DISPlay?                                    查询返回 1

**:LA:BUS<n>:BASE**

■ **命令格式：**

:LA:BUS<n>:BASE {ASCII | BINary | HEX | DEC| WAV}  
:LA:BUS<n>:BASE?

■ **功能描述：**

用于设置示波器的 LA 总线显示格式。  
<n>： 整型数据{1|2}，分别表示总线 1~2。

■ **返回格式：**

查询返回 {ASCII | BINary | HEX | DEC | WAV }。

■ **举例：**

:LA:BUS1:BASE BIN                                    设置 LA 总线 1 解码为二进制显示格式  
:LA:BUS1:BASE?                                    查询返回 BINary

**:LA:BUS<n>:EVENT**

■ **命令格式：**

:LA:BUS<n>:EVENT { {1|ON} | {0|OFF} }  
:LA:BUS<n>:EVENT?

■ **功能描述：**

用于设置示波器的 LA 总线事件列表 ON（打开）或 OFF（关闭）。  
<n>： 整型数据{1|2}，分别表示总线 1~2。

■ **返回格式：**

查询返回 1 或 0，分别代表 ON 或 OFF。

■ **举例：**

:LA:BUS1:EVENT ON                                    打开 LA 总线 1 事件列表  
:LA:BUS1:EVENT?                                    查询返回 1

**:LA:BUS<n>:DATA?**

■ **命令格式：**

:LA:BUS<n>:DATA?

■ **功能描述：**

用于读取示波器的 LA 解码事件表数据。

■ **返回格式：**

查询返回 LA 解码事件表数据，返回的数据符合附录 2：IEEE 488.2 二进制数据格式。

■ **举例：**

:LA:BUS1:DATA? 查询返回数据总线 1 的并行解码事件列表数据：

```
#9000000089PAR,  
ID,BUS1DATA,  
0,0xFF  
1,0xFE  
2,0xF3  
3,0xF5  
4,0xF6
```

PAR 表示并行解码，后面紧跟 CSV 格式的事件表数据，事件表数据的具体格式由不同设备自动适配，数据以逗号分开，会根据解码列表的行自动换行，数据值与设定的进制显示有关。

**:LA:BUS<n>:SLOPe**

■ **命令格式：**

```
:LA:BUS<n>:SLOPe {POSitive|NEGative}  
:LA:BUS<n>:SLOPe?
```

■ **功能描述：**

用于设置示波器的 LA 总线对数据通道进行采样时钟通道的边沿类型。

<n>：整型数据{1|2}，分别表示总线 1~2。

■ **返回格式：**

查询返回{POSitive|NEGative}。

■ **举例：**

```
:LA:BUS1:SLOPe POSitive      设置 LA 总线 1 时钟源边沿为上升沿  
:LA:BUS1:SLOPe?             查询返回 POSitive
```

**:LA:BUS<n>:CLK**

■ **命令格式：**

```
:LA:BUS<n>:CLK {<Dx> | CHANnel1 | CHANnel2 | OFF}  
:LA:BUS<n>:CLK?
```

■ **功能描述：**

用于设置示波器的 LA 总线时钟源。

<n>：整型数据{1|2}，分别表示总线 1~2。

<Dx>：{D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。

■ **返回格式：**

查询返回{<Dx> | CHANnel1 | CHANnel2 | OFF}。

■ **举例：**

:LA:BUS1:CLK D0                                    设置 LA 总线 1 时钟源为 D0 通道

:LA:BUS1:CLK?                                      查询返回 D0

### :LA:BUS<n>:POLarity

■ **命令格式：**

:LA:BUS<n>:POLarity {NEGative|POSitive}

:LA:BUS<n>:POLarity?

■ **功能描述：**

用于设置示波器的 LA 总线位序，即数据极性。

<n>： 整型数据{1|2}， 分别表示总线 1~2。

NEGative 表示位序从高到低， POSitive 表示位序从低到高。

■ **返回格式：**

查询返回{NEGative|POSitive}。

■ **举例：**

:LA:BUS1:POLarity POSitive                    设置 LA 总线 1 数据位序从低到高

:LA:BUS1:POLarity?                            查询返回 POSitive

### :LA:BUS<n>:NREJect

■ **命令格式：**

:LA:BUS<n>:NREJect { {1|ON} | {0|OFF} }

:LA:BUS<n>:NREJect?

■ **功能描述：**

用于打开或关闭示波器的 LA 总线并行解码噪声抑制功能。

<n>： 整型数据{1|2}， 分别表示总线 1~2。

■ **返回格式：**

查询返回 1 或 0， 分别代表 ON 或 OFF。

■ **举例：**

:LA:BUS1:NREJect ON                            打开 LA 总线 1 并行解码噪声抑制功能

:LA:BUS1:NREJect?                            查询返回 1

### :LA:BUS<n>:NRTime

■ **命令格式：**

:LA:BUS<n>:NRTime <time>

:LA:BUS<n>:NRTime?

■ **功能描述:**

用于示波器的 LA 总线并行解码噪声抑制时间，默认单位为 s。

<n>: 整型数据{1|2}，分别表示总线 1~2。

■ **返回格式:**

查询返回噪声抑制时间，单位为 s。

■ **举例:**

:LA:BUS1:NRTime 50us                    设置 LA 总线并行解码噪声抑制时间 50us

:LA:BUS1:NRTime?                        查询返回 5.000e-005

**:LA:BUS<n>:SOURce**

■ **命令格式:**

:LA:BUS<n>:SOURce <digital0>[, digital1~ digital15, CHANnel1, CHANnel2]

:LA:BUS<n>:SOURce?

■ **功能描述:**

用于设置示波器的 LA 总线并行解码当前选中的通道源。

<n>: 整型数据{1|2}，分别表示总线 1~2。

<digital>: 表示{D0|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|D15}。

■ **返回格式:**

查询返回当前选中的通道源。

■ **举例:**

:LA:BUS1:SOURce D0,D3,CHANnel1            LA 总线 1 通道源选中 D0、D3 和通道一

:LA:BUS1:SOURce?                        查询返回 D0,D3,CHANnel1

## BODe 命令

命令用于对波特图功能的相关设置，仪器需要具备此功能指令才能生效。

**:BODe:APPLy**

■ **命令格式:**

:BODe:APPLy {{1 | ON} | {0 | OFF}}

:BODe:APPLy?

■ **功能描述:**

用于设置示波器波特图使能或关闭。

■ **返回格式:**

查询返回全键盘锁定状态，0 表示未打开，1 表示打开。

■ 举例：

:BODe:APPLy ON

打开波特图功能

:BODe:APPLy?

查询返回 1，表示打开

**:BODe:EVENT**

■ 命令格式：

:BODe:EVENTt {{1 | ON} | {0 | OFF}}

:BODe:EVENT?

■ 功能描述：

用于设置示波器波特图事件列表打开或关闭。

■ 返回格式：

查询返回波特图事件列表状态，0 表示未打开，1 表示打开。

■ 举例：

:BODe:EVENT ON

打开波特图事件列表功能

:BODe:EVENT?

查询返回 1，表示打开

**:BODe:EVENT:DATA?**

■ 命令格式：

:BODe:DATA?

■ 功能描述：

用于查询波特图事件列表数据。

■ 返回格式：

查询返回波特图事件表数据，返回的数据符合 [附录 2：IEEE 488.2 二进制数据格式](#)。

■ 举例：

:BODe:EVENT:DATA? 查询返回波特图事件列表数据：

#9000000727BODE,

index,Freq,AMP(Vpp),Gain(dB),Phase( $\hat{A}^\circ$ ),

1,3.700000e+03,3.000000e+00,-1.039986e+02,8.887821e+01,

2,8.784684e+03,3.000000e+00,-9.819112e+01,-7.619118e+01,

3,2.085694e+04,3.000000e+00,-9.908194e+01,-6.782645e+01,

**:BODe:SWEep:MODE**

■ 命令格式：

:BODe:SWEep:MODE { SINGle | CONTInue}

:BODe:SWEep:MODE?

■ 功能描述：

用于设置示波器波特图的扫描方式，SINGle 表示单次；CONTInue 表示连续。

■ **返回格式：**

查询返回{ SINGle | CONTInue}。

■ **举例：**

:BODe:SWEEp:MODE SINGle	设置波特图扫描方式为单次扫描
:BODe:SWEEp:MODE?	查询返回 SINGle

**:BODe:SWEEp:POINts**

■ **命令格式：**

:BODe:SWEEp:POINts <points>  
:BODe:SWEEp:POINts?

■ **功能描述：**

用于设置示波器波特图扫描点数，<points>取值范围 1~1000。

■ **返回格式：**

查询返回扫描点数。

■ **举例：**

:BODe:SWEEp:POINts 1000	设置示波器波特图扫描点数为 1000
:BODe:SWEEp:POINts?	查询返回 1000

**:BODe:SWEEp:FREQuency:STARt**

■ **命令格式：**

:BODe:SWEEp:FREQuency:STARt <freq>  
:BODe:SWEEp:FREQuency:STARt?

■ **功能描述：**

设置示波器波特图扫描起始频率。

■ **返回格式：**

查询返回示波器波特图扫描起始频率，单位为 Hz。

■ **举例：**

:BODe:SWEEp:FREQuency:STARt 1KHz	设置起始频率为 1KHz
:BODe:SWEEp:FREQuency:STARt?	查询返回 1.000e003

**:BODe:SWEEp:FREQuency:END**

■ **命令格式：**

:BODe:SWEEp:FREQuency:END <freq>  
:BODe:SWEEp:FREQuency:END?

■ **功能描述：**

设置示波器波特图扫描截止频率。

■ **返回格式:**

查询返回示波器波特图扫描截止频率, 单位为 Hz。

■ **举例:**

:BODE:SWEEP:FREQUENCY:END 1KHz	设置截止频率为 1KHz
:BODE:SWEEP:FREQUENCY:END?	查询返回 1.000e003

**:BODE:AMPLITUDE:MODE**

■ **命令格式:**

:BODE:AMPLITUDE:MODE { FIXEd | VARiable }  
:BODE:AMPLITUDE:MODE?

■ **功能描述:**

用于设置示波器波特图幅度模式, FIXEd 表示固定; VARiable 表示可变。

■ **返回格式:**

查询返回 { FIXEd | VARiable }。

■ **举例:**

:BODE:AMPLITUDE:MODE FIXEd	设置波特图幅度模式为固定幅度
:BODE:AMPLITUDE:MODE?	查询返回 FIXEd

**:BODE:AMPLITUDE<n>**

■ **命令格式:**

:BODE:AMPLITUDE<n> <amp>  
:BODE:AMPLITUDE<n>?

■ **功能描述:**

设置示波器波特图幅度。

<n>: 取值范围 1~8; <amp>: 取值范围 10mV~3V。

■ **返回格式:**

查询返回 1.000e-003, 单位为 V。

■ **举例:**

:BODE:AMPLITUDE1 10mV	设置示波器波特图幅度 1 为 10mV
:BODE:AMPLITUDE1?	查询返回 1.000e-003

**:BODE:SOURCE:OFFSET**

■ **命令格式:**

:BODE:SOURCE:OFFSET <offset>  
:BODE:SOURCE:OFFSET?



■ **功能描述:**

设置示波器波特图信源直流偏置。

<offset>: 取值范围 0V~1V。

■ **返回格式:**

查询返回示波器波特图信源直流偏置, 单位为 V。

■ **举例:**

:BODe:SOURce:OFFSet 10mV

设置示波器波特图信源直流偏置为 10mV

:BODe:SOURce:OFFSet?

查询返回 1.000e-003

### **:BODe:SOURce:LOAD**

■ **命令格式:**

:BODe:SOURce:LOAD {50OHM | HIGH}

:BODe:SOURce:LOAD?

■ **功能描述:**

设置示波器波特图信源阻抗。

50OHM 表示阻抗 50Ω; HIGH 表示高阻抗。

■ **返回格式:**

查询返回{50OHM | HIGH}。

■ **举例:**

:BODe:SOURce:LOAD HIGH

设置示波器波特图信源阻抗为高阻抗

:BODe:SOURce:LOAD?

查询返回 HIGH

### **:BODe:DUT:INPut**

■ **命令格式:**

:BODe:DUT:INPut {CHANnel1| CHANnel2| CHANnel3| CHANnel4}

:BODe:DUT:INPut?

■ **功能描述:**

设置示波器波特图 DUT 输入通道。

■ **返回格式:**

查询返回{CHANnel1| CHANnel2| CHANnel3| CHANnel4}。

■ **举例:**

:BODe:DUT:INPut CHANnel1

设置示波器波特图 DUT 输入通道为通道一

:BODe:DUT:INPut?

查询返回 CHANnel1

### **:BODe:DUT:OUTPut**

■ **命令格式:**

:BODe:DUT:OUTPut {CHANnel1| CHANnel2| CHANnel3| CHANnel4}

:BODe:DUT:OUTPut?

■ **功能描述:**

设置示波器波特图 DUT 输出通道。

■ **返回格式:**

查询返回{CHANnel1| CHANnel2| CHANnel3| CHANnel4}。

■ **举例:**

:BODe:DUT:OUTPut CHANnel1

设置示波器波特图 DUT 输出通道为通道一

:BODe:DUT:OUTPut?

查询返回 CHANnel1

### :BODe:GAIN:DISPlay

■ **命令格式:**

:BODe:GAIN:DISPlay {{1 | ON} | {0 | OFF}}

:BODe:GAIN:DISPlay?

■ **功能描述:**

用于设置示波器波特图增益显示功能打开或关闭。

■ **返回格式:**

查询返回波特图增益显示功能状态，0 表示未打开，1 表示打开。

■ **举例:**

:BODe:GAIN:DISPlay ON

打开波特图事件增益显示功能

:BODe:GAIN:DISPlay?

查询返回 1，表示打开

### :BODe:GAIN:SCALE

■ **命令格式:**

:BODe:GAIN:SCALE <scale>

:BODe:GAIN:SCALE?

■ **功能描述:**

设置示波器波特图增益挡位。

<scale>: 取值范围 1~500dB。

■ **返回格式:**

查询返回示波器波特图增益挡位，单位为 dB。

■ **举例:**

:BODe:GAIN:SCALE 10

设置示波器波特图增益挡位为 10dB

:BODe:GAIN:SCALE?

查询返回 1.000e001

### :BODe:GAIN:OFFSet

■ **命令格式:**

:BODe:GAIN:OFFSet <offset>

:BODe:GAIN:OFFSet?

■ **功能描述:**

设置示波器波特图增益偏置。

<offset>: 取值范围-250dB~250dB。

■ **返回格式:**

查询返回示波器波特图增益偏置, 单位为 dB。

■ **举例:**

:BODe:GAIN:OFFSet 10

设置示波器波特图增益偏置为 10dB

:BODe:GAIN:OFFSet?

查询返回 1.000e001

### :BODe:PHASe:DISPlay

■ **命令格式:**

:BODe:PHASe:DISPlay {{1 | ON} | {0 | OFF}}

:BODe:PHASe:DISPlay?

■ **功能描述:**

用于设置示波器波特图相位显示功能打开或关闭。

■ **返回格式:**

查询返回波特图相位显示功能状态, 0 表示未打开, 1 表示打开。

■ **举例:**

:BODe:PHASe:DISPlay ON

打开波特图事件相位显示功能

:BODe:PHASe:DISPlay?

查询返回 1, 表示打开

### :BODe:PHASe:SCALE

■ **命令格式:**

:BODe:PHASe:SCALE <scale>

:BODe:PHASe:SCALE?

■ **功能描述:**

设置示波器波特图相位挡位。

<scale>: 取值范围 1°~180°。

■ **返回格式:**

查询返回示波器波特图相位挡位, 单位为 °。

■ **举例:**

:BODe:PHASe:SCALE 10

设置示波器波特图相位挡位为 10°

:BODe:PHASe:SCALE?

查询返回 1.000e001

### :BODe:PHASe:OFFSet

■ **命令格式:**

:BODe:PHASe:OFFSet <offset>

:BODe:PHASe:OFFSet?

■ **功能描述:**

设置示波器波特图相位偏置。

<offset>: 取值范围-180°~180°。

■ **返回格式:**

查询返回示波器波特图相位偏置, 单位为 °。

■ **举例:**

:BODE:PHASe:OFFSet 10

设置示波器波特图相位偏置为 10°

:BODE:PHASe:OFFSet?

查询返回 1.000e001

**:BODE:DISPlay:AUTO**

■ **命令格式:**

:BODE:DISPlay:AUTO

■ **功能描述:**

设置示波器波特图自动显示。

■ **举例:**

:BODE:DISPlay:AUTO

示波器波特图自动显示

**:BODE:FREQuency:STARt**

■ **命令格式:**

:BODE:FREQuency:STARt <freq>

:BODE:FREQuency:STARt?

■ **功能描述:**

设置示波器波特图显示的起始频率。

■ **返回格式:**

查询返回示波器波特图扫描起始频率, 单位为 Hz。

■ **举例:**

:BODE:FREQuency:STARt 1KHz

设置波特图显示起始频率为 1KHz

:BODE:FREQuency:STARt?

查询返回 1.000e003

**:BODE:FREQuency:END**

■ **命令格式:**

:BODE:SWEep:FREQuency:END <freq>

:BODE:SWEep:FREQuency:END?

■ **功能描述:**

设置示波器波特图显示的截止频率。

■ **返回格式:**

查询返回示波器波特图显示的截止频率, 单位为 Hz。

■ **举例:**

:BODE:FREQuency:END 1KHz

设置波特图显示截止频率为 1KHz

:BODE:FREQuency:END?

查询返回 1.000e003

## :BODE:CURSor

### ■ 命令格式:

:BODE:CURSor {{1 | ON} | {0 | OFF}}

:BODE:CURSor?

### ■ 功能描述:

用于设置示波器波特图光标功能打开或关闭。

### ■ 返回格式:

查询返回波特图光标功能状态，0 表示未打开，1 表示打开。

### ■ 举例:

:BODE:CURSor ON

打开波特图光标功能

:BODE:CURSor?

查询返回 1，表示打开

## :BODE:CURSor:MODE

### ■ 命令格式:

:BODE:CURSor:MODE { INDePendent | TRACKing }

:BODE:CURSor:MODE?

### ■ 功能描述:

用于设置示波器波特图光标模式。

INDePendent 表示独立；TRACKing 表示跟踪。

### ■ 返回格式:

查询返回示波器波特图光标模式。

### ■ 举例:

:BODE:CURSor:MODE TRACKing

设置波特图光标为跟踪模式

:BODE:CURSor:MODE?

查询返回 1，表示打开

## 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 接口与 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

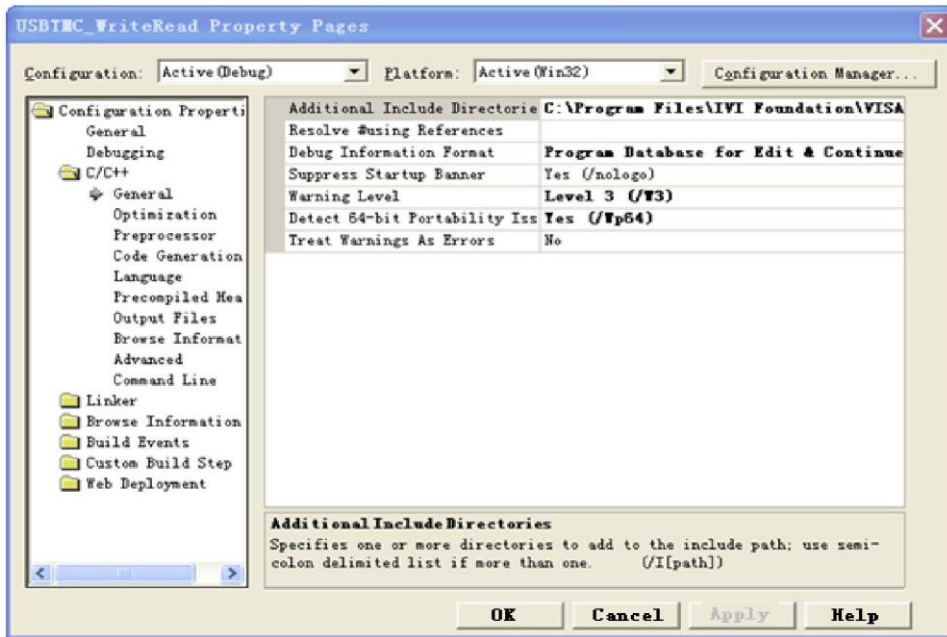
- 环境：Window 系统, Visual Studio。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"\*IDN?"命令来查询设备信息。
- 步骤：
  1. 打开 Visual Studio 软件，新建一个 VC++ win32 console project。
  2. 设置调用 NI-VISA 库的项目环境，分别为静态库和动态库。
    - a) 静态库：

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：

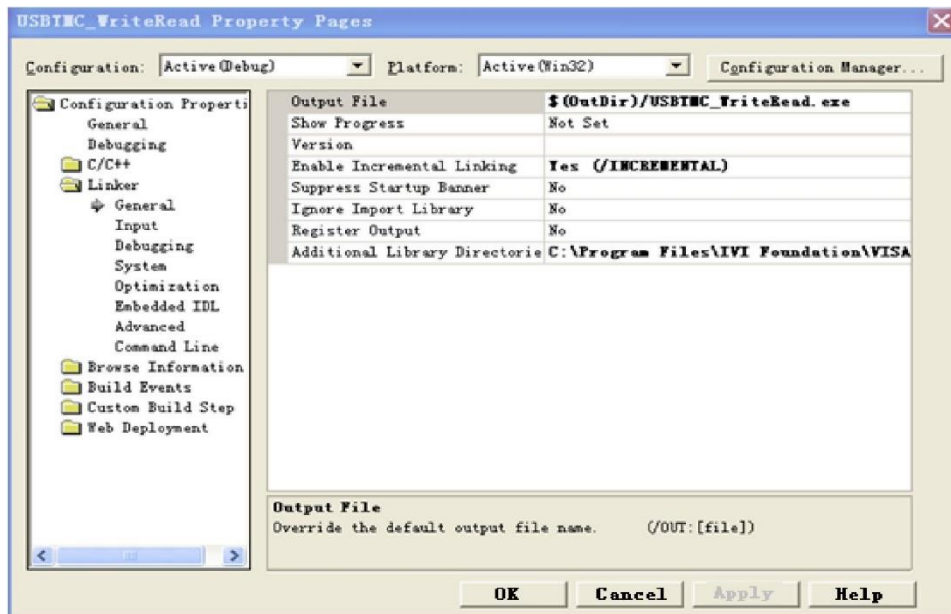
```
#include "visa.h"  
#pragma comment(lib,"visa32.lib")
```

- b) 动态库：

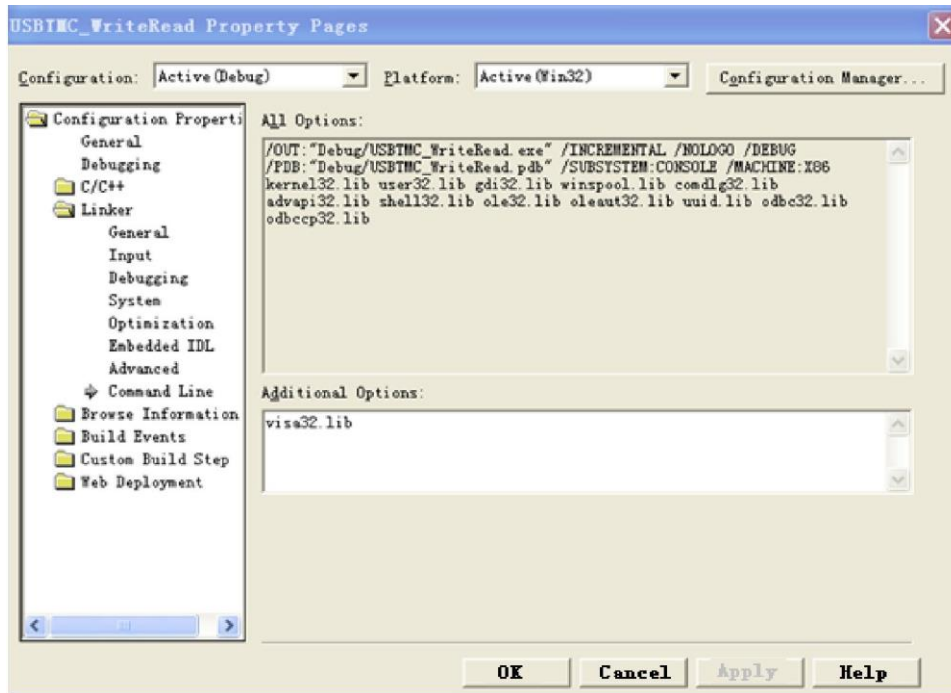
点击"project>>properties"，在属性对话框左侧选择"c/c++---General"中，将 "Additional Include Directories"项的值设置为 NI-VISA 的安装路径，(例如：C:\ProgramFiles\IVI Foundation\VISA\WinNT\include),如下图所示：



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, "*idn?\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();
    }
}
```

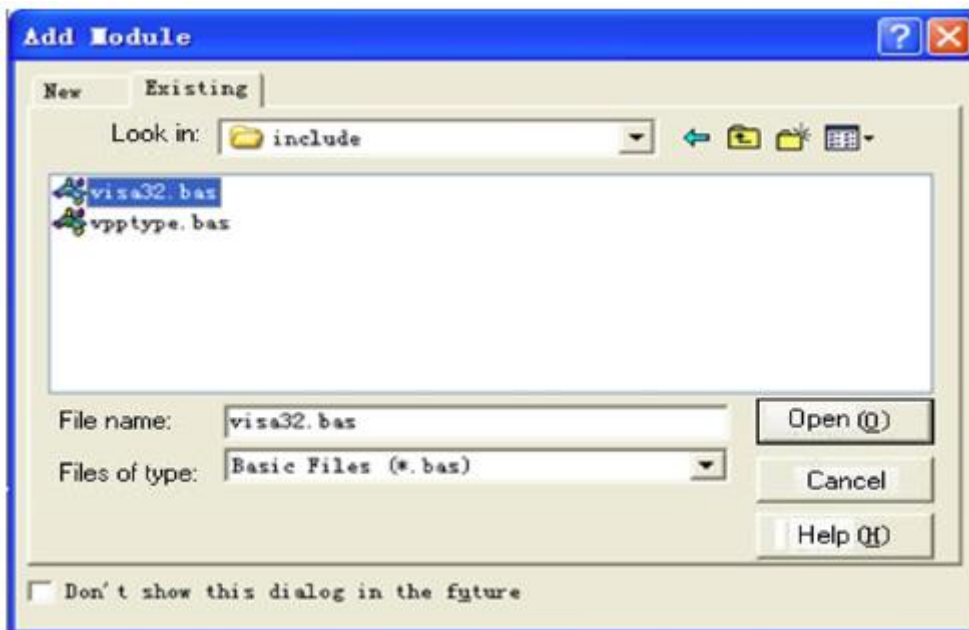
- b) TCP/IP 示例

```
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

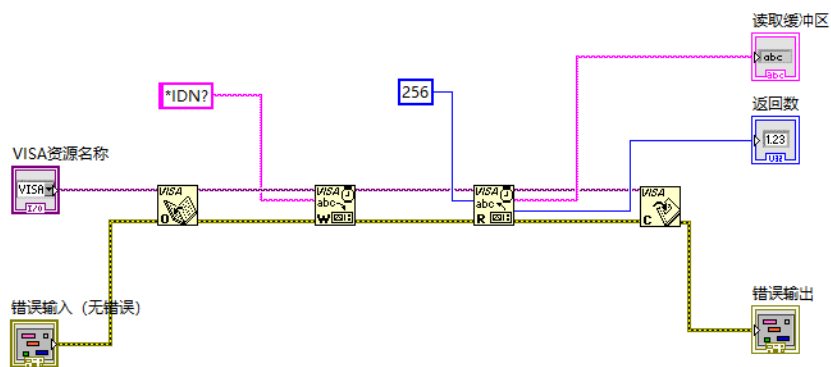
```
PrivateFunction tcp_ip_test(ByVal ip AsString) AsLong
Dim outputBuffer AsString * VI_FIND_BUFLen
Dim defaultRM AsLong
Dim instrsesn AsLong
Dim status AsLong
Dim count AsLong

' 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

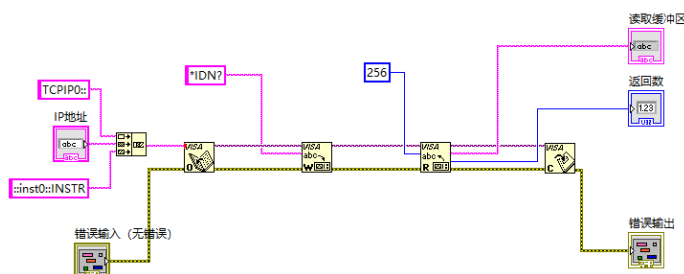
' Now we will open a session via TCP/IP device
status = viOpen(defaultRM, "TCPIP0::" + ip + "::inst0::INSTR", VI_LOAD_CONFIG, VI_NULL, instrsesn)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "An error occurred opening the session"
    viClose(defaultRM)
    tcp_ip_test = status
ExitFunction
EndIf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

## 设置带宽限制

当在观察低频信号时需要减少信号中的高频噪声，则衰减信号中 20MHz 以上的高频信号。可使用如下命令设置带宽限制，如设置通道 1：

```
CHANnel1:BWLimit ON      #打开通道 1 的带宽限制。
CHANnel1:BWLimit?       #查询返回 1，表示已经打开通道 1 的带宽限制。
```

## 设置偏置电压

设置通道偏置电压，可直接通过如下命令进行设置，如设置通道 1 的偏置电压值：

```
CHANnel1:OFFSet 1V      #通道 1 向上移动 1V，即偏置电压为：1V。
CHANnel1:OFFSet?       #查询通道的偏置电压值。
```

## 设置伏格档位

设置通道伏格档位，可直接通过如下命令进行设置，如设置通道 1 的伏格档位。

```
CHANnel1:SCALE 500mV   #设置通道 1 的伏格档位为：500mV。
CHANnel1:SCALE?       #查询通道 1 的伏格档位值。
```

## 设置时基档位

设置示波器的时基档位，可直接通过如下命令进行设置。

```
TIMebase:SCALE 0.005   #设置示波器的时基档位为：5ms。
TIMebase:SCALE?       #查询示波器的时基档位。
```

## 查询幅度值

当需要查询幅度测量结果时，可直接通过如下命令进行查询，而不需要打开测量窗口。  
如查询通道 1 波形的幅度值。

MEASure:VPP? CHANnel1 #查询通道 1 波形的幅度值。

## 查询上升延迟时间值

当需要查询上升延迟测量时间时，可直接通过如下命令进行查询，而不需要打开测量窗口。如查询通道 1 与通道 2 的上升延迟值。

MEASure:PDElay? CHANnel1,CHANnel2 #查询通道 1 与通道 2 上升延迟时间值。

## 附录 1：按键列表

按键	功能描述	LED 灯
CH1	通道一开关	√
CH2	通道二开关	√
CH3	通道三开关	√
CH4	通道四开关	√
AUTO	自动设置示波器的各项控制值,以显示适宜观察的波形	
RS	控制示波器的运行状态,连续发送该命令,示波器将在停止和运行状态切换	√
TMENu	触发菜单	
SINGle	单次触发	√
TFORe	强制触发	
HELP	帮助系统	
HMENu	水平系统菜单	
DISPlay	显示菜单	
MATH	数学运算功能及其菜单	√
REF	参考波形功能及其菜单	√
F1	选择当前菜单的第一个菜单项	
F2	选择当前菜单的第二个菜单项	
F3	选择当前菜单的第三个菜单项	
F4	选择当前菜单的第四个菜单项	
F5	选择当前菜单的第五个菜单项	
MENu	菜单显示功能开关	
PSCReen	一键打印或者一键保存屏幕图像	
MEASure	测量功能	
CURSor	光标测量功能及其菜单	
ACQuire	采样菜单	
STORage	存储菜单	

UTILity	系统辅助菜单	
CLEar	清除示波器屏幕波形	
DECode	解码菜单	
DEFault	恢复默认设置	
FKNob	多功能旋钮	
FKNLeft	多功能旋钮左旋	
FKNRight	多功能旋钮右旋	
VPKNob	垂直位置旋钮	
VPKNLeft	垂直位置旋钮左旋	
VPKNRight	垂直位置旋钮右旋	

HPKNob	水平位置旋钮	
HPKNLeft	水平位置旋钮左旋	
HPKNRight	水平位置旋钮右旋	
TPKNob	触发位置旋钮	
TPKNLeft	触发位置旋钮左旋	
TPKNRight	触发位置旋钮右旋	
VBKNob	电压基准旋钮	
VBKNLeft	电压基准旋钮左旋	
VBKNRight	电压基准旋钮右旋	
TBKNob	时间基准旋钮	
TBKNLeft	时间基准旋钮左旋	
TBKNRight	时间基准旋钮右旋	
NUM0	数字键 0	
NUM1	数字键 1	
NUM2	数字键 2	
NUM3	数字键 3	
NUM4	数字键 4	
NUM5	数字键 5	
NUM6	数字键 6	
NUM7	数字键 7	
NUM8	数字键 8	
NUM9	数字键 9	
DOT	数字键小数点	
SYMBol	数字键符号	
BACKspace	删除键	
ENTER	确认键	
LA	LA 模块按键	√
AWG	信号源模块按键	√
AUTO 状态灯	无按键,只有灯	√

Normal 状态灯	无按键,只有灯	√
Single 状态灯	无按键,只有灯	√
MODE	状态模式切换按键	
RECSet	波形录制设置按键	
PGDN	下一页按键	
RECSTOP	录制停止	√
RECSTART	录制开始	√
PLAYPAUSE	录制波形的播放/暂停按键	√
SHKNLeft	飞梭左旋	
SHKNRight	飞梭右旋	

## 附录 2：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

## 附录 3：光标测量 XY 模式列表

位序	名称
1	光标 Ax 值
2	光标 Bx 值
3	Bx-Ax 差值
4	通道 1 的 Ay 值
5	通道 1 的 By 值
6	通道 1 的 By-Ay 差值
7	通道 2 的 Ay 值
8	通道 2 的 By 值
9	通道 2 的 By-Ay 差值
10	两通道的 Ay 极坐标半径
11	两通道的 By 极坐标半径
12	两通道的 By-Ay 极坐标半径
13	两通道的 Ay 极坐标夹角
14	两通道的 By 极坐标夹角
15	两通道的 By-Ay 极坐标夹角



16	通道 2 的 Ay 与通道 1 的 Ay 之积
17	通道 2 的 By 与通道 1 的 By 之积
18	通道 2 的 By-Ay 与通道 1 的 By-Ay 之积
19	通道 2 的 Ay 与通道 1 的 Ay 之比
20	通道 2 的 By 与通道 1 的 By 之比
21	通道 2 的 By-Ay 与通道 1 的 By-Ay 之比