

TO1000 系列示波器 SCPI 编程 说明书

深圳麦科信仪器有限公司

2017 年 05 月

序 言

本手册指导用户如何使用 SCPI 命令通过 USB 接口编程控制 T01000 系列数字示波器。T01000 系列 可通过 USB 与计算机进行通信。

本手册中包含的材料“按现状”提供，在将来版本中如有更改，恕不另行通知。

对于本手册可能包含的错误，或因手册所提供的信息及演绎的功能以及因使用本手册而导致的任何偶然或继发的损失，Micsig 概不负责。

目 录

1 SCPI 简介	4
1.1 命令格式	4
1.2 符号说明	4
1.3 参数类型	4
1.4 命令缩写	5
2 命令系统	6
本章主要内容	6
2.1 公用命令	7
2.2 菜单功能命令	10
2.3 采样命令子系统	12
2.4 通道命令子系统	13
2.5 数学命令子系统	16
2.6 光标命令子系统	28
2.7 显示命令子系统	32
2.8 测量命令子系统	35
2.9 触发命令子系统	46
2.10 界面命令子系统	70
2.11 时基命令子系统	72

2.12 存储命令子系统.....	74
2.13 总线配置命令子系统.....	75
2.14 PASS/FAIL 命令子系统.....	80
2.15 LAN 命令子系统.....	83
2.16 参考波形命令子系统.....	86
2.17WIFI 命令子系统.....	89
2.18 搜索命令子系统.....	92
2.19 波形命令子系统.....	114
2.20 系统.....	124
2.21 AUTO 设置子系统	126
3 编程实例	130
Excel 编程实例.....	131
Matlab 编程实例	135
LabVIEW 编程实例.....	137
Visual Basic 编程实例	143
Visual C++编程实例	145

1 SCPI 简介

SCPI (Standard Commands for Programmable Instruments 的缩写), 即程控仪器(可编程仪器)标准命令集。SCPI 是一种建立在现有标准 IEEE 488.1 和 IEEE 488.2 基础上, 并遵循了 IEEE 754 标准中浮点运算规则、ISO 646 信息交换 7 位编码符号(相当于 ASCII 编程)等多种标准的标准化仪器编程语言。SCPI 命令为树状层次结构, 包括多个子系统, 每个子系统由一个根关键字和一个或数个层次关键字构成。

本章主要内容

- 命令格式
- 符号说明
- 参数类型
- 命令缩写

1.1 命令格式

SCPI 命令为树状层次结构, 包括多个子系统, 每个子系统由一个根关键字和一个或数个层次关键字构成。命令行通常以冒号 “:” 开始; 关键字之间用冒号 “:” 分隔, 关键字后面跟随可选的参数设置; 命令行后面添加问号 “?”, 表示对此功能进行查询; 命令和参数以 “空格” 分开。

1.2 符号说明

1、 大括号 {}

大括号中的内容为参数选项。参数项之间通常用竖线 “|” 分隔。使用命令时, 必须选择其中一个参数。

2、 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。

3、 方括号 []

方括号中的内容是可省略的。

4、 三角括号 <>

三角括号中的参数必须用一个有效值来替换。

1.3 参数类型

1、 布尔型 (Bool)

参数取值为 “OFF” 、 “ON” 、 “0” 、 “1” 。

2、 离散型 (Discrete)

参数取值为所列举的选项。

3、 整形 (Integer)

除非另有说明, 参数在有效值范围内可以使任意整数 (NR1 格式)。注意, 此时请不要设置参数为小数格式, 否则将出现异常。

4、 实数 (Real)

参数在有效值范围内可以使任意实数，该命令接受小数 (NR2 格式) 和科学计数 (NR3 格式) 格式的参数输入。

5、 ASCII 字符串 (ASCII String)

参数取值为 ASCII 字符的组合。

1.4 命令缩写

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

2 命令系统

本章将逐条介绍 TO1000 命令集中各命令的格式、功能、参数以及使用说明等。

本章主要内容

- 公用命令
- 菜单功能命令
- 采样命令子系统
- 通道命令子系统
- 数学命令子系统
- 光标命令子系统
- 显示命令子系统
- 测量命令子系统
- 触发命令子系统
- 界面命令子系统
- 时基命令子系统
- 存储命令子系统
- 总线配置命令子系统
- PASS/FAIL 命令子系统
- LAN 命令子系统
- 参考波形命令子系统
- WIFI 命令子系统
- 搜索命令子系统
- 波形命令子系统
- 系统
- AUTO 设置子系统

2.1 公用命令

***CLS**

功能：清除寄存器错误状态，包括标准事件寄存器、查询事件寄存器、操作状态寄存器、状态字节寄存器。

格式：*CLS

***ESE**

功能：为标准事件寄存器组设置使能寄存器。

格式：*ESE <value>

*ESE?

其中，<value>，整型，0 至 255，判定标准事件寄存器中哪些位为 1 时，会引起状态字节寄存器中 ESC 位置 1。

说明：

标准事件寄存器的位 1 和位 6 未使用，始终视为 0，因此<value>的取值范围为 00000000（十进制 0）和 11111111（十进制 255）之间位 1 和位 6 为 0 的二进制数对应的十进制数。

ESE 寄存器各个位的定义：

位	7	6	5	4	3	2	1	0
权值	128	64	32	16	8	4	2	1
名称	PON	URQ	CME	EXE	DDE	QYE	RQL	OPC
使能	Power On	User Request	Command Error	Execution Error	Dev.Dependent Error	Query Error	Request Control	Operation Complete

返回格式：

查询返回一个整数，该整数等于寄存器中所有已设置位的权值之和。例如，如果位 3（十进制为 8）和位 5（十进制为 32）被使能，则返回“40”。

举例：

下面的命令将使能寄存器的位 3（十进制为 8）使能。

*ESE 8

下面的查询返回“8”

*ESE?

***ESR?**

功能：为标准事件寄存器组查询事件寄存器的值。

格式：*ESR?

说明：

标准事件寄存器的位 1 和位 6 未使用，始终视为 0，因此<value>的取值范围为 00000000（十进制 0）和 11111111（十进制 255）之间位 1 和位 6 为 0 的二进制数对应的十进制数。

ESE 寄存器各个位的定义：

位	7	6（未使用）	5	4	3	2	1（未使用）	0
权值	128	64	32	16	8	4	2	1
名称	PON	URQ	CME	EXE	DDE	QYE	RQL	OPC
使能	Power On	User Request	Command Error	Execution Error	Dev.Dependent Error	Query Error	Request Control	Operation Complete

返回格式：

查询返回一个整数，该整数等于寄存器中所有已设置位的权值之和。例如，如果位 3（十进制为 8）和位 5（十进制为 32）被使能，则返回“40”。

举例：

下面的查询返回“12”（位 3 和位 2 已设置）。

*ESR?

***IDN**

功能：读取示波器相关信息。包括版本号，制造商，产品型号，产品序列号。

格式：*IDN?

返回格式：

Micsig,<model>,<serial numbe>,X.X.XXX

<model>: 仪器型号。

<serial numbe>: 仪器序列号。

X.X.XXX: 仪器软件版本。

举例：

Micsig, TO202A, 232000054,4.0.155.

***OPC**

功能：当前操作完成后，标准事件寄存器的 OPC 位被置 1

格式：*OPC

***OPC?**

功能：查询当前操作是否完成。当前操作完成则返回“1”，否则返回“0”。

格式：*OPC?

***PSC**

功能：控制示波器上电时是否会产生一个服务请求。

格式：*PSC <value>

其中，<value>，离散型，{{1|ON}}|{0|OFF}}。1|ON，当示波器上电时，所有使能寄存器的值被清零。0|OFF，当示波器上电时，所有使能寄存器的值为上次保存的值。

***RST**

功能：将设备恢复到出厂默认值。

格式：*RST

***SRE**

功能：为状态字节寄存器组设置使能寄存器。

格式：*SRE <value>

***SRE?**

其中，<value>，整型，0 至 255。

说明：

状态字节寄存器的位 0 和位 1 未使用，始终视为 0，因此<value>的取值范围为 00000000 和 11111111 之间位 0 和位 1 为 0 的二进制数对应的十进制数。

SRE 寄存器各个位的定义：

位	7	6	5	4	3	2	1（未使用）	0（未使用）
权值	128	64	32	16	8	4	2	1
名称	OPER	---	ESB	MAV	---	MSG	USR	TRG
使能	Operation Status Reg	Not used	Event Status Bit	Message Available	Not used	Message	User	Trigger

返回格式：

查询返回一个整数，该整数等于寄存器中所有已设置位的权值之和。例如，如果位 3 和位 5 被使能，则返回“40”。

举例：

下面的命令将使能寄存器的位 3（十进制为 8）使能。

***SRE 8**

下面的查询返回“8”

***SRE?**

***STB?**

功能：为状态字节寄存器组查询条件寄存器。

格式：*STB?

说明：

状态字节寄存器的位 0 和位 1 未使用，始终视为 0，因此<value>的取值范围为 00000000 和 11111111 之间位 0 和位 1 为 0 的二进制数对应的十进制数。

SRE 寄存器各个位的定义：

位	7	6	5	4	3	2	1（未使用）	0（未使用）
权值	128	64	32	16	8	4	2	1
名称	OPER	---	ESB	MAV	---	MSG	USR	TRG
使能	Operation Status Reg	Not used	Event Status Bit	Message Available	Not used	Message	User	Trigger

返回格式：

查询返回一个整数，该整数等于寄存器中所有已设置位的权值之和。例如，如果位 3 和位 5 被使能，则返回“40”。

举例：

下面的查询返回“40”（位 3 和位 5 已设置）。

***SRE?**

***TST?**

功能：执行一次自检并返回自检结果。

格式：*TST?

说明：

返回值是一个十进制整数，范围为 0 至 2147483647 ($2^{31}-1$)，对应 32bit 的二进制数，每 bit 代表的自检项目如下：

BIT0: SYSTEM VOLTAGE 3.3

BIT1: ANA VOL

BIT2: STORAGE VOL

BIT3: DIG CORE VOL

BIT4: DIG PER VOL

BIT8: BATT

BIT9: FAN1

BIT10: FAN2

BIT12: TEMPE1

BIT13: TEMPE2

BIT16: TMO

注：BIT5~BIT7， BIT11， BIT14~BIT15， BIT17~BIT31 为预留位。

返回格式：

查询返回一个十进制整数，对应二进制位上为“0”表示自检通过，为“1”表示自检失败。

***WAI**

功能：等待操作完成

格式：*WAI

2.2 菜单功能命令

2.2.1 MENU:AUTO

功能：自动配置，可快速将示波器自动配置为对输入信号显示最佳效果。自动配置内容包括：适用于单个通道和多个通道；自动调整信号水平档位、垂直档位和触发电平；示波器波形反向关闭、带宽设置为全带宽、耦合方式为 DC 耦合、采样方式为正常采样；触发设置为边沿触发、触发模式为自动。

格式：MENU:AUTO

2.2.2 MENU:RUN

功能：使示波器开始运行，符合触发条件，开始采集数据。

格式：MENU:RUN

2.2.3 MENU:STOP

功能：使示波器停止运行，数据采集停止。

格式：MENU:STOP

2.2.4 MENU:SINGLE

功能：将示波器设置为单序列，示波器捕获并显示单次采集。

格式：MENU:SINGLE

2.2.5 MENU:MULTiple

功能：将示波器设置为连续触发方式。

格式: MENU:MULTiple

2.2.6 MENU:HALF

2.2.6.1 MENU:HALF:CHANnel

功能: 将通道位置设置为垂直零点位置 (波形显示区垂直中心)。

格式: MENU:HALF:CHANnel <channel>

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4|REF|MATH}。

2.2.6.2 MENU:HALF:TRIGpos

功能: 设置触发位置到屏幕中间。

格式: MENU:HALF:TRIGpos <source>

其中, <source>, 离散型, {CH1|CH2|CH3|CH4|REF|MATH}。

2.2.6.3 MENU:HALF:XCURsor

功能: 设置通道的垂直光标在 50%处。

格式: MENU:HALF:XCURsor <channel>

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4|REF|MATH}。

2.2.6.4 MENU:HALF:YCURsor

功能: 设置通道的水平光标在 50%处。

格式: MENU:HALF:YCURsor <channel>

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4|REF|MATH}。

2.2.6.5 MENU:HALF:LEVel

功能: 将触发电平设置为触发信号幅值的中间位置。

格式: MENU:HALF:LEVel [<channel>]

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4|REF|MATH}。 , 默认为当前通道。

2.2.7 MENU:HOMepage

功能: 设置示波器回到主界面。

格式: MENU:HOMepage

2.2.8 MENU:RETurn

功能: 设置退出示波器程序, 返回主界面。

格式: MENU:RETurn

2.2.9 MENU:LOCK

功能: 锁定示波器屏幕。

格式: MENU:LOCK

2.2.10 MENU:UNLock

功能: 解锁示波器屏幕。

格式: MENU:UNLock

2.2.11 MENU:COUNter

功能：频率计的打开与关闭。

格式： MENU:COUNter<bool>
MENU:COUNter?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。

返回格式：查询返回“0”或“1”。

2.2.12 MENU:MEASure

功能：打开测量界面。

格式： MENU : MEASure

2.2.13 MENU:TRIGger

功能：打开触发界面。

格式： MENU : TRIGger

2.3 采样命令子系统

2.3.1 :SAMPle:TYPE

功能：设置采样方式。

格式： :SAMPle:TYPE <type>
:SAMPle:TYPE?

其中，<type>，离散型，{NORMal|MEAN|PEAK|ENVelop|HIGHres|SEGmented}

返回格式： 查询返回“NORMal”，“MEAN”，“PEAK”，“ENVelop”，“HIGHres”，或“SEGmented”。

举例：

下面的命令选择包络采样模式。

:SAMPle:TYPE ENVelop

下面的查询返回“ENVelop”。

:SAMPle:TYPE?

2.3.2 :SAMPle:MEAN

功能：设置平均采样次数。所设置的值为2的整数倍数。

格式： :SAMPle:MEAN <count>
:SAMPle:MEAN?

其中，<count>，离散型，{2|4|8|16|32|64|128|256}

返回格式： 查询返回一个整数。

举例：

下面的命令将平均采样次数设置为“32”。

:SAMPle:MEAN 32

下面的查询返回“32”。

:SAMPle:MEAN?

2.3.3 :SAMPle:ENVelop

功能：设置包络采样次数。所设置的值为2的整数倍数或无穷。

格式： :SAMPle:ENVelop <count>
:SAMPle:ENVelop?

其中, <count>, 离散型, {2|4|8|16|32|64|128|256|inf}。

返回格式: 查询返回一个整数。

举例:

下面的命令将包络采样次数设置为“32”。

:SAMPle: ENVelop 32

下面的查询返回“32”。

:SAMPle: ENVelop?

2.3.4 :SAMPle:SEGmented

功能: 设置分段存储的段数。

格式: :SAMPle:SEGmented <segmented>

其中, <segmented>, 整型。

2.3.5 :SAMPle:SRATe

功能: 查询当前的采样率。

格式: :SAMPle:SRATe?

2.3.6 :SAMPle:MDEPth

功能: 查询示波器当前存储深度。

格式: :SAMPle:MDEPth?

2.4 通道命令子系统

2.4.1:CHANnel:DISPlay <n>,<bool>

功能: 通道的打开或关闭

格式: :CHANnel:DISPlay <n>,<bool>

: CHANnel:DISPlay? <n>

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; <bool>, 布尔型, {{0|OFF}|{1|ON}}。

返回格式: 查询返回“0”或“1”。

举例:

下面的命令打开通道 1 的显示。

:CHANnel :DIAPlay CH1 ,ON 或:CHANnel :DIAPlay CH1, 1

下面的查询返回“1”。

:CHANnel:display? CH1

2.4.2:CHANnel:INVerse

功能: 打开或关闭通道的反相显示。

格式: :CHANnel:INVerse <n>,<bool>

: CHANnel:INVerse? <n>

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; <bool>, 布尔型, {{0|OFF}|{1|ON}}。

返回格式: 查询返回“0”或“1”。

举例:

下面的命令打开通道 1 的反相显示。

:CHANnel: INVerse CH1,ON 或:CHANnel: INVerse CH1 ,1

下面的查询返回“1”。

:CHANnel: INVerse? CH1

2.4.3:CHANnel:BAND <n>

功能：设置通道的带宽限制为“20M”或“全带宽”。

格式：:CHANnel:BAND <n>,<type>

: CHANnel:BAND? <n>

参数：<n>，离散型，{CH1|CH2|CH3|CH4}; <type>，离散型，{20M|FULL|HIGH|LOW};
<freq>，实型，{30000~本机最大带宽}，只在“HIGH”“LOW”下有效。

返回格式：查询返回“20M”、“FULL”、“HIGH”、“LOW”。

举例：

下面的命令设置通道 1 的带宽限制为 20MHz。

:CHANnel: BAND CH1,20M,0.1

【注意】“0.1”可以为任意值，在 20M 和 FULL 下该值无效

:CHANnel :BAND? CH1

2.4.4 :CHANnel:PRTY <n>

功能：设置通道的探针类型为“电压”或“电流”。

格式：:CHANnel:PRTY <n>,<type>

: CHANnel:PRTY? <n>

其中，<n>，离散型，{CH1|CH2|CH3|CH4}; <type>，离散型，{VOL|CUR}。

返回格式：查询返回“VOL”或“CUR”。

举例：

下面的命令绘制通道 1 的探针类型为电压。

:CHANnel: PRTY CH1,VOL

下面的查询返回“VOL”。

:CHANnel: BAND? CH1

2.4.5 :CHANnel:PROBe <n>

功能：设置探头的衰减比。

格式：:CHANnel:PROBe <n>,<atten>

: CHANnel:PROBe? <n>

其中，<n>，离散型，{CH1|CH2|CH3|CH4}; <atten>，离散型，
{0.001|0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5|10|20|50|100|200|500|1000}。

返回格式：查询返回“0.001”，“0.002”，“0.005”，“0.01”，“0.02”，“0.05”，
“0.1”，“0.2”，“0.2”，“1”，“2”，“5”，“10”，“20”，“50”，“100”，
“200”，“500”，“1000”。

举例：

下面的命令设置通道 1 所接入探头的衰减比为 10。

:CHANnel:PROBe CH1,10

下面的查询返回“10”。

:CHANnel:PROBe? CH1

2.4.6 :CHANnel:COUPLe <n>

功能：设置通道输入耦合方式为“AC”、“DC”或“GND”。

格式: :CHANnel:COUPle <n>, <couple>

:CHANnel:COUPle? <n>

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; < couple >, 离散型, {AC|DC|GND}。

返回格式: 查询返回 “AC”, “DC” 或 “GND”。

举例:

下面的命令设置通道 1 的输入耦合方式为 “AC”。

:CHANnel:COUPle CH1,AC

下面的查询返回 “SC”。

:CHANnel:COUPle? CH1

2.4.7 :CHANnel:INPutres <n>

功能: 设置通道的输入阻抗为 “MEGA(1M Ω)” 或 “FIFTy (50 Ω)”。

格式: :CHANnel: INPutres <n>,<input>

:CHANnel<n>: INPutres?

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; < input>, 离散型, {MEGA|FIFTy}。

返回格式: 查询返回 “MEGA” 或 “FIFTy”。

举例:

下面的命令设置通道 1 的输入阻抗为 1M Ω 。

:CHANnel: INPutres CH1,MEGA

下面的查询返回 “MEGA”。

:CHANnel: INPutres? CH1

2.4.8 :CHANnel:EXTent <n>

功能: 设置指定通道波形显示的垂直档位。

格式: :CHANnel: EXTent <n>,<extent>

:CHANnel: EXTent? <n>

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; < extent >, 离散型,
{0.001|0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}*该通道的探针倍数 prob。

返回格式: 查询以科学计数形式返回垂直档位值。

举例:

下面的命令设置通道 1 的垂直档位为 1V/div。

:CHANnel:EXTent CH1,1

下面的查询返回 “1.000000e+00”。

:CHANnel:EXTent? CH1

2.4.9 :CHANnel:POSition <n>

功能: 设置指定通道波形显示的垂直偏移。

格式: :CHANnel: POSition <n>,<pos>

:CHANnel: POSition? <n>

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; <offset>, 实型。

返回格式: 查询以科学计数形式返回偏移值。

举例:

下面的命令设置通道 1 的垂直偏移为 0.01V。

:CHANnel: POSition CH1,0.01

下面的查询返回 “1.000000e-02”

:CHANnel: POSition? CH1

2.4.10 :CHANnel:VERNier <n>

功能： 打开或关闭指定通道的垂直档位微调功能。

格式： :CHANnel:VERNier <n>,<bool>

:CHANnel:VERNier? <n>

其中: <n>, 离散型, {CH1|CH2|CH3|CH4}; <bool>, 布尔型, {{0|OFF}|{1|ON}}。

返回格式： 查询返回 “0” 或 “1”。

举例：

下面的命令打开通道 1 的垂直档位微调功能。

:CHANnel: VERNier CH1,ON 或:CHANnel:VERNier CH1,1

下面的查询返回 “1”。

:CHANnel: VERNier? CH1

2.4.11 :CHANnel:PLUS:EXTent<n>, <enum>

功能： 设置指定通道波形显示的垂直档位。

格式： :CHANnel: PLUS:EXTent <n>,< enum >

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; < enum >, 枚举型, 1 为加 1 个档位, -1 为减 1 个档位;

举例：

下面的命令设置通道 1 的垂直档位在原基础上加一档。

:CHANnel: PLUS:EXTent CH1,1

2.4.12:CHANnel: PLUS :POSition <n>,< enum >

功能： 设置指定通道波形显示的垂直偏移。

格式： :CHANnel: PLUS:POSition <n>,< enum >

其中, <n>, 离散型, {CH1|CH2|CH3|CH4}; < enum >, 枚举型, 1 为加 1 个单位, -1 为减 1 个单位;

举例：

下面的命令设置通道 1 的垂直偏移在原基础上减 1 个单位。

:CHANnel: PLUS:POSition CH1,-1

2.5 数学命令子系统

2.5.1 :MATH:DISPlay

功能： 打开或关闭数学运算类型。

格式： :MATH:DISPlay <bool>

:MATH:DISPlay?

其中, <bool>, 布尔型, {{0|OFF}|{1|ON}}。

2.5.2 :MATH:MODE

功能： 选择数学运算类型。

格式： :MATH:MODE <mode>

:MATH:MODE?

其中, <mode>, 离散型, {ADD|SUB|MUL|DIV|FFT| LOGIC|FILTer|ADVAnced}。

返回格式: 查询返回“ADD”、“SUB”、“MUL”、“DIV”、“FFT”、“LOGIC”、“FILTer”、“ADVAnced”。

举例:

下面的命令选择 FFT 运算。

:MATH:MODE FFT

下面的查询返回“FFT”。

:MATH:MODE?

2.5.3 :MATH:ADD

2.5.3.1 :MATH:ADD:S1

功能: 选择加法运算的信源 1

格式: :MATH:ADD:S1 <source>

:MATH:ADD:S1?

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

返回格式: 查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例:

下面的命令选择通道 1 作为信源 1。

:MATH:ADD:S1 CH1

下面的查询返回“CH1”。

:MATH:ADD:S1?

2.5.3.2 :MATH:ADD:S2

功能: 选择加法运算的信源 2。

格式: :MATH:ADD:S2 <source>

:MATH:ADD:S2?

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

返回格式: 查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例:

下面的命令选择通道 1 作为信源 2。

:MATH:ADD:S2 CH1

下面的查询返回“CH1”。

:MATH:ADD:S2?

2.5.3.3 :MATH:ADD:EXTent

功能: 设置加法运算结果的垂直档位。

格式: :MATH:ADD:EXTent < extent >

:MATH:ADD:EXTent?

其中, < extent >, 实型, {0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式: 查询以科学计数形式返回档位值。

举例:

下面的命令设置加法运算结果的垂直档位为 1。

:MATH:ADD:EXTent 1

下面的查询返回“1.000000e+00”。

:MATH:ADD:EXTent?

2.5.3.4 :MATH:ADD:OFFSet

功能：设置加法运算结果的垂直偏移。

格式：:MATH:ADD:OFFSet <offset>

:MATH:ADD:OFFSet?

其中，<offset>，实型，

2.5.3.5 :MATH:ADD: PLUS :EXTent

功能：设置加法运算结果的垂直档位。

格式：:MATH: ADD: PLUS:EXTent <enum >

其中，<enum>，枚举型，1 为在原基础上加 1 档，-1 为减 1 档。

举例：

下面的命令设置加法运算结果的垂直档位在原基础上加 1 档。

:MATH: ADD: PLUS:EXTent 1

2.5.3.6 :MATH: ADD: PLUS: OFFSet

功能：设置加法运算结果的垂直偏移。

格式：:MATH: ADD: PLUS: OFFSet <enum >

其中，<enum>，枚举型，1 为在原基础上加 1 档，-1 为减 1 档。

举例：

下面的命令设置加法运算结果的垂直偏移在原基础上加 1 档。

:MATH: ADD: PLUS:OFFSet 1

2.5.4 :MATH:SUB

2.5.4.1 :MATH:SUB:S1

功能：选择减法运算的信源 S1.

格式：:MATH:SUB:S1 <source>

:MATH:SUB:S1?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 1.

:MATH:SUB:S1 CH1

下面的查询返回“CH1”。

:MATH:SUB:S1?

2.5.4.2 :MATH:SUB:S2

功能：选择减法运算的信源 S2.

格式：:MATH:SUB:S2 <source>

:MATH:SUB:S2?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 2.

:MATH:SUB:S2 CH1

下面的查询返回“CH1”。

:MATH:SUB:S2?

2.5.4.3 :MATH:SUB:EXTent

功能： 设置减法运算结果的垂直档位。

格式： **:MATH:SUB:EXTent <extent >**

:MATH:SUB:EXTent?

其中， < extent >， 实型， {0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式： 查询以科学计数形式返回档位值。

举例：

下面的命令设置减法运算结果的垂直档位为 1。

:MATH:SUB:EXTent 1

下面的查询返回“1.000000e+00”。

:MATH:SUB:EXTent?

2.5.4.4 :MATH:SUB:OFFSet

功能： 设置减法运算结果的垂直偏移。

格式： **:MATH:SUB:OFFSet <offset>**

:MATH:SUB:OFFSet?

其中， <offset>， 实型，

2.5.4.5 :MATH: SUB: PLUS: EXTent

功能： 设置加法运算结果的垂直档位。

格式： **:MATH: SUB: PLUS: EXTent <enum >**

其中， < enum >， 枚举型， 1 为在原基础上加 1 档， -1 为减 1 档。

举例：

下面的命令设置加法运算结果的垂直档位在原基础上加 1 档。

:MATH: SUB: PLUS: EXTent 1

功能： 设置加法运算结果的垂直偏移。

格式： **:MATH: SUB: PLUS: OFFSet <enum >**

其中， < enum >， 枚举型， 1 为在原基础上加 1 档， -1 为减 1 档。

举例：

下面的命令设置加法运算结果的垂直偏移在原基础上加 1 档。

:MATH:SUB: PLUS: OFFSet 1

2.5.5 :MATH:MUL

2.5.5.1 :MATH:MUL:S1

功能： 选择乘法运算的信源 S1.

格式： **:MATH:MUL:S1 <source>**

:MATH:MUL:S1?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 1。

:MATH:MUL:S1 CH1

下面的查询返回“CH1”。

:MATH:MUL:S1?

2.5.5.2 :MATH:MUL:S2

功能：选择乘法运算的信源 S2。

格式：:MATH:MUL:S2 <source>

:MATH:MUL:S2?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 2。

:MATH:MUL:S2 CH1

下面的查询返回“CH1”。

:MATH:MUL:S2?

2.5.5.3 :MATH:MUL:EXTent

功能：设置乘法运算结果的垂直档位。

格式：:MATH:MUL:EXTent <extent>

:MATH:MUL:EXTent?

其中，<extent>，实型，{0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式：查询以科学计数形式返回档位值。

举例：

下面的命令设置乘法运算结果的垂直档位为 1。

:MATH:MUL:EXTent 1

下面的查询返回“1.000000e+00”。

:MATH:MUL:EXTent?

2.5.5.4 :MATH:MUL:OFFSet

功能：设置乘法运算结果的垂直偏移。

格式：:MATH:MUL:OFFSet <offset>

:MATH:MUL:OFFSet?

其中，<offset>，实型，

2.5.5.5 :MATH: MUL:PLUS:EXTent

功能：设置乘法运算结果的垂直档位。

格式：:MATH: MUL: PLUS:EXTent <enum>

其中，<enum>，枚举型，1 为在原基础上加 1 档，-1 为减 1 档。

举例：

下面的命令设置乘法运算结果的垂直档位在原基础上加 1 档。

:MATH: MUL: PLUS:EXTent 1

2.5.5.6 :MATH: MUL: PLUS: OFFSet

功能：设置乘法运算结果的垂直偏移。

格式：:MATH: MUL: PLUS: OFFSet <enum>

其中，<enum>，枚举型，1 为在原基础上加 1 档，-1 为减 1 档。

举例：

下面的命令设置乘法运算结果的垂直偏移在原基础上加 1 档。

:MATH: MUL: PLUS:OFFSet 1

2.5.6 :MATH:DIV

2.5.6.1 :MATH: DIV:S1

功能：选择除法运算的信源 S1.

格式：:MATH: DIV:S1 <source>

:MATH: DIV:S1?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 1.

:MATH:DIV:S1 CH1

下面的查询返回“CH1”。

:MATH:DIV:S1?

2.5.6.2 :MATH: DIV:S2

功能：选择除法运算的信源 S2.

格式：:MATH: DIV:S2 <source>

:MATH: DIV:S2?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 2.

:MATH:DIV:S2 CH1

下面的查询返回“CH1”。

:MATH:DIV:S2?

2.5.6.3 :MATH: DIV:EXTent

功能：设置除法运算结果的垂直档位。

格式：:MATH: DIV:EXTent <extent>

:MATH: DIV:EXTent?

其中，<extent>，实型，{0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式：查询以科学计数形式返回档位值。

举例：

下面的命令设置除法运算结果的垂直档位为 1。

:MATH:DIV:EXTent 1

下面的查询返回 “1.000000e+00”。

:MATH:DIV:EXTent?

2.5.6.4 :MATH:DIV:OFFSet

功能：设置乘法运算结果的垂直偏移。

格式：**:MATH:DIV:OFFSet <offset>**

:MATH:DIV:OFFSet?

其中，<offset>，实型，

2.5.6.5 :MATH: DIV: PLUS:EXTent

功能：设置除法运算结果的垂直档位。

格式：**:MATH: DIV: PLUS:EXTent <enum >**

其中，<enum>，枚举型，1 为在原基础上加 1 档，-1 为减 1 档。

举例：

下面的命令设置除法运算结果的垂直档位在原基础上加 1 档。

:MATH: DIV: PLUS:EXTent 1

2.5.6.6 :MATH: DIV: PLUS: OFFSet

功能：设置除法运算结果的垂直偏移。

格式：**:MATH: DIV: PLUS: OFFSet <enum >**

其中，<enum>，枚举型，1 为在原基础上加 1 档，-1 为减 1 档。

举例：

下面的命令设置除法运算结果的垂直偏移在原基础上加 1 档。

:MATH: DIV: PLUS:OFFSet 1

2.5.7 :MATH:FFT

3.2.5.7.1 :MATH:FFT:SOURce

功能：选择 FFT 运算的信源。

格式：**:MATH:FFT:SOURce <source>**

:MATH:FFT:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回 “CH1”、“CH2”、“CH3” 或 “CH4”。

举例：

下面的命令选择通道 1 作为信源。

:MATH:FFT:SOURce CH1

下面的查询返回 “CH1”。

:MATH:FFT:SOURce?

2.5.7.2 :MATH:FFT:WINDow

功能：选择 FFT 运算的窗函数。

格式：**:MATH:FFT:WINDow <source>**

:MATH:FFT:WINDow?

其中, <source>, 离散型, {RECTangle|HAMMing|BLACkman|HANNing}。

返回格式: 查询返回“RECTangle”、“HAMMing”、“BLACkman”或“HANNing”。

举例:

下面的命令选择 HANNing 窗函数。

:MATH:FFT:WINDow HANNing

下面的查询返回“HANNing”。

:MATH:FFT:WINDow?

2.5.7.3 :MATH:FFT:TYPE

功能: 选择 FFT 波形的显示方式“线性”或“对数”。

格式: :MATH:FFT:TYPE <type>

:MATH:FFT:TYPE?

其中, <type>, 离散型, {LINE|DB}。

返回格式: 查询返回“LINE”或“DB”。

举例:

下面的命令选择对数显示方式。

:MATH:FFT:TYPE DB

下面的查询返回“DB”。

:MATH:FFT:TYPE?

2.5.7.4 :MATH:FFT:EXTent

功能: 设置 FFT 运算结果的垂直档位。

格式: :MATH: FFT:EXTent <extent>

:MATH: FFT:EXTent?

其中, < extent >, 实型, {0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式: 查询以科学计数形式返回档位值。

举例:

下面的命令设置 FFT 运算结果的垂直档位为 1。

:MATH: FFT:EXTent 1

下面的查询返回“1.000000e+00”。

:MATH: FFT:EXTent?

2.5.7.4 :MATH:FFT:OFFSet

功能: 设置乘法运算结果的垂直偏移。

格式: :MATH:FFT:OFFSet <offset>

:MATH:FFT:OFFSet?

其中, <offset>, 实型,

2.5.7.5 :MATH: FFT: PLUS:EXTent

功能: 设置 FFT 运算结果的垂直档位。

格式: :MATH: FFT: PLUS:EXTent <enum >

其中, < enum >, 枚举型, 1 为在原基础上加 1 档, -1 为减 1 档。

举例:

下面的命令设置 FFT 运算结果的垂直档位在原基础上加 1 档。

:MATH: FFT: PLUS:EXTent 1

2.5.7.6 :MATH: FFT: PLUS: OFFSet

功能：设置 FFT 运算结果的垂直偏移。

格式：:MATH: FFT : PLUS: OFFSet <enum>

其中，<enum>，枚举型，1 为在原基础上加 1 档，-1 为减 1 档。

举例：

下面的命令设置 FFT 运算结果的垂直偏移在原基础上加 1 档。

:MATH: FFT: PLUS:OFFSet 1

2.5.8 :MATH:LOGic

2.5.8.1 :MATH: LOGic:S1

功能：选择逻辑运算的信源 S1.

格式：:MATH: LOGic:S1 <source>

:MATH: LOGic:S1?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 1.

:MATH: LOGic:S1 CH1

下面的查询返回“CH1”。

:MATH: LOGic:S1?

说明：如果当前的逻辑运算类型为“NOT”时，则使用该命令指定相应的信源，而不是使用命令:MATH: LOGic:S2 指定的信源。

2.5.8.2 :MATH: LOGic:S2

功能：选择逻辑运算的信源 S2.

格式：:MATH: LOGic:S2 <source>

:MATH: LOGic:S2?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令选择通道 1 作为信源 2.

:MATH: LOGic:S2 CH1

下面的查询返回“CH1”。

:MATH: LOGic:S2?

说明：如果当前的逻辑运算类型为“NOT”时，则不使用该命令指定相应的信源，而是使用命令:MATH: LOGic:S1 指定的信源。

2.5.8.3 :MATH:LOGic:OPERator

功能：选择逻辑运算的逻辑状态。

格式：:MATH:LOGic:OPERator <operator>

:MATH:LOGic:OPERator?

其中，<operator>，离散型，{AND|OR|NOT|XOR}。

返回格式： 查询返回 “AND”、“OR”、“NOT” 或 “XOR”。

举例：

下面的命令选择 XOR 运算。

:MATH:LOGic:OPERator XOR

下面的查询返回 “XOR”。

:MATH:LOGic:OPERator?

2.5.8.4 :MATH: LOGic:EXTent

功能： 设置逻辑运算结果的垂直档位。

格式： :MATH: LOGic:EXTent <extent>

:MATH: LOGic:EXTent?

其中， < extent >， 实型， {0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式： 查询以科学计数形式返回档位值。

举例： 下面的命令设置逻辑运算结果的垂直档位为 1.

:MATH: LOGic:EXTent 1

下面的查询返回 “1.000000e+00”。

:MATH: LOGic:EXTent?

2.5.8.5 :MATH: LOGic:OFFSet

功能： 设置逻辑运算结果的垂直偏移。

格式： :MATH: LOGic:OFFSet <offset>

:MATH: LOGic:OFFSet?

其中， <offset>， 实型，

2.5.9 :MATH:FILTer

2.5.9.1 :MATH:FILTer:SOURce

功能： 设置数学滤波器的滤波源。

格式： :MATH:FILTer:SOURce <source>

:MATH:FILTer:SOURce?

其中， <source>， 离散型， {CH1|CH2|CH3|CH4}。

返回格式： 查询返回 “CH1”、“CH2”、“CH3” 或 “CH4 ”。

举例：

下面的命令选择通道 1 作为信源。

:MATH:FILTer:SOURce CH1

下面的查询返回 “CH1”。

:MATH:FILTer:SOURce?

2.5.9.2 :MATH:FILTer:FORMat

功能： 设置数学滤波器的滤波形式。

格式： :MATH:FILTer:FORMat <format>

:MATH:FILTer:FORMat?

其中， <format>， 离散型， {HIGHpass|LOWPass|BPASs|BSLOp}。

返回格式： 查询返回 “HIGHpass”、“LOWPass”、“BPASS” 或 “BSLOp”。

举例：

下面的命名选择 LOWPass 滤波模式。

:MATH:FILTer:FORMat LOWPass

下面的查询返回 “LOWPass”。

:MATH:FILTer:FORMat?

2.5.9.3 :MATH:FILTer:BANDwidth

功能： 设置数学滤波器的带宽。

格式： **:MATH:FILTer:BANDwidth <bandwidth>**

:MATH:FILTer:BANDwidth?

其中，<bandwidth>，整型，带宽输入范围根据示波器的带宽作为最大输入值，最小为 0Hz。

返回格式： 查询以科学计数形式返回带宽。

举例：

下面的命令设置滤波器的带宽为 1000HZ。

:MATH:FILTer:BANDwidth 1000

下面的查询返回 “1.000000e+03”。

:MATH:FILTer:BANDwidth?

2.5.9.4:MATH: FILTer:EXTent

功能： 设置数学滤波器滤波结果的垂直档位。

格式： **:MATH: FILTer:EXTent <extent>**

:MATH: FILTer:EXTent?

其中， < extent >，实型，{0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式： 查询以科学计数形式返回档位值。

举例：

下面的命令设置滤波器滤波结果的垂直档位为 1.

:MATH: FILTer:EXTent 1

下面的查询返回 “1.000000e+00”。

:MATH: FILTer:EXTent?

2.5.8.5 :MATH: FILTer:OFFSet

功能： 设置数学滤波器滤波结果的垂直偏移。

格式： **:MATH: FILTer:OFFSet <offset>**

:MATH: FILTer:OFFSet?

其中，<offset>，实型，

3.2.5.10 :MATH:ADVanced

2.5.10.1 :MATH:ADVanced:EXPRession

功能： 设置高级运算的表达式。

格式： **MATH:ADVanced:EXPRession <string>**

MATH:ADVanced:EXPRession?

其中，<string>，ASC II 字符串。

返回格式： 查询以字符串形式返回当前的表达式。

举例：

下面的命令设置表达式为“CH1+CH2”。

MATH:ADVanced:EXPRession CH1+CH2

下面的查询返回“CH1+CH2”。

MATH:ADVanced:EXPRession?

2.5.10.2 :MATH:ADVanced:VAR1

功能： 设置高级运算表达式中的变量 1。

格式： :MATH:ADVanced:VAR1 <value>

其中，<value>，实型，-9.9E+37 至 9.9E+37。

返回格式： 查询以科学计数形式返回当前变量 1 的值。

举例： 下面的命令设置变量 1 的值为 100。

:MATH:ADVanced:VAR1 100

下面的查询返回“1.000000e+02”。

:MATH:ADVanced:VARiable1?

2.5.10.3:MATH:ADVanced:VAR2

功能： 设置高级运算表达式中的变量 2

格式： :MATH:ADVanced:VAR2 <value>

其中，<value>，实型，-9.9E+37 至 9.9E+37。

返回格式： 查询以科学计数形式返回当前变量 2 的值。

举例： 下面的命令设置变量 2 的值为 100。

:MATH:ADVanced:VAR2100

下面的查询返回“1.000000e+02”。

:MATH:ADVanced:VAR2?

2.5.10.4 :MATH:ADVanced: EXTent

功能： 设置高级运算结果的垂直档位。

格式： :MATH: ADVanced:EXTent <extent>

:MATH: ADVanced:EXTent?

其中， < extent >， 实型， {0.002|0.005|0.01|0.02|0.05|0.1|0.2|0.5|1|2|5}。

返回格式： 查询以科学计数形式返回档位值。

举例：

下面的命令设置高级运算结果的垂直档位为 1。

:MATH: ADVanced:EXTent 1

下面的查询返回“1.000000e+00”。

:MATH: ADVanced:EXTent?

2.5.8.5 :MATH: ADVanced:OFFSet

功能： 设置高级运算果的垂直偏移。

格式： :MATH: ADVanced:OFFSet <offset>

:MATH: ADVanced:OFFSet?

其中， <offset>， 实型，

2.6 光标命令子系统

2.6.1 :CURSor:HORizontal

功能：打开或关闭水平光标功能。

格式：:CURSor:HORizontal <bool>

:CURSor: HORizontal?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。

2.6.2 :CURSor:VERTical

功能：打开或关闭垂直光标功能。

格式：:CURSor:VERTical <bool>

:CURSor: VERTical?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。

2.6.3 :CURSor:CX1

功能：设置垂直光标 X1 的位置。

格式：:CURSor:CX1 <px>

:CURSor:CX1?

其中，<px>，整型，以像素为单位。

返回格式：查询返回一个整数。

举例：

下面的命令设置垂直光标 X1 的水平位置为“100”。

:CURSor:CX1 100

下面的查询翻译“100”。

:CURSor:CX1?

2.6.4 :CURSor:CX2

功能：设置垂直光标 X2 的位置。

格式：:CURSor:CX2<px>

:CURSor:CX2?

其中，<px>，整型，以像素为单位。

返回格式：查询返回一个整数。

举例：

下面的命令设置垂直光标 X2 的水平位置为“100”。

:CURSor:CX2 100

下面的查询翻译“100”。

:CURSor:CX2?

2.6.5 :CURSor:CY1

功能：设置水平光标 1 的位置。

格式：:CURSor:CY1<px>

:CURSor:CY1?

其中，<px>，整型，以像素为单位。

返回格式：查询返回一个整数。

举例：

下面的命令设置水平光标 Y1 的垂直位置为 “100”。

:CURSor:CY1100

下面的查询翻译 “100”。

:CURSor:CY1?

2.6.6 :CURSor:CY2

功能：设置水平光标 2 的位置。

格式：:CURSor:CY2<px>

:CURSor:CY2?

其中，<px>，整型，以像素为单位。

返回格式：查询返回一个整数。

举例：

下面的命令设置水平光标 Y2 的垂直位置为 “100”。

:CURSor:CY2 100

下面的查询翻译 “100”。

:CURSor:CY2?

2.6.7 :CURSor:X1Value

功能：查询垂直光标 X1 的 x 值。

格式：:CURSor:X1Value?

查询值的单位由当前水平单位决定。

返回格式：查询以科学计数形式返回光标 X1 处的 X 值。

举例：

下面的查询返回 “-0.000000e-02”

:CURSor:X1Value?

2.6.8 :CURSor:X2Value

功能：查询垂直光标 X2 的 x 值。

格式：:CURSor:X2Value?

查询值的单位由当前水平单位决定。

返回格式：查询以科学计数形式返回光标 X2 处的 X 值。

举例：

下面的查询返回 “-0.000000e-02”

:CURSor:X2Value?

2.6.9 :CURSor:Y1Value

功能：查询水平光标 Y1 的 y 值。

格式：:CURSor:Y1Value?

查询值的单位由当前垂直单位决定。

返回格式：查询以科学计数形式返回光标 A 处的 Y 值。

举例：

下面的查询返回 “-0.000000e-02”

:CURSor:Y1Value?

2.6.10 :CURSor:Y2Value

功能： 查询水平光标 Y2 的 y 值。

格式： :CURSor:Y2Value?

查询值的单位由垂直单位决定。

返回格式： 查询以科学计数形式返回光标 B 处的 Y 值。

举例：

下面的查询返回 “-0.000000e-02”

:CURSor:Y2Value?

2.6.11 CURSor:XDELta

功能： 查询垂直光标 X1 和 X2 之间的差值 Δ ，单位与水平单位相同。

格式： :CURSor:XDELta?

返回格式： 查询以科学计数形式返回当前差值 ΔX 。

举例：

下面的查询返回 “1.000000e-03”。

:CURSor:XDELta?

2.6.12 :CURSor:YDELta

功能： 查询水平光标 Y1 和 Y2 之间的差值 Δ ，单位与垂直单位相同。

格式： :CURSor:YDELta?

返回格式： 查询以科学计数形式返回当前差值 ΔX 。

举例：

下面的查询返回 “1.000000e-03”。

:CURSor:YDELta?

2.6.11 CURSor:FREQ?

功能： 查询垂直光标 X1 和 X2 之间的 $1/\Delta$ ，单位为 Hz。

格式： :CURSor: FREQ?

返回格式： 查询以科学计数形式返回当前数值。

举例：

下面的查询返回 “1.000000e -03”。

:CURSor:FREQ?

2.6.13 :CURSor:RATio

功能： 查询水平光标 A 和 B 之间的差值 Δ 与垂直光标 A 和 B 之间的差值 Δ 之间的比值。

格式： :CURSor:RATio?

返回格式： 查询以科学计数形式返回的值。

举例：

下面的查询返回 “3.200000e-02”。

:CURSor:RATio?

2.6.14 CURSor:SOURce

功能： 设置光标测量的通道源。

格式： :CURSor:SOURce <source>

:CURSor:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4| REF1| REF2| REF3| REF4|MATH}。

返回格式： 查询返回 “CH1”、“CH2”、“CH3”、“CH4”、“REF1”、“REF2”、“REF3”、“REF4”、或 “MATH”。

举例：

下面的命令设置通道 1 为测量源。

:CURSor:SOURce CH1

下面的查询返回 “CH1”。

:CURSor:SOURce?

2.6.15 :CURSor:PLUS:CX1

功能： 设置垂直光标 A 的位置。

格式： :CURSor:PLUS:CXA <enum>

其中，<enum>，枚举型，1 为加 1 个单位，-1 为减 1 个单位。

举例：

下面的命令设置垂直光标 A 的水平位置减一个单位。

:CURSor: PLUS:CXA 0

2.6.16:CURSor: PLUS:CX2

功能：设置垂直光标 B 的位置。

格式：:CURSor: PLUS:CXB <enum>

其中，<enum>，枚举型，1 为加 1 个单位，-1 为减 1 个单位。

举例：

下面的命令设置垂直光标 B 的水平位置减一个单位。

:CURSor: PLUS:CXB 0

2.6.17:CURSor: PLUS:CX1

功能：设置水平光标 A 的位置。

格式：:CURSor: PLUS:CXA <enum>

其中，<enum>，枚举型，1 为加 1 个单位，-1 为减 1 个单位。

举例：

下面的命令设置水平光标 A 的垂直位置减一个单位。

:CURSor: PLUS:CXA 0

2.6.18:CURSor: PLUS:CX2

功能：设置水平光标 B 的位置。

格式：:CURSor: PLUS:CXB <enum>

其中，<enum>，枚举型，1 为加 1 个单位，-1 为减 1 个单位。

举例：

下面的命令设置水平光标 B 的垂直位置减一个单位。

:CURSor: PLUS:CXB 0

2.7 显示命令子系统

2.7.1:DISPlay:WAVEform

功能：设置屏幕中波形的显示方式，“点显示”或“线显示”。

格式：:DISPlay:WAVEform <type>

:DISPlay:WAVEform?

其中，<type>，离散型，{VECTors|DOTS}。

返回格式： 查询返回 “VECTors” 或 “DOTS”。

举例：

下面的命令设置波形显示方式为 “DOTS”。

:DISPlay:WAVeform DOTS

下面的查询返回 “DOTS”。

:DISPlay:WAVeform?

2.7.2 :DISPlay:BRIGhtness

功能： 设置屏幕中波形显示的亮度。

格式： :DISPlay:BRIGhtness <time>

:DISPlay:BRIGhtness?

其中，<time>，整型，0 至 100.

返回格式： 查询返回整数。

举例：

下面的命令设置波形显示的亮度为 80.

:DISPlay:BRIGhtness 80

下面的查询返回 “80”。

:DISPlay:BRIGhtness?

2.7.3 :DISPlay:GRATicule

功能： 设置屏幕显示的网格类型。

格式： :DISPlay:GRATicule <type>

:DISPlay:GRATicule?

其中，<type>，离散型，{FULL|GRID|RETical|FRAMe}。

返回格式： 查询返回 “FULL”、“GRID”、“RETical” 或 “FRAMe”。

举例：

下面的命令色织屏幕网格类型为 FULL.

:DISPlay:GRATicule FULL

下面的查询返回 “FULL”。

:DISPlay:GRATicule?

2.7.4 :DISPlay:INTensity

功能： 设置屏幕中网格显示的亮度。

格式： :DISPlay:INTensity <time>

:DISPlay:INTensity?

其中，<time>，整型，0 至 100.

返回格式： 查询返回整数。

举例：

下面的命令设置屏幕网格的亮度为 80.

:DISPlay: INTensity 80

下面的查询返回 “80” 。

:DISPlay: INTensity?

2.7.5 :DISPlay:PERSist

2.7.5.1 :DISPlay:PERSist:MODE

功能： 设置余辉显示模式。

格式： :DISPlay:PERSist:MODE <mode>

:DISPlay:PERSist:MODE?

其中，<mode>，离散型，{AUTO|NORMal|INFinite|none}。

2.7.5.2 :DISPlay:PERSist:ADJust

功能： 设置余辉普通显示模式下余辉时间

格式： :DISPlay:PERSist:ADJust <time>

:DISPlay:PERSist:ADJust?

其中，<time>，整型，以 ms 为单位，

100,200,300,400,500,600,700,800,900,1000,2000,3000,4000,5000,6000,7000, 8000, 9000, 10000

2.4.5.3 :DISPlay:PERSist:CLEar

功能： 清除余辉显示

格式： :DISPlay:PERSist:CLEar

2.7.6 :DISPlay:HIGH

功能：打开或关闭高刷新

格式：:DISPlay:HIGH <bool>

:DISPlay:HIGH?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。。

2.7.7 :DISPlay:HORRef

功能：设置屏幕水平展开中心模式，“触发点”或“屏幕中心”。

格式：:DISPlay:HORRef <mode>

:DISPlay:HORRef?

其中，<mode>，离散型，{CENTER|TRIGpos}。

2.7.8:DISPlay:ZOOM

功能：打开或关闭 ZOOM

格式：:DISPlay:ZOOM <bool>

:DISPlay:ZOOM?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。

2.8 测量命令子系统

:MEASure: <item>[,<src>[,<src>]] 添加指定通道的测量项到界面上

:MEASure: <item>? [,<src>[,<src>]] 查询指定通道测量项的值

2.8.1 :

:MEASure: PERiod <source>

:MEASure: PERiod?

功能：添加指定通道波形的周期测量项到界面上

查询指定通道波形的周期测量值。

格式：:MEASure: PERiod <source>

:MEASure: PERiod? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明：<source>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量记过。

举例： 下面的查询呢返回通道 1 的周期测量值，如： 1.000000e-02

:MEASure: PERiod? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: PERiod?

下面的添加通道 1 的周期测量

:MEASure: PERiod CH1

2.8.2 :MEASure:FREQ?

功能： 查询指定通道波形的频率测量值。

格式： :MEASure:FREQ? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明： <source>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量记过。

举例： 下面的查询呢返回通道 1 的频率测量值，如： 1.000000e-02

:MEASure: FREQ? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: FREQ?

2.8.3 :MEASure:RISetime?

功能： 查询指定通道波形的上升时间测量值。

格式： :MEASure:RISetime? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明： <source>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量记过。

举例： 下面的查询呢返回通道 1 的上升时间测量值，如： 1.000000e-02

:MEASure: RISetime? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: RISetime?

2.8.4 :MEASure:FALLtime?

功能： 查询指定通道波形的下降时间测量值。

格式： :MEASure:FALLtime? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明： <source>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量记过。

举例： 下面的查询呢返回通道 1 的下降时间测量值，如：1.000000e-02

:MEASure: FALLtime? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: FALLtime?

2.8.5 :MEASure:DElay?

功能： 查询通道间延迟测量的结果。

格式： :MEASure:DElay?{<sourceA>,<sourceB>}

其中，<sourceA>,<sourceB>，离散型，
{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明： <sourceA>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量结果。

举例： 下面的查询呢返回通道 1 和通道 2 的延迟测量值，如：1.000000e-02

:MEASure: DElay? CH1,CH2

如果当前示波器设置的测量源也是通道 1 和通道 2，则直接使用下面的命令：

:MEASure: DElay?

2.8.6 :MEASure:PDUTy?

功能： 查询指定通道波形的正占空比测量值。

格式： : MEASure:PDUTy? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明： <source>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量记过。

举例： 下面的查询呢返回通道 1 的正占空比测量值，如：1.000000e-02

:MEASure: PDUTy? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: PDUTy?

2.8.7 :MEASure:NDUTy?

功能： 查询指定通道波形的负占空比测量值。

格式： : MEASure:NDUTy? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明： <source>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量记过。

举例： 下面的查询呢返回通道 1 的负占空比测量值，如：1.000000e-02

:MEASure: NDUTy? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure:NDUTy?

2.8.8 :MEASure:PWIDth?

功能： 查询指定通道波形的正脉宽测量值。

格式： :MEASure:PWIDth? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明： <source>可省略，默认为示波器当前指定的通道源。

返回格式： 查询以科学计数形式返回测量记过。

举例： 下面的查询呢返回通道 1 的正脉宽测量值，如：1.000000e-02

:MEASure: PWIDth? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: PWIDth?

2.8.9 :MEASure:NWIDth?

功能： 查询指定通道波形的负脉宽测量值。

格式： :MEASure:NWIDth? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的负脉宽测量值, 如: 1.000000e-02

:MEASure: NWIDth? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure: NWIDth?

2.8.10 :MEASure:BURStw?

功能: 查询指定通道波形的突发脉冲宽度测量值。

格式: :MEASure:BURStw? <source>

其中, <source>, 离散型, {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的突发脉冲宽度测量值, 如: 1.000000e-02

:MEASure: BURStw? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure: BURStw?

2.8.11 :MEASure:ROV?

功能: 查询指定通道波形的正向超调测量值。

格式: :MEASure:ROV? <source>

其中, <source>, 离散型, 离散型, {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的正向超调测量值, 如: 1.000000e-02

:MEASure: ROV? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure: ROV?

2.8.12 :MEASure:FOV?

功能：查询指定通道波形的负向超调测量值。

格式：:MEASure:FOV? <source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明：<source>可省略，默认为示波器当前指定的通道源。

返回格式：查询以科学计数形式返回测量记过。

举例：下面的查询呢返回通道 1 的负向超调测量值，如：1.000000e-02

:MEASure: FOV? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: FOV?

2.8.13 :MEASure:PHASe?

功能：查询指定通道间相位差测量的结果。

格式：:MEASure:PHASe?{<chanA>,<chanB>}

其中，<chanA>,<chanB>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

2.8.14 :MEASure:PKPK?

功能：查询指定通道波形的峰峰值。

格式：:MEASure:PKPK?<source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明：<source>可省略，默认为示波器当前指定的通道源。

返回格式：查询以科学计数形式返回测量记过。

举例：下面的查询呢返回通道 1 的峰峰测量值，如：1.000000e-02

:MEASure: PKPK? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: PKPK?

2.8.15 :MEASure:AMP?

功能：查询指定通道波形的幅度测量值。

格式：:MEASure:AMP?<source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的峰峰测量值, 如: 1.000000e-02

:MEASure: PKPK? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure: PKPK?

2.8.16 :MEASure:HIGH?

功能: 查询指定通道波形的高值。

格式: :MEASure:HIGH?<source>

其中, <source>, 离散型, {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的高值, 如: 1.000000e-02

:MEASure: HIGH? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure: HIGH?

2.8.17 :MEASure:LOW?

功能: 查询指定通道波形的低值。

格式: :MEASure:LOW?<source>

其中, <source>, 离散型, {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的低值, 如: 1.000000e-02

:MEASure: LOW? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure: LOW?

2.8.18 :MEASure:MAX?

功能：查询指定通道波形的最大值。

格式：:MEASure:MAX?<source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明：<source>可省略，默认为示波器当前指定的通道源。

返回格式：查询以科学计数形式返回测量记过。

举例：下面的查询呢返回通道 1 的最大值，如：1.000000e-02

:MEASure: MAX? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: MAX?

2.8.19 :MEASure:MIN?

功能：查询指定通道波形的最小值。

格式：:MEASure:MIN?<source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明：<source>可省略，默认为示波器当前指定的通道源。

返回格式：查询以科学计数形式返回测量记过。

举例：下面的查询呢返回通道 1 的最小值，如：1.000000e-02

:MEASure: MIN? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: MIN?

2.8.20:MEASure:RMS?

功能：查询指定通道波形的均方根值。

格式：:MEASure:RMS?<source>

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明：<source>可省略，默认为示波器当前指定的通道源。

返回格式：查询以科学计数形式返回测量记过。

举例：下面的查询呢返回通道 1 的均方根值，如：1.000000e-02

:MEASure: RMS? CH1

如果当前示波器设置的测量源也是通道 1，则直接使用下面的命令：

:MEASure: RMS?

2.8.21 :MEASure:CRMS?

功能：查询指定通道波形的周期均方根值。

格式: :MEASure:CRMS?<source>

其中, <source>, 离散型, {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的周期均方根值, 如: 1.000000e-02

:MEASure:C RMS? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure:C RMS?

2.8.22 :MEASure:MEAN?

功能: 查询指定通道波形的平均值。

格式: :MEASure:MEAN?<source>

其中, <source>, 离散型, {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的平均值, 如: 1.000000e-02

:MEASure: MEAN? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure: MEAN?

2.8.23 :MEASure:CMEan?

功能: 查询指定通道波形的周期平均值。

格式: :MEASure:CMEan?<source>

其中, <source>, 离散型, {CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

说明: <source>可省略, 默认为示波器当前指定的通道源。

返回格式: 查询以科学计数形式返回测量记过。

举例: 下面的查询呢返回通道 1 的周期平均值, 如: 1.000000e-02

:MEASure: CMEan? CH1

如果当前示波器设置的测量源也是通道 1, 则直接使用下面的命令:

:MEASure:C MEan?

2.8.24 :MEASure:AREa

功能：查询指定通道波形的面积。

格式：:MEASure:AREa<source>

:MEASure:AREa?

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

2.8.25 :MEASure:CARea

功能：查询指定通道波形的周期面积。

格式：:MEASure:CARea <source>

:MEASure:CARea?

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

2.8.26 :MEASure:(暂无)

功能：查询指定通道波形的有功功率。

格式：:MEASure: <source>

:MEASure:AREA?

其中，<source>，离散型，{CH1|CH2|CH3|CH4|REF1|REF2|REF3|REF4|MATH}。

2.8.27 :MEASure:(暂无)

功能：查询指定通道波形的无功功率。

格式：:MEASure: <source>

:MEASure:AREA?

其中，<source>，离散型，CH1-CH4，MATH,REF。

2.8.28 :MEASure:(暂无)

功能：查询指定通道波形的视在功率。

格式：:MEASure: <source>

:MEASure:AREA?

其中，<source>，离散型，CH1-CH4，MATH,REF。

2.8.29 :MEASure:(暂无)

功能：查询指定通道波形的功率功率。

格式：:MEASure: <source>

:MEASure:AREA?

其中，<source>，离散型，CH1-CH4，MATH,REF。

2.8.30 :MEASure: (暂无)

功能：查询指定通道波形的总功率。

格式：:MEASure: <source>

:MEASure:AREA?

其中，<source>，离散型，CH1-CH4，MATH,REF。

2.8.31:MEASure:CLEAr

功能：清除打开的测量项中的任一项或所有项。

格式：:MEASure:CLEAr <item>

其中，<item>，离散型，{ITEM1|ITEM2|ITEM3|ITEM4|ITEM5|ALL}。

2.8.32 :MESAure:STATistic

2.8.32.1 :MEASure:STATistic:DISPlay

功能：打开或关闭统计功能（显示最后打开的最多 5 项测量值）。

格式：:MEASure:STATistic:DISPlay <bool>

:MEASure:STATistic:DISPlay?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。

2.8.32.2 : MEASure:STATistic:RESet

功能：清楚历史统计数据并重新统计。

格式：: MEASure:STATistic:RESet

2.8.33 :MEASure:ADISplay

功能：打开或关闭全部测量。

格式：:MEASure:ADISplay <bool>

:MEASure:ADISplay?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。

2.8.34 :MEASure:SCOPe

功能：设置测量范围为屏幕区域、光标区域或整个存储深度。

: :MEASure: SCOPe <area>

:MEASure: SCOPe?

其中，<area>，离散型，{SCReen|CREGion|DEPTh}。

2.8.35 :MEASure:COUNter:SOURce

功能： 设置或查询频率计的测量源。

:MEASure: COUNter: SOURce<sour>

:MEASure:COUNter:SOURce?

其中，<sour>，离散型，{CH1|CH2|CH3|CH4}。

2.8.36 :MEASure:COUNter:VALue?

功能： 查询频率计的测量结果,默认单位为 Hz。

:MEASure:COUNter:VALue?

查询以科学计数形式返回当前测量值。若当前未打开频率计功能，则返回 0.0000000e+00

2.9 触发命令子系统

2.9.1 :TRIGger:TYPE

功能： 选择触发类型。

格式： :TRIGger:TYPE <type>

:TRIGger:TYPE?

其中，<type>，离散型，
{EDGE|PULSe|LOGIc|Runt|B|SLOPe|TIMEout|NEDGE|SETUp|VIDEo|UART|LIN|SPI|CAN|I2C|1553B|429}。

返回格式： 查询返回当前使用的触发类型。

举例：

下面的命令选择边沿触发。

:TRIGger:TYPE EDGE

下面的查询返回“EDGE”。

:TRIGger:TYPE?

2.9.2 :TRIGger:HOLDoff

功能： 设置触发释抑时间。

格式： :TRIGger:HOLDoff <value>

:TRIGger:HOLDoff?

其中，<value>，实型，200ns 至 10s。

返回格式： 查询以科学计数形式返回触发释抑时间。

举例：

下面的命令设置触发释抑时间为 200ns

`:TRIGger:HOLDoff 0.0000002`

下面的查询返回 “2.000000e-07” 。

`:TRIGger:HOLDoff?`

2.9.3 TRIGger:MODE

功能： 设置触发方式：自动或普通。

格式： `:TRIGger:MODE <mode>`

`:TRIGger:MODE?`

其中，<mode>，离散型，{AUTO|NORMal}。

返回格式： 查询返回 “AUTO” 或 “NORMal” 。

举例：

下面的命令选择自动触发模式。

`:TRIGger:MODE AUTO`

下面的查询返回 “AUTO” 。

`:TRIGger:MODE?`

2.9.4 :TRIGger:STATus

功能： 查询当前的触发状态。

格式： `:TRIGger:STATus?`

返回格式： 查询返回 “RUN” 、 “WAIT” 、 “AUTO” 、 “STOP” 。

2.9.5 :TRIGger:EDGE

2.9.5.1 :TRIGger:EDGE:SOURce

功能： 选择边沿触发的触发源。

格式： `:TRIGger:EDGE:SOURce <source>`

`:TRIGger:EDGE:SOURce?`

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式： 查询返回 “CH1”、“CH2”、“CH3” 或 “CH4”。

举例：

下面的命令设置通道 1 为触发源。

`:TRIGger:EDGE:SOURce CH1`

下面的查询返回 “CH1”。

`:TRIGger:EDGE:SOURce?`

2.9.5.2 :TRIGger:EDGE:SLOPe

功能： 选择边沿触发的边沿类型。

格式： `:TRIGger:EDGE: SLOPe <edge>`

`:TRIGger:EDGE: SLOPe?`

其中，<edge>，离散型，{RISE|FALL|DUAL}。

返回格式： 查询返回 “RISE”、“FALL” 或 “DUAL”。

举例：

下面的命令选择上升沿触发。

`:TRIGger:EDGE: SLOPe RISE`

下面的查返回 “RISE”。

`:TRIGger:EDGE: SLOPe?`

2.9.5.3 :TRIGger:EDGE:LEVel

功能： 设置边沿触发时的触发电平

格式： `:TRIGger:EDGE:LEVel <level>`

`:TRIGger:EDGE:LEVel?`

其中，<level>，实型。

返回格式： 查询以科学计数形式返回触发电平值。

举例：

下面的命令设置触发电平为 150mV。

`:TRIGger:EDGE:LEVel 0.15`

下面的查询返回 “1.500000e-01”。

:TRIGger:EDGE:LEVel?

2.9.5.4 :TRIGger:EDGE:COUPle

功能：设置边沿触发耦合方式。

格式：:TRIGger:EDGE:COUPle <couple>

:TRIGger:EDGE:COUPle?

其中，<couple>，离散型，{DC|AC|HFRej|LFRej|Noiserej}。

返回格式：查询返回“DC”、“AC”、“HFRej”、“LFRej”或“Noiserej”。

举例：

下面的命令选择 DC 耦合方式。

:TRIGger:EDGE:COUPle DC

下面的查询返回“DC”。

:TRIGger:EDGE:COUPle?

2.9.5.5 :TRIGger:EDGE: PLUS:LEVel

功能：设置边沿触发时的触发电平

格式：:TRIGger:EDGE: PLUS:LEVel <enum>

其中，<enum>，枚举型，1 为加一个单位，-1 为减一个单位；

2.9.6 :TRIGger:PULSe

2.9.6.1 :TRIGger:PULSe:SOURce

功能：设置脉宽触发的触发源。

格式：:TRIGger:PULSe:SOURce <source>

:TRIGger:PULSe:SOURce

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式：查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令设置通道 1 为触发源。

:TRIGger: PULSe:SOURce CH1

下面的查询返回“CH1”。

:TRIGger:PULSe:SOURce?

2.9.6.2 :TRIGger:PULSe:POLarity

功能： 设置脉宽触发的极性。

格式： :TRIGger:PULSe:POLarity <polarity>

:TRIGger:PULSe:POLarity?

其中，<polarity>，离散型，{POSitive|NEGative}。

返回格式： 查询返回“POSitive”或“NEGative”。

举例：

下面的命令选择上升沿触发。

:TRIGger:PULSe:POLarity POSitive

下面的查返回“POSitive”。

:TRIGger:PULSe:POLarity?

2.9.6.3 :TRIGger:PULSe:WIDTh

功能： 设置脉宽触发时的脉冲宽度值。

格式： :TRIGger:PULSe:WIDTh<width>

:TRIGger:PULSe:WIDTh?

其中，<width>，实型，40ns 至 10s。

返回格式： 查询返回实数。

举例：

下面的命令设置脉宽值为 4ns。

:TRIGger:PULSe:WIDTh 4.000000e-08

下面的查询返回“4.000000e-08”。

:TRIGger:PULSe:WIDTh?

2.9.6.4 :TRIGger:PULSe:CONDition

功能： 设置脉宽触发条件。

格式： :TRIGger:PULSe:CONDition<condition>

:TRIGger:PULSe:CONDition?

其中，<condition>，离散型，{GREat|LESS|EQUAL|UNEQUAL}。

GREat:示波器输入信号脉宽大于指定的脉冲宽度；

LESS：示波器输入信号脉宽小于指定的脉冲宽度；

EQUAL：示波器输入信号脉宽等于指定的脉冲宽度；

UNEQUAL：示波器输入信号脉宽不等于指定的脉冲宽度；

2.9.6.5 :TRIGger:PULSe:LEVel

功能：设置脉宽触发时的触发电平

格式：:TRIGger: PULSe:LEVel <level>

:TRIGger: PULSe:LEVel?

其中，<level>，实型。

返回格式：查询以科学计数形式返回触发电平值。

举例：

下面的命令设置触发电平为 150mV。

:TRIGger: PULSe:LEVel 0.15

下面的查询返回“1.500000e-01”。

:TRIGger: PULSe:LEVel?

2.9.6.6 :TRIGger:PULSe: PLUS:LEVel

功能：设置脉宽触发时的触发电平

格式：:TRIGger: PULSe: PLUS:LEVel <enum>

其中，<enum>，枚举型，1 为加一个单位，-1 为减一个单位；

2.9.7 :TRIGger:LOGic

2.9.7.1 :TRIGger:LOGic:STATus

功能：设置逻辑触发中各通道的逻辑状态

格式：:TRIGger:LOGic:STATus<channel>,<status>

:TRIGger:LOGic:STATus? <channel>

其中，<channel>，离散型，{CH1|CH2|CH3|CH4}。<status>，离散型，{HIGH|LOW|NONE}。

2.9.7.2 :TRIGger:LOGic:FUNCTion

功能：设置逻辑触发的比较函数。

格式：:TRIGger:LOGic:FUNCTion <function>

:TRIGger:LOGic: FUNCTion?

其中，<function>，离散型，“AND”、“OR”、“NAND”或“NOR”。

2.9.7.3 :TRIGger:LOGic:CONDition

功能：设置逻辑触发条件。

格式：:TRIGger: LOGic:CONDition<condition>

:TRIGger: LOGic:CONDition?

其中，<condition>，离散型，{GREat|LESS|EQUal|UNEQual|TRUE|FALSe}。

GREat:逻辑状态为真的保持时间大于触发逻辑时间时触发；

LESS : 逻辑状态为真的保持时间小于触发逻辑时间时触发；

EQUal: 逻辑状态为真的保持时间等于触发逻辑时间时触发；

UNEQual: 逻辑状态为真的保持时间不等于触发逻辑时间时触发；

TRUL: 逻辑状态为真时触发；

FALSe: 逻辑状态为假时触发。

2.9.7.4 : TRIGger: LOGic:TIME

功能：设置触发逻辑时间。

格式：:TRIGger: LOGic:TIME<time>

:TRIGger: LOGic:TIME?

其中，<time>，实型，200ns 至 10s。

2.9.7.5 :TRIGger: LOGic:LEVel

功能：设置逻辑触发时的各通道触发电平

格式：:TRIGger: LOGic:LEVel <channel>,<level>

:TRIGger: LOGic:LEVel? <channel>

其中， <channel>,离散型，{CH1|CH2|CH3|CH4}; <level>，实型。

2.9.7.6 :TRIGger: LOGic: PLUS:LEVel

功能：设置逻辑触发时的各通道触发电平

格式: :TRIGger: LOGic: PLUS:LEVel <channel>,<enum>

其中, <channel>,离散型, {CH1|CH2|CH3|CH4}; <enum>, 枚举型, 1 为加一个单位, -1 为减一个单位;

2.9.8 :TRIGger:B

2.9.8.1 :TRIGger:B:SOURce

功能: 设置 B 触发的触发源。

格式: :TRIGger:B:SOURce <source>

:TRIGger:B:SOURce

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

2.9.8.2 :TRIGger:B:EDGE

功能: 设置 B 触发的触发斜率。

格式: :TRIGger:B:EDGE <edge>

:TRIGger:B:EDGE?

其中, <edge>, 离散型, {RISE|FALL}。

2.9.8.3 :TRIGger:B:COUPle

功能: 设置 B 触发耦合方式。

格式: :TRIGger:B:COUPle <couple>

:TRIGger:B:COUPle?

其中, <couple>, 离散型, {DC|AC|HFRej|LFRej|Noiserej}。

2.9.8.4 :TRIGger:B:SEQUence

功能: 设置 B 触发的触发类型 (B 在 A 后触发时间/事件)。

格式: :TRIGger:B:SEQUence <sequence>

:TRIGger:B:SEQUence?

其中, <sequence>, 离散型, {time|event}。

2.9.8.5 :TRIGger:B: LEVel

功能: 设置 B 触发时的触发电平

格式: :TRIGger:B:LEVel <level>

:TRIGger:B:LEVel?

其中，<level>，实型。

2.9.9 :TRIGger:Runt

2.9.9.1 :TRIGger:Runt:SOURce

功能：设置矮脉宽触发的触发源。

格式：:TRIGger:Runt:SOURce <source>

:TRIGger:Runt:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.9.9.2 :TRIGger:Runt:POLArity

功能：设置矮脉宽触发的脉冲极性。

格式：:TRIGger:Runt:POLArity <polarity>

:TRIGger:Runt:POLArity?

其中，<polarity>，离散型，{POSItive|NEGAtive}。

2.9.9.3 :TRIGger:Runt: CONDition

功能：设置脉宽限制条件。

格式：:TRIGger: Runt:CONDition<condition>

:TRIGger: Runt:CONDition?

其中，<condition>，离散型，{GREAt|LESS|BETWeen|NONE}。

GREAt:示波器输入信号脉宽大于指定的脉冲宽度；

LESS : 示波器输入信号脉宽小于指定的脉冲宽度；

BETWeen: 示波器输入信号脉宽在指定的脉冲宽度之间；

NONE: 无关；

2.9.9.4 :TRIGger:SLOPe:HTIME

功能：设置矮脉宽发时的时间上限。

格式：:TRIGger:SLOPe:HTIME <time>

:TRIGger:SLOPe:HTIME?

其中，<time>，实型，8ns 至 10s。

2.9.9.5 :TRIGger:SLOPe:LTIME

功能：设置矮脉宽触发时的时间下限。

格式: :TRIGger:SLOPe:LTIMe <time>

:TRIGger:SLOPe:LTIMe?

其中, <time>, 实型, 8ns 至 10s。

2.9.9.6 :TRIGger:SLOPe:BTIMe

功能: 设置矮脉宽触发时的时间区间。

格式: :TRIGger:SLOPe:BTIMe <htime>,<lttime>

:TRIGger:SLOPe:BTIMe? <type>

其中, <htime>,<lttime>, 实型, 8ns 至 10s。 (high>low)

< type >, 离散型, {HIGH|LOW }

2.9.9.7 :TRIGger:Runt:HLEVel

功能: 设置矮脉宽触发时的高电平。

格式: :TRIGger: Runt:HLEVEl <level>

:TRIGger: Runt:HLEVEl?

其中, <level>, 实型。

2.9.9.8 :TRIGger:Runt:LLEVel

功能: 设置矮脉宽触发时的低电平。

格式: :TRIGger: Runt:LLEVEl <level>

:TRIGger: Runt:LLEVEl?

其中, <level>, 实型。

2.9.5.9 : TRIGger:Runt: PLUS:HLEVel

功能: 设置矮脉宽触发时的高电平。

格式,:TRIGger:Runt: PLUS:HLEVel <enum>

其中, < enum >, 枚举型, 1 为加一个单位, -1 为减一个单位;

2.9.5.10 : TRIGger:Runt: PLUS:LLEVel

功能: 设置矮脉宽触发时的低电平。

格式,:TRIGger:Runt: PLUS:LLEVel <enum>

其中, < enum >, 枚举型, 1 为加一个单位, -1 为减一个单位;

2.9.10 :TRIGger:SLOPe

2.9.10.1 :TRIGger:SLOPe:SOURce

功能：设置斜率触发的触发源。

格式：:TRIGger: SLOPe:SOURce <source>

:TRIGger: SLOPe:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.9.10.2 :TRIGger:SLOPe:EDGE

功能：设置斜率触发沿。

格式：:TRIGger:SLOPe: EDGE <edge>

:TRIGger:SLOPe: EDGE?

其中，<edge>，离散型，{RISE|FALL|EITHer}。

2.9.10.3 :TRIGger: SLOPe: CONDition

功能：设置斜率触发的限制条件。

格式：:TRIGger: SLOPe:CONDition<condition>

:TRIGger: SLOPe:CONDition?

其中，<condition>，离散型，{GREat|LESS|BETWeen}。

GREat:示波器输入信号斜率大于指定的时间设置；

LESS : 示波器输入信号斜率小于指定的时间设置；

BETWeen: 示波器输入信号斜率大于指定的时间上限且小于指定的时间下限。

2.9.10.4 :TRIGger:SLOPe:HTIME

功能：设置斜率触发时的时间上限。

格式：:TRIGger:SLOPe:HTIME <time>

:TRIGger:SLOPe:HTIME?

其中，<time>，实型，8ns 至 10s。

2.9.10.5 :TRIGger:SLOPe:LTIME

功能：设置斜率触发时的时间下限。

格式：:TRIGger:SLOPe:LTIME <time>

:TRIGger:SLOPe:LTIME?

其中，<time>，实型，8ns 至 10s。

2.9.10.6 :TRIGger:SLOPe:BTIMe

功能：设置斜率触发时的时间区间。

格式：:TRIGger:SLOPe:BTIMe <htime>,<lttime>

:TRIGger:SLOPe:BTIMe? <type>

其中，<htime>,<lttime>，实型，8ns 至 10s。(high>low)

< type >，离散型，{HIGH|LOW }

2.9.10.7 :TRIGger:SLOPe:HLEVel

功能：设置斜率触发时的高电平。

格式：:TRIGger: SLOPe:HLEVEL <level>

:TRIGger: SLOPe:HLEVEL?

其中，<level>，实型。

2.9.10.8 :TRIGger:SLOPe: LLEVel

功能：设置斜率触发时的低电平。

格式：:TRIGger: SLOPe:LLEVEL <level>

:TRIGger: SLOPe:LLEVEL?

其中，<level>，实型。

2.9.10.9 :TRIGger:SLOPe: PLUS:HLEVel

功能：设置斜率触发时的高电平。

格式：:TRIGger: SLOPe: PLUS:HLEVel <enum>

其中，< enum >，枚举型，1 为加一个单位，-1 为减一个单位；

2.9.10.10 :TRIGger:SLOPe: PLUS:LLEVel

功能：设置斜率触发时的低电平。

格式：:TRIGger: SLOPe: PLUS:LLEVel <enum>

其中，< enum >，枚举型，1 为加一个单位，-1 为减一个单位；

2.9.11 :TRIGger:TIMEout

2.9.11.1 :TRIGger:TIMEout:SOURce

功能：设置超时触发的触发源。

格式：:TRIGger:TIMEout:SOURce <source>

:TRIGger:TIMEout:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}

2.9.11.2 :TRIGger:TIMEout:POLarity

功能：设置超时触发极性。

格式：:TRIGger:TIMEout:POLarity <polarity>

:TRIGger:TIMEout: POLarity?

其中，<polarity>，离散型，{POSitive|NEGative|EITHer}。

2.9.11.3 :TRIGger:TIMEout:TIME

功能：设置超时触发的超时时间。

格式：:TRIGger:TIMEout:TIME <time>

:TRIGger:TIMEout:TIME?

其中，<time>，实型，8ns 至 10s。

2.9.12 :TRIGger:NEDGE

2.9.12.1 :TRIGger:NEDGE:SOURce

功能：设置第 N 边沿触发的触发源。

格式：:TRIGger:NEDGE:SOURce <source>

:TRIGger:NEDGE:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.9.12.2 :TRIGger:NEDGE:SLOPe

功能：设置第 N 边沿触发的边沿类型。

格式：:TRIGger:NEDGE:SLOPe <slope>

:TRIGger:NEDGE:SLOPe?

其中，<slope>，离散型，{RISE|FALL}。

2.9.12.3 :TRIGger:NEDGE:IDLE

功能：设置第 N 边沿触发中开始边沿计数之前的空闲时间。

格式：:TRIGger:NEDGE:IDLE <time>

:TRIGger:NEDGE:IDLE?

其中，<time>，实型，8ns 至 10s。

2.9.12.4 :TRIGger:NEDGE:EDGE

功能：设置第 N 边沿触发的 N 的数值。

格式: :TRIGger:NEDGE:EDGE <number>

:TRIGger:NEDGE:EDGE?

其中, <number>, 实型, 1 至 65535。

2.9.12.5:TRIGger: NEDGE:LEVel

功能: 设置第 N 边沿触发时的触发电平

格式: :TRIGger: NEDGE:LEVel <level>

:TRIGger: NEDGE:LEVel?

其中, <level>, 实型。

2.9.12.6:TRIGger: NEDGE: PLUS:LEVel

功能: 设置第 N 边沿触发时的触发电平

格式: :TRIGger: NEDGE: PLUS:LEVel <enum>

其中, <enum>, 枚举型, 1 为加一个单位, -1 为减一个单位;

2.9.13 :TRIGger:SETup

2.9.13.1 :TRIGger:SETup:CLOCK

功能: 设置建立保持时间触发的时钟信号源。

格式: :TRIGger:SETup:CLOCK <source>

:TRIGger:SETup:CLOCK?

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

2.9.12 :TRIGger:SETup:DATA

功能: 设置建立保持时间触发的数据信号源。

格式: :TRIGger:SETup:DATA<source>

:TRIGger:SETup:DATA?

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

2.9.13.3 :TRIGger:SETup:CEdGe

功能: 设置建立保持时间触发的时钟边沿类型。

格式: :TRIGger:SETup:CEdGe <edge>

:TRIGger:SETup:CEdGe?

其中, <edge>, 离散型, {RISE|FALL}。

2.9.13.4 :TRIGger:SETup:STIME

功能：设置建立保持时间触发的建立时间。

格式：:TRIGger:SETup:STIME <time>

:TRIGger:SETup:STIME?

其中，<time>，实型，40ns~10s。

2.9.13.4 :TRIGger:SETup:HTIME

功能：设置建立保持时间触发的保持时间。

格式：:TRIGger:SETup:HTIME <time>

:TRIGger:SETup:HTIME?

其中，<time>，实型，40ns~10s。

2.9.13.5 :TRIGger:SETup:CLEVel

功能：设置建立保持时间触发的时钟源触发电平。

格式：:TRIGger:SETup:CLEVel <level>

:TRIGger:SETup:CLEVel?

其中，<level>，实型。

2.9.13.6 :TRIGger:SETup:DLEVel

功能：设置建立保持时间触发的数据源触发电平。

格式：:TRIGger:SETup:DLEVel <level>

:TRIGger:SETup:DLEVel?

其中，<level>，实型。

2.9.14 :TRIGger:VIDeo

2.9.14.1 :TRIGger:VIDeo:SOURce

功能：设置视频触发的触发源。

格式：:TRIGger:VIDeo:SOURce <source>

:TRIGger:VIDeo:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.9.14.2 :TRIGger:VIDeo:POLarity

功能：设置视频触发的极性

格式：:TRIGger:VIDeo:POLarity <polarity>

:TRIGger:VIDeo:POLarity?

其中，<polarity>，离散型，{POSitive|NEGAtive}。

2.9.14.3 :TRIGger:VIDeo:STANdard

功能：选择视频触发时的视频标准。

格式：:TRIGger:VIDeo:STANdard <standard>

:TRIGger:VIDeo:STANdard?

其中，<standard>，离散型，{PAL|SECAm|NTSC|720P|1080I|1080P}。

2.9.14.4 :TRIGger:VIDeo:AMODE

功能：选择触发标准为 PAL、SECAm、NTSC、1080I 时视频触发的同步类型。

格式：:TRIGger:VIDeo:AMODE <mode>

:TRIGger:VIDeo:AMODE?

其中，<mode>，离散型，{ODDField|EVENfield|AFIEld|ALIne|LINE}。

2.9.14.5 :TRIGger:VIDeo:BMODE

功能：选择触发标准为 720P、1080P 时视频触发的同步类型。

格式：:TRIGger:VIDeo:BMODE <mode>

:TRIGger:VIDeo:BMODE?

其中，<mode>，离散型，{AFIEld|ALIne|LINE}。

2.9.14.6 :TRIGger:VIDeo:AFRequence

功能：选择触发标准为 720P、1080I 时视频触发的信号频率。

格式：:TRIGger:VIDeo:AFRequence <frequency>

:TRIGger:VIDeo:AFRequence?

其中，<frequency>，离散型，{60Hz|50Hz}。

2.9.14.7 :TRIGger:VIDeo:BFRRequence

功能：选择触发标准为 1080P 时视频触发的信号频率。

格式：:TRIGger:VIDeo:BFRRequence <frequency>

:TRIGger:VIDeo:BFRRequence?

其中，<frequency>，离散型，{60Hz|50Hz|30Hz|25Hz|24Hz}。

2.9.15 :TRIGger:UART

2.9.15.1 :TRIGger:UART:SOURce

功能：设置 UART 触发的触发源。

格式：:TRIGger:UART:SOURce <source>

:TRIGger:UART:SOURce?

其中，<source>，离散型，{S1|S2}。

2.9.15.2 :TRIGger:UART:TYPE

功能：设置 UART 触发的触发条件。

格式：:TRIGger:UART:TYPE <type>

:TRIGger:UART:TYPE?

其中，<type>，离散型，{START|STOP|DATA|0:DATA|1:DATA|X:DATA|PARity}。

当总线设置中总线字长设置为 9bit 时，触发类型 DATA 不能设置；

当总线设置中总线字长设置为 5bit、6bit、7bit、8bit 时，触发类型中 0:DATA、1:DATA、X:DATA 不能进行设置。

2.9.15.3 :TRIGger:UART:RELAtion

功能：当 UART 总线触发条件选择为 DATA、0:DATA、1:DATA、X:DATA 时，设置 UART 总线触发关系。

格式：:TRIGger:UART:RELAtion <RELATION>

:TRIGger:UART:RELAtion?

其中，<RELATION>，离散型，{GREAt|LESS|EQUAL|UNEQual}。

GREAt:示波器输入数据大于指定的触发数据；

LESS：示波器输入数据小于指定的触发数据；

EQUAL: 示波器输入数据等于指定的触发数据；

UNEQual: 示波器输入数据不等于指定的触发数据；

2.9.15.4 :TRIGger: UART :DATA

功能：当 UART 总线触发条件选择为 DATA、0:DATA、1:DATA、X:DATA 时，设置 UART 总线触发数据。

格式：:TRIGger: UART :DATA <data>

:TRIGger: UART :DATA?

其中，<data>，整型，16 进制，0 至 FF。

2.9.15.5 :TRIGger:UART:LEVEL

功能：设置 UART 触发时的触发电平。

格式：:TRIGger:UART:LEVEL <level>

:TRIGger:UART:LEVEL?

其中，<level>，整型。

2.9.16 :TRIGger:LIN

2.9.16.1 :TRIGger:LIN:SOURce

功能：设置 LIN 触发的触发源。

格式：:TRIGger:LIN:SOURce <source>

:TRIGger:LIN:SOURCE?

其中，<source>，离散型，{S1|S2}。

2.9.16.2 :TRIGger:LIN:TYPE

功能：设置 LIN 触发的触发条件。

格式：:TRIGger:LIN:TYPE <type>

:TRIGger:LIN:TYPE?

其中，<type>，离散型，{SRISe|FID|IDATa}。

SRISe，同步上升沿；FID，帧 ID；IDATa，帧 ID 和数据。

2.9.16.3 :TRIGger:LIN:ID

功能：当 LIN 总线触发条件为 FID 或 IDATa 时，设置 LIN 触发的触发 ID 值。

格式：:TRIGger:LIN:ID <data>

:TRIGger:LIN:ID?

其中，<data>，整型，16 进制，0 至 3F。

2.9.16.4 :TRIGger:LIN:DATA

功能：当 LIN 总线触发条件为 IDATa 时，设置 LIN 触发的触发数据。

格式：:TRIGger:LIN:DATA <data>

:TRIGger:LIN:DATA?

其中，<data>，整型，16 进制，0 至 FFFF,FFFF,FFFF,FFFF。

2.9.16.5 :TRIGger:LIN:LEVel

功能：设置 LIN 触发时的触发电平。

格式: :TRIGger: LIN:LEVel <level>

:TRIGger: LIN:LEVel?

其中, <level>, 整型。

2.9.17 :TRIGger:CAN

2.9.17.1 :TRIGger:CAN:SOURce

功能: 设置 CAN 触发的触发源。

格式: :TRIGger:CAN:SOURce <source>

:TRIGger:CAN:SOURce?

其中, <source>, 离散型, {S1|S2}。

2.9.17.2 :TRIGger:CAN:TYPE

功能: 设置 CAN 触发的触发条件

格式: :TRIGger:CAN:TYPE <type>

:TRIGger:CAN:TYPE?

其中, <type>, 离散型,
{FSTArt|RFID|DFID|RDID|IDATa|WRFR|AERRor|ACKError|OVERload}。

FSTArt, 帧起始; RFID, 远程帧 ID; DFID 数据帧 ID; RDID, 远程帧/数据帧 ID; IDATa, 数据帧 ID 和数据; WRFR, 错误帧; AERRor, 所有错误; ACKError, 确认错误; OVERload, 过载帧。

2.9.17.3 :TRIGger:CAN:ID

功能: 当 CAN 触发的触发条件为 RFID、DFID、IDATa 或 RDID 时, 设置 CAN 触发的触发 ID 值。

格式: :TRIGger:CAN:ID <data>

:TRIGger:CAN:ID?

其中, <data>, 整型, 16 进制, 0 至 FFFF,FFFF。

2.9.17.4 :TRIGger:CAN:DLC

功能: 当 CAN 触发的触发条件为 IDATa 时, 设置 CAN 触发的 DLC 值。

格式: :TRIGger:CAN:DLC <data>

:TRIGger:CAN:DLC?

其中, <data>, 整型, 0 至 8。

2.9.17.5 :TRIGger:CAN:DATA

功能：当 CAN 触发的触发条件为 IDATa 时，设置 CAN 触发的触发数据值。

格式：:TRIGger:CAN:DATA <data>

:TRIGger:CAN:DATA?

其中，<data>，整型，16 进制，数据位数由 DLC 确定。

2.9.17.6 :TRIGger:CAN:LEVel

功能：设置 CAN 触发时的触发电平。

格式：:TRIGger:CAN:LEVel <level>

:TRIGger:CAN:LEVel?

其中，<level>，整型。

2.9.18 :TRIGger:SPI

2.9.18.1 :TRIGger:SPI:DATA

功能：设置 SPI 触发下的数据值。

格式：:TRIGger:SPI:DATA <data>

:TRIGger:SPI:DATA?

其中，<data>，整型，二进制。

2.9.18.2 :TRIGger: SPI:SOURce

功能：设置 SPI 触发的触发源。

格式：:TRIGger: SPI:SOURce <source>

:TRIGger: SPI:SOURce?

其中，<source>，，离散型，{S1|S2}。

2.9.18.3 :TRIGger: SPI:LEVel

功能：设置 SPI 触发时的触发电平。

格式：:TRIGger: SPI:LEVel <level>

:TRIGger: SPI:LEVe?

其中，<level>，整型。

2.9.19 :TRIGger:IIC

2.9.19.1 :TRIGger:IIC:SOURce

功能：设置 IIC 触发的触发源。

格式：:TRIGger:IIC:SOURce <source>

:TRIGger:IIC:SOURce?

其中，<source>，，离散型，{S1|S2}。

2.9.19.2 :TRIGger:IIC:TYPE

功能：设置 IIC 触发的触发类型。

格式：:TRIGger:IIC:TYPE <type>

:TRIGger:IIC:TYPE?

其中，<type>，离散型，
{START|STOP|ACKLost|NACKaddress|REStart|RDATA|FRAM1|FRAM2}。

START，起始条件；STOP，停止条件；ACKLost，确认丢失；NACKaddress，地址字段无确认；REStart，重新启动；RDATA，EEPROM 数据读取；FRAM1，帧型 1；FRAM2，帧型 2。

2.9.19.3 :TRIGger:IIC:ADDRess

功能：当 IIC 触发条件为 NACKaddress、FRAM1 或 FRAM2 时，设置 IIC 总线触发的触发地址。

格式：:TRIGger:IIC:ADDRess <data>

:TRIGger:IIC:ADDRess?

其中，<data>，整型，16 进制，0 至 FF。

2.9.19.4 :TRIGger:IIC:RELation

功能：当 IIC 触发条件为 RDATA 时，设置 IIC 总线触发的触发关系。

格式：:TRIGger:IIC:RELation <relation>

:TRIGger:IIC:RELation

其中，<RELATION>，离散型，{GREAt|LESS|EQUAL|UNEQual}。

GREAt:示波器输入数据大于指定的触发数据；

LESS：示波器输入数据小于指定的触发数据；

EQUAL:示波器输入数据等于指定的触发数据；

UNEQual:示波器输入数据不等于指定的触发数据；

2.9.19.5 :TRIGger:IIC:DATA

功能：当 IIC 触发条件为 RDATA、FRAM1 或 FRAM2 时，设置 IIC 总线触发的触发数据。

格式: :TRIGger:IIC:DATA <data>

:TRIGger:IIC:DATA?

其中, <data>, 整型, 十六进制。

2.9.19.6 :TRIGger: IIC:LEVel

功能: 设置 IIC 触发时的触发电平。

格式: :TRIGger: IIC:LEVel <level>

:TRIGger: IIC:LEVel?

其中, <level>, 整型。

2.9.20 :TRIGger:1553B

2.9.20.1 :TRIGger:1553B:SOURce

功能: 设置 1553B 总线触发的触发源。

格式: :TRIGger:1553B:SOURce <source>

:TRIGger:1553B:SOURce?

其中, <source>, , 离散型, {S1|S2}。

2.9.20.2 :TRIGger:1553B:TYPE

功能: 设置 1553B 总线触发的触发条件。

格式: :TRIGger:1553B:TYPE <type>

:TRIGger:1553B:TYPE?

其中, <type>, 离散型,
{CSSYnc|DWSYnc|CSWOrd|DWORD|RTADdress|OPERror|MERRor|AERRor}。

CSSYnc, 指令/状态字同步头; DWSYnc, 数据字同步头; CSWOrd, 指令/状态字;
DWORD, 数据字; RTADdress, 远程终端地址; OPERror, 奇校验错误; MERRor, 曼彻
斯特码错误; AERRor, 所有错误。

2.9.20.3 :TRIGger:1553B:CSWOrd

功能: 当 1553B 触发条件为 CSWOrd 时, 设置 1553B 总线触发的指令/状态字数值。

格式: :TRIGger:1553B:CSWOrd <data>

:TRIGger:1553B:CSWOrd?

其中, <data>, 整型, 0 至 FFFF。

2.9.20.4 :TRIGger:1553B: DWORD

功能：当 1553B 触发条件为 DWORD 时，设置 1553B 总线触发的触发数据值。

格式：:TRIGger:1553B: DWORD <data>

:TRIGger:1553B: DWORD?

其中，<data>，整型，0 至 FFFF。

2.9.20.5 :TRIGger:1553B: RTADdress

功能：当 1553B 总线触发条件为 RTADdress 时，设置 1553B 总线触发的远程终端地址。

格式：:TRIGger:1553B: RTADdress<address>

:TRIGger:1553B: RTADdress?

其中，<address>，整型，0 至 FF。

2.9.20.6 :TRIGger: 1553B:LEVEL

功能：设置 1553B 触发时的触发电平。

格式：:TRIGger: 1553B:LEVEL <level>

:TRIGger: 1553B:LEVEL?

其中，<level>，整型。

2.9.21 :TRIGger:429

2.9.21.1 :TRIGger:429:SOURce

功能：设置 429 总线触发的触发源。

格式：:TRIGger:429:SOURce <source>

:TRIGger:429:SOURce?

其中，<source>，，离散型，{S1|S2}。

2.9.21.2 :TRIGger:429:TYPE

功能：设置 429 总线触发的触发条件。

格式：:TRIGger:429:TYPE <type>

:TRIGger:429:TYPE?

其中，<type>，离散型，

{WBEGin|WEND|WORD|LABEL|SDI|DATA|SSM|LSDI|LDATa|LSSM|WERRor|WINTerval|VE
RRor|AERRor|ALLO|ALL1}。

WBEGin, 字起始; WEND, 字结束; WORD, 全字匹配; LSDI, LABEL+SDI; LDATa, LABEL+DATA; LSSM, LABEL+SSM; WERRor, 字错误; WINTerval, 字间隙错误; VERRor, 校验错误; AERRor, 所有错误; ALLO, 所有 0 位; ALL1, 所有 1 位。

2.9.21.3 :TRIGger:429:WORD

功能: 当 429 总线触发条件为 WORD 时, 设置 429 总线触发的触发字值。

格式: :TRIGger:429:WORD <data>

:TRIGger:429:WORD?

其中, <data>, 整型, 0 至 FFFFFFFF。

2.9.21.4 :TRIGger:429:LABEL

功能: 当 429 总线触发条件为 LABEL、LSDI、LDATa 或 LSSM 时, 设置 429 总线触发的触发 LABEL 值。

格式: :TRIGger:429:LABEL <data>

:TRIGger:429:LABEL?

其中, <data>, 整型, 八进制, 0 至 3FF。

2.9.21.5 :TRIGger:429:SDI

功能: 当 429 总线触发条件为 SDI 或 LSDI 时, 设置 429 总线触发的触发 SDI 值。

格式: :TRIGger:429:SDI <data>

:TRIGger:429:SDI?

其中, <data>, 整型, 0 至 3。

2.9.21.6 :TRIGger:429:DATA

功能: 当 429 总线触发条件为 DATA 或 LDATa 时, 设置 429 总线触发的触发数据值。

格式: :TRIGger:429:DATA <data>

:TRIGger:429:DATA?

其中, <data>, 整型, 0 至 FFFFFFFF。

2.9.21.7 :TRIGger:429:SSM

功能: 当 429 总线触发条件为 SSM 或 LSSM 时, 设置 429 总线触发的触发数据值。

格式: :TRIGger:429:SSM <data>

:TRIGger:429:SSM?

其中, <data>, 整型, 0 至 3。

2.9.21.8 :TRIGger: 429:LEVEL

功能：设置 429 触发时的触发电平。

格式：:TRIGger:429:LEVEL <level>

:TRIGger: 429:LEVEL?

其中，<level>，整型。

2.10 界面命令子系统

2.10.1 :INTERface:SCOPE

功能：打开示波器功能。

格式：:INTERface:SCOPE

2.10.2 :INTERface:TIME

功能：设置系统时间

格式：:INTERface:TIME<year>,<month>,<day>,<hour>,<minute>

:INTERface:TIME?

其中，<year>,<month>,<day>,<hour>,<minute>，整型。

2.10.3 :INTERface:PHOTOS

功能：打开图片查看功能。

格式：:INTERface:PHOTOS<n>

其中，<n>，整型，图片名称

2.10.4 :INTERface:SETTing

2.10.4.1 :INTERface:SETTing:BRIGhtness

功能：设置示波器背景光亮度

格式：:INTERface:SETTing:BRIGhtness<time>

:INTERface:SETTing:BRIGhtness?

其中，<time>，整型，0 至 100.

2.10.4.2 :INTERface:SETTing:VOLume

功能：设置示波器声音大小。

格式：:INTERface:SETTing: VOLume <time>

:INTERface:SETTing: VOLume?

其中，<time>，整型，0 至 100.

2.10.4.3 :INTerface:SETTing:STANdby

功能：设置示波器待机时间。

格式：:INTerface:SETTing:STANdby<time>

:INTerface:SETTing:STANdby?

其中，<time>，整型，0 至 60min。为 0 时 即表示待机失效。

2.10.4.4 :INTerface:SETTing:SHUTdown

功能：设置示波器关机时间。

格式：:INTerface:SETTing:SHUTdown<time>

:INTerface:SETTing:SHUTdown?

其中，<time>，整型，0 至 60min。为 0 时 即表示待机失效。

2.10.4.5 :INTerface:SETTing:USB

功能：设置示波器中 USB 连接方式。

格式：:INTerface:SETTing:USB<mode>

:INTerface:SETTing:USB?

其中，<mode>，离散型，“USBconnect”、“USBvirtualLAN”或“USBstorage”。

2.10.4.6 :INTerface:SETTing:LANGuage

功能：设置示波器显示的语言。

格式：:INTerface:SETTing:LANGuage<lang>

:INTerface:SETTing:LANGuage?

其中，<lang>，离散型。“中文”、“english”……

2.10.5 :INTerface:SYSTem

功能：查询示波器系统信息。包括型号、带宽、序列号、出厂日期、安装模式、硬件版本、等内容信息。

格式：:INTerface:SYSTem?

2.10.6 :INTerface:BATTery

功能：查询示波器电池电量。

格式：:INTerface:BATTery?

2.10.7 :INTerface:SHUTdown

功能：示波器关机。

格式: :INTErface:SHUTdown

2.10.8:INTErface:REStArt

功能: 示波器重启

格式: :INTErface:REStArt

2.10.9:INTErface:STANdbY

功能: 示波器待机

格式: :INTErface: STANdbY

2.10.10:INTErface:LOCK

功能: 示波器锁屏

格式: :INTErface:LOCK

2.10.11:INTErface:UNLock

功能: 示波器解屏

格式: :INTErface:UNLock

2.11 时基命令子系统

2.11.1 :TIMebase:EXTent

功能: 设置水平时基档位。

格式: :TIMebase:EXTent<extent>

:TIMebase:EXTent?

其中, < extent >, 实型。单位: S

2.11.2 :TIMebase:MODE

功能: 设置屏幕时基显示方式。“YT”、“ROLL”或“XY”。

格式: :TIMebase:MODE<mode>

:TIMebase:MODE?

其中, <mode>, 离散型, “YT”、“ROLL”或“XY”。

2.11.3 :TIMebase:XY1:DISPlay

功能: 打开或关闭通道 1 和通道 2 的 XY 模式显示。

格式: :TIMebase:XY1:DISPlay <bool>

:TIMebase:XY1:DISPlay?

其中, <bool>, 布尔型, {{0|OFF}}|{1|ON}}.

2.11.4 :TIMebase:POSition

功能：设置波形显示的水平偏移。

格式：:TIMebase: POSition <position>

:TIMebase: POSition?

其中， <POSition>， 实型。

返回格式：查询以科学计数形式返回偏移值。

举例：

下面的命令设置水平偏移为 2us。

:TIMebase: POSition 0.000002

下面的查询返回 “2.000000e-06”

:TIMebase: POSition?

2.11.5 :TIMebase:VERNier <n>

功能：打开或关闭波形水平微调功能。

格式：:TIMebase:VERNier <bool>

:TIMebase:VERNier?

其中： <bool>， 布尔型， {{0|OFF}}|{1|ON}}。

返回格式：查询返回 “0” 或 “1” 。

举例：

下面的命令打开水平档位微调功能。

:TIMebase: VERNier CH1 ON 或:TIMebase:VERNier CH1 1

下面的查询返回 “1” 。

:TIMebase: VERNier? CH1

2.11.6 :TIMebase:PLUS:EXTent

功能：设置水平时基档位。

格式：:TIMebase: PLUS:EXTent,<enum>

其中， <enum>枚举型， 1 为在原基础上加 1 个档位， -1 为在原基础上减 1 个档位；

2.11.7 :TIMebase: PLUS:OFFset

功能：设置波形显示的水平偏移。

格式: :TIMebase: PLUS:OFFset <enum>

其中, <enum>枚举型, 1 为在原基础上加 1 个单位, -1 为在原基础上减 1 个单位;

2.12 存储命令子系统

2.12.1 :STORage:SAVE

功能: 存储指定通道的波形到指定位置。

格式: :STORage:SAVE<channel>,<save>

:STORage:SAVE<channel>

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4|MATH}; <save>, 离散型, {LOCal|UDISK}, 默认 LOCal

2.12.2 :STORage:LOAD

功能: 载入 ref。

格式: :STORage:LOAD <source>,<bool>

:STORage:LOAD

其中, <source>, 离散型, {REF1| REF 2| REF 3| REF4 }; <bool>, 布尔型, {{0|OFF}}{1|ON}}。

2.12.3:CAPTure

功能: 屏幕截图。

格式: :STORage:CAPTure

2.12.4 :STORage:DEPT h

功能: 设置示波器存储深度。

格式: :STORage:CAPTure<capture>

:STORage:CAPTure?

其中, <capture>, 离散型, 根据机型不同, 支持的也不同, 如果不支持, 则默认为 auto, 比如 12M 可以设置为 12000000

2.12.5 :STORage:CONSave

功能: 存储示波器设置。

格式: :STORage:CONSave

2.12.6 :STORage:CONLoad

功能: 调用示波器设置。

2.12.7 :STORage:RECORD

功能：示波器录制功能的打开与关闭。

格式：:STORage:RECOrd <operate>

:STORage:RECOrd?

其中，<operate>，离散型，{RECOrd|STOP}。

2.12.8 :STORage:PLAY

功能：示波器回放功能的打开和关闭。

格式：:STORage:PLAY <operate>

:STORage:PLAY?

其中，<operate>，离散型，{RECOrd|STOP}。

2.12.8.1 :STORage:PLAY:SPEed

功能：示波器回放快进选项。

格式：:STORage:PLAY:SPEed <speed>

:STORage:PLAY:SPEed?

其中，<speed>，离散型，{2X|4X|8X|16X|32X|64X}。

2.12.8.1 :STORage:PLAY:BACK

功能：示波器回放后退选项。

格式：:STORage:PLAY: BACK <speed>

:STORage:PLAY: BACK?

其中，<speed>，离散型，{2X|4X|8X|16X|32X|64X}。

2.13 总线配置命令子系统

2.13.1 :BUS:TYPE

功能：设置总线 S1 或 S2 的总线类型。

格式：:BUS <channel>:TYPE <type>

:BUS <channel>:TYPE?

其中，<channel>，离散型，{S1|S2};

<type>，离散型，{UART|LIN|SPI|CAN|IIC|1553B|429}。

2.13.2 :BUS:UART

2.13.2.1 :BUS:UART:RX

功能：设置 UART 总线配置的 RX 通道源。

格式：:BUS:UART:RX <channel>

:BUS:UART:RX?

其中，<channel>，离散型，{CH1|CH2|CH3|CH4}。

2.13.2.2 :BUS:UART:IDLElvl

功能：设置 UART 总线配置的空闲电平状态。

格式：:BUS:UART:IDLElvl <state>

:BUS:UART:IDLElvl?

其中，<state>，离散型，{high|low}。

2.13.2.3 :BUS:UART:BAUDrate

功能：选择 UART 总线配置的波特率。单位：b/s

格式：:BUS:UART:BAUDrate <baudrate>

:BUS:UART:BAUDrate?

其中，<baudrate>，离散型，
{1200|2400|4800|9600|19200|38400|43000|56000|57600|115200}。

2.13.2.4 :BUS:UART:CHECK

功能：选择 UART 总线配置的校验方式。

格式：:BUS:UART:CHECK <check>

:BUS:UART:CHECK?

其中，<check>，离散型，{NONE|ODD|EVEN}。

2.13.2.5 :BUS:UART:USERbaud

功能：选择 UART 总线配置时用户自定义的波特率。单位：b/s

格式：:BUS:UART:USERbaud <baudrate>

:BUS:UART:USERbaud?

其中，<baudrate>，整型，1200 至 8000000。

2.12.2.6 :BUS:UART:WIDTH

功能：选择 UART 总线配置时的数据位宽。

格式：:BUS:UART:WIDTH <width>

:BUS:UART:WIDTH?

其中，<width>，离散型，{5|6|7|8|9}。

2.13.3 :BUS:LIN

2.13.3.1 :BUS:LIN:CHANnel

功能：选择 LIN 总线配置的通道源。

格式：:BUS:LIN:CHANnel <channel>

:BUS:LIN:CHANnel?

其中，<channel>，离散型，{CH1|CH2|CH3|CH4}。

2.13.3.2 :BUS:LIN:IDLElvl

功能：设置 LIN 总线配置的空闲电平状态。

格式：:BUS:LIN:IDLElvl <state>

:BUS:LIN:IDLElvl?

其中，<state>，离散型，{high|low}。

2.13.3.3 :BUS:LIN:BAUDrate

功能：选择 LIN 总线配置的波特率。单位，b/s。

格式：:BUS:LIN:BAUDrate <baudrate>

:BUS:LIN:BAUDrate?

其中，<baudrate>，离散型，{2400|9600|19200}。

2.13.3.4 :BUS:LIN:USERbaud

功能：选择 LIN 总线配置时用户自定义的波特率。单位：b/s

格式：:BUS:LIN:USERbaud <baudrate>

:BUS:LIN:USERbaud?

其中，<baudrate>，整型，2400 至 625000。

2.13.4 :BUS:SPI

2.13.4.1 :BUS:SPI:CLK

功能：选择 SPI 总线配置的时钟源。

格式：:BUS:SPI:CLK <channel>

:BUS:SPI:CLK?

其中，<channel>，离散型，{CH1|CH2|CH3|CH4}。

2.13.4.2 :BUS:SPI:DATA

功能：选择 SPI 总线配置的数据源。

格式：:BUS:SPI:DATA <channel>

:BUS:SPI:DATA?

其中，<channel>，离散型，{CH1|CH2|CH3|CH4}。

2.12.4.3 :BUS:SPI:WIDTh

功能：选择 SPI 总线配置时的数据位宽。

格式：:BUS:SPI:WIDTh <width>

:BUS:SPI:WIDTh?

其中，<width>，离散型，{4|8|16|24|32}。

2.12.4.4 :BUS:SPI:IDLElvl

功能：选择 SPI 总线配置的空闲电平状态。

格式：:BUS:SPI:IDLElvl <state>

:BUS:SPI:IDLElvl?

其中，<state>，离散型，{high|low}。

2.12.4.5 :BUS:SPI:SLOPe

功能：选择 SPI 总线配置的时钟边沿类型。

格式：:BUS:SPI:SLOPe <slope>

:BUS:SPI:SLOPe?

其中，<slope>，离散型，{POSItive|NEGAtive}。

2.13.5 :BUS:CAN

2.13.5.1 :BUS:CAN:CHANnel

功能：选择 CAN 总线配置的通道源。

格式：:BUS:CAN:CHANnel <channel>

:BUS:CAN:CHANnel?

其中，<channel>，离散型，{CH1|CH2|CH3|CH4}。

2.13.5.2 :BUS:CAN: :IDLElvl

功能：设置 CAN 总线配置的空闲电平状态。

格式: :BUS:CAN:IDLElvl <state>

:BUS:CAN:IDLElvl?

其中, <state>, 离散型, {high|low}。

2.13.5.3 :BUS:CAN:BAUDrate

功能: 选择 CAN 总线配置的波特率。单位, b/s。

格式: :BUS:CAN:BAUDrate <baudrate>

:BUS:CAN:BAUDrate?

其中, <baudrate>, 离散型,
{10000|20000|33300|50000|62500|83300|100000|125000|250000|500000|800000|1000000}。

2.13.5.4 :BUS:CAN:USERbaud

功能: 选择 CAN 总线配置时用户自定义的波特率。单位: b/s

格式: :BUS:CAN:USERbaud <baudrate>

:BUS:CAN:USERbaud?

其中, <baudrate>, 整型, 10000 至 1000000。

2.13.6 :BUS:IIC

2.13.6.1 :BUS:IIC:SDA

功能: 设置 IIC 总线配置的串行数据通道源。

格式: :BUS:IIC:SDA <channel>

:BUS:IIC:SDA?

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4}。

2.13.6.2 :BUS:IIC:SCL

功能: 设置 IIC 总线配置的串行时钟的通道源。

格式: :BUS:IIC:SCL <channel>

:BUS:IIC:SCL?

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4}。

2.13.7 :BUS:1553B

功能: 设置 1553B 总线配置的通道源。

格式: :BUS:1553B <channel>

:BUS:1553B?

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4}。

2.13.8 :BUS:429

2.13.8.1 :BUS:429:SOURce

功能: 设置 429 总线配置的通道源。

格式: :BUS:429:SOURce <channel>

:BUS:429:SOURce?

其中, <channel>, 离散型, {CH1|CH2|CH3|CH4}。

2.13.8.2 :BUS:429:FORMat

功能: 设置 429 总线配置的格式

格式: :BUS:429:FORMat <format>

:BUS:429:FORMat?

其中, <format>, 离散型, {LSDS|LDSS|LDAT}。

LSDS, LABEL+SDI+DATA+SSM; LDSS, LABEL+DATA+SSM; LDAT, LABEL+DATA。

2.13.8.3 :BUS:429:DISPlay

功能: 设置 429 总线配置的显示模式。

格式: :BUS:429:DISPlay <diaplay>

:BUS:429:DISPlay?

其中, <diaplay>, 离散型, {BINArY|HEX}。

2.13.8.4 :BUS:429:BANDrate

功能: 设置 429 总线配置的波特率。

格式: :BUS:429:BANDrate <bandrate>

:BUS:429:BANDrate?

其中, <bandrate>, 离散型, {12.5|100}。

2.14 PASS/FAIL 命令子系统

2.14.1 :MASK:SOURce

功能: 设置 pass/fial 测试的通道源。

格式: :MASK:SOURce <source>

:MASK:SOURce?

其中, <source>, 离散型, {CH1| CH2| CH3| CH4}。

返回格式: 查询返回 “CH1”、 “CH2”、 “CH3” 或 “CH4”。

举例:

下面的命令设置通道 1 为测试源

:MASK:SOURce CH1

下面的查询返回 “CH1”

:MASK:SOURce?

2.14.2 :MASK:RANGe

功能: 设置模板测试的测试区域。

格式: :MASK:RANGe <range>

:MASK: RANGe?

其中, <until>, 离散型, {SCREen|CURSor}。

返回格式: 查询返回 “SCREen” 或 “CURSor”。

举例:

下面的命令设置 pass/fail 测试的测试区域为 SCREen。

:MASK:RANGe SCREen

下面的查询返回 “SCREen”

:MASK:RANGe?

2.14.3 :MASK:STATistic

功能: 打开或关闭 pass/fail 测试时的统计功能状态, 统计信息包括通过、失败、和总的测试帧数。

格式: :MASK:STATistic <bool>

:MASK:STATistic?

其中, <bool>, 布尔型, {{0|OFF}|{1|ON}}。

返回格式: 查询返回 “0” 或 “1”。

举例:

下面的命令打开统计功能。

:MASK:STATistic ON 或 :MASK:STATistic 1

下面的查询返回“1”

:MASK:STATistic?

2.14.4 :MASK:RESet

功能：复位模板测试统计信息。

格式：:MASK:RESet

2.14.5 :MASK:SOOutput

功能：打开或关闭“输出即停”。

打开：当检测到失败的波形，示波器会停止测试并进入“STOP”状态。此时，测试结果保持在屏幕的显示。

关闭：检测到失败的波形，示波器会继续测试，屏幕上的测试结果不断更新。

格式：:MASK:SOOutput <bool>

:MASK: SOOutput?

其中，<bool>，布尔型，{{0|OFF}}|{1|ON}}。

返回格式：查询返回“0”或“1”。

举例：

下面的命令打开输出即停模式。

:MASK:SOOutput ON 或:MASK:SOOutput 1

下面的查询返回“1”

:MASK:SOOutput?

2.14.6 :MASK:AUXout

功能：设置模板测试的完成响应。打开时输出一个辅助信号脉冲（负脉冲），关闭无输出。

格式：:MASK:AUXout <bool>

:MASK:AUXout?

其中，<bool>，布尔型，{{0|OFF}}|{1|ON}}。

返回格式：查询返回“0”或“1”。

举例：

下面的命令打开完成响应模式。

:MASK: AUXout ON 或:MASK: AUXout 1

下面的查询返回“1”

:MASK: AUXout?

2.14.7 :MASK:ENABle

功能：打开或关闭模板测试。

格式：:MASK:ENABle <bool>

:MASK:ENABle?

其中，<bool>，布尔型，{{0|OFF}}|{1|ON}}。

2.14.8:MASK:OPERate

功能：控制通过/失败测试的运行和停止。

格式：:MASK:OPERate<operate>

:MASK:ENABle?

其中，<operte>，离散型，{RUN|STOP}。

2.14.9:MASK:X

功能：设置通过/失败测试的规则中的“水平调整”参数。

格式：:MASK:X<x>

:MASK:X?

其中，<x>，实型，范围为 0.02 至 4。

2.14.10:MASK:Y

功能：设置通过/失败测试的规则中的“垂直调整”参数。

格式：:MASK:Y<y>

:MASK:Y?

其中，<Y>，实型，范围为 0.03 至 4。

2.15 LAN 命令子系统

2.15.1 :LAN:IPSETting

功能：设置 IP 配置模式。

格式: :LAN: IPSETting <mode >

:LAN: IPSETting?

其中, < mode>, 离散型, {auto|manual}。

返回格式: 查询返回当前的 IP 配置模式。

举例:

下面的命令设置 IP 配置模式为 auto。

:LAN: IPSETting auto

下面的查询返回 “auto” 。

:LAN: IPSETting?

2.15.2:LAN:IPADdress

功能: 设置仪器的 IP 地址。

格式: :LAN: IPADdress<string>

:LAN: IPADdress?

其中, < string>, ASC II 字符串, nnn,nnn,nnn,nnn。

返回格式: 查询以字符串形式返回当前的 IP 地址。

举例:

下面的命令设置 IP 地址为 192.168.1.12。

:LAN: IPADdress 192.168.1.12

下面的查询返回 “192.168.1.12” 。

:LAN: IPADdress?

2.15.3 :LAN:SUBMask

功能: 设置仪器的子网掩码地址。

格式: :LAN: SUBMask <string>

:LAN: SUBMask?

其中, < string>, ASC II 字符串, nnn,nnn,nnn,nnn。

返回格式: 查询以字符串形式返回当前的子网掩码地址。

举例:

下面的命令设置子网掩码地址为 255.255.255.255。

```
:LAN: SUBMask 255.255.255.255
```

下面的查询返回“255.255.255.255”。

```
:LAN: SUBMask?
```

2.15.4 :LAN:ROUTer

功能：设置仪器的 Router 地址。

格式：:LAN: ROUTer <string>

```
:LAN: ROUTer?
```

其中，< string>，ASC II 字符串，nnn,nnn,nnn,nnn。

返回格式：查询以字符串形式返回当前的 Router 地址。

举例：

下面的命令设置 Router 地址为 255.255.255.0。

```
:LAN: ROUTer 255.255.255.0
```

下面的查询返回“255.255.255.0”。

```
:LAN: ROUTer?
```

2.15.5 :LAN:DNS

功能：设置仪器的域名服务器地址。

格式：:LAN: DNS <string>

```
:LAN: DNS?
```

其中，< string>，ASC II 字符串，nnn,nnn,nnn,nnn。

返回格式：查询以字符串形式返回当前的域名服务器地址。

举例：

下面的命令设置域名服务器地址为 192.168.1.1。

```
:LAN: DNS 192.168.1.1
```

下面的查询返回“192.168.1.1”。

```
:LAN: DNS?
```

2.15.6 :LAN:MAC?

功能：查询仪器 MAC 地址。。

返回格式：查询以字符串形式返回 MAC 地址。

2.15.7 :LAN:VISA?

功能：查询仪器 VISA 地址。。

返回格式：查询以字符串形式返回 VISA 地址。

2.16 参考波形命令子系统

2.16.1 :REFeRence:DISPlay

功能：打开或关闭 REF 功能。

格式：:REFeRence:DISPlay<bool>

:REFeRence:DISPlay?

其中，<bool>，布尔型，{{0|OFF}|{1||ON}}。

返回格式：查询返回“1”或“0”。

举例：

下面的命令打开 REF 功能。

:REFeRence:DISPlay ON

下面的命令返回 1.

:REFeRence:DISPlay?

2.16.2 :REFeRence:ENABle<n>

功能：打开或关闭指定的参考通道。

格式：:REFeRence:ENABle <n>,<bool>

:REFeRence:ENABle? <n>

其中，<n>，离散型，{REF1| REF2| REF3| REF4}; <bool>，布尔型，{{0|OFF}|{1||ON}}。

返回格式：查询返回“1”或“0”。

举例：

下面的命令打开 REF1。

:REFeRence:ENABle 1 ,ON

下面的命令返回 1.

:REFeRence:ENABle? 1

2.16.3 :REFeRence:HSCale<n>

功能：设置参考通道的水平档位。

格式：:REFeRence: VSCale<n>,<scale>

:REFeRence: VSCale? <n>

其中，<n>，离散型，{REF1| REF2| REF3| REF4}; <scale>，实型，1ns~1ks。

返回格式：查询以科学计数形式返回水平档位。

2.16.4 :REFeRence:VSCale<n>

功能：设置参考通道的垂直档位。

格式：:REFeRence: VSCale<n>,<scale>

:REFeRence: VSCale? <n>

其中，<n>，离散型，{REF1| REF2| REF3| REF4}; <scale>，实型，5mV~5V。

返回格式：查询以科学计数形式返回垂直档位。

举例：

下面的命令设置参考通道 1 的垂直档位为 2V。

:REFeRence: VSCale ref12

下面的命令返回 2.000000e+00.

:REFeRence ref1: VSCale?

2.16.5 :REFeRence:CURREnt<n>

功能：选择当前参考通道。

格式：:REFeRence: CURREnt<n>

其中，<n>，离散型，{REF1| REF2| REF3| REF4}。

2.4.9 :REFeRence:POSition <n>

功能：设置指定参考通道波形显示的垂直偏移。

格式：: REFeRence: POSition <n>,<pos>

: REFeRence: POSition? <n>

其中，<n>，离散型，{REF1| REF2| REF3| REF4}; <pos>，**实型**。

返回格式：查询以科学计数形式返回偏移值。

举例：

下面的命令设置通道 1 的垂直偏移为 0.01V。

:REfERENCE: POSition R1,0.01

下面的查询返回 “1.000000e-02”

:REfERENCE: POSition? REF1

2.11.4 :REfERENCE:TIMebase:POSition<n>

功能：设置波形显示的水平偏移。

格式： :REfERENCE:TIMebase:POSition <n>,<pos>

:REfERENCE:TIMebase:POSition? <n>

其中，<n>，离散型，{REF1| REF2| REF3| REF4}; <pos>，实型。

返回格式：查询以科学计数形式返回偏移值。

举例：

下面的命令设置 REF1 水平偏移为 2us。

:REfERENCE:TIMebase:POSition REF1, 0.000002

下面的查询返回 “2.000000e-06”

:REfERENCE:TIMebase:POSition? REF1

2.16.6 :REfERENCE:PLUS:HSCale<n>

功能：设置参考通道的水平档位。

格式： :REfERENCE:PLUS: HSCale <n> ,<bool>

其中，<n>，离散型，{REF1| REF2| REF3| REF4};

<bool>，布尔型，1 为加一个档位，0 为减一个档位。

举例：

下面的命令表示 REF1 的水平档位加一个单位

:REfERENCE: PLUS:HSCale REF1 ,1

2.16.7:REfERENCE:PLUS:VSCale<n>

功能：设置参考通道的垂直档位。

格式： :REfERENCE: PLUS:VSCale <n> ,<bool>

其中, <n>, 离散型, {REF1| REF2| REF3| REF4};

<bool>, 布尔型, 1 为加一个单位, 0 为减一个单位。

举例:

下面的命令表示 REF1 的垂直档位加一个单位

:REFeRence: PLUS:VSCale REF1 ,1

2.16.8 :REFeRence:PLUS :TIMebase:POSition<n>

功能: 设置参考通道的水平偏移。

格式: :REFeRence: PLUS :TIMebase:POSition <n> ,<bool>

其中, <n>, 离散型, {REF1| REF2| REF3| REF4};

<bool>, 布尔型, 1 为加一个单位, 0 为减一个单位。

举例:

下面的命令表示 REF1 的水平偏移加一个单位

:REFeRence: PLUS :TIMebase:POSition REF1 ,1

2.16.9:REFeRence:PLUS: POSition <n>

功能: 设置参考通道的垂直偏移。

格式: :REFeRence: PLUS: POSition <n> ,<bool>

其中, <n>, 离散型, {REF1| REF2| REF3| REF4};

<bool>, 布尔型, 1 为加一个单位, 0 为减一个单位。

举例:

下面的命令表示 REF1 的垂直偏移加一个单位

:REFeRence: PLUS: POSition REF1 ,1

2.17WIFI 命令子系统

2.17.1:WIFI:OPEN

功能: 打开或关闭 WIFI 功能。

格式: :WIFI:OPEN<bool>

:WIFI:OPEN?

其中, <bool>, 布尔型, {{0|OFF}|{1|ON}}。

返回格式: 查询返回 “0” 或 “1”。

举例:

下面的命令设置打开 WIFI 功能。

:WIFI:OPEN ON

下面的命令返回 1.

:WIFI:OPEN?

2.17.2 :WIFI:IPSetting

功能: 设置 WIFI 的 IP 配置模式。

格式: :WIFI: IPSetting <mode >

:WIFI: IPSetting?

其中, < mode>, 离散型, {auto|manual}。

返回格式: 查询返回当前的 WIFI IP 配置模式。

举例:

下面的命令设置 WIFI 的 IP 配置模式为 auto。

:WIFI: IPSetting auto

下面的查询返回 “auto”。

:WIFI: IPSetting?

2.17.3:WIFI:IPADdress

功能: 设置仪器的 WIFI IP 地址。

格式: :WIFI: IPADdress<string>

:WIFI: IPADdress?

其中, < string>, ASC II 字符串, nnn,nnn,nnn,nnn。

返回格式: 查询以字符串形式返回当前的 WIFI IP 地址。

举例:

下面的命令设置 WIFI IP 地址为 192.168.1.12。

:WIFI: IPADdress 192.168.1.12

下面的查询返回“192.168.1.12”。

:WIFI: IPADdress?

2.17.4 :WIFI:SUBMask

功能：设置仪器的 WIFI 子网掩码地址。

格式：:WIFI: SUBMask <string>

:WIFI: SUBMask?

其中，< string>，ASC II 字符串，nnn,nnn,nnn,nnn。

返回格式：查询以字符串形式返回当前的 WIFI 子网掩码地址。

举例：

下面的命令设置 WIFI 子网掩码地址为 255.255.255.255。

:LAN: SUBMask 255.255.255.255

下面的查询返回“255.255.255.255”。

:LAN: SUBMask?

2.17.5 :WIFI:ROUTer

功能：设置仪器的 WIFI Router 地址。

格式：:WIFI: ROUTer <string>

:WIFI: ROUTer?

其中，< string>，ASC II 字符串，nnn,nnn,nnn,nnn。

返回格式：查询以字符串形式返回当前的 Router 地址。

举例：

下面的命令设置 WIFI Router 地址为 255.255.255.0。

:WIFI: ROUTer 255.255.255.0

下面的查询返回“255.255.255.0”。

:WIFI: ROUTer?

2.17.6:WIFI:DNS

功能：设置仪器的 WIFI 域名服务器地址。

格式：:WIFI: DNS <string>

:WIFI: DNS?

其中，<string>，ASC II 字符串，nnn,nnn,nnn,nnn。

返回格式：查询以字符串形式返回当前的 WIFI 域名服务器地址。

举例：

下面的命令设置 WIFI 域名服务器地址为 192.168.1.1。

:WIFI: DNS 192.168.1.1

下面的查询返回“192.168.1.1”。

:WIFI: DNS?

2.18 搜索命令子系统(暂无)

2.18.1 :SEARCh:TYPE

功能：选择搜索类型。

格式：:SEARCh:TYPE <type>

:SEARCh:TYPE?

其中，<type>，离散型，
{EDGE|PULSe|LOGIc|DWARt|B|SLOPe|TIMEout|NEDGE|SETUp|VIDEo|UART|LIN|SPI|CAN
|I2C|1553B|429}。

返回格式：查询返回当前使用的搜索类型。

举例：

下面的命令选择边沿搜索。

:SEARCh:TYPE EDGE

下面的查询返回“EDGE”。

:SEARCh:TYPE?

2.18.2 :SEARCh:HOLDoff

功能：设置搜索释抑时间。

格式：:SEARCh:HOLDoff <value>

:SEARCh:HOLDoff?

其中，<value>，实型，200ns 至 10s。

返回格式：查询以科学计数形式返回搜索释抑时间。

举例：

下面的命令设置搜索释抑时间为 200ns

:SEARCh:HOLDoff 0.0000002

下面的查询返回 “2.000000e-07” 。

:SEARCh:HOLDoff?

2.18.3 SEARCh:MODE

功能： 设置搜索方式：自动或普通。

格式： :SEARCh:MODE <mode>

:SEARCh:MODE?

其中，<mode>，离散型，{AUTO|NORMAl}。

返回格式： 查询返回 “AUTO” 或 “NORMAl” 。

举例：

下面的命令选择自动搜索模式。

:SEARCh:MODE AUTO

下面的查询返回 “AUTO” 。

:SEARCh:MODE?

2.18.4 :SEARCh:STATus

功能： 查询当前的搜索状态。

格式： :SEARCh:STATus?

返回格式： 查询返回 “RUN” 、 “WAIT” 、 “AUTO” 、 “STOP” 。

2.18.5 :SEARCh:EDGE

2.18.5.1 :SEARCh:EDGE:SOURce

功能： 选择边沿搜索的搜索源。

格式： :SEARCh:EDGE:SOURce <source>

:SEARCh:EDGE:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式： 查询返回 “CH1” 、 “CH2” 、 “CH3” 或 “CH4” 。

举例：

下面的命令设置通道 1 为搜索源。

```
:SEARCh:EDGE:SOURce CH1
```

下面的查询返回“CH1”。

```
:SEARCh:EDGE:SOURce?
```

2.18.5.2 :SEARCh:EDGE:SLOPe

功能：选择边沿搜索的边沿类型。

格式：:SEARCh:EDGE: SLOPe <edge>

```
:SEARCh:EDGE: SLOPe?
```

其中，<edge>，离散型，{RISE|FALL|DUAL}。

返回格式：查询返回“RISE”、“FALL”或“DUAL”。

举例：

下面的命令选择上升沿搜索。

```
:SEARCh:EDGE: SLOPe RISE
```

下面的查返回“RISE”。

```
:SEARCh:EDGE: SLOPe?
```

2.18.5.3 :SEARCh:EDGE:LEVEL

功能：设置边沿搜索时的阈值电平

格式：:SEARCh:EDGE:LEVEL <level>

```
:SEARCh:EDGE:LEVEL?
```

其中，<level>，**实型**。

返回格式：查询以科学计数形式返回阈值电平值。

举例：

下面的命令设置阈值电平为 150mV。

```
:SEARCh:EDGE:LEVEL 0.15
```

下面的查询返回“1.500000e-01”。

```
:SEARCh:EDGE:LEVEL?
```

2.18.5.4 :SEARch:EDGE:COUPle

功能： 设置边沿搜索耦合方式。

格式： :SEARch:EDGE:COUPle <couple>

:SEARch:EDGE:COUPle?

其中，<couple>，离散型，{DC|AC|HFRej|LFRej|Noiserej}。

返回格式： 查询返回“DC”、“AC”、“HFRej”、“LFRej”或“Noiserej”。

举例：

下面的命令选择 DC 耦合方式。

:SEARch:EDGE:COUPle DC

下面的查询返回“DC”。

:SEARch:EDGE:COUPle?

2.18.6 :SEARch:PULSe

2.18.6.1 :SEARch:PULSe:SOURce

功能： 设置脉宽搜索的搜索源。

格式： :SEARch:PULSe:SOURce <source>

:SEARch:PULSe:SOURce

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

返回格式： 查询返回“CH1”、“CH2”、“CH3”或“CH4”。

举例：

下面的命令设置通道 1 为搜索源。

:SEARch: PULSe:SOURce CH1

下面的查询返回“CH1”。

:SEARch: PULSe:SOURce?

2.18.6.2 :SEARch:PULSe:POLArity

功能： 设置脉宽搜索的极性。

格式： :SEARch:PULSe:POLArity <polarity>

:SEARch:PULSe:POLArity?

其中, <polarity>, 离散型, {POSitive|NEGAtive}。

返回格式: 查询返回“POSitive”或“NEGAtive”。

举例:

下面的命令选择上升沿搜索。

:SEARCh:PULSe:POLARity POSitive

下面的查返回“POSitive”。

:SEARCh:PULSe:POLARity?

2.18.6.3 :SEARCh:PULSe:WIDTh

功能: 设置脉宽搜索时的脉冲宽度值。

格式: :SEARCh:PULSe:WIDTh<width>

:SEARCh:PULSe:WIDTh?

其中, <width>, 实型, 40ns 至 10s。

返回格式: 查询返回实数。

举例:

下面的命令设置脉宽值为 4ns。

:SEARCh:PULSe:WIDTh 4.000000e-08

下面的查询返回“4.000000e-08”。

:SEARCh:PULSe:WIDTh?

2.18.6.4 :SEARCh:PULSe:CONDition

功能: 设置脉宽搜索条件。

格式: :SEARCh:PULSe:CONDition<condition>

:SEARCh:PULSe:CONDition?

其中, <condition>, 离散型, {GREAt|LESS|EQUAL|UNEQual}。

GREAt:示波器输入信号脉宽大于指定的脉冲宽度;

LESS : 示波器输入信号脉宽小于指定的脉冲宽度;

EQUAL: 示波器输入信号脉宽等于指定的脉冲宽度;

UNEQual: 示波器输入信号脉宽不等于指定的脉冲宽度;

2.18.6.5 :SEARch:PULSe:LEVEl

功能： 设置脉宽搜索时的阈值电平

格式： :SEARch: PULSe:LEVEl <level>

:SEARch: PULSe:LEVEl?

其中，<level>，实型。

返回格式： 查询以科学计数形式返回阈值电平值。

举例：

下面的命令设置阈值电平为 150mV。

:SEARch: PULSe:LEVEl 0.15

下面的查询返回 “1.500000e-01”。

:SEARch: PULSe:LEVEl?

2.18.7 :SEARch:LOGIc

2.18.7.1 :SEARch:LOGIc:STATus

功能： 设置逻辑搜索中各通道的逻辑状态

格式： :SEARch:LOGIc <channel>:STATus <status>

:SEARch:LOGIc <channel>:STATus?

其中，<channel>，离散型，{CH1|CH2|CH3|CH4}。<status>，离散型，{high|ow|none}。

2.18.7.2 :SEARch:LOGIc:FUNCTion

功能： 设置逻辑搜索的比较函数。

格式： :SEARch:LOGIc:FUNCTion <function>

:SEARch:LOGIc: FUNCTion?

其中，<function>，离散型，“AND”、“OR”、“NAND”或“NOR”。

2.18.7.3 :SEARch:LOGIc:CONDition

功能： 设置逻辑搜索条件。

格式： :SEARch: LOGIc:CONDition<condition>

:SEARch: LOGIc:CONDition?

其中，<condition>，离散型，{GREAT|LESS|EQUAL|UNEQUAL|TRUE|FALSE}。

GREAT:逻辑状态为真的保持时间大于搜索逻辑时间时搜索；

LESS : 逻辑状态为真的保持时间小于搜索逻辑时间时搜索;

EQUAL: 逻辑状态为真的保持时间等于搜索逻辑时间时搜索;

UNEQUAL: 逻辑状态为真的保持时间不等于搜索逻辑时间时搜索;

TRUL: 逻辑状态为真时搜索;

FALSe: 逻辑状态为假时搜索。

2.18.7.4 :SEARCh: LOGIc:TIME

功能: 设置搜索逻辑时间。

格式: :SEARCh: LOGIc:TIME<time>

:SEARCh: LOGIc:TIME?

其中, <time>, 实型, 200ns 至 10s。

2.18.7.5 :SEARCh: LOGIc:LEVEL

功能: 设置逻辑搜索时的各通道阈值电平

格式: :SEARCh: LOGIc <channel>:LEVEL <level>

:SEARCh: LOGIc <channel>:LEVEL?

其中, <channel>,离散型, {CH1|CH2|CH3|CH4}; <level>, 实型。

2.18.8 :SEARCh:B

2.18.8.1 :SEARCh:B:SOURce

功能: 设置 B 搜索的搜索源。

格式: :SEARCh:B:SOURce <source>

:SEARCh:B:SOURce

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

2.18.8.2 :SEARCh:B:EDGE

功能: 设置 B 搜索的搜索斜率。

格式: :SEARCh:B:EDGE <edge>

:SEARCh:B:EDGE?

其中, <edge>, 离散型, {RISE|FALL}。

2.18.8.3 :SEARCh:B:COUPle

功能: 设置 B 搜索耦合方式。

格式: :SEARch:B:COUPle <couple>

:SEARch:B:COUPle?

其中, <couple>, 离散型, {DC|AC|HFRej|LFRej|Noiserej}。

2.18.8.4 :SEARch:B:SEQUence

功能: 设置 B 搜索的搜索类型 (B 在 A 后搜索时间/事件)。

格式: :SEARch:B:SEQUence <sequence>

:SEARch:B:SEQUence?

其中, <sequence>, 离散型, {time|event}。

2.18.8.5 :SEARch:B: LEVEL

功能: 设置 B 搜索时的阈值电平

格式: :SEARch:B:LEVEL <level>

:SEARch:B:LEVEL?

其中, <level>, 实型。

2.18.9 :SEARch:DWART

2.18.9.1 :SEARch:DWART:SOURce

功能: 设置矮脉宽搜索的搜索源。

格式: :SEARch:DWART:SOURce <source>

:SEARch:DWART:SOURce?

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

2.18.9.2 :SEARch:DWART:POLArity

功能: 设置矮脉宽搜索的脉冲极性。

格式: :SEARch:DWART:POLArity <polarity>

:SEARch:DWART:POLArity?

其中, <polarity>, 离散型, {POSitive|NEGAtive}。

2.18.9.3 :SEARch:DWART: CONDition

功能: 设置脉宽限制条件。

格式: :SEARch: DWART:CONDition<condition>

:SEARch: DWART:CONDition?

其中，<condition>，离散型，{GREAT|LESS|EQUAL|UNEQUAL}。

GREAT:示波器输入信号脉宽大于指定的脉冲宽度；

LESS：示波器输入信号脉宽小于指定的脉冲宽度；

EQUAL: 示波器输入信号脉宽等于指定的脉冲宽度；

UNEQUAL: 示波器输入信号脉宽不等于指定的脉冲宽度；

2.18.9.4 :SEARCh:DWART:WIDTh

功能：设置矮脉宽搜索时的脉冲宽度值。

格式：:SEARCh: DWART:WIDTh<width>

:SEARCh: DWART:WIDTh?

其中，<width>，实型，40ns 至 10s。

2.18.9.5 :SEARCh:DWART:HLEVel

功能：设置矮脉宽触发时的高电平。

格式：:SEARCh: DWART:HLEVel <level>

:SEARCh: DWART:HLEVel?

其中，<level>，实型。

2.18.9.6 :SEARCh:DWART:LLEVel

功能：设置矮脉宽搜索时的低电平。

格式：:SEARCh: DWART:LLEVel <level>

:SEARCh: DWART:LLEVel?

其中，<level>，实型。

2.18.10 :SEARCh:SLOPe

2.18.10.1 :SEARCh:SLOPe:SOURce

功能：设置斜率搜索的搜索源。

格式：:SEARCh: SLOPe:SOURce <source>

:SEARCh: SLOPe:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.18.10.2 :SEARCh:SLOPe:POLArity

功能：设置斜率搜索的极性（正/负）。

格式: :SEARch:SLOPe:POLArity <polarity>

:SEARch:SLOPe:POLArity?

其中, <polarity>, 离散型, {POSItive|NEGAtive}。

2.18.10.3 :SEARch: SLOPe: CONDition

功能: 设置斜率搜索的限制条件。

格式: :SEARch: SLOPe:CONDition<condition>

:SEARch: SLOPe:CONDition?

其中, <condition>, 离散型, {GREAt|LESS|GRLEss}。

GREAt:示波器输入信号斜率大于指定的时间设置;

LESS : 示波器输入信号斜率小于指定的时间设置;

GRLEss: 示波器输入信号斜率大于指定的时间上限且小于指定的时间下限。

2.18.10.4 :SEARch:SLOPe:HTIMe

功能: 设置斜率搜索时的时间上限。

格式: :SEARch:SLOPe:HTIMe <time>

:SEARch:SLOPe:HTIMe?

其中, <time>, 实型, 10ns 至 10s。

2.18.10.5 :SEARch:SLOPe:LTIMe

功能: 设置斜率搜索时的时间下限。

格式: :SEARch:SLOPe:LTIMe <time>

:SEARch:SLOPe:LTIMe?

其中, <time>, 实型, 10ns 至 10s。

2.18.10.6 :SEARch:SLOPe:UPLVI

功能: 设置斜率搜索时的阈值电平上限。

格式: :SEARch:SLOPe:UPLVI <level>

:SEARch:SLOPe:UPLVI?

其中, <level>, 实型。

2.18.10.6 :SEARch:SLOPe:LOWLvl

功能: 设置斜率搜索时的阈值电平上限。

格式: :SEARch:SLOPe: LOWLvl <level>

:SEARch:SLOPe: LOWLvl?

其中, <level>, 实型。

2.18.11 :SEARch:TIMEout

2.18.11.1 :SEARch:TIMEout:SOURce

功能: 设置超时搜索的搜索源。

格式: :SEARch:TIMEout:SOURce <source>

:SEARch:TIMEout:SOURce?

其中, <source>, 离散型, {CH1|CH2|CH3|CHANne

2.18.11.2 :SEARch:TIMEout:EDGE

功能: 设置超时搜索的边沿类型。

格式: :SEARch:TIMEout:EDGE <edge>

:SEARch:TIMEout:EDGE?

其中, <edge>, 离散型, {RISE|FALL|DUAL}。

2.18.11.3 :SEARch:TIMEout:TIME

功能: 设置超时搜索的超时时间。

格式: :SEARch:TIMEout:TIME <time>

:SEARch:TIMEout:TIME?

其中, <time>, 实型, 20ns 至 10s。

2.18.12 :SEARch:NEDGE

2.18.12.1 :SEARch:NEDGE:SOURce

功能: 设置第 N 边沿搜索的搜索源。

格式: :SEARch:NEDGE:SOURce <source>

:SEARch:NEDGE:SOURce?

其中, <source>, 离散型, {CH1|CH2|CH3|CH4}。

2.18.12.2 :SEARch:NEDGE:SLOPe

功能: 设置第 N 边沿搜索的边沿类型。

格式: :SEARch:NEDGE:SLOPe <slope>

:SEARch:NEDGE:SLOPe?

其中，<slope>，离散型，{RISE|FALL}。

2.18.12.3 :SEARch:NEDGE:IDLE

功能：设置第 N 边沿搜索中开始边沿计数之前的空闲时间。

格式：:SEARch:NEDGE:IDLE <time>

:SEARch:NEDGE:IDLE?

其中，<time>，实型，20ns 至 10s。

2.18.12.4 :SEARch:NEDGE:EDGE

功能：设置第 N 边沿搜索的 N 的数值。

格式：:SEARch:NEDGE:EDGE <number>

:SEARch:NEDGE:EDGE?

其中，<number>，实型，1 至 10000。

2.18.12.5:SEARch: NEDGE:LEVEL

功能：设置第 N 边沿搜索时的阈值电平

格式：:SEARch: NEDGE:LEVEL <level>

:SEARch: NEDGE:LEVEL?

其中，<level>，实型。

2.18.13 :SEARch:SETUp

2.18.13.1 :SEARch:SETUp:CLOCK

功能：设置建立保持时间搜索的时钟信号源。

格式：:SEARch:SETUp:CLOCK <source>

:SEARch:SETUp:CLOCK?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.18.12 :SEARch:SETUp:DATA

功能：设置建立保持时间搜索的数据信号源。

格式：:SEARch:SETUp:DATA<source>

:SEARch:SETUp:DATA?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.18.13.3 :SEARch:SETUp:CEdGe

功能：设置建立保持时间搜索的时钟边沿类型。

格式：:SEARch:SETUp:CEdGe <edge>

:SEARch:SETUp:CEdGe?

其中，<edge>，离散型，{RISE|FALL}。

2.18.13.4 :SEARch:SETUp:STIME

功能：设置建立保持时间搜索的建立时间。

格式：:SEARch:SETUp:STIME <time>

:SEARch:SETUp:STIME?

其中，<time>，实型，40ns~10s。

2.18.13.4 :SEARch:SETUp:HTIME

功能：设置建立保持时间搜索的保持时间。

格式：:SEARch:SETUp:HTIME <time>

:SEARch:SETUp:HTIME?

其中，<time>，实型，40ns~10s。

2.18.13.5 :SEARch:SETUp:CLEVel

功能：设置建立保持时间搜索的时钟源阈值电平。

格式：:SEARch:SETUp:CLEVel <level>

:SEARch:SETUp:CLEVel?

其中，<level>，实型。

2.18.13.6 :SEARch:SETUp:DLEVel

功能：设置建立保持时间搜索的数据源阈值电平。

格式：:SEARch:SETUp:DLEVel <level>

:SEARch:SETUp:DLEVel?

其中，<level>，实型。

2.18.14 :SEARch:VIDEo

2.18.14.1 :SEARch:VIDEo:SOURce

功能：设置视频搜索的搜索源。

格式：:SEARch:VIDEo:SOURce <source>

:SEARch:VIDEo:SOURce?

其中，<source>，离散型，{CH1|CH2|CH3|CH4}。

2.18.14.2 :SEARch:VIDEo:POLArity

功能：设置视频搜索的极性

格式：:SEARch:VIDEo:POLArity <polarity>

:SEARch:VIDEo:POLArity?

其中，<polarity>，离散型，{POSitive|NEGAtive}。

2.18.14.3 :SEARch:VIDEo:STANdard

功能：选择视频搜索时的视频标准。

格式：:SEARch:VIDEo:STANdard <standard>

:SEARch:VIDEo:STANdard?

其中，<standard>，离散型，{PAL|SECAm|NESC|720P|1080I|1080P}。

2.18.14.4 :SEARch:VIDEo:AMODE

功能：选择搜索标准为 PAL、SECAm、NESC、1080I 时视频搜索的同步类型。

格式：:SEARch:VIDEo:AMODE <mode>

:SEARch:VIDEo:AMODE?

其中，<mode>，离散型，{ODDField|EVENfield|AFIEld|ALINE|LINE}。

2.18.14.5 :SEARch:VIDEo:BMODE

功能：选择搜索标准为 720P、1080P 时视频搜索的同步类型。

格式：:SEARch:VIDEo:BMODE <mode>

:SEARch:VIDEo:BMODE?

其中，<mode>，离散型，{AFIEld|ALINE|LINE}。

2.18.14.6 :SEARch:VIDEo:AFREquence

功能：选择搜索标准为 720P、1080I 时视频搜索的信号频率。

格式：:SEARch:VIDEo:AFREquence <frequency>

:SEARch:VIDEo:AFREquence?

其中，<frequency>，离散型，{60Hz|50Hz}。

2.18.14.7 :SEARch:VIDEo:BFREquence

功能：选择搜索标准为 1080P 时视频搜索的信号频率。

格式：:SEARch:VIDEO:BFREquence <frequency>

:SEARch:VIDEO:BFREquence?

其中，<frequency>，离散型，{60Hz|50Hz|30Hz|25Hz|24Hz}。

2.18.15 :SEARch:UART

2.18.15.1 :SEARch:UART:SOURce

功能：设置 UART 搜索的搜索源。

格式：:SEARch:UART:SOURce <source>

:SEARch:UART:SOURce?

其中，<source>，离散型，{S1|S2}。

2.18.15.2 :SEARch:UART:TYPE

功能：设置 UART 搜索的搜索条件。

格式：:SEARch:UART:TYPE <type>

:SEARch:UART:TYPE?

其中，<type>，离散型，{START|STOP|DATA|0:DATA|1:DATA|X:DATA|PARity}。

当总线设置中总线字长设置为 9bit 时，搜索类型 DATA 不能设置；

当总线设置中总线字长设置为 5bit、6bit、7bit、8bit 时，搜索类型中 0:DATA、1:DATA、X:DATA 不能进行设置。

2.18.15.3 :SEARch:UART:RELAtion

功能：当 UART 总线搜索条件选择为 DATA、0:DATA、1:DATA、X:DATA 时，设置 UART 总线搜索关系。

格式：:SEARch:UART:RELAtion <RELATION>

:SEARch:UART:RELAtion?

其中，<RELATION>，离散型，{GREAt|LESS|EQUAL|UNEQual}。

GREAt:示波器输入数据大于指定的搜索数据；

LESS：示波器输入数据小于指定的搜索数据；

EQUAL：示波器输入数据等于指定的搜索数据；

UNEQual：示波器输入数据不等于指定的搜索数据；

2.18.15.4 :SEARCh: UART :DATA

功能：当 UART 总线搜索条件选择为 DATA、0:DATA、1:DATA、X:DATA 时，设置 UART 总线搜索数据。

格式：:SEARCh: UART :DATA <data>

:SEARCh: UART :DATA?

其中，<data>，整型，16 进制，0 至 FF。

2.18.15.5 :SEARCh: UART:LEVEL

功能：设置 UART 搜索时的阈值电平。

格式：:SEARCh: UART:LEVEL <level>

:SEARCh: UART:LEVEL?

其中，<level>，整型。

2.18.16 :SEARCh:LIN

2.18.16.1 :SEARCh:LIN:SOURCce

功能：设置 LIN 搜索的搜索源。

格式：:SEARCh:LIN:SOURCce <source>

:SEARCh:LIN:SOURCE?

其中，<source>，离散型，{S1|S2}。

2.18.16.2 :SEARCh:LIN:TYPE

功能：设置 LIN 搜索的搜索条件。

格式：:SEARCh:LIN:TYPE <type>

:SEARCh:LIN:TYPE?

其中，<type>，离散型，{SRISe|FID|IDATa}。

SRISe，同步上升沿；FID，帧 ID；IDATa，帧 ID 和数据。

2.18.16.3 :SEARCh:LIN:ID

功能：当 LIN 总线搜索条件为 FID 或 IDATa 时，设置 LIN 搜索的搜索 ID 值。

格式：:SEARCh:LIN:ID <data>

:SEARCh:LIN:ID?

其中，<data>，整型，16 进制，0 至 3F。

2.18.16.4 :SEARCh:LIN:DATA

功能：当 LIN 总线搜索条件为 IDATa 时，设置 LIN 搜索的搜索数据。

格式：:SEARch:LIN:DATA <data>

:SEARch:LIN:DATA?

其中，<data>，整型，16 进制，0 至 FFFF,FFFF,FFFF,FFFF。

2.18.16.5 :SEARch:LIN:LEVEL

功能：设置 LIN 搜索时的阈值电平。

格式：:SEARch:LIN:LEVEL <level>

:SEARch:LIN:LEVEL?

其中，<level>，整型。

2.18.17 :SEARch:CAN

2.18.17.1 :SEARch:CAN:SOURce

功能：设置 CAN 搜索的搜索源。

格式：:SEARch:CAN:SOURce <source>

:SEARch:CAN:SOURce?

其中，<source>，离散型，{S1|S2}。

2.18.17.2 :SEARch:CAN:TYPE

功能：设置 CAN 搜索的搜索条件

格式：:SEARch:CAN:TYPE <type>

:SEARch:CAN:TYPE?

其中，<type>，离散型，
{FSTArt|RFID|DFID|RDID|IDATa|WRFR|AERRor|ACKError|OVERload}。

FSTArt，帧起始；RFID，远程帧 ID；DFID 数据帧 ID；RDID，远程帧/数据帧 ID；IDATa，数据帧 ID 和数据；WRFR，错误帧；AERRor，所有错误；ACKError，确认错误；OVERload，过载帧。

2.18.17.3 :SEARch:CAN:ID

功能：当 CAN 搜索的搜索条件为 RFID、DFID、IDATa 或 RDID 时，设置 CAN 搜索的搜索 ID 值。

格式：:SEARch:CAN:ID <data>

:SEARch:CAN:ID?

其中，<data>，整型，16 进制，0 至 FFFF,FFFF。

2.18.17.4 :SEARCh:CAN:DLC

功能：当 CAN 搜索的搜索条件为 IDATa 时，设置 CAN 搜索的 DLC 值。

格式：:SEARCh:CAN:DLC <data>

:SEARCh:CAN:DLC?

其中，<data>，整型，0 至 8.

2.18.17.5 :SEARCh:CAN:DATA

功能：当 CAN 搜索的搜索条件为 IDATa 时，设置 CAN 搜索的搜索数据值。

格式：:SEARCh:CAN:DATA <data>

:SEARCh:CAN:DATA?

其中，<data>，整型，16 进制，数据位数由 DLC 确定。

2.18.17.6 :SEARCh:CAN:LEVEL

功能：设置 CAN 搜索时的阈值电平。

格式：:SEARCh:CAN:LEVEL <level>

:SEARCh:CAN:LEVEL?

其中，<level>，整型。

2.18.18 :SEARCh:SPI

2.18.18.1 :SEARCh:SPI:DATA

功能：设置 SPI 触发下的数据值。

格式：:SEARCh:SPI:DATA <data>

:SEARCh:SPI:DATA?

其中，<data>，整型，二进制。

2.18.18.2 :SEARCh: SPI:SOURce

功能：设置 SPI 搜索的搜索源。

格式：:SEARCh: SPI:SOURce <source>

:SEARCh: SPI:SOURce?

其中，<source>，，离散型，{S1|S2}。

2.18.18.3 :SEARCh: SPI:LEVEL

功能：设置 SPI 搜索时的阈值电平。

格式: :SEARCh: SPI:LEVEL <level>

:SEARCh: SPI:LEVEL?

其中, <level>, 整型。

2.18.19 :SEARCh:IIC

2.18.19.1 :SEARCh:IIC:SOURce

功能: 设置 IIC 搜索的搜索源。

格式: :SEARCh:IIC:SOURce <source>

:SEARCh:IIC:SOURce?

其中, <source>, , 离散型, {S1|S2}。

2.18.19.2 :SEARCh:IIC:TYPE

功能: 设置 IIC 搜索的搜索类型。

格式: :SEARCh:IIC:TYPE <type>

:SEARCh:IIC:TYPE?

其中, <type>, 离散型,
{START|STOP|ACKLost|NACKaddress|REStart|RDATa|FRAM1|FRAM2}。

START, 起始条件; STOP, 停止条件; ACKLost, 确认丢失; NACKaddress, 地址字段无确认; REStart, 重新启动; RDATa, EEPROM 数据读取; FRAM1, 帧型 1; FRAM2, 帧型 2。

2.18.19.3 :SEARCh:IIC:ADDRes

功能: 当 IIC 搜索条件为 NACKaddress、FRAM1 或 FRAM2 时, 设置 IIC 总线搜索的搜索地址。

格式: :SEARCh:IIC:ADDRes <data>

:SEARCh:IIC:ADDRes?

其中, <data>, 整型, 16 进制, 0 至 FF。

2.18.19.4 :SEARCh:IIC:RELAtion

功能: 当 IIC 搜索条件为 RDATa 时, 设置 IIC 总线搜索的搜索关系。

格式: :SEARCh:IIC:RELAtion <relation>

:SEARCh:IIC:RELAtion

其中, <RELATION>, 离散型, {GREAt|LESS|EQUAL|UNEQual}。

GREAT:示波器输入数据大于指定的搜索数据;

LESS : 示波器输入数据小于指定的搜索数据;

EQUAL: 示波器输入数据等于指定的搜索数据;

UNEQual: 示波器输入数据不等于指定的搜索数据;

2.18.19.5 :SEARch:IIC:DATA

功能: 当 IIC 搜索条件为 RDATA、FRAM1 或 FRAM2 时, 设置 IIC 总线搜索的搜索数据。

格式: :SEARch:IIC:DATA <data>

:SEARch:IIC:DATA?

其中, <data>, 整型, 十六进制。

2.18.19.6 :SEARch: IIC:LEVEL

功能: 设置 IIC 搜索时的阈值电平。

格式: :SEARch: IIC:LEVEL <level>

:SEARch: IIC:LEVEL?

其中, <level>, 整型。

2.18.20 :SEARch:1553B

2.18.20.1 :SEARch:1553B:SOURce

功能: 设置 1553B 总线搜索的搜索源。

格式: :SEARch:1553B:SOURce <source>

:SEARch:1553B:SOURce?

其中, <source>, , 离散型, {S1|S2}。

2.18.20.2 :SEARch:1553B:TYPE

功能: 设置 1553B 总线搜索的搜索条件。

格式: :SEARch:1553B:TYPE <type>

:SEARch:1553B:TYPE?

其中, <type>, 离散型,
{CSSYnc|DWSYnc|CSWOrd|DWOOrd|RTADdress|OPERror|MERRor|AERRor}。

CSSync, 指令/状态字同步头; DWSync, 数据字同步头; CSWord, 指令/状态字; DWord, 数据字; RTAddress, 远程终端地址; OPErr, 奇校验错误; MERR, 曼彻斯特码错误; AERR, 所有错误。

2.18.20.3 :SEARch:1553B:CSWord

功能: 当 1553B 搜索条件为 CSWord 时, 设置 1553B 总线搜索的指令/状态字数值。

格式: :SEARch:1553B:CSWord <data>

:SEARch:1553B:CSWord?

其中, <data>, 整型, 0 至 FFFF。

2.18.20.4 :SEARch:1553B: DWord

功能: 当 1553B 搜索条件为 DWord 时, 设置 1553B 总线搜索的搜索数据值。

格式: :SEARch:1553B: DWord <data>

:SEARch:1553B: DWord?

其中, <data>, 整型, 0 至 FFFF。

2.18.20.5 :SEARch:1553B: RTAddress

功能: 当 1553B 总线搜索条件为 RTAddress 时, 设置 1553B 总线搜索的远程终端地址。

格式: :SEARch:1553B: RTAddress<address>

:SEARch:1553B: RTAddress?

其中, <address>, 整型, 0 至 FF。

2.18.20.6 :SEARch: 1553B:LEVEL

功能: 设置 1553B 搜索时的阈值电平。

格式: :SEARch: 1553B:LEVEL <level>

:SEARch: 1553B:LEVEL?

其中, <level>, 整型。

2.18.21 :SEARch:429

2.18.21.1 :SEARch:429:SOURce

功能: 设置 429 总线搜索的搜索源。

格式: :SEARch:429:SOURce <source>

:SEARch:429:SOURce?

其中，<source>，，离散型，{S1|S2}。

2.18.21.2 :SEARCh:429:TYPE

功能：设置 429 总线搜索的搜索条件。

格式：:SEARCh:429:TYPE <type>

:SEARCh:429:TYPE?

其中，<type>，离散型，
{WBEGin|WEND|WORD|LABEL|SDI|DATA|SSM|LSDI|LDATa|LSSM|WERRor|WINTerval|VERRor|AERRor|ALLO|ALL1}。

WBEGin，字起始；WEND，字结束；WORD，全字匹配；LSDI，LABEL+SDI；LDATa，LABEL+DATA；LSSM，LABEL+SSM；WERRor，字错误；WINTerval，字间隙错误；VERRor，校验错误；AERRor，所有错误；ALLO，所有 0 位；ALL1，所有 1 位。

2.18.21.3 :SEARCh:429:WORD

功能：当 429 总线搜索条件为 WORD 时，设置 429 总线搜索的搜索字值。

格式：:SEARCh:429:WORD <data>

:SEARCh:429:WORD?

其中，<data>，整型，0 至 FFFFFFFF。

2.18.21.4 :SEARCh:429:LABEL

功能：当 429 总线搜索条件为 LABEL、LSDI、LDATa 或 LSSM 时，设置 429 总线搜索的搜索 LABEL 值。

格式：:SEARCh:429:LABEL <data>

:SEARCh:429:LABEL?

其中，<data>，整型，八进制，0 至 3FF。

2.18.21.5 :SEARCh:429:SDI

功能：当 429 总线搜索条件为 SDI 或 LSDI 时，设置 429 总线搜索的搜索 SDI 值。

格式：:SEARCh:429:SDI <data>

:SEARCh:429:SDI?

其中，<data>，整型，0 至 3。

2.18.21.6 :SEARCh:429:DATA

功能：当 429 总线搜索条件为 DATA 或 LDATa 时，设置 429 总线搜索的搜索数据值。

格式：:SEARCh:429:DATA <data>

:SEARch:429:DATA?

其中，<data>，整型，0 至 FFFFFFFF。

2.18.21.7 :SEARch:429:SSM

功能：当 429 总线搜索条件为 SSM 或 LSSM 时，设置 429 总线搜索的搜索数据值。

格式：:SEARch:429:SSM <data>

:SEARch:429:SSM?

其中，<data>，整型，0 至 3。

2.18.21.8 :SEARch: 429:LEVEL

功能：设置 429 搜索时的阈值电平。

格式：:SEARch:429:LEVEL <level>

:SEARch: 429:LEVEL?

其中，<level>，整型。

2.19 波形命令子系统

:WAVEform:BEIn

:WAVEform:DATA?

:WAVEform:END

:WAVEform:FORMat

:WAVEform:MODE

:WAVEform:POINts

:WAVEform:PREamble?

:WAVEform:RESet

:WAVEform:SOURce

:WAVEform:STARt

:WAVEform:STATus?

:WAVEform:STOP

:WAVEform:XINCrement?

:WAVEform:XORigin?

:WAVEform:XREFerence?

:WAVEform:YINCrement?

:WAVeform:YORigin?
:WAVeform:YREFerence?

2.19.1 :WAVeform:BEGin

命令格式:WAVeform:BEGin
功能描述 启动波形的读取。

2.19.2 :WAVeform:DATA?

命令格式 :WAVeform:DATA?

功能描述 读取波形数据。

该命令受:WAVeform:FORMat, :WAVeform:MODE, :WAVeform:POINts, :WAVeform:SOURce等命令设置的影响。

说明

屏幕波形数据读取流程:

- | | |
|-----------------------|------------|
| S1. :WAV:SOURce CHAN1 | 设置读取的源 |
| S2. :WAV:MODE NORM | 波形模式为 NORM |
| S3. :WAV:DATA? | 获取缓存中的数据 |

内存波形数据读取:

- | | |
|-----------------------|------------------|
| S1. :STOP | 内存波形只能在停止状态下进行读取 |
| S2. :WAV:SOURce CHAN1 | 设置读取的源 |
| S3. :WAV:MODE RAW | 波形模式为 RAW |
| S4. :WAV:RESet | 复位波形读取 |
| S5. :WAV:BEGin | 开始波形读取 |
| S6. :WAV:STATus? | 获取状态 |
| 1) IDLE | 波形读取线程结束 |
| :WAV:DATA? | 获取缓存中的数据 |
| :WAV: END | 波形读取完成 |
| 2) READ | 波形读取线程正在运行 |
| :WAV:DATA? | 获取缓存中的数据 |
| S6 | 继续读取波形数据 |

返回格式

读取到的数据由 3 个部分组成，分别是 TMC 数据描述头、数据长度和波形数据。

#9dddddddXXXX...q\n

其中的

#9 表示 TMC 数据描述头

dddddddd 表示数据流中有效的波形点数

XXXX...q 表述波形数据，例如 80\82\81\90\00\08\。。。。。

读取内存数据时，每次读回的数据可能只是内存中一块区域的数据。分块读回的数据，每块开头都含有类似 #9XXXXXXXX 的描述符，其中 XXXXXXXX 表示本次传输块中的波形点数。相邻两块间的波形数据连续。

2.19.3 :WAVeform:END

命令格式 :WAVeform:END

功能描述 停止波形的读取。

2.19.4 :WAVeform:FORMat

命令格式 :WAVeform:FORMat <format>

:WAVeform:FORMat?

功能描述 设置波形数据的返回格式。

参数

名称	类型	范围	默认值
<format>	离散型	{WORD BYTE}	BYTE

返回格式 查询返回“WORD”或“BYTE”。

举例

下面的命令选择 WORD 格式。

:WAVeform:FORMat WORD

下面的查询返回“WORD”。

:WAVeform:FORMat?

WORD: 是允许传输的数据是 16 位二进制，16 位二进制转换 16 进制，2 个字节；也就是说一个点是 2 个字节；如果数据小于 16 位，则右侧补 0；

BYTE: 允许传输的数据是 8 位二进制，转换成 16 进制就是 1 个字节；假如在高分辨率的情况下，数据超出 8 位，则取左侧部分；

如果数据有空洞，则用 0 表示；

2.19.5 :WAVeform:MODE

命令格式

:WAVeform:MODE <mode>
:WAVeform:MODE?

功能描述 设置波形的读取模式。

参数

名称	类型	范围	默认值
<mode>	离散型	{NORMal MAXimum RAW}	NORMal

说明

不同模式下，:WAVeform:POINts 命令返回的波形点数不一样。

NORMal: 返回当前显示的波形点数。

MAXimum: 返回当前状态下的最大有效数据点数。运行状态下返回屏幕显示的数据点数，停止状态下返回内存数据点数。

RAW: 返回当前系统内存数据点数。只在停止状态下有效。

返回格式 查询返回“NORM”、“MAX”或“RAW”。

举例

下面的命令选择 RAW 模式。

:WAVeform:MODE RAW

下面的查询返回“RAW”。

:WAVeform:MODE?

2.19.6 :WAVeform:POINts

命令格式

:WAVeform:POINts <point>
:WAVeform:POINts?

功能描述 设置需要读取的波形点数。受当前波形读取模式的限制（参考:WAVeform:MODE 命令）。

参数

名称	类型	范围	默认值
<point>	整型	1 至 最大存储深度	----

返回格式 查询返回一个整数。

举例

下面的命令设置返回点数为 2048。

```
:WAVEform:POINts 2048
```

下面的查询返回“2048”。

```
:WAVEform:POINts?
```

2.19.7 :WAVEform:PREamble?

命令格式 :WAVEform:PREamble?

功能描述 查询并返回全部的波形参数。

返回格式

查询返回以“,”间隔的 10 个波形参数:

<format>,<type>,<points>,<count>,<xincrement>,<xorigin>,<xreference>,<yincrement>,<yorigin>,<yreference>

<format>: 0 (BYTE) 或 1 (WORD)。参考:WAVEform:FORMat 命令。

<type>: 0 (NORMal)、1 (MAXimum) 或 2 (RAW)。参考:WAVEform:MODE 命令。

<points>: 1 至最大存储深度的整数。参考:WAVEform:POINts 命令。

<count>: 在平均采样方式下为平均次数 (参考:ACQuire:AVERages 命令), 其他方式下为 1。

<xincrement>: X 方向上的相邻两点之间的时间差。参考:WAVEform:XINCrement?命令。

<xorigin>: X 方向上从触发点到“参考时间基准”的时间。参考:WAVEform:XORigin?命令。

<xreference>: X 方向上数据点的参考时间基准。参考:WAVEform:XREFerence?命令。

<yincrement>: Y 方向上的单位电压值。参考:WAVEform:YINCrement?命令。

<yorigin>: Y 方向上相对于“垂直参考位置”(参考:WAVEform:YREFerence?命令)的垂直偏移。参考:WAVEform:YORigin?命令。

<yreference>: Y 方向的垂直参考位置。参考:WAVEform:YREFerence?命令。

举例

下面的查询返回“0,2,0,1,0.000000,-0.001488,0,0.062500,3.968750,127”。

```
:WAVEform:PREamble?
```

2.19.8 :WAVEform:RESet

命令格式 :WAVEform:RESet

功能描述 复位波形的读取。

2.19.9 :WAVeform:SOURce

命令格式 :WAVeform:SOURce <source>

:WAVeform:SOURce?

功能描述 设置波形读取的通道源。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3 CH4}	CHANnel1

返回格式

查询返回“CHAN1”、“CHAN2”、“CHAN3”或“CHAN4”。

举例

下面的命令选择通道 2 为通道源。

:WAVeform:SOURce CHANnel2

下面的查询返回“CHAN2”。

:WAVeform:SOURce?

2.19.10 :WAVeform:STARt

命令格式

:WAVeform:STARt <sta>

:WAVeform:STARt?

功能描述 设置内存中波形被读取的起始位置。

参数

名称	类型	范围	默认值
<sta>	整型	0 至(存储深度-1)	----

说明

有关“存储深度”的说明，请参考:ACQuire:MDEPth?命令。

返回格式 查询返回一个整数。

举例

下面的命令设置起始位置为 100。

:WAVeform:STARt 100

下面的查询返回“100”。

:WAVeform:STARt?

2.19.11 :WAVeform:STATus?

命令格式 :WAVeform:STATus?

功能描述 查询并返回当前的波形读取状态。
返回格式 查询返回“IDLE”或“READ”。

2.19.12 :WAVeform:STOP

命令格式

:WAVeform:STOP <sta>
:WAVeform:STOP?

功能描述 设置内存中波形被读取的停止位置。

参数

名称	类型	范围	默认值
<sta>	整型	1 至(存储深度-1)	----

说明 有关“存储深度”的说明，请参考:ACQuire:MDEPth?命令。

返回格式 查询返回一个整数。

举例 下面的命令设置停止位置为 200。

:WAVeform:STOP 200

下面的查询返回“200”。

:WAVeform:STOP?

2.19.13 :WAVeform:XINCrement?

命令格式

:WAVeform:XINCrement?

功能描述

查询指定源（参考:WAVeform:SOURce 命令）X 方向上相邻两点之间的时间差。单位为 s。

返回格式

查询以科学计数形式返回时间差值。

举例

下面的查询返回“2.000000e-08”。

```
:WAVeform:XINCrement?
```

2.19.14 :WAVeform:XORigin?

命令格式

```
:WAVeform:XORigin?
```

功能描述

查询指定源（参考:WAVeform:SOURce 命令）X 方向上从触发点到“参考时间基准”（参考:WAVeform:XREFerence?命令）的时间。单位为 s。

返回格式

查询以科学计数形式返回时间值。

举例

下面的查询返回“-7.000000e-06”。

```
:WAVeform:XORigin?
```

2.19.15 :WAVeform:XREFerence?

命令格式

:WAVeform:XREference?

功能描述

查询指定源（参考:WAVeform:SOURce 命令）X 方向上数据点的参考时间基准。单位为 s，采用科学计数法,同上。

返回格式

查询以整数形式返回时间基准。

举例

下面的查询返回“0”。

```
:WAVeform:XREference?
```

2.19.16 :WAVeform:YINCrement?

命令格式

:WAVeform:YINCrement?

功能描述

查询指定源（参考:WAVeform:SOURce 命令）Y 方向上的单位电压值。单位与源所选单位一致。

返回格式

查询以科学计数形式返回电压值。

举例

下面的查询返回“3.125000e-03V”。

```
:WAVeform:YINCrement?
```

2.19.17 :WAVeform:YORigin?

命令格式

```
:WAVeform:YORigin?
```

功能描述

查询指定源（参考:WAVeform:SOURce 命令）Y 方向上相对于“垂直参考位置”（参考:WAVeform:YREference?命令）的垂直偏移。单位与源所选单位一致。

返回格式

查询以科学计数形式返回偏移值。

举例

下面的查询返回“3.968750e+00V”。

```
:WAVeform:YORigin?
```

2.19.18 :WAVeform:YREference?(暂时不做)

命令格式

:WAVeform:YREference?

功能描述

查询指定源（参考:WAVeform:SOURce 命令）Y 方向的垂直参考位置。单位与源所选单位一致。

返回格式

查询以整数形式返回参考位置。

举例

下面的查询返回“127”。

```
:WAVeform:YREference?
```

错误信息

无读取值

超出范围

2.20 系统

2.20.1 :SYS: TOUCH

命令格式:SYS:TOUCh <event>,<x>,<y>

功能描述 屏幕事件

event 0 抬起无事件

event 1 鼠标左键按下

event 2 鼠标右键按下

x 鼠标横轴坐标

y 鼠标纵轴坐标

2.20.2 :SYS:SCReen?

命令格式 :SYS:SCReen?

功能描述 获取屏幕图像。受命令设置的影响。

说明

屏幕波形数据读取流程:

返回 BMP 格式的一帧图像。

返回格式

读取到的数据由 3 个部分组成，分别是 TMC 数据描述头、数据长度和波形数据。

#9dddddddXXXX...

其中的

#9 表示 TMC 数据描述头

dddddddd 表示数据流中有效的数据长度

XXXX...表述 bmp 数据，

如#9001920054XXXX...

#9 固定头数据

001920054 代表后面数据长度为 1920054 字节(800*600*4+54)

XXXX... 为完整的 bmp 数据。

注意。接收数据缓冲 要大于 1920054+11

2.20.3:WLANAP:OPEN

功能：打开或关闭 WIFI 功能。

格式：:WLANAP:OPEN<bool>

:WLANAP:OPEN?

其中，<bool>，布尔型，{{0|OFF}|{1|ON}}。

返回格式：查询返回“0”或“1”。

举例：

下面的命令设置打开 WLANAP 功能。

:WLANAP:OPEN ON

下面的命令返回 1.

:WLANAP:OPEN?

:CALibrate:START

命令格式 **:CALibrate:START**

功能描述 示波器开始执行自校准操作。

说明 ☐ 自校准操作可迅速使示波器达到最佳工作状态，以取得最精确的测量值。

☐ 执行自校准之前，请确保所有通道均未接入信号直至自校准操作结束。

☐ 自校准过程中，大部分按键的功能已经被禁用。您可以发送**:CALibrate:QUIT**命令终止自

校准操作。

:CALibrate:QUIT

命令格式 **:CALibrate:QUIT**

功能描述 在任意时刻放弃自校准操作。

相关命令 **:CALibrate:START**

2.21 AUTO 设置子系统

2.21.1 :AUTO:SET:CHANnel <bool>

功能：autoset 通道自动开启与关闭的使能

格式： **:AUTO:SET:CHANnel <bool>**

:AUTO:SET:CHANnel?

其中, <bool>, 布尔型, {{0|OFF}|{1|ON}}。

返回格式: 查询返回“0”或“1”。

举例:

下面的命令打开通道 1 的显示。

:AUTO:SET:CHANnel ON 或:AUTO:SET:CHANnel 1

下面的查询返回“1”。

:AUTO:SET:CHANnel?

2.21.2 :AUTO:SET:LEVel <level>

功能: auto 中通道自动开启与关闭的有效阈值

格式: :AUTO:SET:LEVel <level>

:AUTO:SET:LEVel?

其中, <level>, 实型, 0.001V~99V。

返回格式: 查询以科学计数形式返回有效阈值。

举例:

下面的命令设置有效阈值为 150mV。

:AUTO:SET:LEVnel 0.15

下面的查询返回“1.500000e-01”。

:AUTO:SET:LEVel?

2.21.3 :AUTO:SET:SOURce <source>

功能: autoset 执行时, 优先选择触发源的规则, 当前优先|最大值优先

格式: :AUTO:SET: SOURce <source>

:AUTO:SET:SOURce?

其中, <source>, 离散型, {CURrent|MAX}。

返回格式: 查询返回“CURrent”或“MAX”。

举例:

下面的命令设置自动时, 触发源最大值优先

:AUTO:SET:SOURce MAX

下面的查询返回“MAX”。

:AUTO:SET:SOURce?

2.21.4 :AUTO:RANge <bool>

功能：按 Auto 键执行的模式，分为 autoset 和 autorange

格式：:AUTO:RANge <bool>

:AUTO:RANge?

其中，<bool>，布尔型， {{0|OFF}}|{1|ON}}，0 的时候位 AutoSet 模式，1 的时候位 AutoRange 模式

返回格式：查询返回“0”或“1”。

举例：

下面的命令 Autorange 使能。

:AUTO:RANge ON 或:AUTO:RANge 1

下面的查询返回“1”。

:AUTO:RANge?

2.21.5 :AUTO:RANge:VERtical <bool>

功能：AutoRange 过程中垂直刻度系数是否自动

格式：:AUTO:RANge:VERtical <bool>

:AUTO:RANge:VERtical?

其中，<bool>，布尔型， {{0|OFF}}|{1|ON}}

返回格式：查询返回“0”或“1”。

举例：

下面的命令打开 VERtical 自动。

:AUTO:RANge:VERtical ON 或:AUTO:RANge:VERtical 1

下面的查询返回“1”。

:AUTO:RANge:VERtical?

2.21.6 :AUTO:RANge:HORizontal <bool>

功能：AutoRange 过程中时基是否自动

格式：:AUTO:RANge: HORizontal <bool>

:AUTO:RANge: HORizontal?

其中, <bool>, 布尔型, {{0|OFF}}|{1||ON}}

返回格式: 查询返回“0”或“1”。

举例:

下面的命令打开 HORizontal 自动。

:AUTO:RANge: HORizontal ON 或: AUTO:RANge: HORizontal 1

下面的查询返回“1”。

:AUTO:RANge: HORizontal?

2.21.6 :AUTO:RANge:LEVel <bool>

功能: AutoRange 过程中触发电平是否自动

格式: :AUTO:RANge: :LEVel <bool>

:AUTO:RANge: LEVel?

其中, <bool>, 布尔型, {{0|OFF}}|{1||ON}}

返回格式: 查询返回“0”或“1”。

举例:

下面的命令打开 LEVel 自动。

:AUTO:RANge:LEVel ON 或: AUTO:RANge:LEVel 1

下面的查询返回“1”。

:AUTO:RANge:LEVel?

3 编程实例

本章例举了在 Excel、Matlab、LabVIEW、Visual Basic 6.0 和 Visual C++ 6.0 开发环境中如何使用命令实现示波器常用功能的编程实例。这些实例都是基于 VISA (Virtual Instrument Software Architecture) 库编程实现的。

本章主要内容：

- Excel 编程实例
- Matlab 编程实例
- LabVIEW 编程实例
- Visual Basic 编程实例
- Visual C++编程实例

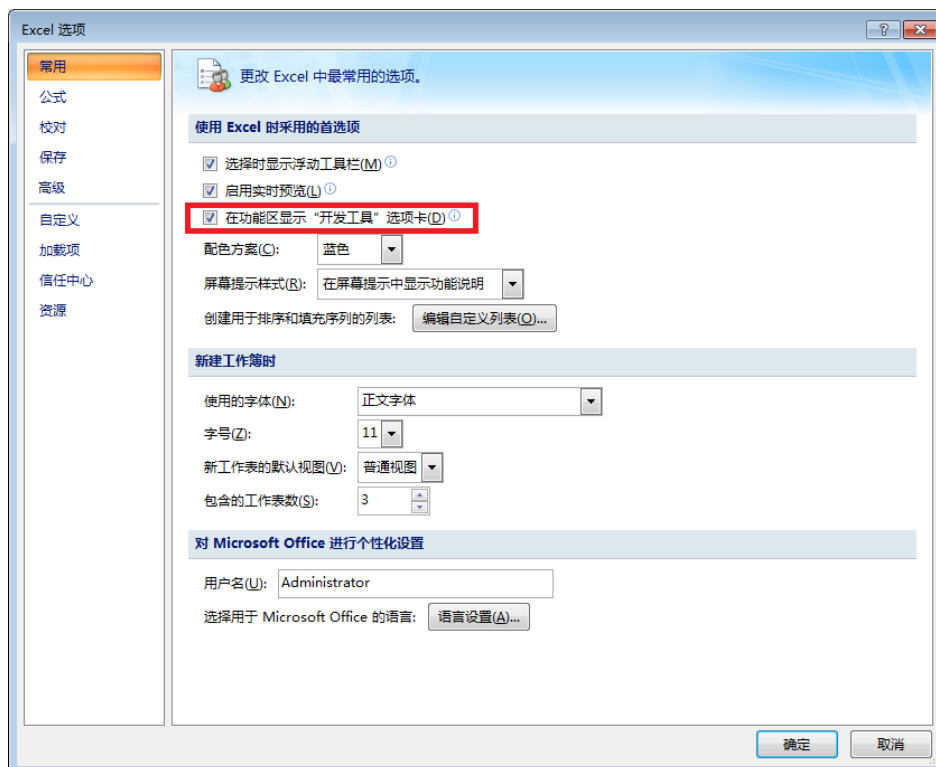
Excel 编程实例

本例使用的程序： Microsoft Excel 2007

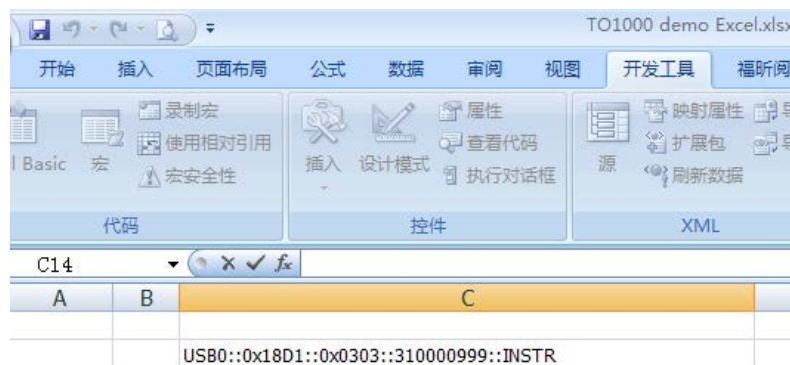
本例实现的功能： 发送*IDN?命令，读取设备信息。

新建一个启用宏的 Excel 文件，本例中命名为 TO1000_Demo_Excel.xlsm。

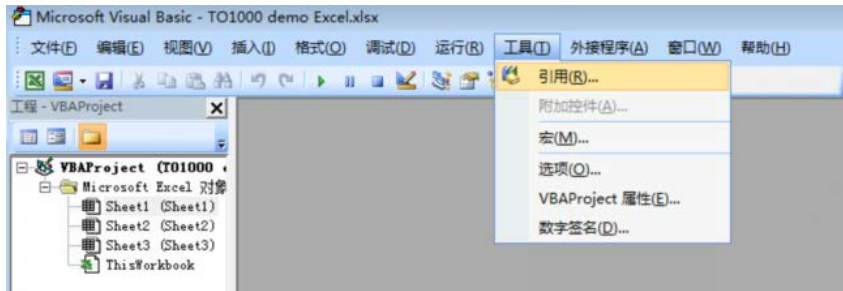
运行 TO1000_Demo_Excel.xlsm 文件，单击 Excel 文件左上角的 Office 按钮，点击“Excel 选项”，打开如下图所示界面，勾选“在功能区显示“开发工具”选项卡(D)”，点击“确定”。此时，Excel 的菜单栏将显示“开发工具”菜单。



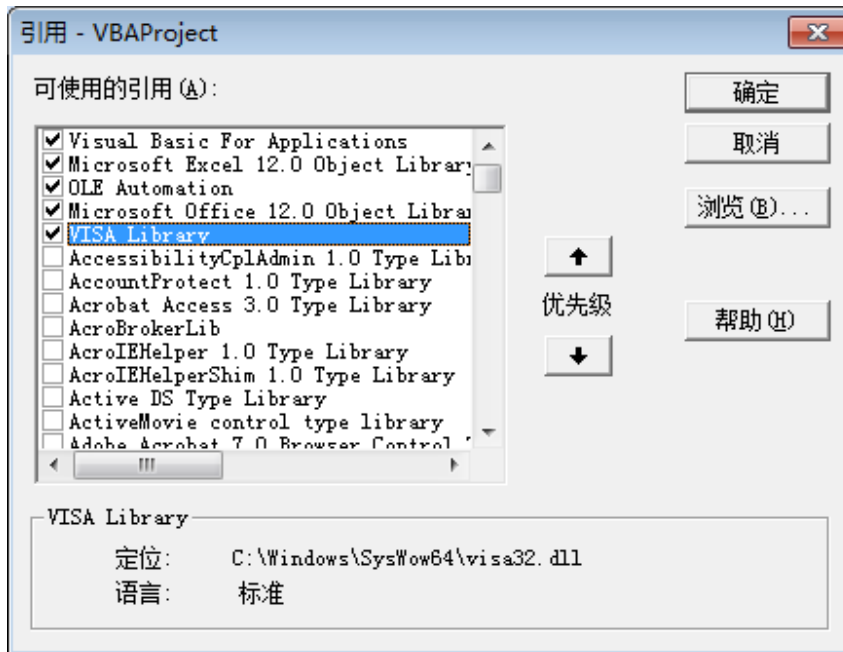
在 Excel 的一个单元格中填入设备资源描述符，例如设备资源描述符为 USB0::0x18D1::0x0303::310000999::INSTR，将其填入 SHEET1.CELLS(2,3)（即 Sheet1 的 C2 单元格）中。单击“开发工具”菜单选择 Visual Basic 选项，打开 Microsoft Visual Basic。



1. 在 Visual Basic 页面的菜单栏选择“工具（T）”单击“引用（R）”。



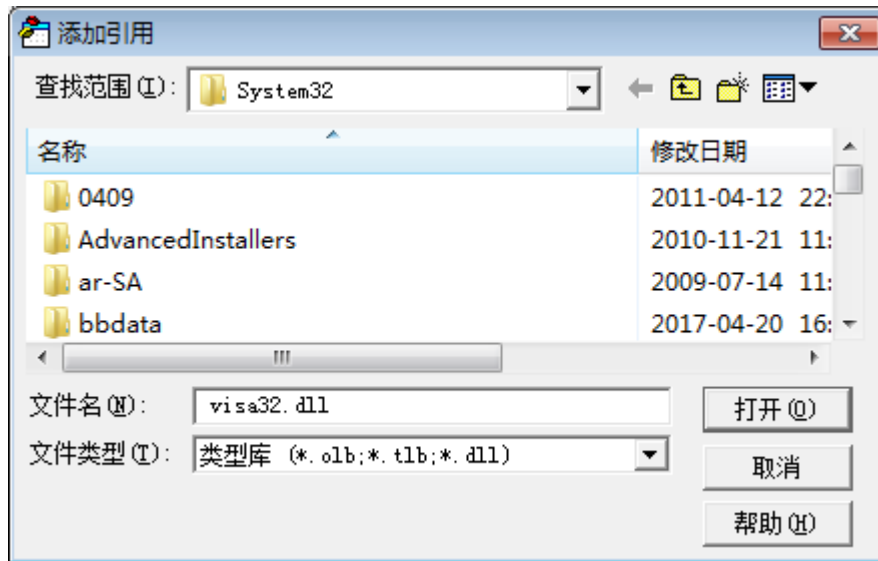
在弹出的对话框中选中 VISA Library，单击确定按钮即可引用 VISA Library。



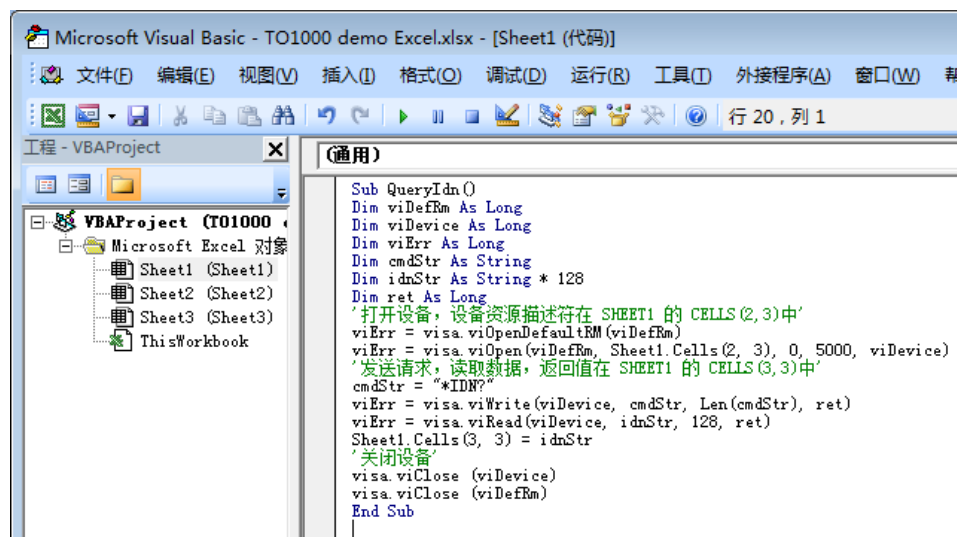
说明：

如果您在上图左侧的列表中无法找到 VISA Library，请按照如下方法查找：

- (1) 请确保您的计算机已经安装 NI-VISA 库。
- (2) 点击右侧的“浏览（B）...”进行查找，查找范围为 C:\WINDOWS\system32，文件名为 visa32.dll，如下图所示：



2. 在“开发工具”菜单下点击“查看代码”，进入 Microsoft Visual Basic 页面，添加如下代码并保存。



说明：

若第 2 步新建的 Excel 文件不是启用宏的文件，此时，将弹出“无法在未启用宏的工作簿中保存以下功能”的提示消息，此时，请根据提示将工作簿保存为启用宏的文件（文件后缀名变成.xlsx）即可。

代码文本：

```
Sub QueryIdn()  
Dim viDefRm As Long  
Dim viDevice As Long  
Dim viErr As Long  
Dim cmdStr As String  
Dim idnStr As String * 128
```

Dim ret As Long

‘打开设备，设备资源描述符在 SHEET1 的 CELLS(2,3)中’

viErr = visa.viOpenDefaultRM(viDefRm)

viErr = visa.viOpen(viDefRm, Sheet1.Cells(2, 3), 0, 5000, viDevice)

‘发送请求，读取数据，返回值在 SHEET1 的 CELLS(3,3)中’

cmdStr = "*IDN?"

viErr = visa.viWrite(viDevice, cmdStr, Len(cmdStr), ret)

viErr = visa.viRead(viDevice, idnStr, 128, ret)

Sheet1.Cells(3,3) = idnStr

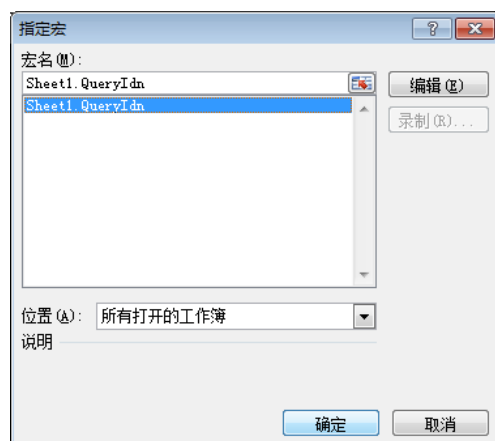
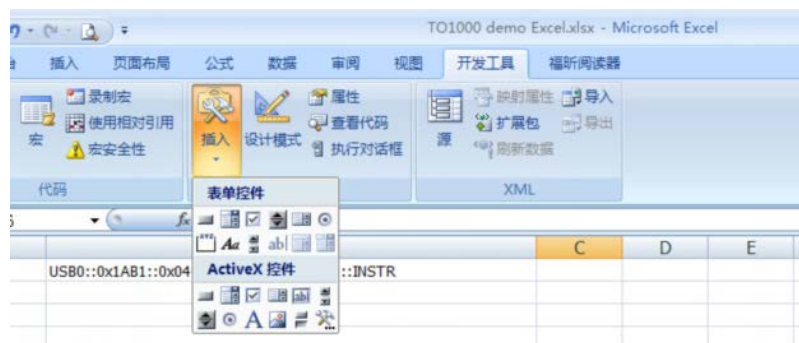
‘关闭设备’

visa.viClose (viDevice)

visa.viClose (viDefRm)

End Sub

3. 添加按钮控件： 在“开发工具”菜单下点击“插入”， 在“ 表单控件” 选择按钮后放在 Excel 单元格中。此时，弹出“指定宏”界面，选中“Sheet1.QueryIdn”， 单击“确定”即可。



按钮默认的名称为“按钮 1”。右击按钮，在弹出的菜单中先选择“编辑文字(X)”，将按钮名称改为“ *IDN?”。

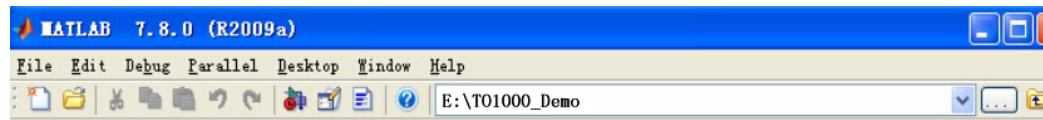
4. 点击“ *IDN?”按钮，即发送请求，读取数据，返回值在 SHEET1 的 CELLS (3,3) 中。

Matlab 编程实例

本例使用的程序： MATLAB R2009a

本例实现的功能： 对波形数据进行 FFT 计算，并绘制波形。运行 Matlab 软件并修改。

1. 运行 Matlab 软件并修改当前路径（即修改软件上方的 Current Directory）。本实例将当前路径修改为 E:\TO1000_Demo。



2. 点击 Matlab 界面的 File → New → Blank M-File 创建一个空白的 M 文件。

3. 在 M 文件中添加如下代码：

%创建 VISA 对象

```
TO1000 = visa('ni','USB0::0x18D1::0x0303::310000999::INSTR');
```

%设置设备属性，本例中设置输入缓存的长度为 2048

```
TO1000.InputBufferSize = 2048;
```

%打开设备

```
fopen(TO1000);
```

%读取波形

```
fprintf(TO1000, ':wav:data? ');
```

%请求数据

```
[data,len]= fread(TO1000,2048);
```

%关闭设备

```
fclose(TO1000);
```

```
delete(TO1000);
```

```
clear TO1000;
```

%数据处理。读取的波形数据含有 TMC 头，长度为 11 个字节，其中前 2 个字节分别为 TMC 头标志符#和宽度描述符 9，接着的 9 个字节为数据长度，然后是波形数据，最后一个字节为结束符 0x0A。所以，读取的有效波形数据点为 12 到倒数第 2 个点。

```
wave = data(12:len-1);
```

```
wave = wave';
```

```
subplot(211);
```

```
plot(wave);
```

```
fftSpec = fft(wave',2048);
```

```
fftRms = abs(fftSpec);
```

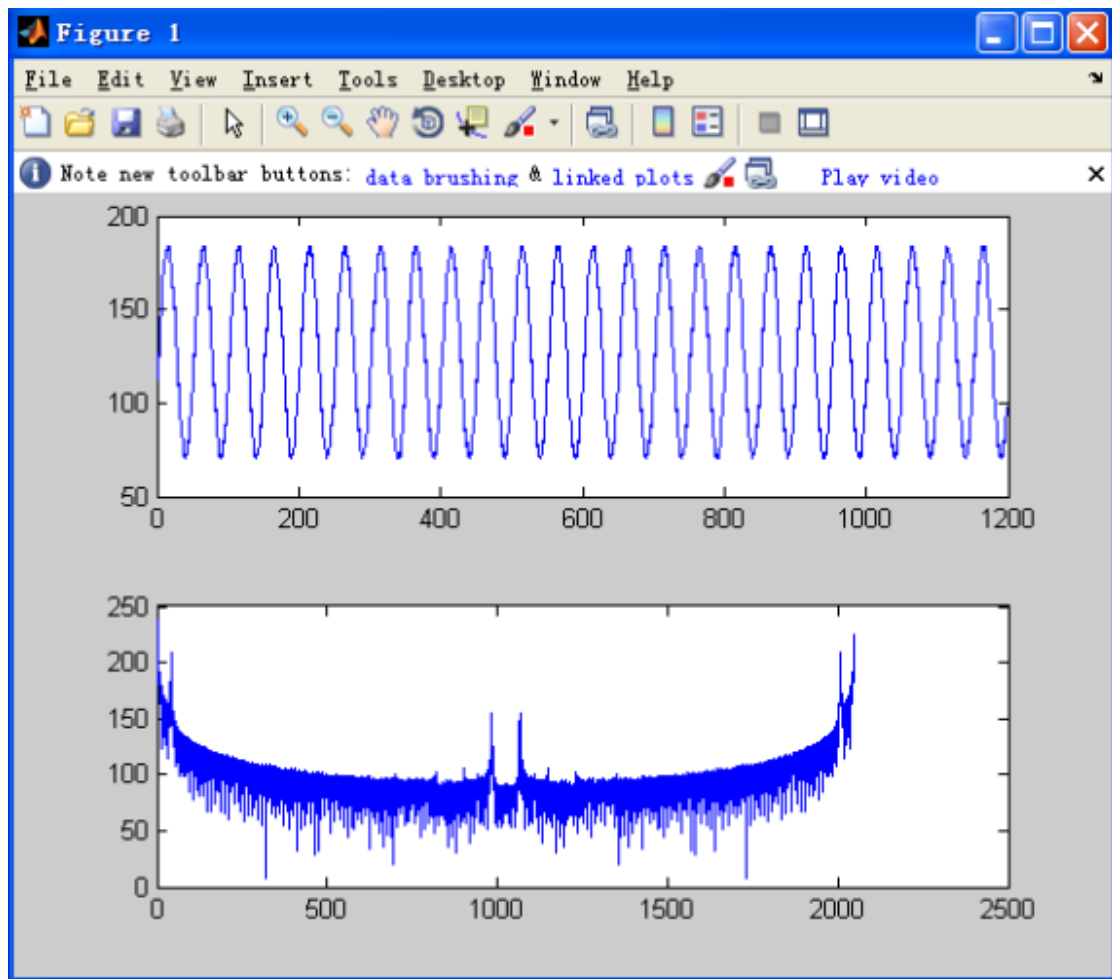
```
fftLg = 20*log(fftRms);
```

```
subplot(212);
```

```
plot(fftLg);
```

4. 将 M 文件保存在当前路径下。本实例的 M 文件命名为 TO1000_Demo_MATLAB.m。

5. 运行 M 文件，显示如下运行结果：

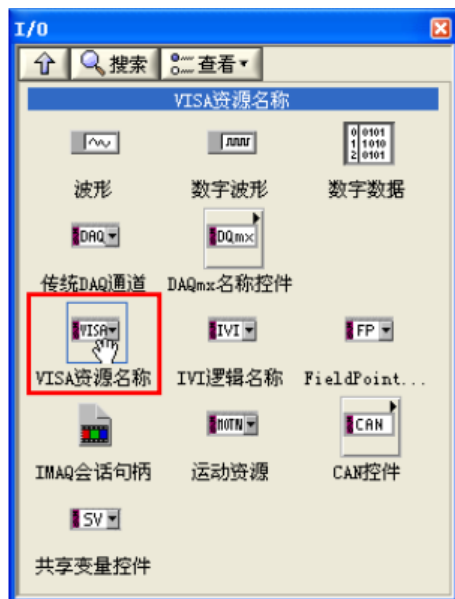


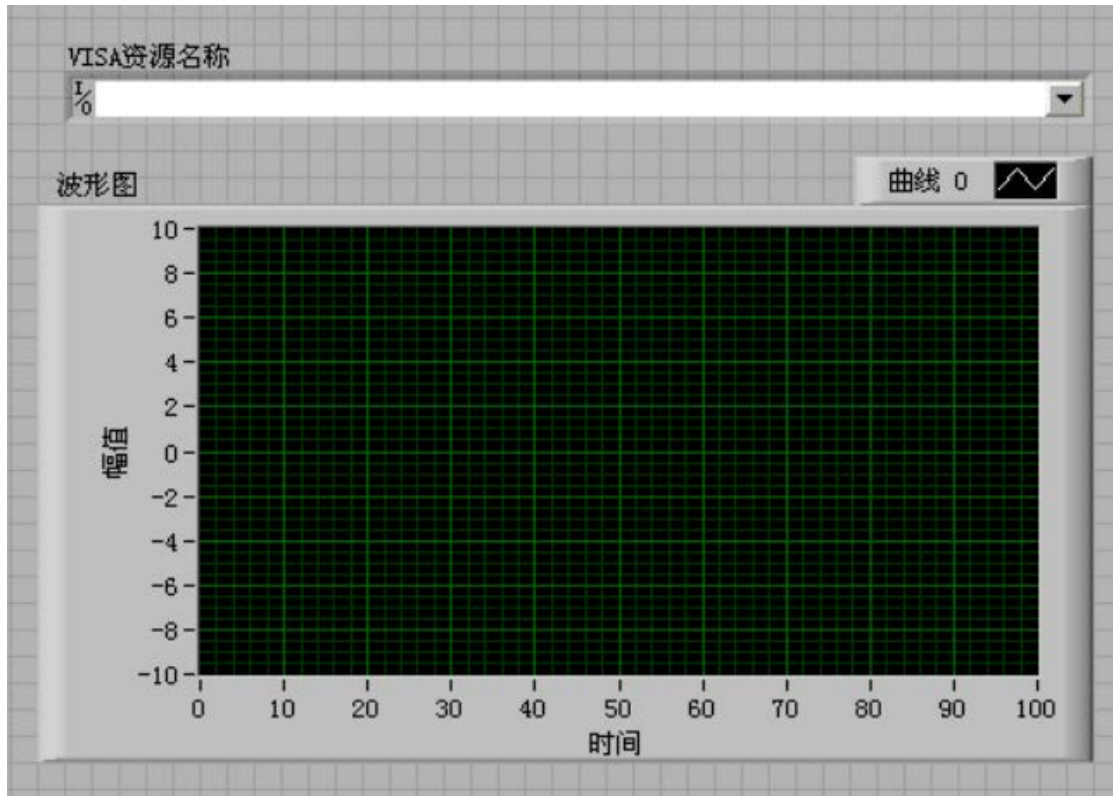
LabVIEW 编程实例

本例使用的程序： LabVIEW 8.2

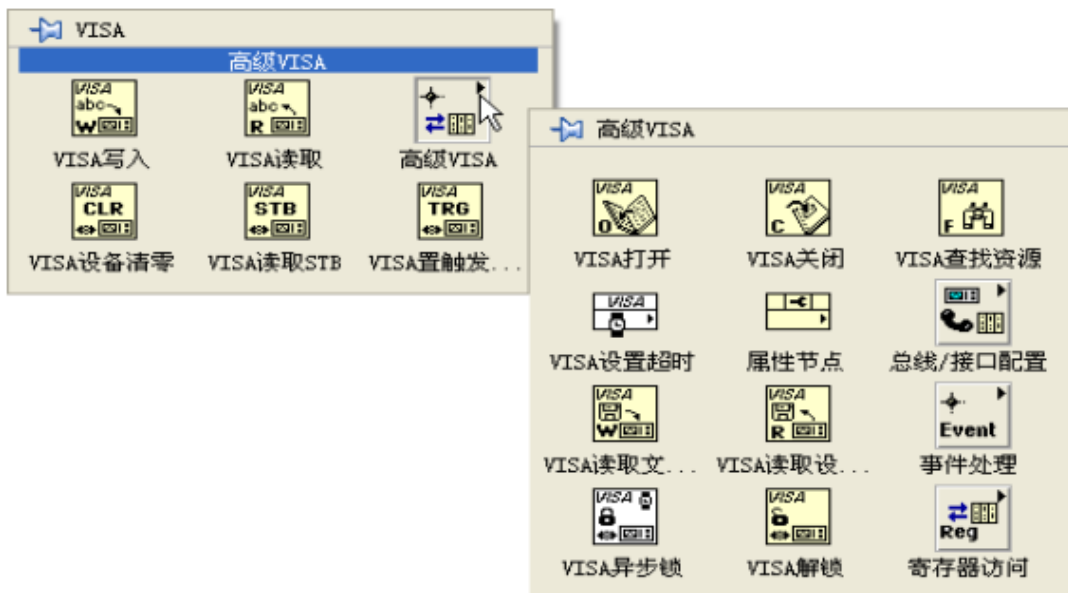
本例实现的功能： 读取 CH1 的屏幕波形数据。

1. 运行 LabVIEW 8.2，新建一个 VI 文件，命名为 TO1000_Demo。
2. 添加控件。在前面板界面右击，I/O 中选择 VISA 资源名称和图形中选择“波形图”，完成添加，如下列图所示：





3. 打开程序框图面板，选择仪器 I/O>VISA 分别添加以下函数， VISA 写入、 VISA 读取、 VISA 打开、 VISA 关闭函数。

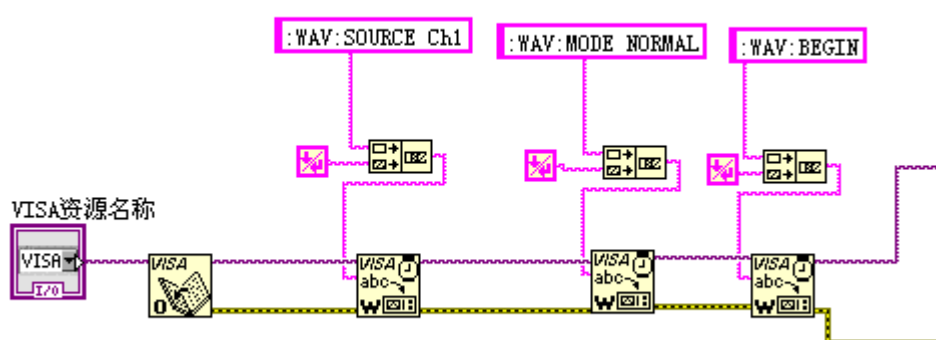


4. 将 VISA 资源名和 VISA 打开相连，将所有函数的 VISA 资源名称输出和 VISA 资源名称连接，错误输出和错误输入连接，如下图所示：

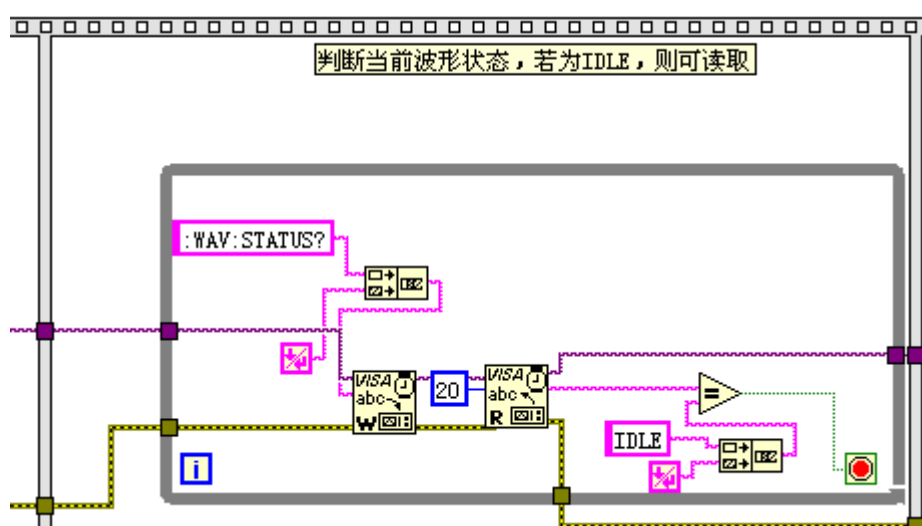
VISA资源名称



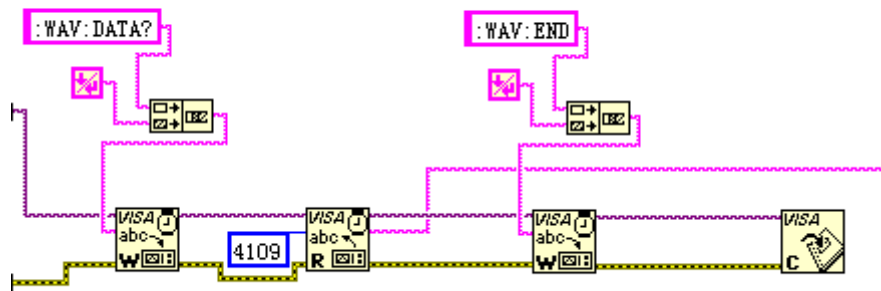
5. 在 VISA 写入控件的写入缓冲区添加文本框，分别写入：“:WAV:SOUR CH1”、“:WAV:MODE NORMAL”和“:WAV:BEGin”。第一条命令设置通道源为 CH1，第二条命令是设置普通模式读取取状态，第三条命令是下发开始读取指令。这三条共同完成读取前的配置工作。



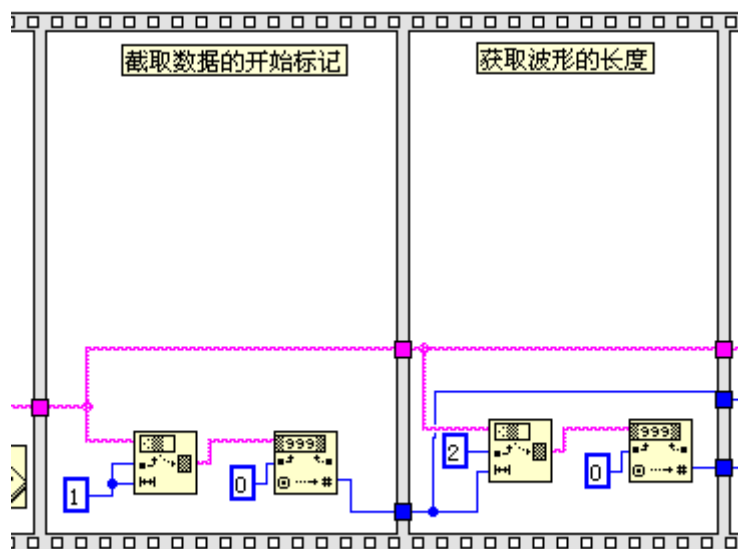
6. 发送完开始读取命令，需查询当前波形状态，当读取到的波形状态为 IDLE 时，才可读取波形数据。

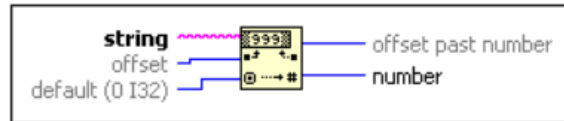
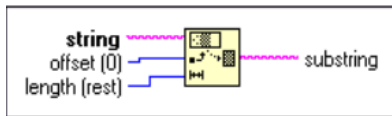
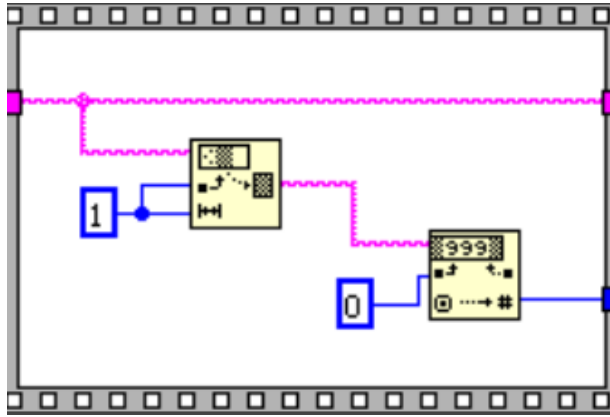


7. 在 VISA 写入控件的写入缓冲区添加文本框，写入：“:WAV:DATA?”，读取波形数据通过 VISA 读取函数完成，VISA 读取函数要求输入读取的字节总数，本例中读取的波形数据长度总字节数等于 4109（13 开始字节+4096 波形数据），读取完波形后，发送“:WAV:END”结束读取波形，VISA 操作完成后关闭 VISA 资源。

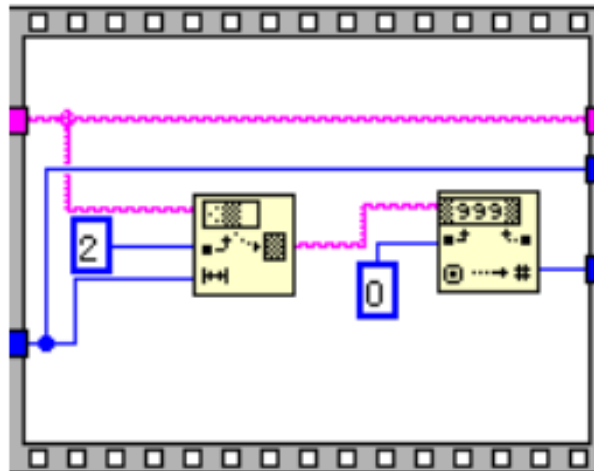


6.读取的数据格式为TMC头+波形数据点+结束符。TMC头为#NXXXXXX 的形式,#为 TMC 规定的头标志符,N 表示后面含有 N 个字节,以 ASCII 字符的形式描述波形数据点的长度,结束符用于表示通讯的终止。例如,一次读取的数据为: #90000004096XXXX 表示 9 个字节描述数据的长度,000004096 表示波形数据的长度,即 4096 字节。提取出 N 的数值,通过使用“截取字符串”和“十进制数字字符串至数值转换”两个函数完成。

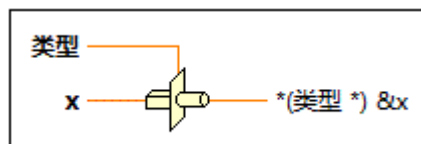
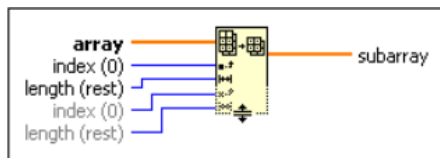


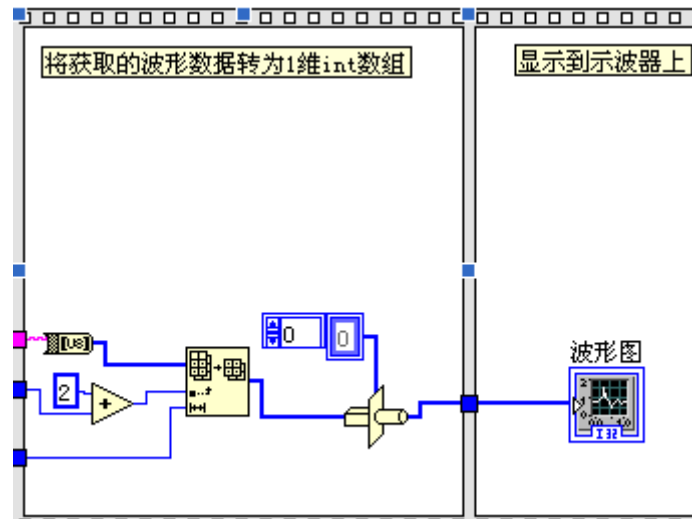


提取出有效的波形数据长度：

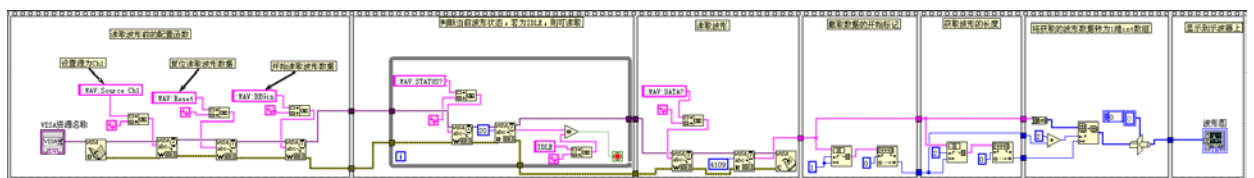


7 通过“字符串至字节数组转换”将字符数据转换为数组形式，然后通过“数组子集”函数去掉头部的 TMC 数据头，然后通过强制转换，将 U8 数组转为 I32 数组，即可在波形图控件上显示波形数据。





8.完整的程序框图如下所示:



9.在 VISA 资源名称列表框中选择设备资源，启动运行。

Visual Basic 编程实例

本例使用的程序： Visual Basic 6.0

本例实现的功能： 控制任意一个通道的开关状态。

进入 Visual Basic 6.0 编程环境，按照下列步骤操作：

1. 建立一个标准应用程序工程（ Standard EXE），命名为 Demo。
2. 打开 Project>Add Module 的 Existing 选项卡，找到之前 NI-VISA 安装路径下的 include 文件夹中的 visa32.bas 文件并添加。



3. 在 Demo 中添加如下四个按钮，分别代表 CH1~CH4。添加四个 Label: Label1(0), Label1(1), Label1(2), Label1(3), 分别显示 CH1~CH4 的状态（打开时显示通道的颜色，关闭时显示成灰色）。如下图所示：



4. 打开 Project+Project1 Properties 中的 General 选项卡，在 Startup Object 下拉框中选择 Form1。
5. 双击 CH1 按钮进入编程环境，添加如下代码，即可实现对 CH1~CH4 的控制。以下为 CH1 的代码， 其它通道代码类似。

```

Dim defrm As Long
Dim vi As Long
Dim strRes As String * 200
Dim list As Long
Dim nmatches As Long

Dim matches As String * 200

'保留获取设备号
Dim s32Disp As Integer
' 获得 visa 的 usb 资源
Call viOpenDefaultRM(defrm)
Call viFindRsrc(defrm, "USB?* ", list, nmatches, matches)
' 打开设备
Call viOpen(defrm, matches, 0, 0, vi)
' 发送询问 CH1 状态命令
Call viVPrintf(vi, ":CHAN1:DISP?" + Chr$(10), 0)
' 获取 CH1 状态
Call viVScanf(vi, "%t", strRes)
s32Disp = CInt(strRes)
If (s32Disp = 1) Then
' 发送设置命令
Call viVPrintf(vi, ":CHAN1:DISP 0" + Chr$(10), 0)
Label1(0).ForeColor = &H808080 '灰色
Else
Call viVPrintf(vi, ":CHAN1:DISP 1" + Chr$(10), 0)
Label1(0).ForeColor = &HFFFF& '黄色
End If
' 关闭资源
Call viClose(vi)
Call viClose(defrm)

```

6. 保存、运行整个工程，可得到 demo 的单个可执行程序。当示波器与 PC 成功相连时，可实现对任意一个通道的开/关控制。

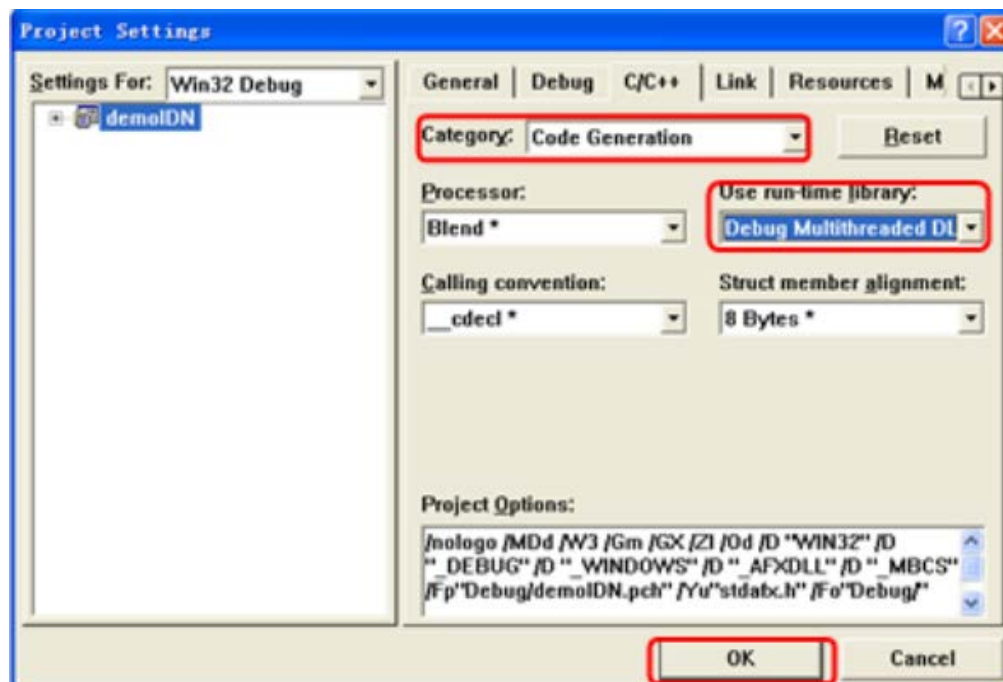
Visual C++编程实例

本例使用的程序： Visual C++6.0

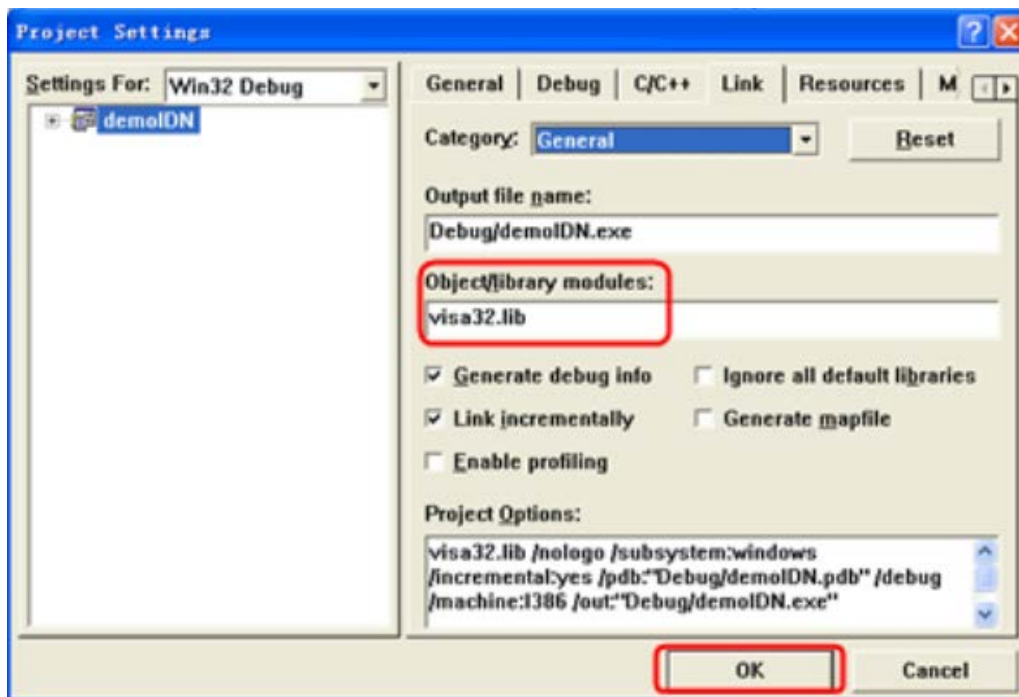
本例实现的功能： 查找仪器地址、连接仪器、发送命令并读取返回值。

进入 Visual C++6.0 编程环境，按照下列步骤操作：

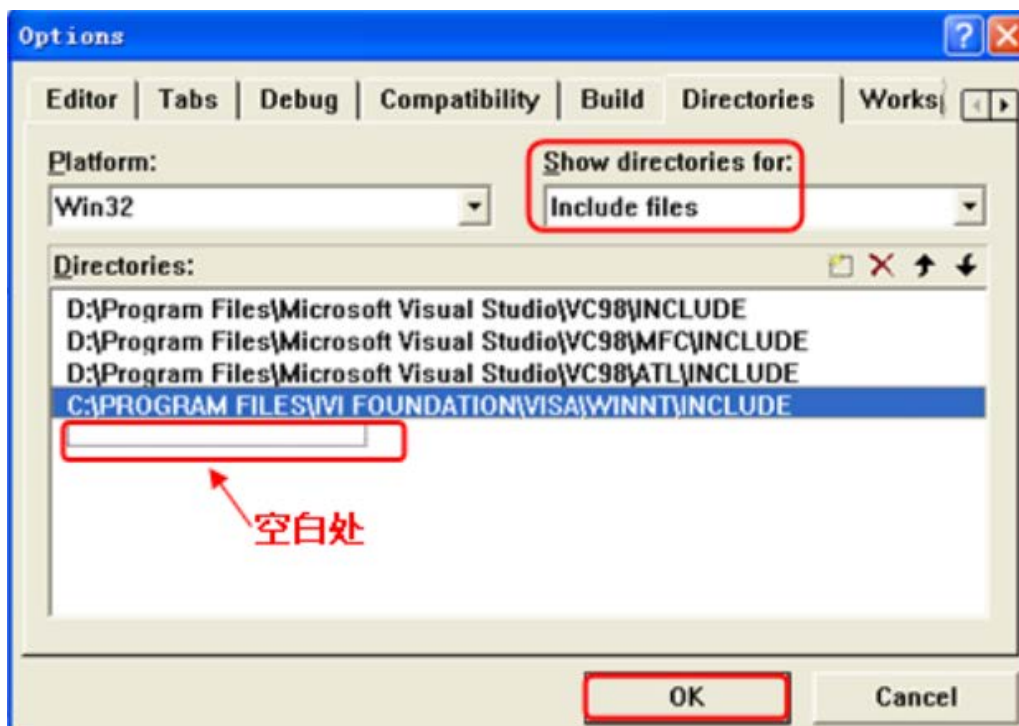
1. 建立一个基于对话框的 MFC 的工程。
2. 打开 Project\Settings 中的 C/C++选项卡，在 Category 中选 Code Generation，在 Use run-time library 中选 Debug Multithreaded DLL。点 OK 关闭对话框。



3. 打开 Project\Settings 中的 Link 选项卡，在 Object/library modules 中手动添加 visa32.lib。



4. 打开 Tools\Options 中的 Directories 选项卡。在 Show directories for 中选择 Include files，双击 Directories 选框中的空白处添加 Include 的路径：C:\Program Files\IVI Foundation\VISA\WinNT\include。在 Show directories for 中选择 Library files，双击 Directories 选框中的空白处添加 Lib 的路径：C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc。



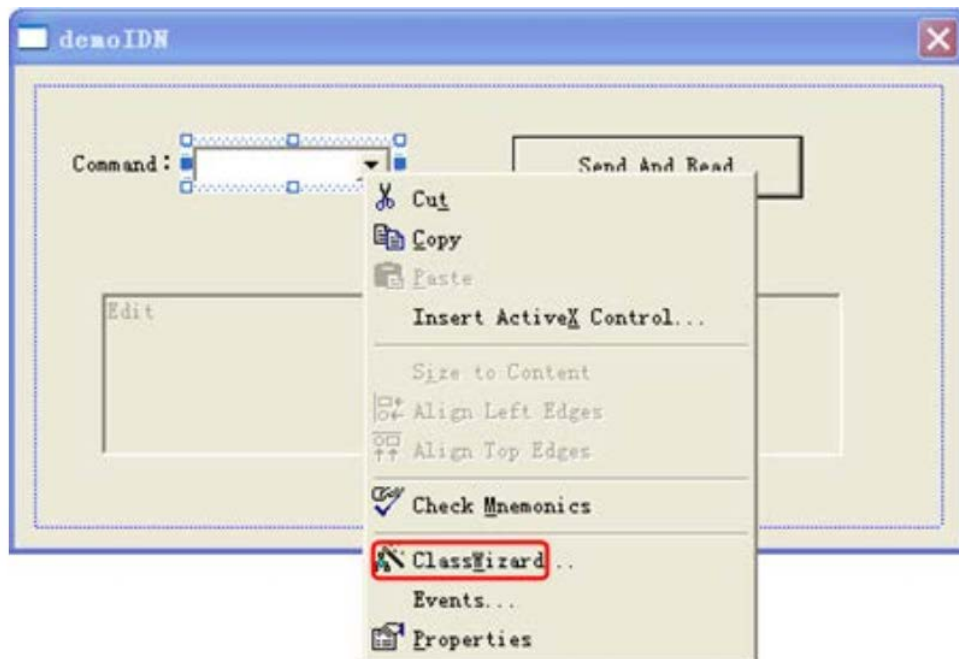
说明：到这里， VISA 库添加完毕。

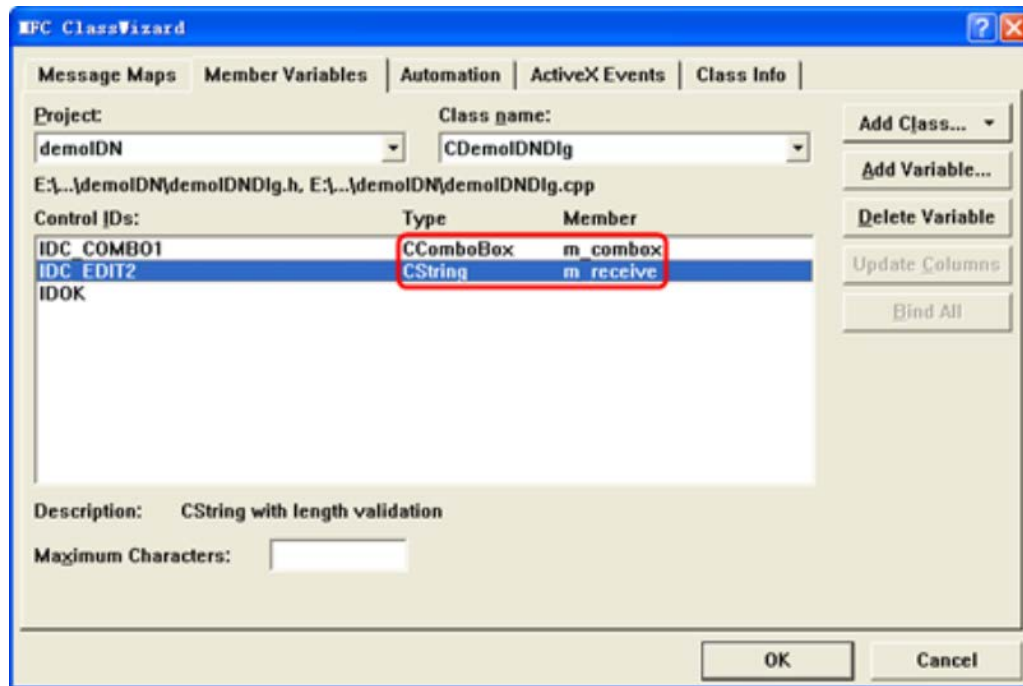
5. 添加 Text、 Combo Box、 Button 和 Edit Box 控件。布局如下所示：



6. 修改控件属性。

- 1) 将 Text 命名为“Command”。
 - 2) 打开 Combo Box 属性中的 Data 项，手动输入命令： *IDN?
 - 3) 打开 Edit Box 属性中的 General 项，选中 Disabled。
 - 4) 将 Button 命名为 Send and Read。
7. 为 Combo Box 和 Edit 控件分别添加变量 m_combox 和 m_receive。





8. 添加代码。

双击“Send and Read”进入编程环境，首先请在头文件中对 visa 库“#include <visa.h>”进行声明，然后添加如下代码：

```
ViSession defaultRM, vi;
char buf [256] = {0};
CString s,strTemp;
char* stringTemp;
ViChar buffer [VI_FIND_BUFLLEN];
ViRsrc matches=buffer;
ViUInt32 nmatches;
ViFindList list;
viOpenDefaultRM (&defaultRM);

//获取 visa 的 USB 资源
viFindRsrc(defaultRM, "USB?*", &list,&nmatches, matches);
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);
//发送接收到的命令
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);
strTemp = strTemp + "\n";
stringTemp = (char*)(LPCTSTR)strTemp;
viPrintf (vi,stringTemp);
//读取结果
viScanf (vi, "%t\n", &buf);
//将结果显示出来
UpdateData (TRUE);
m_receive = buf;
```

```
UpdateData (FALSE);  
viClose (vi);  
viClose (defaultRM);
```

9.保存、编译和运行工程，可得到单个可执行文件。当示波器与 PC 成功相连时，选择 *IDN?按“ Send and Read”按键，将显示示波器返回的结果。

