User Manual

示波器编程手册

ZUS5000/ZUS6000 系列示波器

UM01010101 1.01 Date:2025/2/12

类别	内容
关键词	ZUS5000、ZUS6000系列示波器程控命令
摘要	详细描述ZUS5000、ZUS6000系列示波器支持的各种SCPI协议明细

修订历史

版本	日期	原因
V1.0	2024/4/15	完成初稿
V1.01	2025/2/12	1.新增参考波形相关指令； 2.新增时序相关指令； 3.新增截图设置文件名和存储路径相关指令； 4.新增通道标签； 5.修正 ZOOM 功能使能状态指令； 6.新增 ZUS5000 按键指令； 7.新增读取波形指令； 8.新增:SYSTem:ERRor 指令； 9.新增协议解码和协议触发指令; 10.新增编程例程说明。

目 录

1. SCPI 简介

1.1 概述

SCPI（Standard Commands for Programmable Instruments 的缩写），即可编程仪器标准命令，定义了一套可用于控制可编程测试测量仪器的标准语法和命令。

SCPI 于 1990 年 IEEE 488.2 协议一起面世。在 IEEE 488 协议中，IEEE 488.1 指定了物理和电气总线，IEEE 488.2 指定了协议和数据格式，但是都没有指定配套使用的指令集。不同的制造商，甚至不同的型号、相同类型的仪器都需要使用不同的命令集。SCPI 创建了一个标准，可以在所有的制造商和所有型号中通用。它需要使用 IEEE 488.2 的数据格式，但不必非得是 488.1 总线，也可用于串口（RS-232）、以太网、USB 接口、VXIbus 等若干硬件总线。

SCPI 命令是 ASCII 字符串，通过物理传输层传入。命令由一连串的关键字构成，有的还需要包括参数。在使用中，即可以写全名，也可以是仅包含大写字母的缩写。通常仪器对于查询命令的反馈也为 ASCII 代码。在传输大量数据时，二进制数据也是可以使用的。

1.2 命令语法

符号说明

冒号:

除了通用命令，绝大多数命令以冒号“:”开始，各级的关键字之间用冒号“:”分隔。

注：以上的符号中，除了冒号“:”外，其他的符号仅起解释作用，并不随命令一起发送。

大括号{ }

大括号中的内容为参数选项，各个参数项之间用竖线“|”分隔，使用命令时，必须选择其中一个参数。如{ON|OFF}作为参数时，表示 ON 和 OFF 必须而且只能选择一个。

中括号[ ]

方括号表示其中的内容是可以省略不写的。

三角括号< >

三角括号表示其中的内容必须使用一个有效值来替换，同时将< >删除。

竖线 |

竖线用于分隔多个参数选项，使用命令时，必须选择其中一个参数。

问号?

使用查询功能的命令须以“？”结尾，不可查询的命令不能以“？”结尾。

大小写和缩写

SCPI 命令一般有英文字母组成，并且不区分字母的大小写；但是为了便于书写，用户在书写时可以省略 SCPI 命令中的部分字母。具体而言，书写时，命令集里完整命令的大写字母不可省略，而小写字母则可以省略。

分隔

下面介绍不同命令、命令和参数、参数和参数之间分隔的规则。

1. **命令的分隔**

命令中间不允许用空格分隔

例如：“:CHANnel1 :COUPling AC”是错的。

不同级别的命令

SCPI 命令中的冒号“：”，除通用命令外，用于分隔不同级别的命令。

例如：“:CHANnel1:COUPling”中，“CHANnel1”是第一级命令，“COUPling”是第二级命令。

2. **参数的分隔**

参数与命令的分隔

当命令带有参数时，用一个英文空格将命令和参数分隔。正确和错误示范如下所示：

```
:CHANnel1:COUPling AC   √
:CHANnel1:COUPlingAC    ×
```

命令中多个参数之间的分隔

命令带有多参数时，用逗号将不同参数分隔。如“:MENU:SET STORAGE,REVERSE,ON”

3. **命令结束符**

每条命令结束时，为提高执行效率，应给该命令添加结束符“\n”或“;”。建议使用“\n”作为命令结束符，如：

```
:CURSor:MODE OFF\n
```

1.3 参数类型

布尔型（Bool）

布尔型的参数取值 0 或者 OFF 或者 FALSE、1 或者 ON 或者 TRUE，例如：

```
:DISPlay:FREEze 0
:DISPlay:FREEze OFF
:DISPlay:FREEze 1
:DISPlay:FREEze ON
```

都是有效的命令。

注：这里说明的是设置的参数，如果是作为返回值，则只返回 0 或者 1。

离散型（Discrete）

离散型又可称为枚举类型。如设置：

```
:MATH:MODE <mode>
```

其中的<mode>即为离散型，其可取的值为 OFF、ADD、SUBTract、MULTiply、DIVision、DIFFerential、INTegral 或者 FFT。

整型（Integer）

除非特殊说明或者限定了范围，整型可以是有效范围内的任意整数。此时的命令的参数必须为整数，不能为小数格式，否则将出现错误。例如：

```
:CURSor:X1Position <pos>
```

上述的指令设置 X1 光标的位置，其中的参数<pos>即为整型，且其范围是有限制的，为 0~699。

实型（Real）

实型的参数在有效范围内可以是任意实数，实型参数可以以小数格式和科学计数法的格式出现，例如：

```
:TRIGger:PULSe:UWIDth <uwidth>
```

其中参数<uwidth>为实型，其范围为 1.0E-9~4，若设置参数为 100ms，可以表示为 0.1 或者 1.0E-1。

1.4 结束符

所有指令的返回值都会以结束符 \n 结尾。

2. IEEE 488.2 通用命令

通用命令定义了兼容 IEEE488.2 标准的仪器所应支持的标准命令，用于仪器识别、复位、读取仪器设置以及确定仪器状态是如何读取和清除的。

查询仪器基本信息或执行常用基本操作。这些命令通常以“*”开头，命令关键字长度为 3 个字符。

- *CLS　　此命令用于清除以下状态：事件队列，标准事件状态寄存器，状态字节寄存器（除了 MAV 位）。
- *ESE　　设置和查询标准事件寄存器的使能寄存器。
- *ESR　　查询标准事件寄存器的值。
- *IDN?　　查询当前设备的信息。
- *OPC　　在所有操作完成之后，将标准事件寄存器的 OPC 位置 1。
- *OPT?　　查询并返回所有已安装的插件。
- *PSC　　设置上电清除操作功能状态。
- *RCL　　从指定位置加载系统配置参数。
- *RST　　除通信设置外的所有设置参数均恢复到出厂值。
- *SAV　　保存系统参数到指定位置。
- *SRE　　设置和查询状态字节寄存器的使能寄存器。
- *STB?　　查询状态字节寄存器的值。
- *TRG　　执行一次单次触发。
- *TST?　　执行一次自校验，并返回一个整形值。
- *WAI　　等待所有操作完成之后，才执行下一条命令。

*CLS(Clear Status)

命令格式

*CLS

功能描述

此命令用于清除以下状态：事件队列，标准事件状态寄存器，状态字节寄存器（除了MAV 位）。

参数说明

无。

实例说明

无。

*ESE(Standard Event Status Enable)

命令格式

*ESE <Int>

*ESE?

功能描述

设置标准事件寄存器的使能寄存器。

查询标准事件寄存器的使能寄存器。

参数说明

<Int> 范围 0~255 的一个整数，每个二进制位表示标准事件寄存器的一个使能位。

实例说明

设置标准事件寄存器的使能寄存器为 128。

*ESE 128

*ESR(Standard Event Status Register)

命令格式

*ESR?

功能描述

查询标准事件寄存器的值，读取后，清零该寄存器。

参数说明

无。

实例说明

查询标准事件寄存器的值。

*ESR? -> 64 返回 0~255 的一个十进制整数。

*IDN(Identification Number)

命令格式

*IDN?

功能描述

查询当前设备的信息。

参数说明

无。

实例说明

*IDN? ->

Zhiyuan Instruments,ZUS5054Pro,7842001732407100010,S0.01,1.3.17.15927(2024-09-03 19:04:26)

返回信息包括：<生产厂商>,<型号>,<序列号>,<固件版本>。

*OPC(Operation Complete)

命令格式

*OPC

*OPC?

功能描述

在所有操作（包含本命令）完成之后，将标准事件寄存器的 OPC 位置 1。

用于等待完成所有操作完成后，向输出缓冲区返回 1。*OPC?和 OPC 命令的不同之处在于，这种方式 IO 驱动会等待响应，可以避免应用程序显式循环。

参数说明

无。

实例说明

*OPC? -> 1

在 ZUS6000 系列示波器中，查询该位时，设备总是返回 1。

*OPT

命令格式

*OPT?

功能描述

查询并返回所有已安装的插件，返回格式为：<插件 1>:<描述>,<插件 2>:<描述>...。

参数说明

无。

实例说明

*OPT? -> DSO:READY 2018/6/5 20:18:55

*PSC

命令格式

*PSC <bool>

功能描述

设置上电清除操作功能状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开上电清除操作功能；

{OFF | FALSE | 0} 关闭上电清除操作功能。

实例说明

*PSC ON 打开上电清除功能。

*RCL

命令格式

*RCL <String>

功能描述

从指定位置加载系统配置参数，相关命令：*SAV。

参数说明

<String> 表示加载的系统参数的名称，不能包含非法字符：/\:?"<>|。

实例说明

*RCL config　　从 config 加载系统配置参数。

*RST(Reset)

命令格式

*RST

功能描述

除通信设置外的所有设置参数均恢复到出厂值。

参数说明

无。

实例说明

无。

*SAV

命令格式

*SAV <String>

功能描述

保存系统参数到指定位置，相关命令：*RCL。

参数说明

<String> 表示保存的系统参数的名称，不能包含非法字符：/\:?"<>|。

实例说明

*SAV config　　保存系统参数到 config。

*SRE(Service Request Enable)

命令格式

*SRE <Int>

*SRE?

功能描述

设置状态字节寄存器的使能寄存器。

查询状态字节寄存器的使能寄存器。

参数说明

<Int> 范围 0~255 的一个整数，每个二进制位表示状态字节寄存器的一个使能位。

实例说明

*SRE 32　　设置状态字节寄存器的使能寄存器为 32。

*STB(Read Status Byte)

命令格式

*STB?

功能描述

查询状态字节寄存器的值，读取后清零该寄存器，但不包括 MAV 位。

参数说明

无。

实例说明

*STB? -> 32　　返回 0~255 的一个十进制整数。

*TRG

命令格式

*TRG

功能描述

执行一次单次触发，相当于按下 Single。

参数说明

无。

实例说明

无。

*TST(Self Test)

命令格式

*TST?

功能描述

执行一次自校验，并返回一个整形值，0 表示无错误，1 表示错误。

参数说明

无。

实例说明

*TST? -> 0

*WAI

命令格式

*WAI

功能描述

等待所有操作（包含本命令）完成之后，才执行下一条命令。

参数说明

无。

实例说明

无。

3. 捕获设置相关命令

:ACQuire 命令用于查询和设置采样配置。

- :ACQuire:AVERages 设置和查询平均捕获的平均次数。
- :ACQuire:MAREa 设置和查询存储区域。
- :ACQuire:DEPTh? 查询当前存储深度。
- :ACQuire:MDEPth 设置存储深度。
- :ACQuire:SRATe? 查询采样率。
- :ACQuire:TYPE 设置和查询捕获模式。

:ACQuire:AVERages

命令格式

:ACQuire:AVERages <UInt>

:ACQuire:AVERages?

功能描述

设置平均捕获的平均次数。

查询平均捕获的平均次数。

参数说明

<UInt> 平均次数，范围 2~65536 且必须为 2 的整数次幂。

返回格式

查询命令以整数形式返回平均捕获的平均次数。

实例说明

:ACQuire:AVERages 2	设置平均捕获的平均次数为 2。
:ACQuire:AVERages? -> 2	当前平均捕获的平均次数为 2。

:ACQuire:MAREa

命令格式

:ACQuire:MAREa <String>

:ACQuire:MAREa?

功能描述

设置存储区域。

查询存储区域。

参数说明

<String> 范围 {AUTO | FIXEd}，

分别表示自动设置，固定。

返回格式

查询命令返回 AUTO 或 FIXEd。

实例说明

:ACQuire:MAREa AUTO　　设置存储区域为自动设置。

:ACQuire:MAREa? ->AUTO　　当前存储区域为自动设置。

:ACQuire:DEPTh

命令格式

:ACQuire:DEPTh?

功能描述

查询当前存储深度。

参数说明

无。

返回格式

查询命令以整数形式返回当前存储深度。

实例说明

:ACQuire:DEPTh? -> 500000　　当前存储深度为 500Kpts。

:ACQuire:MDEPth

命令格式

:ACQuire:MDEPth <UInt>

:ACQuire:MDEPth?

功能描述

设置存储深度。

查询存储深度。

参数说明

<UInt> 存储深度。

根据单/双通道的不同，可选的存储深度也不同：

单通道：{10K | 100K | 1M | 10M | 20M | 50M | 100M | 125M | 250M | 500M}；

多通道：{10K | 100K | 1M | 10M | 20M | 50M | 100M | 125M | 250M}。

注：ZDS5054 示波器单通道时可选的最大存储深度为 250M，多通道可选的最大存储深度为 125M。

返回格式

查询命令以整数形式返回存储深度。

实例说明

:ACQuire:MDEPth 10K	设置 10Kpts 的存储深度。
:ACQuire:MDEPth? -> 250000000	当前存储深度为 250Mpts。

:ACQuire:SRATe

命令格式

:ACQuire:SRATe?

功能描述

查询采样率。

参数说明

无。

返回格式

查询命令以实型形式返回采样率。

实例说明

:ACQuire:SRATe? -> 2.5e+09　　代表 2.50GSa/s。

:ACQuire:TYPE

命令格式

:ACQuire:TYPE <String>

:ACQuire:TYPE?

功能描述

设置捕获模式。

查询捕获模式。

参数说明

<String> 范围 {NORMal | PEAK | HRESolution | AVERages}，分别代表标准、峰值、高分辨率、平均。

返回格式

查询命令返回 NORMal、PEAK、HRESolution 或 AVERages。

实例说明

:ACQuire:TYPE NORMal	设置捕获模式为标准。
:ACQuire:TYPE? -> NORMal	查询捕获模式为标准。

4. 信号发生器相关命令

:AFG 命令用于查询和修改 AFG 相关配置。

- :AFG:APPLy:SYNC　　AFG 相位同步。
- :AFG:CHANnel<x>:SOURce　　设置和查询 AFG 通道开关状态。
- :AFG:CHANnel<x>:APPLy:PARAM　　设置和查询 AFG 参数设置。
- :AFG:CHANnel<x>:MODUlated:PARAM　　设置和查询 AFG 高级功能“调制”的选项。
- :AFG:CHANnel<x>:SWEEP:PARAM　　设置和查询 AFG 高级功能“扫频”的选项。
- :AFG:CHANnel<x>:BURST:PARAM　　设置和查询 AFG 高级功能“猝发”的选项。
- :AFG:CHANnel<x>:WAVE:USER:PATH　　加载自定义波形文件。

:AFG:APPLy:SYNC

命令格式

:AFG:APPLy:SYNC

功能描述

AFG 相位同步。

参数说明

无。

实例说明

:AFG:APPLy:SYNC　　设置 AFG 相位同步。

:AFG:CHANnel<x>:SOURce

命令格式

:AFG:CHANnel<x>:SOURce <Bool>

:AFG:CHANnel<x>:SOURce?

功能描述

设置 AFG 通道开关状态。

查询 AFG 通道开关状态。

参数说明

<x> 需要查询或改变的 AFG 通道编号，1 或 2。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开 AFG 通道；

{OFF | FALSE | 0} 关闭 AFG 通道。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:AFG:CHANnel1:SOURce ON	打开 AFG1。
:AFG:CHANnel1:SOURce? -> 1	查询 AFG1 为开启状态。

:AFG:CHANnel<x>:APPLy:PARAM

命令格式

:AFG:CHANnel<x>:APPLy:PARAM <String_1>,<String_2>

:AFG:CHANnel<x>:APPLy:PARAM? <String_1>

功能描述

设置 AFG 参数设置。

查询 AFG 参数设置。

参数说明

<x> AFG 通道编号，1 或 2。

<String_1>：

范围 {STATE | WAVE | FREQuency | VAMP | VOFFSET | PHASEANGle |Resistance | RAMPSYMMetry | DCycle | ADVAnced}，

分别表示开关，波形，频率，电压幅度，电压偏移，相位，阻抗，对称度，占空比，高级功能选择。

<String_2>：

当<String_1>为 STATE 时，范围 {0 | 1}，开关 AFG；

当<String_1>为 WAVE 时，范围 {SINusoid | RAMP | PULSe | SQUare | NOISe | DC | USER}，分别表示正弦波，三角波，脉冲波，方波，噪声，直流，自定义；

当<String_1>为 Resistance 时，范围 {1M | 50}，单位/Ω；

当<String_1>为 FREQuency、VAMP、VOFFSET、PHASEANGle、RAMSYMMetry、DCycle 时，填入数值。

当<String_1>为 ADVAnced 时，范围 {OFF | SWEEP | MODUlated | BURST}，分别表示关闭，扫频，调制，猝发。

返回格式

查询命令会根据<String_1>的不同返回结果有所不同，可参考参数说明。

实例说明

:AFG:CHANnel1:APPLy:PARAM Resistance,1M 设置 AFG 通道 1 的负载阻抗为 1MΩ。

:AFG:CHANnel1:APPLy:PARAM? WAVE,1 -> sinusoid 查询 AFG 通道 1 当前输出的波形为正弦波。

:AFG:CHANnel<x>:MODUlated:PARAM

命令格式

:AFG:CHANnel<x>:MODUlated:PARAM <String_1>,<String_2>[,<Int>]

:AFG:CHANnel<x>:MODUlated:PARAM? <String_1>[,<String_2>]

功能描述

设置 AFG 高级功能“调制”的选项。

查询 AFG 高级功能“调制”的选项。

参数说明

<x> AFG 通道编号，1 或 2。

<String_1>

范围{TYPE | WAVE | FREQuency | RAMPSYMMetry | AMADJUSTment | FMFREQuency | PMPHAse | BPSKPHAse | BFSKFREQuency | PWMDEVAtion}，

分别表示调制类型，调制波形，调幅频率，对称度，调制度，调相频率，调相偏差，BPSK 相移相位，BFSK 速率 PWM 占空比偏差。

<String_2>

当<String_1>为“TYPE”或“FREQuency”时，范围 {AM | FM | PM | BASK | BFSK | BPSK | PWM}，分别表示调幅，调频，调相位，BASK，BFSK，BPSK，PWM；

当<String_1>为“WAVE”时，范围 {SINusoid | RAMP | SQUare |NOISe}，分别表示正弦，三角波，方波，噪声。

其余情况下不需要<String_2>参数。

<Int>：

当<String_1>为“TYPE”和“WAVE”时，不需要该参数，其余情况下填数值。

实例说明

:AFG:CHANnel1:MODUlated:PARAM TYPE, AM 设置 AFG1 调制类型为调幅。

:AFG:CHANnel1:MODUlated:PARAM? TYPE -> AM 查询 AFG1 调制类型为调幅。

:AFG:CHANnel<x>:SWEEP:PARAM

命令格式

:AFG:CHANnel<x>:SWEEP:PARAM <String_1>,<String_2>

:AFG:CHANnel<x>:SWEEP:PARAM? <String_1>

功能描述

设置 AFG 高级功能“扫频”的选项。

查询 AFG 高级功能“扫频”的选项。

参数说明

<x> AFG 通道编号，1 或 2。

<String_1>：

范围{STARTFREQuency | ENDFREQuency | TIME}，

分别表示扫频开始频率，扫频结束频率，扫频时间。

<String_2>：

填入数值。

实例说明

:AFG:CHANnel1:SWEEP:PARAM STARTFREQuency,500 设置 AFG1 的扫频开始频率为 500Hz。

:AFG:CHANnel1:SWEEP:PARAM? STARTFREQuency, 1 -> 500 查询当前 AFG1 的扫频开始频率为 500Hz。

:AFG:CHANnel<x>:BURST:PARAM

命令格式

:AFG:CHANnel<x>:BURST:PARAM <String_1>,<String_2>

:AFG:CHANnel<x>:BURST:PARAM? <String_1>

功能描述

设置 AFG 高级功能“猝发”的选项。

查询 AFG 高级功能“猝发”的选项。

参数说明

<x> AFG 通道编号，1 或 2。

<String_1>

范围{BURSTTIME | BURSTCNT}，

分别表示猝发时间，猝发次数。

<String_2>

填入数值。

实例说明

:AFG:CHANnel1:BURST:PARAM BURSTTIME,0.02 设置 AFG1 猝发时间为 20ms。

:AFG:CHANnel<x>:WAVE:USER:PATH

命令格式

:AFG:CHANnel<x>:WAVE:USER:PATH <String>

功能描述

加载自定义波形文件。

参数说明

<String> 导入路径和文件名，如“/flash/wave/test.csv”。

实例说明

:AFG:CHANnel1:WAVE:USER:PATH /flash/wave/test.csv

加载名为 test 的自定义波形文件。

5. 自校准相关命令

:CALibrate 命令用于查询和设置校准。

- :CALibrate:COUNTer 查询开机后执行自校准次数。
- :CALibrate:DATE? 查询校准日期。
- :CALibrate:DATETIME? 查询校准日期和时间。
- :CALibrate:LOCK? 查询校准是否正在进行。
- :CALibrate:QUIT 退出校准模式。
- :CALibrate:STARt 开始自校准。
- :CALibrate:TIME? 查询校准时间。
- :CALibrate:VERSion? 查询校准参数版本。

:CALibrate:COUNTer

命令格式

:CALibrate:COUNTer?

功能描述

查询开机后执行自校准次数。

参数说明

无。

返回格式

查询命令以整数形式返回执行自校准次数。

实例说明

:CALibrate:COUNTer? -> 1　　开机后已执行一次自校准。

:CALibrate:DATE

命令格式

:CALibrate:DATE?

功能描述

查询校准日期。

参数设置

无。

实例说明

:CALibrate:DATE? -> 2023-08-22　　查询校准日期。

:CALibrate:DATETIME

命令格式

:CALibrate:DATETIME?

功能描述

查询校准日期和时间。

参数设置

无。

实例说明

:CALibrate:DATETIME? -> 2023-08-22 14:54:07　　查询校准日期和时间。

:CALibrate:LOCK

命令格式

:CALibrate:LOCK?

功能描述

查询校准是否正在进行。

参数设置

无。

实例说明

:CALibrate:LOCK? -> 0　　目前未进行自校准。

:CALibrate:QUIT

命令格式

:CALibrate:QUIT

功能描述

退出校准模式。

参数设置

无。

实例说明

无。

:CALibrate:STARt

命令格式

:CALibrate:STARt

功能描述

开始自校准。

参数说明

无。

实例说明

:CALibrate:STARt　　开始自校准。

:CALibrate:TIME

命令格式

:CALibrate:TIME?

功能描述

查询校准时间。

参数设置

无。

实例说明

:CALibrate:TIME? -> 14:54:07　　查询校准时间。

:CALibrate:VERSion

命令格式

:CALibrate:VERSion?

功能描述

查询校准参数版本。

参数设置

无。

实例说明

:CALibrate:VERSion? -> 1.01　　查询校准参数版本。

6. 通道相关指令

:CHANnel 命令用于查询和设置模拟通道信息。

- :CHANnel<x>:DISPlay 设置和查询垂直通道状态。
- :CHANnel<x>:SHOW 设置和查询垂直通道显示状态。
- :CHANnel<x>:COUPling 设置和查询垂直通道的耦合类型。
- :CHANnel<x>:SCALe 设置和查询垂直通道的档位。
- :CHANnel<x>:OFFSet 设置和查询垂直通道的偏移。
- :CHANnel<x>:WBRightness 设置和查询垂直通道的波形亮度。
- :CHANnel<x>:BWLimit 设置和查询垂直通道的带宽限制。
- :CHANnel<x>:UNITs 设置和查询垂直通道的探头类型。
- :CHANnel<x>:TERMination 设置和查询垂直通道的阻抗。
- :CHANnel<x>:INVert 设置和查询垂直通道的反相开关。
- :CHANnel<x>:FREQuency? 查询垂直通道的硬件频率计。
- :CHANnel<x>:BAUDRAte? 查询垂直通道的硬件频率计的波特率。
- :CHANnel<x>:PROBe:Final 设置和查询垂直通道的外部衰减。
- :CHANnel<x>:FINe 设置和查询垂直通道的档位调节。
- :CHANnel<x>:DELAy 设置和查询垂直通道的延时校正。
- :CHANnel<x>:LABEL 设置和查询垂直通道的标签名。
- :CHANnel<x>:LABEL:CLEAR 清除垂直通道的标签名。

:CHANnel<x>:DISPlay

命令格式

:CHANnel<x>:DISPlay <Bool>

:CHANnel<x>:DISPlay?

功能描述

设置垂直通道状态。

查询垂直通道状态。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开垂直通道；

{OFF | FALSE | 0} 关闭垂直通道。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:CHANnel1:DISPlay ON	打开垂直通道 1。
:CHANnel1:DISPlay? -> 1	查询垂直通道 1 为打开状态。

:CHANnel<x>:SHOW

命令格式

:CHANnel<x>:SHOW <Bool>

:CHANnel<x>:SHOW?

功能描述

设置垂直通道显示状态。

查询垂直通道显示状态。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 显示垂直通道；

{OFF | FALSE | 0} 隐藏垂直通道。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：显示，隐藏。

实例说明

:CHANnel1:SHOW ON	显示垂直通道 1。
:CHANnel1:SHOW? -> 1	查询当前垂直通道 1 为显示状态。

:CHANnel<x>:COUPling

命令格式

:CHANnel<x>:COUPling <String>

:CHANnel<x>:COUPling?

功能描述

设置垂直通道的耦合类型。

查询垂直通道的耦合类型。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<String> 范围{DC | AC | GND}。

返回格式

查询命令会返回 DC、AC 或 GND。

实例说明

:CHANnel1:COUPling DC	设置通道 1 通道耦合为 DC。
:CHANnel1:COUPling? -> DC	查询当前通道 1 通道耦合为 DC。

:CHANnel<x>:SCALe

命令格式

:CHANnel<x>:SCALe <String>

:CHANnel<x>:SCALe?

功能描述

设置垂直通道的档位。

查询垂直通道的档位。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<String> 垂直档位，可带单位，如“5mV”，也可用科学计数法表示。

返回格式

查询命令以实型形式返回垂直通道的档位，单位 V。

实例说明

:CHANnel1:SCALe 5mV	设置垂直通道 1 垂直档位为 5mV。
:CHANnel1:SCALe? -> 0.005	查询当前垂直通道 1 垂直档位为 5mV。

:CHANnel<x>:OFFSet

命令格式

:CHANnel<x>:OFFSet <String>

:CHANnel<x>:OFFSet?

功能描述

设置垂直通道的偏移。

查询垂直通道的偏移。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<String> 偏移量，可带单位，如“5mV”，也可用科学计数法表示。

返回格式

查询命令以实型形式返回垂直通道的偏移量，单位 V。

实例说明

:CHANnel1:OFFSet 5mV	设置垂直通道 1 垂直偏移为 5mV。
:CHANnel1:OFFSet? -> 0.005	查询当前垂直通道 1 垂直偏移为 5mV。

:CHANnel<x>:WBRightness

命令格式

:CHANnel<x>:WBRightness <UInt>

:CHANnel<x>:WBRightness?

功能描述

设置垂直通道的波形亮度。

查询垂直通道的波形亮度。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<UInt> 亮度，范围 0~100，单位%。

返回格式

查询命令以整数形式返回垂直通道的波形亮度，单位百分比。

实例说明

:CHANnel1:WBRightness 60	设置通道 1 的波形亮度为 60%。
:CHANnel1:WBRightness? -> 60	查询当前通道 1 的波形亮度为 60%。

:CHANnel<x>:BWLimit

命令格式

:CHANnel<x>:BWLimit <String>

:CHANnel<x>:BWLimit?

功能描述

设置垂直通道的带宽限制。

查询垂直通道的带宽限制。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<String> 范围{FULL | 20MHz | 500MHz}。

注：ZUS6054 示波器无 500MHz 选项。

返回格式

查询命令返回 FULL、20MHz 或 500MHz。

实例说明

:CHANnel1:BWLimit 500MHz　　设置通道 1 的带宽限制为 500MHz。

:CHANnel1:BWLimit? -> 500MHz　　查询当前通道 1 的带宽限制为 500MHz。

:CHANnel<x>:UNITs

命令格式

:CHANnel<x>:UNITs <String>

:CHANnel<x>:UNITs?

功能描述

设置垂直通道的探头类型。

查询垂直通道的探头类型。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<String> 范围 {VOLTAGE | CURRENT}，

分别表示电压、电流。

返回格式

查询命令返回 VOLTAGE 或 CURRENT。

实例说明

:CHANnel1:UNITs VOLTAGE	设置通道 1 的探头类型为电压。
:CHANnel1:UNITs? -> CURRENT	查询当前通道 1 的探头类型为电流。

:CHANnel<x>:TERMination

命令格式

:CHANnel<x>:TERMination <String>

:CHANnel<x>:TERMination?

功能描述

设置垂直通道的阻抗。

查询垂直通道的阻抗。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<String> 范围{1M | 50}，单位Ω。

返回格式

查询命令返回 1M 或 50，单位Ω。

实例说明

:CHANnel1:TERMination 1M	设置通道 1 的输入阻抗为 1MΩ。
:CHANnel1:TERMination? -> 1M	查询通道 1 的输入阻抗为 1MΩ。

:CHANnel<x>:INVert

命令格式

:CHANnel<x>:INVert <Bool>

:CHANnel<x>:INVert?

功能描述

设置垂直通道的反相开关。

查询垂直通道的反相开关。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开垂直通道的反相开关；

{OFF | FALSE | 0}，关闭垂直通道的反相开关。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:CHANnel1:INVert ON	打开通道 1 的反相开关。
:CHANnel1:INVert? -> 0	查询当前通道 1 的反相开关为关闭状态。

:CHANnel<x>:FREQuency

命令格式

:CHANnel<x>:FREQuency?

功能描述

查询垂直通道的硬件频率计。

参数说明

<x> 需要查询的垂直通道编号，1、2、3 或 4。

返回格式

查询命令以实型的形式返回垂直通道的硬件频率计，单位 Hz。

实例说明

:CHANnel1:FREQuency? -> 0　　查询当前通道 1 的硬件频率计为 0Hz。

:CHANnel<x>:BAUDRAte

命令格式

:CHANnel<x>:BAUDRAte?

功能描述

查询垂直通道的硬件频率计的波特率。

参数说明

<x> 需要查询的垂直通道编号，1、2、3 或 4。

返回格式

查询命令以实型的形式返回垂直通道的硬件频率计的波特率，单位 Bd。

实例说明

:CHANnel1:BAUDRAte? -> 0　　查询当前通道 1 的硬件频率计为 0Bd。

:CHANnel<x>:PROBe:Final

命令格式

:CHANnel<x>:PROBe:Final <Double>

:CHANnel<x>:PROBe:Final?

功能描述

设置垂直通道的外部衰减。

查询垂直通道的外部衰减。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<Double> 衰减比，范围 0.01X~10MX。

返回格式

查询命令以实型形式返回垂直通道的外部衰减。

实例说明

:CHANnel1:PROBe:Final 100	设置通道 1 的探头比为 100X。
:CHANnel1:PROBe:Final? ->10	查询当前通道 1 的探头比为 10X。

:CHANnel<x>:FINe

命令格式

:CHANnel<x>:FINe <String>

:CHANnel<x>:FINe?

功能描述

设置垂直通道的档位调节。

查询垂直通道的档位调节。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<String> 范围{COARSe | FINe}，

分别表示粗调、细调。

返回格式

查询命令返回 COARSe 或 FINe。

实例说明

:CHANnel1:FINe COARSe	设置通道 1 的档位调节为粗调。
:CHANnel1:FINe? -> FINe	查询当前通道 1 的档位调节为细调。

:CHANnel<x>:DELAy

命令格式

:CHANnel<x>:DELAy <UInt>

:CHANnel<x>:DELAy?

功能描述

设置垂直通道的延时校正。

查询垂直通道的延时校正。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<UInt> 范围-100~100，单位为 ns。

返回格式

查询命令以整数形式返回垂直通道的延时校正。

实例说明

:CHANnel1:DELAy 0	设置通道 1 延时校正为 0ns。
:CHANnel1:DELAy? ->0	查询当前通道 1 的延时校正为 0ns。

:CHANnel<x>:LABEL

命令格式

:CHANnel<x>:LABEL <String>

:CHANnel<x>:LABEL?

功能描述

设置垂直通道的标签名。

查询垂直通道的标签名。

参数说明

<String> 任意标签名。

返回格式

查询命令以字符串形式返回标签名。

实例说明

:CHANnel1:LABEL SCK	设置通道 1 的标签名为 SCK。
:CHANnel1:LABEL? -> SCK	查询当前通道 1 的标签名为 SCK。

:CHANnel<x>:LABEL:CLEAR

命令格式

:CHANnel<x>:LABEL:CLEAR

功能描述

清除垂直通道的标签名。

参数说明

无。

实例说明

:CHANnel1:LABEL:CLEAR　　清除通道 1 的标签名。

7. 光标相关命令

:CURSor 命令用于查询和设置光标的配置。

- :CURSor:STATES 设置和查询光标的开关状态。
- :CURSor:MODE 设置和查询光标类型。
- :CURSor:SHOW:MODE 设置和查询光标显示模式。
- :CURSor:CHANNel:MODE 设置和查询光标通道类型。
- :CURSor:CHANNel 当光标通道类型为相同时，设置和查询光标测量通道选择。
- :CURSor:X1CHANNel 当光标通道类型为区分时，设置和查询光标测量光标 A 的通道选择。
- :CURSor:X2CHANNel 当光标通道类型为区分时，设置和查询光标测量光标 B 的通道选择。
- :CURSor:X1Position 设置和查询示波器的垂直光标 A 的水平位置。
- :CURSor:X2Position 设置和查询示波器的垂直光标 B 的水平位置。
- :CURSor:X1Value 设置和查询示波器的垂直光标 A 的水平位置。
- :CURSor:X2Value 设置和查询示波器的垂直光标 B 的水平位置。
- :CURSor:X1Wave? 查询示波器的垂直光标 A 的当前测量值。
- :CURSor:X2Wave? 查询示波器的垂直光标 B 的当前测量值。
- :CURSor:XDELta? 查询示波器的垂直光标 A 和垂直光标 B 的的水平位置差值 ΔX。
- :CURSor:IXDElta? 查询示波器的垂直光标 1 和垂直光标 2 的的水平位置差值 ΔX 的倒数。
- :CURSor:Y1Position 设置和查询示波器的水平光标 A 的垂直位置。
- :CURSor:Y2Position 设置和查询示波器的水平光标 B 的垂直位置。
- :CURSor:Y1Value 设置和查询示波器的水平光标 A 的垂直位置。
- :CURSor:Y2Value 设置和查询示波器的水平光标 B 的垂直位置。
- :CURSor:YDELta? 查询示波器的水平光标 A 和水平光标 B 的的垂直位置差值 ΔY。

:CURSor:STATES

命令格式

:CURSor:STATES <bool>

:CURSor:STATES?

功能描述

设置光标的开关状态。

查询光标的开关状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开光标功能；

{OFF | FALSE | 0}，关闭光标功能。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:CURSor:STATES ON　　打开光标功能。

:CURSor:STATES? -> 0　　查询当前光标功能为关闭状态。

:CURSor:MODE

命令格式

:CURSor:MODE <String>

:CURSor:MODE?

功能描述

设置光标类型。

查询光标类型。

参数说明

<String> 范围{VERTical | HORIzontal | ALL}，

分别表示垂直光标、水平光标、网格光标。

返回格式

查询命令返回 VERTical、HORIzontal 或 ALL。

实例说明

:CURSor:MODE VERTical	设置光标类型为垂直光标。
:CURSor:MODE? -> ALL	查询当前光标类型为网格光标。

:CURSor:SHOW:MODE

命令格式

:CURSor:SHOW:MODE <String>

:CURSor:SHOW:MODE?

功能描述

设置光标显示模式。

查询光标显示模式。

参数说明

<String> 范围{LINE | TABLE}，

分别表示线、表格。

返回格式

查询命令返回 LINE 或 TABLE。

实例说明

:CURSor:SHOW:MODE TABLE 设置光标显示模式为表格模式。

:CURSor:SHOW:MODE? -> TABLE 查询当前光标显示模式为表格模式。

:CURSor:CHANNel:MODE

命令格式

:CURSor:CHANNel:MODE <String>

:CURSor:CHANNel:MODE?

功能描述

设置光标通道类型。

查询光标通道类型。

参数说明

<String> 范围{ SAME | SPLIT}，

分别表示相同、区分。

返回格式

查询命令返回 SAME 或 SPLIT。

实例说明

:CURSor:CHANNel:MODE SAME 设置光标通道类型为相同。

:CURSor:CHANNel:MODE? -> SAME 查询当前光标通道类型为相同。

:CURSor:CHANNel

命令格式

:CURSor:CHANNel <String>[,<String_CH>]...[,<String_CH>]

:CURSor:CHANNel?

功能描述

当光标通道类型为相同时，设置光标测量通道选择。

当光标通道类型为相同时，查询光标测量通道选择。

参数说明

<String> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 | Trend }

<String_CH> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | Trend }

注 1：光标通道类型设置为**相同**时有效，相关指令:CURSor:CHANNel:MODE。

注 2：光标显示模式为“**线**”时，仅可设置一个参数；光标显示模式为“**表格**”时，可设置多个参数。相关指令:CURSor:SHOW:MODE。

返回格式

当光标显示模式查询命令为“线”时，查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、MATH1、MATH2、MATH3、MATH4、REF1、REF2、REF3、REF4 或 Trend 中的一个参数；

当光标显示模式查询命令为“表格”时，查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、MATH1、MATH2、MATH3、MATH4、REF1、REF2、REF3、REF4 或 Trend 中的多个参数。

实例说明

:CURSor:CHANNel CHANnel1　　设置光标测量通道选择为通道 1。

:CURSor:CHANNel? -> CHANnel1,CHANnel2,CHANnel3　　查询光标测量通道选择为通道 1、通道 2 和通道 3。

:CURSor:X1CHANNel

命令格式

:CURSor:X1CHANNel <String>[,<String_CH>]...[,<String_CH>]

:CURSor:X1CHANNel?

功能描述

当光标通道类型为区分时，设置光标测量光标 A 的通道选择。

当光标通道类型为区分时，查询光标测量光标 A 的通道选择。

参数说明

<String> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 | Trend }

<String_CH> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | Trend }

注 1：光标通道类型设置为**区分**时有效，相关指令:CURSor:CHANNel:MODE。

注 2：光标显示模式为“**线**”时，仅可设置一个参数；光标显示模式为“**表格**”时，可设置多个参数。相关指令:CURSor:SHOW:MODE。

返回格式

当光标显示模式查询命令为“线”时，查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、MATH1、MATH2、MATH3、MATH4、REF1、REF2、REF3、REF4 或 Trend 中的一个参数；

当光标显示模式查询命令为“表格”时，查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、MATH1、MATH2、MATH3、MATH4、REF1、REF2、REF3、REF4 或 Trend 中的多个参数。

实例说明

:CURSor:X1CHANNel CHANnel1　　设置光标测量光标 A 的通道选择为通道 1。

:CURSor:X1CHANNel? -> CHANnel1,CHANnel2,CHANnel3　　查询光标测量光标 A 的通道选择为通道 1、通道 2 和通道 3。

:CURSor:X2CHANNel

命令格式

:CURSor:X2CHANNel <String>[,<String_CH>]...[,<String_CH>]

:CURSor:X2CHANNel?

功能描述

当光标通道类型为区分时，设置光标测量光标 B 的通道选择。

当光标通道类型为区分时，查询光标测量光标 B 的通道选择。

参数说明

<String> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 | Trend }

<String_CH> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | Trend }

注 1：光标通道类型设置为**区分**时有效，相关指令:CURSor:CHANNel:MODE。

注 2：光标显示模式为“**线**”时，仅可设置一个参数；光标显示模式为“**表格**”时，可设置多个参数。相关指令:CURSor:SHOW:MODE。

返回格式

当光标显示模式查询命令为“线”时，查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、MATH1、MATH2、MATH3、MATH4、REF1、REF2、REF3、REF4 或 Trend 中的一个参数；

当光标显示模式查询命令为“表格”时，查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、MATH1、MATH2、MATH3、MATH4、REF1、REF2、REF3、REF4 或 Trend 中的多个参数。

实例说明

:CURSor:X2CHANNel CHANnel1　　设置光标测量光标 B 的通道选择为通道 1。

:CURSor:X2CHANNel? -> CHANnel1,CHANnel2,CHANnel3　　查询光标测量光标 B 的通道选择为通道 1、通道 2 和通道 3。

:CURSor:X1Position

命令格式

:CURSor:X1Position <UInt>

:CURSor:X1Position?

功能描述

设置示波器的垂直光标 A 的水平位置。

查询示波器的垂直光标 A 的水平位置。

时间和位置换算公式：$\frac{\text{需设置的时刻}-\text{水平偏移量}}{\text{水平档位}} * 50$

参数说明

<UInt> 范围：-250~249。

返回格式

查询命令以整数形式返回示波器的垂直光标 A 的水平位置。

实例说明

:CURSor:X1Position 0	设置垂直光标 A 的水平位置为 0。
:CURSor:X1Position? -> 0	查询当前垂直光标 A 的水平位置为 0。

:CURSor:X2Position

命令格式

:CURSor:X2Position <UInt>

:CURSor:X2Position?

功能描述

设置示波器的垂直光标 B 的水平位置。

查询示波器的垂直光标 B 的水平位置。

时间和位置换算公式：$\frac{\text{需设置的时刻}-\text{水平偏移量}}{\text{水平档位}} * 50$

参数说明

<UInt> 范围：-250~249。

返回格式

查询命令以整数形式返回示波器的垂直光标 B 的水平位置。

实例说明

:CURSor:X2Position 0	设置垂直光标 B 的水平位置为 0。
:CURSor:X2Position? -> 0	查询当前垂直光标 B 的水平位置为 0。

:CURSor:X1Value

命令格式

:CURSor:X1Value <Double>

:CURSor:X1Value?

功能描述

设置示波器的垂直光标 A 的水平位置。

查询示波器的垂直光标 A 的水平位置。

参数说明

<Double> 时间字符串，单位 s。

返回格式

查询命令以实型形式返回示波器的垂直光标 A 的水平位置。

实例说明

:CURSor:X1Value 0.05	将垂直光标 A 放置在 50ms 的位置。
:CURSor:X1Value? -> 0.05	查询当前垂直光标 A 的水平位置为 50ms。

:CURSor:X2Value

命令格式

:CURSor:X2Value <Double>

:CURSor:X2Value?

功能描述

设置示波器的垂直光标 B 的水平位置。

查询示波器的垂直光标 B 的水平位置。

参数说明

<Double> 时间字符串，单位 s。

返回格式

查询命令以实型形式返回示波器的垂直光标 B 的水平位置。

实例说明

:CURSor:X2Value 0.05	将垂直光标 B 放置在 50ms 的位置。
:CURSor:X2Value? -> 0.05	查询当前垂直光标 B 的水平位置为 50ms。

:CURSor:X1Wave

命令格式

:CURSor:X1Wave? <String>

功能描述

查询示波器的垂直光标 A 的当前测量值。

参数说明

<String> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 | Trend }。

返回格式

查询命令以实型形式返回示波器的垂直光标 A 的当前测量值，单位 V 或 A。

注：无效时返回“---”。

实例说明

:CURSor:X1Value? -> 0.5　　查询垂直光标 A 的当前测量值为 500mV。

:CURSor:X2Wave

命令格式

:CURSor:X2Wave? <String>

功能描述

查询示波器的垂直光标 B 的当前测量值。

参数说明

<String> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 | Trend }。

返回格式

查询命令以实型形式返回示波器的垂直光标 B 的当前测量值，单位 V 或 A。

注：无效时返回“---”。

实例说明

:CURSor:X1Value? -> 0.5　　查询垂直光标 B 的当前测量值为 500mV。

:CURSor:XDELta

命令格式

:CURSor:XDELta?

功能描述

查询示波器的垂直光标 A 和垂直光标 B 的的水平位置差值 ΔX。

参数说明

无。

返回格式

查询命令以实型形式返回水平位置差值 ΔX，单位 s。

实例说明

:CURSor:XDELta? ->0.0998　　查询当前垂直光标 A 和垂直光标 B 的的水平位置差值为 99.8ms。

:CURSor:IXDElta

命令格式

:CURSor:IXDElta?

功能描述

查询示波器的垂直光标 1 和垂直光标 2 的的水平位置差值 ΔX 的倒数。

参数说明

无。

返回格式

查询命令以实型形式返回水平位置差值 ΔX 的倒数，单位 Hz。

实例说明

:CURSor:IXDElta? ->10.02004　　查询当前垂直光标 A 和垂直光标 B 的的水平位置差值的倒数为 10.02004Hz。

:CURSor:Y1Position

命令格式

:CURSor:Y1Position <UInt>

:CURSor:Y1Position?

功能描述

设置示波器的水平光标 A 的垂直位置。

查询示波器的水平光标 A 的垂直位置。

电压和位置换算公式：$\frac{\text{需设置的电压+垂直偏移量}}{\text{垂直档位}} * 50$

参数说明

<UInt> 范围：-200~199。

返回格式

查询命令以整数形式返回示波器的水平光标 A 的垂直位置。

实例说明

:CURSor:Y1Position 0	设置水平光标 A 的垂直位置为 0。
:CURSor:Y1Position? -> 0	查询当前水平光标 A 的垂直位置为 0。

:CURSor:Y2Position

命令格式

:CURSor:Y2Position <UInt>

:CURSor:Y2Position?

功能描述

设置示波器的水平光标 B 的垂直位置。

查询示波器的水平光标 B 的垂直位置。

电压和位置换算公式：$\frac{\text{需设置的电压+垂直偏移量}}{\text{垂直档位}} * 50$

参数说明

<UInt> 范围：-200~199。

返回格式

查询命令以整数形式返回示波器的水平光标 B 的垂直位置。

实例说明

:CURSor:Y2Position 0	设置水平光标 B 的垂直位置为 0。
:CURSor:Y2Position? -> 0	查询当前水平光标 B 的垂直位置为 0。

:CURSor:Y1Value

命令格式

:CURSor:Y1Value <Double>

:CURSor:Y1Value?

功能描述

设置示波器的水平光标 A 的垂直位置。

查询示波器的水平光标 A 的垂直位置。

参数说明

<Double> 电压/电流值。

返回格式

查询命令以实型形式返回示波器的水平光标 A 的垂直位置，单位 V 或 A。

实例说明

:CURSor:Y1Value -3　　将水平光标 A 放置在-3V 的位置。

:CURSor:Y1Value? ->-3　　查询水平光标 A 的垂直位置为-3A。

:CURSor:Y2Value

命令格式

:CURSor:Y2Value <Double>

:CURSor:Y2Value?

功能描述

设置示波器的水平光标 B 的垂直位置。

查询示波器的水平光标 B 的垂直位置。

参数说明

<Double> 电压/电流值。

返回格式

查询命令以实型形式返回示波器的水平光标 B 的垂直位置，单位 V 或 A。

实例说明

:CURSor:Y2Value -3	将水平光标 B 放置在-3V 的位置。
:CURSor:Y2Value? ->-3	查询水平光标 B 的垂直位置为-3A。

:CURSor:YDELta

命令格式

:CURSor:YDELta?

功能描述

查询示波器的水平光标 A 和水平光标 B 的的垂直位置差值 ΔY。

参数说明

无。

返回格式

查询命令以实型形式返回垂直位置差值 ΔY，单位 V 或 A。

实例说明

:CURSor:YDELta? -> 74.4　　查询示波器的水平光标 A 和水平光标 B 的的水平位置差值 ΔY 为 74.4A。

8. 解码相关命令

:DECODe 命令组用于查询和修改解码相关配置。

- :DECODe<x>:STATE 设置解码通道使能状态。
- :DECODe<x>:PLUG 设置解码类型。
- :DECODe<x>:EVENTtable 设置解码事件表的开关状态。
- :DECODe<x>:REPORT:HTML 导出解码网页报表。
- :DECODe<x>:REPORT:CSV 导出解码 CSV 文件。
- :DECODe<x>:REPORT:COUNTer? 查询开机后执行解码报表导出次数。

:DECODe<x>:STATE

命令格式

:DECODe<x>:STATE <Bool>

功能描述

设置解码通道使能状态。

参数说明

<x> 解码通道序号，1 或 2。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开解码功能；

{OFF | FALSE | 0}，关闭解码功能。

实例说明

:DECODe1:STATE 1　　使能解码 1。

:DECODe<x>:PLUG

命令格式

:DECODe<x>:PLUG <String>

:DECODe<x>:PLUG?

功能描述

设置解码类型。

查询解码类型

参数说明

<x> 解码通道序号，1 或 2。

<String> 范围{UART | IIC | IICDEVICE | SPI | MODBUS | MIPIDSI | MIPIRFFE | CAN | CANFD | LIN | FLEXRAY | SENT | MVB | WTB | 1553B | ISO7816 | TDM | IIS | USB | USBPD | QC | SDSPI | SDSD | WIEGAND | DS18B20 | PS2 | MDIO | DALI | HDQ | ONEWIRE | MANCHESTER | DIFFMANCHESTER | MILLER | DHT11 | SHT11 | DMX512 | I3C | ARINC429 | NEC | RC5 | RC6}

分别表示 UART、I2C、I2C device、SPI、Modbus、MIPI_DSI、MIPI_RFFE、CAN、CAN FD、LIN、FlexRay、SENT、MVB、WTB、1553B、IS07816、TDM、I2S、USB、USB_PD、QC2.0\3.0、SD_SPI、SD_SD、Wiegand、DS18B20、PS/2、MDIO、DALI、HDQ、1-Wire、Manchester、Diff-Mancheter、Miller、DHT11、SHT11、DMX512、I3C、ARINC429、NEC、RC5、RC6。

注：ZUS5054 示波器标配 CAN、USB、UART、I2C、I2C device、SPI、I2S、ModBus、1553B、PS/2，其余协议触发类型可选配。

实例说明

:DECODe1:PLUG UART	设置解码类型为 UART。
:DECODe1:PLUG? -> UART	查询解码类型为 UART。

:DECODe<x>:EVENTtable

命令格式

:DECODe<x>:EVENTtable <Bool>

功能描述

设置解码事件表的开关状态。

参数说明

<x> 解码通道序号，1 或 2。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开解码事件表；

{OFF | FALSE | 0}，关闭解码事件表。

实例说明

:DECODe1:EVENTtable 1　　打开解码 1 的事件表。

:DECODe<x>:REPORT:HTML

命令格式

:DECODe<x>:REPORT:HTML <String>

功能描述

导出解码网页报表。

参数说明

<x> 解码通道序号，1 或 2。

<String> 导出路径和文件名，如“/flash/decode1.html”。

注 1：导出网页报表前请先打开事件表，相关指令:DECODe<x>:EVENTtable。

注 2：文件名必须以“*.html”为尾缀。

实例说明

:DECODe1:REPORT:HTML /Udisk0/decode1.html

将本次解码结果以网页报表的格式导入 U 盘，并将文件名命名为 decode1。

:DECODe<x>:REPORT:CSV

命令格式

:DECODe<x>:REPORT:CSV <String>

功能描述

导出解码 CSV 文件。

参数说明

<x> 解码通道序号，1 或 2。

<String> 导出路径和文件名，如“/flash/decode1.csv”。

注 1：导出 CSV 文件前请先打开事件表，相关指令:DECODe<x>:EVENTtable。

注 1：文件名必须以“*.csv”为尾缀。

实例说明

:DECODe1:REPORT:CSV /Udisk0/decode1.csv

将本次解码结果以 CSV 文件格式导入 U 盘，并将文件名命名为 decode1。

:DECODe<x>:REPORT:COUNTer

命令格式

:DECODe<x>:REPORT:COUNTer?

功能描述

查询开机后执行解码报表导出次数。

参数说明

<x> 解码通道序号，1 或 2。

返回格式

查询命令以整数形式返回开机后执行解码报表导出次数。

实例说明

:DECODe1:REPORT:COUNTer? -> 1

开机后解码报表共导出 1 次。

:DECODe<x>:1553B:SOURce

命令格式

:DECODe<x>:1553B:SOURce <String>

:DECODe<x>:1553B:SOURce?

功能描述

设置 1553B 协议解码的信源。

查询 1553B 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1、2、3 或 4。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:1553B:SOURce CHANnel1	设置 1553B 协议解码信源为通道 1。
:DECODe1:1553B:SOURce? -> CHANnel1	查询 1553B 协议解码信源为通道 1。

:DECODe<x>:1WIRe:SOURce

命令格式

:DECODe<x>:1WIRe:SOURce <String>

:DECODe<x>:1WIRe:SOURce?

功能描述

设置 1WIRe 协议解码的信源。

查询 1WIRe 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:1WIRe:SOURce CHANnel1	设置 1WIRe 协议解码信源为通道 1。
:DECODe1:1WIRe:SOURce? -> CHANnel1	查询 1WIRe 协议解码信源为通道 1。

:DECODe<x>:1WIRe:SPEEdmode

命令格式

:DECODe<x>:1WIRe:SPEEdmode <String>

:DECODe<x>:1WIRe:SPEEdmode?

功能描述

设置 1WIRe 协议解码的速度模式。

查询 1WIRe 协议解码的速度模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { STANdard | DRIVer }，分别表示标准、驱动。

返回格式

查询命令返回 STANdard 或 DRIVer。

实例说明

:DECODe1:1WIRe:SPEEdmode STANdard 设置 1WIRe 协议解码速度模式为标准。

:DECODe1:1WIRe:SPEEdmode? -> STANdard 查询 1WIRe 协议解码速度模式为标准。

:DECODe<x>:1WIRe:HOFFset

命令格式

:DECODe<x>:1WIRe:HOFFset <Int>

:DECODe<x>:1WIRe:HOFFset?

功能描述

设置 1-Wire 协议解码的采样偏移。

查询 1-Wire 协议解码的采样偏移。

注：速度模式为**驱动**时有效，相关指令:DECODe<x>:1WIRe:SPEEdmode，单位为 us。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 1~16。

返回格式

查询命令以整数形式返回 1-Wire 协议解码的采样偏移。

实例说明

:DECODe1:1WIRe:HOFFset 6	设置 1-Wire 协议解码的采样偏移为 6us。
:DECODe1:1WIRe:HOFFset? -> 6	查询 1-Wire 协议解码的采样偏移为 6us。

:DECODe<x>:1WIRe:LOFFset

命令格式

:DECODe<x>:1WIRe:LOFFset <Int>

:DECODe<x>:1WIRe:LOFFset?

功能描述

设置 1-Wire 协议解码的采样偏移。

查询 1-Wire 协议解码的采样偏移。

注：速度模式为**标准**时有效，相关指令:DECODe<x>:1WIRe:SPEEdmode，单位为 us。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 1~60。

返回格式

查询命令以整数形式返回 1-Wire 协议解码的采样偏移。

实例说明

:DECODe1:1WIRe:LOFFset 6	设置 1-Wire 协议解码的采样偏移为 6us。
:DECODe1:1WIRe:LOFFset? -> 6	查询 1-Wire 协议解码的采样偏移为 6us。

:DECODe<x>:ARINC429:SOURce

命令格式

:DECODe<x>:ARINC429:SOURce <String>

:DECODe<x>:ARINC429:SOURce?

功能描述

设置 ARINC429 协议解码的数据信源。

查询 ARINC429 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:ARINC429:SOURce CHANnel1 设置 ARINC429 协议解码数据信源为通道 1。

:DECODe1:ARINC429:SOURce? -> CHANnel1 查询 ARINC429 协议解码数据信源为通道 1。

:DECODe<x>:ARINC429:SPEEd

命令格式

:DECODe<x>:ARINC429:SPEEd <Int>

:DECODe<x>:ARINC429:SPEEd?

功能描述

设置 ARINC429 协议解码的传输速度。

查询 ARINC429 协议解码的传输速度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 1000~100000。

返回格式

查询命令以整数形式返回 ARINC429 协议解码的传输速度。

实例说明

:DECODe1:ARINC429:SPEEd 1000 设置 ARINC429 协议解码的传输速度为 1000。

:DECODe1:ARINC429:SPEEd? -> 1000 查询 ARINC429 协议解码的传输速度为 1000。

:DECODe<x>:ARINC429:MODEl

命令格式

:DECODe<x>:ARINC429:MODEl <String>

:DECODe<x>:ARINC429:MODEl?

功能描述

设置 ARINC429 协议解码的传输模式。

查询 ARINC429 协议解码的传输模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { BNR | BCD }。

返回格式

查询命令返回 BNR 或 BCD。

实例说明

:DECODe1:ARINC429:MODEl BNR 设置 ARINC429 协议解码传输模式为 BNR。

:DECODe1:ARINC429:MODEl? -> BNR 查询 ARINC429 协议解码传输模式为 BNR。

:DECODe<x>:ARINC429:HIGHthresh

命令格式

:DECODe<x>:ARINC429:HIGHthresh <Double>

:DECODe<x>:ARINC429:HIGHthresh?

功能描述

设置 ARINC429 协议解码的高阈值。

查询 ARINC429 协议解码的高阈值。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 阈值，范围-100.00~100.00，单位 V。

注：高阈值取值范围会受低阈值阈值范围影响。

返回格式

查询命令以实型形式返回 ARINC429 协议解码的高阈值大小。

实例说明

:DECODe1:ARINC429:HIGHthresh -1　　设置 ARINC429 协议解码的高阈值为-1V。

:DECODe1:ARINC429:HIGHthresh? -> -1.00　　查询 ARINC429 协议解码的高阈值为-1V。

:DECODe<x>:ARINC429:LOWThresh

命令格式

:DECODe<x>:ARINC429:LOWThresh <Double>

:DECODe<x>:ARINC429:LOWThresh?

功能描述

设置 ARINC429 协议解码的低阈值。

查询 ARINC429 协议解码的低阈值。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 阈值，范围-100.00~100.00，单位 V。

注：低阈值取值范围会受高阈值阈值范围影响。

返回格式

查询命令以实型形式返回 ARINC429 协议解码的低阈值大小。

实例说明

:DECODe1:ARINC429:LOWThresh -1　　设置 ARINC429 协议解码的低阈值为-1V。

:DECODe1:ARINC429:LOWThresh? -> -1.00　　查询 ARINC429 协议解码的低阈值为-1V。

:DECODe<x>:CAN:SOURce

命令格式

:DECODe<x>:CAN:SOURce <String>

:DECODe<x>:CAN:SOURce?

功能描述

设置 CAN 协议解码的信源。

查询 CAN 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:CAN:SOURce CHANnel1　　设置 CAN 协议解码信源为通道 1。

:DECODe1:CAN:SOURce? -> CHANnel1　　查询 CAN 协议解码信源为通道 1。

:DECODe<x>:CAN:BUSType

命令格式

:DECODe<x>:CAN:BUSType <String>

:DECODe<x>:CAN:BUSType?

功能描述

设置 CAN 协议解码的总线类型。

查询 CAN 协议解码的总线类型。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CANL | CANH | CANDiff }。

返回格式

查询命令返回 CANL、CANH 或 CANDiff。

实例说明

:DECODe1:CAN:BUSType CANL	设置 CAN 协议解码的总线类型为 CANL。
:DECODe1:CAN:BUSType? -> CANL	查询 CAN 协议解码的总线类型为 CANL。

:DECODe<x>:CAN:BAUDrate

命令格式

:DECODe<x>:CAN:BAUDrate <Double>

:DECODe<x>:CAN:BAUDrate?

功能描述

设置 CAN 协议解码的波特率。

查询 CAN 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 波特率，范围 1.00k~10000.00k。

返回格式

查询命令以实型形式返回 CAN 协议解码的波特率大小。

实例说明

:DECODe1:CAN:BAUDrate 500	设置 CAN 协议解码的波特率为 500k。
:DECODe1:CAN:BAUDrate? -> 500.00	查询 CAN 协议解码的波特率为 500k。

:DECODe<x>:CAN:SAMPlepos

命令格式

:DECODe<x>:CAN:SAMPlepos <Int>

:DECODe<x>:CAN:SAMPlepos?

功能描述

设置 CAN 协议解码的采样位置。

查询 CAN 协议解码的采样位置。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 50~95，单位%。

返回格式

查询命令以整数形式返回 CAN 协议解码的采样位置。

实例说明

:DECODe1:CAN:SAMPlepos 70	设置 CAN 协议解码的采样位置为 70%。
:DECODe1:CAN:SAMPlepos? -> 70	查询 CAN 协议解码的采样位置为 70%。

:DECODe<x>:CANFd:SOURce

命令格式

:DECODe<x>:CANFd:SOURce <String>

:DECODe<x>:CANFd:SOURce?

功能描述

设置 CAN-FD 协议解码的信源。

查询 CAN-FD 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:CANFd:SOURce CHANnel1　　设置 CAN-FD 协议解码信源为通道 1。

:DECODe1:CANFd:SOURce? -> CHANnel1　　查询 CAN-FD 协议解码信源为通道 1。

:DECODe<x>:CANFd:BUSType

命令格式

:DECODe<x>:CANFd:BUSType <String>

:DECODe<x>:CANFd:BUSType?

功能描述

设置 CAN-FD 协议解码的总线类型。

查询 CAN-FD 协议解码的总线类型。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CANL | CANH | CANDiff }。

返回格式

查询命令返回 CANL、CANH 或 CANDiff。

实例说明

:DECODe1:CANFd:BUSType CANL 设置 CAN-FD 协议解码的总线类型为 CANL。

:DECODe1:CANFd:BUSType? -> CANL 查询 CAN-FD 协议解码的总线类型为 CANL。

:DECODe<x>:CANFd:BAUDrate

命令格式

:DECODe<x>:CANFd:BAUDrate <Double>

:DECODe<x>:CANFd:BAUDrate?

功能描述

设置 CAN-FD 协议解码的波特率。

查询 CAN-FD 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 波特率，范围 1.00k~10000.00k。

返回格式

查询命令以实型形式返回 CANFd 协议解码的波特率大小。

实例说明

:DECODe1:CANFd:BAUDrate 500	设置 CAN-FD 协议解码的波特率为 500k。
:DECODe1:CANFd:BAUDrate? -> 500.00	查询 CAN-FD 协议解码的波特率为 500k。

:DECODe<x>:CANFd:FDBAud

命令格式

:DECODe<x>:CANFd:FDBAud <Double>

:DECODe<x>:CANFd:FDBAud?

功能描述

设置 CAN-FD 协议解码的 FD 波特率。

查询 CAN-FD 协议解码的 FD 波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> FD 波特率，范围 1.00k~10000.00k。

注：FD 波特率设置始终要大于或等于波特率设置，相关指令:DECODe<x>:CANFd:BAUDrate。

返回格式

查询命令以实型形式返回 CAN-FD 协议解码的波特率大小。

实例说明

:DECODe1:CANFd:FDBAud 500	设置 CAN-FD 协议解码的 FD 波特率为 500k。
:DECODe1:CANFd:FDBAud? -> 500.00	查询 CAN-FD 协议解码的 FD 波特率为 500k。

:DECODe<x>:CANFd:SAMPlepos

命令格式

:DECODe<x>:CANFd:SAMPlepos <Int>

:DECODe<x>:CANFd:SAMPlepos?

功能描述

设置 CAN-FD 协议解码的采样位置。

查询 CAN-FD 协议解码的采样位置。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 50~95，单位%。

返回格式

查询命令以整数形式返回 CAN-FD 协议解码的采样位置。

实例说明

:DECODe1:CANFd:SAMPlepos 70　　设置 CAN-FD 协议解码的采样位置为 70%。

:DECODe1:CANFd:SAMPlepos? -> 70　　查询 CAN-FD 协议解码的采样位置为 70%。

:DECODe<x>:DALI:SOURce

命令格式

:DECODe<x>:DALI:SOURce <String>

:DECODe<x>:DALI:SOURce?

功能描述

设置 DALI 协议解码的信源。

查询 DALI 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:DALI:SOURce CHANnel1	设置 DALI 协议解码信源为通道 1。
:DECODe1:DALI:SOURce? -> CHANnel1	查询 DALI 协议解码信源为通道 1。

:DECODe<x>:DHT11:SOURce

命令格式

:DECODe<x>:DHT11:SOURce <String>

:DECODe<x>:DHT11:SOURce?

功能描述

设置 DHT11 协议解码的信源。

查询 DHT11 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:DHT11:SOURce CHANnel1	设置 DHT11 协议解码信源为通道 1。
:DECODe1:DHT11:SOURce? -> CHANnel1	查询 DHT11 协议解码信源为通道 1。

:DECODe<x>:DMANchester:SOURce

命令格式

:DECODe<x>:DMANchester:SOURce <String>

:DECODe<x>:DMANchester:SOURce?

功能描述

设置 DManchester 协议解码的信源。

查询 DManchester 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:DMANchester:SOURce CHANnel1 设置 DManchester 协议解码信源为通道 1。

:DECODe1:DMANchester:SOURce? -> CHANnel1 查询 DManchester 协议解码信源为通道 1。

:DECODe<x>:DMANchester:BITLen

命令格式

:DECODe<x>:DMANchester:BITLen <String>

:DECODe<x>:DMANchester:BITLen?

功能描述

设置 DManchester 协议解码的位时长。

查询 DManchester 协议解码的位时长。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 100ns~1ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 DManchester 协议解码的位时长。

实例说明

:DECODe1:DMANchester:BITLen 2e-4 设置 DManchester 协议解码的位时长为 200us。

:DECODe1:DMANchester:BITLen? -> 1.000ms 查询 DManchester 协议解码的位时长为 1ms。

:DECODe<x>:DMANchester:FRAMestart

命令格式

:DECODe<x>:DMANchester:FRAMestart <String>

:DECODe<x>:DMANchester:FRAMestart?

功能描述

设置 DManchester 协议解码的包起始位。

查询 DManchester 协议解码的包起始位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 0 | 1}。

返回格式

查询命令返回 0 或 1。

实例说明

:DECODe1:DMANchester:FRAMestart 1　　设置 DManchester 协议解码的包起始位为 1。

:DECODe1:DMANchester:FRAMestart? -> 1　　查询 DManchester 协议解码的包起始位为 1。

:DECODe<x>:DMANchester:DATAstart

命令格式

:DECODe<x>:DMANchester:DATAstart <String>

:DECODe<x>:DMANchester:DATAstart?

功能描述

设置 DManchester 协议解码的帧起始位。

查询 DManchester 协议解码的帧起始位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 0 | 1 | NONE }。

返回格式

查询命令返回 0，1 或 NONE。

实例说明

:DECODe1:DMANchester:DATAstart NONE 设置 DManchester 协议解码的帧起始位为 NONE。

:DECODe1:DMANchester:DATAstart? -> NONE 查询 DManchester 协议解码的帧起始位为 NONE。

:DECODe<x>:DMANchester:TRANsmode

命令格式

:DECODe<x>:DMANchester:TRANsmode <String>

:DECODe<x>:DMANchester:TRANsmode?

功能描述

设置 DManchester 协议解码的传输模式。

查询 DManchester 协议解码的传输模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { LSB | MSB}。

返回格式

查询命令返回 LSB 或 MSB。

实例说明

:DECODe1:DMANchester:TRANsmode LSB 设置 DManchester 协议解码的传输模式为 LSB。

:DECODe1:DMANchester:TRANsmode? -> LSB 查询 DManchester 协议解码的传输模式为 LSB。

:DECODe<x>:DMANchester:BITCnt

命令格式

:DECODe<x>:DMANchester:BITCnt <Int>

:DECODe<x>:DMANchester:BITCnt?

功能描述

设置 DManchester 协议解码的数据位宽。

查询 DManchester 协议解码的数据位宽。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 1~32。

返回格式

查询命令以整数形式返回 DManchester 协议解码的数据位宽。

实例说明

:DECODe1:DMANchester:BITCnt 8	设置 DManchester 协议解码的数据位宽为 8。
:DECODe1:DMANchester:BITCnt? -> 8	查询 DManchester 协议解码的数据位宽为 8

:DECODe<x>:DMANchester:PARItybit

命令格式

:DECODe<x>:DMANchester:PARItybit <String>

:DECODe<x>:DMANchester:PARItybit?

功能描述

设置 DManchester 协议解码的校验位。

查询 DManchester 协议解码的校验位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { ODD | EVEN | NONE }，分别表示奇校验、偶校验、无。

返回格式

查询命令返回 ODD，EVEN 或 NONE。

实例说明

:DECODe1:DMANchester:PARItybit NONE 设置 DManchester 协议解码的校验位为 NONE。

:DECODe1:DMANchester:PARItybit? -> NONE 查询 DManchester 协议解码的校验位为 NONE。

:DECODe<x>:DMANchester:IGNOrebits

命令格式

:DECODe<x>:DMANchester:IGNOrebits <Int>

:DECODe<x>:DMANchester:IGNOrebits?

功能描述

设置 DManchester 协议解码的起始忽略。

查询 DManchester 协议解码的起始忽略。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 DManchester 协议解码的起始忽略。

实例说明

:DECODe1:DMANchester:IGNOrebits 8 设置 DManchester 协议解码的起始忽略为 8。

:DECODe1:DMANchester:IGNOrebits? -> 8 查询 DManchester 协议解码的起始忽略为 8。

:DECODe<x>:DMANchester:SERERatebits

命令格式

:DECODe<x>:DMANchester:SERERatebits <Int>

:DECODe<x>:DMANchester:SERERatebits?

功能描述

设置 DManchester 协议解码的间隔位数。

查询 DManchester 协议解码的间隔位数。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 DManchester 协议解码的间隔位数。

实例说明

:DECODe1:DMANchester:SERERatebits 8 设置 DManchester 协议解码的间隔位数为 8。

:DECODe1:DMANchester:SERERatebits? -> 8 查询 DManchester 协议解码的间隔位数为 8。

:DECODe<x>:DMX512:SOURce

命令格式

:DECODe<x>:DMX512:SOURce <String>

:DECODe<x>:DMX512:SOURce?

功能描述

设置 DMX512 协议解码的信源。

查询 DMX512 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:DMX512:SOURce CHANnel1 设置 DMX512 协议解码信源为通道 1。

:DECODe1:DMX512:SOURce? -> CHANnel1 查询 DMX512 协议解码信源为通道 1。

:DECODe<x>:DMX512:BUSType

命令格式

:DECODe<x>:DMX512:BUSType <String>

:DECODe<x>:DMX512:BUSType?

功能描述

设置 DMX512 协议解码的总线类型。

查询 DMX512 协议解码的总线类型。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { - | + | COMMon}。

返回格式

查询命令返回-，+或 COMMon。

实例说明

:DECODe1:DMX512:BUSType + 设置 DMX512 协议解码的总线类型为+。

:DECODe1:DMX512:BUSType? -> + 查询 DMX512 协议解码的总线类型为+。

:DECODe<x>:DS18B20:SOURce

命令格式

:DECODe<x>:DS18B20:SOURce <String>

:DECODe<x>:DS18B20:SOURce?

功能描述

设置 DS18B20 协议解码的信源。

查询 DS18B20 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:DS18B20:SOURce CHANnel1 设置 DS18B20 协议解码信源为通道 1。

:DECODe1:DS18B20:SOURce? -> CHANnel1 查询 DS18B20 协议解码信源为通道 1。

:DECODe<x>:DS18B20:PRECision

命令格式

:DECODe<x>:DS18B20:PRECision <String>

:DECODe<x>:DS18B20:PRECision?

功能描述

设置 DS18B20 协议解码的温度分辨率。

查询 DS18B20 协议解码的温度分辨率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 12bit | 11bit | 10bit | 9bit}。

返回格式

查询命令返回 12bit，11bit，10bit 或 9bit。

实例说明

:DECODe1:DS18B20:PRECision 12bit 设置 DS18B20 协议解码的温度分辨率为 12bit。

:DECODe1:DS18B20:PRECision? -> 12bit 查询 DS18B20 协议解码的温度分辨率为 12bit。

:DECODe<x>:DSI:D+

命令格式

:DECODe<x>:DSI:D+ <String>

:DECODe<x>:DSI:D+?

功能描述

设置 MIPI-DSI 协议解码的 D+的信源。

查询 MIPI-DSI 协议解码的 D+的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:DSI:D+ CHANnel1	设置 MIPI-DSI 协议解码的 D+信源为通道 1。
:DECODe1:DSI:D+? -> CHANnel1	查询 MIPI-DSI 协议解码的 D+信源为通道 1。

:DECODe<x>:DSI:D-

命令格式

:DECODe<x>:DSI:D- <String>

:DECODe<x>:DSI:D-?

功能描述

设置 MIPI-DSI 协议解码的 D-的信源。

查询 MIPI-DSI 协议解码的 D-的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:DSI:D- CHANnel1　　设置 MIPI-DSI 协议解码的 D-信源为通道 1。

:DECODe1:DSI:D-? -> CHANnel1　　查询 MIPI-DSI 协议解码的 D-信源为通道 1。

:DECODe<x>:FLEXray:TXD

命令格式

:DECODe<x>:FLEXray:TXD <String>

:DECODe<x>:FLEXray:TXD?

功能描述

设置 FlexRay 协议解码的 TxD 的信源。

查询 FlexRay 协议解码的 TxD 的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:FLEXray:TXD CHANnel1　　设置 FlexRay 协议解码的 TxD 信源为通道 1。

:DECODe1:FLEXray:TXD? -> CHANnel1　查询 FlexRay 协议解码的 TxD 信源为通道 1。

:DECODe<x>:FLEXray:SPEEd

命令格式

:DECODe<x>:FLEXray:SPEEd <String>

:DECODe<x>:FLEXray:SPEEd?

功能描述

设置 FlexRay 协议解码的通信速率。

查询 FlexRay 协议解码的通信速率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 2.5Mbps | 5Mbps | 10Mbps }。

返回格式

查询命令返回 2.5Mbps、5Mbps 或 10Mbps。

实例说明

:DECODe1:FLEXray:SPEEd 10Mbps 设置 FlexRay 协议解码的通信速率为 10Mbps。

:DECODe1:FLEXray:SPEEd? -> 10Mbps 查询 FlexRay 协议解码的通信速率为 10Mbps。

:DECODe<x>:FLEXray:CHANnel

命令格式

:DECODe<x>:FLEXray:CHANnel <String>

:DECODe<x>:FLEXray:CHANnel?

功能描述

设置 FlexRay 协议解码的信道。

查询 FlexRay 协议解码的信道。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { A | B },分别表示 ChannelA、ChannelB。

返回格式

查询命令返回 A 或 B。

实例说明

:DECODe1:FLEXray:CHANnel A	设置 FlexRay 协议解码的信道为 ChannelA。
:DECODe1:FLEXray:CHANnel? -> A	查询 FlexRay 协议解码的信道为 ChannelA。

:DECODe<x>:HDQ:SOURce

命令格式

:DECODe<x>:HDQ:SOURce <String>

:DECODe<x>:HDQ:SOURce?

功能描述

设置 HDQ 协议解码的信源。

查询 HDQ 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:HDQ:SOURce CHANnel1	设置 HDQ 协议解码信源为通道 1。
:DECODe1:HDQ:SOURce? -> CHANnel1	查询 HDQ 协议解码信源为通道 1。

:DECODe<x>:HDQ:DATAlen

命令格式

:DECODe<x>:HDQ:DATAlen <String>

:DECODe<x>:HDQ:DATAlen?

功能描述

设置 HDQ 协议解码的数据长度。

查询 HDQ 协议解码的数据长度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 8bit | 16bit }。

返回格式

查询命令返回 8bit 或 16bit。

实例说明

:DECODe1:HDQ:DATAlen 8bit	设置 HDQ 协议解码数据长度为 8 位。
:DECODe1:HDQ:DATAlen? -> 8bit	查询 HDQ 协议解码数据长度为 8 位。

:DECODe<x>:I3C:SCL

命令格式

:DECODe<x>:I3C:SCL <String>

:DECODe<x>:I3C:SCL?

功能描述

设置 I3C 协议解码的时钟信源。

查询 I3C 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:I3C:SCL CHANnel1	设置 I3C 协议解码的时钟信源为通道 1。
:DECODe1:I3C:SCL? -> CHANnel1	查询 I3C 协议解码的时钟信源为通道 1。

:DECODe<x>:I3C:SDA

命令格式

:DECODe<x>:I3C:SDA <String>

:DECODe<x>:I3C:SDA?

功能描述

设置 I3C 协议解码的数据信源。

查询 I3C 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:I3C:SDA CHANnel1	设置 I3C 协议解码的数据信源为通道 1。
:DECODe1:I3C:SDA? -> CHANnel1	查询 I3C 协议解码的数据信源为通道 1。

:DECODe<x>:I3C:MODE

命令格式

:DECODe<x>:I3C:MODE <String>

:DECODe<x>:I3C:MODE?

功能描述

设置 I3C 协议解码的传输模式。

查询 I3C 协议解码的传输模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { SDR }。

返回格式

查询命令返回 SDR。

实例说明

:DECODe1:I3C:MODE SDR　　设置 I3C 协议解码的传输模式为 SDR。

:DECODe1:I3C:MODE? -> SDR　　查询 I3C 协议解码的传输模式为 SDR。

:DECODe<x>:I3C:ADDRtype

命令格式

:DECODe<x>:I3C:ADDRtype <String>

:DECODe<x>:I3C:ADDRtype?

功能描述

设置 I3C 协议解码的地址类型。

查询 I3C 协议解码的地址类型。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 7bit | 8bit }。

返回格式

查询命令返回 7bit 或 8bit。

实例说明

:DECODe1:I3C:ADDRtype 8bit	设置 I3C 协议解码的地址类型为 8 位。
:DECODe1:I3C:ADDRtype? -> 8bit	查询 I3C 协议解码的地址类型为 8 位。

:DECODe<x>:IIC:SCL

命令格式

:DECODe<x>:IIC:SCL <String>

:DECODe<x>:IIC:SCL?

功能描述

设置 IIC 协议解码的时钟信源。

查询 IIC 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:IIC:SCL CHANnel1	设置 IIC 协议解码的时钟信源为通道 1。
:DECODe1:IIC:SCL? -> CHANnel1	查询 IIC 协议解码的时钟信源为通道 1。

:DECODe<x>:IIC:SDA

命令格式

:DECODe<x>:IIC:SDA <String>

:DECODe<x>:IIC:SDA?

功能描述

设置 IIC 协议解码的数据信源。

查询 IIC 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:IIC:SDA CHANnel1	设置 IIC 协议解码的数据信源为通道 1。
:DECODe1:IIC:SDA? -> CHANnel1	查询 IIC 协议解码的数据信源为通道 1。

:DECODe<x>:IIC:ADDRtype

命令格式

:DECODe<x>:IIC:ADDRtype <String>

:DECODe<x>:IIC:ADDRtype?

功能描述

设置 IIC 协议解码的地址类型。

查询 IIC 协议解码的地址类型。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 7bit | 8bit | 10bit }。

返回格式

查询命令返回 7bit、8bit 或 10bit。

实例说明

:DECODe1:IIC:ADDRtype 8bit	设置 IIC 协议解码的地址类型为 8 位。
:DECODe1:IIC:ADDRtype? -> 8bit	查询 IIC 协议解码的地址类型为 8 位。

:DECODe<x>:IIC:SPEEd

命令格式

:DECODe<x>:IIC:SPEEd <String>

:DECODe<x>:IIC:SPEEd?

功能描述

设置 IIC 协议解码的传输速度。

查询 IIC 协议解码的传输速度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { NORMal | HIGH}。

返回格式

查询命令返回 NORMal 或 HIGH。

实例说明

:DECODe1:IIC:SPEEd NORMal	设置 IIC 协议解码的传输速度为 NORMal。
:DECODe1:IIC:SPEEd? -> NORMal	查询 IIC 协议解码的传输速度为 NORMal。

:DECODe<x>:IICDev:SCL

命令格式

:DECODe<x>:IICDev:SCL <String>

:DECODe<x>:IICDev:SCL?

功能描述

设置 IIC Device 协议解码的时钟信源。

查询 IIC Device 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:IICDev:SCL CHANnel1　　设置 IIC Device 协议解码的时钟信源为通道 1。

:DECODe1:IICDev:SCL? -> CHANnel1　　查询 IIC Device 协议解码的时钟信源为通道 1。

:DECODe<x>:IICDev:SDA

命令格式

:DECODe<x>:IICDev:SDA <String>

:DECODe<x>:IICDev:SDA?

功能描述

设置 IIC Device 协议解码的数据信源。

查询 IIC Device 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:IICDev:SDA CHANnel1　　设置 IIC Device 协议解码的数据信源为通道 1。

:DECODe1:IICDev:SDA? -> CHANnel1　　查询 IIC Device 协议解码的数据信源为通道 1。

:DECODe<x>:IICDev:DEVIce

命令格式

:DECODe<x>:IICDev:DEVIce <String>

:DECODe<x>:IICDev:DEVIce?

功能描述

设置 IIC Device 协议解码的设备型号。

查询 IIC Device 协议解码的设备型号。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { NAU8810 | NAU8811 | NAU8812 | NAU8814 | NAU88C10 | NAU8820 | NAU8822 | NAU88C22 | NAU88L24 | NAU88L25 | NAU8401 | NAU8402 | NAU8501 | NAU8502 | NAU85L40 }。

返回格式

查询命令返回 NAU8810、NAU8811、NAU8812、NAU8814、NAU88C10、NAU8820、NAU8822、NAU88C22、NAU88L24、NAU88L25、NAU8401、NAU8402、NAU8501、NAU8502 或 NAU85L40。

实例说明

:DECODe1:IICDev:DEVIce NAU8810　　设置 IIC Device 协议解码的设备型号为 NAU8810。

:DECODe1:IICDev:DEVIce? -> NAU8810　　查询 IIC Device 协议解码的设备型号为 NAU8810。

:DECODe<x>:IIS:SCK

命令格式

:DECODe<x>:IIS:SCK <String>

:DECODe<x>:IIS:SCK?

功能描述

设置 IIS 协议解码的时钟信源。

查询 IIS 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:IIS:SCK CHANnel1　　设置 IIS 协议解码的时钟信源为通道 1。

:DECODe1:IIS:SCK? -> CHANnel1　　查询 IIS 协议解码的时钟信源为通道 1。

:DECODe<x>:IIS:WS

命令格式

:DECODe<x>:IIS:WS <String>

:DECODe<x>:IIS:WS?

功能描述

设置 IIS 协议解码的字选择信源。

查询 IIS 协议解码的字选择信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:IIS:WS CHANnel1	设置 IIS 协议解码的字选择信源为通道 1。
:DECODe1:IIS:WS? -> CHANnel1	查询 IIS 协议解码的字选择信源为通道 1。

:DECODe<x>:IIS:SD

命令格式

:DECODe<x>:IIS:SD <String>

:DECODe<x>:IIS:SD?

功能描述

设置 IIS 协议解码的数据信源。

查询 IIS 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:IIS:SD CHANnel1	设置 IIS 协议解码的数据信源为通道 1。
:DECODe1:IIS:SD? -> CHANnel1	查询 IIS 协议解码的数据信源为通道 1。

:DECODe<x>:IIS:TYPE

命令格式

:DECODe<x>:IIS:TYPE <String>

:DECODe<x>:IIS:TYPE?

功能描述

设置 IIS 协议解码的协议格式。

查询 IIS 协议解码的协议格式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { IIS | LEFT | RIGHt}。

返回格式

查询命令返回 IIS、LEFT 或 RIGHt。

实例说明

:DECODe1:IIS:TYPE IIS　　设置 IIS 协议解码的协议格式为 IIS。

:DECODe1:IIS:TYPE? -> IIS　　查询 IIS 协议解码的协议格式为 IIS。

:DECODe<x>:IIS:BITCnt

命令格式

:DECODe<x>:IIS:BITCnt <Int>

:DECODe<x>:IIS:BITCnt?

功能描述

设置 IIS 协议解码的数据长度。

查询 IIS 协议解码的数据长度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 4~32。

返回格式

查询命令以整数形式返回 IIS 协议解码的数据长度。

实例说明

:DECODe1:IIS:BITCnt 8	设置 IIS 协议解码的数据长度为 8。
:DECODe1:IIS:BITCnt? -> 8	查询 IIS 协议解码的数据长度为 8。

:DECODe<x>:ISO7816:RST

命令格式

:DECODe<x>:ISO7816:RST <String>

:DECODe<x>:ISO7816:RST?

功能描述

设置 ISO7816 协议解码的复位信源。

查询 ISO7816 协议解码的复位信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:ISO7816:RST CHANnel1　　设置 ISO7816 协议解码的复位信源为通道 1。

:DECODe1:ISO7816:RST? -> CHANnel1　查询 ISO7816 协议解码的复位信源为通道 1。

:DECODe<x>:ISO7816:DAT

命令格式

:DECODe<x>:ISO7816:DAT <String>

:DECODe<x>:ISO7816:DAT?

功能描述

设置 ISO7816 协议解码的数据信源。

查询 ISO7816 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:ISO7816:DAT CHANnel1 设置 ISO7816 协议解码的数据信源为通道 1。

:DECODe1:ISO7816:DAT? -> CHANnel1 查询 ISO7816 协议解码的数据信源为通道 1。

:DECODe<x>:ISO7816:BAUDrate

命令格式

:DECODe<x>:ISO7816:BAUDrate <Double>

:DECODe<x>:ISO7816:BAUDrate?

功能描述

设置 ISO7816 协议解码的波特率。

查询 ISO7816 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 波特率，范围 1.00k~15.00k。

返回格式

查询命令以实型形式返回 ISO7816 协议解码的波特率大小。

实例说明

:DECODe1:ISO7816:BAUDrate 15	设置 ISO7816 协议解码的波特率为 15k。
:DECODe1:ISO7816:BAUDrate? -> 15.00	查询 ISO7816 协议解码的波特率为 15k。

:DECODe<x>:ISO7816:CRCLen

命令格式

:DECODe<x>:ISO7816:CRCLen <String>

:DECODe<x>:ISO7816:CRCLen?

功能描述

设置 ISO7816 协议解码的 CRC 长度。

查询 ISO7816 协议解码的 CRC 长度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 1 | 2}。

返回格式

查询命令返回 1 或 2。

实例说明

:DECODe1:ISO7816:CRCLen 1	设置 ISO7816 协议解码的 CRC 长度为 1。
:DECODe1:ISO7816:CRCLen? -> 1	查询 ISO7816 协议解码的 CRC 长度为 1。

:DECODe<x>:LIN:SOURce

命令格式

:DECODe<x>:LIN:SOURce <String>

:DECODe<x>:LIN:SOURce?

功能描述

设置 LIN 协议解码的信源。

查询 LIN 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:LIN:SOURce CHANnel1	设置 LIN 协议解码信源为通道 1。
:DECODe1:LIN:SOURce? -> CHANnel1	查询 LIN 协议解码信源为通道 1。

:DECODe<x>:LIN:BAUDrate

命令格式

:DECODe<x>:LIN:BAUDrate <Int>

:DECODe<x>:LIN:BAUDrate?

功能描述

设置 LIN 协议解码的波特率。

查询 LIN 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 波特率，范围 1000~20000。

返回格式

查询命令以整型形式返回 LIN 协议解码的波特率大小。

实例说明

:DECODe1:LIN:BAUDrate 9600	设置 LIN 协议解码的波特率为 9600。
:DECODe1:LIN:BAUDrate? -> 9600	查询 LIN 协议解码的波特率为 9600。

:DECODe<x>:LIN:VERSion

命令格式

:DECODe<x>:LIN:VERSion <String>

:DECODe<x>:LIN:VERSion?

功能描述

设置 LIN 协议解码的版本。

查询 LIN 协议解码的版本。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { LIN1.3 | LIN2}。

返回格式

查询命令返回 LIN1.3 或 LIN2。

实例说明

:DECODe1:LIN:VERSion LIN1.3　　设置 LIN 协议解码的版本为 LIN1.3。

:DECODe1:LIN:VERSion? -> LIN1.3　　查询 LIN 协议解码的版本为 LIN1.3。

:DECODe<x>:MANChester:SOURce

命令格式

:DECODe<x>:MANChester:SOURce <String>

:DECODe<x>:MANChester:SOURce?

功能描述

设置 Manchester 协议解码的信源。

查询 Manchester 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:MANChester:SOURce CHANnel1 设置 Manchester 协议解码信源为通道 1。

:DECODe1:MANChester:SOURce? -> CHANnel1 查询 Manchester 协议解码信源为通道 1。

:DECODe<x>:MANChester:TYPE

命令格式

:DECODe<x>:MANChester:TYPE <String>

:DECODe<x>:MANChester:TYPE?

功能描述

设置 Manchester 协议解码的编码模式。

查询 Manchester 协议解码的编码模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { GE | IEEE}。

返回格式

查询命令返回 GE 或 IEEE。

实例说明

:DECODe1:MANChester:TYPE GE 设置 Manchester 协议解码的编码模式为 GE。

:DECODe1:MANChester:TYPE? -> GE 查询 Manchester 协议解码的编码模式为 GE。

:DECODe<x>:MANChester:BITLen

命令格式

:DECODe<x>:MANChester:BITLen <String>

:DECODe<x>:MANChester:BITLen?

功能描述

设置 Manchester 协议解码的位时长。

查询 Manchester 协议解码的位时长。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 100ns~1ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 Manchester 协议解码的位时长。

实例说明

:DECODe1:MANChester:BITLen 2e-4 设置 Manchester 协议解码的位时长为 200us。

:DECODe1:MANChester:BITLen? -> 1.000ms 查询 Manchester 协议解码的位时长为 1ms。

:DECODe<x>:MANChester:FRAMestart

命令格式

:DECODe<x>:MANChester:FRAMestart <String>

:DECODe<x>:MANChester:FRAMestart?

功能描述

设置 Manchester 协议解码的包起始位。

查询 Manchester 协议解码的包起始位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 0 | 1}。

返回格式

查询命令返回 0 或 1。

实例说明

:DECODe1:MANChester:FRAMestart 1　　设置 Manchester 协议解码的包起始位为 1。

:DECODe1:MANChester:FRAMestart? -> 1　　查询 Manchester 协议解码的包起始位为 1。

:DECODe<x>:MANChester:DATAstart

命令格式

:DECODe<x>:MANChester:DATAstart <String>

:DECODe<x>:MANChester:DATAstart?

功能描述

设置 Manchester 协议解码的帧起始位。

查询 Manchester 协议解码的帧起始位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 0 | 1 | NONE }。

返回格式

查询命令返回 0，1 或 NONE。

实例说明

:DECODe1:MANChester:DATAstart NONE 设置 Manchester 协议解码的帧起始位为 NONE。

:DECODe1:MANChester:DATAstart? -> NONE 查询 Manchester 协议解码的帧起始位为 NONE。

:DECODe<x>:MANChester:TRANsmode

命令格式

:DECODe<x>:MANChester:TRANsmode <String>

:DECODe<x>:MANChester:TRANsmode?

功能描述

设置 Manchester 协议解码的传输模式。

查询 Manchester 协议解码的传输模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { LSB | MSB}。

返回格式

查询命令返回 LSB 或 MSB。

实例说明

:DECODe1:MANChester:TRANsmode LSB 设置 Manchester 协议解码的传输模式为 LSB。

:DECODe1:MANChester:TRANsmode? -> LSB 查询 Manchester 协议解码的传输模式为 LSB。

:DECODe<x>:MANChester:BITCnt

命令格式

:DECODe<x>:MANChester:BITCnt <Int>

:DECODe<x>:MANChester:BITCnt?

功能描述

设置 Manchester 协议解码的数据位宽。

查询 Manchester 协议解码的数据位宽。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 1~32。

返回格式

查询命令以整数形式返回 Manchester 协议解码的数据位宽。

实例说明

:DECODe1:MANChester:BITCnt 8	设置 Manchester 协议解码的数据位宽为 8。
:DECODe1:MANChester:BITCnt? -> 8	查询 Manchester 协议解码的数据位宽为 8

:DECODe<x>:MANChester:PARItybit

命令格式

:DECODe<x>:MANChester:PARItybit <String>

:DECODe<x>:MANChester:PARItybit?

功能描述

设置 Manchester 协议解码的校验位。

查询 Manchester 协议解码的校验位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { ODD | EVEN | NONE }，分别表示奇校验、偶校验、无。

返回格式

查询命令返回 ODD，EVEN 或 NONE。

实例说明

:DECODe1:MANChester:PARItybit NONE 设置 Manchester 协议解码的校验位为 NONE。

:DECODe1:MANChester:PARItybit? -> NONE 查询 Manchester 协议解码的校验位为 NONE。

:DECODe<x>:MANChester:IGNOrebits

命令格式

:DECODe<x>:MANChester:IGNOrebits <Int>

:DECODe<x>:MANChester:IGNOrebits?

功能描述

设置 Manchester 协议解码的起始忽略。

查询 Manchester 协议解码的起始忽略。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Manchester 协议解码的起始忽略。

实例说明

:DECODe1:MANChester:IGNOrebits 8 设置 Manchester 协议解码的起始忽略为 8。

:DECODe1:MANChester:IGNOrebits? -> 8 查询 Manchester 协议解码的起始忽略为 8。

:DECODe<x>:MANChester:SERatebits

命令格式

:DECODe<x>:MANChester:SERatebits <Int>

:DECODe<x>:MANChester:SERatebits?

功能描述

设置 Manchester 协议解码的间隔位数。

查询 Manchester 协议解码的间隔位数。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Manchester 协议解码的间隔位数。

实例说明

:DECODe1:MANChester:SERatebits 8 设置 Manchester 协议解码的间隔位数为 8。

:DECODe1:MANChester:SERatebits? -> 8 查询 Manchester 协议解码的间隔位数为 8。

:DECODe<x>:MDIO:MDC

命令格式

:DECODe<x>:MDIO:MDC <String>

:DECODe<x>:MDIO:MDC?

功能描述

设置 MDIO 协议解码的 MDC 的信源。

查询 MDIO 协议解码的 MDC 的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:MDIO:MDC CHANnel1　　设置 MDIO 协议解码的 MDC 信源为通道 1。

:DECODe1:MDIO:MDC? -> CHANnel1　　查询 MDIO 协议解码的 MDC 信源为通道 1。

:DECODe<x>:MDIO:MDIO

命令格式

:DECODe<x>:MDIO:MDIO <String>

:DECODe<x>:MDIO:MDIO?

功能描述

设置 MDIO 协议解码的 MDIO 的信源。

查询 MDIO 协议解码的 MDIO 的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:MDIO:MDIO CHANnel1 设置 MDIO 协议解码的 MDIO 信源为通道 1。

:DECODe1:MDIO:MDIO? -> CHANnel1 查询 MDIO 协议解码的 MDIO 信源为通道 1。

:DECODe<x>:MILLer:SOURce

命令格式

:DECODe<x>:MILLer:SOURce <String>

:DECODe<x>:MILLer:SOURce?

功能描述

设置 Miller 协议解码的信源。

查询 Miller 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:MILLer:SOURce CHANnel1	设置 Miller 协议解码信源为通道 1。
:DECODe1:MILLer:SOURce? -> CHANnel1	查询 Miller 协议解码信源为通道 1。

:DECODe<x>:MILLer:BITRate

命令格式

:DECODe<x>:MILLer:BITRate <Double>

:DECODe<x>:MILLer:BITRate?

功能描述

设置 Miller 协议解码的比特率。

查询 Miller 协议解码的比特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 比特率，范围 0.10k~1000000.00k。

返回格式

查询命令以实型形式返回 Miller 协议解码的比特率大小。

实例说明

:DECODe1:MILLer:BITRate 15	设置 Miller 协议解码的比特率为 15k。
:DECODe1:MILLer:BITRate? -> 15.00	查询 Miller 协议解码的比特率为 15k。

:DECODe<x>:MILLer:IDLELevel

命令格式

:DECODe<x>:MILLer:IDLELevel <String>

:DECODe<x>:MILLer:IDLELevel?

功能描述

设置 Miller 协议解码的空闲电平。

查询 Miller 协议解码的空闲电平。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { LOW | HIGH }。

返回格式

查询命令返回 LOW 或 HIGH。

实例说明

:DECODe1:MILLer:IDLELevel LOW 设置 Miller 协议解码的空闲电平为 LOW。

:DECODe1:MILLer:IDLELevel? -> LOW 查询 Miller 协议解码的空闲电平为 LOW。

:DECODe<x>:MILLer:FIRStbit

命令格式

:DECODe<x>:MILLer:FIRStbit <String>

:DECODe<x>:MILLer:FIRStbit?

功能描述

设置 Miller 协议解码的第一个比特。

查询 Miller 协议解码的第一个比特。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 0 | 1 }。

返回格式

查询命令返回 0 或 1。

实例说明

:DECODe1:MILLer:FIRStbit 0	设置 Miller 协议解码的第一个比特为 0。
:DECODe1:MILLer:FIRStbit? -> 0	查询 Miller 协议解码的第一个比特为 0。

:DECODe<x>:MILLer:DATAmode

命令格式

:DECODe<x>:MILLer:DATAmode <String>

:DECODe<x>:MILLer:DATAmode?

功能描述

设置 Miller 协议解码的数据模式。

查询 Miller 协议解码的数据模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { MSB | LSB}。

返回格式

查询命令返回 MSB 或 LSB。

实例说明

:DECODe1:MILLer:DATAmode LSB 设置 Miller 协议解码的数据模式为 LSB。

:DECODe1:MILLer:DATAmode? -> LSB 查询 Miller 协议解码的数据模式为 LSB。

:DECODe<x>:MILLer:BITCnt

命令格式

:DECODe<x>:MILLer:BITCnt <Int>

:DECODe<x>:MILLer:BITCnt?

功能描述

设置 Miller 协议解码的数据位宽。

查询 Miller 协议解码的数据位宽。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 1~32。

返回格式

查询命令以整数形式返回 Miller 协议解码的数据位宽。

实例说明

:DECODe1:MILLer:BITCnt 8	设置 Miller 协议解码的数据位宽为 8。
:DECODe1:MILLer:BITCnt? -> 8	查询 Miller 协议解码的数据位宽为 8

:DECODe<x>:MODBus:SOURce

命令格式

:DECODe<x>:MODBus:SOURce <String>

:DECODe<x>:MODBus:SOURce?

功能描述

设置 Modbus 协议解码的数据信源。

查询 Modbus 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:MODBus:SOURce CHANnel1　　设置 Modbus 协议解码数据信源为通道 1。

:DECODe1:MODBus:SOURce? -> CHANnel1　查询 Modbus 协议解码数据信源为通道 1。

:DECODe<x>:MODBus:BAUDrate

命令格式

:DECODe<x>:MODBus:BAUDrate <Int>

:DECODe<x>:MODBus:BAUDrate?

功能描述

设置 Modbus 协议解码的波特率。

查询 Modbus 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 波特率，范围 0~20000000。

返回格式

查询命令以整型形式返回 Modbus 协议解码的波特率大小。

实例说明

:DECODe1:MODBus:BAUDrate 9600	设置 Modbus 协议解码的波特率为 9600。
:DECODe1:MODBus:BAUDrate? -> 9600	查询 Modbus 协议解码的波特率为 9600。

:DECODe<x>:MODBus:PARItymode

命令格式

:DECODe<x>:MODBus:PARItymode <String>

:DECODe<x>:MODBus:PARItymode?

功能描述

设置 Modbus 协议解码的校验位。

查询 Modbus 协议解码的校验位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { ODD | EVEN | NONE }，分别表示奇校验、偶校验、无。

返回格式

查询命令返回 ODD，EVEN 或 NONE。

实例说明

:DECODe1:MODBus:PARItymode NONE 设置 Modbus 协议解码的校验位为 NONE。

:DECODe1:MODBus:PARItymode? -> NONE 查询 Modbus 协议解码的校验位为 NONE。

:DECODe<x>:MODBus:REVErse

命令格式

:DECODe<x>:MODBus:REVErse <String>

:DECODe<x>:MODBus:REVErse?

功能描述

设置 Modbus 协议解码的电平反相。

查询 Modbus 协议解码的电平反相。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:DECODe1:MODBus:REVErse FALSe 设置 Modbus 协议解码的电平反相为 FALSe。

:DECODe1:MODBus:REVErse? -> FALSe 查询 Modbus 协议解码的电平反相为 FALSe。

:DECODe<x>:MODBus:TRANmode

命令格式

:DECODe<x>:MODBus:TRANmode <String>

:DECODe<x>:MODBus:TRANmode?

功能描述

设置 Modbus 协议解码的传输模式。

查询 Modbus 协议解码的传输模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { RTU | ASCIi }。

返回格式

查询命令返回 RTU 或 ASCIi。

实例说明

:DECODe1:MODBus:TRANmode RTU 设置 Modbus 协议解码的传输模式为 RTU。

:DECODe1:MODBus:TRANmode? -> RTU 查询 Modbus 协议解码的传输模式为 RTU。

:DECODe<x>:MVB:SOURce

命令格式

:DECODe<x>:MVB:SOURce <String>

:DECODe<x>:MVB:SOURce?

功能描述

设置 MVB 协议解码的数据信源。

查询 MVB 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:MVB:SOURce CHANnel1　　设置 MVB 协议解码数据信源为通道 1。

:DECODe1:MVB:SOURce? -> CHANnel1　　查询 MVB 协议解码数据信源为通道 1。

:DECODe<x>:MVB:BAUDrate

命令格式

:DECODe<x>:MVB:BAUDrate <Double>

:DECODe<x>:MVB:BAUDrate?

功能描述

设置 MVB 协议解码的波特率。

查询 MVB 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 波特率，范围 0.00M~8.00M。

返回格式

查询命令以实型形式返回 MVB 协议解码的波特率大小。

实例说明

:DECODe1:MVB:BAUDrate 2	设置 MVB 协议解码的波特率为 2M。
:DECODe1:MVB:BAUDrate? -> 2.00	查询 MVB 协议解码的波特率为 2M。

:DECODe<x>:MVB:MEDIum

命令格式

:DECODe<x>:MVB:MEDIum <String>

:DECODe<x>:MVB:MEDIum?

功能描述

设置 MVB 协议解码的介质。

查询 MVB 协议解码的介质。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { ESD | EMD | OGF }。

返回格式

查询命令返回 ESD、EMD 或 OGF。

实例说明

:DECODe1:MVB:MEDIum ESD	设置 MVB 协议解码的介质为 ESD。
:DECODe1:MVB:MEDIum? -> ESD	查询 MVB 协议解码的介质为 ESD。

:DECODe<x>:NEC:SOURce

命令格式

:DECODe<x>:NEC:SOURce <String>

:DECODe<x>:NEC:SOURce?

功能描述

设置 NEC 协议解码的信源。

查询 NEC 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:NEC:SOURce CHANnel1	设置 NEC 协议解码信源为通道 1。
:DECODe1:NEC:SOURce? -> CHANnel1	查询 NEC 协议解码信源为通道 1。

:DECODe<x>:NEC:INVErt

命令格式

:DECODe<x>:NEC:INVErt <String>

:DECODe<x>:NEC:INVErt?

功能描述

设置 NEC 协议解码的电平反相。

查询 NEC 协议解码的电平反相。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:DECODe1:NEC:INVErt FALSe	设置 NEC 协议解码的电平反相为 FALSe。
:DECODe1:NEC:INVErt? -> FALSe	查询 NEC 协议解码的电平反相为 FALSe。

:DECODe<x>:NEC:DEMOdulate

命令格式

:DECODe<x>:NEC:DEMOdulate <String>

:DECODe<x>:NEC:DEMOdulate?

功能描述

设置 NEC 协议解码的载波解调。

查询 NEC 协议解码的载波解调。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:DECODe1:NEC:DEMOdulate FALSe	设置 NEC 协议解码的载波解调为 FALSe。
:DECODe1:NEC:DEMOdulate? -> FALSe	查询 NEC 协议解码的载波解调为 FALSe。

:DECODe<x>:PS2:CLK

命令格式

:DECODe<x>:PS2:CLK <String>

:DECODe<x>:PS2:CLK?

功能描述

设置 PS/2 协议解码的时钟信源。

查询 PS/2 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:PS2:CLK CHANnel1	设置 PS/2 协议解码的时钟信源为通道 1。
:DECODe1:PS2:CLK? -> CHANnel1	查询 PS/2 协议解码的时钟信源为通道 1。

:DECODe<x>:PS2:DAT

命令格式

:DECODe<x>:PS2:DAT <String>

:DECODe<x>:PS2:DAT?

功能描述

设置 PS/2 协议解码的数据信源。

查询 PS/2 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:PS2:DAT CHANnel1	设置 PS/2 协议解码的数据信源为通道 1。
:DECODe1:PS2:DAT? -> CHANnel1	查询 PS/2 协议解码的数据信源为通道 1。

:DECODe<x>:QC:D+

命令格式

:DECODe<x>:QC:D+ <String>

:DECODe<x>:QC:D+?

功能描述

设置 QC2.0/3.0 协议解码的 D+的信源。

查询 QC2.0/3.0 协议解码的 D+的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:QC:D+ CHANnel1	设置 QC2.0/3.0 协议解码的 D+信源为通道 1。
:DECODe1:QC:D+? -> CHANnel1	查询 QC2.0/3.0 协议解码的 D+信源为通道 1。

:DECODe<x>:QC:D-

命令格式

:DECODe<x>:QC:D- <String>

:DECODe<x>:QC:D-?

功能描述

设置 QC2.0/3.0 协议解码的 D-的信源。

查询 QC2.0/3.0 协议解码的 D-的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:QC:D- CHANnel1	设置 QC2.0/3.0 协议解码的 D-信源为通道 1。
:DECODe1:QC:D-? -> CHANnel1	查询 QC2.0/3.0 协议解码的 D-信源为通道 1。

:DECODe<x>:QC:CLASs

命令格式

:DECODe<x>:QC:CLASs <String>

:DECODe<x>:QC:CLASs?

功能描述

设置 QC2.0/3.0 协议解码的类别。

查询 QC2.0/3.0 协议解码的类别。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { A | B }，分别表示 ClassA、ClassB。

返回格式

查询命令返回 A 或 B。

实例说明

:DECODe1:QC:CLASs A	设置 QC2.0/3.0 协议解码的类别为 ClassA。
:DECODe1:QC:CLASs? -> A	查询 QC2.0/3.0 协议解码的类别为 ClassA。

:DECODe<x>:QC:LOWThres

命令格式

:DECODe<x>:QC:LOWThres <Double>

:DECODe<x>:QC:LOWThres?

功能描述

设置 QC2.0/3.0 协议解码的低阈值。

查询 QC2.0/3.0 协议解码的低阈值。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 阈值，范围 0.25~0.45，单位 V。

返回格式

查询命令以实型形式返回 QC2.0/3.0 协议解码的低阈值大小。

实例说明

:DECODe1:QC:LOWThres 0.3	设置 QC2.0/3.0 协议解码的低阈值为 0.3V。
:DECODe1:QC:LOWThres? -> 0.30	查询 QC2.0/3.0 协议解码的低阈值为 0.3V。

:DECODe<x>:QC:HIGHthres

命令格式

:DECODe<x>:QC:HIGHthres <Double>

:DECODe<x>:QC:HIGHthres?

功能描述

设置 QC2.0/3.0 协议解码的高阈值。

查询 QC2.0/3.0 协议解码的高阈值。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 阈值，范围 1.50~2.20，单位 V。

返回格式

查询命令以实型形式返回 QC2.0/3.0 协议解码的高阈值大小。

实例说明

:DECODe1:QC:HIGHthres 1.65	设置 QC2.0/3.0 协议解码的高阈值为 1.65V。
:DECODe1:QC:HIGHthres? -> 1.65	查询 QC2.0/3.0 协议解码的高阈值为 1.65V。

:DECODe<x>:RC<x>:SOURce

命令格式

:DECODe<x>:RC<x>:SOURce <String>

:DECODe<x>:RC<x>:SOURce?

功能描述

设置 RC5 和 RC6 协议解码的信源。

查询 RC5 和 RC6 协议解码的信源。

参数说明

<x1> 需要查询或改变的解码通道编号，1 或 2。

<x2> RC 的类型，范围 { 5 | 6 }。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:RC6:SOURce CHANnel1　　设置 RC6 协议解码信源为通道 1。

:DECODe1:RC6:SOURce? -> CHANnel1　　查询 RC6 协议解码信源为通道 1。

:DECODe<x>:RC6:MODE

命令格式

:DECODe<x>:RC6:MODE <String>

:DECODe<x>:RC6:MODE?

功能描述

设置 RC6 协议解码的模式。

查询 RC6 协议解码的模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { MODE0 | RC6-20 }。

返回格式

查询命令返回 MODE0 或 RC6-20。

实例说明

:DECODe1:RC6:MODE MODE0	设置 RC6 协议解码的模式为 MODE0。
:DECODe1:RC6:MODE? -> MODE0	查询 RC6 协议解码的模式为 MODE0。

:DECODe<x1>:RC<x2>:INVErt

命令格式

:DECODe<x>:RC<x>:INVErt <String>

:DECODe<x>:RC<x>:INVErt?

功能描述

设置 RC5 和 RC6 协议解码的电平反相。

查询 RC5 和 RC6 协议解码的电平反相。

参数说明

<x1> 需要查询或改变的解码通道编号，1 或 2。

<x2> RC 的类型，范围 { 5 | 6 }。

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:DECODe1:RC6:INVErt FALSe　　设置 RC6 协议解码的电平反相为 FALSe。

:DECODe1:RC6:INVErt? -> FALSe　　查询 RC6 协议解码的电平反相为 FALSe。

:DECODe<x1>:RC<x2>:DEMOdulate

命令格式

:DECODe<x>:RC<x>:DEMOdulate <String>

:DECODe<x>:RC<x>:DEMOdulate?

功能描述

设置 RC5 和 RC6 协议解码的载波解调。

查询 RC5 和 RC6 协议解码的载波解调。

参数说明

<x1> 需要查询或改变的解码通道编号，1 或 2。

<x2> RC 的类型，范围 { 5 | 6 }。

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:DECODe1:RC6:DEMOdulate FALSe　　设置 RC6 协议解码的载波解调为 FALSe。

:DECODe1:RC6:DEMOdulate? -> FALSe　　查询 RC6 协议解码的载波解调为 FALSe。

:DECODe<x>:RFFE:SCL

命令格式

:DECODe<x>:RFFE:SCL <String>

:DECODe<x>:RFFE:SCL?

功能描述

设置 MIPI-RFFE 协议解码的时钟信源。

查询 MIPI-RFFE 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:RFFE:SCL CHANnel1　　设置 MIPI-RFFE 协议解码的时钟信源为通道 1。

:DECODe1:RFFE:SCL? -> CHANnel1　　查询 MIPI-RFFE 协议解码的时钟信源为通道 1。

:DECODe<x>:RFFE:SDA

命令格式

:DECODe<x>:RFFE:SDA <String>

:DECODe<x>:RFFE:SDA?

功能描述

设置 MIPI-RFFE 协议解码的数据信源。

查询 MIPI-RFFE 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:RFFE:SDA CHANnel1　　设置 MIPI-RFFE 协议解码的数据信源为通道 1。

:DECODe1:RFFE:SDA? -> CHANnel1　　查询 MIPI-RFFE 协议解码的数据信源为通道 1。

:DECODe<x>:RFFE:CLOCk

命令格式

:DECODe<x>:RFFE:CLOCk <Double>

:DECODe<x>:RFFE:CLOCk?

功能描述

设置 MIPI-RFFE 协议解码的时钟频率。

查询 MIPI-RFFE 协议解码的时钟频率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 阈值，范围 32.00K~52000.00K，单位 Hz。

返回格式

查询命令以实型形式返回 MIPI-RFFE 协议解码的时钟频率大小。

实例说明

:DECODe1:RFFE:CLOCk 36　　设置 MIPI-RFFE 协议解码的时钟频率为 36KHz。

:DECODe1:RFFE:CLOCk? -> 36.00　　查询 MIPI-RFFE 协议解码的时钟频率为 36KHz。

:DECODe<x>:SDSD:SCK

命令格式

:DECODe<x>:SDSD:SCK <String>

:DECODe<x>:SDSD:SCK?

功能描述

设置 SD-SD 协议解码的时钟信源。

查询 SD-SD 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SDSD:SCK CHANnel1　　设置 SD-SD 协议解码的时钟信源为通道 1。

:DECODe1:SDSD:SCK? -> CHANnel1　　查询 SD-SD 协议解码的时钟信源为通道 1。

:DECODe<x>:SDSD:CMD

命令格式

:DECODe<x>:SDSD:CMD <String>

:DECODe<x>:SDSD:CMD?

功能描述

设置 SD-SD 协议解码的命令信源。

查询 SD-SD 协议解码的命令信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SDSD:CMD CHANnel1　　设置 SD-SD 协议解码的命令信源为通道 1。

:DECODe1:SDSD:CMD? -> CHANnel1　　查询 SD-SD 协议解码的命令信源为通道 1。

:DECODe<x>:SDSD:BUSType

命令格式

:DECODe<x>:SDSD:BUSType <String>

:DECODe<x>:SDSD:BUSType?

功能描述

设置 SD-SD 协议解码的总线类型。

查询 SD-SD 协议解码的总线类型。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { SD | MMC }。

返回格式

查询命令返回 SD 或 MMC。

实例说明

:DECODe1:SDSD:BUSType SD	设置 SD-SD 协议解码的总线类型为 SD。
:DECODe1:SDSD:BUSType? -> SD	查询 SD-SD 协议解码的总线类型为 SD。

:DECODe<x>:SDSPi:SCK

命令格式

:DECODe<x>:SDSPi:SCK <String>

:DECODe<x>:SDSPi:SCK?

功能描述

设置 SD-SPI 协议解码的时钟信源。

查询 SD-SPI 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SDSPi:SCK CHANnel1　　设置 SD-SPI 协议解码的时钟信源为通道 1。

:DECODe1:SDSPi:SCK? -> CHANnel1　　查询 SD-SPI 协议解码的时钟信源为通道 1。

:DECODe<x>:SDSPi:CMD

命令格式

:DECODe<x>:SDSPi:CMD <String>

:DECODe<x>:SDSPi:CMD?

功能描述

设置 SD-SPI 协议解码的命令信源。

查询 SD-SPI 协议解码的命令信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SDSPi:CMD CHANnel1　　设置 SD-SPI 协议解码的命令信源为通道 1。

:DECODe1:SDSPi:CMD? -> CHANnel1　　查询 SD-SPI 协议解码的命令信源为通道 1。

:DECODe<x>:SENT:SOURce

命令格式

:DECODe<x>:SENT:SOURce <String>

:DECODe<x>:SENT:SOURce?

功能描述

设置 SENT 协议解码的数据信源。

查询 SENT 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SENT:SOURce CHANnel1	设置 SENT 协议解码数据信源为通道 1。
:DECODe1:SENT:SOURce? -> CHANnel1	查询 SENT 协议解码数据信源为通道 1。

:DECODe<x>:SENT:PULSes

命令格式

:DECODe<x>:SENT:PULSes <String>

:DECODe<x>:SENT:PULSes?

功能描述

设置 SENT 协议解码的数据脉冲个数。

查询 SENT 协议解码的数据脉冲个数。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 1 | 2 | 3 | 4 | 5 | 6}。

返回格式

查询命令返回 1、2、3、4、5 或 6。

实例说明

:DECODe1:SENT:PULSes 1	设置 SENT 协议解码的数据脉冲个数为 1。
:DECODe1:SENT:PULSes? -> 1	查询 SENT 协议解码的数据脉冲个数为 1。

:DECODe<x>:SENT:TICK

命令格式

:DECODe<x>:SENT:TICK <String>

:DECODe<x>:SENT:TICK?

功能描述

设置 SENT 协议解码的时间片宽度。

查询 SENT 协议解码的时间片宽度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 500ns~90us，可带单位，如“600ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SENT 协议解码的时间片宽度。

实例说明

:DECODe1:SENT:TICK 2e-5	设置 SENT 协议解码的时间片宽度为 20us。
:DECODe1:SENT:TICK? -> 20.00us	查询 SENT 协议解码的时间片宽度为 20us。

:DECODe<x>:SENT:PAUSepulse

命令格式

:DECODe<x>:SENT:PAUSepulse <String>

:DECODe<x>:SENT:PAUSepulse?

功能描述

设置 SENT 协议解码使用停止脉冲状态。

查询 SENT 协议解码使用停止脉冲状态。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { OFF | ON }。

返回格式

查询命令返回 OFF 或 ON。

实例说明

:DECODe1:SENT:PAUSepulse OFF 设置 SENT 协议解码使用停止脉冲状态为 OFF。

:DECODe1:SENT:PAUSepulse? -> OFF 查询 SENT 协议解码使用停止脉冲状态为 OFF。

:DECODe<x>:SHT11:CLK

命令格式

:DECODe<x>:SHT11:CLK <String>

:DECODe<x>:SHT11:CLK?

功能描述

设置 SHT11 协议解码的时钟信源。

查询 SHT11 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SHT11:CLK CHANnel1　　设置 SHT11 协议解码的时钟信源为通道 1。

:DECODe1:SHT11:CLK? -> CHANnel1　　查询 SHT11 协议解码的时钟信源为通道 1。

:DECODe<x>:SHT11:DAT

命令格式

:DECODe<x>:SHT11:DAT <String>

:DECODe<x>:SHT11:DAT?

功能描述

设置 SHT11 协议解码的数据信源。

查询 SHT11 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SHT11:DAT CHANnel1　　设置 SHT11 协议解码的数据信源为通道 1。

:DECODe1:SHT11:DAT? -> CHANnel1　　查询 SHT11 协议解码的数据信源为通道 1。

:DECODe<x>:SHT11:SHOWorg

命令格式

:DECODe<x>:SHT11:SHOWorg <String>

:DECODe<x>:SHT11:SHOWorg?

功能描述

设置 SHT11 协议解码是否显示原始数据。

查询 SHT11 协议解码是否显示原始数据。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { NO | YES }。

返回格式

查询命令返回 NO 或 YES。

实例说明

:DECODe1:SHT11:SHOWorg NO　　设置 SHT11 协议解码显示原始数据为 NO。

:DECODe1:SHT11:SHOWorg? -> NO　　查询 SHT11 协议解码显示原始数据为 NO。

:DECODe<x>:SHT11:VDD

命令格式

:DECODe<x>:SHT11:VDD <String>

:DECODe<x>:SHT11:VDD?

功能描述

设置 SHT11 协议解码的工作电压。

查询 SHT11 协议解码的工作电压。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 5.0V | 4.0V | 3.5V | 3.0V | 2.5V }。

返回格式

查询命令返回 5.0V、4.0V、3.5V、3.0V 或 2.5V。

实例说明

:DECODe1:SHT11:VDD 4.0V 设置 SHT11 协议解码的工作电压为 4.0V。

:DECODe1:SHT11:VDD? -> 4.0V 查询 SHT11 协议解码的工作电压为 4.0V。

:DECODe<x>:SHT11:ADC

命令格式

:DECODe<x>:SHT11:ADC <String>

:DECODe<x>:SHT11:ADC?

功能描述

设置 SHT11 协议解码的 ADC 分辨率。

查询 SHT11 协议解码的 ADC 分辨率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 12/14bit | 8/12bit }。

返回格式

查询命令返回 12/14bit 或 8/12bit。

实例说明

:DECODe1:SHT11:ADC 8/12bit	设置 SHT11 协议解码的 ADC 分辨率为 8/12bit。
:DECODe1:SHT11:ADC? -> 8/12bit	查询 SHT11 协议解码的 ADC 分辨率为 8/12bit。

:DECODe<x>:SHT11:TEMPunit

命令格式

:DECODe<x>:SHT11:TEMPunit <String>

:DECODe<x>:SHT11:TEMPunit?

功能描述

设置 SHT11 协议解码的温度单位。

查询 SHT11 协议解码的温度单位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { C | F }，分别表示摄氏度℃、华氏度℉。

返回格式

查询命令返回 C 或 F。

实例说明

:DECODe1:SHT11:TEMPunit C	设置 SHT11 协议解码的温度单位为摄氏度℃。
:DECODe1:SHT11:TEMPunit? -> C	查询 SHT11 协议解码的温度单位为摄氏度℃。

:DECODe<x>:SHT11:TP

命令格式

:DECODe<x>:SHT11:TP <Int>

:DECODe<x>:SHT11:TP?

功能描述

设置 SHT11 协议解码的湿度补偿。

查询 SHT11 协议解码的湿度补偿。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 湿度补偿，范围-20~100，单位℃。

返回格式

查询命令以整型形式返回 SHT11 协议解码的湿度补偿大小。

实例说明

:DECODe1:SHT11:TP 25	设置 SHT11 协议解码的湿度补偿为 25℃。
:DECODe1:SHT11:TP? ->25	查询 SHT11 协议解码的湿度补偿为 25℃。

:DECODe<x>:SPC:SOURce

命令格式

:DECODe<x>:SPC:SOURce <String>

:DECODe<x>:SPC:SOURce?

功能描述

设置 SPC 协议解码的数据信源。

查询 SPC 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SPC:SOURce CHANnel1　　设置 SPC 协议解码数据信源为通道 1。

:DECODe1:SPC:SOURce? -> CHANnel1　　查询 SPC 协议解码数据信源为通道 1。

:DECODe<x>:SPC:DATAfieldnumber

命令格式

:DECODe<x>:SPC:DATAfieldnumber <String>

:DECODe<x>:SPC:DATAfieldnumber?

功能描述

设置 SPC 协议解码的数据字段个数。

查询 SPC 协议解码的数据字段个数。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 3 | 4 | 5 | 6 }。

返回格式

查询命令返回 3、4、5 或 6。

实例说明

:DECODe1:SPC:DATAfieldnumber 4　　设置 SPC 协议解码的数据字段个数为 4。

:DECODe1:SPC:DATAfieldnumber? -> 4　　查询 SPC 协议解码的数据字段个数为 4。

:DECODe<x>:SPC:UNITtime

命令格式

:DECODe<x>:SPC:UNITtime <String>

:DECODe<x>:SPC:UNITtime?

功能描述

设置 SPC 协议解码的时间片宽度。

查询 SPC 协议解码的时间片宽度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 500.0ns~3.880us，可带单位，如“600ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SPC 协议解码的时间片宽度。

实例说明

:DECODe1:SPC:UNITtime 2e-6	设置 SPC 协议解码的时间片宽度为 2us。
:DECODe1:SPC:UNITtime? -> 2.000us	查询 SPC 协议解码的时间片宽度为 2us。

:DECODe<x>:SPI:SCK

命令格式

:DECODe<x>:SPI:SCK <String>

:DECODe<x>:SPI:SCK?

功能描述

设置 SPI 协议解码的时钟信源。

查询 SPI 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SPI:SCK CHANnel1	设置 SPI 协议解码的时钟信源为通道 1。
:DECODe1:SPI:SCK? -> CHANnel1	查询 SPI 协议解码的时钟信源为通道 1。

:DECODe<x>:SPI:SDA

命令格式

:DECODe<x>:SPI:SDA <String>

:DECODe<x>:SPI:SDA?

功能描述

设置 SPI 协议解码的数据信源。

查询 SPI 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:SPI:SDA CHANnel1	设置 SPI 协议解码的数据信源为通道 1。
:DECODe1:SPI:SDA? -> CHANnel1	查询 SPI 协议解码的数据信源为通道 1。

:DECODe<x>:SPI:CS

命令格式

:DECODe<x>:SPI:CS <String>

:DECODe<x>:SPI:CS?

功能描述

设置 SPI 协议解码的片选信源。

查询 SPI 协议解码的片选信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | NONE }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4 或 NONE。

实例说明

:DECODe1:SPI:CS CHANnel1	设置 SPI 协议解码的片选信源为通道 1。
:DECODe1:SPI:CS? -> CHANnel1	查询 SPI 协议解码的片选信源为通道 1。

:DECODe<x>:SPI:SAMPmode

命令格式

:DECODe<x>:SPI:SAMPmode <String>

:DECODe<x>:SPI:SAMPmode?

功能描述

设置 SPI 协议解码的工作方式。

查询 SPI 协议解码的工作方式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { POLPha00 | POLPha01 | POLPha10 | POLPha11 }，分别表示 POL:PHA 0:0、POL:PHA 0:1、POL:PHA 1:0、POL:PHA 1:1。

返回格式

查询命令返回 POLPha00、POLPha01、POLPha10 或 POLPha11。

实例说明

:DECODe1:SPI:SAMPmode POLPha00 设置 SPI 协议解码的工作方式为 POL:PHA 0:0。

:DECODe1:SPI:SAMPmode? -> POLPha00 查询 SPI 协议解码的工作方式为 POL:PHA 0:0。

:DECODe<x>:SPI:TRANsmode

命令格式

:DECODe<x>:SPI:TRANsmode <String>

:DECODe<x>:SPI:TRANsmode?

功能描述

设置 SPI 协议解码的传输模式。

查询 SPI 协议解码的传输模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { LSB | MSB}。

返回格式

查询命令返回 LSB 或 MSB。

实例说明

:DECODe1:SPI:TRANsmode LSB	设置 SPI 协议解码的传输模式为 LSB。
:DECODe1:SPI:TRANsmode? -> LSB	查询 SPI 协议解码的传输模式为 LSB。

:DECODe<x>:SPI:DATAlen

命令格式

:DECODe<x>:SPI:DATAlen <Int>

:DECODe<x>:SPI:DATAlen?

功能描述

设置 SPI 协议解码的数据宽度。

查询 SPI 协议解码的数据宽度。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 数据宽度，范围 4~32。

返回格式

查询命令以整型形式返回 SPI 协议解码的数据宽度大小。

实例说明

:DECODe1:SPI:DATAlen 8	设置 SPI 协议解码的数据宽度为 8。
:DECODe1:SPI:DATAlen? ->8	查询 SPI 协议解码的数据宽度为 8。

:DECODe<x>:SPI:OVERtime

命令格式

:DECODe<x>:SPI:OVERtime <String>

:DECODe<x>:SPI:OVERtime?

功能描述

设置 SPI 协议解码的超时值。

查询 SPI 协议解码的超时值。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 40.00ns~1.000s，可带单位，如“600ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SPI 协议解码的超时值。

实例说明

:DECODe1:SPI:OVERtime 2e-6	设置 SPI 协议解码的超时值为 2us。
:DECODe1:SPI:OVERtime? -> 2.000us	查询 SPI 协议解码的超时值为 2us。

:DECODe<x>:TDM:CLK

命令格式

:DECODe<x>:TDM:CLK <String>

:DECODe<x>:TDM:CLK?

功能描述

设置 TDM 协议解码的时钟信源。

查询 TDM 协议解码的时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:TDM:CLK CHANnel1　　设置 TDM 协议解码的时钟信源为通道 1。

:DECODe1:TDM:CLK? -> CHANnel1　　查询 TDM 协议解码的时钟信源为通道 1。

:DECODe<x>:TDM:DAT

命令格式

:DECODe<x>:TDM:DAT <String>

:DECODe<x>:TDM:DAT?

功能描述

设置 TDM 协议解码的数据信源。

查询 TDM 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:TDM:DAT CHANnel1　　设置 TDM 协议解码的数据信源为通道 1。

:DECODe1:TDM:DAT? -> CHANnel1　　查询 TDM 协议解码的数据信源为通道 1。

:DECODe<x>:TDM:FS

命令格式

:DECODe<x>:TDM:FS <String>

:DECODe<x>:TDM:FS?

功能描述

设置 TDM 协议解码的帧同步时钟信源。

查询 TDM 协议解码的帧同步时钟信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:TDM:FS CHANnel1　　设置 TDM 协议解码的帧同步时钟信源为通道 1。

:DECODe1:TDM:FS? -> CHANnel1　　查询 TDM 协议解码的帧同步时钟信源为通道 1。

:DECODe<x>:TDM:CHNCnt

命令格式

:DECODe<x>:TDM:CHNCnt <String>

:DECODe<x>:TDM:CHNCnt?

功能描述

设置 TDM 协议解码的通道个数。

查询 TDM 协议解码的通道个数。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 }。

返回格式

查询命令返回 1、2、3、4、5、6、7 或 8。

实例说明

:DECODe1:TDM:CHNCnt 8	设置 TDM 协议解码的通道个数为 8。
:DECODe1:TDM:CHNCnt? -> 8	查询 TDM 协议解码的通道个数为 8。

:DECODe<x>:TDM:BITCnt

命令格式

:DECODe<x>:TDM:BITCnt <Int>

:DECODe<x>:TDM:BITCnt?

功能描述

设置 TDM 协议解码的数据位宽。

查询 TDM 协议解码的数据位宽。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 1~32。

返回格式

查询命令以整数形式返回 TDM 协议解码的数据位宽。

实例说明

:DECODe1:TDM:BITCnt 8	设置 TDM 协议解码的数据位宽为 8。
:DECODe1:TDM:BITCnt? -> 8	查询 TDM 协议解码的数据位宽为 8

:DECODe<x>:UART:SOURce

命令格式

:DECODe<x>:UART:SOURce <String>

:DECODe<x>:UART:SOURce?

功能描述

设置 UART 协议解码的信源。

查询 UART 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:UART:SOURce CHANnel1　　设置 UART 协议解码信源为通道 1。

:DECODe1:UART:SOURce? -> CHANnel1　　查询 UART 协议解码信源为通道 1。

:DECODe<x>:UART:BAUDrate

命令格式

:DECODe<x>:UART:BAUDrate <Int>

:DECODe<x>:UART:BAUDrate?

功能描述

设置 UART 协议解码的波特率。

查询 UART 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 波特率，范围 100~20000000。

返回格式

查询命令以整型形式返回 UART 协议解码的波特率大小。

实例说明

:DECODe1:UART:BAUDrate 9600	设置 UART 协议解码的波特率为 9600。
:DECODe1:UART:BAUDrate? -> 9600	查询 UART 协议解码的波特率为 9600。

:DECODe<x>:UART:STOPbit

命令格式

:DECODe<x>:UART:STOPbit <String>

:DECODe<x>:UART:STOPbit?

功能描述

设置 UART 协议解码的结束位宽。

查询 UART 协议解码的结束位宽。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 1 | 1.5 | 2 }。

返回格式

查询命令返回 1、1.5 或 2。

实例说明

:DECODe1:UART:STOPbit 1	设置 UART 协议解码的结束位宽为 1。
:DECODe1:UART:STOPbit? -> 1	查询 UART 协议解码的结束位宽为 1。

:DECODe<x>:UART:PARItymode

命令格式

:DECODe<x>:UART:PARItymode <String>

:DECODe<x>:UART:PARItymode?

功能描述

设置 UART 协议解码的校验位。

查询 UART 协议解码的校验位。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { EVEN | ODD | MARK | SPACe | NONE }。

返回格式

查询命令返回 EVEN、ODD、MARK、SPACe 或 NONE。

实例说明

:DECODe1:UART:PARItymode NONE　　设置 UART 协议解码的校验位为 NONE。

:DECODe1:UART:PARItymode? -> NONE　　查询 UART 协议解码的校验位为 NONE。

:DECODe<x>:UART:DATAMode

命令格式

:DECODe<x>:UART:DATAMode <String>

:DECODe<x>:UART:DATAMode?

功能描述

设置 UART 协议解码的数据模式。

查询 UART 协议解码的数据模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { LSB | MSB}。

返回格式

查询命令返回 LSB 或 MSB。

实例说明

:DECODe1:UART:DATAMode LSB	设置 UART 协议解码的数据模式为 LSB。
:DECODe1:UART:DATAMode? -> LSB	查询 UART 协议解码的数据模式为 LSB。

:DECODe<x>:UART:REVErse

命令格式

:DECODe<x>:UART:REVErse <String>

:DECODe<x>:UART:REVErse?

功能描述

设置 UART 协议解码的电平反相。

查询 UART 协议解码的电平反相。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:DECODe1:UART:REVErse FALSe　　设置 UART 协议解码的电平反相为 FALSe。

:DECODe1:UART:REVErse? -> FALSe　　查询 UART 协议解码的电平反相为 FALSe。

:DECODe<x>:UART:DATALength

命令格式

:DECODe<x>:UART:DATALength <String>

:DECODe<x>:UART:DATALength?

功能描述

设置 UART 协议解码的数据位宽。

查询 UART 协议解码的数据位宽。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 4 | 5 | 6 | 7 | 8 | 9 | 12 | 16 }。

返回格式

查询命令返回 4、5、6、7、8、9、12 或 16。

实例说明

:DECODe1:UART:DATALength 8	设置 UART 协议解码的数据位宽为 8。
:DECODe1:UART:DATALength? -> 8	查询 UART 协议解码的数据位宽为 8。

:DECODe<x>:USB:D+

命令格式

:DECODe<x>:USB:D+ <String>

:DECODe<x>:USB:D+?

功能描述

设置 USB 协议解码的 D+的信源。

查询 USB 协议解码的 D+的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:USB:D+ CHANnel1	设置 USB 协议解码的 D+信源为通道 1。
:DECODe1:USB:D+? -> CHANnel1	查询 USB 协议解码的 D+信源为通道 1。

:DECODe<x>:USB:D-

命令格式

:DECODe<x>:USB:D- <String>

:DECODe<x>:USB:D-?

功能描述

设置 USB 协议解码的 D-的信源。

查询 USB 协议解码的 D-的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:USB:D- CHANnel1	设置 USB 协议解码的 D-信源为通道 1。
:DECODe1:USB:D-? -> CHANnel1	查询 USB 协议解码的 D-信源为通道 1。

:DECODe<x>:USB:USBMode

命令格式

:DECODe<x>:USB:USBMode <String>

:DECODe<x>:USB:USBMode?

功能描述

设置 USB 协议解码的 USB 模式。

查询 USB 协议解码的 USB 模式。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { LOWSpeed | FULLspeed }，分别表示低速、全速。

返回格式

查询命令返回 LOWSpeed 或 FULLspeed。

实例说明

:DECODe1:USB:USBMode LOWSpeed	设置 USB 协议解码的 USB 模式为低速。
:DECODe1:USB:USBMode? -> LOWSpeed	查询 USB 协议解码的 USB 模式为低速。

:DECODe<x>:USBPd:SOURce

命令格式

:DECODe<x>:USBPd:SOURce <String>

:DECODe<x>:USBPd:SOURce?

功能描述

设置 USB-PD 协议解码的信源。

查询 USB-PD 协议解码的信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:USBPd:SOURce CHANnel1　　设置 USB-PD 协议解码信源为通道 1。

:DECODe1:USBPd:SOURce? -> CHANnel1　　查询 USB-PD 协议解码信源为通道 1。

:DECODe<x>:USBPd:BITRate

命令格式

:DECODe<x>:USBPd:BITRate <Double>

:DECODe<x>:USBPd:BITRate?

功能描述

设置 USB-PD 协议解码的比特率。

查询 USB-PD 协议解码的比特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 比特率，范围 1.00k~10000.00k。

返回格式

查询命令以实型形式返回 USB-PD 协议解码的比特率大小。

实例说明

:DECODe1:USBPd:BITRate 15	设置 USB-PD 协议解码的比特率为 15k。
:DECODe1:USBPd:BITRate? -> 15.00	查询 USB-PD 协议解码的比特率为 15k。

:DECODe<x>:USBPd:ERROrrange

命令格式

:DECODe<x>:USBPd:ERROrrange <Double>

:DECODe<x>:USBPd:ERROrrange?

功能描述

设置 USB-PD 协议解码的误差范围。

查询 USB-PD 协议解码的误差范围。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 误差范围，范围 0.10~0.50。

返回格式

查询命令以实型形式返回 USB-PD 协议解码的误差范围大小。

实例说明

:DECODe1:USBPd:ERROrrange 0.3	设置 USB-PD 协议解码的误差范围为 0.3。
:DECODe1:USBPd:ERROrrange? -> 0.30	查询 USB-PD 协议解码的误差范围为 0.3。

:DECODe<x1>:WIEGand:DAT<x2>

命令格式

:DECODe<x1>:WIEGand:DAT<x2> <String>

:DECODe<x1>:WIEGand:DAT<x2>?

功能描述

设置 Wiegand 协议解码的 DAT0 和 DAT1 的信源。

查询 Wiegand 协议解码的 DAT0 和 DAT1 的信源。

参数说明

<x1> 需要查询或改变的解码通道编号，1 或 2。

<x2> DAT 的编号，范围{ 0 | 1 }。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:WIEGand:DAT0 CHANnel1　　设置 Wiegand 协议解码的 DAT0 信源为通道 1。

:DECODe1:WIEGand:DAT0? -> CHANnel1　　查询 Wiegand 协议解码的 DAT0 信源为通道 1。

:DECODe<x>:WIEGand:TYPE

命令格式

:DECODe<x>:WIEGand:TYPE <String>

:DECODe<x>:WIEGand:TYPE?

功能描述

设置 Wiegand 协议解码的协议类型。

查询 Wiegand 协议解码的协议类型。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { 26bit | 39bit | 44bit | CUSTom }。

返回格式

查询命令返回 26bit、39bit、44bit 或 CUSTom。

实例说明

:DECODe1:WIEGand:TYPE CUSTom 设置 Wiegand 协议解码的协议类型为自定义。

:DECODe1:WIEGand:TYPE? -> CUSTom 查询 Wiegand 协议解码的协议类型为自定义。

:DECODe<x>:WIEGand:SPACe

命令格式

:DECODe<x>:WIEGand:SPACe <String>

:DECODe<x>:WIEGand:SPACe?

功能描述

设置 Wiegand 协议解码的帧间隔。

查询 Wiegand 协议解码的帧间隔。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 10.00ms~1.000s，可带单位，如“600ms”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 Wiegand 协议解码的帧间隔。

实例说明

:DECODe1:WIEGand:SPACe 2e-2　　设置 Wiegand 协议解码的帧间隔为 20ms。

:DECODe1:WIEGand:SPACe? -> 20.00ms　　查询 Wiegand 协议解码的帧间隔为 20ms。

:DECODe<x>:WIEGand:OEMLength

命令格式

:DECODe<x>:WIEGand:OEMLength <Int>

:DECODe<x>:WIEGand:OEMLength?

功能描述

设置 Wiegand 协议解码的 OEM 位宽。

查询 Wiegand 协议解码的 OEM 位宽。

注：协议类型设置为**自定义**时有效，相关指令:DECODe<x>:WIEGand:TYPE。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Wiegand 协议解码的 OEM 位宽。

实例说明

:DECODe1:WIEGand:OEMLength 8　　设置 Wiegand 协议解码的 OEM 位宽为 8。

:DECODe1:WIEGand:OEMLength? -> 8　　查询 Wiegand 协议解码的 OEM 位宽为 8

:DECODe<x>:WIEGand:FCLEngth

命令格式

:DECODe<x>:WIEGand:FCLEngth <Int>

:DECODe<x>:WIEGand:FCLEngth?

功能描述

设置 Wiegand 协议解码的 FC 位宽。

查询 Wiegand 协议解码的 FC 位宽。

注：协议类型设置为**自定义**时有效，相关指令:DECODe<x>:WIEGand:TYPE。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Wiegand 协议解码的 FC 位宽。

实例说明

:DECODe1:WIEGand:FCLEngth 8	设置 Wiegand 协议解码的 FC 位宽为 8。
:DECODe1:WIEGand:FCLEngth? -> 8	查询 Wiegand 协议解码的 FC 位宽为 8

:DECODe<x>:WIEGand:CCLEngth

命令格式

:DECODe<x>:WIEGand:CCLEngth <Int>

:DECODe<x>:WIEGand:CCLEngth?

功能描述

设置 Wiegand 协议解码的 CC 位宽。

查询 Wiegand 协议解码的 CC 位宽。

注：协议类型设置为**自定义**时有效，相关指令:DECODe<x>:WIEGand:TYPE。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Wiegand 协议解码的 CC 位宽。

实例说明

:DECODe1:WIEGand:CCLEngth 8	设置 Wiegand 协议解码的 CC 位宽为 8。
:DECODe1:WIEGand:CCLEngth? -> 8	查询 Wiegand 协议解码的 CC 位宽为 8

:DECODe<x>:WTB:SOURce

命令格式

:DECODe<x>:WTB:SOURce <String>

:DECODe<x>:WTB:SOURce?

功能描述

设置 WTB 协议解码的数据信源。

查询 WTB 协议解码的数据信源。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:DECODe1:WTB:SOURce CHANnel1	设置 WTB 协议解码数据信源为通道 1。
:DECODe1:WTB:SOURce? -> CHANnel1	查询 WTB 协议解码数据信源为通道 1。

:DECODe<x>:WTB:BAUDrate

命令格式

:DECODe<x>:WTB:BAUDrate <Double>

:DECODe<x>:WTB:BAUDrate?

功能描述

设置 WTB 协议解码的波特率。

查询 WTB 协议解码的波特率。

参数说明

<x> 需要查询或改变的解码通道编号，1 或 2。

<Double> 波特率，范围 0.00M~8.00M。

返回格式

查询命令以实型形式返回 WTB 协议解码的波特率大小。

实例说明

:DECODe1:WTB:BAUDrate 2　　设置 WTB 协议解码的波特率为 2M。

:DECODe1:WTB:BAUDrate? -> 2.00　　查询 WTB 协议解码的波特率为 2M。

9. 显示相关命令

:DISPlay 命令用于查询和设置显示配置。

- :DISPlay:VECTors 设置和查询显示类型。
- :DISPlay:GBRightness 设置和查询网格亮度。
- :DISPlay:WBRightness 设置和查询波形亮度。
- :DISPlay:BACKBRightness 设置和查询背光亮度。
- :DISPlay:PERSistence 设置和查询余晖时间。
- :DISPlay:COLOrgraded 设置和查询色温开关状态。
- :DISPlay:FREEze 设置和查询冻结开关状态。
- :DISPlay:DATA? 读取示波器当前的位图数据流。
- :DISPlay:DATA:PNG? 读取示波器当前的位图数据流。

:DISPlay:VECTors

命令格式

:DISPlay:VECTors <Bool>

:DISPlay:VECTors?

功能描述

设置显示类型。

查询当前显示类型。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，将显示类型设置成点显示模式；

{OFF | FALSE | 0}，将显示类型设置成线显示模式。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:DISPlay:VECTors TRUE	将显示类型设置成点显示模式。
:DISPlay:VECTors? -> 0	查询当前显示类型为线显示。

:DISPlay:GBRightness

命令格式

:DISPlay:GBRightness <UInt>

:DISPlay:GBRightness?

功能描述

设置网格亮度。

查询网格亮度。

参数说明

<UInt> 范围 0~100，单位百分比。

返回格式

查询命令以整数形式返回网格亮度。

实例说明

:DISPlay:GBRightness 50	设置网格亮度为 50%。
:DISPlay:GBRightness? -> 50	查询当前网格亮度为 50%。

:DISPlay:WBRightness

命令格式

:DISPlay:WBRightness <String>, <UInt>

:DISPlay:WBRightness?

功能描述

设置波形亮度。

查询波形亮度。

参数说明

<String> 范围 { ALL | CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | M1 | M2 | M3 | M4 | R1 | R2 | R3 | R4 }。

<UInt> 范围 0~100，单位百分比。

返回格式

查询命令以整数形式返回波形亮度。

注：查询时<String>不可为"ALL"。

实例说明

:DISPlay:WBRightness ALL,60	设置所有通道波形亮度为 60%。
:DISPlay:WBRightness? CHANnel1 -> 60	查询当前通道 1 波形亮度为 60%。

:DISPlay:BACKBRightness

命令格式

:DISPlay:BACKBRightness <UInt>

:DISPlay:BACKBRightness?

功能描述

设置背光亮度。

查询背光亮度。

参数说明

<UInt> 范围 5~100，单位百分比。

返回格式

查询命令以整数形式返回背光亮度。

实例说明

:DISPlay:BACKBRightness 99	设置背光亮度为 99%。
:DISPlay:BACKBRightness? -> 60	查询当前背光亮度为 60%。

:DISPlay:PERSistence

命令格式

:DISPlay:PERSistence <String>

:DISPlay:PERSistence?

功能描述

设置余晖时间。

查询余晖时间。

参数说明

<String> 范围{ OFF | 0.1 | 0.2 | 0.5 | 1 | 2 | 5 | 10 | 20 | 50 | INFinite }，单位为 s。

分别表示关闭、100ms、200ms、500ms、1s、5s、10s、20s、无限。

返回格式

查询命令返回 OFF、0.1、0.2、0.5、1、2、5、10、20、50 或 INFinite。

实例说明

:DISPlay:PERSistence OFF	关闭余晖。
:DISPlay:PERSistence? -> 1	代表 1s 的余晖。

:DISPlay:COLOrgraded

命令格式

:DISPlay:COLOrgraded <Bool>

:DISPlay:COLOrgraded?

功能描述

设置色温开关状态。

查询色温开关状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开色温；

{OFF | FALSE | 0}，关闭色温。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:DISPlay:COLOrgraded ON	打开色温。
:DISPlay:COLOrgraded? -> 0	查询当前色温为关闭状态。

:DISPlay:FREEze

命令格式

:DISPlay:FREEze <Bool>

:DISPlay:FREEze?

功能描述

设置冻结开关状态。

查询冻结开关状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开冻结；

{OFF | FALSE | 0}，关闭冻结。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:DISPlay:FREEze ON	打开冻结。
:DISPlay:FREEze? -> 0	查询当前冻结为关闭状态。

:DISPlay:DATA

命令格式

:DISPlay:DATA?

功能描述

读取示波器当前的位图数据流。

参数说明

无。

返回格式

查询命令返回 #<dig><len><data><end>。

数据流格式如下所示：

数据流长度格式	
数据	说明
#<dig><len>	文件头以#起始，<dig>表示后面用<dig>个 10 进制的字符来表示数据流的长度，也即<len>字节
数据流	
数据	说明
<data>	具体的位图数据
文件尾	
数据	说明
<end>	\n

注 1：<dig>:范围 0~9，指定随后<len>长度；

注 2：<len>:表示二进制文件的大小;

注 3：<data>:表示具体的位图数据，将<data><end>另存为“xxx.BMP”文件即可得到示波器当前界面截图。

实例说明

:DISPlay:DATA? -> #76220854BM…

请参考 Demo 编程例程说明中的“例程 2:截图”。

:DISPlay:DATA:PNG

命令格式

:DISPlay:DATA:PNG?

功能描述

读取示波器当前的位图数据流。

参数说明

无。

返回格式

查询命令返回 #<dig><len><data><end>。

数据流格式如下所示：

数据流长度格式	
数据	说明
#<dig><len>	文件头以#起始，<dig>表示后面用<dig>个 10 进制的字符来表示数据流的长度，也即<len>字节
数据流	
数据	说明
<data>	具体的位图数据
文件尾	
数据	说明
<end>	\n

注 1：<dig>:范围 0~9，指定随后<len>长度；

注 2：<len>:表示二进制文件的大小;

注 3：<data>:表示具体的位图数据，将<data><end>另存为“xxx.png”文件即可得到示波器当前界面截图。

实例说明

:DISPlay:DATA:PNG? -> #6103817 塒 NG…

请参考 Demo 编程例程说明中的“例程 3:截图 png”。

10. FFT 相关命令

:FFT 命令用于查询和设置 FFT 的配置。

- :FFT:STATe 设置和查询 FFT 的开关状态。
- :FFT:SOURce 设置和查询 FFT 的信源。
- :FFT:WINDow 设置和查询 FFT 的窗函数。
- :FFT:VSMOde 设置和查询 FFT 的显示模式。
- :FFT:HOR:AUTo 设置和查询 FFT 水平轴自动跨度状态。
- :FFT:HOR:SCOpe 设置和查询 FFT 水平跨度。
- :FFT:HOR:CENTer 设置和查询 FFT 水平轴中心。
- :FFT:VERT:AUTo 设置和查询 FFT 垂直轴自动跨度状态。
- :FFT:VERT:SCOpe 设置和查询 FFT 垂直跨度。
- :FFT:VERT:CENTer 设置和查询 FFT 垂直轴中心。
- :FFT:TABLe? 查询 FFT 所有报表结果。
- :FFT:ITEM? 查询 FFT 某行报表结果。

:FFT:STATe

命令格式

:FFT:STATe <Bool>

:FFT:STATe?

功能描述

设置 FFT 的开关状态。

查询 FFT 的开关状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开 FFT 功能；

{OFF | FALSE | 0}，关闭 FFT 功能。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:FFT:STATe ON	打开 FFT 功能。
:FFT:STATe? -> 0	查询当前 FFT 功能为关闭状态。

:FFT:SOURce

命令格式

:FFT:SOURce <String>

:FFT:SOURce?

功能描述

设置 FFT 的信源。

查询 FFT 的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | Math1 | Math2 | Math3 | Math4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、Math1、Math2、Math3 或 Math4。

实例说明

:FFT:SOURce CHANnel1	设置 FFT 信源为通道 1。
:FFT:SOURce? -> CHANnel1	查询 FFT 信源为通道 1。

:FFT:WINDow

命令格式

:FFT:WINDow <String>

:FFT:WINDow?

功能描述

设置 FFT 的窗函数。

查询 FFT 的窗函数。

参数说明

<String> 范围 {Rectangle | Hanning | Hamming | Blackman | Flattop}，分别表示矩形窗、汉宁窗、海明窗、布莱克曼窗、平顶窗。

返回格式

查询命令返回 Rectangle、Hanning、Hamming、Blackman 或 Flattop。

实例说明

:FFT:WINDow Rectangle	设置 FFT 窗函数为矩形窗。
:FFT:WINDow? -> Rectangle	查询当前 FFT 窗函数为矩形窗。

:FFT:VSMOde

命令格式

:FFT:VSMOde <String>

:FFT:VSMOde?

功能描述

设置 FFT 的显示模式。

查询 FFT 的显示模式。

参数说明

<String> 范围：{Ampl | dBm | Vrms | PSD}。

返回格式

查询命令返回 Ampl、dBm、Vrms 或 PSD。

实例说明

:FFT:VSMOde Ampl	设置 FFT 的显示模式为 Ampl。
:FFT:VSMOde? -> dBm	查询当前 FFT 的显示模式为 dBm。

:FFT:HOR:AUTo

命令格式

:FFT:HOR:AUTo <Bool>

:FFT:HOR:AUTo?

功能描述

设置 FFT 水平轴自动跨度状态。

查询 FFT 水平轴自动跨度状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开 FFT 水平轴自动跨度；

{OFF | FALSE | 0}，关闭 FFT 水平轴自动跨度。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:FFT:HOR:AUTo ON　　打开 FFT 水平轴自动跨度。

:FFT:HOR:AUTo? -> 0　　查询当前 FFT 水平轴自动跨度为关闭状态。

:FFT:HOR:SCOpe

命令格式

:FFT:HOR:SCOpe <Double>

:FFT:HOR:SCOpe?

功能描述

设置 FFT 水平跨度。

查询 FFT 水平跨度。

注：FFT 水平轴自动跨度设置为**关闭**时有效，相关指令:FFT:HOR:AUTo。

参数说明

<Double> 频率值，范围 1nHz~100GHz，单位 Hz。

返回格式

查询命令以实型形式返回 FFT 水平跨度，单位 Hz。

实例说明

:FFT:HOR:SCOpe 1000	设置 FFT 水平跨度为 1kHz。
:FFT:HOR:SCOpe? ->1000	查询 FFT 水平跨度为 1kHz。

:FFT:HOR:CENTer

命令格式

:FFT:HOR:CENTer <Double>

:FFT:HOR:CENTer?

功能描述

设置 FFT 水平轴中心。

查询 FFT 水平轴中心。

注：FFT 水平轴自动跨度设置为**关闭**时有效，相关指令:FFT:HOR:AUTo。

参数说明

<Double> 频率值，范围-100GHz~100GHz，单位 Hz。

返回格式

查询命令以实型形式返回 FFT 水平轴中心，单位 Hz。

实例说明

:FFT:HOR:CENTer 1000	设置 FFT 水平轴中心为 1kHz。
:FFT:HOR:CENTer? ->1000	查询 FFT 水平轴中心为 1kHz。

:FFT:VERT:AUTo

命令格式

:FFT:VERT:AUTo <Bool>

:FFT:VERT:AUTo?

功能描述

设置 FFT 垂直轴自动跨度状态。

查询 FFT 垂直轴自动跨度状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开 FFT 垂直轴自动跨度；

{OFF | FALSE | 0}，关闭 FFT 垂直轴自动跨度。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:FFT:VERT:AUTo ON　　打开 FFT 垂直轴自动跨度。

:FFT:VERT:AUTo? -> 0　　查询当前 FFT 垂直轴自动跨度为关闭状态。

:FFT:VERT:SCOpe

命令格式

:FFT:VERT:SCOpe <Double>

:FFT:VERT:SCOpe?

功能描述

设置 FFT 垂直跨度。

查询 FFT 垂直跨度。

注：FFT 垂直轴自动跨度设置为**关闭**时有效，相关指令:FFT:VERT:AUTo。

参数说明

<Double> 电压/功率值，范围 1ndBm~100GdBm，单位 dBm 或 V。

返回格式

查询命令以实型形式返回 FFT 垂直跨度，单位 dBm 或 V。

实例说明

:FFT:VERT:SCOpe 1000	设置 FFT 垂直跨度为 1kdBm。
:FFT:VERT:SCOpe? ->1000	查询 FFT 垂直跨度为 1kdBm。

:FFT:VERT:CENTer

命令格式

:FFT:VERT:CENTer <Double>

:FFT:VERT:CENTer?

功能描述

设置 FFT 垂直轴中心。

查询 FFT 垂直轴中心。

注：FFT 垂直轴自动跨度设置为**关闭**时有效，相关指令:FFT:VERT:AUTo。

参数说明

<Double> 电压/功率值，范围-100GdBm~100GdBm，单位 dBm 或 V。

返回格式

查询命令以实型形式返回 FFT 垂直轴中心，单位 dBm 或 V。

实例说明

:FFT:VERT:CENTer 1000	设置 FFT 垂直轴中心为 1kdBm。
:FFT:VERT:CENTer? ->1000	查询 FFT 垂直轴中心为 1kdBm。

:FFT:TABLe

命令格式

:FFT:TABLe?

功能描述

查询 FFT 所有报表结果。

参数说明

无。

返回格式

查询命令以字符串形式返回 FFT 所有报表结果。

实例说明

无。

:FFT:ITEM

命令格式

:FFT:ITEM? <UInt>

功能描述

查询 FFT 某行报表结果。

参数说明

<UInt> 范围{0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20}。

返回格式

查询命令以字符串形式返回 FFT 某行报表结果。

实例说明

:FFT:ITEM? 1 -> 1,1250000000.000000,-44.466801,0.000000;

查询 FFT 序列为 1 的报表结果。

11. 按键相关命令

:KEY 命令用于发送特定按键消息

- :KEY　　发送按键事件，模拟按键按下的效果。

:KEY

命令格式

:KEY <UInt>

功能描述

发送按键事件，模拟按键按下的效果。

参数说明

<UInt> 按键键值。

键值查找表

按键键值对应的查找表如下所示：

表 11.1　ZUS6000 键值查找表

功能	键值	功能	键值	功能	键值	功能	键值
X-Key	**169**	Cursor	130	Zoom	145	Ref	**167**
		Utility	166	Acquire	146	Math	159
Single	132	Search	168				
Run/Stop	131	Measure	137	Vert Division +	160	Trigger	151
AutoSetup	133	Clear	135	Vert Division -	161	Trig Offset -	148
Default	134	Printscreen	136	Vert Division Push	162	Trig Offset +	149
Touch	138			Vert Offset -	163	Trig Offset Push	150
				Vert Offset +	164	Auto/Normal	152
A -	40	Hrzt Division +	139	Vert Offset Push	165		
A +	38	Hrzt Division -	140				
A Push	128	Hrzt Division Push	141	Channel1	153		
B -	37	Hrzt Offset -	142	Channel2	154		
B +	39	Hrzt Offset +	143	Channel3	155		
B Push	13	Hrzt Offset Push	144	Channel4	156		

表 11.2　ZUS5000 键值查找表

功能	键值	功能	键值	功能	键值	功能	键值
Single	133	Persist	**129**	Zoom	146	Math	160
Run/Stop	132	T	130	Acquire	147		
AutoSetup	134	Cursor	131			Trigger	152
Default	135	Clear	136	Vert Division +	161	Trig Offset -	149
		Printscreen	137	Vert Division -	162	Trig Offset +	150
A -	40	Measure	138	Vert Division Push	163	Trig Offset Push	151
A +	38	Touch	139	Vert Offset -	164	Auto/Normal	153
A Push	128			Vert Offset +	165		
B -	37	Hrzt Division +	140	Vert Offset Push	166		
B +	39	Hrzt Division -	141				
B Push	13	Hrzt Division Push	142	Channel1	154		
		Hrzt Offset -	143	Channel2	155		
		Hrzt Offset +	144	Channel3	156		
		Hrzt Offset Push	145	Channel4	157		

实例说明

:KEY 132　　点击“single”按键

12. 数学运算相关命令

:MATH 命令用于查询和设置数学通道的配置。

- :MATH<x>:STAte 设置和查询数学通道状态。
- :MATH<x>:DISPlay 设置和查询数学通道显示状态。
- :MATH<x>:SCALe 设置和查询数学通道的档位。
- :MATH<x>:OFFSet 设置和查询数学通道的偏移。
- :MATH<x>:CALAREa 设置和查询数学通道的计算区域。
- :MATH<x>:FORMula 设置和查询数学通道的公式。

:MATH<x>:STAte

命令格式

:MATH<x>:STAte <Bool>

:MATH<x>:STAte?

功能描述

设置数学通道状态。

查询数学通道状态。

参数说明

<x> 需要查询或改变的数学通道编号，1、2、3 或 4。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开数学通道；

{OFF | FALSE | 0} 关闭数学通道。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:MATH1:STAte ON	打开数学通道 1。
:MATH1:STAte? -> 1	查询当前数学通道 1 为打开状态。

:MATH<x>:DISPlay

命令格式

:MATH<x>:DISPlay <Bool>

:MATH<x>:DISPlay?

功能描述

设置数学通道显示状态。

查询数学通道显示状态。

参数说明

<x> 需要查询或改变的数学通道编号，1、2、3 或 4。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 显示数学通道；

{OFF | FALSE | 0} 隐藏数学通道。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：显示，隐藏。

实例说明

:MATH1:DISPlay ON	显示数学通道 1。
:MATH1:DISPlay? -> 1	查询当前数学通道 1 为显示状态。

:MATH<x>:SCALe

命令格式

:MATH<x>:SCALe <Double>

:MATH<x>:SCALe?

功能描述

设置数学通道的档位。

查询数学通道的档位。

参数说明

<x> 需要查询或改变的数学通道编号，1、2、3 或 4。

<Double> 档位，可带单位，如“2mV”，也可用科学计数法表示。

返回格式

查询命令以实型形式返回数学通道的档位，单位 V。

实例说明

:MATH1:SCALe 2mV	设置数学通道 1 垂直档位为 2mV。
:MATH1:SCALe? -> 0.002	查询当前数学通道 1 垂直档位为 2mV。

:MATH<x>:OFFSet

命令格式

:MATH<x>:OFFSet <Double>

:MATH<x>:OFFSet?

功能描述

设置数学通道的偏移。

查询数学通道的偏移。

参数说明

<x> 需要查询或改变的数学通道编号，1、2、3 或 4。

<Double> 偏移量，可带单位，如“2mV”，也可用科学计数法表示。

返回格式

查询命令以实型形式返回数学通道的偏移量，单位 V。

实例说明

:MATH1:OFFSet 2mV	设置数学通道 1 垂直偏移为 2mV。
:MATH1:OFFSet? -> 0.002	查询当前数学通道 1 垂直偏移为 2mV。

:MATH<x>:CALARea

命令格式

:MATH<x>:CALARea <String>

:MATH<x>:CALARea?

功能描述

设置数学通道的计算区域。

查询数学通道的计算区域。

参数说明

<x> 需要查询或改变的数学通道编号，1、2、3 或 4。

<String> 范围{RECORD | MAIN | Zoom1 | Zoom2 | CURSOR}，

分别表示全内存、主时基、Zoom1 区域、Zoom2 区域、光标区域。

返回格式

查询命令返回 RECORD、MAIN、Zoom1、Zoom2 或 CURSOR。

实例说明

:MATH1:CALARea RECORD　　设置数学通道 1 的计算区域为全内存。

:MATH1:CALARea? -> RECORD　　查询当前数学通道 1 的计算区域为全内存。

:MATH<x>:FOCUStab

命令格式

:MATH<x>:FOCUStab <String>

功能描述

设置数学通道的运算类型。

参数说明

<x> 需要查询或改变的数学通道编号，1、2、3 或 4。

<String> 范围{0 | 1 | 2}，

分别表示基本运算、滤波运算和高级运算。

返回格式

查询命令返回 0、1 或 2。

实例说明

:MATH1:FOCUStab 2	设置数学通道 1 的运算类型为高级运算。
:MATH1:FOCUStab? -> 2	查询当前数学通道 1 的运算类型为高级运算。

:MATH<x>:FORMula

命令格式

:MATH<x>:FORMula <String>

:MATH<x>:FORMula?

功能描述

设置数学通道的公式。

查询数学通道的公式。

注 1：运算类型设置为**高级运算**时有效，相关指令:MATH<x>:FOCUStab

注 2：输入滤波运算公式时，参数需用双引号分隔，例：

:MATH1:FORMula "FIRLowPass(C1,9)"

参数说明

<x> 需要查询或改变的数学通道编号，1、2、3 或 4。

<String> 数学表达式。

返回格式

查询命令以字符串形式返回数学表达式。

实例说明

:MATH1:FORMula C1+C1	设置数学通道 1 的数学表达式为 C1+C1。
:MATH1:FORMula? -> C1-C1	查询当前数学通道 1 的数学表达式为 C1-C1。

13. 测量相关命令

:MEASure 命令用于查询和设置测量的配置。

- :MEASure:ENABle 设置和查询测量使能。
- :MEASure:CLEAr 清除测量结果。
- :MEASure:THReshold:TYPE 设置和查询测量的阈值类型。
- :MEASure:THReshold:PERcent 设置和查询测量基顶基底值/最大最小值的阈值。
- :MEASure:THReshold:ABSolute 设置和查询测量绝对值的阈值。
- :MEASure:SCOPE:MODE 设置和查询测量区域。
- :MEASure<x>:ADD 增加测量项。
- :MEASure<x>:REMOve 删除测量项。
- :MEASure:REMOve:ALL 删除所有测量项。
- :MEASure<x>:TYPE 设置测量项的测量类型。
- :MEASure<x>:SOURce<x> 设置测量项的信源。
- :MEASure<x>:SELEction 设置平均、直流有效值、交流有效值、比率、面积、负面积和正面积等测量项的类型选择。
- :MEASure<x>:EDGE<x> 设置延迟和相位的测量项信源的边沿类型。
- :MEASure<x>:NUM<x> 设置延迟测量项信源的测量边沿。
- :MEASure<x>:SETTINGTYPE 设置和查询测量项使用的阈值和测量范围应用于全局或局部测量项。
- :MEASure<x>:LOCAL:SCOPE:MODE 设置和查询局部测量范围。
- :MEASure<x>:LOCAL:THReshold:TYPE 设置和查询测量的局部阈值类型。
- :MEASure<x>:LOCAL:THReshold:PERcent 设置和查询局部测量基顶基底值/最大最小值的阈值。
- :MEASure<x>:LOCAL:THReshold:ABSolute 设置和查询局部测量绝对值的阈值。
- :MEASure<x>:STAte? 查询某一测量项状态。
- :MEASure<x>:CURRent? 查询测量项的当前值。
- :MEASure<x>:MAXImum? 查询测量项的最大值。
- :MEASure<x>:MINImum? 查询测量项的最小值。
- :MEASure<x>:AVERage? 查询测量项的平均值。
- :MEASure<x>:DEViation? 查询测量项的标准方差值。
- :MEASure<x>:COUNt? 查询测量项的计数值。
- :MEASure<x>:AVAilable? 查询测量项的测量状态。
- :MEASure:OPEN 使能测量功能。
- :MEASure:CLOSe 关闭测量功能。

:MEASure:STAtics	设置和查询测量统计开关状态。
:MEASure:FREQuency:METER:TYPe	设置和查询硬件频率计的显示类型。
:MEASure:ITEM:STAtics?	查询是否存在特定测量项。

:MEASure:ENABle

命令格式

:MEASure:ENABle <Bool>

:MEASure:ENABle?

功能描述

设置测量使能。

查询测量使能。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开测量功能；

{OFF | FALSE | 0} 关闭测量功能。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:MEASure:ENABle ON	打开测量功能。
:MEASure:ENABle? -> 0	查询当前测量功能为关闭状态。

:MEASure:CLEAr

命令格式

:MEASure:CLEAr

功能描述

清除测量结果。

参数说明

无。

实例说明

无。

:MEASure:THReshold:TYPE

命令格式

:MEASure:THReshold:TYPE <String>

:MEASure:THReshold:TYPE?

功能描述

设置测量的阈值类型。

查询测量的阈值类型。

参数说明

<String> 范围 { TOPBase | MAXMin | ABSolute }，

分别表示基顶基底值、最大最小值、绝对值。

返回格式

查询命令返回 TOPBase、MAXMin 或 ABSolute。

实例说明

:MEASure:THReshold:TYPE TOPBase 设置测量阈值类型为基顶基底值。

:MEASure:THReshold:TYPE? -> MAXMin 查询当前测量阈值类型为最大最小值。

:MEASure:THReshold:PERcent

命令格式

:MEASure:THReshold:PERcent <String>,<UInt>

:MEASure:THReshold:PERcent? <String>

功能描述

设置测量基顶基底值/最大最小值的阈值。

查询测量基顶基底值/最大最小值的阈值。

参数说明

<String> 范围 { LOWer | MIDdle | UPper }，

分别表示低阈值、中阈值、高阈值。

<UInt> 10~90 的整数，单位百分比。

返回格式

查询命令以整数形式返回基顶基底值/最大最小值的阈值大小，单位百分比。

实例说明

:MEASure:THReshold:PERcent LOWer,10　　设置基顶基底值/最大最小值的低阈值为 10%。

:MEASure:THReshold:PERcent? MIDdle -> 50　　查询当前基顶基底值/最大最小值的中阈值为 50%。

:MEASure:THReshold:ABSolute

命令格式

:MEASure:THReshold:ABSolute <String>,<Double>

:MEASure:THReshold:ABSolute? <String>

功能描述

设置测量绝对值的阈值。

查询测量绝对值的阈值。

参数说明

<String> 范围 { LOWer | MIDdle | UPper }，

分别表示低阈值、中阈值、高阈值。

<Double> 阈值，范围-40000.0~40000.0。

返回格式

查询命令以实型形式返回绝对值的阈值大小。

实例说明

:MEASure:THReshold:ABSolute LOWer,-1	设置绝对值的低阈值为-1。
:MEASure:THReshold:ABSolute? LOWer -> -1	查询当前绝对值的低阈值为-1。

:MEASure:SCOPE:MODE

命令格式

:MEASure:SCOPE:MODE <String>

:MEASure:SCOPE:MODE?

功能描述

设置测量区域。

查询测量区域。

参数说明

<String> 范围 { RECORD | MAIN | Zoom1 | Zoom2 | CURSOR }，

分别表示全内存、主时基、Zoom1 区域、Zoom2 区域、光标区域。

返回格式

查询命令返回 RECORD、MAIN、Zoom1、Zoom2 或 CURSOR。

实例说明

:MEASure:SCOPE:MODE RECORD	设置测量区域为全内存。
:MEASure:SCOPE:MODE? -> RECORD	查询当前测量区域为全内存。

:MEASure<x>:ADD

命令格式

:MEASure<x>:ADD

功能描述

增加测量项。

参数说明

<x> 测量项编号。

实例说明

:MEASure1:ADD 添加测量 1。

:MEASure<x>:REMOve

命令格式

:MEASure<x>:REMOve

功能描述

删除测量项。

参数说明

<x> 测量项编号。

实例说明

:MEASure1:REMOve　　删除测量 1。

:MEASure:REMOve:ALL

命令格式

:MEASure:REMOve:ALL

功能描述

删除所有测量项。

参数说明

无。

实例说明

无。

:MEASure<x>:TYPE

命令格式

:MEASure<x>:TYPE <String>

功能描述

设置测量项的测量类型。

参数说明

<x> 测量项编号。

<String>

电压类测量项目：

{ Pk-Pk | Ampl | Max | Min | Top | Base | Over | Pre | Avg | DCRMS | ACRMS | Ratio | V-Mean }，分别表示峰峰值、幅度、最大值、最小值、顶部值、底部值、过冲、预冲、平均值、直流有效值、交流有效值、比率、校准平均值。

时间类测量项目：

{ Period | Freq | RiseTime | FallTime | PulseWD | Duty | BurstWD | PulseTrain | X | Delay | Phase | SetupHold | Baudrate }，分别表示周期、频率、上升时间、下降时间、脉宽、占空比、突发宽度、脉冲串长度、位置、延迟、相位、建立保持、波特率。

其他类：

{ EdgeCount | PulseCnt | TriggerCnt | Area | PArea | NArea }，分别表示边沿计数、脉冲计数、触发计数、面积、负面积、正面积。

实例说明

:MEASure1:TYPE Ampl　　设置测量 1 的测量类型为幅度。

:MEASure<x>:SOURce<x>

命令格式

:MEASure<x1>:SOURce<x2> <String>

功能描述

设置测量项的信源。

参数说明

<x1> 测量项编号。

<x2> 范围{ 1 | 2 }，信源编号。

注：单个信源默认编号 1；有多个信源时，时钟信源默认编号 1，数据信源默认编号 2。

<String> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 }。

实例说明

:MEASure1:SOURce1 CHANnel1　　设置测量 1 的信源 1 为通道 1。

:MEASure<x>:SELEction

命令格式

:MEASure<x>:SELEction <String>

功能描述

设置平均、直流有效值、交流有效值、比率、面积、负面积和正面积等测量项的类型选择。

参数说明

<x> 测量项编号。

<String> 范围

{DC | CYCLe}，类型选择为周期；

{AC | DISPlay}，类型选择为全屏。

实例说明

:MEASure1:SELEction DC　　设置测量 1 的类型选择为周期。

:MEASure<x>:EDGE<x>

命令格式

:MEASure<x1>:EDGE<x2> <String>

功能描述

设置延迟和相位的测量项信源的边沿类型。

参数说明

<x1> 测量项编号。

<x2> 范围{ 1 | 2 }，信源编号。

<String> 范围{ Rise | Fall }，分别表示上升沿、下降沿。

实例说明

:MEASure1:EDGE1 Rise　　设置测量 1 的信源 1 的边沿类型为上升沿。

:MEASure<x>:NUM<x>

命令格式

:MEASure<x1>:NUM<x2> <UInt>

功能描述

设置延迟测量项信源的测量边沿。

参数说明

<x1> 测量项编号。

<x2> 范围{ 1 | 2 }，信源编号。

<UInt> 边沿数。

实例说明

:MEASure1:NUM1 1　　设置测量 1 的信源 1 的测量边沿为 1。

:MEASure<x>:SETTINGTYPE

命令格式

:MEASure<x>:SETTINGTYPE <String>

:MEASure<x>:SETTINGTYPE?

功能描述

设置测量项使用的阈值和测量范围应用于全局或局部测量项。

查询测量项使用的阈值和测量范围的应用范围。

参数说明

<x> 测量项编号。

<String> 范围{Global | Local}，分别表示全局、局部。

返回格式

查询命令返回 Global 或 Local。

实例说明

:MEASure1:SETTINGTYPE Global 设置测量 1 的阈值和测量范围应用于全局测量项。

:MEASure1:SETTINGTYPE? -> Local 查询测量 1 的阈值和测量范围仅应用于测量 1。

:MEASure<x>:LOCAL:SCOPE:MODE

命令格式

:MEASure<x>:LOCAL:SCOPE:MODE <String>

:MEASure<x>:LOCAL:SCOPE:MODE?

功能描述

设置局部测量范围。

查询局部测量范围。

注：测量项使用的阈值和测量范围应用设置为**局部**时有效，相关指令:MEASure<x>:SETTINGTYPE。

参数说明

<x> 测量项编号。

<String> 范围{ RECORD | MAIN | Zoom1 | Zoom2 | CURSOR }

分别表示全内存、主时基、Zoom1 区域、Zoom2 区域、光标区域。

返回格式

查询命令返回 RECORD、MAIN、Zoom1、Zoom2 或 CURSOR。

实例说明

:MEASure1:LOCAL:SCOPE:MODE RECORD 设置测量 1 测量范围为全内存。

:MEASure1:LOCAL:SCOPE:MODE? -> RECORD 查询当前测量 1 测量范围为全内存。

:MEASure<x>:LOCAL:THReshold:TYPE

命令格式

:MEASure<x>:LOCAL:THReshold:TYPE <String>

:MEASure<x>:LOCAL:THReshold:TYPE?

功能描述

设置测量的局部阈值类型。

查询测量的局部阈值类型。

注：测量项使用的阈值和测量范围应用设置为**局部**时有效，相关指令:MEASure<x>:SETTINGTYPE。

参数说明

<x> 测量项编号。

<String> 范围{ TOPBase | MAXMin | ABSolute }

分别表示基顶基底值、最大最小值、绝对值。

返回格式

查询命令返回 TOPBase、MAXMin 或 ABSolute。

实例说明

:MEASure1:LOCAL:THReshold:TYPE TOPBase 设置测量 1 的阈值类型为基顶基底值。

:MEASure1:LOCAL:THReshold:TYPE? -> MAXMin 查询当前测量 1 的阈值类型为最大最小值。

:MEASure<x>:LOCAL:THReshold:PERcent

命令格式

:MEASure<x>:LOCAL:THReshold:PERcent <String>,<UInt>

:MEASure<x>:LOCAL:THReshold:PERcent? <String>

功能描述

设置局部测量基顶基底值/最大最小值的阈值。

查询局部测量基顶基底值/最大最小值的阈值。

注：测量项使用的阈值和测量范围应用设置为**局部**时有效，相关指令:MEASure<x>:SETTINGTYPE。

参数说明

<x> 测量项编号。

<String> 范围{ LOWer | MIDdle | UPper }，分别表示低阈值、中阈值、高阈值。

<UInt> 范围 10~90，单位百分比。

返回格式

查询命令以整数形式返回局部测量基顶基底值/最大最小值的阈值大小，单位百分比。

实例说明

:MEASure1:LOCAL:THReshold:PERcent LOWer,10 设置测量 1 基顶基底值/最大最小值的低阈值为 10%

:MEASure1:LOCAL:THReshold:PERcent? MIDdle -> 50 查询当前测量 1 的基顶基底值/最大最小值的中阈值为 50%。

:MEASure<x>:LOCAL:THReshold:ABSolute

命令格式

:MEASure<x>:LOCAL:THReshold:ABSolute <String>, <Double>

:MEASure<x>:LOCAL:THReshold:ABSolute? <String>

功能描述

设置局部测量绝对值的阈值。

查询局部测量绝对值的阈值。

注：测量项使用的阈值和测量范围应用设置为**局部**时有效，相关指令:MEASure<x>:SETTINGTYPE。

参数说明

<x> 测量项编号。

<String> 范围{ LOWer | MIDdle | UPper }，分别表示低阈值、中阈值、高阈值。

<Double> 阈值，范围-40000.0~40000.0。

返回格式

查询命令以实型形式返回局部测量基顶基底值/最大最小值的阈值大小，单位百分比。

实例说明

:MEASure1:LOCAL:THReshold:ABSolute LOWer, -1　　设置测量 1 的绝对值的低阈值为-1。

:MEASure1:LOCAL:THReshold:ABSolute? LOWer -> -1　　查询当前测量 1 的绝对值的低阈值为-1。

:MEASure<x>:STAte

命令格式

:MEASure<x>:STAte?

功能描述

查询某一测量项状态。

参数说明

<x> 需要查询的测量项编号。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:MEASure1:STAte? -> 1　　查询测量项 1 为打开状态。

:MEASure<x>:CURRent

命令格式

:MEASure<x>:CURRent?

功能描述

查询测量项的当前值。

参数说明

<x> 测量项编号。

返回格式

查询命令以实型形式返回测量项的当前值。

实例说明

无。

:MEASure<x>:MAXImum

命令格式

:MEASure<x>:MAXImum?

功能描述

查询测量项的最大值。

参数说明

<x> 测量项编号。

返回格式

查询命令以实型形式返回测量项的最大值。

实例说明

无。

:MEASure<x>:MINImum

命令格式

:MEASure<x>:MINImum?

功能描述

查询测量项的最小值。

参数说明

<x> 测量项编号。

返回格式

查询命令以实型形式返回测量项的最小值。

实例说明

无。

:MEASure<x>:AVERage

命令格式

:MEASure<x>:AVERage?

功能描述

查询测量项的平均值。

参数说明

<x> 测量项编号。

返回格式

查询命令以实型形式返回测量项的平均值。

实例说明

无。

:MEASure<x>:DEViation

命令格式

:MEASure<x>:DEViation?

功能描述

查询测量项的标准方差值。

参数说明

<x> 测量项编号。

返回格式

查询命令以实型形式返回测量项的标准方差值。

实例说明

无。

:MEASure<x>:COUNt

命令格式

:MEASure<x>:COUNt?

功能描述

查询测量项的计数值。

参数说明

<x> 测量项编号。

返回格式

查询命令以整数形式返回测量项的计数值。

实例说明

无。

:MEASure<x>:AVAilable

命令格式

:MEASure<x>:AVAilable?

功能描述

查询测量项的测量状态。

参数说明

<x> 测量项编号。

返回格式

读取状态的命令，会返回三种结果，分别为：Valid，?，Invalid，

对应为：有效的，不准确的（如输入波形超出量程），无效的。

注：无效值在测量结果中显示为“----”。

返回格式

查询命令以字符串形式返回测量项的测量状态。

实例说明

:MEASure1:AVAilable -> Valid　　查询当前测量 1 的测量状态为有效。

:MEASure:OPEN

命令格式

:MEASure:OPEN

功能描述

使能测量功能。

参数说明

无。

实例说明

:MEASure:OPEN 使能测量功能。

:MEASure:CLOSe

命令格式

:MEASure:CLOSe

功能描述

关闭测量功能。

参数说明

无。

实例说明

:MEASure:CLOSe	关闭测量功能。

:MEASure:STAtics

命令格式

:MEASure:STAtics <Bool>

:MEASure:STAtics?

功能描述

设置测量统计开关状态。

查询测量统计开关状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开测量统计开关；

{OFF | FALSE | 0} 关闭测量统计开关。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:MEASure:STAtics ON	打开测量统计开关。
:MEASure:STAtics? -> 1	查询当前测量统计开关为打开状态。

:MEASure:FREQuency:METER:TYPe

命令格式

:MEASure:FREQuency:METER:TYPe <String>

:MEASure:FREQuency:METER:TYPe?

功能描述

设置硬件频率计的显示类型。

查询硬件频率计的显示类型。

参数说明

<String> 范围{OFF | Freq | Baudrate}，分别表示关闭硬件频率计、显示频率、显示波特率。

返回格式

查询命令返回 OFF、Freq 或 Baudrate。

实例说明

:MEASure:FREQuency:METER:TYPe Freq 设置硬件频率计的显示类型为频率。

:MEASure:FREQuency:METER:TYPe? -> Freq 查询当前硬件频率计显示类型为频率。

:MEASure:ITEM:STAtics

命令格式

:MEASure:ITEM:STAtics? <String_1>,<String_CH>[,<String_CH>]

功能描述

查询是否存在特定测量项。

参数说明

<String_1> 测量项名。

电压类测量项目：

{ Pk-Pk | Ampl | Max | Min | Top | Base | Over | Pre | Avg | DCRMS | ACRMS | Ratio | V-Mean }，分别表示峰峰值、幅度、最大值、最小值、顶部值、底部值、过冲、预冲、平均值、直流有效值、交流有效值、比率、校准平均值。

时间类测量项目：

{ Period | Freq | RiseTime | FallTime | PulseWD | Duty | BurstWD | PulseTrain | X | Delay | Phase | SetupHold | Baudrate }，分别表示周期、频率、上升时间、下降时间、脉宽、占空比、突发宽度、脉冲串长度、位置、延迟、相位、建立保持、波特率。

其他类：

{ EdgeCount | PulseCnt | TriggerCnt | Area | PArea | NArea }，分别表示边沿计数、脉冲计数、触发计数、面积、负面积、正面积。

<String_CH>

{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 }。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：存在某个特定测量项，不存在某个特定测量项。

实例说明

:MEASure:ITEM:STAtics? Max, channel1 -> 1　　查询当前存在测量通道 1 的最大值的测量项。

14. 运行控制相关命令

:RUNctrl 命令用于控制示波器的运行状态。

- :RUN 运行
- :STOP 停止
- :SINGle 单次触发
- :DEFault 恢复默认设置
- :DEFault:CANCEL 取消恢复默认设置
- :AUTO 一键捕获
- :AUTO:CANCEL 取消一键捕获

:RUN

命令格式

:RUN

功能描述

使示波器处于运行状态，与在示波器前面板上按下 RUN 按键功能一致。

参数说明

无。

实例说明

无。

:STOP

命令格式

:STOP

功能描述

使示波器处于停止状态，与在示波器前面板上按下 Stop 按键功能一致。

参数说明

无。

实例说明

无。

:SINGle

命令格式

:SINGle

功能描述

执行示波器的单次捕获功能，与在示波器前面板中按下 Single 的功能一致。

参数说明

无。

实例说明

无。

:DEFault

命令格式

:DEFault

功能描述

执行示波器的默认设置功能，与在示波器前面板上按下 Default Setup 按键功能一致。

参数说明

无。

实例说明

无。

:DEFault:CANCEL

命令格式

:DEFault:CANCEL

功能描述

取消默认配置。

参数说明

无。

实例说明

无。

:AUTO

命令格式

:AUTO

功能描述

执行示波器的自动定标功能，与在示波器前面板上按下 Auto Setup 的功能一致。

参数说明

无。

实例说明

无。

:AUTO:CANCEL

命令格式

:AUTO:CANCEL

功能描述

取消自动配置。

参数说明

无。

实例说明

无。

15. 参考波形相关命令

:Ref 命令用于查询和设置参考波形通道信息。

- :REF<x>:STAte 设置和查询参考通道的使能。
- :REF<x>:DISPlay 设置和查询参考通道显示状态。
- :REF<x>:SCALe 设置和查询参考通道的垂直档位。
- :REF<x>:OFFSet 设置和查询参考通道的偏移。
- :REF<x>:UNit 设置参考通道的单位。
- :REF<x>:SOURce 设置和查询参考通道的信源。
- :REF<x>:SAVE 进行暂存波形。
- :REF<x>:CLEAr 清除暂存波形。
- :REF<x>:IMPOrt 导入参考波形数据。
- :REF<x>:EXPOrt 导出参考波形数据。

:REF<x>:STAte

命令格式

:REF<x>:STAte <Bool>

:REF<x>:STAte?

功能描述

设置参考通道的使能。

查询参考通道的使能。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开参考通道；

{OFF | FALSE | 0} 关闭参考通道。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:REF1:STAte 1	打开参考通道 1。
:REF1:STAte? -> 1	查询当前参考通道 1 为打开状态。

:REF<x>:DISPlay

命令格式

:REF<x>:DISPlay <Bool>

:REF<x>:DISPlay?

功能描述

设置参考通道显示状态。

查询参考通道显示状态。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 显示参考通道；

{OFF | FALSE | 0} 隐藏参考通道。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:REF1:DISPlay ON	显示参考通道 1。
:REF1:DISPlay? -> 1	查询当前参考通道 1 为显示状态。

:REF<x>:SCALe

命令格式

:REF<x>:SCALe <Double>

:REF<x>:SCALe?

功能描述

设置参考通道的垂直档位。

查询参考通道的垂直档位。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<Double> 档位，可带单位，如“2mV”，也可用科学计数法表示。

返回格式

查询命令以实型形式返回参考通道的垂直档位。

实例说明

:REF1:SCALe 2mV	设置参考通道 1 垂直档位为 2mV。
:REF1:SCALe? -> 0.002	查询当前参考通道 1 垂直档位为 2mV。

:REF<x>:OFFSet

命令格式

:REF<x>:OFFSet <Double>

:REF<x>:OFFSet?

功能描述

设置参考通道的偏移。

查询参考通道的偏移。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<Double> 偏移量，可带单位，如“2mV”，也可用科学计数法表示。

返回格式

查询命令以实型形式返回参考通道的偏移量。

实例说明

:REF1:OFFSet 2mV	设置参考通道 1 垂直偏移为 2mV。
:REF1:OFFSet? -> 0.002	查询当前参考通道 1 垂直偏移为 2mV。

:REF<x>:UNit

命令格式

:REF<x>:UNit <String>

:REF<x>:UNit?

功能描述

设置参考通道的单位。

查询参考通道的单位。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<String> 参考通道的显示单位。

返回格式

查询命令以实型形式返回参考通道的单位。

实例说明

:REF1:UNit W	设置参考 1 的显示单位为 W。
:REF1:UNit? -> W	查询当前参考 1 的显示单位为 W。

:REF<x>:SOURce

命令格式

:REF<x>:SOURce <String>

:REF<x>:SOURce?

功能描述

设置参考通道的信源。

查询参考通道的信源。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<String> 范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | MATH1 | MATH2 | MATH3 | MATH4 | REF1 | REF2 | REF3 | REF4 }。

注：不可设置本身作为暂存信源。

返回格式

查询命令以实型形式返回参考通道的信源。

实例说明

:REF1:SOURce CHANnel1	设置参考 1 的信源通道为通道 1。
:REF1:SOURce? -> CHANnel1	查询当前参考 1 的信源通道为通道 1。

:REF<x>:SAVE

命令格式

:REF<x>:SAVE

功能描述

进行暂存波形。

注：滚动状态下不支持暂存波形。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

返回格式

无。

实例说明

:REF1:SAVE　　参考 1 进行暂存波形。

:REF<x>:CLEAr

命令格式

:REF<x>:CLEAr

功能描述

清除暂存波形。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

返回格式

无。

实例说明

:REF1:CLEAr　　清除参考 1 暂存波形。

:REF<x>:IMPOrt

命令格式

:REF<x>:IMPOrt <String>

功能描述

导入参考波形数据。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<String> 导入路径和文件名，如“/flash/wave.wfm”。

实例说明

:REF1:IMPOrt /flash/wave.wfm 加载 wave.wfm 波形数据。

:REF<x>:EXPOrt

命令格式

:REF<x>:EXPOrt <String>

功能描述

导出参考波形数据。

参数说明

<x> 需要查询或改变的参考通道编号，1、2、3 或 4。

<String> 导出路径和文件名，如“/flash/wave.wfm”。

注：文件名必须以“*.wfm”为尾缀。

实例说明

:REF1:EXPOrt /flash/wave.wfm　　导出参考 1 波形数据，文件命名为 wave。

16. 存储相关命令

:SERIal 命令导入导出系统设置和一键截图。

- :SERIal 执行一次序列化。
- :SERIal:EXPort 导出系统设置。
- :SERIal:IMPort 导入系统设置。
- :STORage:PRINT 截图。
- :STORage:PRINT:PATH 设置截图保存的路径。

:SERIal

命令格式

:SERIal

功能描述

执行一次序列化。

参数说明

无。

实例说明

无。

:SERIal:EXPort

命令格式

:SERIal:EXPort <String>

功能描述

导出系统设置。

参数说明

<String> 导出路径和文件名，如“/flash/ZTMI001.stp”。

注：文件名必须以“*.stp”为尾缀。

实例说明

:SERIal:EXPort /flash/ZTMI001.stp

将本次系统设置文件以 stp 文件格式导出到示波器本地内存，并将文件名命名为 ZTMI001。

:SERIal:IMPort

命令格式

:SERIal:IMPort <String>

功能描述

导入系统设置。

参数说明

<String> 导入路径和文件名，如“/flash/ZTMI001.stp”。

实例说明

:SERIal:IMPort /flash/ZTMI001.stp

加载 ZTMI001.stp 文件到示波器。

:STORage:PRINT

命令格式

:STORage:PRINT [<String>]

功能描述

截图。

参数说明

<String> 导出文件名，如“123.bmp”或“123.png”。

注：不带参数时按照截图设置界面设置进行截图保存。若文件存在则会覆盖文件。

实例说明

:STORage:PRINT 按照截图设置界面设置进行截图。

:STORage:PRINT 123.bmp 截图导出文件名为 123.bmp 的图片，截图路径为示波器截图设置界面路径。

:STORage:PRINT 123.png 截图导出文件名为 123.png 的图片，截图路径为示波器截图设置界面路径。

:STORage:PRINT:PATH

命令格式

:STORage:PRINT:PATH <String>

功能描述

设置截图保存的路径，若路径不存在则不会生效。

参数设置

<String> 导出路径，如“/flash”。

实例说明

:STORage:PRINT:PATH /flash　　设置截图保存路径为/flash。

17. 系统设置相关命令

:SYStem 命令用于查询和设置系统设置、网络设置、时间设置、以及查询系统信息。

- :SYSTem:ERRor[:NEXT] 查询并删除最新的一条系统错误信息。
- :SYSTem:ERRor:COUNt 查询当前的系统错误个数。
- :SYStem:VERSion:SELL? 查询销售版本。
- :SYStem:VERSion:SOFTware? 查询软件版本。
- :SYStem:DEVice:NAME? 查询设备名称。
- :SYStem:DEVice:ID? 查询序列号。
- :SYStem:MANUFACTURE? 查询生产厂商。
- :SYStem:MAC:ADDRess? 查询 MAC 地址。
- :SYStem:LANGuage 设置和查询当前的系统使用的语言。
- :SYStem:LANGuage:LIST? 查询当前的系统支持的语言。
- :SYStem:DATE 设置和查询示波器的系统日期。
- :SYStem:TIME 设置和查询示波器的系统时间。
- :SYStem:ZONE 设置示波器的时区。
- :SYStem:RESUlt:DOUBle:FORMat 设置和查询示波器 scpi 返回的浮点值精度。
- :SYStem:RESUlt:INT:FORMat 设置和查询示波器 scpi 返回的整数值的格式。
- :SYStem:RESet 恢复默认设置。
- :SYStem:RESet:DEFault 取消恢复默认设置。
- :SYStem:RESet:FACTory 恢复出厂设置
- :SYSTEM:LOCK 设置和查询示波器锁定状态。
- :SYSTEM:TIps 设置示波器显示提示信息。
- [:SYStem]:SCPI:CLIENT? 查询系统示波器连接至客户端的状态。
- [:SYStem]:SCPI:ECHO 设置和查询示波器 scpi 应答状态。
- :SYSTem[:LAN]:IP:TYPE 设置和查询示波器的网络类型。
- :SYSTem[:LAN]:IP:ADDR 设置和查询示波器的静态 IP 网络地址。
- :SYSTem[:LAN]:IP:MASK 设置和查询示波器的静态 IP 的子网掩码。
- :SYSTem[:LAN]:IP:GATEway 设置和查询示波器的静态 IP 的默认网关。

:SYSTem:ERRor

命令格式

:SYSTem:ERRor?

功能描述

查询并删除最新的一条系统错误信息。

返回格式

查询命令返回包含错误编号和错误内容字符串，其中错误编号是一个整数，错误内容是 ASCII 字符串。如返回：-113,”Undefined header”。

:SYSTem:ERRor:COUNt

命令格式

:SYSTem:ERRor:COUNt?

功能描述

查询当前的系统错误个数。

返回格式

查询命令返回以整型表示的错误个数。

:SYStem:VERSion:SELL

命令格式

:SYStem:VERSion:SELL?

功能描述

查询销售版本。

返回格式

查询命令以字符串形式返回当前的销售版本。

实例说明

:SYStem:VERSion:SELL? -> V1.00

查询当前销售版本为 V1.00。

:SYStem:VERSion:SOFTware

命令格式

:SYStem:VERSion:SOFTware?

功能描述

查询软件版本。

返回格式

查询命令以字符串形式返回当前的软件版本。

实例说明

:SYStem:VERSion:SOFTware? -> 1.2.4.2251455

查询当前软件版本为 1.2.4.2251455。

:SYStem:DEVice:NAME

命令格式

:SYStem:DEVice:NAME?

功能描述

查询设备名称。

返回格式

查询命令以字符串形式返回当前的设备名称。

实例说明

:SYStem:DEVice:NAME? -> ZUS6104

查询当前设备名称为 ZUS6104。

:SYStem:DEVice:ID

命令格式

:SYStem:DEVice:ID?

功能描述

查询序列号。

返回格式

查询命令以字符串形式返回序列号。

实例说明

:SYStem:DEVice:ID? -> 7842001042210290010

查询序列号为 7842001042210290010。

:SYStem:MANUFACTURE

命令格式

:SYStem:MANUFACTURE?

功能描述

查询生产厂商。

返回格式

查询命令以字符串形式返回生产厂商。

实例说明

:SYStem:MANUFACTURE? -> Zhiyuan Instruments

查询生产厂商为 Zhiyuan Instruments。

:SYStem:MAC:ADDRess

命令格式

:SYStem:MAC:ADDRess?

功能描述

查询 MAC 地址。

返回格式

查询命令以字符串形式返回 MAC 地址。

实例说明

:SYStem:MAC:ADDRess? -> 00:14:97:01:00:03

查询 MAC 地址为 00:14:97:01:00:03。

:SYStem:LANGuage

命令格式

:SYStem:LANGuage <String>

:SYStem:LANGuage?

功能描述

设置当前的系统使用的语言。

查询当前的系统使用的语言。

参数说明

<String> 范围{Chinese | English}，分别表示中文，英文。

返回格式

查询命令返回 Chinese 或 English。

实例说明

:SYStem:LANGuage Chinese	设置系统使用的语言为中文。
:SYStem:LANGuage? -> Chinese	查询当前系统使用的语言为中文。

:SYStem:LANGuage:LIST

命令格式

:SYStem:LANGuage:LIST?

功能描述

查询当前的系统支持的语言。

返回格式

查询命令以字符串形式返回当前的系统支持的语言。

实例说明

:SYStem:LANGuage:LIST? -> Chinese,English

查询当前系统支持的语言为中文和英文。

:SYStem:DATE

命令格式

:SYStem:DATE <year>,<month>,<day>

:SYStem:DATE?

功能描述

设置示波器的系统日期。

查询示波器的系统日期。

参数说明

<year> 年。

<month> 月。

<day> 日。

返回格式

查询命令返回年月日，以逗号间隔。

实例说明

:SYStem:DATE 2023,4,10	设置示波器的系统日期为 2023/4/10。
:SYStem:DATE? -> 2023,4,10	查询当前示波器的系统日期为 2023/4/10。

:SYStem:TIME

命令格式

:SYStem:TIME <hour>,<minute>,<second>

:SYStem:TIME?

功能描述

设置示波器的系统时间。

查询示波器的系统时间。

参数说明

<hour> 小时，范围 0-23。

<minute> 分钟，范围 0-59。

<second> 秒，范围 0-59。

返回格式

查询命令返回时分秒，以逗号间隔。

实例说明

:SYStem:TIME 19,0,0	设置示波器的系统时间为 19:0:0。
:SYStem:TIME? -> 19,0,0	查询当前示波器的系统时间为 19:0:0。

:SYStem:ZONE

命令格式

:SYStem:ZONE <Int>

功能描述

设置示波器的时区。

参数说明

<Int> 与 0 时区的差值，以分钟为单位，中国+8 时区为-8*60(东 8 区)，一共有东西 12 区。

实例说明

:SYStem:ZONE -8*60　　设置示波器的时区为东 8 区。

:SYStem:RESUlt:DOUBle:FORMat

命令格式

:SYStem:RESUlt:DOUBle:FORMat <String>,<UInt>

:SYStem:RESUlt:DOUBle:FORMat?

功能描述

设置示波器 scpi 返回的浮点值精度。

查询示波器 scpi 返回的浮点值精度。

参数说明

<String> 范围{G | F | E}，分别表示默认显示模式，固定小数位，科学计数显示。

<UInt> 显示位数，默认为 8 位。

返回格式

查询命令以整数形式返回 scpi 返回的浮点值精度。

实例说明

:SYStem:RESUlt:DOUBle:FORMat G,10　　设置示波器 scpi 返回的浮点值精度为默认显示模式，共显示 10 位。

:SYStem:RESUlt:DOUBle:FORMat? -> G,10　　查询示波器 scpi 返回的浮点值精度为默认显示模式，共显示 10 位。

:SYStem:RESUlt:INT:FORMat

命令格式

:SYStem:RESUlt:INT:FORMat <String>

:SYStem:RESUlt:INT:FORMat?

功能描述

设置示波器 scpi 返回的整数值的格式。

查询示波器 scpi 返回的整数值的格式。

参数说明

<String> 范围 {DEC | HEX | OCT}，分别表示十进制，十六进制，八进制。默认为十进制。

返回格式

查询命令返回 DEC、HEX 或 OCT。

实例说明

:SYStem:RESUlt:INT:FORMat HEX 设置示波器 scpi 返回的整数值的格式为十六进制。

:SYStem:RESUlt:INT:FORMat? -> DEC 查询示波器 scpi 返回的整数值的格式为十进制。

:SYStem:RESet

命令格式

:SYStem:RESet

功能描述

将示波器通讯接口设置相关以外的设置恢复为默认值，效果相当于在示波器界面上按下Default Setup。

参数说明

无。

实例说明

无。

:SYStem:RESet:DEFault

命令格式

:SYStem:RESet:DEFault

功能描述

将示波器通讯接口设置相关以外的设置恢复为默认值，效果相当于在示波器界面上按下Default Setup。

参数说明

无。

实例说明

无。

:SYStem:RESet:FACTory

命令格式

:SYStem:RESet:FACTory

功能描述

将示波器通讯接口设置相关以外的设置恢复为出厂状态。

参数说明

无。

实例说明

无。

:SYSTEM:LOCK

命令格式

:SYSTEM:LOCK <bool>

:SYSTEM:LOCK?

功能描述

设置示波器锁定状态。

查询示波器锁定状态。

参数说明

<bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 开启示波器锁定；

{OFF | FALSE | 0} 关闭示波器锁定。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:SYSTEM:LOCK true	开启示波器锁定。
:SYSTEM:LOCK? -> 1	查询当前示波器为锁定状态。

:SYSTEM:TIps

命令格式

:SYSTEM:TIps <String>,<UInt>

功能描述

设置示波器显示提示信息。

参数说明

<String> 显示的信息。

<UInt> 显示时长，单位毫秒。

实例说明

:SYSTEM:TIps Ready to export result,2000

设置示波器显示提示信息为 Ready to export result，显示时长为 2000ms。

[:SYStem]:SCPI:CLIENT

命令格式

[:SYStem]:SCPI:CLIENT?

功能描述

查询系统示波器连接至客户端的状态。

参数说明

无。

实例说明

:SCPI:CLIENT? -> 0: type=tcp;ip=127.0.0.1;port=31722; conect on: 2023-04-11 13:42:25

[:SYStem]:SCPI:ECHO

命令格式

[:SYStem]:SCPI:ECHO <bool>

[:SYStem]:SCPI:ECHO?

功能描述

设置示波器 scpi 应答状态。

查询示波器 scpi 应答状态。

参数说明

<bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 开启 scpi 应答；

{OFF | FALSE | 0} 关闭 scpi 应答。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:SCPI:ECHO true	开启 scpi 应答。
:SCPI:ECHO? ->1	查询当前 scpi 应答状态为开启状态。

:SYSTem[:LAN]:IP:TYPE

命令格式

:SYSTem[:LAN]:IP:TYPE <String>

:SYSTem[:LAN]:IP:TYPE?

功能描述

设置示波器的网络类型。

查询示波器的网络类型。

参数说明

<String> 范围{STATic | DHCP}，分别表示静态 IP，动态地址。

返回格式

查询命令返回 STATic 或 DHCP。

实例说明

:SYSTem:IP:TYPE STATic	设置示波器的网络类型为静态 IP。
:SYSTem:IP:TYPE? -> STATic	查询当前示波器的网络类型为静态 IP。

:SYSTem[:LAN]:IP:ADDR

命令格式

:SYSTem[:LAN]:IP:ADDR <String>

:SYSTem[:LAN]:IP:ADDR?

功能描述

设置示波器的静态 IP 网络地址。

查询示波器的静态 IP 网络地址。

注：示波器的网络类型设置为**静态地址**时有效，相关指令:SYSTem[:LAN]:IP:TYPE。

参数说明

<String> 以.作为分割符号的 IP 地址。

返回格式

查询命令返回示波器的静态 IP 网络地址。

实例说明

:SYSTem:IP:ADDR 192.168.138.101　　设置示波器的静态 IP 网络地址为 192.168.138.101。

:SYSTem:IP:ADDR? -> 192.168.138.101　　查询当前示波器的静态 IP 网络地址为 192.168.138.101。

:SYSTem[:LAN]:IP:MASK

命令格式

:SYSTem[:LAN]:IP:MASK <String>

:SYSTem[:LAN]:IP:MASK?

功能描述

设置示波器的静态 IP 的子网掩码。

查询示波器的静态 IP 的子网掩码。

注：示波器的网络类型设置为**静态地址**时有效，相关指令:SYSTem[:LAN]:IP:TYPE。

参数说明

<String> 以.作为分割符号的子网掩码。

返回格式

查询命令返回示波器的静态 IP 的子网掩码。

实例说明

:SYSTem:IP:MASK 255.255.255.0　　设置示波器的静态 IP 的子网掩码为 255.255.255.0。

:SYSTem:IP:MASK? -> 255.255.255.0　　查询当前示波器的静态 IP 的子网掩码为 255.255.255.0。

:SYSTem[:LAN]:IP:GATEway

命令格式

:SYSTem[:LAN]:IP:GATEway <String>

:SYSTem[:LAN]:IP:GATEway?

功能描述

设置示波器的静态 IP 的默认网关。

查询示波器的静态 IP 的默认网关。

注：示波器的网络类型设置为**静态地址**时有效，相关指令:SYSTem[:LAN]:IP:TYPE。

参数说明

<String> 以.作为分割符号的默认网关。

返回格式

查询命令返回示波器的静态 IP 的默认网关。

实例说明

:SYSTem:IP:GATEway 192.168.138.254　　设置示波器的静态 IP 的默认网关为 192.168.138.254。

:SYSTem:IP:GATEway? -> 192.168.138.254　　查询当前示波器的静态 IP 的默认网关为 192.168.138.254。

18. 时序相关命令

:TA 命令组用于查询和修改时序相关配置。

- :TA:RESULT 查询当前时序分析测量结果。
- :TA:STATE 设置和查询时序分析开关状态。。
- :TA:TYPE 设置和查询时序分析类型。
- :TA:REPORT:HTML 导出时序网页报表。
- :TA:REPORT:CSV 导出时序 CSV 文件。
- :TA:REPORT:COUNT 查询开机后执行时序报表导出次数。
- :TA:TABLE:READ:LINE? 查询时序分析表格某行结果。
- :TA:CONFIG:INI 设置和查询时序参数设置。

:TA:RESULT

命令格式

:TA:RESULT?

功能描述

查询当前时序分析测量结果。

参数设置

无。

返回格式

查询命令返回 Pass、Fail 或 No Test。

注：未打开时序分析则返回 No Test。

实例说明

:TA:RESULT? -> Pass　　当前时序分析测量结果为通过。

:TA:STATE

命令格式

:TA:STATE <bool>

:TA:STATE?

功能描述

设置时序分析开关状态。

查询时序分析开关状态。

参数设置

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1}，打开时序功能；

{OFF | FALSE | 0}，关闭时序功能。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:TA:STATE 1　　打开时序功能。

:TA:STATE? -> 1　　查询当前时序功能为关闭状态。

:TA:TYPE

命令格式

:TA:TYPE <String>

:TA:TYPE?

功能描述

设置时序分析类型。

查询时序分析类型

参数说明

<String> 范围{IIC-TA | IIS-TA | SPI-TA | MIPI-RFFE-TA | CAN-TA | Reset-TA}，

分别表示 IIC 时序、IIS 时序、SPI 时序、MIPI-RFFE 时序、CAN 时序、Reset 时序。

实例说明

:TA:TYPE IIC-TA	设置时序分析类型为 IIC。
:TA:TYPE? -> IIC-TA	查询当前时序分析类型为 IIC。

:TA:REPORT:HTML

命令格式

:TA:REPORT:HTML <String>

功能描述

导出时序网页报表。

参数说明

<String> 导出路径和文件名，如“/flash/IIC-TA.html”。

注：文件名必须以“*.html”为尾缀。

实例说明

:TA:REPORT:HTML /flash/IIC-TA.html

将本次时序结果以网页报表的格式导入示波器本地内存，并将文件名命名为 IIC-TA。

:TA:REPORT:CSV

命令格式

:TA:REPORT:CSV <String>

功能描述

导出时序 CSV 文件。

参数说明

<String> 导出路径和文件名，如“/flash/IIC-TA.csv”。

注：文件名必须以“*.csv”为尾缀。

实例说明

:TA:REPORT:CSV /flash/IIC-TA.csv

将本次时序结果以 CSV 文件格式导入示波器本地内存，并将文件名命名为 IIC-TA。

:TA:REPORT:COUNT

命令格式

:TA:REPORT:COUNT?

功能描述

查询开机后执行时序报表导出次数。

参数说明

无。

返回格式

查询命令以整数形式返回开机后执行时序报表导出次数。

实例说明

:TA:REPORT:COUNT? -> 1

开机后时序报表共导出 1 次。

:TA:TABLE:READ:LINE?

命令格式

:TA:TABLE:READ:LINE? <Int>

功能描述

查询时序分析表格某行结果。

参数说明

<Int> 默认第一行为 0，以此类推。

实例说明

:TA:TABLE:READ:LINE? 0 ->

1,Dominant.CAN_H,3.000V,4.250V,3.379V,3.379V,3.379V,Pass

查询 CAN 时序分析表格第一行结果。

:TA:CONFIG:INI

命令格式

:TA:CONFIG:INI <String>

:TA:CONFIG:INI?

功能描述

设置时序参数设置。

查询时序参数设置。

参数说明

详细内容查看各时序手册关于 ini 文件配置参数说明。

实例说明

:TA:CONFIG:INI PluginName=CAN-TA

设置时序分析类型为 CAN 时序。

:TA:CONFIG:INI ->

```
#3187SerialFrom=TimingAnalyze
PluginName=CAN-TA
Baudrate/K=500
BusType=1
Deviation=1
Dominant(MAX)=4.25
Dominant(MIN)=3
ID=0
Recessive(MAX)=3
Recessive(MIN)=2
Screening=0
Source=0
Thresh(V)=3
```

查询 CAN 时序参数设置。

19. 水平时基相关命令

:TIMebase 命令用于查询和设置水平时基的配置。

- :TIMebase:MODE 设置和查询水平模式。
- :TIMebase:SCALe 设置和查询水平时基档位。
- :TIMebase:OFFSet 设置和查询水平偏移量。
- :TIMebase:ZOOM 切换和查询水平档位时基状态。

:TIMebase:MODE

命令格式

:TIMebase:MODE <String>

:TIMebase:MODE?

功能描述

设置水平模式。

查询水平模式。

参数说明

<String> 范围{ YT | XY | ROLL }，分别表示 YT 模式，XY 模式，滚动模式。

返回格式

查询命令返回 YT、XY 或 ROLL。

实例说明

:TIMebase:MODE XY	设置水平模式为 XY 模式。
:TIMebase:MODE? -> ROLL	查询当前水平模式为滚动模式。

:TIMebase:SCALe

命令格式

:TIMebase:SCALe <Double>

:TIMebase:SCALe?

功能描述

设置水平时基档位。

查询水平时基档位。

参数说明

<Double> 水平时基档位，可带单位，如“2ms”。

范围 { 500ps | 1ns | 2ns | 5ns | 10ns | 20ns | 50ns | 100ns | 200ns | 500ns | 1us | 2us | 5us | 10us | 20us | 50us | 100us | 200us | 500us | 1ms | 2ms | 5ms | 10ms | 20ms | 50ms | 100ms | 200ms | 500ms | 1s | 2s | 5s | 10s | 20s | 50s | 100s | 200s | 500s | 1ks }。

也可用科学计数形式表示，如 2e-5 代表 20us/div 的水平档位。

返回格式

查询命令以实型形式返回水平档位，单位 s。

实例说明

:TIMebase:SCALe 20us	设置水平档位为 20us/div。
:TIMebase:SCALe? -> 0.001	查询当前水平档位为 1ms/div。

:TIMebase:OFFSet

命令格式

:TIMebase:OFFSet <Double>

:TIMebase:OFFSet?

功能描述

设置水平偏移量。

查询水平偏移量。

参数说明

<Double> 偏移量，可带单位，如“2ms”。

也可用科学计数法形式表示，如 2e-5 代表 20us/div 的水平偏移量。

返回格式

查询命令以实型形式返回水平偏移量，单位 S。

实例说明

:TIMebase:OFFSet 2ms	设置水平偏移为 2ms。
:TIMebase:OFFSet? -> 0.002	查询当前水平偏移为 2ms

:TIMebase:ZOOM

命令格式

:TIMebase:ZOOM <String>

:TIMebase:ZOOM?

功能描述

切换水平档位时基状态。

查询水平档位时基状态。

参数说明

<String> 范围{ MAIN | ONEZOOM | TWOZOOM }，分别表示主时基、单 ZOOM、双 ZOOM。

返回格式

查询命令返回 MAIN、ONEZOOM 或 TWOZOOM。

实例说明

:TIMebase:ZOOM ONEZOOM	开启单 ZOOM 功能。
:TIMebase:ZOOM? -> MAIN	查询当前为主时基功能。

20. 触发相关命令

:TRIGger 命令用于查询和设置触发配置。

- :TRIGger:SWEep 设置和查询触发方式。
- :TRIGger:HOLDoff 设置和查询触发释抑。
- :TRIGger:SENSitivity 设置和查询触发灵敏度。
- :TRIGger:COUPling 设置和查询触发耦合。
- :TRIGger:MODE 设置和查询触发类型。
- :TRIGger:LEVEl:CHANnel<x> 设置和查询垂直通道的触发阈值。
- :TRIGger:HLEVel:CHANnel<x> 设置和查询 TH 触发高阈值。
- :TRIGger:LLEVel:CHANnel<x> 设置和查询 TL 触发低阈值。
- :TRIGger:LEVEl:EXT 设置和查询外触发阈值。
- :TRIGger:EDGe:SOURce 设置和查询边沿触发的信源选择。
- :TRIGger:EDGe:TYPE 设置和查询边沿触发的边沿类型。
- :TRIGger:PULSe:SOURce 设置和查询脉宽触发的信源选择。
- :TRIGger:PULSe:WHEN 设置和查询脉宽触发的脉宽类型。
- :TRIGger:PULSe:LOWer 设置和查询脉宽触发的脉宽下限。
- :TRIGger:PULSe:UPPer 设置和查询脉宽触发的脉宽上限。
- :TRIGger:SLOPe:SOURce 设置和查询斜率触发的信源选择。
- :TRIGger:SLOPe:WHEN 设置和查询斜率触发的触发模式。
- :TRIGger:SLOPe:LOWer 设置和查询斜率触发的时间下限。
- :TRIGger:SLOPe:UPPer 设置和查询斜率触发的时间上限。
- :TRIGger:VIDEo:SOURce 设置和查询视频触发的信源选择。
- :TRIGger:VIDEo:POLArity 设置和查询视频触发的极性。
- :TRIGger:VIDEo:STANdard 设置和查询视频触发的视频格式。
- :TRIGger:VIDEo:MODE 设置和查询视频触发的触发模式。
- :TRIGger:VIDEo:LINE 设置和查询视频触发的指定触发行。
- :TRIGger:RUNT:SOURce 设置和查询欠幅触发的信源选择。
- :TRIGger:RUNT:TYPE 设置和查询欠幅触发的脉冲类型。
- :TRIGger:RUNT:WHEN 设置和查询欠幅触发的限定符。
- :TRIGger:RUNT:LOWer 设置和查询欠幅触发的脉宽下限。
- :TRIGger:RUNT:UPPer 设置和查询欠幅触发的脉宽上限。
- :TRIGger:PRUNt:SOURce 设置和查询超幅触发的信源选择。
- :TRIGger:PRUNt:TYPE 设置和查询超幅触发的脉冲类型。
- :TRIGger:PRUNt:WHEN 设置和查询超幅触发的限定符。
- :TRIGger:PRUNt:LOWer 设置和查询超幅触发的脉宽下限。
- :TRIGger:PRUNt:UPPer 设置和查询超幅触发的脉宽上限。

- :TRIGger:PATTern:ASRc 设置和查询码型触发的 A 信源。
- :TRIGger:PATTern:BSRc 设置和查询码型触发的 B 信源。
- :TRIGger:PATTern:APat 设置和查询码型触发的 A 码型。
- :TRIGger:PATTern:BPat 设置和查询码型触发的 B 码型。
- :TRIGger:PATTern:WHEN 设置和查询码型触发的限定符。
- :TRIGger:PATTern:LOWer 设置和查询码型触发的时间下限。
- :TRIGger:PATTern:UPPer 设置和查询码型触发的时间上限。
- :TRIGger:NEDGe:SOURce 设置和查询第 N 边沿触发的信源选择。
- :TRIGger:NEDGe:TYPE 设置和查询第 N 边沿触发的边沿类型。
- :TRIGger:NEDGe:NUM 设置和查询第 N 边沿触发的边沿类型。
- :TRIGger:NEDGe:IDLE 设置和查询第 N 边沿触发的空闲时间。
- :TRIGger:TIMeout:SOURce 设置和查询超时触发的信源选择。
- :TRIGger:TIMeout:TYPE 设置和查询超时触发的触发模式。
- :TRIGger:TIMeout:TIMe 设置和查询超时触发的超时时间。
- :TRIGger:SHOLd:CSrc 设置和查询建立保持触发的时钟通道。
- :TRIGger:SHOLd:DSrc 设置和查询建立保持触发的数据通道。
- :TRIGger:SHOLd:TYPE 设置和查询建立保持触发的采样类型。
- :TRIGger:SHOLd:PATTern 设置和查询建立保持触发的数据类型。
- :TRIGger:SHOLd:MODE 设置和查询建立保持触发的触发类型。
- :TRIGger:SHOLd:STIMe 设置和查询建立保持触发的建立时间。
- :TRIGger:SHOLd:HTIMe 设置和查询建立保持触发的保持时间。

:TRIGger:SWEep

命令格式

:TRIGger:SWEep <String>

:TRIGger:SWEep?

功能描述

设置触发方式。

查询触发方式。

参数说明

<String> 范围 { AUTO | NORmal }，分别表示自动，普通。

返回格式

查询命令返回 AUTO 或 NORmal。

实例说明

:TRIGger:SWEep AUTO	设置触发方式为自动。
:TRIGger:SWEep? -> NORmal	查询当前触发方式为普通。

:TRIGger:HOLDoff

命令格式

:TRIGger:HOLDoff <Double>

:TRIGger:HOLDoff?

功能描述

设置触发释抑。

查询触发释抑。

参数说明

<Double> 范围 0ns~13.7s，用科学计数法形式表示，默认单位 s，如 3.2e-9 代表 3.20ns 的触发释抑，最小分辨率是 3.20ns。

返回格式

查询命令以实型形式返回触发释抑，单位 s。

实例说明

:TRIGger:HOLDoff 3.2e-9　　设置触发释抑为 3.20ns。

:TRIGger:HOLDoff? -> 3.2e-9　　查询当前触发释抑为 3.20ns。

:TRIGger:COUPling

命令格式

:TRIGger:COUPling <String>

:TRIGger:COUPling?

功能描述

设置触发耦合。

查询触发耦合。

参数说明

<String> 范围{DC | AC | LFReject | HFReject}，

分别表示直流、交流、低频抑制、高频抑制。

返回格式

查询命令返回 DC、AC、LFReject 或 HFReject。

实例说明

:TRIGger:COUPling DC	设置触发耦合为直流耦合。
:TRIGger:COUPling? -> DC	查询触发耦合为直流耦合。

:TRIGger:AUTOSENS

命令格式

:TRIGger:AUTOSENS <String>

:TRIGger:AUTOSENS?

功能描述

设置灵敏度模式。

查询灵敏度模式。

参数说明

<String> 范围{0 | 1}，

分别表示手动和自动。

返回格式

查询命令返回 0 或 1。

实例说明

:TRIGger:AUTOSENS 0	设置灵敏度模式为手动。
:TRIGger:AUTOSENS? -> 0	查询灵敏度模式为手动。

:TRIGger:SENSitivity

命令格式

:TRIGger:SENSitivity <Double>

:TRIGger:SENSitivity? -> <Double>

功能描述

设置触发灵敏度。

查询触发灵敏度。

注：灵敏度模式设置为**手动**时有效，相关指令:TRIGger:MODE。

参数说明

<Double> 范围 0.0~1.5，如 0.3 代表 0.3div 的灵敏度。

返回格式

查询命令以实型形式返回触发灵敏度，单位 div。

实例说明

:TRIGger:SENSitivity 0.3	设置触发灵敏度为 0.3div。
:TRIGger:SENSitivity? -> 0.3	查询当前触发灵敏度为 0.3div。

:TRIGger:MODE

命令格式

:TRIGger:MODE <String>

:TRIGger:MODE?

功能描述

设置触发类型。

查询触发类型。

参数说明

<String> 范围{EDGE | PULSe | SLOPe | VIDEo | RUNT | PRUNt | PATTern | NEDGe | TIMeout | SHOLd | UART | IIC | IICDEVICE | SPI | MODBUS | MIPIDSI | MIPIRFFE | CAN | CANFD | LIN | FLEXRAY | SENT | MVB | WTB | 1553B | IS07816 | USB | SDSPI | SDSD | WIEGAND | DS18B20 | PS2 | MDIO | DALI | HDQ | ONEWIRE | DIFFMANCHESTER | MILLER | DHT11 | SHT11 | DMX512}，

分别表示边沿触发、脉宽触发、斜率触发、视频触发、欠幅触发、超幅触发、码型触发、第 N 边沿触发、超时触发、建立保持、UART、I2C、I2C device、SPI、Modbus、MIPI_DSI、MIPI_RFFE、CAN、CAN FD、LIN、FlexRay、SENT、MVB、WTB、1553B、IS07816、USB、SD_SPI、SD_SD、Wiegand、DS18B20、PS/2、MDIO、DALI、HDQ、1-Wire、Diff-Mancheter、Miller、DHT11、SHT11、DMX512。

注：ZUS5054 示波器标配基础触发和以下协议触发类型：CAN、USB、UART、I²C、I²C device、SPI、I²S、ModBus、1553B、PS/2，其余协议触发类型可选配。

返回格式

查询命令返回 EDGE、PULSe、SLOPe、VIDEo、RUNT、PRUNt、PATTern、NEDGe、TIMeout 或 SHOLd 等。

实例说明

:TRIGger:MODE EDGE	设置触发类型为边沿触发。
:TRIGger:MODE? -> EDGE	查询触发类型为边沿触发

:TRIGger:LEVEl:CHANnel<x>

命令格式

:TRIGger:LEVEl:CHANnel<x> <Double>

:TRIGger:LEVEl:CHANnel<x>?

功能描述

设置垂直通道的触发阈值。

查询垂直通道的触发阈值。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<Double> 阈值，单位 V 或 A。

返回格式

查询命令以实型形式返回垂直通道的触发阈值，单位 V 或 A。

实例说明

:TRIGger:LEVEl:CHANnel1 -0.3	设置垂直通道 1 的触发阈值为-300mV。
:TRIGger:LEVEl:CHANnel1? -> -0.3	查询当前垂直通道 1 的触发阈值为-300mV。

:TRIGger:HLEVel:CHANnel<x>

命令格式

:TRIGger:HLEVel:CHANnel<x> <Double>

:TRIGger:HLEVel:CHANnel<x>?

功能描述

设置 TH 触发高阈值。

查询 TH 触发高阈值。

注：触发类型设置为**斜率触发**、**欠幅触发**和**超幅触发**时有效，相关指令:TRIGger:MODE。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<Double> 阈值，单位 V 或 A。

返回格式

查询命令以实型形式返回 TH 触发高阈值，单位 V 或 A。

实例说明

:TRIGger:HLEVEl:CHANnel1 -0.3　　设置垂直通道 1 的 TH 触发高阈值为 -300mV。

:TRIGger:HLEVEl:CHANnel1? -> -0.3　　查询当前垂直通道 1 的 TH 触发高阈值为 -300mV。

:TRIGger:LLEVel:CHANnel<x>

命令格式

:TRIGger:LLEVel:CHANnel<x> <Double>

:TRIGger:LLEVel:CHANnel<x>?

功能描述

设置 TL 触发低阈值。

查询 TL 触发低阈值。

注：触发类型设置为**斜率触发**、**欠幅触发**和**超幅触发**时有效，相关指令:TRIGger:MODE。

参数说明

<x> 需要查询或改变的垂直通道编号，1、2、3 或 4。

<Double> 阈值，单位 V 或 A。

返回格式

查询命令以实型形式返回 TL 触发低阈值，单位 V 或 A。

实例说明

:TRIGger:LLEVEl:CHANnel1 -0.3　　设置垂直通道 1 的 TL 触发低阈值为 -300mV。

:TRIGger:LLEVEl:CHANnel1? -> -0.3　　查询当前垂直通道 1 的 TL 触发低阈值为 -300mV。

:TRIGger:LEVEl:EXT

命令格式

:TRIGger:LEVEl:EXT <Double>

:TRIGger:LEVEl:EXT?

功能描述

设置外触发阈值。

查询外触发阈值。

参数说明

<Double> 阈值，单位 V。

返回格式

查询命令以实型形式返回外触发阈值，单位 V。

实例说明

:TRIGger:LEVEl:EXT -0.3	设置外触发阈值为-300mV。
:TRIGger:LEVEl:EXT? -> -0.3	查询外触发阈值为-300mV。

:TRIGger:EDGe:SOURce

命令格式

:TRIGger:EDGe:SOURce <String>

:TRIGger:EDGe:SOURce?

功能描述

设置边沿触发的信源选择。

查询边沿触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | AC | EXT}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、AC 或 EXT。

实例说明

:TRIGger:EDGe:SOURce CHANnel1 设置边沿触发的信源选择为通道 1。

:TRIGger:EDGe:SOURce? -> CHANnel1 查询当前边沿触发的信源选择为通道 1。

:TRIGger:EDGe:TYPE

命令格式

:TRIGger:EDGe:TYPE <String>

:TRIGger:EDGe:TYPE?

功能描述

设置边沿触发的边沿类型。

查询边沿触发的边沿类型。

参数说明

<String> 范围 {POSitive | NEGative | EITHer}，分别表示上升沿触发，下降沿触发，上升沿下降沿均触发。

返回格式

查询命令返回 POSitive、NEGative 或 EITHer。

实例说明

:TRIGger:EDGe:TYPE POSitive 设置边沿触发的边沿类型为上升沿触发。

:TRIGger:EDGe:TYPE? -> POSitive 查询当前边沿触发的边沿类型为上升沿触发。

:TRIGger:PULSe:SOURce

命令格式

:TRIGger:PULSe:SOURce <String>

:TRIGger:PULSe:SOURce?

功能描述

设置脉宽触发的信源选择。

查询脉宽触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:PULSe:SOURce CHANnel1 设置脉宽触发的信源选择为通道 1。

:TRIGger:PULSe:SOURce? -> CHANnel1 查询当前脉宽触发的信源选择为通道 1。

:TRIGger:PULSe:WHEN

命令格式

:TRIGger:PULSe:WHEN <String>

:TRIGger:PULSe:WHEN?

功能描述

设置脉宽触发的脉宽类型。

查询脉宽触发的脉宽类型。

参数说明

<String> 范围{PGReater | PLESs | PGLess | NGReater | NLESs | NGLess}，

分别表示正脉冲宽度大于脉宽下限时触发，正脉冲宽度小于脉宽上限时触发，正脉冲宽度在指定范围时触发，负脉冲宽度大于脉宽下限时触发，负脉冲宽度小于脉宽上限时触发，负脉冲在指定范围时触发。

返回格式

查询命令返回 PGReater、PLESs、PGLess、NGReater、NLESs 或 NGLess。

实例说明

:TRIGger:PULSe:WHEN PGReater 设置脉宽触发的脉宽类型为正脉冲宽度大于脉宽下限时触发。

:TRIGger:PULSe:WHEN? -> PGReater 查询当前脉宽触发的脉宽类型为正脉冲宽度大于脉宽下限时触发。

:TRIGger:PULSe:LOWer

命令格式

:TRIGger:PULSe:LOWer <String>

:TRIGger:PULSe:LOWer?

功能描述

设置脉宽触发的脉宽下限。

查询脉宽触发的脉宽下限。

参数说明

<String> 范围 1ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回脉宽触发的脉宽下限。

实例说明

:TRIGger:PULSe:LOWer 100ns	设置脉宽触发的脉宽下限为 100ns。
:TRIGger:PULSe:LOWer? -> 100ns	查询当前脉宽触发的脉宽下限为 100ns。

:TRIGger:PULSe:UPPer

命令格式

:TRIGger:PULSe:UPPer <String>

:TRIGger:PULSe:UPPer?

功能描述

设置脉宽触发的脉宽上限。

查询脉宽触发的脉宽上限。

参数说明

<String> 范围 1ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回脉宽触发的脉宽上限。

实例说明

:TRIGger:PULSe:UPPer 100ns	设置脉宽触发的脉宽上限为 100ns。
:TRIGger:PULSe:UPPer? -> 100ns	查询当前脉宽触发的脉宽上限为 100ns。

:TRIGger:SLOPe:SOURce

命令格式

:TRIGger:SLOPe:SOURce <String>

:TRIGger:SLOPe:SOURce?

功能描述

设置斜率触发的信源选择。

查询斜率触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SLOPe:SOURce CHANnel1 设置斜率触发的信源选择为通道 1。

:TRIGger:SLOPe:SOURce? -> CHANnel1 查询当前斜率触发的信源选择为通道 1。

:TRIGger:SLOPe:WHEN

命令格式

:TRIGger:SLOPe:WHEN <String>

:TRIGger:SLOPe:WHEN?

功能描述

设置斜率触发的触发模式。

查询斜率触发的触发模式。

参数说明

<String> 范围{ PGReater | PLESs | PGLess | NGReater | NLESs | NGLess }，

分别表示正斜率大于时间下限时触发，正斜率小于时间上限时触发，正斜率在指定范围时触发，负斜率大于时间下限时触发，负斜率小于时间上限时触发，负斜率在指定范围时触发。

返回格式

查询命令返回 PGReater、PLESs、PGLess、NGReater、NLESs 或 NGLess。

实例说明

:TRIGger:SLOPe:WHEN PGReater　　设置斜率触发的触发模式为正斜率大于时间下限时触发。

:TRIGger:SLOPe:WHEN? -> PGReater　　查询当前斜率触发的触发模式为正斜率大于时间下限时触发。

:TRIGger:SLOPe:LOWer

命令格式

:TRIGger:SLOPe:LOWer <String>

:TRIGger:SLOPe:LOWer?

功能描述

设置斜率触发的时间下限。

查询斜率触发的时间下限。

参数说明

<String> 范围 1ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回斜率触发的时间下限。

实例说明

:TRIGger:SLOPe:LOWer 10ns	设置斜率触发的时间下限为 10ns。
:TRIGger:SLOPe:LOWer? -> 10ns	查询当前斜率触发的时间下限为 10ns。

:TRIGger:SLOPe:UPPer

命令格式

:TRIGger:SLOPe:UPPer <String>

:TRIGger:SLOPe:UPPer?

功能描述

设置斜率触发的时间上限。

查询斜率触发的时间上限。

参数说明

<String> 范围 1ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回斜率触发的时间上限。

实例说明

:TRIGger:SLOPe:UPPer 10ns	设置斜率触发的时间上限为 10ns。
:TRIGger:SLOPe:UPPer? -> 10ns	查询斜率触发的时间上限为 10ns。

:TRIGger:VIDEo:SOURce

命令格式

:TRIGger:VIDEo:SOURce <String>

:TRIGger:VIDEo:SOURce?

功能描述

设置视频触发的信源选择。

查询视频触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:VIDEo:SOURce CHANnel1　　设置视频触发的信源选择为通道 1。

:TRIGger:VIDEo:SOURce? -> CHANnel1　　查询当前视频触发的信源选择为通道 1。

:TRIGger:VIDEo:POLArity

命令格式

:TRIGger:VIDEo:POLArity <String>

:TRIGger:VIDEo:POLArity?

功能描述

设置视频触发的极性。

查询视频触发的极性。

参数说明

<String> 范围 {POSitive | NEGative}，分别表示正极性，负极性。

返回格式

查询命令返回 POSitive 或 NEGative。

实例说明

:TRIGger:VIDEo:POLArity POSitive	设置视频触发的极性为正极性。
:TRIGger:VIDEo:POLArity? -> POSitive	查询当前视频触发的极性为正极性。

:TRIGger:VIDEo:STANdard

命令格式

:TRIGger:VIDEo:STANdard <String>

:TRIGger:VIDEo:STANdard?

功能描述

设置视频触发的视频格式。

查询视频触发的视频格式。

参数说明

<String> 范围 {NTSC | PAL | SECAM}。

返回格式

查询命令返回 NTSC、PAL 或 SECAM。

实例说明

:TRIGger:VIDEo:STANdard NTSC 设置视频触发的视频格式为 NTSC。

:TRIGger:VIDEo:STANdard? -> NTSC 查询当前视频触发的视频格式为 NTSC。

:TRIGger:VIDEo:MODE

命令格式

:TRIGger:VIDEo:MODE <String>

:TRIGger:VIDEo:MODE?

功能描述

设置视频触发的触发模式。

查询视频触发的触发模式。

参数说明

<String> 范围 {ANYLine | GOTOline | ANYFiled | EVENfield | ODDField}，分别表示任意行、指定行、任意场、偶数场、奇数场。

返回格式

查询命令返回 ANYLine、GOTOline、ANYFiled、EVENfield 或 ODDField。

实例说明

:TRIGger:VIDEo:MODE ANYFiled	设置视频触发的触发模式为任意场。
:TRIGger:VIDEo:MODE? -> ANYFiled	查询当前视频触发的触发模式为任意场。

:TRIGger:VIDEo:LINE

命令格式

:TRIGger:VIDEo:LINE <Uint>

:TRIGger:VIDEo:LINE?

功能描述

设置视频触发的指定触发行。

查询视频触发的指定触发行。

参数说明

<Uint> 范围 1~625。

返回格式

查询命令以整数形式返回视频触发的指定触发行。

实例说明

:TRIGger:VIDEo:LINE 1	设置视频触发的指定触发行为第一行。
:TRIGger:VIDEo:LINE? -> 1	查询当前视频触发的指定触发行为第一行。

:TRIGger:RUNT:SOURce

命令格式

:TRIGger:RUNT:SOURce <String>

:TRIGger:RUNT:SOURce?

功能描述

设置欠幅触发的信源选择。

查询欠幅触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:RUNT:SOURce CHANnel1　　设置欠幅触发的信源选择为通道 1。

:TRIGger:RUNT:SOURce? -> CHANnel1　　查询欠幅视频触发的信源选择为通道 1。

:TRIGger:RUNT:TYPE

命令格式

:TRIGger:RUNT:TYPE <String>

:TRIGger:RUNT:TYPE?

功能描述

设置欠幅触发的脉冲类型。

查询欠幅触发的脉冲类型。

参数说明

<String> 范围 {POSitive | NEGative | EITHer}

分别表示正向欠幅脉冲触发、负向欠幅脉冲触发、正向和负向欠幅脉冲均可触发。

返回格式

查询命令返回 POSitive、NEGative 或 EITHer。

实例说明

:TRIGger:RUNT:TYPE POSitive 设置欠幅触发的脉冲类型为正向欠幅脉冲触发。

:TRIGger:RUNT:TYPE? -> POSitive 查询当前欠幅触发的脉冲类型为正向欠幅脉冲触发。

:TRIGger:RUNT:WHEN

命令格式

:TRIGger:RUNT:WHEN <String>

:TRIGger:RUNT:WHEN?

功能描述

设置欠幅触发的限定符。

查询欠幅触发的限定符。

参数说明

<String> 范围 {NONE | GREater | LESS | INRange}，

分别表示不限、大于、小于、区间以内。

返回格式

查询命令返回 NONE、GREater、LESS 或 INRange。

实例说明

:TRIGger:RUNT:WHEN GREater	设置欠幅触发的限定符为大于。
:TRIGger:RUNT:WHEN? -> GREater	查询当前欠幅触发的限定符为大于。

:TRIGger:RUNT:LOWer

命令格式

:TRIGger:RUNT:LOWer <String>

:TRIGger:RUNT:LOWer?

功能描述

设置欠幅触发的脉宽下限。

查询欠幅触发的脉宽下限。

参数说明

<String> 范围 2ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回欠幅触发的脉宽下限。

实例说明

:TRIGger:RUNT:LOWer 200ns	设置欠幅触发的脉宽下限为 200ns。
:TRIGger:RUNT:LOWer? -> 200ns	查询当前欠幅触发的脉宽下限为 200ns。

:TRIGger:RUNT:UPPer

命令格式

:TRIGger:RUNT:UPPer <String>

:TRIGger:RUNT:UPPer?

功能描述

设置欠幅触发的脉宽上限。

查询欠幅触发的脉宽上限。

参数说明

<String> 范围 2ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回欠幅触发的脉宽上限。

实例说明

:TRIGger:RUNT:UPPer 200ns	设置欠幅触发的脉宽上限为 200ns。
:TRIGger:RUNT:UPPer? -> 200ns	查询当前欠幅触发的脉宽上限为 200ns。

:TRIGger:PRUNt:SOURce

命令格式

:TRIGger:PRUNt:SOURce <String>

:TRIGger:PRUNt:SOURce?

功能描述

设置超幅触发的信源选择。

查询超幅触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:PRUNt:SOURce CHANnel1　　设置超幅触发的信源选择为通道 1。

:TRIGger:PRUNt:SOURce? -> CHANnel1　　查询超幅触发的信源选择为通道 1。

:TRIGger:PRUNt:TYPE

命令格式

:TRIGger:PRUNt:TYPE <String>

:TRIGger:PRUNt:TYPE?

功能描述

设置超幅触发的脉冲类型。

查询超幅触发的脉冲类型。

参数说明

<String> 范围 {POSitive | NEGative | EITHer}，

分别表示正向超幅脉冲触发、负向超幅脉冲触发、正向和负向超幅脉冲均可触发。

返回格式

查询命令返回 POSitive、NEGative 或 EITHer。

实例说明

:TRIGger:PRUNt:TYPE POSitive 设置超幅触发的脉冲类型为正向超幅脉冲触发。

:TRIGger:PRUNt:TYPE? -> POSitive 查询当前超幅触发的脉冲类型为正向超幅脉冲触发。

:TRIGger:PRUNt:WHEN

命令格式

:TRIGger:PRUNt:WHEN <String>

:TRIGger:PRUNt:WHEN?

功能描述

设置超幅触发的限定符。

查询超幅触发的限定符。

参数说明

<String>

范围 {NONE | GREater | LESS | INRange}，

分别表示不限、大于、小于、区间以内。

返回格式

查询命令返回 NONE、GREater、LESS 或 INRange。

实例说明

:TRIGger:PRUNt:WHEN GREater	设置超幅触发的限定符为大于。
:TRIGger:PRUNt:WHEN? -> GREater	查询当前超幅触发的限定符为大于。

:TRIGger:PRUNt:LOWer

命令格式

:TRIGger:PRUNt:LOWer <String>

:TRIGger:PRUNt:LOWer?

功能描述

设置超幅触发的脉宽下限。

查询超幅触发的脉宽下限。

参数说明

<String> 范围 4ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回超幅触发的脉宽下限。

实例说明

:TRIGger:PRUNt:LOWer 200ns 设置超幅触发的脉宽下限为 200ns。

:TRIGger:PRUNt:LOWer? -> 200ns 查询当前超幅触发的脉宽下限为 200ns。

:TRIGger:PRUNt:UPPer

命令格式

:TRIGger:PRUNt:UPPer <String>

:TRIGger:PRUNt:UPPer?

功能描述

设置超幅触发的脉宽上限。

查询超幅触发的脉宽上限。

参数说明

<String> 范围 4ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回超幅触发的脉宽上限。

实例说明

:TRIGger:PRUNt:UPPer 200ns	设置超幅触发的脉宽上限为 200ns。
:TRIGger:PRUNt:UPPer? -> 200ns	查询当前超幅触发的脉宽上限为 200ns。

:TRIGger:PATTern:ASRc

命令格式

:TRIGger:PATTern:ASRc <String>

:TRIGger:PATTern:ASRc?

功能描述

设置码型触发的 A 信源。

查询码型触发的 A 信源。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:PATTern:ASRc CHANnel1　　设置码型触发的 A 信源为通道 1。

:TRIGger:PATTern:ASRc? -> CHANnel1　　查询码型触发的 A 信源为通道 1。

:TRIGger:PATTern:BSRc

命令格式

:TRIGger:PATTern:BSRc <String>

:TRIGger:PATTern:BSRc?

功能描述

设置码型触发的 B 信源。

查询码型触发的 B 信源。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:PATTern:BSRc CHANnel1	设置码型触发的 B 信源为通道 1。
:TRIGger:PATTern:BSRc? -> CHANnel1	查询码型触发的 B 信源为通道 1。

:TRIGger:PATTern:APat

命令格式

:TRIGger:PATTern:APat <String>

:TRIGger:PATTern:APat?

功能描述

设置码型触发的 A 码型。

查询码型触发的 A 码型。

参数说明

<String> 范围 {H | L | X | R | F}，

分别表示高电平，低电平，忽略，上升沿，下降沿。

返回格式

查询命令返回 H、L、X、R 或 F。

实例说明

:TRIGger:PATTern:APat H	设置码型触发的 A 码型为高电平。
:TRIGger:PATTern:APat? -> H	查询码型触发的 A 码型为高电平。

:TRIGger:PATTern:BPat

命令格式

:TRIGger:PATTern:BPat <String>

:TRIGger:PATTern:BPat?

功能描述

设置码型触发的 B 码型。

查询码型触发的 B 码型。

参数说明

<String> 范围 {H | L | X | R | F}，

分别表示高电平，低电平，忽略，上升沿，下降沿。

返回格式

查询命令返回 H、L、X、R 或 F。

实例说明

:TRIGger:PATTern:BPat H	设置码型触发的 B 码型为高电平。
:TRIGger:PATTern:BPat? -> H	查询码型触发的 B 码型为高电平。

:TRIGger:PATTern:WHEN

命令格式

:TRIGger:PATTern:WHEN <String>

:TRIGger:PATTern:WHEN?

功能描述

设置码型触发的限定符。

查询码型触发的限定符。

参数说明

<String> 范围 {NONE | GREater | LESS | INRange | OUTRange}，

分别表示不限、大于、小于、区间以内、区间以外。

返回格式

查询命令返回 NONE、GREater、LESS、INRange 或 OUTRange。

实例说明

:TRIGger:PATTern:WHEN GREater	设置码型触发的限定符为大于。
:TRIGger:PATTern:WHEN? -> GREater	查询码型触发的限定符为大于。

:TRIGger:PATTern:LOWer

命令格式

:TRIGger:PATTern:LOWer <String>

:TRIGger:PATTern:LOWer?

功能描述

设置码型触发的时间下限。

查询码型触发的时间下限。

参数说明

<String> 范围 4ns~1.7s，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回码型触发的时间下限。

实例说明

:TRIGger:PATTern:LOWer 10.0ns	设置码型触发的时间下限为 10.0ns。
:TRIGger:PATTern:LOWer? -> 10.0ns	查询码型触发的时间下限为 10.0ns。

:TRIGger:PATTern:UPPer

命令格式

:TRIGger:PATTern:UPPer <String>

:TRIGger:PATTern:UPPer?

功能描述

设置码型触发的时间上限。

查询码型触发的时间上限。

参数说明

<String> 范围 4ns~1.7s，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回码型触发的时间上限。

实例说明

:TRIGger:PATTern:UPPer 10.0ns	设置码型触发的时间上限为 10.0ns。
:TRIGger:PATTern:UPPer? -> 10.0ns	查询码型触发的时间上限为 10.0ns。

:TRIGger:NEDGe:SOURce

命令格式

:TRIGger:NEDGe:SOURce <String>

:TRIGger:NEDGe:SOURce?

功能描述

设置第 N 边沿触发的信源选择。

查询第 N 边沿触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:NEDGe:SOURce CHANnel1　　设置第 N 边沿触发的信源选择为通道 1。

:TRIGger:NEDGe:SOURce? -> CHANnel1　　查询第 N 边沿触发的信源选择为通道 1。

:TRIGger:NEDGe:TYPE

命令格式

:TRIGger:NEDGe:TYPE <String>

:TRIGger:NEDGe:TYPE?

功能描述

设置第 N 边沿触发的边沿类型。

查询第 N 边沿触发的边沿类型。

参数说明

<String> 范围 {POSitive | NEGative}

分别表示上升沿触发，下降沿触发。

返回格式

查询命令返回 POSitive 或 NEGative。

实例说明

:TRIGger:NEDGe:TYPE POSitive 设置第 N 边沿触发的边沿类型为上升沿触发。

:TRIGger:NEDGe:TYPE? -> POSitive 查询当前第 N 边沿触发的边沿类型为上升沿触发。

:TRIGger:NEDGe:NUM

命令格式

:TRIGger:NEDGe:NUM <Int>

:TRIGger:NEDGe:NUM?

功能描述

设置第 N 边沿触发的边沿数。

查询第 N 边沿触发的边沿数。

参数说明

<Int> 范围 1~65535，边沿数。

返回格式

查询命令以整数形式返回第 N 边沿触发的边沿数。

实例说明

:TRIGger:NEDGe:NUM 1	设置第 N 边沿触发的边沿数为 1。
:TRIGger:NEDGe:NUM? -> 1	查询第 N 边沿触发的边沿数为 1。

:TRIGger:NEDGe:IDLE

命令格式

:TRIGger:NEDGe:IDLE <String>

:TRIGger:NEDGe:IDLE?

功能描述

设置第 N 边沿触发的空闲时间。

查询第 N 边沿触发的空闲时间。

参数说明

<String> 范围 10ns~1.7s，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回第 N 边沿触发的空闲时间。

实例说明

:TRIGger:NEDGe:IDLE 2e-4	设置第 N 边沿触发的空闲时间为 200.0us。
:TRIGger:NEDGe:IDLE? -> 200.0us	查询第 N 边沿触发的空闲时间为 200.0us。

:TRIGger:TIMeout:SOURce

命令格式

:TRIGger:TIMeout:SOURce <String>

:TRIGger:TIMeout:SOURce?

功能描述

设置超时触发的信源选择。

查询超时触发的信源选择。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:TIMeout:SOURce CHANnel1	设置超时触发的信源选择为通道 1。
:TRIGger:TIMeout:SOURce? -> CHANnel1	查询超时触发的信源选择为通道 1。

:TRIGger:TIMeout:TYPE

命令格式

:TRIGger:TIMeout:TYPE <String>

:TRIGger:TIMeout:TYPE?

功能描述

设置超时触发的触发模式。

查询超时触发的触发模式。

参数说明

<String>范围 {POSitive | NEGative | EITHer}，

分别表示高电平超时，低电平超时，双沿超时。

返回格式

查询命令返回 POSitive、NEGative 或 EITHer。

实例说明

:TRIGger:TIMeout:TYPE POSitive 设置超时触发的触发模式为高电平超时。

:TRIGger:TIMeout:TYPE? -> NEGative 查询当前超时触发的触发模式为高电平超时。

:TRIGger:TIMeout:TIMe

命令格式

:TRIGger:TIMeout:TIMe <String>

:TRIGger:TIMeout:TIMe?

功能描述

设置超时触发的超时时间。

查询超时触发的超时时间。

参数说明

<String> 范围 8ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回超时触发的超时时间。

实例说明

:TRIGger:TIMeout:TIMe 125ns	设置超时触发的超时时间为 125ns。
:TRIGger:TIMeout:TIMe? -> 125ns	查询超时触发的超时时间为 125ns。

:TRIGger:SHOLd:CSrc

命令格式

:TRIGger:SHOLd:CSrc <String>

:TRIGger:SHOLd:CSrc?

功能描述

设置建立保持触发的时钟通道。

查询建立保持触发的时钟通道。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SHOLd:CSrc CHANnel1　　设置建立保持触发的时钟通道为通道 1。

:TRIGger:SHOLd:CSrc? -> CHANnel1　　查询建立保持触发的时钟通道为通道 1。

:TRIGger:SHOLd:DSrc

命令格式

:TRIGger:SHOLd:DSrc <String>

:TRIGger:SHOLd:DSrc?

功能描述

设置建立保持触发的数据通道。

查询建立保持触发的数据通道。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SHOLd:DSrc CHANnel1　　设置建立保持触发的数据通道为通道 1。

:TRIGger:SHOLd:DSrc? -> CHANnel1　　查询建立保持触发的数据通道为通道 1。

:TRIGger:SHOLd:TYPE

命令格式

:TRIGger:SHOLd:TYPE <String>

:TRIGger:SHOLd:TYPE?

功能描述

设置建立保持触发的采样类型。

查询建立保持触发的采样类型。

参数说明

<String>范围 {POSitive | NEGative}，

分别表示上升沿，下降沿。

返回格式

查询命令返回 POSitive 或 NEGative。

实例说明

:TRIGger:SHOLd:TYPE POSitive	设置建立保持触发的采样类型为上升沿。
:TRIGger:SHOLd:TYPE? -> POSitive	查询建立保持触发的采样类型为上升沿。

:TRIGger:SHOLd:PATTern

命令格式

:TRIGger:SHOLd:PATTern <String>

:TRIGger:SHOLd:PATTern?

功能描述

设置建立保持触发的数据类型。

查询建立保持触发的数据类型。

参数说明

<String>范围 {L | H}，分别表示低电平，高电平。

返回格式

查询命令返回 L 或 H。

实例说明

:TRIGger:SHOLd:PATTern L	设置建立保持触发的数据类型低电平。
:TRIGger:SHOLd:PATTern? -> L	查询建立保持触发的数据类型低电平。

:TRIGger:SHOLd:MODE

命令格式

:TRIGger:SHOLd:MODE <String>

:TRIGger:SHOLd:MODE?

功能描述

设置建立保持触发的触发类型。

查询建立保持触发的触发类型。

参数说明

<String> 范围 {SETup | HOLd}，

分别表示建立触发，保持触发。

返回格式

查询命令返回 SETup 或 HOLd。

实例说明

:TRIGger:SHOLd:MODE SETup	设置建立保持触发的触发类型为建立触发。
:TRIGger:SHOLd:MODE? -> SETup	查询建立保持触发的触发类型为建立触发。

:TRIGger:SHOLd:STIMe

命令格式

:TRIGger:SHOLd:STIMe <String>

:TRIGger:SHOLd:STIMe?

功能描述

设置建立保持触发的建立时间。

查询建立保持触发的建立时间。

参数说明

<String> 范围 2ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回建立保持触发的建立时间。

实例说明

:TRIGger:SHOLd:STIMe 10.0ns	设置建立保持触发的建立时间为 10.0ns。
:TRIGger:SHOLd:STIMe? -> 10.0ns	查询建立保持触发的建立时间为 10.0ns。

:TRIGger:SHOLd:HTIMe

命令格式

:TRIGger:SHOLd:HTIMe <String>

:TRIGger:SHOLd:HTIMe?

功能描述

设置建立保持触发的保持时间。

查询建立保持触发的保持时间。

参数说明

<String> 范围 2ns~800ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回建立保持触发的保持时间。

实例说明

:TRIGger:SHOLd:HTIMe 10.0ns	设置建立保持触发的保持时间为 10.0ns。
:TRIGger:SHOLd:HTIMe? -> 10.0ns	查询建立保持触发的保持时间为 10.0ns。

:TRIGger:1553B:SOURce

命令格式

:TRIGger:1553B:SOURce <String>

:TRIGger:1553B:SOURce?

功能描述

设置 1553B 协议触发的信源。

查询 1553B 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:1553B:SOURce CHANnel1　　设置 1553B 协议触发信源为通道 1。

:TRIGger:1553B:SOURce? -> CHANnel1　　查询 1553B 协议触发信源为通道 1。

:TRIGger:1553B:TRIGMode

命令格式

:TRIGger:1553B:TRIGMode <String>

:TRIGger:1553B:TRIGMode?

功能描述

设置 1553B 协议触发的触发模式。

查询 1553B 协议触发的触发模式。

参数说明

<String> 范围 {STARt | RTADdr | SUBAddr | CODE }，分别表示起始触发、RT 地址触发、子地址触发和数据数/方式码触发。

返回格式

查询命令返回 STARt、RTADdr、SUBAddr 或 CODE。

实例说明

:TRIGger:1553B:TRIGMode STARt　　设置 1553B 协议触发的触发模式为起始触发。

:TRIGger:1553B:TRIGMode? -> STARt　　查询 1553B 协议触发的触发模式为起始触发。

:TRIGger:1553B:RTADdr

命令格式

:TRIGger:1553B:RTADdr <Int>

:TRIGger:1553B:RTADdr?

功能描述

设置 1553B 协议触发的远程终端地址。

查询 1553B 协议触发的远程终端地址。

注：触发模式设置为 **RT 地址触发**、**子地址触发**和**数据数/方式码**时有效，相关指令:TRIGger:1553B:TRIGMode。

参数说明

<Int> 范围 0x00~0x1F。

返回格式

查询命令以十六进制形式返回 1553B 协议触发的远程终端地址。

实例说明

:TRIGger:1553B:RTADdr 0x10	设置 1553B 协议触发的远程终端地址为 0x10。
:TRIGger:1553B:RTADdr? -> 0x10	查询 1553B 协议触发的远程终端地址为 0x10。

:TRIGger:1553B:TRMOde

命令格式

:TRIGger:1553B:TRMOde <String>

:TRIGger:1553B:TRMOde?

功能描述

设置 1553B 协议触发的收发模式。

查询 1553B 协议触发的收发模式。

注：触发模式设置为**子地址触发**和**数据数/方式码**时有效，相关指令:TRIGger:1553B:TRIGMode。

参数说明

<String> 范围 {RECEive | TRANsmit }，分别表示接收、发送。

返回格式

查询命令返回 RECEive 或 TRANsmit。

实例说明

:TRIGger:1553B:TRMOde TRANsmit　　设置 1553B 协议触发的收发模式为发送。

:TRIGger:1553B:TRMOde? -> TRANsmit　　查询 1553B 协议触发的收发模式为发送。

:TRIGger:1553B:FIELd

命令格式

:TRIGger:1553B:FIELd <Int>

:TRIGger:1553B:FIELd?

功能描述

设置 1553B 协议触发的子地址/方式域。

查询 1553B 协议触发的子地址/方式域。

注：触发模式设置为**子地址触发**和**数据数/方式码**时有效，相关指令:TRIGger:1553B:TRIGMode。

参数说明

<Int> 范围 0x00~0x1F。

返回格式

查询命令以十六进制形式返回 1553B 协议触发的子地址/方式域。

实例说明

:TRIGger:1553B:FIELd 0x10	设置 1553B 协议触发的子地址/方式域为 0x10。
:TRIGger:1553B:FIELd? -> 0x10	查询 1553B 协议触发的子地址/方式域为 0x10。

:TRIGger:1553B:CODE

命令格式

:TRIGger:1553B:CODE <Int>

:TRIGger:1553B:CODE?

功能描述

设置 1553B 协议触发的数据数/方式码。

查询 1553B 协议触发的数据数/方式码。

注：触发模式设置为**数据数/方式码触发**有效，相关指令:TRIGger:1553B:TRIGMode。

参数说明

<Int> 范围 0x00~0x1F。

返回格式

查询命令以十六进制形式返回 1553B 协议触发的数据数/方式码。

实例说明

:TRIGger:1553B:CODE 0x10	设置 1553B 协议触发的数据数/方式码为 0x10。
:TRIGger:1553B:CODE? -> 0x10	查询 1553B 协议触发的数据数/方式码为 0x10。

:TRIGger:1WIRe:SOURce

命令格式

:TRIGger:1WIRe:SOURce <String>

:TRIGger:1WIRe:SOURce?

功能描述

设置 1-Wire 协议触发的信源。

查询 1-Wire 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:1WIRe:SOURce CHANnel1	设置 1-Wire 协议触发信源为通道 1。
:TRIGger:1WIRe:SOURce? -> CHANnel1	查询 1-Wire 协议触发信源为通道 1。

:TRIGger:1WIRe:SPEEdmode

命令格式

:TRIGger:1WIRe:SPEEdmode <String>

:TRIGger:1WIRe:SPEEdmode?

功能描述

设置 1-Wire 协议触发的速度模式。

查询 1-Wire 协议触发的速度模式。

参数说明

<String> 范围 {STANdard | DRIVer }，分别表示标准、驱动。

返回格式

查询命令返回 STANdard 或 DRIVer。

实例说明

:TRIGger:1WIRe:SPEEdmode STANdard 设置 1-Wire 协议触发的速度模式为标准。

:TRIGger:1WIRe:SPEEdmode? -> STANdard 查询 1-Wire 协议触发的速度模式为标准。

:TRIGger:1WIRe:TRIGMode

命令格式

:TRIGger:1WIRe:TRIGMode <String>

:TRIGger:1WIRe:TRIGMode?

功能描述

设置 1-Wire 协议触发的触发模式。

查询 1-Wire 协议触发的触发模式。

参数说明

<String> 范围 {STARtfield | COMMand | CUSTom }，分别表示开始段触发、指令触发和自定义触发。

返回格式

查询命令返回 STARtfield、COMMand 或 CUSTom。

实例说明

:TRIGger:1WIRe:TRIGMode STARtfield 设置 1-Wire 协议触发的触发模式为开始段触发。

:TRIGger:1WIRe:TRIGMode? -> STARtfield 查询 1-Wire 协议触发的触发模式为开始段触发。

:TRIGger:1WIRe:TRIGCmd

命令格式

:TRIGger:1WIRe:TRIGCmd <String>

:TRIGger:1WIRe:TRIGCmd?

功能描述

设置 1-Wire 协议触发的触发指令。

查询 1-Wire 协议触发的触发指令。

注：触发模式设置为**指令触发**时有效，相关指令:TRIGger:1WIRe:TRIGMode。

参数说明

<String> 范围 {READ | SEARch | MATCh | ALARm | SKIP }。

返回格式

查询命令返回 READ、SEARch、MATCh、ALARm 或 SKIP。

实例说明

:TRIGger:1WIRe:TRIGCmd READ　　设置 1-Wire 协议触发的触发指令为 READ。

:TRIGger:1WIRe:TRIGCmd? -> READ　　查询 1-Wire 协议触发的触发指令为 READ。

:TRIGger:1WIRe:COMMand<x>

命令格式

:TRIGger:1WIRe:COMMand<x> <Int>

:TRIGger:1WIRe:COMMand<x>?

功能描述

设置 1-Wire 协议触发的指令 1 或指令 2。

查询 1-Wire 协议触发的指令 1 或指令 2。

注：触发模式设置为**自定义触发**时有效，相关指令:TRIGger:1WIRe:TRIGMod。

参数说明

<x> 指令编号，范围 { 1 | 2 }。

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 1-Wire 协议触发的指令 1 或指令 2。

实例说明

:TRIGger:1WIRe:COMMand1 0x10　　设置 1-Wire 协议触发的指令 1 为 0x10。

:TRIGger:1WIRe:COMMand1? -> 0x10　　查询 1-Wire 协议触发的指令 1 为 0x10。

:TRIGger:1WIRe:SPACe

命令格式

:TRIGger:1WIRe:SPACe <Int>

:TRIGger:1WIRe:SPACe?

功能描述

设置 1-Wire 协议触发的间隔位宽。

查询 1-Wire 协议触发的间隔位宽。

注：触发模式设置为**自定义触发**时有效，相关指令:TRIGger:1WIRe:TRIGMode。

参数说明

<Int> 范围 0~255。

返回格式

查询命令以整数形式返回 1-Wire 协议触发的间隔位宽。

实例说明

:TRIGger:1WIRe:SPACe 1	设置 1-Wire 协议触发的间隔位宽为 1。
:TRIGger:1WIRe:SPACe? -> 1	查询 1-Wire 协议触发的间隔位宽为 1。

:TRIGger:CAN:SOURce

命令格式

:TRIGger:CAN:SOURce <String>

:TRIGger:CAN:SOURce?

功能描述

设置 CAN 协议触发的信源。

查询 CAN 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:CAN:SOURce CHANnel1	设置 CAN 协议触发信源为通道 1。
:TRIGger:CAN:SOURce? -> CHANnel1	查询 CAN 协议触发信源为通道 1。

:TRIGger:CAN:TRIGMode

命令格式

:TRIGger:CAN:TRIGMode <String>

:TRIGger:CAN:TRIGMode?

功能描述

设置 CAN 协议触发的触发模式。

查询 CAN 协议触发的触发模式。

参数说明

<String> 范围 {STARtbit | STDData | EXTData | STDRemote | EXTRemote | ERR }，分别表示开始位、标准数据帧、扩展数据帧、标准远程帧、扩展远程帧和错误/过载帧。

返回格式

查询命令返回 STARtbit、STDData、EXTData、STDRemote、EXTRemote 或 ERR。

实例说明

:TRIGger:CAN:TRIGMode STARtbit　　设置 CAN 协议触发的触发模式为开始位。

:TRIGger:CAN:TRIGMode? -> STARtbit　　查询 CAN 协议触发的触发模式为开始位。

:TRIGger:CAN:BUSType

命令格式

:TRIGger:CAN:BUSType <String>

:TRIGger:CAN:BUSType?

功能描述

设置 CAN 协议触发的总线类型。

查询 CAN 协议触发的总线类型。

参数说明

<String> 范围 { CANL | CANH | CANDiff }。

返回格式

查询命令返回 CANL、CANH 或 CANDiff。

实例说明

:TRIGger:CAN:BUSType CANL	设置 CAN 协议触发的总线类型为 CANL。
:TRIGger:CAN:BUSType? -> CANL	查询 CAN 协议触发的总线类型为 CANL。

:TRIGger:CAN:BAUDrate

命令格式

:TRIGger:CAN:BAUDrate <Double>

:TRIGger:CAN:BAUDrate?

功能描述

设置 CAN 协议触发的波特率。

查询 CAN 协议触发的波特率。

参数说明

<Double> 波特率，范围 1.00k~10000.00k。

返回格式

查询命令以实型形式返回 CAN 协议触发的波特率大小。

实例说明

:TRIGger:CAN:BAUDrate 500	设置 CAN 协议触发的波特率为 500k。
:TRIGger:CAN:BAUDrate? -> 500.00	查询 CAN 协议触发的波特率为 500k。

:TRIGger:CAN:ID

命令格式

:TRIGger:CAN:ID <Int>

:TRIGger:CAN:ID?

功能描述

设置 CAN 协议触发的 ID 值。

查询 CAN 协议触发的 ID 值。

注：触发模式设置为**标准数据帧**、**扩展数据帧**、**标准远程帧**和**扩展远程帧**时有效，相关指令:TRIGger:CAN:TRIGMode。

参数说明

<Int> 范围 0x00~0x1FFFFFFF。

注：ID 值范围根据触发模式变化，扩展数据帧和扩展远程帧最大是 0x1FFFFFFF，其余最大是 0x7FF。

返回格式

查询命令以十六进制形式返回 CAN 协议触发的 ID 值。

实例说明

:TRIGger:CAN:ID 0x10	设置 CAN 协议触发的 ID 值为 0x10。
:TRIGger:CAN:ID? -> 0x10	查询 CAN 协议触发的 ID 值为 0x10。

:TRIGger:CAN:DLC

命令格式

:TRIGger:CAN:DLC <String>

:TRIGger:CAN:DLC?

功能描述

设置 CAN 协议触发的 DLC 值。

查询 CAN 协议触发的 DLC 值。

注：触发模式设置为**标准数据帧**、**扩展数据帧**、**标准远程帧**和**扩展远程帧**时有效，相关指令:TRIGger:CAN:TRIGMode。

参数说明

<String> 范围 {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | NONE }。

返回格式

查询命令返回 1、2、3、4、5、6、7、8 或 NONE。

实例说明

:TRIGger:CAN:DLC 2　　设置 CAN 协议触发的 DLC 值为 2。

:TRIGger:CAN:DLC? -> 2　查询 CAN 协议触发的 DLC 值为 2。

:TRIGger:CAN:DATAIndex

命令格式

:TRIGger:CAN:DATAIndex <Int>

:TRIGger:CAN:DATAIndex?

功能描述

设置 CAN 协议触发的数据索引。

查询 CAN 协议触发的数据索引。

注 1：触发模式设置为**标准数据帧**和**扩展数据帧**时有效，相关指令:TRIGger:CAN:TRIGMode。

注 2：当 DLC 值大于 0 时有效，相关指令:TRIGger:CAN:DLC。

参数说明

<Int> 范围 0~7。

注：数据索引范围根据 DLC 值变化。

返回格式

查询命令以整数形式返回 CAN 协议触发的数据索引。

实例说明

:TRIGger:CAN:DATAIndex 1	设置 CAN 协议触发的数据索引为 1。
:TRIGger:CAN:DATAIndex? -> 1	查询 CAN 协议触发的数据索引为 1。

:TRIGger:CAN:TRIGData

命令格式

:TRIGger:CAN:TRIGData <Int>

:TRIGger:CAN:TRIGData?

功能描述

设置 CAN 协议触发的触发数据。

查询 CAN 协议触发的触发数据。

注 1：触发模式设置为**标准数据帧**和**扩展数据帧**时有效，相关指令:TRIGger:CAN:TRIGMode。

注 2：当 DLC 值大于 0 时有效，相关指令:TRIGger:CAN:DLC。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 CAN 协议触发的触发数据。

实例说明

:TRIGger:CAN:TRIGData 0x10	设置 CAN 协议触发的触发数据为 0x10。
:TRIGger:CAN:TRIGData? -> 0x10	查询 CAN 协议触发的触发数据为 0x10。

:TRIGger:CANFd:SOURce

命令格式

:TRIGger:CANFd:SOURce <String>

:TRIGger:CANFd:SOURce?

功能描述

设置 CAN-FD 协议触发的信源。

查询 CAN-FD 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:CANFd:SOURce CHANnel1　　设置 CAN-FD 协议触发信源为通道 1。

:TRIGger:CANFd:SOURce? -> CHANnel1　　查询 CAN-FD 协议触发信源为通道 1。

:TRIGger:CANFd:TRIGMode

命令格式

:TRIGger:CANFd:TRIGMode <String>

:TRIGger:CANFd:TRIGMode?

功能描述

设置 CAN-FD 协议触发的触发模式。

查询 CAN-FD 协议触发的触发模式。

参数说明

<String> 范围 {STARtbit | STDData | EXTData | STDRemote | EXTRemote | ERR | FDSTddata | FDEXtdata }，分别表示开始位、标准数据帧、扩展数据帧、标准远程帧、扩展远程帧、错误/过载帧、FD 标准数据帧和 FD 扩展数据帧。

返回格式

查询命令返回 STARtbit、STDData、EXTData、STDRemote、EXTRemote、ERR、FDSTddata 或 FDEXtdata。

实例说明

:TRIGger:CANFd:TRIGMode STARtbit 设置 CAN-FD 协议触发的触发模式为开始位。

:TRIGger:CANFd:TRIGMode? -> STARtbit 查询 CAN-FD 协议触发的触发模式为开始位。

:TRIGger:CANFd:BUSType

命令格式

:TRIGger:CANFd:BUSType <String>

:TRIGger:CANFd:BUSType?

功能描述

设置 CAN-FD 协议触发的总线类型。

查询 CAN-FD 协议触发的总线类型。

参数说明

<String> 范围 { CANL | CANH | CANDiff }。

返回格式

查询命令返回 CANL、CANH 或 CANDiff。

实例说明

:TRIGger:CANFd:BUSType CANL　　设置 CAN-FD 协议触发的总线类型为 CANL。

:TRIGger:CANFd:BUSType? -> CANL　　查询 CAN-FD 协议触发的总线类型为 CANL。

:TRIGger:CANFd:BAUDrate

命令格式

:TRIGger:CANFd:BAUDrate <Double>

:TRIGger:CANFd:BAUDrate?

功能描述

设置 CAN-FD 协议触发的波特率。

查询 CAN-FD 协议触发的波特率。

参数说明

<Double> 波特率，范围 1.00k~10000.00k。

返回格式

查询命令以实型形式返回 CAN-FD 协议触发的波特率大小。

实例说明

:TRIGger:CANFd:BAUDrate 500	设置 CAN-FD 协议触发的波特率为 500k。
:TRIGger:CANFd:BAUDrate? -> 500.00	查询 CAN-FD 协议触发的波特率为 500k。

:TRIGger:CANFd:FDBAud

命令格式

:TRIGger:CANFd:FDBAud <Double>

:TRIGger:CANFd:FDBAud?

功能描述

设置 CAN-FD 协议触发的 FD 波特率。

查询 CAN-FD 协议触发的 FD 波特率。

参数说明

<Double> FD 波特率，范围 1.00k~10000.00k。

注：FD 波特率设置始终要大于或等于波特率设置，相关指令:TRIGger:CANFd:BAUDrate。

返回格式

查询命令以实型形式返回 CAN-FD 协议触发的 FD 波特率大小。

实例说明

:TRIGger:CANFd:FDBAud 500　　设置 CAN-FD 协议触发的 FD 波特率为 500k。

:TRIGger:CANFd:FDBAud? -> 500.00　　查询 CAN-FD 协议触发的 FD 波特率为 500k。

:TRIGger:CANFd:ID

命令格式

:TRIGger:CANFd:ID <Int>

:TRIGger:CANFd:ID?

功能描述

设置 CAN-FD 协议触发的 ID 值。

查询 CAN-FD 协议触发的 ID 值。

注：触发模式设置为**标准数据帧**、**扩展数据帧**、**标准远程帧**、**扩展远程帧**、**FD 标准数据帧**和 **FD 扩展数据帧**时有效，相关指令:TRIGger:CANFd:TRIGMode。

参数说明

<Int> 范围 0x00~0x1FFFFFFF。

注：ID 值范围根据触发模式变化，扩展数据帧、扩展远程帧和 FD 扩展数据帧最大是 0x1FFFFFFF，其余最大是 0x7FF。

返回格式

查询命令以十六进制形式返回 CAN-FD 协议触发的 ID 值。

实例说明

:TRIGger:CANFd:ID 0x10	设置 CAN-FD 协议触发的 ID 值为 0x10。
:TRIGger:CANFd:ID? -> 0x10	查询 CAN-FD 协议触发的 ID 值为 0x10。

:TRIGger:CANFd:DLC

命令格式

:TRIGger:CANFd:DLC <String>

:TRIGger:CANFd:DLC?

功能描述

设置 CAN-FD 协议触发的 DLC 值。

查询 CAN-FD 协议触发的 DLC 值。

注：触发模式设置为**标准数据帧**、**扩展数据帧**、**标准远程帧**和**扩展远程帧**时有效，相关指令:TRIGger:CANFd:TRIGMode。

参数说明

<String> 范围 {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | NONE }。

返回格式

查询命令返回 1、2、3、4、5、6、7、8 或 NONE。

实例说明

:TRIGger:CANFd:DLC 2　　设置 CAN-FD 协议触发的 DLC 值为 2。

:TRIGger:CANFd:DLC? -> 2　　查询 CAN-FD 协议触发的 DLC 值为 2。

:TRIGger:CANFd:DATAIndex

命令格式

:TRIGger:CANFd:DATAIndex <Int>

:TRIGger:CANFd:DATAIndex?

功能描述

设置 CAN-FD 协议触发的数据索引。

查询 CAN-FD 协议触发的数据索引。

注 1：触发模式设置为**标准数据帧**和**扩展数据帧**时有效，相关指令:TRIGger:CANFd:TRIGMode。

注 2：当 DLC 值大于 0 时有效，相关指令:TRIGger:CANFd:DLC。

参数说明

<Int> 范围 0~7。

注：数据索引范围根据 DLC 值变化。

返回格式

查询命令以整数形式返回 CAN-FD 协议触发的数据索引。

实例说明

:TRIGger:CANFd:DATAIndex 1　　设置 CAN-FD 协议触发的数据索引为 1。

:TRIGger:CANFd:DATAIndex? -> 1　　查询 CAN-FD 协议触发的数据索引为 1。

:TRIGger:CANFd:TRIGData

命令格式

:TRIGger:CANFd:TRIGData <Int>

:TRIGger:CANFd:TRIGData?

功能描述

设置 CAN-FD 协议触发的触发数据。

查询 CAN-FD 协议触发的触发数据。

注 1：触发模式设置为**标准数据帧**和**扩展数据帧**时有效，相关指令:TRIGger:CANFd:TRIGMode。

注 2：当 DLC 值大于 0 时有效，相关指令:TRIGger:CANFd:DLC。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 CAN-FD 协议触发的触发数据。

实例说明

:TRIGger:CANFd:TRIGData 0x10　　设置 CAN-FD 协议触发的触发数据为 0x10。

:TRIGger:CANFd:TRIGData? -> 0x10　　查询 CAN-FD 协议触发的触发数据为 0x10。

:TRIGger:CANFd:FDDLc

命令格式

:TRIGger:CANFd:FDDLc <String>

:TRIGger:CANFd:FDDLc?

功能描述

设置 CAN-FD 协议触发的 FD DLC 值。

查询 CAN-FD 协议触发的 FD DLC 值。

注：触发模式设置为 **FD 标准数据帧**和 **FD 扩展数据帧**时有效，相关指令:TRIGger:CANFd:TRIGMode。

参数说明

<String> 范围 {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | NONE }。

返回格式

查询命令返回 1、2、3、4、5、6、7、8、9、10、11、12、14、15 或 NONE。

实例说明

:TRIGger:CANFd:FDDLc 2　　设置 CAN-FD 协议触发的 FD DLC 值为 2。

:TRIGger:CANFd:FDDLc? -> 2　查询 CAN-FD 协议触发的 FD DLC 值为 2。

:TRIGger:CANFd:BRS

命令格式

:TRIGger:CANFd:BRS <String>

:TRIGger:CANFd:BRS?

功能描述

设置 CAN-FD 协议触发的 BRS 位。

查询 CAN-FD 协议触发的 BRS 位。

注 1:触发模式设置为 **FD 标准数据帧**和 **FD 扩展数据帧**时有效,相关指令:TRIGger:CANFd:TRIGMode。

注 2: 当 FD DLC 值为 **NONE** 时无效，相关指令:TRIGger:CANFd:FDDLc。

参数说明

<String> 范围 { DOMInant | RECEssive }，分别表示显性和隐性。

返回格式

查询命令返回 DOMInant 或 RECEssive。

实例说明

:TRIGger:CANFd:BRS RECEssive　　设置 CAN-FD 协议触发的 BRS 位为隐性。

:TRIGger:CANFd:BRS? -> RECEssive　　查询 CAN-FD 协议触发的 BRS 位为隐性。

:TRIGger:CANFd:ESI

命令格式

:TRIGger:CANFd:ESI <String>

:TRIGger:CANFd:ESI?

功能描述

设置 CAN-FD 协议触发的 ESI 位。

查询 CAN-FD 协议触发的 ESI 位。

注 1:仅触发模式设置为 **FD 标准数据帧**和 **FD 扩展数据帧**有效,相关指令:TRIGger:CANFd:TRIGMode。

注 2: 当 FD DLC 值为 **NONE** 时无效，相关指令:TRIGger:CANFd:FDDLc。

参数说明

<String> 范围 { DOMInant | RECEssive }，分别表示显性和隐性。

返回格式

查询命令返回 DOMInant 或 RECEssive。

实例说明

:TRIGger:CANFd:ESI RECEssive　　设置 CAN-FD 协议触发的 ESI 位为隐性。

:TRIGger:CANFd:ESI? -> RECEssive　　查询 CAN-FD 协议触发的 ESI 位为隐性。

:TRIGger:DALI:SOURce

命令格式

:TRIGger:DALI:SOURce <String>

:TRIGger:DALI:SOURce?

功能描述

设置 DALI 协议触发的信源。

查询 DALI 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:DALI:SOURce CHANnel1　　设置 DALI 协议触发信源为通道 1。

:TRIGger:DALI:SOURce? -> CHANnel1　　查询 DALI 协议触发信源为通道 1。

:TRIGger:DALI:TRIGMode

命令格式

:TRIGger:DALI:TRIGMode <String>

:TRIGger:DALI:TRIGMode?

功能描述

设置 DALI 协议触发的触发模式。

查询 DALI 协议触发的触发模式。

参数说明

<String> 范围 {BEGIn | FORWard19 | FORWard27 | BACKward }。

返回格式

查询命令返回 BEGIn、FORWard19、FORWard27 或 BACKward。

实例说明

:TRIGger:DALI:TRIGMode BEGIn	设置 DALI 协议触发的触发模式为 BEGIn。
:TRIGger:DALI:TRIGMode? -> BEGIn	查询 DALI 协议触发的触发模式为 BEGIn。

:TRIGger:DALI:ADDRess

命令格式

:TRIGger:DALI:ADDRess <Int>

:TRIGger:DALI:ADDRess?

功能描述

设置 DALI 协议触发的地址位。

查询 DALI 协议触发的地址位。

注：触发模式设置为 Forward19 和 Forward27 时有效，相关指令:TRIGger:DALI:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 DALI 协议触发的地址位。

实例说明

:TRIGger:DALI:ADDRess 0x10　　设置 DALI 协议触发的地址位为 0x10。

:TRIGger:DALI:ADDRess? -> 0x10　　查询 DALI 协议触发的地址位为 0x10。

:TRIGger:DALI:DATA

命令格式

:TRIGger:DALI:DATA <Int>

:TRIGger:DALI:DATA?

功能描述

设置 DALI 协议触发的数据位。

查询 DALI 协议触发的数据位。

注：触发模式设置为 **Backward** 时有效，相关指令:TRIGger:DALI:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 DALI 协议触发的数据位。

实例说明

:TRIGger:DALI:DATA 0x10　　设置 DALI 协议触发的数据位为 0x10。

:TRIGger:DALI:DATA? -> 0x10　　查询 DALI 协议触发的数据位为 0x10。

:TRIGger:DALI:CMD<x>

命令格式

:TRIGger:DALI:CMD<x> <Int>

:TRIGger:DALI:CMD<x>?

功能描述

设置 DALI 协议触发的命令 1/命令 2。

查询 DALI 协议触发的命令 1/命令 2。

注：触发模式设置为 **Forward27** 时有效，相关指令:TRIGger:DALI:TRIGMode。

参数说明

<x> 指令编号，范围 { 1 | 2 }。

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 DALI 协议触发的命令 1/命令 2。

实例说明

:TRIGger:DALI:CMD1 0x10　　设置 DALI 协议触发的命令 1 为 0x10。

:TRIGger:DALI:CMD1? -> 0x10　　查询 DALI 协议触发的命令 1 为 0x10。

:TRIGger:DHT11:SOURce

命令格式

:TRIGger:DHT11:SOURce <String>

:TRIGger:DHT11:SOURce?

功能描述

设置 DHT11 协议触发的信源。

查询 DHT11 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:DHT11:SOURce CHANnel1　　设置 DHT11 协议触发信源为通道 1。

:TRIGger:DHT11:SOURce? -> CHANnel1　　查询 DHT11 协议触发信源为通道 1。

:TRIGger:DHT11:TRIGMode

命令格式

:TRIGger:DHT11:TRIGMode <String>

:TRIGger:DHT11:TRIGMode?

功能描述

设置 DHT11 协议触发的触发模式。

查询 DHT11 协议触发的触发模式。

参数说明

<String> 范围 ｛BEGIn｝，表示开始触发。

返回格式

查询命令返回 BEGIn。

实例说明

:TRIGger:DHT11:TRIGMode BEGIn　　设置 DHT11 协议触发的触发模式为开始触发。

:TRIGger:DHT11:TRIGMode? -> BEGIn　　查询 DHT11 协议触发的触发模式为开始触发。

:TRIGger:DMANchester:SOURce

命令格式

:TRIGger:DMANchester:SOURce <String>

:TRIGger:DMANchester:SOURce?

功能描述

设置 DManchester 协议触发的信源。

查询 DManchester 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:DMANchester:SOURce CHANnel1 设置 DManchester 协议触发信源为通道 1。

:TRIGger:DMANchester:SOURce? -> CHANnel1 查询 DManchester 协议触发信源为通道 1。

:TRIGger:DMANchester:BITLen

命令格式

:TRIGger:DMANchester:BITLen <String>

:TRIGger:DMANchester:BITLen?

功能描述

设置 DManchester 协议触发的位时长。

查询 DManchester 协议触发的位时长。

参数说明

<String> 范围 100ns~1ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 DManchester 协议触发的位时长。

实例说明

:TRIGger:DMANchester:BITLen 2e-4 设置 DManchester 协议触发的位时长为 200us。

:TRIGger:DMANchester:BITLen? -> 1.000ms 查询 DManchester 协议触发的位时长为 1ms。

:TRIGger:DMANchester:FRAMestart

命令格式

:TRIGger:DMANchester:FRAMestart <String>

:TRIGger:DMANchester:FRAMestart?

功能描述

设置 DManchester 协议触发的包起始位。

查询 DManchester 协议触发的包起始位。

参数说明

<String> 范围 { 0 | 1}。

返回格式

查询命令返回 0 或 1。

实例说明

:TRIGger:DMANchester:FRAMestart 1　　设置 DManchester 协议触发的包起始位为 1。

:TRIGger:DMANchester:FRAMestart? -> 1　　查询 DManchester 协议触发的包起始位为 1。

:TRIGger:DMANchester:TRIGMode

命令格式

:TRIGger:DMANchester:TRIGMode <String>

:TRIGger:DMANchester:TRIGMode?

功能描述

设置 DManchester 协议触发的触发模式。

查询 DManchester 协议触发的触发模式。

参数说明

<String> 范围 ｛BEGIn｝，表示包起始位。

返回格式

查询命令返回 BEGIn。

实例说明

:TRIGger:DMANchester:TRIGMode BEGIn 设置 DManchester 协议触发的触发模式为包起始位。

:TRIGger:DMANchester:TRIGMode? -> BEGIn 查询 DManchester 协议触发的触发模式为包起始位。

:TRIGger:DMX512:SOURce

命令格式

:TRIGger:DMX512:SOURce <String>

:TRIGger:DMX512:SOURce?

功能描述

设置 DMX512 协议触发的信源。

查询 DMX512 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:DMX512:SOURce CHANnel1　　设置 DMX512 协议触发信源为通道 1。

:TRIGger:DMX512:SOURce? -> CHANnel1　　查询 DMX512 协议触发信源为通道 1。

:TRIGger:DMX512:TRIGMode

命令格式

:TRIGger:DMX512:TRIGMode <String>

:TRIGger:DMX512:TRIGMode?

功能描述

设置 DMX512 协议触发的触发模式。

查询 DMX512 协议触发的触发模式。

参数说明

<String> 范围 { BREAk | Data }，分别表示复位触发和数据触发。

返回格式

查询命令返回 BREAk 或 Data。

实例说明

:TRIGger:DMX512:TRIGMode BREAk 设置 DMX512 协议触发的触发模式为复位触发。

:TRIGger:DMX512:TRIGMode? -> BREAk 查询 DMX512 协议触发的触发模式为复位触发。

:TRIGger:DMX512:BUSType

命令格式

:TRIGger:DMX512:BUSType <String>

:TRIGger:DMX512:BUSType?

功能描述

设置 DMX512 协议触发的总线类型。

查询 DMX512 协议触发的总线类型。

参数说明

<String> 范围 { - | + | COMMon}。

返回格式

查询命令返回-，+或 COMMon。

实例说明

:TRIGger:DMX512:BUSType +	设置 DMX512 协议触发的总线类型为+。
:TRIGger:DMX512:BUSType? -> +	查询 DMX512 协议触发的总线类型为+。

:TRIGger:DMX512:TRIGData

命令格式

:TRIGger:DMX512:TRIGData <Int>

:TRIGger:DMX512:TRIGData?

功能描述

设置 DMX512 协议触发的触发数据。

查询 DMX512 协议触发的触发数据。

注 1：触发模式设置为**数据触发**时有效，相关指令:TRIGger:DMX512:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 DMX512 协议触发的触发数据。

实例说明

:TRIGger:DMX512:TRIGData 0x10　　设置 DMX512 协议触发的触发数据为 0x10。

:TRIGger:DMX512:TRIGData? -> 0x10　　查询 DMX512 协议触发的触发数据为 0x10。

:TRIGger:DS18B20:SOURce

命令格式

:TRIGger:DS18B20:SOURce <String>

:TRIGger:DS18B20:SOURce?

功能描述

设置 DS18B20 协议触发的信源。

查询 DS18B20 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:DS18B20:SOURce CHANnel1　　设置 DS18B20 协议触发信源为通道 1。

:TRIGger:DS18B20:SOURce? -> CHANnel1　　查询 DS18B20 协议触发信源为通道 1。

:TRIGger:DS18B20:TRIGMode

命令格式

:TRIGger:DS18B20:TRIGMode <String>

:TRIGger:DS18B20:TRIGMode?

功能描述

设置 DS18B20 协议触发的触发模式。

查询 DS18B20 协议触发的触发模式。

参数说明

<String> 范围 ｛STARtfield | COMMand｝，分别表示开始段触发和指令触发。

返回格式

查询命令返回 STARtfield 或 COMMand。

实例说明

:TRIGger:DS18B20:TRIGMode STARtfield 设置 DS18B20 协议触发的触发模式为开始段触发。

:TRIGger:DS18B20:TRIGMode? -> STARtfield 查询 DS18B20 协议触发的触发模式为开始段触发。

:TRIGger:DS18B20:ROMCmd

命令格式

:TRIGger:DS18B20:ROMCmd <String>

:TRIGger:DS18B20:ROMCmd?

功能描述

设置 DS18B20 协议触发的 ROM 指令。

查询 DS18B20 协议触发的 ROM 指令。

注：触发模式设置为**指令触发**时有效，相关指令:TRIGger:DS18B20:TRIGMode。

参数说明

<String> 范围 { NONE | READ | SEARch | MATCh | ALARm | SKIP }。

返回格式

查询命令返回 NONE、READ、SEARch、MATCh、ALARm 或 SKIP。

实例说明

:TRIGger:DS18B20:ROMCmd READ 设置 DS18B20 协议触发的 ROM 指令为 READ。

:TRIGger:DS18B20:ROMCmd? -> READ 查询 DS18B20 协议触发的 ROM 指令为 READ。

:TRIGger:DS18B20:RAMCmd

命令格式

:TRIGger:DS18B20:RAMCmd <String>

:TRIGger:DS18B20:RAMCmd?

功能描述

设置 DS18B20 协议触发的 RAM 指令。

查询 DS18B20 协议触发的 RAM 指令。

注：触发模式设置为**指令触发**时有效，相关指令:TRIGger:DS18B20:TRIGMode。

参数说明

<String> 范围 { NONE | CONVert | READ | WRITe | COPY | RESTore | POWEr }。

返回格式

查询命令返回 NONE、CONVert、READ、WRITe、COPY、RESTore 或 POWEr。

实例说明

:TRIGger:DS18B20:RAMCmd CONVert 设置 DS18B20 协议触发的 RAM 指令为 CONVert。

:TRIGger:DS18B20:RAMCmd? -> CONVert 查询 DS18B20 协议触发的 RAM 指令为 CONVert。

:TRIGger:DSI:D+

命令格式

:TRIGger:DSI:D+ <String>

:TRIGger:DSI:D+?

功能描述

设置 MIPI-DSI 协议触发的 D+的信源。

查询 MIPI-DSI 协议触发的 D+的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:DSI:D+ CHANnel1　　设置 MIPI-DSI 协议触发的 D+信源为通道 1。

:TRIGger:DSI:D+? -> CHANnel1　　查询 MIPI-DSI 协议触发的 D+信源为通道 1。

:TRIGger:DSI:D-

命令格式

:TRIGger:DSI:D- <String>

:TRIGger:DSI:D-?

功能描述

设置 MIPI-DSI 协议触发的 D-的信源。

查询 MIPI-DSI 协议触发的 D-的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:DSI:D- CHANnel1	设置 MIPI-DSI 协议触发的 D-信源为通道 1。
:TRIGger:DSI:D-? -> CHANnel1	查询 MIPI-DSI 协议触发的 D-信源为通道 1。

:TRIGger:DSI:TRIGMode

命令格式

:TRIGger:DSI:TRIGMode <String>

:TRIGger:DSI:TRIGMode?

功能描述

设置 MIPI-DSI 协议触发的触发模式。

查询 MIPI-DSI 协议触发的触发模式。

参数说明

<String> 范围 { STARt | TRANsmode | BTA }，分别表示起始触发、传输模式触发和总线转向触发。

返回格式

查询命令返回 STARt、TRANsmode 或 BTA。

实例说明

:TRIGger:DSI:TRIGMode STARt 设置 MIPI-DSI 协议触发的触发模式为起始触发。

:TRIGger:DSI:TRIGMode? -> STARt 查询 MIPI-DSI 协议触发的触发模式为起始触发。

:TRIGger:DSI:TRANsmode

命令格式

:TRIGger:DSI:TRANsmode <String>

:TRIGger:DSI:TRANsmode?

功能描述

设置 MIPI-DSI 协议触发的传输模式。

查询 MIPI-DSI 协议触发的传输模式。

注：触发模式设置为**传输模式触发**时有效，相关指令:TRIGger:DSI:TRIGMode。

参数说明

<String> 范围 { LPDT | ACK | ULPS | RAR | TER }。

返回格式

查询命令返回 LPDT、ACK、ULPS、RAR 或 TER。

实例说明

:TRIGger:DSI:TRANsmode LPDT 设置 MIPI-DSI 协议触发的传输模式为 LPDT。

:TRIGger:DSI:TRANsmode? -> LPDT 查询 MIPI-DSI 协议触发的传输模式为 LPDT。

:TRIGger:DSI:MATChmode

命令格式

:TRIGger:DSI:MATChmode <String>

:TRIGger:DSI:MATChmode?

功能描述

设置 MIPI-DSI 协议触发的匹配模式。

查询 MIPI-DSI 协议触发的匹配模式。

注：传输模式设置为 **LPDT** 时有效，相关指令:TRIGger:DSI:TRANsmode。

参数说明

<String> 范围 { NONE | DATAtype | VIRTualchannel | SHORtpacket | LONGpacket }，分别表示无数据、数据类型、虚拟通道、短数据包和长数据包。

返回格式

查询命令返回 NONE、DATAtype、VIRTualchannel、SHORtpacket 或 LONGpacket。

实例说明

:TRIGger:DSI:MATChmode DATAtype 设置 MIPI-DSI 协议触发的匹配模式为数据类型。

:TRIGger:DSI:MATChmode? -> DATAtype 查询 MIPI-DSI 协议触发的匹配模式为数据类型。

:TRIGger:DSI:DATAtype

命令格式

:TRIGger:DSI:DATAtype <Int>

:TRIGger:DSI:DATAtype?

功能描述

设置 MIPI-DSI 协议触发的数据类型。

查询 MIPI-DSI 协议触发的数据类型。

注：匹配模式设置为**数据类型**、**虚拟通道**、**短数据包**和**长数据包**时有效，相关指令:TRIGger:DSI:MATChmode。

参数说明

<Int> 范围 0x00~0x3F。

返回格式

查询命令以十六进制形式返回 MIPI-DSI 协议触发的数据类型。

实例说明

:TRIGger:DSI:DATAtype 0x10　　设置 MIPI-DSI 协议触发的数据类型为 0x10。

:TRIGger:DSI:DATAtype? -> 0x10　　查询 MIPI-DSI 协议触发的数据类型为 0x10。

:TRIGger:DSI:VIRTualchan

命令格式

:TRIGger:DSI:VIRTualchan <Int>

:TRIGger:DSI:VIRTualchan?

功能描述

设置 MIPI-DSI 协议触发的虚拟通道号。

查询 MIPI-DSI 协议触发的虚拟通道号。

注：匹配模式设置为**虚拟通道**、**短数据包**和**长数据包**时有效，相关指令:TRIGger:DSI:MATChmode。

参数说明

<Int> 范围 0x00~0x03。

返回格式

查询命令以十六进制形式返回 MIPI-DSI 协议触发的虚拟通道号。

实例说明

:TRIGger:DSI:VIRTualchan 0x02　　设置 MIPI-DSI 协议触发的虚拟通道号为 0x02。

:TRIGger:DSI:VIRTualchan? -> 0x02　　查询 MIPI-DSI 协议触发的虚拟通道号为 0x02。

:TRIGger:DSI:DATACnt

命令格式

:TRIGger:DSI:DATACnt <Int>

:TRIGger:DSI:DATACnt?

功能描述

设置 MIPI-DSI 协议触发的 Data0/数据数。

查询 MIPI-DSI 协议触发的 Data0/数据数。

注：匹配模式设置为**短数据包**和**长数据包**时有效，相关指令:TRIGger:DSI:MATChmode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MIPI-DSI 协议触发的 Data0/数据数。

实例说明

:TRIGger:DSI:DATACnt 0x10　　设置 MIPI-DSI 协议触发的 Data0/数据数为 0x10。

:TRIGger:DSI:DATACnt? -> 0x10　　查询 MIPI-DSI 协议触发的 Data0/数据数为 0x10。

:TRIGger:DSI:DATA1

命令格式

:TRIGger:DSI:DATA1 <Int>

:TRIGger:DSI:DATA1?

功能描述

设置 MIPI-DSI 协议触发的 Data1。

查询 MIPI-DSI 协议触发的 Data1。

注：匹配模式设置为**短数据包**时有效，相关指令:TRIGger:DSI:MATChmode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MIPI-DSI 协议触发的 Data1。

实例说明

:TRIGger:DSI:DATA1 0x10　　设置 MIPI-DSI 协议触发的 Data1 为 0x10。

:TRIGger:DSI:DATA1? -> 0x10　查询 MIPI-DSI 协议触发的 Data1 为 0x10。

:TRIGger:FLEXray:TXD

命令格式

:TRIGger:FLEXray:TXD <String>

:TRIGger:FLEXray:TXD?

功能描述

设置 FlexRay 协议触发的 TxD 的信源。

查询 FlexRay 协议触发的 TxD 的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:FLEXray:TXD CHANnel1　　设置 FlexRay 协议触发的 TxD 信源为通道 1。

:TRIGger:FLEXray:TXD? -> CHANnel1　　查询 FlexRay 协议触发的 TxD 信源为通道 1。

:TRIGger:FLEXray:SPEEd

命令格式

:TRIGger:FLEXray:SPEEd <String>

:TRIGger:FLEXray:SPEEd?

功能描述

设置 FlexRay 协议触发的通信速率。

查询 FlexRay 协议触发的通信速率。

参数说明

<String> 范围 { 2.5Mbps | 5Mbps | 10Mbps }。

返回格式

查询命令返回 2.5Mbps、5Mbps 或 10Mbps。

实例说明

:TRIGger:FLEXray:SPEEd 2.5Mbps 设置 FlexRay 协议触发的通信速率为 2.5Mbps。

:TRIGger:FLEXray:SPEEd? -> 2.5Mbps 查询 FlexRay 协议触发的通信速率为 2.5Mbps。

:TRIGger:FLEXray:CHANnel

命令格式

:TRIGger:FLEXray:CHANnel <String>

:TRIGger:FLEXray:CHANnel?

功能描述

设置 FlexRay 协议触发的信道。

查询 FlexRay 协议触发的信道。

参数说明

<String> 范围 { A | B },分别表示 ChannelA、ChannelB。

返回格式

查询命令返回 A 或 B。

实例说明

:TRIGger:FLEXray:CHANnel A	设置 FlexRay 协议触发的信道为 ChannelA。
:TRIGger:FLEXray:CHANnel? -> A	查询 FlexRay 协议触发的信道为 ChannelA。

:TRIGger:FLEXray:TRIGMode

命令格式

:TRIGger:FLEXray:TRIGMode <String>

:TRIGger:FLEXray:TRIGMode?

功能描述

设置 FlexRay 协议触发的触发模式。

查询 FlexRay 协议触发的触发模式。

参数说明

<String> 范围 { TSS | FRAMeid }。

返回格式

查询命令返回 TSS 或 FRAMeid。

实例说明

:TRIGger:FLEXray:TRIGMode TSS	设置 FlexRay 协议触发的触发模式为 TSS。
:TRIGger:FLEXray:TRIGMode? ->TSS	查询 FlexRay 协议触发的触发模式为 TSS。

:TRIGger:FLEXray:FRAMeid

命令格式

:TRIGger:FLEXray:FRAMeid <Int>

:TRIGger:FLEXray:FRAMeid?

功能描述

设置 FlexRay 协议触发的 FrameID。

查询 FlexRay 协议触发的 FrameID。

注：触发模式设置为 **FrameID** 时有效，相关指令:TRIGger:FLEXray:TRIGMode。

参数说明

<Int> 范围 0x00~0x7FF。

返回格式

查询命令以十六进制形式返回 FlexRay 协议触发的 FrameID。

实例说明

:TRIGger:FLEXray:FRAMeid 0x10　　设置 FlexRay 协议触发的 FrameID 为 0x10。

:TRIGger:FLEXray:FRAMeid? -> 0x10　　查询 FlexRay 协议触发的 FrameID 为 0x10。

:TRIGger:FLEXray:PAYLoad

命令格式

:TRIGger:FLEXray:PAYLoad <String>

:TRIGger:FLEXray:PAYLoad?

功能描述

设置 FlexRay 协议触发的负载帧标志。

查询 FlexRay 协议触发的负载帧标志。

注：触发模式设置为 **FrameID** 时有效，相关指令:TRIGger:FLEXray:TRIGMode。

参数说明

<String> 范围 { 0 | 1 }。

返回格式

查询命令返回 0 或 1。

实例说明

:TRIGger:FLEXray:PAYLoad 0	设置 FlexRay 协议触发的负载帧标志为 0。
:TRIGger:FLEXray:PAYLoad? -> 0	查询 FlexRay 协议触发的负载帧标志为 0。

:TRIGger:FLEXray:NULL

命令格式

:TRIGger:FLEXray:NULL <String>

:TRIGger:FLEXray:NULL?

功能描述

设置 FlexRay 协议触发的空帧标志。

查询 FlexRay 协议触发的空帧标志。

注：触发模式设置为 **FrameID** 时有效，相关指令:TRIGger:FLEXray:TRIGMode。

参数说明

<String> 范围 { 0 | 1 }。

返回格式

查询命令返回 0 或 1。

实例说明

:TRIGger:FLEXray:NULL 0　　设置 FlexRay 协议触发的空帧标志为 0。

:TRIGger:FLEXray:NULL? -> 0　　查询 FlexRay 协议触发的空帧标志为 0。

:TRIGger:FLEXray:SYNC

命令格式

:TRIGger:FLEXray:SYNC <String>

:TRIGger:FLEXray:SYNC?

功能描述

设置 FlexRay 协议触发的同步帧标志。

查询 FlexRay 协议触发的同步帧标志。

注：触发模式设置为 **FrameID** 时有效，相关指令:TRIGger:FLEXray:TRIGMode。

参数说明

<String> 范围 { 0 | 1 }。

返回格式

查询命令返回 0 或 1。

实例说明

:TRIGger:FLEXray:SYNC 0　　设置 FlexRay 协议触发的同步帧标志为 0。

:TRIGger:FLEXray:SYNC? -> 0　　查询 FlexRay 协议触发的同步帧标志为 0。

:TRIGger:FLEXray:STARtup

命令格式

:TRIGger:FLEXray:STARtup <String>

:TRIGger:FLEXray:STARtup?

功能描述

设置 FlexRay 协议触发的起始帧标志。

查询 FlexRay 协议触发的起始帧标志。

注：触发模式设置为 **FrameID** 时有效，相关指令:TRIGger:FLEXray:TRIGMode。

参数说明

<String> 范围 { 0 | 1 }。

返回格式

查询命令返回 0 或 1。

实例说明

:TRIGger:FLEXray:STARtup 0　　设置 FlexRay 协议触发的起始帧标志为 0。

:TRIGger:FLEXray:STARtup? -> 0　　查询 FlexRay 协议触发的起始帧标志为 0。

:TRIGger:HDQ:SOURce

命令格式

:TRIGger:HDQ:SOURce <String>

:TRIGger:HDQ:SOURce?

功能描述

设置 HDQ 协议触发的信源。

查询 HDQ 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:HDQ:SOURce CHANnel1　　设置 HDQ 协议触发信源为通道 1。

:TRIGger:HDQ:SOURce? -> CHANnel1　　查询 HDQ 协议触发信源为通道 1。

:TRIGger:HDQ:TRIGMode

命令格式

:TRIGger:HDQ:TRIGMode <String>

:TRIGger:HDQ:TRIGMode?

功能描述

设置 HDQ 协议触发的触发模式。

查询 HDQ 协议触发的触发模式。

参数说明

<String> 范围 { BREAk | COMMand }，分别表示复位段和指令。

返回格式

查询命令返回 BREAk 或 COMMand。

实例说明

:TRIGger:HDQ:TRIGMode BREAk	设置 HDQ 协议触发的触发模式为复位段。
:TRIGger:HDQ:TRIGMode? ->BREAk	查询 HDQ 协议触发的触发模式为复位段。

:TRIGger:HDQ:OPERate

命令格式

:TRIGger:HDQ:OPERate <String>

:TRIGger:HDQ:OPERate?

功能描述

设置 HDQ 协议触发的操作模式。

查询 HDQ 协议触发的操作模式。

注：触发模式设置为**指令**时有效，相关指令:TRIGger:HDQ:TRIGMode。

参数说明

<String> 范围 { READ | WRITe }，分别表示读寄存器和写寄存器。

返回格式

查询命令返回 READ 或 WRITe。

实例说明

:TRIGger:HDQ:OPERate READ	设置 HDQ 协议触发的操作模式为读寄存器。
:TRIGger:HDQ:OPERate? -> READ	查询 HDQ 协议触发的操作模式为读寄存器。

:TRIGger:HDQ:REGAddr

命令格式

:TRIGger:HDQ:REGAddr <Int>

:TRIGger:HDQ:REGAddr?

功能描述

设置 HDQ 协议触发的寄存器地址。

查询 HDQ 协议触发的寄存器地址。

注：触发模式设置为**指令**时有效，相关指令:TRIGger:HDQ:TRIGMode。

参数说明

<Int> 范围 0x00~0x7F。

返回格式

查询命令以十六进制形式返回 HDQ 协议触发的寄存器地址。

实例说明

:TRIGger:HDQ:REGAddr 0x10　　设置 HDQ 协议触发的寄存器地址为 0x10。

:TRIGger:HDQ:REGAddr? -> 0x10　　查询 HDQ 协议触发的寄存器地址为 0x10。

:TRIGger:IIC:SCL

命令格式

:TRIGger:IIC:SCL <String>

:TRIGger:IIC:SCL?

功能描述

设置 IIC 协议触发的时钟信源。

查询 IIC 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:IIC:SCL CHANnel1	设置 IIC 协议触发的时钟信源为通道 1。
:TRIGger:IIC:SCL? -> CHANnel1	查询 IIC 协议触发的时钟信源为通道 1。

:TRIGger:IIC:SDA

命令格式

:TRIGger:IIC:SDA <String>

:TRIGger:IIC:SDA?

功能描述

设置 IIC 协议触发的数据信源。

查询 IIC 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:IIC:SDA CHANnel1	设置 IIC 协议触发的数据信源为通道 1。
:TRIGger:IIC:SDA? -> CHANnel1	查询 IIC 协议触发的数据信源为通道 1。

:TRIGger:IIC:TRIGMode

命令格式

:TRIGger:IIC:TRIGMode <String>

:TRIGger:IIC:TRIGMode?

功能描述

设置 IIC 协议触发的触发模式。

查询 IIC 协议触发的触发模式。

参数说明

<String> 范围 { STARt | END | ADDRess }，分别表示起始位、结束位和地址位。

返回格式

查询命令返回 STARt、END 或 ADDRess。

实例说明

:TRIGger:IIC:TRIGMode STARt　　设置 IIC 协议触发的触发模式为起始位。

:TRIGger:IIC:TRIGMode? -> STARt　　查询 IIC 协议触发的触发模式为起始位。

:TRIGger:IIC:ADDRType

命令格式

:TRIGger:IIC:ADDRType <String>

:TRIGger:IIC:ADDRType?

功能描述

设置 IIC 协议触发的地址类型。

查询 IIC 协议触发的地址类型。

参数说明

<String> 范围 { 7bit | 8bit | 10bit }。

返回格式

查询命令返回 7bit、8bit 或 10bit。

实例说明

:TRIGger:IIC:ADDRType 8bit	设置 IIC 协议触发的地址类型为 8 位。
:TRIGger:IIC:ADDRType? -> 8bit	查询 IIC 协议触发的地址类型为 8 位。

:TRIGger:IIC:ADDREss

命令格式

:TRIGger:IIC:ADDREss <Int>

:TRIGger:IIC:ADDREss?

功能描述

设置 IIC 协议触发的触发地址。

查询 IIC 协议触发的触发地址。

注：触发模式设置为**地址值**时有效，相关指令:TRIGger:IIC:TRIGMode。

参数说明

<Int> 地址类型为 7 位时，范围 0x00~0x7F；地址类型为 8 位时，范围 0x00~0xFF；地址类型为 10 位时，范围 0x00~0x3FF。

返回格式

查询命令以十六进制形式返回 IIC 协议触发的触发地址。

实例说明

:TRIGger:IIC:ADDREss 0x10　　设置 IIC 协议触发的触发地址为 0x10。

:TRIGger:IIC:ADDREss? -> 0x10　　查询 IIC 协议触发的触发地址为 0x10。

:TRIGger:IIC:RWMOde

命令格式

:TRIGger:IIC:RWMOde <String>

:TRIGger:IIC:RWMOde?

功能描述

设置 IIC 协议触发的读写模式。

查询 IIC 协议触发的读写模式。

注：触发模式设置为**地址值**时有效，相关指令:TRIGger:IIC:TRIGMode。

参数说明

<String> 范围 { WRITe | READ }。

返回格式

查询命令返回 WRITe 或 READ。

实例说明

:TRIGger:IIC:RWMOde READ	设置 IIC 协议触发的读写模式为 READ。
:TRIGger:IIC:RWMOde? -> READ	查询 IIC 协议触发的读写模式为 READ。

:TRIGger:IIC:ACKMode

命令格式

:TRIGger:IIC:ACKMode <String>

:TRIGger:IIC:ACKMode?

功能描述

设置 IIC 协议触发的响应类型。

查询 IIC 协议触发的响应类型。

注：触发模式设置为**地址值**时有效，相关指令:TRIGger:IIC:TRIGMode。

参数说明

<String> 范围 { ACK | NACK }。

返回格式

查询命令返回 ACK 或 NACK。

实例说明

:TRIGger:IIC:ACKMode ACK	设置 IIC 协议触发的响应类型为 ACK。
:TRIGger:IIC:ACKMode? -> ACK	查询 IIC 协议触发的响应类型为 ACK。

:TRIGger:IICDev:SCL

命令格式

:TRIGger:IICDev:SCL <String>

:TRIGger:IICDev:SCL?

功能描述

设置 IIC Device 协议触发的时钟信源。

查询 IIC Device 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:IICDev:SCL CHANnel1　　设置 IIC Device 协议触发的时钟信源为通道 1。

:TRIGger:IICDev:SCL? -> CHANnel1　　查询 IIC Device 协议触发的时钟信源为通道 1。

:TRIGger:IICDev:SDA

命令格式

:TRIGger:IICDev:SDA <String>

:TRIGger:IICDev:SDA?

功能描述

设置 IIC Device 协议触发的数据信源。

查询 IIC Device 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:IICDev:SDA CHANnel1　　设置 IIC Device 协议触发的数据信源为通道 1。

:TRIGger:IICDev:SDA? -> CHANnel1　　查询 IIC Device 协议触发的数据信源为通道 1。

:TRIGger:IICDev:DEVIce

命令格式

:TRIGger:IICDev:DEVIce <String>

:TRIGger:IICDev:DEVIce?

功能描述

设置 IIC Device 协议触发的设备型号。

查询 IIC Device 协议触发的设备型号。

参数说明

<String> 范围 { NAU8810 | NAU8811 | NAU8812 | NAU8814 | NAU88C10 | NAU8820 | NAU8822 | NAU88C22 | NAU88L24 | NAU88L25 | NAU8401 | NAU8402 | NAU8501 | NAU8502 | NAU85L40 }。

返回格式

查询命令返回 NAU8810、NAU8811、NAU8812、NAU8814、NAU88C10、NAU8820、NAU8822、NAU88C22、NAU88L24、NAU88L25、NAU8401、NAU8402、NAU8501、NAU8502 或 NAU85L40。

实例说明

:TRIGger:IICDev:DEVIce NAU8810　　设置 IIC Device 协议触发的设备型号为 NAU8810。

:TRIGger:IICDev:DEVIce? -> NAU8810　　查询 IIC Device 协议触发的设备型号为 NAU8810。

:TRIGger:IICDev:TRIGMode

命令格式

:TRIGger:IICDev:TRIGMode <String>

:TRIGger:IICDev:TRIGMode?

功能描述

设置 IIC Device 协议触发的触发模式。

查询 IIC Device 协议触发的触发模式。

参数说明

<String> 范围 { STARt | DEVAddr | REGAddr }，分别表示起始位、设备地址和寄存器地址。

返回格式

查询命令返回 STARt、DEVAddr 或 ADDRess。

实例说明

:TRIGger:IICDev:TRIGMode STARt 设置 IIC Device 协议触发的触发模式为起始位。

:TRIGger:IICDev:TRIGMode? -> STARt 查询 IIC Device 协议触发的触发模式为起始位。

:TRIGger:IICDev:ADDRMode

命令格式

:TRIGger:IICDev:ADDRMode <String>

:TRIGger:IICDev:ADDRMode?

功能描述

设置 IIC Device 协议触发的地址格式。

查询 IIC Device 协议触发的地址格式。

注：触发模式设置为**设备地址**和**寄存器地址**时有效，相关指令:TRIGger:IICDev:TRIGMode。

参数说明

<String> 范围 { 8bit | 7bitW | 7bitR }。

返回格式

查询命令返回 8bit、7bitW 或 7bitR。

实例说明

:TRIGger:IICDev:ADDRMode 8bit 设置 IIC Device 协议触发的地址格式为 8bit。

:TRIGger:IICDev:ADDRMode? -> 8bit 查询 IIC Device 协议触发的地址格式为 8bit。

:TRIGger:IICDev:DEVAddr

命令格式

:TRIGger:IICDev:DEVAddr <Int>

:TRIGger:IICDev:DEVAddr?

功能描述

设置 IIC Device 协议触发的设备地址。

查询 IIC Device 协议触发的设备地址。

注：触发模式设置为**设备地址**和**寄存器地址**时有效，相关指令:TRIGger:IICDev:TRIGMode。

参数说明

<Int> 地址类型为 7 位时，范围 0x00~0x7F；地址类型为 8 位时，范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 IIC Device 协议触发的设备地址。

实例说明

:TRIGger:IICDev:DEVAddr 0x10	设置 IIC Device 协议触发的设备地址为 0x10。
:TRIGger:IICDev:DEVAddr? -> 0x10	查询 IIC Device 协议触发的设备地址为 0x10。

:TRIGger:IICDev:REGAddr

命令格式

:TRIGger:IICDev:REGAddr <Int>

:TRIGger:IICDev:REGAddr?

功能描述

设置 IIC Device 协议触发的寄存器地址。

查询 IIC Device 协议触发的寄存器地址。

注：触发模式设置为**寄存器地址**时有效，相关指令:TRIGger:IICDev:TRIGMode。

参数说明

<Int> 范围 0x00~0x7F。

注：设备型号设置为 NAU88L24、NAU88L25 或 NAU85L40 时，范围为 0x00~0xFFFF，相关指令:TRIGger:IICDev:DEVIce。

返回格式

查询命令以十六进制形式返回 IIC Device 协议触发的寄存器地址。

实例说明

:TRIGger:IICDev:REGAddr 0x10　　设置 IIC Device 协议触发的寄存器地址为 0x10。

:TRIGger:IICDev:REGAddr? -> 0x10　　查询 IIC Device 协议触发的寄存器地址为 0x10。

:TRIGger:ISO7816:RST

命令格式

:TRIGger:ISO7816:RST <String>

:TRIGger:ISO7816:RST?

功能描述

设置 ISO7816 协议触发的复位信源。

查询 ISO7816 协议触发的复位信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:ISO7816:RST CHANnel1　　设置 ISO7816 协议触发的复位信源为通道 1。

:TRIGger:ISO7816:RST? -> CHANnel1　　查询 ISO7816 协议触发的复位信源为通道 1。

:TRIGger:ISO7816:DAT

命令格式

:TRIGger:ISO7816:DAT <String>

:TRIGger:ISO7816:DAT?

功能描述

设置 ISO7816 协议触发的数据信源。

查询 ISO7816 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:ISO7816:DAT CHANnel1　　设置 ISO7816 协议触发的数据信源为通道 1。

:TRIGger:ISO7816:DAT? -> CHANnel1　　查询 ISO7816 协议触发的数据信源为通道 1。

:TRIGger:ISO7816:BAUDrate

命令格式

:TRIGger:ISO7816:BAUDrate <Double>

:TRIGger:ISO7816:BAUDrate?

功能描述

设置 ISO7816 协议触发的波特率。

查询 ISO7816 协议触发的波特率。

参数说明

<Double> 波特率，范围 1.00k~15.00k。

返回格式

查询命令以实型形式返回 ISO7816 协议触发的波特率大小。

实例说明

:TRIGger:ISO7816:BAUDrate 15　　设置 ISO7816 协议触发的波特率为 15k。

:TRIGger:ISO7816:BAUDrate? -> 15.00　　查询 ISO7816 协议触发的波特率为 15k。

:TRIGger:ISO7816:CRCLen

命令格式

:TRIGger:ISO7816:CRCLen <String>

:TRIGger:ISO7816:CRCLen?

功能描述

设置 ISO7816 协议触发的 CRC 长度。

查询 ISO7816 协议触发的 CRC 长度。

参数说明

<String> 范围 { 1 | 2}。

返回格式

查询命令返回 1 或 2。

实例说明

:TRIGger:ISO7816:CRCLen 1　　设置 ISO7816 协议触发的 CRC 长度为 1。

:TRIGger:ISO7816:CRCLen? -> 1　　查询 ISO7816 协议触发的 CRC 长度为 1。

:TRIGger:ISO7816:TRIGMode

命令格式

:TRIGger:ISO7816:TRIGMode <String>

:TRIGger:ISO7816:TRIGMode?

功能描述

设置 ISO7816 协议触发的触发模式。

查询 ISO7816 协议触发的触发模式。

参数说明

<String> 范围 { RST | TS }。

返回格式

查询命令返回 RST 或 TS。

实例说明

:TRIGger:ISO7816:TRIGMode TS	设置 ISO7816 协议触发的触发模式为 TS。
:TRIGger:ISO7816:TRIGMode? -> TS	查询 ISO7816 协议触发的触发模式为 TS。

:TRIGger:LIN:SOURce

命令格式

:TRIGger:LIN:SOURce <String>

:TRIGger:LIN:SOURce?

功能描述

设置 LIN 协议触发的信源。

查询 LIN 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:LIN:SOURce CHANnel1　　设置 LIN 协议触发信源为通道 1。

:TRIGger:LIN:SOURce? -> CHANnel1　　查询 LIN 协议触发信源为通道 1。

:TRIGger:LIN:BAUDrate

命令格式

:TRIGger:LIN:BAUDrate <Int>

:TRIGger:LIN:BAUDrate?

功能描述

设置 LIN 协议触发的波特率。

查询 LIN 协议触发的波特率。

参数说明

<Int> 波特率，范围 1000~20000。

返回格式

查询命令以整型形式返回 LIN 协议触发的波特率大小。

实例说明

:TRIGger:LIN:BAUDrate 9600	设置 LIN 协议触发的波特率为 9600。
:TRIGger:LIN:BAUDrate? -> 9600	查询 LIN 协议触发的波特率为 9600。

:TRIGger:LIN:TRIGMode

命令格式

:TRIGger:LIN:TRIGMode <String>

:TRIGger:LIN:TRIGMode?

功能描述

设置 LIN 协议触发的触发模式。

查询 LIN 协议触发的触发模式。

参数说明

<String> 范围 { BREAk | SYNCh | IDFIeld | DATA }，分别表示同步间隔、同步场、ID 场和数据序列。

返回格式

查询命令返回 BREAk、SYNCh、IDFIeld 或 DATA。

实例说明

:TRIGger:LIN:TRIGMode BREAk　　设置 LIN 协议触发的触发模式为同步间隔。

:TRIGger:LIN:TRIGMode? -> BREAk　　查询 LIN 协议触发的触发模式为同步间隔。

:TRIGger:LIN:TRIGId

命令格式

:TRIGger:LIN:TRIGId <Int>

:TRIGger:LIN:TRIGId?

功能描述

设置 LIN 协议触发的触发 ID 场值。

查询 LIN 协议触发的触发 ID 场值。

注：触发模式设置为 **ID 场**和**数据序列**时有效，相关指令:TRIGger:LIN:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 LIN 协议触发的触发 ID 场值。

实例说明

:TRIGger:LIN:TRIGId 0x10　　设置 LIN 协议触发的触发 ID 场值为 0x10。

:TRIGger:LIN:TRIGId? -> 0x10　　查询 LIN 协议触发的触发 ID 场值为 0x10。

:TRIGger:LIN:DATACnt

命令格式

:TRIGger:LIN:DATACnt <Int>

:TRIGger:LIN:DATACnt?

功能描述

设置 LIN 协议触发的数据个数。

查询 LIN 协议触发的数据个数。

注：触发模式设置为**数据序列**时有效，相关指令:TRIGger:LIN:TRIGMode。

参数说明

<Int> 范围 1~8。

返回格式

查询命令以整型形式返回 LIN 协议触发的数据个数大小。

实例说明

:TRIGger:LIN:DATACnt 2　　设置 LIN 协议触发的数据个数为 2。

:TRIGger:LIN:DATACnt? -> 2　　查询 LIN 协议触发的数据个数为 2。

:TRIGger:LIN:DATAIndex

命令格式

:TRIGger:LIN:DATAIndex <Int>

:TRIGger:LIN:DATAIndex?

功能描述

设置 LIN 协议触发的数据索引。

查询 LIN 协议触发的数据索引。

注：触发模式设置为**数据序列**时有效，相关指令:TRIGger:LIN:TRIGMode。

参数说明

<Int> 范围 0~7。

注：数据索引范围根据数据个数变化。

返回格式

查询命令以整数形式返回 LIN 协议触发的数据索引。

实例说明

:TRIGger:LIN:DATAIndex 1	设置 LIN 协议触发的数据索引为 1。
:TRIGger:LIN:DATAIndex? -> 1	查询 LIN 协议触发的数据索引为 1。

:TRIGger:LIN:TRIGData

命令格式

:TRIGger:LIN:TRIGData <Int>

:TRIGger:LIN:TRIGData?

功能描述

设置 LIN 协议触发的触发数据。

查询 LIN 协议触发的触发数据。

注：触发模式设置为**数据序列**时有效，相关指令:TRIGger:LIN:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 LIN 协议触发的触发数据。

实例说明

:TRIGger:LIN:TRIGData 0x10　　设置 LIN 协议触发的触发数据为 0x10。

:TRIGger:LIN:TRIGData? -> 0x10　　查询 LIN 协议触发的触发数据为 0x10。

:TRIGger:MANChester:SOURce

命令格式

:TRIGger:MANChester:SOURce <String>

:TRIGger:MANChester:SOURce?

功能描述

设置 Manchester 协议触发的信源。

查询 Manchester 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:MANChester:SOURce CHANnel1 设置 Manchester 协议触发信源为通道 1。

:TRIGger:MANChester:SOURce? -> CHANnel1 查询 Manchester 协议触发信源为通道 1。

:TRIGger:MANChester:TYPE

命令格式

:TRIGger:MANChester:TYPE <String>

:TRIGger:MANChester:TYPE?

功能描述

设置 Manchester 协议触发的编码模式。

查询 Manchester 协议触发的编码模式。

参数说明

<String> 范围 { GE | IEEE}。

返回格式

查询命令返回 GE 或 IEEE。

实例说明

:TRIGger:MANChester:TYPE IEEE 设置 Manchester 协议触发的编码模式为 IEEE。

:TRIGger:MANChester:TYPE? -> IEEE 查询 Manchester 协议触发的编码模式为 IEEE。

:TRIGger:MANChester:BITLen

命令格式

:TRIGger:MANChester:BITLen <String>

:TRIGger:MANChester:BITLen?

功能描述

设置 Manchester 协议触发的位时长。

查询 Manchester 协议触发的位时长。

参数说明

<String> 范围 100ns~1ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 Manchester 协议触发的位时长。

实例说明

:TRIGger:MANChester:BITLen 2e-4 设置 Manchester 协议触发的位时长为 200us。

:TRIGger:MANChester:BITLen? -> 1.000ms 查询 Manchester 协议触发的位时长为 1ms。

:TRIGger:MANChester:FRAMestart

命令格式

:TRIGger:MANChester:FRAMestart <String>

:TRIGger:MANChester:FRAMestart?

功能描述

设置 Manchester 协议触发的包起始位。

查询 Manchester 协议触发的包起始位。

参数说明

<String> 范围 { 0 | 1}。

返回格式

查询命令返回 0 或 1。

实例说明

:TRIGger:MANChester:FRAMestart 1　　设置 Manchester 协议触发的包起始位为 1。

:TRIGger:MANChester:FRAMestart? -> 1　　查询 Manchester 协议触发的包起始位为 1。

:TRIGger:MANChester:TRIGMode

命令格式

:TRIGger:MANChester:TRIGMode <String>

:TRIGger:MANChester:TRIGMode?

功能描述

设置 Manchester 协议触发的触发模式。

查询 Manchester 协议触发的触发模式。

参数说明

<String> 范围 { BEGIn }，表示包起始位。

返回格式

查询命令返回 BEGIn。

实例说明

:TRIGger:MANChester:TRIGMode BEGIn 设置 Manchester 协议触发的触发模式为包起始位。

:TRIGger:MANChester:TRIGMode? -> BEGIn 查询 Manchester 协议触发的触发模式为包起始位。

:TRIGger:MDIO:MDC

命令格式

:TRIGger:MDIO:MDC <String>

:TRIGger:MDIO:MDC?

功能描述

设置 MDIO 协议触发的 MDC 的信源。

查询 MDIO 协议触发的 MDC 的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:MDIO:MDC CHANnel1　　设置 MDIO 协议触发的 MDC 信源为通道 1。

:TRIGger:MDIO:MDC? -> CHANnel1　　查询 MDIO 协议触发的 MDC 信源为通道 1。

:TRIGger:MDIO:MDIO

命令格式

:TRIGger:MDIO:MDIO <String>

:TRIGger:MDIO:MDIO?

功能描述

设置 MDIO 协议触发的 MDIO 的信源。

查询 MDIO 协议触发的 MDIO 的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:MDIO:MDIO CHANnel1　　设置 MDIO 协议触发的 MDIO 信源为通道 1。

:TRIGger:MDIO:MDIO? -> CHANnel1　　查询 MDIO 协议触发的 MDIO 信源为通道 1。

:TRIGger:MDIO:CLAUse

命令格式

:TRIGger:MDIO:CLAUse <String>

:TRIGger:MDIO:CLAUse?

功能描述

设置 MDIO 协议触发的条款。

查询 MDIO 协议触发的条款。

参数说明

<String> 范围 { CLAUse22 | CLAUse45}。

返回格式

查询命令返回 CLAUse22 或 CLAUse45。

实例说明

:TRIGger:MDIO:CLAUse CLAUse45　　设置 MDIO 协议触发的条款为 CLAUse45。

:TRIGger:MDIO:CLAUse? -> CLAUse45　查询 MDIO 协议触发的条款为 CLAUse45。

:TRIGger:MDIO:TRIGMode

命令格式

:TRIGger:MDIO:TRIGMode <String>

:TRIGger:MDIO:TRIGMode?

功能描述

设置 MDIO 协议触发的触发模式。

查询 MDIO 协议触发的触发模式。

参数说明

<String> 范围 { ST | OP | PHYAd | REGDev | DATA }。

返回格式

查询命令返回 ST、OP、PHYAd、REGDev 或 DATA。

实例说明

:TRIGger:MDIO:TRIGMode ST	设置 MDIO 协议触发的触发模式为 ST。
:TRIGger:MDIO:TRIGMode? -> ST	查询 MDIO 协议触发的触发模式为 ST。

:TRIGger:MDIO:OP22

命令格式

:TRIGger:MDIO:OP22 <String>

:TRIGger:MDIO:OP22?

功能描述

设置 MDIO 协议触发的条款 22 下的 OP 类型。

查询 MDIO 协议触发的条款 22 下的 OP 类型。

注 1：条款设置为 **CLause22** 时有效，相关指令:TRIGger:MDIO:CLAUse。

注 2：触发模式设置为 **OP**、**PHYAD**、**REG/DEV** 和 **DATA** 时有效，相关指令:TRIGger:MDIO:TRIGMode。

参数说明

<String> 范围 { WRITe | READ }。

返回格式

查询命令返回 WRITe 或 READ。

实例说明

:TRIGger:MDIO:OP22 WRITe　　设置 MDIO 协议触发的条款 22 下的 OP 类型为 WRITe。

:TRIGger:MDIO:OP22? -> WRITe　　查询 MDIO 协议触发的条款 22 下的 OP 类型为 WRITe。

:TRIGger:MDIO:OP45

命令格式

:TRIGger:MDIO:OP45 <String>

:TRIGger:MDIO:OP45?

功能描述

设置 MDIO 协议触发的条款 45 下的 OP 类型。

查询 MDIO 协议触发的条款 45 下的 OP 类型。

注 1：条款设置为 **CLause45** 时有效，相关指令:TRIGger:MDIO:CLAUse。

注 2：触发模式设置为 **OP**、**PHYAD**、**REG/DEV** 和 **DATA** 时有效，相关指令:TRIGger:MDIO:TRIGMode。

参数说明

<String> 范围 { ADDRess | WRITe | READ | POSTread }。

返回格式

查询命令返回 ADDRess、WRITe、READ 或 POSTread。

实例说明

:TRIGger:MDIO:OP45 WRITe 设置 MDIO 协议触发的条款 45 下的 OP 类型为 WRITe。

:TRIGger:MDIO:OP45? -> WRITe 查询 MDIO 协议触发的条款 45 下的 OP 类型为 WRITe。

:TRIGger:MDIO:PHYAd

命令格式

:TRIGger:MDIO:PHYAd <Int>

:TRIGger:MDIO:PHYAd?

功能描述

设置 MDIO 协议触发的芯片地址。

查询 MDIO 协议触发的芯片地址。

注：触发模式设置为 **PHYAD**、**REG/DEV** 和 **DATA** 时有效，相关指令:TRIGger:MDIO:TRIGMode。

参数说明

<Int> 范围 0x00~0x1F。

返回格式

查询命令以十六进制形式返回 MDIO 协议触发的芯片地址。

实例说明

:TRIGger:MDIO:PHYAd 0x10	设置 MDIO 协议触发的芯片地址为 0x10。
:TRIGger:MDIO:PHYAd? -> 0x10	查询 MDIO 协议触发的芯片地址为 0x10。

:TRIGger:MDIO:REGAd

命令格式

:TRIGger:MDIO:REGAd <Int>

:TRIGger:MDIO:REGAd?

功能描述

设置 MDIO 协议触发的寄存器地址。

查询 MDIO 协议触发的寄存器地址。

注 1：条款设置为 **CLause22** 时有效，相关指令:TRIGger:MDIO:CLAUse。

注 2：触发模式设置为 **REG/DEV** 和 **DATA** 时有效，相关指令:TRIGger:MDIO:TRIGMode。

参数说明

<Int> 范围 0x00~0x1F。

返回格式

查询命令以十六进制形式返回 MDIO 协议触发的寄存器地址。

实例说明

:TRIGger:MDIO:REGAd 0x10	设置 MDIO 协议触发的寄存器地址为 0x10。
:TRIGger:MDIO:REGAd? -> 0x10	查询 MDIO 协议触发的寄存器地址为 0x10。

:TRIGger:MDIO:DEVType

命令格式

:TRIGger:MDIO:DEVType <String>

:TRIGger:MDIO:DEVType?

功能描述

设置 MDIO 协议触发的设备类型。

查询 MDIO 协议触发的设备类型。

注 1：条款设置为 **CLause45** 时有效，相关指令:TRIGger:MDIO:CLAUse。

注 2：触发模式设置为 **REG/DEV** 和 **DATA** 时有效，相关指令:TRIGger:MDIO:TRIGMode。

参数说明

<String> 范围 { RESErved | PMDPma | WIS | PCS | PHYXs | DTEXs }。

返回格式

查询命令返回 RESErved、PMDPma、WIS、PCS、PHYXs 或 DTEXs。

实例说明

:TRIGger:MDIO:DEVType RESErved 设置 MDIO 协议触发的设备类型为 RESErved。

:TRIGger:MDIO:DEVType? -> RESErved 查询 MDIO 协议触发的设备类型为 RESErved。

:TRIGger:MDIO:DATA

命令格式

:TRIGger:MDIO:DATA <Int>

:TRIGger:MDIO:DATA?

功能描述

设置 MDIO 协议触发的触发数据。

查询 MDIO 协议触发的触发数据。

注：触发模式设置为 **DATA** 时有效，相关指令:TRIGger:MDIO:TRIGMode。

参数说明

<Int> 范围 0x00~0xFFFF。

返回格式

查询命令以十六进制形式返回 MDIO 协议触发的触发数据。

实例说明

:TRIGger:MDIO:DATA 0x10　　设置 MDIO 协议触发的触发数据为 0x10。

:TRIGger:MDIO:DATA? -> 0x10　　查询 MDIO 协议触发的触发数据为 0x10。

:TRIGger:MILLer:SOURce

命令格式

:TRIGger:MILLer:SOURce <String>

:TRIGger:MILLer:SOURce?

功能描述

设置 Miller 协议触发的信源。

查询 Miller 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:MILLer:SOURce CHANnel1	设置 Miller 协议触发信源为通道 1。
:TRIGger:MILLer:SOURce? -> CHANnel1	查询 Miller 协议触发信源为通道 1。

:TRIGger:MILLer:BITRate

命令格式

:TRIGger:MILLer:BITRate <Double>

:TRIGger:MILLer:BITRate?

功能描述

设置 Miller 协议触发的比特率。

查询 Miller 协议触发的比特率。

参数说明

<Double> 比特率，范围 0.10k~1000000.00k。

返回格式

查询命令以实型形式返回 Miller 协议触发的比特率大小。

实例说明

:TRIGger:MILLer:BITRate 15	设置 Miller 协议触发的比特率为 15k。
:TRIGger:MILLer:BITRate? -> 15.00	查询 Miller 协议触发的比特率为 15k。

:TRIGger:MILLer:IDLELevel

命令格式

:TRIGger:MILLer:IDLELevel <String>

:TRIGger:MILLer:IDLELevel?

功能描述

设置 Miller 协议触发的空闲电平。

查询 Miller 协议触发的空闲电平。

参数说明

<String> 范围 { LOW | HIGH }，分别表示低电平和高电平。

返回格式

查询命令返回 LOW 或 HIGH。

实例说明

:TRIGger:MILLer:IDLELevel LOW　　设置 Miller 协议触发的空闲电平为低电平。

:TRIGger:MILLer:IDLELevel? -> LOW　　查询 Miller 协议触发的空闲电平为低电平。

:TRIGger:MILLer:TRIGMode

命令格式

:TRIGger:MILLer:TRIGMode <String>

:TRIGger:MILLer:TRIGMode?

功能描述

设置 Miller 协议触发的触发模式。

查询 Miller 协议触发的触发模式。

参数说明

<String> 范围 { IDLE }，表示空闲电平。

返回格式

查询命令返回 IDLE。

实例说明

:TRIGger:MILLer:TRIGMode IDLE　　设置 Miller 协议触发的触发模式为空闲电平。

:TRIGger:MILLer:TRIGMode? -> IDLE　　查询 Miller 协议触发的触发模式为空闲电平。

:TRIGger:MODBus:SOURce

命令格式

:TRIGger:MODBus:SOURce <String>

:TRIGger:MODBus:SOURce?

功能描述

设置 Modbus 协议触发的数据信源。

查询 Modbus 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:MODBus:SOURce CHANnel1　　设置 Modbus 协议触发数据信源为通道 1。

:TRIGger:MODBus:SOURce? -> CHANnel1　　查询 Modbus 协议触发数据信源为通道 1。

:TRIGger:MODBus:BAUDrate

命令格式

:TRIGger:MODBus:BAUDrate <Int>

:TRIGger:MODBus:BAUDrate?

功能描述

设置 Modbus 协议触发的波特率。

查询 Modbus 协议触发的波特率。

参数说明

<Int> 波特率，范围 0~20000000。

返回格式

查询命令以整型形式返回 Modbus 协议触发的波特率大小。

实例说明

:TRIGger:MODBus:BAUDrate 9600	设置 Modbus 协议触发的波特率为 9600。
:TRIGger:MODBus:BAUDrate? -> 9600	查询 Modbus 协议触发的波特率为 9600。

:TRIGger:MODBus:PARItymode

命令格式

:TRIGger:MODBus:PARItymode <String>

:TRIGger:MODBus:PARItymode?

功能描述

设置 Modbus 协议触发的校验位。

查询 Modbus 协议触发的校验位。

参数说明

<String> 范围 { ODD | EVEN | NONE }，分别表示奇校验、偶校验、无。

返回格式

查询命令返回 ODD，EVEN 或 NONE。

实例说明

:TRIGger:MODBus:PARItymode NONE　　设置 Modbus 协议触发的校验位为 NONE。

:TRIGger:MODBus:PARItymode? -> NONE　　查询 Modbus 协议触发的校验位为 NONE。

:TRIGger:MODBus:REVErse

命令格式

:TRIGger:MODBus:REVErse <String>

:TRIGger:MODBus:REVErse?

功能描述

设置 Modbus 协议触发的电平反相。

查询 Modbus 协议触发的电平反相。

参数说明

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:TRIGger:MODBus:REVErse FALSe　　设置 Modbus 协议触发的电平反相为 FALSe。

:TRIGger:MODBus:REVErse? -> FALSe　　查询 Modbus 协议触发的电平反相为 FALSe。

:TRIGger:MODBus:TRANmode

命令格式

:TRIGger:MODBus:TRANmode <String>

:TRIGger:MODBus:TRANmode?

功能描述

设置 Modbus 协议触发的传输模式。

查询 Modbus 协议触发的传输模式。

参数说明

<String> 范围 { RTU | ASCIi }。

返回格式

查询命令返回 RTU 或 ASCIi。

实例说明

:TRIGger:MODBus:TRANmode RTU　　设置 Modbus 协议触发的传输模式为 RTU。

:TRIGger:MODBus:TRANmode? -> RTU　　查询 Modbus 协议触发的传输模式为 RTU。

:TRIGger:MODBus:TRIGMode

命令格式

:TRIGger:MODBus:TRIGMode <String>

:TRIGger:MODBus:TRIGMode?

功能描述

设置 MODBus 协议触发的触发模式。

查询 MODBus 协议触发的触发模式。

参数说明

<String> 范围 { BEGIn | ADDR | ADDR+CMD }，分别表示开始触发、地址触发和地址+命令。

返回格式

查询命令返回 BEGIn、ADDR 或 ADDR+CMD。

实例说明

:TRIGger:MODBus:TRIGMode BEGIn 设置 MODBus 协议触发的触发模式为开始触发。

:TRIGger:MODBus:TRIGMode? -> BEGIn 查询 MODBus 协议触发的触发模式为开始触发。

:TRIGger:MODBus:TRIGCmd

命令格式

:TRIGger:MODBus:TRIGCmd <Int>

:TRIGger:MODBus:TRIGCmd?

功能描述

设置 MODBus 协议触发的触发命令。

查询 MODBus 协议触发的触发命令。

注：触发模式设置为**地址+命令**时有效，相关指令:TRIGger:MODBus:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MODBus 协议触发的触发命令。

实例说明

:TRIGger:MODBus:TRIGCmd 0x10　　设置 MODBus 协议触发的触发命令为 0x10。

:TRIGger:MODBus:TRIGCmd? -> 0x10　　查询 MODBus 协议触发的触发命令为 0x10。

:TRIGger:MODBus:TRIGAddr

命令格式

:TRIGger:MODBus:TRIGAddr <Int>

:TRIGger:MODBus:TRIGAddr?

功能描述

设置 MODBus 协议触发的触发地址。

查询 MODBus 协议触发的触发地址。

注：触发模式设置为**地址触发**和**地址+命令**时有效，相关指令:TRIGger:MODBus:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MODBus 协议触发的触发地址。

实例说明

:TRIGger:MODBus:TRIGAddr 0x10　　设置 MODBus 协议触发的触发地址为 0x10。

:TRIGger:MODBus:TRIGAddr? -> 0x10　　查询 MODBus 协议触发的触发地址为 0x10。

:TRIGger:MVB:SOURce

命令格式

:TRIGger:MVB:SOURce <String>

:TRIGger:MVB:SOURce?

功能描述

设置 MVB 协议触发的数据信源。

查询 MVB 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:MVB:SOURce CHANnel1　　设置 MVB 协议触发数据信源为通道 1。

:TRIGger:MVB:SOURce? -> CHANnel1　　查询 MVB 协议触发数据信源为通道 1。

:TRIGger:MVB:BAUDrate

命令格式

:TRIGger:MVB:BAUDrate <Double>

:TRIGger:MVB:BAUDrate?

功能描述

设置 MVB 协议触发的波特率。

查询 MVB 协议触发的波特率。

参数说明

<Double> 波特率，范围 0.00M~8.00M。

返回格式

查询命令以实型形式返回 MVB 协议触发的波特率大小。

实例说明

:TRIGger:MVB:BAUDrate 2　　设置 MVB 协议触发的波特率为 2M。

:TRIGger:MVB:BAUDrate? -> 2.00　　查询 MVB 协议触发的波特率为 2M。

:TRIGger:MVB:MEDIum

命令格式

:TRIGger:MVB:MEDIum <String>

:TRIGger:MVB:MEDIum?

功能描述

设置 MVB 协议触发的介质。

查询 MVB 协议触发的介质。

参数说明

<String> 范围 { ESD | EMD | OGF }。

返回格式

查询命令返回 ESD、EMD 或 OGF。

实例说明

:TRIGger:MVB:MEDIum ESD	设置 MVB 协议触发的介质为 ESD。
:TRIGger:MVB:MEDIum? -> ESD	查询 MVB 协议触发的介质为 ESD。

:TRIGger:MVB:TRIGMode

命令格式

:TRIGger:MVB:TRIGMode <String>

:TRIGger:MVB:TRIGMode?

功能描述

设置 MVB 协议触发的触发模式。

查询 MVB 协议触发的触发模式。

参数说明

<String> 范围 { MSD | SSD | FCODe | ADDR }。

返回格式

查询命令返回 MSD、SSD、FCODe 或 ADDR。

实例说明

:TRIGger:MVB:TRIGMode MSD	设置 MVB 协议触发的触发模式为 MSD。
:TRIGger:MVB:TRIGMode? -> MSD	查询 MVB 协议触发的触发模式为 MSD。

:TRIGger:MVB:IDLELevel

命令格式

:TRIGger:MVB:IDLELevel <String>

:TRIGger:MVB:IDLELevel?

功能描述

设置 MVB 协议触发的空闲电平。

查询 MVB 协议触发的空闲电平。

参数说明

<String> 范围 { LOW | HIGH }，分别表示低电平和高电平。

返回格式

查询命令返回 LOW 或 HIGH。

实例说明

:TRIGger:MVB:IDLELevel LOW	设置 MVB 协议触发的空闲电平为低电平。
:TRIGger:MVB:IDLELevel? -> LOW	查询 MVB 协议触发的空闲电平为低电平。

:TRIGger:MVB:FCODe

命令格式

:TRIGger:MVB:FCODe <Int>

:TRIGger:MVB:FCODe?

功能描述

设置 MVB 协议触发的 F 码。

查询 MVB 协议触发的 F 码。

注：触发模式设置为 **FCode** 和 **Addr** 时有效，相关指令:TRIGger:MVB:TRIGMode。

参数说明

<Int> 范围 0x00~0x0F。

返回格式

查询命令以十六进制形式返回 MVB 协议触发的 F 码。

实例说明

:TRIGger:MVB:FCODe 0x10	设置 MVB 协议触发的 F 码为 0x10。
:TRIGger:MVB:FCODe? -> 0x10	查询 MVB 协议触发的 F 码为 0x10。

:TRIGger:MVB:ADDR

命令格式

:TRIGger:MVB:ADDR <Int>

:TRIGger:MVB:ADDR?

功能描述

设置 MVB 协议触发的地址。

查询 MVB 协议触发的地址。

注：触发模式设置为 **Addr** 时有效，相关指令:TRIGger:MVB:TRIGMode。

参数说明

<Int> 范围 0x00~0xFFF。

返回格式

查询命令以十六进制形式返回 MVB 协议触发的地址。

实例说明

:TRIGger:MVB:ADDR 0x10　　设置 MVB 协议触发的地址为 0x10。

:TRIGger:MVB:ADDR? -> 0x10　　查询 MVB 协议触发的地址为 0x10。

:TRIGger:PS2:CLK

命令格式

:TRIGger:PS2:CLK <String>

:TRIGger:PS2:CLK?

功能描述

设置 PS/2 协议触发的时钟信源。

查询 PS/2 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:PS2:CLK CHANnel1	设置 PS/2 协议触发的时钟信源为通道 1。
:TRIGger:PS2:CLK? -> CHANnel1	查询 PS/2 协议触发的时钟信源为通道 1。

:TRIGger:PS2:DAT

命令格式

:TRIGger:PS2:DAT <String>

:TRIGger:PS2:DAT?

功能描述

设置 PS/2 协议触发的数据信源。

查询 PS/2 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:PS2:DAT CHANnel1	设置 PS/2 协议触发的数据信源为通道 1。
:TRIGger:PS2:DAT? -> CHANnel1	查询 PS/2 协议触发的数据信源为通道 1。

:TRIGger:PS2:TRIGMode

命令格式

:TRIGger:PS2:TRIGMode <String>

:TRIGger:PS2:TRIGMode?

功能描述

设置 PS/2 协议触发的触发模式。

查询 PS/2 协议触发的触发模式。

参数说明

<String> 范围 { BEGIn | DATA }，分别表示起始位和数据。

返回格式

查询命令返回 BEGIn 或 DATA。

实例说明

:TRIGger:PS2:TRIGMode BEGIn　　设置 PS/2 协议触发的触发模式为起始位。

:TRIGger:PS2:TRIGMode? -> BEGIn　　查询 PS/2 协议触发的触发模式为起始位。

:TRIGger:PS2:TRIGDIrect

命令格式

:TRIGger:PS2:TRIGDIrect <String>

:TRIGger:PS2:TRIGDIrect?

功能描述

设置 PS/2 协议触发的数据传输方向。

查询 PS/2 协议触发的数据传输方向。

参数说明

<String> 范围 { TOHOst | TODEv }，分别表示发向主机和发向从机。

返回格式

查询命令返回 TOHOst 或 TODEv。

实例说明

:TRIGger:PS2:TRIGDIrect TOHOst 设置 PS/2 协议触发的数据传输方向为发向主机。

:TRIGger:PS2:TRIGDIrect? -> TOHOst 查询 PS/2 协议触发的数据传输方向为发向主机。

:TRIGger:PS2:TRIGDAta

命令格式

:TRIGger:PS2:TRIGDAta <Int>

:TRIGger:PS2:TRIGDAta?

功能描述

设置 PS/2 协议触发的触发数据。

查询 PS/2 协议触发的触发数据。

注：触发模式设置为**数据**时有效，相关指令:TRIGger:PS2:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 PS/2 协议触发的触发数据。

实例说明

:TRIGger:PS2:TRIGDAta 0x10	设置 PS/2 协议触发的触发数据为 0x10。
:TRIGger:PS2:TRIGDAta? -> 0x10	查询 PS/2 协议触发的触发数据为 0x10。

:TRIGger:RFFE:SCL

命令格式

:TRIGger:RFFE:SCL <String>

:TRIGger:RFFE:SCL?

功能描述

设置 MIPI-RFFE 协议触发的时钟信源。

查询 MIPI-RFFE 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:RFFE:SCL CHANnel1　　设置 MIPI-RFFE 协议触发的时钟信源为通道 1。

:TRIGger:RFFE:SCL? -> CHANnel1　查询 MIPI-RFFE 协议触发的时钟信源为通道 1。

:TRIGger:RFFE:SDA

命令格式

:TRIGger:RFFE:SDA <String>

:TRIGger:RFFE:SDA?

功能描述

设置 MIPI-RFFE 协议触发的数据信源。

查询 MIPI-RFFE 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:RFFE:SDA CHANnel1　　设置 MIPI-RFFE 协议触发的数据信源为通道 1。

:TRIGger:RFFE:SDA? -> CHANnel1　查询 MIPI-RFFE 协议触发的数据信源为通道 1。

:TRIGger:RFFE:CLOCk

命令格式

:TRIGger:RFFE:CLOCk <Double>

:TRIGger:RFFE:CLOCk?

功能描述

设置 MIPI-RFFE 协议触发的时钟频率。

查询 MIPI-RFFE 协议触发的时钟频率。

参数说明

<Double> 阈值，范围 32.00K~52000.00K，单位 Hz。

返回格式

查询命令以实型形式返回 MIPI-RFFE 协议触发的时钟频率大小。

实例说明

:TRIGger:RFFE:CLOCk 36　　设置 MIPI-RFFE 协议触发的时钟频率为 36KHz。

:TRIGger:RFFE:CLOCk? -> 36.00　　查询 MIPI-RFFE 协议触发的时钟频率为 36KHz。

:TRIGger:RFFE:TRIGMode

命令格式

:TRIGger:RFFE:TRIGMode <String>

:TRIGger:RFFE:TRIGMode?

功能描述

设置 MIPI-RFFE 协议触发的触发模式。

查询 MIPI-RFFE 协议触发的触发模式。

参数说明

<String> 范围 { BEGIn | SLAVeaddr | CMD | DATA }，分别表示起始触发、从机地址、指令触发和数据触发。

返回格式

查询命令返回 BEGIn、SLAVeaddr，CMD 或 DATA。

实例说明

:TRIGger:RFFE:TRIGMode BEGIn 设置 MIPI-RFFE 协议触发的触发模式为起始触发。

:TRIGger:RFFE:TRIGMode? -> BEGIn 查询 MIPI-RFFE 协议触发的触发模式为起始触发。

:TRIGger:RFFE:SLAVeaddr

命令格式

:TRIGger:RFFE:SLAVeaddr <Int>

:TRIGger:RFFE:SLAVeaddr?

功能描述

设置 MIPI-RFFE 协议触发的设备地址。

查询 MIPI-RFFE 协议触发的设备地址。

注：触发模式设置为**从机地址**、**指令触发**和**数据触发**时有效，相关指令:TRIGger:RFFE:TRIGMode。

参数说明

<Int> 范围 0x00~0x0F。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的设备地址。

实例说明

:TRIGger:RFFE:SLAVeaddr 0x10　　设置 MIPI-RFFE 协议触发的设备地址为 0x10。

:TRIGger:RFFE:SLAVeaddr? -> 0x10　　查询 MIPI-RFFE 协议触发的设备地址为 0x10。

:TRIGger:RFFE:TRIGCmd

命令格式

:TRIGger:RFFE:TRIGCmd <String>

:TRIGger:RFFE:TRIGCmd?

功能描述

设置 MIPI-RFFE 协议触发的指令触发指令。

查询 MIPI-RFFE 协议触发的指令触发指令。

注：触发模式设置为**指令触发**时有效，相关指令:TRIGger:RFFE:TRIGMode。

参数说明

<String> 范围 { REG0W | REGWrite | REGRead | EXTWrite | EXTRead | EXTWL | EXTRL | MSTRead | MSTWrite | MOH | ISID | CUSTom }，分别表示 Reg0 write、Reg write、Reg read、ExtReg W、ExtReg R、ExtReg WL、ExtReg RL、Master R、Master W、MOH、ISID 和自定义。

返回格式

查询命令返回 REG0W、REGWrite、REGRead、EXTWrite、EXTRead、EXTWL、EXTRL、MSTRead、MSTWrite、MOH、ISID 或 CUSTom。

实例说明

:TRIGger:RFFE:TRIGCmd REG0W 设置 MIPI-RFFE 协议触发的指令触发指令为 Reg0 write。

:TRIGger:RFFE:TRIGCmd? -> REG0W 查询 MIPI-RFFE 协议触发的指令触发指令为 Reg0 write。

:TRIGger:RFFE:CMDVal

命令格式

:TRIGger:RFFE:CMDVal <Int>

:TRIGger:RFFE:CMDVal?

功能描述

设置 MIPI-RFFE 协议触发的指令值。

查询 MIPI-RFFE 协议触发的指令值。

注：指令触发指令设置为**自定义**时有效，相关指令:TRIGger:RFFE:TRIGCmd。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的指令值。

实例说明

:TRIGger:RFFE:CMDVal 0x10	设置 MIPI-RFFE 协议触发的指令值为 0x10。
:TRIGger:RFFE:CMDVal? -> 0x10	查询 MIPI-RFFE 协议触发的指令值为 0x10。

:TRIGger:RFFE:CMDMask

命令格式

:TRIGger:RFFE:CMDMask <Int>

:TRIGger:RFFE:CMDMask?

功能描述

设置 MIPI-RFFE 协议触发的指令掩码。

查询 MIPI-RFFE 协议触发的指令掩码。

注：指令触发指令设置为**自定义**时有效，相关指令:TRIGger:RFFE:TRIGCmd。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的指令掩码。

实例说明

:TRIGger:RFFE:CMDMask 0x10　　设置 MIPI-RFFE 协议触发的指令掩码为 0x10。

:TRIGger:RFFE:CMDMask? -> 0x10　　查询 MIPI-RFFE 协议触发的指令掩码为 0x10。

:TRIGger:RFFE:DATACmd

命令格式

:TRIGger:RFFE:DATACmd <String>

:TRIGger:RFFE:DATACmd?

功能描述

设置 MIPI-RFFE 协议触发的数据触发指令。

查询 MIPI-RFFE 协议触发的数据触发指令。

注：触发模式设置为**数据触发**时有效，相关指令:TRIGger:RFFE:TRIGMode。

参数说明

<String> 范围 { REG0W | REGWrite | REGRead }，分别表示 Reg0 write、Reg write 和 Reg read。

返回格式

查询命令返回 REG0W、REGWrite 或 REGRead。

实例说明

:TRIGger:RFFE:DATACmd REG0W 设置 MIPI-RFFE 协议触发的数据触发指令为 Reg0 write。

:TRIGger:RFFE:DATACmd? -> REG0W 查询 MIPI-RFFE 协议触发的数据触发指令为 Reg0 write。

:TRIGger:RFFE:REGData

命令格式

:TRIGger:RFFE:REGData <Int>

:TRIGger:RFFE:REGData?

功能描述

设置 MIPI-RFFE 协议触发的数据。

查询 MIPI-RFFE 协议触发的数据。

注：触发模式设置为**数据触发**时有效，相关指令:TRIGger:RFFE:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的数据。

实例说明

:TRIGger:RFFE:REGData 0x10	设置 MIPI-RFFE 协议触发的数据为 0x10。
:TRIGger:RFFE:REGData? -> 0x10	查询 MIPI-RFFE 协议触发的数据为 0x10。

:TRIGger:RFFE:REG0Mask

命令格式

:TRIGger:RFFE:REG0Mask <Int>

:TRIGger:RFFE:REG0Mask?

功能描述

设置 MIPI-RFFE 协议触发的 Reg0 数据掩码。

查询 MIPI-RFFE 协议触发的 Reg0 数据掩码。

注：数据触发指令设置为 **Reg0 write** 时有效，相关指令:TRIGger:RFFE:DATACmd。

参数说明

<Int> 范围 0x00~0x7F。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的 Reg0 数据掩码。

实例说明

:TRIGger:RFFE:REG0Mask 0x10　　设置 MIPI-RFFE 协议触发的 Reg0 数据掩码为 0x10。

:TRIGger:RFFE:REG0Mask? -> 0x10　　查询 MIPI-RFFE 协议触发的 Reg0 数据掩码为 0x10。

:TRIGger:RFFE:REGAddr

命令格式

:TRIGger:RFFE:REGAddr <Int>

:TRIGger:RFFE:REGAddr?

功能描述

设置 MIPI-RFFE 协议触发的寄存器地址。

查询 MIPI-RFFE 协议触发的寄存器地址。

注：数据触发指令设置为 **Reg write** 和 **Reg read** 时有效，相关指令:TRIGger:RFFE:DATACmd。

参数说明

<Int> 范围 0x00~0x1F。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的寄存器地址。

实例说明

:TRIGger:RFFE:REGAddr 0x10　　设置 MIPI-RFFE 协议触发的寄存器地址为 0x10。

:TRIGger:RFFE:REGAddr? -> 0x10　　查询 MIPI-RFFE 协议触发的寄存器地址为 0x10。

:TRIGger:RFFE:ADDRMask

命令格式

:TRIGger:RFFE:ADDRMask <Int>

:TRIGger:RFFE:ADDRMask?

功能描述

设置 MIPI-RFFE 协议触发的地址掩码。

查询 MIPI-RFFE 协议触发的地址掩码。

注：数据触发指令设置为 **Reg write** 和 **Reg read** 时有效，相关指令:TRIGger:RFFE:DATACmd。

参数说明

<Int> 范围 0x00~0x1F。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的地址掩码。

实例说明

:TRIGger:RFFE:ADDRMask 0x10　　设置 MIPI-RFFE 协议触发的地址掩码为 0x10。

:TRIGger:RFFE:ADDRMask? -> 0x10　　查询 MIPI-RFFE 协议触发的地址掩码为 0x10。

:TRIGger:RFFE:DATAMask

命令格式

:TRIGger:RFFE:DATAMask <Int>

:TRIGger:RFFE:DATAMask?

功能描述

设置 MIPI-RFFE 协议触发的数据掩码。

查询 MIPI-RFFE 协议触发的数据掩码。

注：数据触发指令设置为 **Reg write** 和 **Reg read** 时有效，相关指令:TRIGger:RFFE:DATACmd。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 MIPI-RFFE 协议触发的数据掩码。

实例说明

:TRIGger:RFFE:DATAMask 0x10　　设置 MIPI-RFFE 协议触发的数据掩码为 0x10。

:TRIGger:RFFE:DATAMask? -> 0x10　　查询 MIPI-RFFE 协议触发的数据掩码为 0x10。

:TRIGger:SDSD:SCK

命令格式

:TRIGger:SDSD:SCK <String>

:TRIGger:SDSD:SCK?

功能描述

设置 SD-SD 协议触发的时钟信源。

查询 SD-SD 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SDSD:SCK CHANnel1	设置 SD-SD 协议触发的时钟信源为通道 1。
:TRIGger:SDSD:SCK? -> CHANnel1	查询 SD-SD 协议触发的时钟信源为通道 1。

:TRIGger:SDSD:CMD

命令格式

:TRIGger:SDSD:CMD <String>

:TRIGger:SDSD:CMD?

功能描述

设置 SD-SD 协议触发的命令信源。

查询 SD-SD 协议触发的命令信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SDSD:CMD CHANnel1　　设置 SD-SD 协议触发的命令信源为通道 1。

:TRIGger:SDSD:CMD? -> CHANnel1　　查询 SD-SD 协议触发的命令信源为通道 1。

:TRIGger:SDSD:TRIGMode

命令格式

:TRIGger:SDSD:TRIGMode <String>

:TRIGger:SDSD:TRIGMode?

功能描述

设置 SD-SD 协议触发的触发模式。

查询 SD-SD 协议触发的触发模式。

参数说明

<String> 范围 { STARt | CMD | CMDArg }，分别表示开始位、命令、命令及参数。

返回格式

查询命令返回 STARt、CMD 或 CMDArg。

实例说明

:TRIGger:SDSD:TRIGMode STARt　　设置 SD-SD 协议触发的触发模式为开始位。

:TRIGger:SDSD:TRIGMode? -> STARt　　查询 SD-SD 协议触发的触发模式为开始位。

:TRIGger:SDSD:IDLETime

命令格式

:TRIGger:SDSD:IDLETime <String>

:TRIGger:SDSD:IDLETime?

功能描述

设置 SD-SD 协议触发的空闲时间。

查询 SD-SD 协议触发的空闲时间。

参数说明

<String> 范围 40ns~1ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SD-SD 协议触发的空闲时间。

实例说明

:TRIGger:SDSD:IDLETime 2e-4	设置 SD-SD 协议触发的空闲时间为 200.0us。
:TRIGger:SDSD:IDLETime? -> 200.0us	查询 SD-SD 协议触发的空闲时间为 200.0us。

:TRIGger:SDSD:BUSType

命令格式

:TRIGger:SDSD:BUSType <String>

:TRIGger:SDSD:BUSType?

功能描述

设置 SD-SD 协议触发的总线类型。

查询 SD-SD 协议触发的总线类型。

参数说明

<String> 范围 { SD | MMC }。

返回格式

查询命令返回 SD 或 MMC。

实例说明

:TRIGger:SDSD:BUSType SD	设置 SD-SD 协议触发的总线类型为 SD。
:TRIGger:SDSD:BUSType? -> SD	查询 SD-SD 协议触发的总线类型为 SD。

:TRIGger:SDSD:CMDValue

命令格式

:TRIGger:SDSD:CMDValue <Int>

:TRIGger:SDSD:CMDValue?

功能描述

设置 SD-SD 协议触发的 CMD 值。

查询 SD-SD 协议触发的 CMD 值。

注：触发模式设置为**命令**和**命令及参数**时有效，相关指令:TRIGger:SDSD:TRIGMode。

参数说明

<Int> 范围 0~59。

返回格式

查询命令以整型形式返回 SD-SD 协议触发的 CMD 值。

实例说明

:TRIGger:SDSD:CMDValue 0	设置 SD-SD 协议触发的 CMD 值为 0。
:TRIGger:SDSD:CMDValue? -> 0	查询 SD-SD 协议触发的 CMD 值为 0。

:TRIGger:SDSD:DATAIndex

命令格式

:TRIGger:SDSD:DATAIndex <String>

:TRIGger:SDSD:DATAIndex?

功能描述

设置 SD-SD 协议触发的数据索引。

查询 SD-SD 协议触发的数据索引。

注：触发模式设置为**命令及参数**时有效，相关指令:TRIGger:SDSD:TRIGMode。

参数说明

<String> 范围 { ARG1 | ARG2 | ARG3 | ARG4 }。

返回格式

查询命令返回 ARG1、ARG2、ARG3 或 ARG4。

实例说明

:TRIGger:SDSD:DATAIndex ARG1　　设置 SD-SD 协议触发的数据索引为 ARG1。

:TRIGger:SDSD:DATAIndex? -> ARG1　　查询 SD-SD 协议触发的数据索引为 ARG1。

:TRIGger:SDSD:CMDArg

命令格式

:TRIGger:SDSD:CMDArg <Int>

:TRIGger:SDSD:CMDArg?

功能描述

设置 SD-SD 协议触发的 ARG 值。

查询 SD-SD 协议触发的 ARG 值。

注：触发模式设置为**命令及参数**时有效，相关指令:TRIGger:SDSD:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 SD-SD 协议触发的 ARG 值。

实例说明

:TRIGger:SDSD:CMDArg 0x10	设置 SD-SD 协议触发的 ARG 值为 0x10。
:TRIGger:SDSD:CMDArg? -> 0x10	查询 SD-SD 协议触发的 ARG 值为 0x10。

:TRIGger:SDSPi:SCK

命令格式

:TRIGger:SDSPi:SCK <String>

:TRIGger:SDSPi:SCK?

功能描述

设置 SD-SPI 协议触发的时钟信源。

查询 SD-SPI 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SDSPi:SCK CHANnel1	设置 SD-SPI 协议触发的时钟信源为通道 1。
:TRIGger:SDSPi:SCK? -> CHANnel1	查询 SD-SPI 协议触发的时钟信源为通道 1。

:TRIGger:SDSPi:CMD

命令格式

:TRIGger:SDSPi:CMD <String>

:TRIGger:SDSPi:CMD?

功能描述

设置 SD-SPI 协议触发的命令信源。

查询 SD-SPI 协议触发的命令信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SDSPi:CMD CHANnel1　　设置 SD-SPI 协议触发的命令信源为通道 1。

:TRIGger:SDSPi:CMD? -> CHANnel1　　查询 SD-SPI 协议触发的命令信源为通道 1。

:TRIGger:SDSPi:TRIGMode

命令格式

:TRIGger:SDSPi:TRIGMode <String>

:TRIGger:SDSPi:TRIGMode?

功能描述

设置 SD-SPI 协议触发的触发模式。

查询 SD-SPI 协议触发的触发模式。

参数说明

<String> 范围 { STARt | CMD | CMDArg }，分别表示开始位、命令、命令及参数。

返回格式

查询命令返回 STARt、CMD 或 CMDArg。

实例说明

:TRIGger:SDSPi:TRIGMode STARt　　设置 SD-SPI 协议触发的触发模式为开始位。

:TRIGger:SDSPi:TRIGMode? -> STARt　　查询 SD-SPI 协议触发的触发模式为开始位。

:TRIGger:SDSPi:IDLETime

命令格式

:TRIGger:SDSPi:IDLETime <String>

:TRIGger:SDSPi:IDLETime?

功能描述

设置 SD-SPI 协议触发的空闲时间。

查询 SD-SPI 协议触发的空闲时间。

参数说明

<String> 范围 40ns~1ms，可带单位，如“100ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SD-SPI 协议触发的空闲时间。

实例说明

:TRIGger:SDSPi:IDLETime 2e-4	设置 SD-SPI 协议触发的空闲时间为 200.0us。
:TRIGger:SDSPi:IDLETime? -> 200.0us	查询 SD-SPI 协议触发的空闲时间为 200.0us。

:TRIGger:SDSPi:CMDValue

命令格式

:TRIGger:SDSPi:CMDValue <Int>

:TRIGger:SDSPi:CMDValue?

功能描述

设置 SD-SPI 协议触发的 CMD 值。

查询 SD-SPI 协议触发的 CMD 值。

注：触发模式设置为**命令**和**命令及参数**时有效，相关指令:TRIGger:SDSPi:TRIGMode。

参数说明

<Int> 范围 0~59。

返回格式

查询命令以整型形式返回 SD-SPI 协议触发的 CMD 值。

实例说明

:TRIGger:SDSPi:CMDValue 0	设置 SD-SPI 协议触发的 CMD 值为 0。
:TRIGger:SDSPi:CMDValue? -> 0	查询 SD-SPI 协议触发的 CMD 值为 0。

:TRIGger:SDSPi:DATAIndex

命令格式

:TRIGger:SDSPi:DATAIndex <String>

:TRIGger:SDSPi:DATAIndex?

功能描述

设置 SD-SPI 协议触发的数据索引。

查询 SD-SPI 协议触发的数据索引。

注：触发模式设置为**命令及参数**时有效，相关指令:TRIGger:SDSPi:TRIGMode。

参数说明

<String> 范围 { ARG1 | ARG2 | ARG3 | ARG4 }。

返回格式

查询命令返回 ARG1、ARG2、ARG3 或 ARG4。

实例说明

:TRIGger:SDSPi:DATAIndex ARG1　　设置 SD-SPI 协议触发的数据索引为 ARG1。

:TRIGger:SDSPi:DATAIndex? -> ARG1　　查询 SD-SPI 协议触发的数据索引为 ARG1。

:TRIGger:SDSPi:CMDArg

命令格式

:TRIGger:SDSPi:CMDArg <Int>

:TRIGger:SDSPi:CMDArg?

功能描述

设置 SD-SPI 协议触发的 ARG 值。

查询 SD-SPI 协议触发的 ARG 值。

注：触发模式设置为**命令及参数**时有效，相关指令:TRIGger:SDSPi:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 SD-SPI 协议触发的 ARG 值。

实例说明

:TRIGger:SDSPi:CMDArg 0x10	设置 SD-SPI 协议触发的 ARG 值为 0x10。
:TRIGger:SDSPi:CMDArg? -> 0x10	查询 SD-SPI 协议触发的 ARG 值为 0x10。

:TRIGger:SENT:SOURce

命令格式

:TRIGger:SENT:SOURce <String>

:TRIGger:SENT:SOURce?

功能描述

设置 SENT 协议触发的数据信源。

查询 SENT 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SENT:SOURce CHANnel1　　设置 SENT 协议触发数据信源为通道 1。

:TRIGger:SENT:SOURce? -> CHANnel1　　查询 SENT 协议触发数据信源为通道 1。

:TRIGger:SENT:PULSes

命令格式

:TRIGger:SENT:PULSes <String>

:TRIGger:SENT:PULSes?

功能描述

设置 SENT 协议触发的数据脉冲个数。

查询 SENT 协议触发的数据脉冲个数。

参数说明

<String> 范围 { 1 | 2 | 3 | 4 | 5 | 6 }。

返回格式

查询命令返回 1、2、3、4、5 或 6。

实例说明

:TRIGger:SENT:PULSes 1　　设置 SENT 协议触发的数据脉冲个数为 1。

:TRIGger:SENT:PULSes? -> 1　　查询 SENT 协议触发的数据脉冲个数为 1。

:TRIGger:SENT:TICK

命令格式

:TRIGger:SENT:TICK <String>

:TRIGger:SENT:TICK?

功能描述

设置 SENT 协议触发的时间片宽度。

查询 SENT 协议触发的时间片宽度。

参数说明

<String> 范围 500ns~90us，可带单位，如“600ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SENT 协议触发的时间片宽度。

实例说明

:TRIGger:SENT:TICK 2e-5	设置 SENT 协议触发的时间片宽度为 20us。
:TRIGger:SENT:TICK? -> 20.00us	查询 SENT 协议触发的时间片宽度为 20us。

:TRIGger:SENT:TRIGMode

命令格式

:TRIGger:SENT:TRIGMode <String>

:TRIGger:SENT:TRIGMode?

功能描述

设置 SENT 协议触发的触发模式。

查询 SENT 协议触发的触发模式。

参数说明

<String> 范围 { SYNC | STATus | SD | SDC }，分别表示同步场触发、状态场触发、状态+数据触发和状态+数据+校验和触发。

返回格式

查询命令返回 SYNC、STATus、SD 或 SDC。

实例说明

:TRIGger:SENT:TRIGMode SYNC　　设置 SENT 协议触发的触发模式为同步场触发。

:TRIGger:SENT:TRIGMode? -> SYNC　　查询 SENT 协议触发的触发模式为同步场触发。

:TRIGger:SENT:TRIGStatus

命令格式

:TRIGger:SENT:TRIGStatus <Int>

:TRIGger:SENT:TRIGStatus?

功能描述

设置 SENT 协议触发的状态数据。

查询 SENT 协议触发的状态数据。

注：触发模式设置为**状态场触发**、**状态+数据触发**和**状态+数据+校验和触发**时有效，相关指令:TRIGger:SENT:TRIGMode。

参数说明

<Int> 范围 0x00~0x0F。

返回格式

查询命令以十六进制形式返回 SENT 协议触发的状态数据。

实例说明

:TRIGger:SENT:TRIGStatus 0x01　　设置 SENT 协议触发的状态数据为 0x01。

:TRIGger:SENT:TRIGStatus? -> 0x01　　查询 SENT 协议触发的状态数据为 0x01。

:TRIGger:SENT:DATAIndex

命令格式

:TRIGger:SENT:DATAIndex <Int>

:TRIGger:SENT:DATAIndex?

功能描述

设置 SENT 协议触发的数据索引。

查询 SENT 协议触发的数据索引。

注：触发模式设置为**状态+数据触发**和**状态+数据+校验和触发**时有效，相关指令:TRIGger:SENT:TRIGMode。

参数说明

<Int> 范围 1~6。

返回格式

查询命令以整型形式返回 SENT 协议触发的数据索引。

实例说明

:TRIGger:SENT:DATAIndex 1　　设置 SENT 协议触发的数据索引为 6。

:TRIGger:SENT:DATAIndex? -> 1　　查询 SENT 协议触发的数据索引为 6。

:TRIGger:SENT:DATAVal

命令格式

:TRIGger:SENT:DATAVal <Int>

:TRIGger:SENT:DATAVal?

功能描述

设置 SENT 协议触发的触发数据值。

查询 SENT 协议触发的触发数据值。

注：触发模式设置为**状态+数据触发**和**状态+数据+校验和触发**时有效，相关指令:TRIGger:SENT:TRIGMode。

参数说明

<Int> 范围 0x00~0x0F。

返回格式

查询命令以十六进制形式返回 SENT 协议触发的触发数据值。

实例说明

:TRIGger:SENT:DATAVal 0x01　　设置 SENT 协议触发的触发数据值为 0x01。

:TRIGger:SENT:DATAVal? -> 0x01　　查询 SENT 协议触发的触发数据值为 0x01。

:TRIGger:SENT:CHECksum

命令格式

:TRIGger:SENT:CHECksum <Int>

:TRIGger:SENT:CHECksum?

功能描述

设置 SENT 协议触发的校验和。

查询 SENT 协议触发的校验和。

注：触发模式设置为**状态+数据+校验和触发**时有效，相关指令:TRIGger:SENT:TRIGMode。

参数说明

<Int> 范围 0x00~0x0F。

返回格式

查询命令以十六进制形式返回 SENT 协议触发的校验和。

实例说明

:TRIGger:SENT:CHECksum 0x01　　设置 SENT 协议触发的校验和为 0x01。

:TRIGger:SENT:CHECksum? -> 0x01　　查询 SENT 协议触发的校验和为 0x01。

:TRIGger:SHT11:CLK

命令格式

:TRIGger:SHT11:CLK <String>

:TRIGger:SHT11:CLK?

功能描述

设置 SHT11 协议触发的时钟信源。

查询 SHT11 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SHT11:CLK CHANnel1　　设置 SHT11 协议触发的时钟信源为通道 1。

:TRIGger:SHT11:CLK? -> CHANnel1　　查询 SHT11 协议触发的时钟信源为通道 1。

:TRIGger:SHT11:DAT

命令格式

:TRIGger:SHT11:DAT <String>

:TRIGger:SHT11:DAT?

功能描述

设置 SHT11 协议触发的数据信源。

查询 SHT11 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SHT11:DAT CHANnel1　　设置 SHT11 协议触发的数据信源为通道 1。

:TRIGger:SHT11:DAT? -> CHANnel1　　查询 SHT11 协议触发的数据信源为通道 1。

:TRIGger:SHT11:TRIGMode

命令格式

:TRIGger:SHT11:TRIGMode <String>

:TRIGger:SHT11:TRIGMode?

功能描述

设置 SHT11 协议触发的触发模式。

查询 SHT11 协议触发的触发模式。

参数说明

<String> 范围 { STARt }，表示起始位触发。

返回格式

查询命令返回 STARt。

实例说明

:TRIGger:SHT11:TRIGMode STARt 设置 SHT11 协议触发的触发模式为起始位。

:TRIGger:SHT11:TRIGMode? -> STARt 查询 SHT11 协议触发的触发模式为起始位。

:TRIGger:SPC:SOURce

命令格式

:TRIGger:SPC:SOURce <String>

:TRIGger:SPC:SOURce?

功能描述

设置 SPC 协议触发的数据信源。

查询 SPC 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SPC:SOURce CHANnel1	设置 SPC 协议触发数据信源为通道 1。
:TRIGger:SPC:SOURce? -> CHANnel1	查询 SPC 协议触发数据信源为通道 1。

:TRIGger:SPC:DATAfieldnumber

命令格式

:TRIGger:SPC:DATAfieldnumber <String>

:TRIGger:SPC:DATAfieldnumber?

功能描述

设置 SPC 协议触发的数据字段个数。

查询 SPC 协议触发的数据字段个数。

参数说明

<String> 范围 { 3 | 4 | 5 | 6 }。

返回格式

查询命令返回 3、4、5 或 6。

实例说明

:TRIGger:SPC:DATAfieldnumber 4　　设置 SPC 协议触发的数据字段个数为 4。

:TRIGger:SPC:DATAfieldnumber? -> 4　　查询 SPC 协议触发的数据字段个数为 4。

:TRIGger:SPC:UNITtime

命令格式

:TRIGger:SPC:UNITtime <String>

:TRIGger:SPC:UNITtime?

功能描述

设置 SPC 协议触发的时间片宽度。

查询 SPC 协议触发的时间片宽度。

参数说明

<String> 范围 500.0ns~3.880us，可带单位，如“600ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SPC 协议触发的时间片宽度。

实例说明

:TRIGger:SPC:UNITtime 2e-6	设置 SPC 协议触发的时间片宽度为 2us。
:TRIGger:SPC:UNITtime? -> 2.000us	查询 SPC 协议触发的时间片宽度为 2us。

:TRIGger:SPC:TRIGMode

命令格式

:TRIGger:SPC:TRIGMode <String>

:TRIGger:SPC:TRIGMode?

功能描述

设置 SPC 协议触发的触发模式。

查询 SPC 协议触发的触发模式。

参数说明

<String> 范围 { TRIG | SYNC | STATus | DATA }，分别表示触发场触发、同步场触发、状态场触发和数据场触发。

返回格式

查询命令返回 TRIG、SYNC、STATus 或 DATA。

实例说明

:TRIGger:SPC:TRIGMode TRIG　　设置 SPC 协议触发的触发模式为触发场触发。

:TRIGger:SPC:TRIGMode? -> TRIG　　查询 SPC 协议触发的触发模式为触发场触发。

:TRIGger:SPC:PROTocolmode

命令格式

:TRIGger:SPC:PROTocolmode <String>

:TRIGger:SPC:PROTocolmode?

功能描述

设置 SPC 协议触发的协议模式。

查询 SPC 协议触发的协议模式。

参数说明

<String> 范围 { SYNChronous | RANGeselection | IDSElection }，分别表示同步模式、范围选择和 ID 选择。

返回格式

查询命令返回 SYNChronous、RANGeselection 或 IDSElection。

实例说明

:TRIGger:SPC:PROTocolmode SYNChronous 设置 SPC 协议触发的协议模式为同步模式。

:TRIGger:SPC:PROTocolmode? -> SYNChronous 查询 SPC 协议触发的协议模式为同步模式。

:TRIGger:SPC:SLAVeunittime

命令格式

:TRIGger:SPC:SLAVeunittime <String>

:TRIGger:SPC:SLAVeunittime?

功能描述

设置 SPC 协议触发的从机单位时间。

查询 SPC 协议触发的从机单位时间。

注：触发模式设置为**同步场触发**、**状态场触发**和**数据场触发**时有效，相关指令:TRIGger:SPC:TRIGMode。

参数说明

<String> 范围 500.0ns~3.880us，可带单位，如“600ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SPC 协议触发的从机单位时间。

实例说明

:TRIGger:SPC:SLAVeunittime 2e-6　　设置 SPC 协议触发的从机单位时间为 2us。

:TRIGger:SPC:SLAVeunittime? -> 2.000us　　查询 SPC 协议触发的从机单位时间为 2us。

:TRIGger:SPC:TRIGStatus

命令格式

:TRIGger:SPC:TRIGStatus <Int>

:TRIGger:SPC:TRIGStatus?

功能描述

设置 SPC 协议触发的状态数据。

查询 SPC 协议触发的状态数据。

注：触发模式设置为**状态场触发**和**数据场触发**时有效，相关指令:TRIGger:SPC:TRIGMode。

参数说明

<Int> 范围 0x00~0x0F。

返回格式

查询命令以十六进制形式返回 SPC 协议触发的状态数据。

实例说明

:TRIGger:SPC:TRIGStatus 0x01	设置 SPC 协议触发的状态数据为 0x01。
:TRIGger:SPC:TRIGStatus? -> 0x01	查询 SPC 协议触发的状态数据为 0x01。

:TRIGger:SPC:COMParetype34

命令格式

:TRIGger:SPC:COMParetype34 <String>

:TRIGger:SPC:COMParetype34?

功能描述

设置 SPC 协议触发数据字段个数为 3 和 4 时的比较模式。

查询 SPC 协议触发数据字段个数为 3 和 4 时的比较模式。

注 1：数据字段个数设置为 **3** 和 **4** 时有效，相关指令:TRIGger:SPC:DATAfieldnumber。

注 2：触发模式设置为**数据场触发**时有效，相关指令:TRIGger:SPC:TRIGMode。

参数说明

<String> 范围 ｛HALLGREAtequal | HALLLESSequal｝，分别表示霍尔值大于和霍尔值小于。

返回格式

查询命令返回 HALLGREAtequal 或 HALLLESSequal。

实例说明

:TRIGger:SPC:COMParetype34 HALLGREAtequal 设置 SPC 协议触发数据字段个数为 3 和 4 时的比较模式为霍尔值大于。

:TRIGger:SPC:COMParetype34? -> HALLGREAtequal 查询 SPC 协议触发数据字段个数为 3 和 4 时的比较模式为霍尔值大于。

:TRIGger:SPC:COMParetype56

命令格式

:TRIGger:SPC:COMParetype56 <String>

:TRIGger:SPC:COMParetype56?

功能描述

设置 SPC 协议触发数据字段个数为 5 和 6 时的比较模式。

查询 SPC 协议触发数据字段个数为 5 和 6 时的比较模式。

注 1：数据字段个数设置为 **5** 和 **6** 时有效，相关指令:TRIGger:SPC:DATAfieldnumber。

注 2：触发模式设置为**数据场触发**时有效，相关指令:TRIGger:SPC:TRIGMode。

参数说明

<String> 范围 { HALLGREAtequal | HALLLESSequal | TEMPGREAtequal | TEMPLESSequal }，分别表示霍尔值大于、霍尔值小于、温度大于和温度小于。

返回格式

查询命令返回 HALLGREAtequal 、 HALLLESSequal 、 TEMPGREAtequal 或 TEMPLESSequal。

实例说明

:TRIGger:SPC:COMParetype56 HALLGREAtequal 设置 SPC 协议触发数据字段个数为 5 和 6 时的比较模式为霍尔值大于。

:TRIGger:SPC:COMParetype56? -> HALLGREAtequal 查询 SPC 协议触发数据字段个数为 5 和 6 时的比较模式为霍尔值大于。

:TRIGger:SPC:HALL

命令格式

:TRIGger:SPC:HALL <Int>

:TRIGger:SPC:HALL?

功能描述

设置 SPC 协议触发的霍尔值。

查询 SPC 协议触发的霍尔值。

注 1：触发模式设置为**数据场触发**时有效，相关指令:TRIGger:SPC:TRIGMode。

注 2：比较模式设置为**霍尔值大于**和**霍尔值小于**时有效，相关指令:TRIGger:SPC:COMParetype34 和:TRIGger:SPC:COMParetype56。

参数说明

<Int> 范围 0x00~0xFFFF。

返回格式

查询命令以十六进制形式返回 SPC 协议触发的霍尔值。

实例说明

:TRIGger:SPC:HALL 0x01　　设置 SPC 协议触发的霍尔值为 0x01。

:TRIGger:SPC:HALL? -> 0x01　　查询 SPC 协议触发的霍尔值为 0x01。

:TRIGger:SPC:TEMPerature

命令格式

:TRIGger:SPC:TEMPerature <Int>

:TRIGger:SPC:TEMPerature?

功能描述

设置 SPC 协议触发的温度值。

查询 SPC 协议触发的温度值。

注 1：触发模式设置为**数据场触发**时有效，相关指令:TRIGger:SPC:TRIGMode。

注 2：比较模式设置为**温度大于**和**温度小于**时有效，相关指令:TRIGger:SPC:COMParetype56。

参数说明

<Int> 范围-55~200。

返回格式

查询命令以整型形式返回 SPC 协议触发的温度值。

实例说明

:TRIGger:SPC:TEMPerature 80	设置 SPC 协议触发的温度值为 80。
:TRIGger:SPC:TEMPerature? -> 80	查询 SPC 协议触发的温度值为 80。

:TRIGger:SPI:SCK

命令格式

:TRIGger:SPI:SCK <String>

:TRIGger:SPI:SCK?

功能描述

设置 SPI 协议触发的时钟信源。

查询 SPI 协议触发的时钟信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SPI:SCK CHANnel1　　设置 SPI 协议触发的时钟信源为通道 1。

:TRIGger:SPI:SCK? -> CHANnel1　　查询 SPI 协议触发的时钟信源为通道 1。

:TRIGger:SPI:SDA

命令格式

:TRIGger:SPI:SDA <String>

:TRIGger:SPI:SDA?

功能描述

设置 SPI 协议触发的数据信源。

查询 SPI 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:SPI:SDA CHANnel1	设置 SPI 协议触发的数据信源为通道 1。
:TRIGger:SPI:SDA? -> CHANnel1	查询 SPI 协议触发的数据信源为通道 1。

:TRIGger:SPI:CS

命令格式

:TRIGger:SPI:CS <String>

:TRIGger:SPI:CS?

功能描述

设置 SPI 协议触发的片选信源。

查询 SPI 协议触发的片选信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | NONE }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4 或 NONE。

实例说明

:TRIGger:SPI:CS CHANnel1	设置 SPI 协议触发的片选信源为通道 1。
:TRIGger:SPI:CS? -> CHANnel1	查询 SPI 协议触发的片选信源为通道 1。

:TRIGger:SPI:SAMPmode

命令格式

:TRIGger:SPI:SAMPmode <String>

:TRIGger:SPI:SAMPmode?

功能描述

设置 SPI 协议触发的工作方式。

查询 SPI 协议触发的工作方式。

参数说明

<String> 范围 { POLPha00 | POLPha01 | POLPha10 | POLPha11 }，分别表示 POL:PHA 0:0、POL:PHA 0:1、POL:PHA 1:0、POL:PHA 1:1。

返回格式

查询命令返回 POLPha00、POLPha01、POLPha10 或 POLPha11。

实例说明

:TRIGger:SPI:SAMPmode POLPha00　　设置 SPI 协议触发的工作方式为 POL:PHA 0:0。

:TRIGger:SPI:SAMPmode? -> POLPha00　查询 SPI 协议触发的工作方式为 POL:PHA 0:0。

:TRIGger:SPI:TRANsmode

命令格式

:TRIGger:SPI:TRANsmode <String>

:TRIGger:SPI:TRANsmode?

功能描述

设置 SPI 协议触发的传输模式。

查询 SPI 协议触发的传输模式。

参数说明

<String> 范围 { LSB | MSB}。

返回格式

查询命令返回 LSB 或 MSB。

实例说明

:TRIGger:SPI:TRANsmode LSB	设置 SPI 协议触发的传输模式为 LSB。
:TRIGger:SPI:TRANsmode? -> LSB	查询 SPI 协议触发的传输模式为 LSB。

:TRIGger:SPI:DATAlen

命令格式

:TRIGger:SPI:DATAlen <Int>

:TRIGger:SPI:DATAlen?

功能描述

设置 SPI 协议触发的数据宽度。

查询 SPI 协议触发的数据宽度。

参数说明

<Int> 数据宽度，范围 4~32。

返回格式

查询命令以整型形式返回 SPI 协议触发的数据宽度大小。

实例说明

:TRIGger:SPI:DATAlen 8	设置 SPI 协议触发的数据宽度为 8。
:TRIGger:SPI:DATAlen? ->8	查询 SPI 协议触发的数据宽度为 8。

:TRIGger:SPI:OVERtime

命令格式

:TRIGger:SPI:OVERtime <String>

:TRIGger:SPI:OVERtime?

功能描述

设置 SPI 协议触发的超时值。

查询 SPI 协议触发的超时值。

参数说明

<String> 范围 40.00ns~1.000s，可带单位，如“600ns”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 SPI 协议触发的超时值。

实例说明

:TRIGger:SPI:OVERtime 2e-6	设置 SPI 协议触发的超时值为 2us。
:TRIGger:SPI:OVERtime? -> 2.000us	查询 SPI 协议触发的超时值为 2us。

:TRIGger:SPI:TRIGMode

命令格式

:TRIGger:SPI:TRIGMode <String>

:TRIGger:SPI:TRIGMode?

功能描述

设置 SPI 协议触发的触发模式。

查询 SPI 协议触发的触发模式。

参数说明

<String> 范围 { BEGIn | DATA }，分别表示起始触发和数据触发。

返回格式

查询命令返回 BEGIn 或 DATA。

实例说明

:TRIGger:SPI:TRIGMode BEGIn	设置 SPI 协议触发的触发模式为起始触发。
:TRIGger:SPI:TRIGMode? -> BEGIn	查询 SPI 协议触发的触发模式为起始触发。

:TRIGger:SPI:TRIGData

命令格式

:TRIGger:SPI:TRIGData <Int>

:TRIGger:SPI:TRIGData?

功能描述

设置 SPI 协议触发的触发数据。

查询 SPI 协议触发的触发数据。

注：触发模式设置为**数据触发**时有效，相关指令:TRIGger:SPI:TRIGMode。

参数说明

<Int> 触发数据值。

注：触发数据范围根据数据宽度变化，相关指令:TRIGger:SPI:DATAlen。

数据宽度大小	触发数据范围	数据宽度大小	触发数据范围	数据宽度大小	触发数据范围
4	0x00~0x0F	14	0x00~0x3FFF	24	0x00~0xFFFFFF
5	0x00~0x1F	15	0x00~0x7FFF	25	0x00~0x1FFFFFF
6	0x00~0x3F	16	0x00~0xFFFF	26	0x00~0x3FFFFFF
7	0x00~0x7F	17	0x00~0x1FFFF	27	0x00~0x7FFFFFF
8	0x00~0xFF	18	0x00~0x3FFFF	28	0x00~0xFFFFFFF
9	0x00~0x1FF	19	0x00~0x7FFFF	29	0x00~0x1FFFFFFF
10	0x00~0x3FF	20	0x00~0xFFFFF	30	0x00~0x3FFFFFFF
11	0x00~0x7FF	21	0x00~0x1FFFFF	31	0x00~0x7FFFFFFF
12	0x00~0xFFF	22	0x00~0x3FFFFF	32	0x00~0xFFFFFFFF
13	0x00~0x1FFF	23	0x00~0x7FFFFF		

返回格式

查询命令以十六进制形式返回 SPI 协议触发的触发数据。

实例说明

:TRIGger:SPI:TRIGData 0x10　　设置 SPI 协议触发的触发数据为 0x10。

:TRIGger:SPI:TRIGData? -> 0x10　　查询 SPI 协议触发的触发数据为 0x10。

:TRIGger:UART:SOURce

命令格式

:TRIGger:UART:SOURce <String>

:TRIGger:UART:SOURce?

功能描述

设置 UART 协议触发的信源。

查询 UART 协议触发的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:UART:SOURce CHANnel1　　设置 UART 协议触发信源为通道 1。

:TRIGger:UART:SOURce? -> CHANnel1　　查询 UART 协议触发信源为通道 1。

:TRIGger:UART:BAUDrate

命令格式

:TRIGger:UART:BAUDrate <Int>

:TRIGger:UART:BAUDrate?

功能描述

设置 UART 协议触发的波特率。

查询 UART 协议触发的波特率。

参数说明

<Int> 波特率，范围 100~20000000。

返回格式

查询命令以整型形式返回 UART 协议触发的波特率大小。

实例说明

:TRIGger:UART:BAUDrate 9600	设置 UART 协议触发的波特率为 9600。
:TRIGger:UART:BAUDrate? -> 9600	查询 UART 协议触发的波特率为 9600。

:TRIGger:UART:STOPbit

命令格式

:TRIGger:UART:STOPbit <String>

:TRIGger:UART:STOPbit?

功能描述

设置 UART 协议触发的结束位宽。

查询 UART 协议触发的结束位宽。

参数说明

<String> 范围 { 1 | 1.5 | 2 }。

返回格式

查询命令返回 1、1.5 或 2。

实例说明

:TRIGger:UART:STOPbit 1	设置 UART 协议触发的结束位宽为 1。
:TRIGger:UART:STOPbit? -> 1	查询 UART 协议触发的结束位宽为 1。

:TRIGger:UART:PARItymode

命令格式

:TRIGger:UART:PARItymode <String>

:TRIGger:UART:PARItymode?

功能描述

设置 UART 协议触发的校验模式。

查询 UART 协议触发的校验模式。

参数说明

<String> 范围 { EVEN | ODD | MARK | SPACe | NONE }。

返回格式

查询命令返回 EVEN、ODD、MARK、SPACe 或 NONE。

实例说明

:TRIGger:UART:PARItymode NONE　　设置 UART 协议触发的校验模式为 NONE。

:TRIGger:UART:PARItymode? -> NONE　　查询 UART 协议触发的校验模式为 NONE。

:TRIGger:UART:DATAMode

命令格式

:TRIGger:UART:DATAMode <String>

:TRIGger:UART:DATAMode?

功能描述

设置 UART 协议触发的数据模式。

查询 UART 协议触发的数据模式。

参数说明

<String> 范围 { LSB | MSB}。

返回格式

查询命令返回 LSB 或 MSB。

实例说明

:TRIGger:UART:DATAMode LSB	设置 UART 协议触发的数据模式为 LSB。
:TRIGger:UART:DATAMode? -> LSB	查询 UART 协议触发的数据模式为 LSB。

:TRIGger:UART:REVErse

命令格式

:TRIGger:UART:REVErse <String>

:TRIGger:UART:REVErse?

功能描述

设置 UART 协议触发的电平反相。

查询 UART 协议触发的电平反相。

参数说明

<String> 范围 { FALSe | TRUE }。

返回格式

查询命令返回 FALSe 或 TRUE。

实例说明

:TRIGger:UART:REVErse FALSe	设置 UART 协议触发的电平反相为 FALSe。
:TRIGger:UART:REVErse? -> FALSe	查询 UART 协议触发的电平反相为 FALSe。

:TRIGger:UART:TRIGMode

命令格式

:TRIGger:UART:TRIGMode <String>

:TRIGger:UART:TRIGMode?

功能描述

设置 UART 协议触发的触发模式。

查询 UART 协议触发的触发模式。

参数说明

<String> 范围 { BEGIn | DATA }，分别表示开始触发和数据触发。

返回格式

查询命令返回 BEGIn 或 DATA。

实例说明

:TRIGger:UART:TRIGMode BEGIn　　设置 UART 协议触发的触发模式为开始触发。

:TRIGger:UART:TRIGMode? -> BEGIn　　查询 UART 协议触发的触发模式为开始触发。

:TRIGger:UART:DATALen

命令格式

:TRIGger:UART:DATALen <String>

:TRIGger:UART:DATALen?

功能描述

设置 UART 协议触发的数据位宽。

查询 UART 协议触发的数据位宽。

参数说明

<String> 范围 { 4 | 5 | 6 | 7 | 8 | 9 | 12 | 16 }。

返回格式

查询命令返回 4、5、6、7、8、9、12 或 16。

实例说明

:TRIGger:UART:DATALen 8	设置 UART 协议触发的数据位宽为 8。
:TRIGger:UART:DATALen? -> 8	查询 UART 协议触发的数据位宽为 8。

:TRIGger:UART:TRIGData

命令格式

:TRIGger:UART:TRIGData <Int>

:TRIGger:UART:TRIGData?

功能描述

设置 UART 协议触发的触发数据。

查询 UART 协议触发的触发数据。

注：触发模式设置为**数据触发**时有效，相关指令:TRIGger:UART:TRIGMode。

参数说明

<Int> 触发数据值。

注：触发数据范围根据数据宽度变化，相关指令:TRIGger:UART:DATALen。

数据宽度大小	触发数据范围
4	0x00~0x0F
5	0x00~0x1F
6	0x00~0x3F
7	0x00~0x7F
8	0x00~0xFF
9	0x00~0x1FF
12	0x00~0xFFF
16	0x00~0xFFFF

返回格式

查询命令以十六进制形式返回 UART 协议触发的触发数据。

实例说明

:TRIGger:UART:TRIGData 0x10　　设置 UART 协议触发的触发数据为 0x10。

:TRIGger:UART:TRIGData? -> 0x10　　查询 UART 协议触发的触发数据为 0x10。

:TRIGger:USB:D+

命令格式

:TRIGger:USB:D+ <String>

:TRIGger:USB:D+?

功能描述

设置 USB 协议触发的 D+的信源。

查询 USB 协议触发的 D+的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:USB:D+ CHANnel1	设置 USB 协议触发的 D+信源为通道 1。
:TRIGger:USB:D+? -> CHANnel1	查询 USB 协议触发的 D+信源为通道 1。

:TRIGger:USB:D-

命令格式

:TRIGger:USB:D- <String>

:TRIGger:USB:D-?

功能描述

设置 USB 协议触发的 D-的信源。

查询 USB 协议触发的 D-的信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:USB:D- CHANnel1　　设置 USB 协议触发的 D-信源为通道 1。

:TRIGger:USB:D-? -> CHANnel1　　查询 USB 协议触发的 D-信源为通道 1。

:TRIGger:USB:USBMode

命令格式

:TRIGger:USB:USBMode <String>

:TRIGger:USB:USBMode?

功能描述

设置 USB 协议触发的 USB 模式。

查询 USB 协议触发的 USB 模式。

参数说明

<String> 范围 { LOWSpeed | FULLspeed }，分别表示低速、全速。

返回格式

查询命令返回 LOWSpeed 或 FULLspeed。

实例说明

:TRIGger:USB:USBMode LOWSpeed　　设置 USB 协议触发的 USB 模式为低速。

:TRIGger:USB:USBMode? -> LOWSpeed　　查询 USB 协议触发的 USB 模式为低速。

:TRIGger:USB:LOWTrigmode

命令格式

:TRIGger:USB:LOWTrigmode <String>

:TRIGger:USB:LOWTrigmode?

功能描述

设置 USB 协议触发 USB 模式为低速时的触发模式。

查询 USB 协议触发 USB 模式为低速时的触发模式。

注：USB 模式设置为**低速**时有效，相关指令:TRIGger:USB:USBMode。

参数说明

<String> 范围 { OUT | IN | SOF | SETUp | DATA0 | DATA1 | ACK | NAK | STALl | SYNC | EOP }，分别表示输出包、输入包、起始包、建立包、DATA0 包、DATA1 包、回应包、无回应包、停止包、包同步和包结束符。

返回格式

查询命令返回 OUT、IN、SOF、SETUp、DATA0、DATA1、ACK、NAK、STALl、SYNC 或 EOP。

实例说明

:TRIGger:USB:LOWTrigmode OUT 设置 USB 协议触发 USB 模式为低速时的触发模式为输出包。

:TRIGger:USB:LOWTrigmode? -> OUT 查询 USB 协议触发 USB 模式为低速时的触发模式为输出包。

:TRIGger:USB:FULLTrigmode

命令格式

:TRIGger:USB:FULLTrigmode <String>

:TRIGger:USB:FULLTrigmode?

功能描述

设置 USB 协议触发 USB 模式为全速时的触发模式。

查询 USB 协议触发 USB 模式为全速时的触发模式。

注：USB 模式设置为**全速**时有效，相关指令:TRIGger:USB:USBMode。

参数说明

<String> 范围 { OUT | IN | SOF | SETUp | DATA0 | DATA1 | ACK | NAK | STALl | PRE | SYNC | EOP }，分别表示输出包、输入包、起始包、建立包、DATA0 包、DATA1 包、回应包、无回应包、停止包、前同步包、包同步和包结束符。

返回格式

查询命令返回 OUT、IN、SOF、SETUp、DATA0、DATA1、ACK、NAK、STALl、PRE、SYNC 或 EOP。

实例说明

:TRIGger:USB:FULLTrigmode OUT　　设置 USB 协议触发 USB 模式为全速时的触发模式为输出包。

:TRIGger:USB:FULLTrigmode? -> OUT　　查询 USB 协议触发 USB 模式为全速时的触发模式为输出包。

:TRIGger:USB:TRIGAdd

命令格式

:TRIGger:USB:TRIGAdd <String>

:TRIGger:USB:TRIGAdd?

功能描述

设置 USB 协议触发的扩展参数。

查询 USB 协议触发的扩展参数。

注：触发模式设置为**输出包**、**输入包**、**起始包**、**建立包**、**DATA0 包**和 **DATA1 包**时有效，相关指令:TRIGger:USB:LOWTrigmode 和:TRIGger:USB:FULLTrigmode。

参数说明

<String> 范围 { OFF | ON }。

返回格式

查询命令返回 OFF 或 ON。

实例说明

:TRIGger:USB:TRIGAdd OFF	设置 USB 协议触发的扩展参数为 OFF。
:TRIGger:USB:TRIGAdd? -> OFF	查询 USB 协议触发的扩展参数为 OFF。

:TRIGger:USB:ADDRval

命令格式

:TRIGger:USB:ADDRval <Int>

:TRIGger:USB:ADDRval?

功能描述

设置 USB 协议触发的地址值。

查询 USB 协议触发的地址值。

注 1：触发模式设置为**输出包**、**输入包**和**建立包**时有效，相关指令:TRIGger:USB:LOWTrigmode 和:TRIGger:USB:FULLTrigmode。

注 2：扩展参数设置为 **ON** 时有效，相关指令:TRIGger:USB:TRIGAdd。

参数说明

<Int> 范围 0x00~0x7F。

返回格式

查询命令以十六进制形式返回 USB 协议触发的地址值。

实例说明

:TRIGger:USB:ADDRval 0x01	设置 USB 协议触发的地址值为 0x01。
:TRIGger:USB:ADDRval? -> 0x01	查询 USB 协议触发的地址值为 0x01。

:TRIGger:USB:PORTval

命令格式

:TRIGger:USB:PORTval <Int>

:TRIGger:USB:PORTval?

功能描述

设置 USB 协议触发的端口值。

查询 USB 协议触发的端口值。

注 1：触发模式设置为**输出包**、**输入包**和**建立包**时有效，相关指令:TRIGger:USB:LOWTrigmode 和:TRIGger:USB:FULLTrigmode。

注 2：扩展参数设置为 **ON** 时有效，相关指令:TRIGger:USB:TRIGAdd。

参数说明

<Int> 范围 0~15。

返回格式

查询命令以整型形式返回 USB 协议触发的端口值大小。

实例说明

:TRIGger:USB:PORTval 2	设置 USB 协议触发的端口值为 2。
:TRIGger:USB:PORTval? -> 2	查询 USB 协议触发的端口值为 2。

:TRIGger:USB:FRAMenum

命令格式

:TRIGger:USB:FRAMenum <Int>

:TRIGger:USB:FRAMenum?

功能描述

设置 USB 协议触发的帧号值。

查询 USB 协议触发的帧号值。

注 1：触发模式设置为**起始包**时有效，相关指令:TRIGger:USB:LOWTrigmode 和:TRIGger:USB:FULLTrigmode。

注 2：扩展参数设置为 **ON** 时有效，相关指令:TRIGger:USB:TRIGAdd。

参数说明

<Int> 范围 0~2047。

返回格式

查询命令以整型形式返回 USB 协议触发的帧号值大小。

实例说明

:TRIGger:USB:FRAMenum 2	设置 USB 协议触发的帧号值为 2。
:TRIGger:USB:FRAMenum? -> 2	查询 USB 协议触发的帧号值为 2。

:TRIGger:USB:DATACnt

命令格式

:TRIGger:USB:DATACnt <Int>

:TRIGger:USB:DATACnt?

功能描述

设置 USB 协议触发的数据个数。

查询 USB 协议触发的数据个数。

注 1：触发模式设置为 **DATA0 包**和 **DATA1 包**时有效，相关指令:TRIGger:USB:LOWTrigmode 和:TRIGger:USB:FULLTrigmode。

注 2：扩展参数设置为 **ON** 时有效，相关指令:TRIGger:USB:TRIGAdd。

参数说明

<Int> 范围 1~10。

返回格式

查询命令以整型形式返回 USB 协议触发的数据个数大小。

实例说明

:TRIGger:USB:DATACnt 2　　设置 USB 协议触发的数据个数为 2。

:TRIGger:USB:DATACnt? -> 2　　查询 USB 协议触发的数据个数为 2。

:TRIGger:USB:DATAIndex

命令格式

:TRIGger:USB:DATAIndex <Int>

:TRIGger:USB:DATAIndex?

功能描述

设置 USB 协议触发的数据索引。

查询 USB 协议触发的数据索引。

注 1：触发模式设置为 **DATA0 包**和 **DATA1 包**时有效，相关指令:TRIGger:USB:LOWTrigmode 和:TRIGger:USB:FULLTrigmode。

注 2：扩展参数设置为 **ON** 时有效，相关指令:TRIGger:USB:TRIGAdd。

参数说明

<Int> 范围 0~9。

注：数据索引范围根据数据个数变化。

返回格式

查询命令以整数形式返回 USB 协议触发的数据索引。

实例说明

:TRIGger:USB:DATAIndex 1	设置 USB 协议触发的数据索引为 1。
:TRIGger:USB:DATAIndex? -> 1	查询 USB 协议触发的数据索引为 1。

:TRIGger:USB:TRIGData

命令格式

:TRIGger:USB:TRIGData <Int>

:TRIGger:USB:TRIGData?

功能描述

设置 USB 协议触发的触发数据。

查询 USB 协议触发的触发数据。

注 1：触发模式设置为 **DATA0 包**和 **DATA1 包**时有效，相关指令:TRIGger:USB:LOWTrigmode 和:TRIGger:USB:FULLTrigmode。

注 2：扩展参数设置为 **ON** 时有效，相关指令:TRIGger:USB:TRIGAdd。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 USB 协议触发的触发数据。

实例说明

:TRIGger:USB:TRIGData 0x10　　设置 USB 协议触发的触发数据为 0x10。

:TRIGger:USB:TRIGData? -> 0x10　　查询 USB 协议触发的触发数据为 0x10。

:TRIGger:WIEGand:DAT<x>

命令格式

:TRIGger:WIEGand:DAT<x> <String>

:TRIGger:WIEGand:DAT<x>?

功能描述

设置 Wiegand 协议触发的 DAT0 和 DAT1 的信源。

查询 Wiegand 协议触发的 DAT0 和 DAT1 的信源。

参数说明

<x> DAT 的编号，范围{ 0 | 1 }。

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:WIEGand:DAT0 CHANnel1　　设置 Wiegand 协议触发的 DAT0 信源为通道 1。

:TRIGger:WIEGand:DAT0? -> CHANnel1　查询 Wiegand 协议触发的 DAT0 信源为通道 1。

:TRIGger:WIEGand:TYPE

命令格式

:TRIGger:WIEGand:TYPE <String>

:TRIGger:WIEGand:TYPE?

功能描述

设置 Wiegand 协议触发的协议类型。

查询 Wiegand 协议触发的协议类型。

参数说明

<String> 范围 { 26bit | 39bit | 44bit | CUSTom }。

返回格式

查询命令返回 26bit、39bit、44bit 或 CUSTom。

实例说明

:TRIGger:WIEGand:TYPE CUSTom 设置 Wiegand 协议触发的协议类型为自定义。

:TRIGger:WIEGand:TYPE? -> CUSTom 查询 Wiegand 协议触发的协议类型为自定义。

:TRIGger:WIEGand:TRIGMode

命令格式

:TRIGger:WIEGand:TRIGMode <String>

:TRIGger:WIEGand:TRIGMode?

功能描述

设置 Wiegand 协议触发的触发模式。

查询 Wiegand 协议触发的触发模式。

参数说明

<String> 范围 { STARt | DATA }，分别表示开始位触发和数据触发。

返回格式

查询命令返回 STARt 或 DATA。

实例说明

:TRIGger:WIEGand:TRIGMode STARt 设置 Wiegand 协议触发的触发模式为开始位触发。

:TRIGger:WIEGand:TRIGMode? -> STARt 查询 Wiegand 协议触发的触发模式为开始位触发。

:TRIGger:WIEGand:SPACe

命令格式

:TRIGger:WIEGand:SPACe <String>

:TRIGger:WIEGand:SPACe?

功能描述

设置 Wiegand 协议触发的帧间隔。

查询 Wiegand 协议触发的帧间隔。

参数说明

<String> 范围 10.00ms~1.000s，可带单位，如“600ms”，不加单位时默认单位为秒。

返回格式

查询命令以字符串形式返回 Wiegand 协议触发的帧间隔。

实例说明

:TRIGger:WIEGand:SPACe 2e-2	设置 Wiegand 协议触发的帧间隔为 20ms。
:TRIGger:WIEGand:SPACe? -> 20.00ms	查询 Wiegand 协议触发的帧间隔为 20ms。

:TRIGger:WIEGand:MATChmode2639

命令格式

:TRIGger:WIEGand:MATChmode2639 <String>

:TRIGger:WIEGand:MATChmode2639?

功能描述

设置 Wiegand 协议触发协议类型为 26 位和 39 位时的匹配模式。

查询 Wiegand 协议触发协议类型为 26 位和 39 位时的匹配模式。

注 1：协议类型设置为 **26 位**和 **39 位**时有效，相关指令:TRIGger:WIEGand:TYPE。

注 2：触发模式设置为**数据触发**时有效，相关指令:TRIGger:WIEGand:TRIGMode。

参数说明

<String> 范围 { NONE | FC | CC | FC+CC }。

返回格式

查询命令返回 NONE、FC、CC 或 FC+CC。

实例说明

:TRIGger:WIEGand:MATChmode2639 NONE 设置 Wiegand 协议触发协议类型为 26 位和 39 位时的匹配模式为 NONE。

:TRIGger:WIEGand:MATChmode2639? -> NONE 查询 Wiegand 协议触发协议类型为 26 位和 39 位时的匹配模式为 NONE。

:TRIGger:WIEGand:MATChmode44all

命令格式

:TRIGger:WIEGand:MATChmode44all <String>

:TRIGger:WIEGand:MATChmode44all?

功能描述

设置 Wiegand 协议触发协议类型为 44 位和自定义时的匹配模式。

查询 Wiegand 协议触发协议类型为 44 位和自定义时的匹配模式。

注 1：协议类型设置为 **44 位**和**自定义**时有效，相关指令:TRIGger:WIEGand:TYPE。

注 2：触发模式设置为**数据触发**时有效，相关指令:TRIGger:WIEGand:TRIGMode。

参数说明

<String> 范围 { NONE | OEM | FC | CC | OEM+FC | OEM+CC | FC+CC | ALL }。

返回格式

查询命令返回 NONE、OEM、FC、CC、OEM+FC、OEM+CC 或 ALL。

实例说明

:TRIGger:WIEGand:MATChmode44all NONE 设置 Wiegand 协议触发协议类型为 44 位和自定义时的匹配模式为 NONE。

:TRIGger:WIEGand:MATChmode44all? -> NONE 查询 Wiegand 协议触发协议类型为 44 位和自定义时的匹配模式为 NONE。

:TRIGger:WIEGand:OEMLength

命令格式

:TRIGger:WIEGand:OEMLength <Int>

:TRIGger:WIEGand:OEMLength?

功能描述

设置 Wiegand 协议触发的 OEM 位宽。

查询 Wiegand 协议触发的 OEM 位宽。

注 1：协议类型设置为**自定义**时有效，相关指令:TRIGger:WIEGand:TYPE。

注 2：触发模式设置为**数据触发**时有效，相关指令:TRIGger:WIEGand:TRIGMode。

参数说明

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Wiegand 协议触发的 OEM 位宽。

实例说明

:TRIGger:WIEGand:OEMLength 8	设置 Wiegand 协议触发的 OEM 位宽为 8。
:TRIGger:WIEGand:OEMLength? -> 8	查询 Wiegand 协议触发的 OEM 位宽为 8

:TRIGger:WIEGand:FCLEngth

命令格式

:TRIGger:WIEGand:FCLEngth <Int>

:TRIGger:WIEGand:FCLEngth?

功能描述

设置 Wiegand 协议触发的 FC 位宽。

查询 Wiegand 协议触发的 FC 位宽。

注：协议类型设置为**自定义**时有效，相关指令:TRIGger:WIEGand:TYPE。

参数说明

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Wiegand 协议触发的 FC 位宽。

实例说明

:TRIGger:WIEGand:FCLEngth 8	设置 Wiegand 协议触发的 FC 位宽为 8。
:TRIGger:WIEGand:FCLEngth? -> 8	查询 Wiegand 协议触发的 FC 位宽为 8

:TRIGger:WIEGand:CCLEngth

命令格式

:TRIGger:WIEGand:CCLEngth <Int>

:TRIGger:WIEGand:CCLEngth?

功能描述

设置 Wiegand 协议触发的 CC 位宽。

查询 Wiegand 协议触发的 CC 位宽。

注：协议类型设置为**自定义**时有效，相关指令:TRIGger:WIEGand:TYPE。

参数说明

<Int> 范围 0~32。

返回格式

查询命令以整数形式返回 Wiegand 协议触发的 CC 位宽。

实例说明

:TRIGger:WIEGand:CCLEngth 8	设置 Wiegand 协议触发的 CC 位宽为 8。
:TRIGger:WIEGand:CCLEngth? -> 8	查询 Wiegand 协议触发的 CC 位宽为 8

:TRIGger:WIEGand:OEMValue

命令格式

:TRIGger:WIEGand:OEMValue <Int>

:TRIGger:WIEGand:OEMValue?

功能描述

设置 Wiegand 协议触发的 OEM 触发值。

查询 Wiegand 协议触发的 OEM 触发值。

注：匹配模式设置为 **OEM**、**OEM+FC**、**OEM+CC** 和 **All** 时有效，相关指令:TRIGger:WIEGand:MATChmode44all。

参数说明

<Int> 协议类型设置为 44 位，范围 0x00~0xFF；协议类型设置为自定义，范围根据 OEM 位宽值改变，如下表所示。

OEM 位宽值	OEM 触发值范围	OEM 位宽值	OEM 触发值范围	OEM 位宽值	OEM 触发值范围
0	0x00	11	0x00~0x7FF	22	0x00~0x3FFFFF
1	0x00~0x01	12	0x00~0xFFF	23	0x00~0x7FFFFF
2	0x00~0x03	13	0x00~0x1FFF	24	0x00~0xFFFFFF
3	0x00~0x07	14	0x00~0x3FFF	25	0x00~0x1FFFFFF
4	0x00~0x0F	15	0x00~0x7FFF	26	0x00~0x3FFFFFF
5	0x00~0x1F	16	0x00~0xFFFF	27	0x00~0x7FFFFFF
6	0x00~0x3F	17	0x00~0x1FFFF	28	0x00~0xFFFFFFF
7	0x00~0x7F	18	0x00~0x3FFFF	29	0x00~0x1FFFFFFF
8	0x00~0xFF	19	0x00~0x7FFFF	30	0x00~0x3FFFFFFF
9	0x00~0x1FF	20	0x00~0xFFFFF	31	0x00~0x7FFFFFFF
10	0x00~0x3FF	21	0x00~0x1FFFFF	32	0x00~0xFFFFFFFF

注：OEM 位宽值指令:TRIGger:WIEGand:OEMLength。

返回格式

查询命令以十六进制形式返回 Wiegand 协议触发的 OEM 触发值。

实例说明

:TRIGger:WIEGand:OEMValue 0x10　　设置 Wiegand 协议触发的 OEM 触发值为 0x10。

:TRIGger:WIEGand:OEMValue? -> 0x10　查询 Wiegand 协议触发的 OEM 触发值为 0x10。

:TRIGger:WIEGand:FCVAlue

命令格式

:TRIGger:WIEGand:FCVAlue <Int>

:TRIGger:WIEGand:FCVAlue?

功能描述

设置 Wiegand 协议触发的 FC 触发值。

查询 Wiegand 协议触发的 FC 触发值。

注：匹配模式设置为 **FC**、**OEM+FC**、**FC+CC** 和 **All** 时有效，相关指令:TRIGger:WIEGand:MATChmode2639 和:TRIGger:WIEGand:MATChmode44all。

参数说明

<Int> 协议类型设置为 26 位，范围 0x00~0xFF；协议类型设置为 39 位，范围 0x00~0x1FFFF；协议类型设置为 44 位，范围 0x00~0xFFF；协议类型设置为自定义，范围根据 FC 位宽值改变，如下表所示。

FC 位宽值	FC 触发值范围	FC 位宽值	FC 触发值范围	FC 位宽值	FC 触发值范围
0	0x00	11	0x00~0x7FF	22	0x00~0x3FFFFF
1	0x00~0x01	12	0x00~0xFFF	23	0x00~0x7FFFFF
2	0x00~0x03	13	0x00~0x1FFF	24	0x00~0xFFFFFF
3	0x00~0x07	14	0x00~0x3FFF	25	0x00~0x1FFFFFF
4	0x00~0x0F	15	0x00~0x7FFF	26	0x00~0x3FFFFFF
5	0x00~0x1F	16	0x00~0xFFFF	27	0x00~0x7FFFFFF
6	0x00~0x3F	17	0x00~0x1FFFF	28	0x00~0xFFFFFFF
7	0x00~0x7F	18	0x00~0x3FFFF	29	0x00~0x1FFFFFFF
8	0x00~0xFF	19	0x00~0x7FFFF	30	0x00~0x3FFFFFFF
9	0x00~0x1FF	20	0x00~0xFFFFF	31	0x00~0x7FFFFFFF
10	0x00~0x3FF	21	0x00~0x1FFFFF	32	0x00~0xFFFFFFFF

注：FC 位宽值指令:TRIGger:WIEGand:FCLEngth。

返回格式

查询命令以十六进制形式返回 Wiegand 协议触发的 FC 触发值。

实例说明

:TRIGger:WIEGand:FCVAlue 0x10　　设置 Wiegand 协议触发的 FC 触发值为 0x10。

:TRIGger:WIEGand:FCVAlue? -> 0x10　　查询 Wiegand 协议触发的 FC 触发值为 0x10。

:TRIGger:WIEGand:CCVAlue

命令格式

:TRIGger:WIEGand:CCVAlue <Int>

:TRIGger:WIEGand:CCVAlue?

功能描述

设置 Wiegand 协议触发的 CC 触发值。

查询 Wiegand 协议触发的 CC 触发值。

注：匹配模式设置为 **CC**、**OEM+CC**、**FC+CC** 和 **All** 时有效，相关指令:TRIGger:WIEGand:MATChmode2639 和:TRIGger:WIEGand:MATChmode44all。

参数说明

<Int> 协议类型设置为 26 位，范围 0x00~0xFFFF；协议类型设置为 39 位，范围 0x00~0xFFFFF；协议类型设置为 44 位，范围 0x00~0x3FFFFF；协议类型设置为自定义，范围根据 CC 位宽值改变，如下表所示。

CC 位宽值	CC 触发值范围	CC 位宽值	CC 触发值范围	CC 位宽值	CC 触发值范围
0	0x00	11	0x00~0x7FF	22	0x00~0x3FFFFF
1	0x00~0x01	12	0x00~0xFFF	23	0x00~0x7FFFFF
2	0x00~0x03	13	0x00~0x1FFF	24	0x00~0xFFFFFF
3	0x00~0x07	14	0x00~0x3FFF	25	0x00~0x1FFFFFF
4	0x00~0x0F	15	0x00~0x7FFF	26	0x00~0x3FFFFFF
5	0x00~0x1F	16	0x00~0xFFFF	27	0x00~0x7FFFFFF
6	0x00~0x3F	17	0x00~0x1FFFF	28	0x00~0xFFFFFFF
7	0x00~0x7F	18	0x00~0x3FFFF	29	0x00~0x1FFFFFFF
8	0x00~0xFF	19	0x00~0x7FFFF	30	0x00~0x3FFFFFFF
9	0x00~0x1FF	20	0x00~0xFFFFF	31	0x00~0x7FFFFFFF
10	0x00~0x3FF	21	0x00~0x1FFFFF	32	0x00~0xFFFFFFFF

注：CC 位宽值指令:TRIGger:WIEGand:CCLEngth。

返回格式

查询命令以十六进制形式返回 Wiegand 协议触发的 CC 触发值。

实例说明

:TRIGger:WIEGand:CCVAlue 0x10　　设置 Wiegand 协议触发的 CC 触发值为 0x10。

:TRIGger:WIEGand:CCVAlue? -> 0x10　　查询 Wiegand 协议触发的 CC 触发值为 0x10。

:TRIGger:WTB:SOURce

命令格式

:TRIGger:WTB:SOURce <String>

:TRIGger:WTB:SOURce?

功能描述

设置 WTB 协议触发的数据信源。

查询 WTB 协议触发的数据信源。

参数说明

<String> 范围 { CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3 或 CHANnel4。

实例说明

:TRIGger:WTB:SOURce CHANnel1	设置 WTB 协议触发数据信源为通道 1。
:TRIGger:WTB:SOURce? -> CHANnel1	查询 WTB 协议触发数据信源为通道 1。

:TRIGger:WTB:BAUDrate

命令格式

:TRIGger:WTB:BAUDrate <Double>

:TRIGger:WTB:BAUDrate?

功能描述

设置 WTB 协议触发的波特率。

查询 WTB 协议触发的波特率。

参数说明

<Double> 波特率，范围 0.00M~8.00M。

返回格式

查询命令以实型形式返回 WTB 协议触发的波特率大小。

实例说明

:TRIGger:WTB:BAUDrate 2	设置 WTB 协议触发的波特率为 2M。
:TRIGger:WTB:BAUDrate? -> 2.00	查询 WTB 协议触发的波特率为 2M。

:TRIGger:WTB:TRIGMode

命令格式

:TRIGger:WTB:TRIGMode <String>

:TRIGger:WTB:TRIGMode?

功能描述

设置 WTB 协议触发的触发模式。

查询 WTB 协议触发的触发模式。

参数说明

<String> 范围 { STARt | DATAFlg | DESTaddr | LINKctrl | SOURceaddr | DATASize }，分别表示开始位、帧数据标志、目标地址、链路控制、源地址和数据个数。

返回格式

查询命令返回 STARt、DATAFlg、DESTaddr、LINKctrl、SOURceaddr 或 DATASize。

实例说明

:TRIGger:WTB:TRIGMode STARt　　设置 WTB 协议触发的触发模式为开始位。

:TRIGger:WTB:TRIGMode? -> STARt　　查询 WTB 协议触发的触发模式为开始位。

:TRIGger:WTB:DESTaddr

命令格式

:TRIGger:WTB:DESTaddr <Int>

:TRIGger:WTB:DESTaddr?

功能描述

设置 WTB 协议触发的目标地址。

查询 WTB 协议触发的目标地址。

注：触发模式设置为**目标地址**、**链路控制**、**源地址**和**数据个数**时有效，相关指令:TRIGger:WTB:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 WTB 协议触发的目标地址。

实例说明

:TRIGger:WTB:DESTaddr 0x10　　设置 WTB 协议触发的目标地址为 0x10。

:TRIGger:WTB:DESTaddr? -> 0x10　　查询 WTB 协议触发的目标地址为 0x10。

:TRIGger:WTB:LINKctrl

命令格式

:TRIGger:WTB:LINKctrl <Int>

:TRIGger:WTB:LINKctrl?

功能描述

设置 WTB 协议触发的链路控制。

查询 WTB 协议触发的链路控制。

注：触发模式设置为**链路控制**、**源地址**和**数据个数**时有效，相关指令:TRIGger:WTB:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 WTB 协议触发的链路控制。

实例说明

:TRIGger:WTB:LINKctrl 0x10　　设置 WTB 协议触发的链路控制为 0x10。

:TRIGger:WTB:LINKctrl? -> 0x10　　查询 WTB 协议触发的链路控制为 0x10。

:TRIGger:WTB:SRCAddr

命令格式

:TRIGger:WTB:SRCAddr <Int>

:TRIGger:WTB:SRCAddr?

功能描述

设置 WTB 协议触发的源地址。

查询 WTB 协议触发的源地址。

注：触发模式设置为**源地址**和**数据个数**时有效，相关指令:TRIGger:WTB:TRIGMode。

参数说明

<Int> 范围 0x00~0xFF。

返回格式

查询命令以十六进制形式返回 WTB 协议触发的源地址。

实例说明

:TRIGger:WTB:SRCAddr 0x10　　设置 WTB 协议触发的源地址为 0x10。

:TRIGger:WTB:SRCAddr? -> 0x10　　查询 WTB 协议触发的源地址为 0x10。

:TRIGger:WTB:DATASize

命令格式

:TRIGger:WTB:DATASize <Int>

:TRIGger:WTB:DATASize?

功能描述

设置 WTB 协议触发的数据个数。

查询 WTB 协议触发的数据个数。

注：触发模式设置为**数据个数**时有效，相关指令:TRIGger:WTB:TRIGMode。

参数说明

<Int> 范围 0x00~0x80。

返回格式

查询命令以十六进制形式返回 WTB 协议触发的数据个数。

实例说明

:TRIGger:WTB:DATASize 0x10　　设置 WTB 协议触发的数据个数为 0x10。

:TRIGger:WTB:DATASize? -> 0x10　　查询 WTB 协议触发的数据个数为 0x10。

21. 趋势图相关命令

:TRENd 命令用于查询和设置趋势图配置。

- :TRENd:STATE 设置和查询趋势图使能状态。
- :TRENd:SOURce 设置和查询趋势图的信源。
- :TRENd:TYPE 设置和查询趋势图类型。
- :TRENd:THResholds:TYPE 设置和查询趋势图的阈值类型。
- :TRENd:THResholds:PERcent 设置和查询趋势图基顶基底值/最大最小值的阈值。
- :TRENd:THResholds:ABSolute 设置和查询趋势图绝对值的阈值。
- :TRENd:SCOpe:MODE 设置和查询趋势图的计算区域。
- :TRENd:DIV 设置和查询趋势图的垂直档位。
- :TRENd:OFFSET 设置和查询趋势图的垂直偏移。

:TRENd:STATE

命令格式

:TRENd:STATE <Bool>

:TRENd:STATE?

功能描述

设置趋势图使能状态。

查询趋势图使能状态。

参数说明

<Bool> 支持"ON"/"OFF"/"TRUE"/"FALSE"/"0"/"1"的 bool 类型参数。

{ON | TRUE | 1} 打开趋势图的使能；

{OFF | FALSE | 0} 关闭趋势图的使能。

返回格式

查询命令会返回两种结果，分别为：1 和 0，

对应为：打开，关闭。

实例说明

:TRENd:STATE 1	开启趋势图使能。
:TRENd:STATE -> 1	查询趋势图使能状态为打开状态。

:TRENd:SOURce

命令格式

:TRENd:SOURce <string>

:TRENd:SOURce?

功能描述

设置趋势图的信源。

查询趋势图的信源。

参数说明

<String> 范围 {CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 | M1 | M2 | M3 | M4 | R1 | R2 | R3 | R4}。

返回格式

查询命令返回 CHANnel1、CHANnel2、CHANnel3、CHANnel4、M1、M2、M3、M4、R1、R2、R3 或 R4。

实例说明

:TRENd:SOURce CHANnel1　　设置趋势图的通道源为通道 1。

:TRENd:SOURce? -> CHANnel1　　查询趋势图的通道源为通道 1。

:TRENd:TYPE

命令格式

:TRENd:TYPE <string>

:TRENd:TYPE?

功能描述

设置趋势图类型。

查询趋势图类型。

参数说明

<String> 范围 {FREQ | PERIod | DUTY}，

分别表示频率，周期，占空比。

返回格式

查询命令返回 FREQ、PERIod 或 DUTY。

实例说明

:TRENd:TYPE FREQ	设置趋势图类型为频率。
:TRENd:TYPE? -> FREQ	查询趋势图类型为频率。

:TRENd:THResholds:TYPE

命令格式

:TRENd:THResholds:TYPE <String>

:TRENd:THResholds:TYPE?

功能描述

设置趋势图的阈值类型。

查询趋势图的阈值类型。

参数说明

<String> 范围{ TOPBase | MAXMin | ABSolute }，

分别表示基顶基底值、最大最小值、绝对值。

返回格式

查询命令返回 TOPBase、MAXMin 或 ABSolute。

实例说明

:TRENd:THResholds:TYPE TOPBase 设置趋势图的阈值类型为基顶基底值。

:TRENd:THResholds:TYPE? -> TOPBase 查询趋势图的阈值类型为基顶基底值。

:TRENd:THResholds:PERcent

命令格式

:TRENd:THResholds:PERcent <String>, <UInt>

:TRENd:THResholds:PERcent? <String>

功能描述

设置趋势图基顶基底值/最大最小值的阈值。

查询趋势图基顶基底值/最大最小值的阈值。

参数说明

<String> 范围{ LOWer | MIDdle | UPper }，

分别表示低阈值、中阈值、高阈值。

<UInt> 范围 0~100，单位百分比。

返回格式

查询命令以实型形式返回趋势图基顶基底值/最大最小值的阈值大小，单位百分比。

实例说明

:TRENd:THResholds:PERcent MIDdle,50　　设置趋势图基顶基底值/最大最小值的中阈值为 50%。

:TRENd:THResholds:PERcent? MIDdle -> 50　　查询趋势图基顶基底值/最大最小值的中阈值为 50%。

:TRENd:THResholds:ABSolute

命令格式

:TRENd:THResholds:ABSolute <String>, <Double>

:TRENd:THResholds:ABSolute? <String>

功能描述

设置趋势图绝对值的阈值。

查询趋势图绝对值的阈值。

参数说明

<String>

范围{ LOWer | MIDdle | UPper }，

分别表示低阈值、中阈值、高阈值。

<Double> 阈值，范围-40000.0~40000.0。

返回格式

查询命令以实型形式返回趋势图绝对值的阈值大小。

实例说明

:TRENd:THResholds:ABSolute LOWer, -1　　设置趋势图绝对值的低阈值为-1。

:TRENd:THResholds:ABSolute? LOWer -> -1　　查询趋势图绝对值的低阈值为-1。

:TRENd:SCOpe:MODE

命令格式

:TRENd:SCOpe:MODE <String>

:TRENd:SCOpe:MODE?

功能描述

设置趋势图的计算区域。

查询趋势图的计算区域。

参数说明

<String> 范围{ RECORD | MAIN | Zoom1 | Zoom2 | CURSOR }，

分别表示全内存、主时基、Zoom1 区域、Zoom2 区域、光标区域。

返回格式

查询命令返回 RECORD、MAIN、Zoom1、Zoom2 或 CURSOR。

实例说明

:TRENd:SCOpe:MODE RECORD	设置趋势图的计算区域为全内存。
:TRENd:SCOpe:MODE? -> RECORD	查询趋势图的计算区域为全内存。

:TRENd:DIV

命令格式

:TRENd:DIV <String>,<Double>

:TRENd:DIV? <String>

功能描述

设置趋势图的垂直档位。

查询趋势图的垂直档位。

参数说明

<String> 范围 {FREQ | PERIod | DUTY}，

分别表示频率，周期，占空比。

<Double> 档位。

返回格式

查询命令以实型形式返回趋势图的垂直档位。

当**<String>**为 FREQ 时，默认单位为 Hz/div；

当**<String>**为 PERIod 时，默认单位为 s/div；

当**<String>**为 DUTY 时，默认单位为%/div。

实例说明

:TRENd:DIV FREQ,2000　　设置趋势图的垂直档位为 2kHz/div。

:TRENd:DIV? FREQ -> 2000　　查询趋势图的垂直档位为 2kHz/div。

:TRENd:OFFSET

命令格式

:TRENd:OFFSET <String>,<Double>

:TRENd:OFFSET? <String>

功能描述

设置趋势图的垂直偏移。

查询趋势图的垂直偏移。

参数说明

<String> 范围 {FREQ | PERIod | DUTY}，

分别表示频率，周期，占空比。

<Double> 偏移量。

返回格式

查询命令以实型形式返回趋势图的垂直偏移量。

当**<String>**为 FREQ 时，默认单位为 Hz；

当**<String>**为 PERIod 时，默认单位为 s；

当**<String>**为 DUTY 时，默认单位为%。

实例说明

:TRENd:OFFSET FREQ,2000	设置趋势图的垂直偏移为 2kHz。
:TRENd:OFFSET? FREQ -> 2000	查询趋势图的垂直偏移为 2kHz。

22. 读取波形相关命令

:WAVE 命令用读取波形原始数据与采样状态。

- :WAVE:READ? 读取通道波形的原始数据。

:WAVE:READ?

命令格式

:WAVE:READ? <String1>,<String2>

功能描述

读取通道波形的原始数据。

参数说明

<String1>范围范围{ CHANnel1 | CHANnel2 | CHANnel3 | CHANnel4 }；

<String2>范围{ SCREEN | MEMORY }，分别表示全屏，内存。

返回格式

查询命令返回#<dig><len><file>

数据流格式如下所示：

文件头	
数据	说明
#<dig><len>	文件头以#起始，<dig>表示后面用<dig>个 10 进制的字符来表示数据流的长度，也即<len>字节
数据流	
数据	说明
<file>	WFM 文件流

注 1：<dig>:范围 0~A，其中 A 表示 10，指定随后<len>长度；

注 2：<len>:表示二进制文件的大小;

注 3：<file>:表示 wfm 文件的二进制流。

wfm 文件流：

```
/*****************************************************************************************
**  二进制文件格式：<discribe> + <data>
*****************************************************************************************/
struct discribe
{
    char        cFileType[4];                //固定为“WFM”
    char        cDevName[64];                //设备名称
    char        cFirmwareVersion[128];       //软件版本
    char        cDataFormat[40];             //数据格式，Vx.xx
    uint32_t    uReserved;                   //保留
    uint32_t    uDataType;                   //数据类型[注]
    uint32_t    uReserved0;                  //保留
    double      dfHrztDivision;              //水平档位，单位为秒
    double      dfHrztOffset;                //水平偏移，单位为秒
    double      dfVertDivision;              //垂直档位，单位为伏
    double      dfVertOffset;                //垂直偏移，单位为伏
    double      dfStartTime;                 //数据起始时间，单位为秒
```

```
    double      dfEndTime;                  //数据结束时间，单位为秒
    double      dfSampleRate;               //采样率，单位 Hz
    double      TriggerTime;                //触发时间
    uint32_t    iDataLength;                //原始数据点数
    uint32_t    uReserved1;                 //保留
    double      dkoef;                      //数据比例系数（探头比）
    char        cUnit[64];                  //通道单位
};
```

Data 段的数据格式根据 uDataType 数据类型决定。

数据类型解析			
uDataType 值	类型	字节长度（Byte）	描述
0	uint8_t	1	原始 ADC
1	int8_t	1	原始 ADC
2	uint16_t	2	原始 ADC
3	int16_t	2	原始 ADC
4	uint32_t	4	原始 ADC
5	int32_t	4	原始 ADC
6	float	4	实际值
7	double	8	实际值

原始 ADC 数据解析

假设 ADC=X 值时，div 为垂直档位，offset 为垂直偏移，ADC 值转电压值公式如下所示：

$$\text{电压} = （\mathrm{X} - 2048） * \frac{\mathrm{div}}{400} - \mathrm{offset}$$

实例说明

当前存储深度为 100Kpts，读取通道 1 原始数据。

:WAVE:READ? CHANnel1,SCREEN -> #6200392???…

请参考 Demo 编程例程说明中的“例程 4:读取波形”。

23. :ZOOM 相关命令组

:ZOOM 命令用于查询和设置双时基（ZOOM）的配置。

- :ZOOM:SCALe 设置和查询 Zoom 的水平时基。
- :ZOOM:OFFSet 设置和查询 Zoom 窗口的水平偏移。
- :ZOOM:VERTical:SCALe 设置和查询 Zoom 的垂直缩放倍率。
- :ZOOM:VERTical:OFFSet 设置和查询 Zoom 窗口的垂直偏移像素。

:ZOOM:SCALe

命令格式

:ZOOM:SCALe <String>, <Double>

:ZOOM:SCALe? <String>

功能描述

设置 Zoom 的水平时基。

查询 Zoom 的水平时基。

参数说明

<String> 范围 { Zoom1 | Zoom2 }。

<Double> 缩放区域水平档位。

范围：500ps/div~当前主时基水平时基，用科学计数形式表示，如 2e-5 代表 20us/div 的水平档位。

注：多通道时最小时基档位为 1ns/div。

返回格式

查询命令以实型形式返回水平档位，单位 s/div。

实例说明

:ZOOM:SCALe Zoom1, 2e-5	设置 Zoom1 的水平时基为 20us/div。
:ZOOM:SCALe? Zoom1 ->2e-5	查询 Zoom1 的水平时基为 20us/div。

:ZOOM:OFFSet

命令格式

:ZOOM:OFFSet <String>, <Double>

:ZOOM:OFFSet? <String>

功能描述

设置 Zoom 窗口的水平偏移。

查询 Zoom 窗口的水平偏移。

参数说明

<String> 范围 { Zoom1 | Zoom2 }。

<Double> 偏移量，单位 s，可用科学计数形式表示，如 2e-5 代表 20us。

返回格式

查询命令以实型形式返回水平偏移，单位 s。

实例说明

:ZOOM:OFFSet Zoom1,2e-5	设置 Zoom1 的水平偏移为-20us。
:ZOOM:OFFSet? Zoom1 -> 2e-5	查询 Zoom1 的水平偏移为-20us。

:ZOOM:VERTical:SCALe

命令格式

:ZOOM:VERTical:SCALe <String>, <Double>

:ZOOM:VERTical:SCALe? <String>

功能描述

设置 Zoom 的垂直缩放倍率。

查询 Zoom 的垂直缩放倍率。

参数说明

<String> 范围 { Zoom1 | Zoom2 }。

<Double> 放大倍率，如 2.0、3.5。

返回格式

查询命令以实型形式返回垂直缩放倍率。

实例说明

:ZOOM:VERTical:SCALe Zoom1, 2.0	设置 Zoom1 的垂直缩放倍率为 2 倍。
:ZOOM:VERTical:SCALe? Zoom1 -> 2.0	查询 Zoom1 的垂直缩放倍率为 2 倍。

:ZOOM:VERTical:OFFSet

命令格式

:ZOOM:VERTical:OFFSet <String>, <Double>

:ZOOM:VERTical:OFFSet? <String>

功能描述

设置 Zoom 窗口的垂直偏移像素。

查询 Zoom 窗口的垂直偏移像素。

参数说明

<String> 范围 { Zoom1 | Zoom2 }。

<Double> 垂直偏移像素。

返回格式

查询命令以实型形式返回 Zoom 窗口的垂直偏移像素。

实例说明

:ZOOM:VERTical:OFFSet Zoom1, 0　　设置 Zoom1 窗口的垂直偏移像素为 0（此时 Zoom1 缩放窗口位于主时基中间位置）

:ZOOM:VERTical:OFFSet? Zoom1 -> 0　　查询 Zoom1 窗口的垂直偏移像素为 0（此时 Zoom1 缩放窗口位于主时基中间位置）

24. 编程例程说明

NI-VISA（Virtual Instrument Software Architecture，以下简称为 VISA）是美国国家仪器 NI（National1nstrLlrnent）公司开发的一种用来与各种仪器总线进行通信的高级应用编程接口。VISA 软件是一个综合软件包，不受平台、总线和环境的限制，可用来对 USB、GPIB、串口、VXI、PXI、USB 和以太网系统进行配置、编程和调试。

VISA 是一款可与仪器总线通信的高级应用程序接口（API）。VISA 独立于平台、总线和环境。换言之，无论是在运行 Windows 2000 操作系统的计算机上借助 LabVIEW 创建与 USB 设备通信的程序，还是在运行 MacOSX 操作系统的计算机上借助 C 创建与 GPIB 设备通信的程序，均可使用相同的 API。

注：关于 VISA 的 API 介绍可参考 ni-visa.chm 手册，位于 C:\Program Files\IVI Foundation\VISA\WinNT\NIvisa 目录下，见图 24.1 所示。

VISA 提供其它环境所需的库及头文件，使用时包含该库及头文件即可。如图 24.2 所示，其中各个文件夹包含的信息如下：

- Bin：包含动态链接库文件；
- Include：包含所需的头文件；
- Lib：包含 32 位的静态链接库；
- Lib_x64：包含 64 位的静态链接库；
- NIvisa：NI 公司提供的各类测试软件。

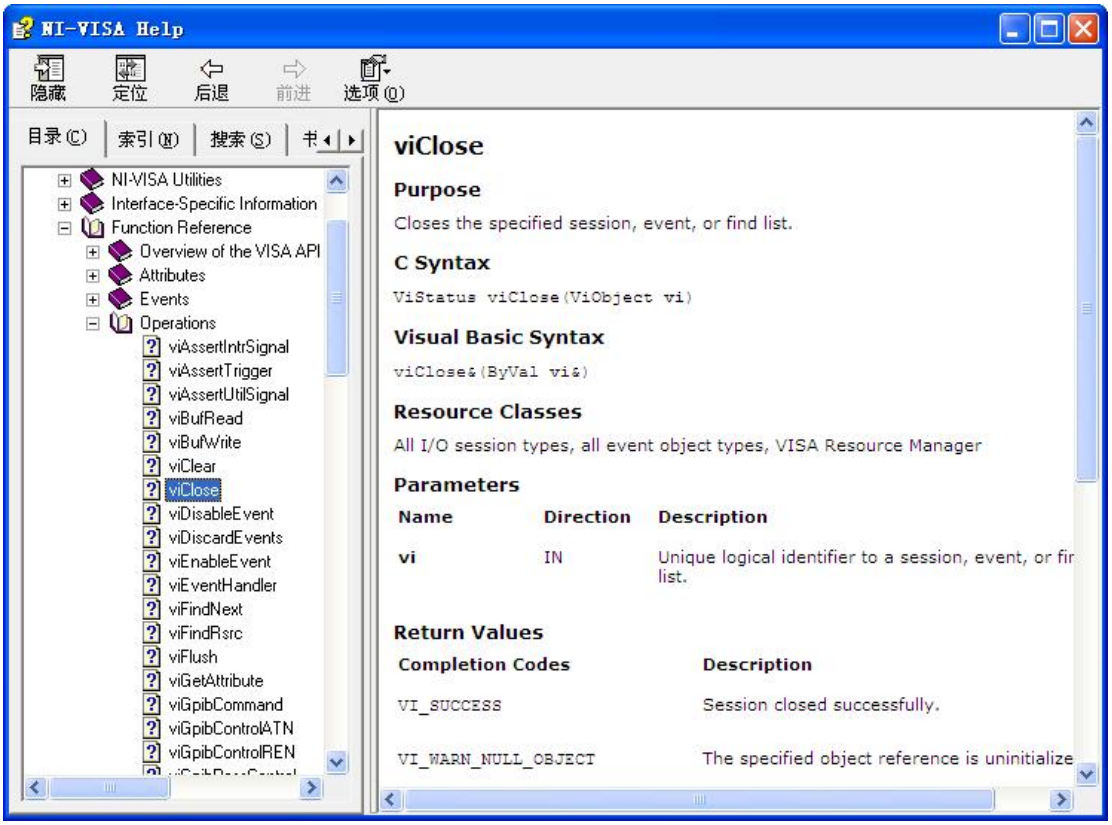

图 24.1 NI-VISA API 介绍

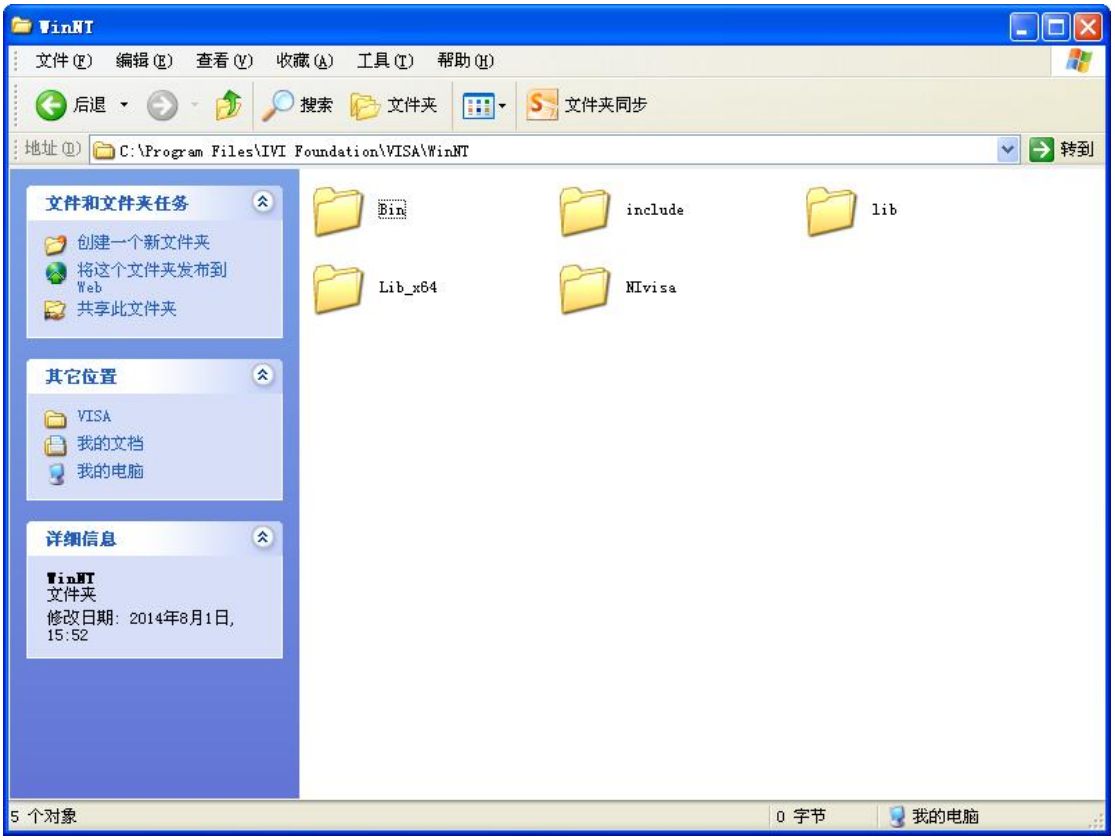

图 24.2　VISA 库及头文件信息

一个 VISA 的典型应用分为以下 4 个步骤，如图 24.3 所示。

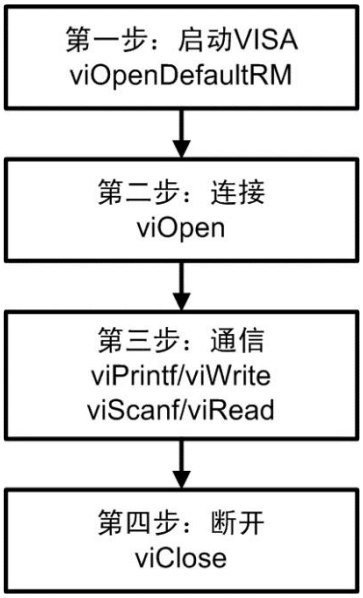

图 24.3　VISA 通信流程步骤

NI-Visa C++编程实例

程序清单 24.1 提供了一个简单的 VS 控制台示例程序，该程序展示了通过 VISA 的 USB 接口来读取设备 ID。

程序清单 24.1　VC++控制台程序

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

using namespace std;

int _tmain(int argc, _TCHAR* argv[])
{
    ViSession defaultRM, vi;
    ViByte buf[256] = {0};

    ViStatus status;
    ViChar buffer [VI_FIND_BUFLEN];
    ViRsrc matches=buffer;
    ViUInt32 nmatches;
    ViFindList list;
    viOpenDefaultRM(&defaultRM);

    try
    {



/*****************************************************************/
/* USB 操作                                                      */
/*****************************************************************/
     //获取 visa 的 USB 资源并打开
     status = viFindRsrc(defaultRM, "USB?*", &list,&nmatches,
           matches);
     if (status<VI_SUCCESS) throw 1;

     status = viOpen(defaultRM, matches, VI_NULL, VI_NULL, &vi);
    if (status<VI_SUCCESS) throw 2;



/*****************************************************************/
/* TCP 操作设备 TCP 端口为 5025                                  */
/*****************************************************************/
```

```
// 打开网络资源*/格式为 TCPIP0::IP 地址::端口::SOCKET
// status = viOpen(defaultRM,//"TCPIP0::192.168.138.46::5025::SOCKET",
     VI_NULL, VI_NULL, &vi);


    // 发送获取 ID 的命令
    status = viPrintf(vi, "*IDN?\n");
    cout<<"*IDN?\n";
    if (status<VI_SUCCESS) throw 3;

    // 读取设置 ID
    status = viScanf(vi, "%t\n", buf);
    if (status<VI_SUCCESS) throw 4;
    cout<<buf<<endl;

    // 关闭
    viClose(vi);
    viClose(defaultRM);
    }
    catch (...)
    {
       cout<<"测试错误！"<<status<<endl;
    }

    system("pause");
    return 0;
}
```

C++ 以太网通信实例

下列程序通过以太网连接示波器，实现测量、截图、保存波形等功能。

程序清单 24.2 VC++控制台程序

```
#include <iostream>
#include <fstream>
#include <string>
#include <WinSock2.h>
#pragma comment(lib, "ws2_32.lib")
using namespace std;

void main()
{
	WSADATA wsData;
	WSAStartup(MAKEWORD(2, 2), &wsData);
	SOCKET sock = socket(AF_INET, SOCK_STREAM, 0);
	SOCKADDR_IN addrSrv;
	addrSrv.sin_family = AF_INET;
	addrSrv.sin_port = htons(5025);  // 端口号：5025
	addrSrv.sin_addr.S_un.S_addr = inet_addr("192.168.138.49");
	// 连接设备
	connect(sock, (sockaddr*)&addrSrv, sizeof(addrSrv));

  // 发送*IDN 获取示波器设备信息
	char context[1024] = {};
	send(sock, "*IDN?\n", strlen("*IDN?\n"), 0);
	recv(sock, context, sizeof(context)-1, 0);
	cout << context << endl;

  // 例程 1：测量功能的设置与读取
	// 使能测量功能
	send(sock, ":MEASure:ENABle ON\n", strlen(":MEASure:ENABle ON\n"), 0);
	// 添加测量 1
	send(sock, ":MEASure1:ADD\n", strlen(":MEASure1:ADD\n"), 0);
	// 设置测量 1 类型为峰峰值
	send(sock, ":MEASure1:TYPE VPP\n", strlen(":MEASure1:TYPE VPP\n"), 0);
	// 读取测量当前值
	send(sock, ":MEASure1:CURRent?\n", strlen(":MEASure1:CURRent?\n"), 0);

	char value[1024] = {};
	recv(sock, value, sizeof(value)-1, 0);
	cout << value << endl;
```

```
//例程 2： 截图
send(sock, ":DISPlay:DATA?\n", strlen(":DISPlay:DATA?\n"), 0);
char temp[3] = {}; // 文件头以#起始，后面的那个数字表示后面用几个 10 进制的位来表示数据流的长度
recv(sock, temp, 2, 0);

int len = atoi(temp+1);
char * pic_length = new char[len];
recv(sock, pic_length, len, 0);
int bmp_length = atoi(pic_length) + 1;
delete[] pic_length;
// 获取数据并保存
int size = bmp_length > 1024*1024?1024*1024 : bmp_length;
fstream pic_bmp;
pic_bmp.open("test.bmp", ios::in | ios::out | ios::binary | ios::trunc);
char *buf1 = new char[size];
do
{
        int real_size = recv(sock, buf1, size, 0);
        bmp_length -= real_size;
        pic_bmp.write(buf1, real_size);
} while (bmp_length > 0);
pic_bmp.close();
delete[] buf1;

//例程 3： 截图 png
send(sock, ":DISPlay:DATA:PNG?\n", strlen(":DISPlay:DATA:PNG?\n"), 0);
char temp2[3] = {}; // 文件头以#起始，后面的那个数字表示后面用几个 10 进制的位来表示数据流的长度
recv(sock, temp2, 2, 0);

int len2 = atoi(temp2+1);
char *pic_length2 = new char[len2];
recv(sock, pic_length2, len2, 0);
int png_length = atoi(pic_length2) + 1;
delete[] pic_length2;
// 获取数据并保存
int size2 = png_length > 1024*1024?1024*1024 : png_length;
fstream pic_png;
pic_png.open("test.png", ios::in | ios::out | ios::binary | ios::trunc);
char *buf2 = new char[size2];
do
{
        int real_size = recv(sock, buf2, size2, 0);
```

```
                png_length -= real_size;
                pic_png.write(buf2, real_size);
        } while (png_length > 0);
        pic_png.close();
        delete buf2;

        //例程 4：  读取波形
        send(sock, ":WAVE:READ? CHANnel1,MEMORY\n", strlen(":WAVE:READ? CHANnel1,MEMORY\n"), 0);

        // 获取长度
        char temp3[3] = {}; // 文件头以#起始，后面的那个数字表示后面用几个 10 进制的位来表示数据流的长度
        recv(sock, temp3, 2, 0);

        int len3 = stoi(temp3+1, 0, 16);
        char *length_temp = new char[len3];
        recv(sock, length_temp, len3, 0);
        int wave_length = atoi(length_temp);
        delete[] length_temp;

        int size3 = wave_length > 1024*1024?1024*1024 : wave_length;

        // 获取数据并保存
        fstream wave_file;
        wave_file.open("test.wfm", ios::in | ios::out | ios::binary | ios::trunc);
        char* buf3 = new char[size3];
        do
        {
                int real_size = recv(sock, buf3, size3, 0);
                wave_length -= real_size;
                wave_file.write(buf3, real_size);
        } while (wave_length > 0);
        wave_file.close();
        delete[] buf3;
        closesocket(sock);
}
```

C# 以太网通信实例

下列 C#程序通过以太网连接示波器，设备信息的获取功能。

程序清单 24.3 C#控制台程序

```
using System;
using System.Text;
using System.Net;
using System.Net.Sockets;
namespace test
{
    class Program
    {
        static void Main(string[] args)
        {
            Socket s = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
            IPEndPoint point = new IPEndPoint(IPAddress.Parse("172.16.23.223"), 5025);
            s.Connect(point);
            s.Send(Encoding.ASCII.GetBytes("*IDN?\n"));
            byte[] receive = new byte[1024];
            s.Receive(receive);
            string str = Encoding.ASCII.GetString(receive);
            Console.WriteLine(str);
        }
    }
}
```

Python 以太网通信实例

下列程序通过以太网连接示波器，实现设备信息的获取功能。

程序清单 24.4 python 控制台程序

```
import socket
import time

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
host = "192.168.138.49"
port = 5025
addr = (host, port)
s.connect(addr)

# 发送*IDN 获取示波器设备信息
cmd = "*IDN?\n".encode()
s.send(cmd)
response=s.recv(1024).decode().strip()
print(response)

# 例程 1：测量功能的设置与读取
cmd = ":MEASure:ENABle ON\n".encode()
s.send(cmd)
cmd = ":MEASure1:ADD\n".encode()
s.send(cmd)
cmd = ":MEASure1:TYPE Ampl\n".encode()
s.send(cmd)
time.sleep(1)
cmd = ":MEASure1:CURRent?\n".encode()
s.send(cmd)
response=s.recv(1024).decode().strip()
print(response)

# 例程 2： 截图
cmd = ":DISPlay:DATA?\n".encode()
s.send(cmd)
ch = s.recv(1).decode()
if ch == "#":
    pic_length = s.recv(1).decode()
    pic_length = int(pic_length, 16)
    length = s.recv(pic_length).decode()
    length = int(length)
    size = 1024 * 1024
    f = open("test.bmp", "wb")
```

```
        while length > 0:
            buf = s.recv(size)
            length = length - len(buf)
            f.write(buf)
        f.close()

# 例程 3： 截图 png
cmd = ":DISPlay:DATA:png?\n".encode()
s.send(cmd)
ch = s.recv(1).decode()
if ch == "#":
        pic_length = s.recv(1).decode()
        pic_length = int(pic_length, 16)
        length = s.recv(pic_length).decode()
        length = int(length)
        size = 1024 * 1024
        f = open("test.png", "wb")

        while length > 0:
            buf = s.recv(size)
            length = length - len(buf)
            f.write(buf)
        f.close()

# 例程 4： 读取波形
cmd = ":WAVE:READ? CHANnel1,MEMORY\n".encode()
s.send(cmd)
ch = s.recv(1).decode()
if ch == "#":
        wave_length = s.recv(1).decode()
        wave_length = int(wave_length, 16)
        length = s.recv(wave_length).decode()
        length = int(length)
        size = 1024 * 1024
        f = open("test.wfm", "wb")

        while length > 0:
            buf = s.recv(size)
            length = length - len(buf)
            f.write(buf)
        f.close()
s.close()
```

25. 免责声明

本着为用户提供更好服务的原则，广州致远仪器有限公司（下称“致远仪器”）在本手册中将尽可能地为用户呈现详实、准确的产品信息。但介于本手册的内容具有一定的时效性，致远仪器不能完全保证该文档在任何时段的时效性与适用性。致远仪器有权在没有通知的情况下对本手册上的内容进行更新，恕不另行通知。为了得到最新版本的信息，请尊敬的用户定时访问致远仪器官方网站或者与致远仪器工作人员联系。感谢您的包容与支持！

赋能高效测试， 共创美好生活

Empower efficient testing, co-create a better life

广州致远仪器有限公司

更多详情请访问
www.zlgtmi.com

欢迎拨打全国服务热线
400-888-4005