

RIGOL

编程手册

DS1000E, DS1000D 系列数字示波器

2013 年 6 月

RIGOL Technologies, Inc.

保证和声明

版权

© 2009 北京普源精电科技有限公司版权所有。

商标信息

RIGOL 是北京普源精电科技有限公司的注册商标。

文档编号

PGA07005-1110

声明

- 本公司产品受已获准及尚在审批的中华人民共和国专利的保护。
- 本公司保留改变规格及价格的权利。
- 本手册提供的信息取代以往出版的所有资料。
- 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，**RIGOL** 概不负责。
- 未经 **RIGOL** 事先书面许可不得影印复制或改编本手册的任何部分。

产品认证

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

联系我们

如您在使用此产品的过程中有任何问题或需求，可与 **RIGOL** 联系：

服务与支持热线：400 620 0002

网址：www.rigol.com

目 录

| | |
|-------------------------|------------|
| 第 1 章 编程概述 | 1-1 |
| 通信接口 | 1-2 |
| 命令介绍 | 1-3 |
| 命令格式..... | 1-3 |
| 符号说明..... | 1-3 |
| 命令输入..... | 1-4 |
| 参数类型..... | 1-4 |
| 第 2 章 命令系统 | 2-1 |
| 通用命令 | 2-2 |
| SYSTem 命令 | 2-4 |
| ACQuire 命令 | 2-6 |
| DISPlay 命令..... | 2-10 |
| TIMebase 命令..... | 2-15 |
| TRIGger 命令..... | 2-19 |
| 触发控制..... | 2-22 |
| 边沿触发..... | 2-26 |
| 脉宽触发..... | 2-27 |
| 视频触发..... | 2-29 |
| 斜率触发..... | 2-32 |
| 码型触发..... | 2-35 |
| 持续时间触发..... | 2-36 |
| 交替触发..... | 2-38 |
| STORage 命令..... | 2-47 |
| MATH 命令..... | 2-49 |
| CHANnel 命令 | 2-52 |
| MEASure 命令..... | 2-58 |
| WAVEform 命令..... | 2-67 |
| LA 命令..... | 2-70 |
| KEY 命令 | 2-74 |
| 其他命令 | 2-88 |
| 第 3 章 编程实例 | 3-1 |
| 编程准备 | 3-1 |

| | |
|-----------------------------|----------|
| 基于 RIGOL USB 驱动的编程 | 3-2 |
| Visual C++ 6.0 编程实例 | 3-2 |
| Visual Basic 6.0 编程实例 | 3-7 |
| 基于 VISA 库的编程 | 3-16 |
| Visual C++ 6.0 编程实例 | 3-16 |
| Visual Basic 6.0 编程实例 | 3-21 |
| LabVIEW 8.2 编程实例 | 3-23 |
| 附录 命令速查 A-Z | 1 |

第1章 编程概述

本章介绍如何通过远程接口命令对 DS1000E, DS1000D 系列数字示波器进行编程操作，以及接口命令的格式和输入方法。

本章内容如下：

- 通信接口
- 命令介绍
 - 命令格式
 - 符号说明
 - 命令输入
 - 参数类型

通信接口

DS1000E, DS1000D 系列数字示波器可通过 USB 或 RS232 接口与计算机进行通信。程序编译所使用的命令通常是以 ASCII 字符串形式植入主机中以便用户进行操控的。

您可以使用计算机和示波器进行的操作包括：

- 设置示波器；
- 相关测量；
- 从示波器获得数据（信号波形和测量数据）。

设备连接：

- USB：使用 USB 数据线将示波器与计算机相连。
- RS232：使用 RS232 串口线将示波器与计算机相连。

命令介绍

命令格式

DS1000E, DS1000D系列数字示波器的命令系统为树状层次结构，每个子系统由一个根关键字和一个或多个层次关键字构成。命令行通常以冒号“:”开始（IEEE命令除外），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；命令和参数要以“空格”分开。

例如：

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

TRIGger是命令的根关键字，EDGE和SLOPe分别是第二级、第三级关键字，各级关键字用冒号“:”分开。大括号中的内容表示用户可设置的参数；问号“?”表示查询；命令；:TRIGger:EDGE:SLOPe和参数之间用“空格”分开。在一些带多个参数的命令中，通常用逗号“,”分隔参数，例如：

```
:TRIGger:DURation:PATtern <value>,<mask>
```

符号说明

下面的符号不是命令中的内容，但是通常用于辅助说明命令中的参数。

1. 大括号 { }

大括号中的内容为命令中的必选参数，每次只能从中选择一项。各个参数间以竖线“|”分隔。

例如：:MEASure:TOTal {ON|OFF}，该命令可选择 ON 或 OFF 作为参数。

2. 方括号 []

方括号中的内容为可选项。

例如：:TIMebase[:DELayed]:OFFSet <offset>，该命令中的[:DELayed]可省略。

3. 三角括号 < >

三角括号中的参数必须用一个有效值来替换。

例如：:DISPlay:BRIGhtness <ncount>，该命令中的<ncount>要用一个数值来代替，如:DISPlay:BRIGhtness 25

命令输入

DS1000E, DS1000D 系列所有的命令对大小写输入敏感。您可以采用全拼大写或缩写模式输入命令。请注意：若要使用命令缩写形式，必须完整输入命令格式中指定的大写字母，例如：

```
:TRIGger:EDGE:SLOPe
```

可缩写成：

```
:TRIG:EDGE:SLOP
```

参数类型

命令中通常有 5 种参数类型，参数的设置方法根据参数类型而定。

1. 布尔型

参数取值为“OFF”或“ON”。例如：`:MEASure:TOTal {ON|OFF}`
“ON”表示打开（启用）功能，“OFF”表示关闭（禁用）功能。

2. 连续整型

参数取值为连续的整数，例如：`:DISPlay:BRIGhtness <ncount>`
<ncount>可取 0~32（包含 0 和 32）之间的整数。

3. 连续实型

参数在有效值范围内和精度要求下，可以任意取值。例如：
`:TRIGger:EDGE:SENSitivity <count>`
<count>可取 0.1~1（包含 0.1 和 1）之间的任意实数。

4. 离散型

参数取值为所列举的值，例如：`:ACQuire:AVERages <count>`
<count>只能取 2, 4, 8, 16, 32, 64, 128, 256。

5. ASCII 字符串

参数取值为 ASCII 字符的组合。例如：`:TRIGger:MODE <mod>`
<mod>可取“EDGE”、“PULSe”、“VIDEO”、“SLOPe”、“PATtern”、“DURation”或“ALternation”。

第2章 命令系统

本章将详细介绍 DS1000E, DS1000D 系列数字示波器命令系统中的每一条命令，包括命令格式、功能描述、使用方法以及相关注意事项。

DS1000E, DS1000D 系列包含的命令系统如下：

- 通用命令
- SYSTem 命令
- ACQuire 命令
- DISPlay 命令
- TIMebase 命令
- TRIGger 命令
 - 触发控制*
 - 边沿触发*
 - 脉宽触发*
 - 视频触发*
 - 斜率触发*
 - 码型触发*
 - 持续时间触发*
 - 交替触发*
- STORage 命令
- MATH 命令
- CHANnel 命令
- MEASure 命令
- WAVeform 命令
- LA 命令
- KEY 命令
- 其他命令

通用命令

IEEE标准定义了一些用于查询仪器基本信息或执行常用基本操作的通用命令，这些命令通常以“*”开头，命令关键字长度为3个字符。

DS1000E, DS1000D 系列数字示波器支持的通用命令包括：

- *IDN?
- *RST

下面将详细介绍每条命令的格式、功能和使用方法。

1. *IDN?

命令格式:

*IDN?

功能描述:

该命令查询仪器 ID 字符串。返回包含 4 个由逗号 “,” 分隔的字段：制造商，型号，序列号，由点号分隔并由数字组成的版本号。

返回格式:

RIGOL TECHNOLOGIES,<示波器型号>,<产品序列号>,<软件版本号>

举例:

RIGOL TECHNOLOGIES,DS1102E,DS1EB104702974,00.02.01.01.00

2. *RST

命令格式:

*RST

功能描述:

该命令用于设置系统参数复位。

SYSTEM命令

SYSTEM 命令用于对示波器进行最基本的操作，主要包括运行控制、屏幕数据截取等操作。

SYSTEM 命令包括：

- :RUN
- :STOP
- :AUTO
- :HARDcopy

下面将详细介绍每条命令的格式、功能和使用方法。

1. :RUN

命令格式:

:RUN

功能描述:

执行该命令，示波器将开始波形采样工作，如需停止工作，请执行:**STOP**命令。

2. :STOP

命令格式:

:STOP

功能描述:

执行该命令，示波器将停止波形采样工作，如需恢复工作，请执行:**RUN**命令。

3. :AUTO

命令格式:

:AUTO

功能描述:

该命令用于自动设置仪器控制值，使输入的波形达到最佳显示效果。

4. :HARDcopy

命令格式:

:HARDcopy

功能描述:

该命令用于截取示波器当前的屏幕信息，以“HardCopyxxx.bmp”格式拷屏位图到 U 盘中（注意：不能在文件系统中使用该命令）。

ACQUIRE命令

ACQUIRE 命令用于设置示波器的采样方式。

ACQUIRE 命令包括：

- :ACQUIRE:TYPE
- :ACQUIRE:MODE
- :ACQUIRE:AVERAGES
- :ACQUIRE:SAMPLINGRATE?
- :ACQUIRE:MEMDEPTH <depth>

下面将详细介绍每条命令的格式、功能和使用方法。

1. :ACQuire:TYPE

命令格式:

:ACQuire:TYPE <type>
:ACQuire:TYPE?

功能描述:

该命令用于设置和查询示波器当前所要采用的获取方式。 <type>可取 NORMAL（普通采样）、AVERAge（平均采样）或 PEAKdetect（峰值检测）。

返回格式:

查询返回 NORMAL、AVERAGE 或 PEAKDETECT。

举例:

:ACQ:TYPE AVER 设置平均采样方式
:ACQ:TYPE? 查询返回 AVERAGE

2. :ACQuire:MODE

命令格式:

:ACQuire:MODE <mode>
:ACQuire:MODE?

功能描述:

该命令用于设置和查询示波器当前所要采用的采样方式。 <mode>可取 RTIME（实时采样）或 ETIME（等效采样）。

返回格式:

查询返回 REAL_TIME 或 EQUAL_TIME。

举例:

:ACQ:MODE ETIM 设置等效采样方式
:ACQ:MODE? 查询返回 EQUAL_TIME

3. :ACQuire:AVERAges

命令格式:

```
:ACQuire:AVERages <count>  
:ACQuire:AVERages?
```

功能描述:

该命令用于设置和查询平均采样方式中的平均次数。<count>可取 2~256 之间 2 的 N 次幂整数。

返回格式:

查询返回 2、4、8、16、32、64、128 或 256。

举例:

```
:ACQ:AVER 16      设置平均采样次数为 16  
:ACQ:AVER?       查询返回 16
```

4. :ACQuire:SAMPlingrate?**命令格式:**

```
:ACQuire:SAMPlingrate? {CHANnel<n>|DIGITAL}
```

功能描述:

该命令用于查询模拟通道或数字通道（仅 DS1000D 系列）的采样率。<n>可取 1 或 2。

返回格式:

查询返回采样率的设置值。

举例:

```
:ACQ:SAMP? CHANnel2      查询 CH2 的采样率  
100000000.000000        查询返回 100M
```

5. :ACQuire:MEMDepth <depth>**命令格式:**

```
:ACQuire:MEMDepth <depth>  
:ACQuire:MEMDepth?
```

功能描述:

该命令用于设置和查询示波器的存储深度。<depth>可取 LONG（长存储）或 NORMAL（普通存储）。

返回格式：

查询返回 LONG 或 NORMAL。

举例：

| | |
|----------------|------------|
| :ACQ:MEMD LONG | 设置存储深度为长存储 |
| :ACQ:MEMD? | 查询返回 LONG |

DISPlay命令

DISPlay 命令用于设置示波器的显示方式。

DISPlay 命令包括：

- :DISPlay:TYPE
- :DISPlay:GRID
- :DISPlay:PERSist
- :DISPlay:MNUDisplay
- :DISPlay:MNUStatus
- :DISPlay:CLEar
- :DISPlay:BRIGhtness
- :DISPlay:INTensity

下面将详细介绍每条命令的格式、功能和使用方法。

1. :DISPlay:TYPE

命令格式:

```
:DISPlay:TYPE <type>  
:DISPlay:TYPE?
```

功能描述:

该命令用于设置和查询采样点之间的显示方式。<type>可取 VECTors（矢量显示）或 DOTS（点显示）。

返回格式:

查询返回 VECTORS 或 DOTS。

举例:

```
:DISP:TYPE VECT          设置显示类型为矢量显示  
:DISP:TYPE?             查询返回 VECTORS
```

2. :DISPlay:GRID

命令格式:

```
:DISPlay:GRID <grid>  
:DISPlay:GRID?
```

功能描述:

该命令用于设置和查询屏幕网格的状态。<grid>可取 FULL（打开背景网格及坐标）、HALF（关闭背景网格）或 NONE（关闭背景网格及坐标）。

返回格式:

查询返回 FULL、HALF 或 NONE。

举例:

```
:DISP:GRID FULL          打开背景网格及坐标  
:DISP:GRID?             查询返回 FULL
```

3. :DISPlay:PERSist

命令格式:

:DISPlay:PERsist {ON|OFF}
:DISPlay:PERsist?

功能描述:

该命令用于设置和查询波形保持功能的开关状态。“ON”表示记录点一直保持，直至波形保持功能关闭，“OFF”表示记录点以高刷新率变化。

返回格式:

查询返回 ON 或 OFF。

举例:

:DISP:PERs ON 打开波形保持
:DISP:PERs? 查询返回 ON

4. :DISPlay:MNUDisplay**命令格式:**

:DISPlay:MNUDisplay <time>
:DISPlay:MNUDisplay?

功能描述:

该命令用于设置和查询系统自动隐藏菜单的时间。<time>可取 1s、2s、5s、10s、20s 或 INFinite（无限）。

返回格式:

查询返回 1s、2s、5s、10s、20s 或 INFinite。

举例:

:DISP:MNUD 10 设置菜单显示时间为 10s
:DISP:MNUD? 查询返回 10s

5. :DISPlay:MNUStatus**命令格式:**

:DISPlay:MNUStatus {ON|OFF}
:DISPlay:MNUStatus?

功能描述:

该命令用于设置和查询操作菜单的开关状态。

返回格式:

查询返回 ON 或 OFF。

举例:

```
:DISP:MNUS ON      打开操作菜单
:DISP:MNUS?        查询返回 ON
```

6. :DISPlay:CLEAr**命令格式:**

```
:DISPlay:CLEAr
```

功能描述:

该命令用于清除开启波形保持时屏幕上保持的过期波形。

7. :DISPlay:BRIGhtness**命令格式:**

```
:DISPlay:BRIGhtness <ncount>
:DISPlay:BRIGhtness?
```

功能描述:

该命令用于设置和查询网格亮度。从暗到亮的，<ncount>的取值范围为：0~32。

返回格式:

查询返回 ncount 的设置值。

举例:

```
:DISP:BRIG 10      设置网格亮度为 10
:DISP:BRIG?        查询返回 10
```

8. :DISPlay:INTensity

命令格式:

:DISPlay:INTensity <count>

:DISPlay:INTensity?

功能描述:

该命令用于设置和查询波形亮度。<count>可取 0~32 之间的整数。

返回格式:

查询返回 count 的设置值。

举例:

:DISP:INT 12 设置波形亮度为 12

:DISP:INT? 查询返回 12

TIMebase命令

TIMebase 命令用于改变水平刻度（时基）和触发在内存中的水平位置（触发位移）。改变水平刻度会使波形相对屏幕中心扩张或收缩，改变水平位置则使波形相对于触发点的位置有偏移。

TIMebase 命令包括：

- :TIMebase:MODE
- :TIMebase[:DELAyed]:OFFSet
- :TIMebase[:DELAyed]:SCALe
- :TIMebase:FORMat

下面将详细介绍每条命令的格式、功能和使用方法。

1. :TIMebase:MODE

命令格式:

```
:TIMebase:MODE <mode>  
:TIMebase:MODE?
```

功能描述:

该命令用于设置和查询水平时基的扫描方式。<mode>可取 MAIN（主时基）或 DELayed（延迟扫描时基）。

返回格式:

查询返回 MAIN 或 DELAYED。

举例:

```
:TIM:MODE MAIN          设置水平时基为主时基  
:TIM:MODE?              查询返回 MAIN
```

2. :TIMebase[:DELaYed]:OFFSet

命令格式:

```
:TIMebase[:DELaYed]:OFFSet <offset>  
:TIMebase[:DELaYed]:OFFSet?
```

功能描述:

该命令用于调整 MAIN（主时基）或 DELayed（延迟扫描）的时基偏移量（即波形位置相对于触发中点的偏移量）。其中:

当 NORMAL 模式时, <offset>的取值范围为: 1s ~ 内存终点;

当 STOP 模式时, <offset>的取值范围为: -500s ~ +500s;

当 SCAN 模式时, <offset>的取值范围为: $-6 * \text{Scale} \sim +6 * \text{Scale}$, Scale 表示当前的水平档位, 单位为 s/div。

选择 MAIN 状态时, 需省略[:DELaYed]选项。

返回格式:

查询返回 offset 的设置值, 单位为 s。

举例:

```
:TIM:MODE MAIN          设置主时基  
:TIM:OFFS 1             设置时基偏移量为 1s
```

:TIM:OFFS? 查询返回 1.000e+00

3. :TIMebase[:DELAyed]:SCALE

命令格式:

:TIMebase[:DELAyed]:SCALE <scale_val>
:TIMebase[:DELAyed]:SCALE?

功能描述:

该命令用于设置和查询 MAIN（主时基）或 DELayed（延迟扫描时基）状态下的水平刻度，即 s/div（秒/格）。其中：

当 YT 模式时，<scale_val>的取值范围为：2ns - 50s；

当 ROLL 模式时，<scale_val>的取值范围为：500ms - 50s；

选择 MAIN 状态时，需省略[:DELAyed]选项。

返回格式:

查询返回水平时基的设置值，单位为 s。

举例:

:TIM:MODE MAIN 设置主时基
:TIM:SCALE 2 设置时基档位为 2s
:TIM:SCALE? 查询返回 2.000e+00

4. :TIMebase:FORMat

命令格式:

:TIMebase:FORMat <value>
:TIMebase:FORMat?

功能描述:

该命令用于设置和查询水平时基。<value>可取 XY、YT 或 SCANNing。

返回格式:

查询返回 X-Y、Y-T 或 SCANNING。

举例:

:TIM:FORM YT 设置时基格式为 Y-T 模式

:TIM:FORM?

查询返回 Y-T

TRIGger命令

为了得到稳定的波形显示，需要对示波器的触发系统进行设置。触发决定了示波器何时开始采集数据和显示波形。一旦触发被正确设定，它可以将不稳定的显示转换成有意义的波形。

示波器在开始采集数据时，先收集足够的数据用来在触发点的左方画出波形，示波器在等待触发条件发生的同时连续的采集数据，当检测到触发后，示波器将连续的采集足够的数据以在触发点的右方画出波形。

DS1000E, DS1000D 系列数字示波器的触发方式有：Edge（边沿）、Pulse（脉宽）、Video（视频）、Slope（斜率）、Pattern（码型）、Duration（持续时间）和 Alternation（交替）触发。

TRIGger 命令包括：

触发控制命令

- :TRIGger:MODE
- :TRIGger<mode>:SOURce
- :TRIGger<mode>:LEVel
- :TRIGger<mode>:SWEep
- :TRIGger<mode>:COUPling
- :TRIGger:HOLDoff
- :TRIGger:STATus?
- :Trig%50
- :FORCetrig

边沿触发命令

- :TRIGger:EDGE:SLOPe
- :TRIGger:EDGE:SENSitivity

脉宽触发命令

- :TRIGger:PULSe:MODE
- :TRIGger:PULSe:SENSitivity
- :TRIGger:PULSe:WIDTh

视频触发命令

- :TRIGger:VIDEO:MODE
- :TRIGger:VIDEO:POLarity
- :TRIGger:VIDEO:STANdard
- :TRIGger:VIDEO:LINE
- :TRIGger:VIDEO:SENSitivity

斜率触发命令

- :TRIGger:SLOPe:TIME
- :TRIGger:SLOPe:SENSitivity
- :TRIGger:SLOPe:MODE
- :TRIGger:SLOPe:WINDow
- :TRIGger:SLOPe:LEVelA
- :TRIGger:SLOPe:LEVelB

码型触发命令

- :TRIGger:PATtern:PATtern

持续时间触发命令

- :TRIGger:DURation:PATtern
- :TRIGger:DURation:TIME
- :TRIGger:DURation:QUALifier

交替触发命令

- :TRIGger:ALternation:SOURce
- :TRIGger:ALternation:TYPE
- :TRIGger:ALternation:TimeSCALe
- :TRIGger:ALternation:TimeOFFSet
- :TRIGger:ALternation<mode>:LEVel
- :TRIGger:ALternation:EDGE:SLOPe
- :TRIGger:ALternation<mode>:MODE
- :TRIGger:ALternation<mode>:TIME
- :TRIGger:ALternation:VIDEO:POLarity
- :TRIGger:ALternation:VIDEO:STANdard
- :TRIGger:ALternation:VIDEO:LINE
- :TRIGger:ALternation:SLOPe:WINDow
- :TRIGger:ALternation:SLOPe:LEVelA
- :TRIGger:ALternation:SLOPe:LEVelB
- :TRIGger:ALternation<mode>:COUPling

- :TRIGger:ALTerNation<mode>:HOLDoff
- :TRIGger:ALTerNation<mode>:SENSitivity

下面将详细介绍每条命令的格式、功能和使用方法。

触发控制

1. :TRIGger:MODE

命令格式:

```
:TRIGger:MODE <mode>
```

```
:TRIGger:MODE?
```

功能描述:

该命令用于设置和查询触发模式。<mode>可取 EDGE、PULSe、VIDEO、SLOPe、PATTern、DURation 或 ALTernation。

返回格式:

查询返回 EDGE、PULSE、VIDEO、SLOPE、PATTERN、DURATION 或 ALTERNATION。

举例:

```
:TRIG:MODE EDGE          设置边沿触发模式
```

```
:TRIG:MODE?              查询返回 EDGE
```

2. :TRIGger<mode>:SOURce

命令格式:

```
:TRIGger<mode>:SOURce <src>
```

```
:TRIGger<mode>:SOURce?
```

功能描述:

该命令用于设置和查询触发信源。<src>可取输入通道 (CHANnel1、CHANnel2)、外部触发通道 (EXT、AC Line (市电) 或数字通道 (Digital))。

当 mode 为:EDGE 时, <src>可取 CHANnel<n>、EXT、ACLIne 或 DIGItal<m>;

当 mode 为:PULSe 时, <src>可取 CHANnel<n>、EXT 或 DIGItal<m>;

当 mode 为:SLOPe 时, <src>可取 CHANnel<n>或 EXT;

当 mode 为:VIDEO 时, <src>可取 CHANnel<n>或 EXT;

其中, <n>可取 1 或 2; <m>的取值范围为 0~15。

返回格式:

查询返回 CH1、CH2、EXT、ACLIne、D0~D15。

举例:

:TRIG:EDGE:SOUR CHAN1 设置边沿触发信源为通道 1
:TRIG:EDGE:SOUR? 查询返回 CH1

3. :TRIGger<mode>:LEVel**命令格式:**

:TRIGger<mode>:LEVel <level>
:TRIGger<mode>:LEVel?

功能描述:

该命令用于设置和查询触发电平。<level>可取:EDGE、:PULSe 或:VIDEo, 其取值范围为: $-6 * \text{Scale} \sim +6 * \text{Scale}$, Scale 表示当前的垂直档位, 单位为 V/div。

返回格式:

查询返回 level 的设置值, 单位为 V。

举例:

:TRIG:EDGE:LEV 1 设置边沿触发的触发电平为 1
:TRIG:EDGE:LEV? 查询返回 1.00e+00

4. :TRIGger<mode>:SWEep**命令格式:**

:TRIGger<mode>:SWEep {AUTO|NORMAl|SINGLe}
:TRIGger<mode>:SWEep?

功能描述:

该命令用于设置和查询触发方式。<mode>可取:EDGE、:PULSe、:SLOPe、:PATTern 或:DURation。

返回格式:

查询返回 AUTO、NORMAL 或 SINGLE。

举例:

:TRIG:EDGE:SWE AUTO 设置自动触发方式
:TRIG:EDGE:SWE? 查询返回 AUTO

5. :TRIGger<mode>:COUPling

命令格式:

```
:TRIGger<mode>:COUPling {DC|AC|HF|LF}  
:TRIGger<mode>:COUPling?
```

功能描述:

该命令用于设置和查询耦合方式。其中:

DC (直流): 让信号的所有分量通过;

AC (交流): 阻挡“直流”成分并衰减 10Hz 以下的信号;

HF (高频抑制): 衰减超过 150kHz 的高频分量;

LF (低频抑制): 阻挡直流分量并衰减低于 8kHz 的低频成分;

<mode>可取:EDGE、:PULSe 或:SLOPe。

返回格式:

查询返回 DC、AC、HF 或 LF。

举例:

```
:TRIG:EDGE:COUP DC      设置耦合方式为直流  
:TRIG:EDGE:COUP?      查询返回 DC
```

6. :TRIGger:HOLDoff

命令格式:

```
:TRIGger:HOLDoff <count>  
:TRIGger:HOLDoff?
```

功能描述:

该命令用于设置和查询触发释抑时间。<count>的取值范围为: 500ns~1.5s。

返回格式:

查询返回 count 的设置值, 单位为 s。

举例:

```
:TRIG:HOLD 0.0005      设置触发释抑时间为 500 微秒  
:TRIG:HOLDoff?      查询返回 5.000e-04
```

7. :TRIGger:STATus?

命令格式:

:TRIGger:STATus?

功能描述:

该命令用于查询当前示波器的运行状态。这些状态有：RUN（运行）、STOP（停止）、T`D（已触发）、WAIT（等待）和 AUTO（自动）。

返回格式:

查询返回 RUN、STOP、T`D、WAIT 或 AUTO。

8. :Trig%50

命令格式:

:Trig%50

功能描述:

该命令用于设定触发电平值至信号幅值的垂直中点。

9. :FORCetrig

命令格式:

:FORCetrig

功能描述:

该命令用于强制示波器产生触发信号，主要应用于“普通”和“单次”触发模式。

边沿触发

1. :TRIGger:EDGE:SLOPe

命令格式:

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

功能描述:

该命令用于设置和查询边沿触发的边沿类型为 POSitive(上升沿)或 NEGative(下降沿)。

返回格式:

查询返回 POSITIVE 或 NEGATIVE。

举例:

```
:TRIG:EDGE:SLOP POS      设置触发边沿为上升沿  
:TRIG:EDGE:SLOP?        查询返回 POSITIVE
```

2. :TRIGger:EDGE:SENSitivity

命令格式:

```
:TRIGger:EDGE:SENSitivity <count>  
:TRIGger:EDGE:SENSitivity?
```

功能描述:

该命令用于设置和查询触发灵敏度。<count>的取值范围为; 0.1div~1div。

返回格式:

查询返回 count 的设置值, 单位为 div。

举例:

```
:TRIG:EDGE:SENS 0.5      设置触发灵敏度为 0.5div  
:TRIG:EDGE:SENS?        查询返回 5.00e-01
```

脉宽触发

1. :TRIGger:PULSe:MODE

命令格式:

:TRIGger:PULSe:MODE <mode>

:TRIGger:PULSe:MODE?

功能描述:

该命令用于设置和查询脉宽触发的脉冲条件。<mode>可取+GREATERthan（正脉宽大于）、+LESSthan（正脉宽小于）、+EQUAl（正脉宽等于）、-GREATERthan（负脉宽大于）、-LESSthan（负脉宽小于）或-EQUAl（负脉宽等于）。

返回格式:

查询返回+GREATER THAN、+LESS THAN、+EQUAL、-GREATER THAN、-LESS THAN 或-EQUAL。

举例:

:TRIG:PULS:MODE +GRE

设置脉冲条件为“正脉宽大于”

:TRIG:PULS:MODE?

查询返回+GREATER THAN

2. :TRIGger:PULSe:SENSitivity

命令格式:

:TRIGger:PULSe:SENSitivity <count>

:TRIGger:PULSe:SENSitivity?

功能描述:

该命令用于设置和查询触发灵敏度。<count>的取值范围为：0.1div~1div。

返回格式:

查询返回 count 的设置值，单位为 div。

举例:

:TRIG:PULS:SENS 0.5

设置触发灵敏度为 0.5div

:TRIG:PULS:SENS?

查询返回 5.00e-01

3. :TRIGger:PULSe:WIDTh

命令格式:

:TRIGger:PULSe:WIDTh <wid>
:TRIGger:PULSe:WIDTh?

功能描述:

该命令用于设置和查询脉冲宽度。<wid>的取值范围为：20ns ~ 10s。

返回格式:

查询返回 wid 的设置值，单位为 s。

举例:

| | |
|-----------------------|----------------|
| :TRIG:PULS:WIDT 0.001 | 设置脉冲宽度为 1ms |
| :TRIG:PULS:WIDT? | 查询返回 1.000e-03 |

视频触发

1. :TRIGger:VIDEO:MODE

命令格式:

```
:TRIGger:VIDEO:MODE <mode>  
:TRIGger:VIDEO:MODE?
```

功能描述:

该命令用于设置和查询视频触发的同步方式。<mode>可取 ODDfield（奇数场）、EVENfield（偶数场）、LINE（指定行）或 ALLlines（所有行）。

返回格式:

查询返回 ODD FIELD、EVEN FIELD、LINE 或 ALL LINES。

举例:

```
:TRIG:VIDEO:MODE EVEN      设置触发同步方式为偶数场  
:TRIG:VIDEO:MODE?         查询返回 EVEN FIELD
```

2. :TRIGger:VIDEO:POLarity

命令格式:

```
:TRIGger:VIDEO:POLarity {POSitive|NEGative}  
:TRIGger:VIDEO:POLarity?
```

功能描述:

该命令用于设置和查询视频触发的视频极性为 POSitive（正极性）或 NEGative（负极性）。

返回格式:

查询返回 POSITIVE 或 NEGATIVE。

举例:

```
:TRIG:VIDEO:POL POS      设置视频极性为正极性  
:TRIG:VIDEO:POL?       查询返回 POSITIVE
```

3. :TRIGger:VIDEO:STANdard

命令格式:

:TRIGger:VIDEO:STANdard {NTSC|PALSecam}
:TRIGger:VIDEO:STANdard?

功能描述:

该命令用于设置和查询视频触发的视频标准。

返回格式:

查询返回 NTSC、PAL/SECAM。

举例:

| | |
|-----------------------|-------------------|
| :TRIG:VIDEO:STAN PALS | 选择视频标准为 PAL/SECAM |
| :TRIG:VIDEO:STAN? | 查询返回 PAL/SECAM |

4. :TRIGger:VIDEO:LINE

命令格式:

:TRIGger:VIDEO:LINE <value>
:TRIGger:VIDEO:LINE?

功能描述:

该命令用于设置和查询同步的指定行数。NTSC 标准下，<value>的取值范围为：1~525；PAL/SECAM 标准下，<value>的取值范围为：1~625。

返回格式:

查询返回当前所指定的行数。

举例:

| | |
|---------------------|---------------|
| :TRIG:VIDEO:LINE 25 | 设置同步的指定行数为 25 |
| :TRIG:VIDEO:LINE? | 查询返回 25 |

5. :TRIGger:VIDEO:SENSitivity

命令格式:

:TRIGger:VIDEO:SENSitivity <count>

:TRIGger:VIDEO:SENSitivity?

功能描述:

该命令用于设置和查询触发灵敏度。<count>的取值范围为；0.1div ~1div。

返回格式:

查询返回 count 的设置值，单位 div。

举例:

:TRIG:VIDEO:SENS 0.5 设置触发灵敏度为 0.5div

:TRIG:VIDEO:SENS? 查询返回 5.00e-01

斜率触发

1. :TRIGger:SLOPe:TIME

命令格式:

:TRIGger:SLOPe:TIME <count>

:TRIGger:SLOPe:TIME?

功能描述:

该命令用于调整和查询斜率触发的时间设置。<count>的取值范围为：20ns~10s。

返回格式:

查询返回 count 的设置值，单位为 s。

举例:

:TRIG:SLOP:TIME 0.01 设置斜率时间为 10ms

:TRIG:SLOP:TIME? 查询返回 1.000e-02

2. :TRIGger:SLOPe:SENSitivity

命令格式:

:TRIGger:SLOPe:SENSitivity <count>

:TRIGger:SLOPe:SENSitivity?

功能描述:

该命令用于设置和查询触发灵敏度。<count>的取值范围为：0.1div ~1div。

返回格式:

查询返回 count 的设置值，单位为 div。

举例:

:TRIG:SLOP:SENS 0.5 设置触发灵敏度为 0.5div

:TRIG:SLOP:SENS? 查询返回 5.00e-01

3. :TRIGger:SLOPe:MODE

命令格式:

```
:TRIGger:SLOPe:MODE <mode>
:TRIGger:SLOPe:MODE?
```

功能描述:

该命令用于设置和查询斜率触发的斜率条件。<mode>可取+GREaterthan（正斜率大于）、+LESSthan（正斜率小于）、+EQUal（正斜率等于）、-GREaterthan（负斜率大于）、-LESSthan（负斜率小于）或-EQUal（负斜率等于）。

返回格式:

查询返回+GREATER THAN、+LESS THAN、+EQUAL、-GREATER THAN、-LESS THAN 或-EQUAL。

举例:

```
:TRIG:SLOP:MODE +GRE          设置斜率条件为“正斜率大于”
:TRIG:SLOPe:MODE?            查询返回+GREATER THAN
```

4. :TRIGger:SLOPe:WINDow**命令格式:**

```
:TRIGger:SLOPe:WINDow <count>
:TRIGger:SLOPe:WINDow?
```

功能描述:

该命令用于设置和查询能被 LEVEL 调节的触发电平类型。

当斜率条件为+GREaterthan、+LESSthan 或+ EQUal 时，<count>可取 PA（上升沿 Level A）、PB（上升沿 Level B）或 PAB（上升沿 Level AB）；

当斜率条件为-GREaterthan、-LESSthan 或-EQUal 时，<count>可取 NA（下降沿 Level A）、NB（下降沿 LevelB）或 NAB（下降沿 LevelIAB）。

返回格式:

查询返回 P_WIN_A、P_WIN_B、P_WIN_AB、N_WIN_A、N_WIN_B 或 N_WIN_AB。

举例:

```
:TRIG:SLOP:WIND PA          设置触发电平类型为上升沿 Level A
:TRIG:SLOP:WIND?          查询返回 P_WIN_A
```

5. :TRIGger:SLOPe:LEVelA

命令格式:

:TRIGger:SLOPe:LEVelA <value>
:TRIGger:SLOPe:LEVelA?

功能描述:

该命令用于设置和查询触发电平的上边界 Level A。<value>的取值范围为：
LevelB~+ 6*Scale，Scale 表示当前的垂直档位，单位为 V/div。

返回格式:

查询返回 value 的设置值，单位为 V。

举例:

:TRIG:SLOP:LEVA 2 设置触发电平上边界为 2V
:TRIG:SLOP:LEVA? 查询返回 2.000e+00

6. :TRIGger:SLOPe:LEVelB

命令格式:

:TRIGger:SLOPe:LEVelB <value>
:TRIGger:SLOPe:LEVelB?

功能描述:

该命令用于设置和查询触发电平的下边界 LEVel B。<value>的取值范围为：
-6*Scale~LevelA，Scale 表示当前的垂直档位，单位为 V/div。

返回格式:

查询返回 value 的设置值，单位为 V。

注：Level A（上边界）的最小值不能小于 Level B（下边界）的最大值。

举例:

:TRIG:SLOP:LEVB -1.5 设置触发电平下边界为-1.5V
:TRIG:SLOP:LEVB? 查询返回-1.500e+00

码型触发

1. :TRIGger:PATtern:PATtern

命令格式:

```
:TRIGger:PATtern:PATtern <value>,<mask>, [<edge source>,<edge>]  
:TRIGger:PATtern:PATtern?
```

功能描述:

该命令用于设置和查询数字信号的码型。其中:

<value>取值为 16 位无符号整型数据, High 为 1, Low 为 0;

<mask>取值为 16 位无符号整型数据, enable 为 1, X 为 0;

<edge source>的取值范围为: 0~15 (DIG0~DIG15);

<edge>在上升沿处为 1, 下降沿处为 0。

返回格式:

查询依次返回 value、mask、edge source 和 edge 的设定值, 其中 value、mask 返回的是十进制数据。

举例:

```
:TRIG:PATT:PATT 65535,65535,2,1    设置码型  
:TRIG:PATT:PATT?                  查询返回 65535, 65535, DIG2, Positive
```

持续时间触发

1. :TRIGger:DURation:PATtern

命令格式:

```
:TRIGger:DURation:PATtern <value>,<mask>  
:TRIGger:DURation:PATtern?
```

功能描述:

该命令用于设置和查询数字信号的码型。其中:
<value>取值为 16 位无符号整型数据, High 为 1, Low 为 0;
<mask>取值为 16 位无符号整型数据, enable 为 1, X 为 0。

返回格式:

查询依次返回 value 和 mask 的设置值。

举例:

```
:TRIG:DUR:PATT 65535,65535      设置码型  
:TRIG:DUR:PATT?                查询返回 65535,65535
```

2. :TRIGger:DURation:TIME

命令格式:

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

功能描述:

该命令用于设置和查询持续触发的限定时间。<time>的取值范围为: 2ns~10s。

返回格式:

查询返回 time 的设置值。

举例:

```
:TRIG:DUR:TIME 0.05             设置持续时间限定符时间为 50ms  
:TRIG:DUR:TIME?                查询返回 5.00e-02
```

3. :TRIGger:DURation:QUALifier

命令格式:

:TRIGger:DURation:QUALifier <qual>
:TRIGger:DURation:QUALifier?

功能描述:

该命令用于设置和查询时间限定条件。<qual>可取 GREATERthan（大于）、LESSthan（小于）或 EQUAL（等于）。

返回格式:

查询返回 GREATER THAN、LESS THAN 或 EQUAL。

举例:

| | |
|--------------------|-------------------|
| :TRIG:DUR:QUAL GRE | 设置时间限定条件为“大于” |
| :TRIG:DUR:QUAL? | 查询返回 GREATER THAN |

交替触发

1. :TRIGger:ALTerNation:SOURce

命令格式:

```
:TRIGger:ALTerNation:SOURce <src>  
:TRIGger:ALTerNation:SOURce?
```

功能描述:

该命令用于设置和查询将要进行操作的通道。<src>可取 CHANnel1 或 CHANnel2。

返回格式:

查询返回 CH1 或 CH2。

举例:

```
:TRIG:ALT:SOUR CHAN2      选择通道 2 进行操作  
:TRIG:ALT:SOUR?           查询返回 CH2
```

2. :TRIGger:ALTerNation:TYPE

命令格式:

```
:TRIGger:ALTerNation:TYPE <value>  
:TRIGger:ALTerNation:TYPE?
```

功能描述:

该命令用于设置和查询交替触发模式下的触发类型。<value>可取 EDGE（边沿）、PULSe（脉宽）、SLOPe（斜率）或 VIDEO（视频）触发。

返回格式:

查询返回 EDGE、PULSE、SLOPE 或 VIDEO。

举例:

```
:TRIG:ALT:TYPE EDGE      设置触发类型为边沿触发  
:TRIG:ALT:TYPE?         查询返回 EDGE
```

3. :TRIGger:ALTerNation:TimeSCALe

命令格式:

:TRIGger:ALTernation:TimeSCALe <value>
:TRIGger:ALTernation:TimeSCALe?

功能描述:

该命令用于设置和查询当前通道的时基。<value>的取值范围为：2ns~20ms。

返回格式:

查询返回 value 的设置值，单位为 s。

举例:

| | |
|-----------------------|----------------|
| :TRIG:ALT:TSCAL 0.001 | 设置时基为 1ms |
| :TRIG:ALT:TSCAL? | 查询返回 1.000e-03 |

4. :TRIGger:ALTernation:TimeOFFSet**命令格式:**

:TRIGger:ALTernation:TimeOFFSet <value>
:TRIGger:ALTernation:TimeOFFSet?

功能描述:

该命令用于设置和查询当前通道的水平时基偏移。其中，
当 NORMAL 模式时，<value>的取值范围为：1s ~ 内存终点；
当 STOP 模式时，<value>的取值范围为：-500s ~ +500s；
当 ROLL 模式时，<value>的取值范围为：-6*scale ~ +6*Scale。
Scale 表示当前的水平档位，单位为 s/div。

返回格式:

查询返回 value 的设置值，单位为 s。

举例:

| | |
|------------------------|--------------------|
| :TRIG:ALT:TOFFS 0.0002 | 设置当前通道的水平时基为 200us |
| :TRIG:ALT:TOFFS? | 查询返回 2.000e-04 |

5. :TRIGger:ALTernation<mode>:LEVel**命令格式:**

```
:TRIGger:ALTernation<mode>:LEVel <value>
:TRIGger:ALTernation<mode>:LEVel?
```

功能描述:

该命令用于设置和查询当前通道的触发电平。<mode>可取:EDGE、:PULSe或:VIDEO。<value>的取值范围为: $-6*Scale \sim +6*Scale$, Scale 表示当前的垂直档位, 单位为 V/div。

返回格式:

查询返回 value 的设置值, 单位为 V。

举例:

```
:TRIG:ALT:EDGE:LEV 2      设置当前通道的触发电平为 2V
:TRIG:ALT:EDGE:LEV?      查询返回 2.00e+00
```

6. :TRIGger:ALTernation:EDGE:SLOPe**命令格式:**

```
:TRIGger:ALTernation:EDGE:SLOPe <value>
:TRIGger:ALTernation:EDGE:SLOPe?
```

功能描述:

该命令用于设置和查询当前操作状态下边沿触发的边沿类型。<value>可取 POSitive (上升沿) 或 NEGative (下降沿)。

返回格式:

查询返回 POSITIVE 或 NEGATIVE。

举例:

```
:TRIG:ALT:EDGE:SLOP POS    设置边沿类型
:TRIG:ALT:EDGE:SLOP?      查询返回 POSITIVE
```

7. :TRIGger:ALTernation<mode>:MODE**命令格式:**

```
:TRIGger:ALTernation<mode>:MODE <value>
:TRIGger:ALTernation<mode>:MODE?
```

功能描述:

该命令用于设置或查询当前操作状态下脉宽触发的脉冲条件、斜率触发的斜率条件或视频触发的同步设置。<mode>可取:PULSe、:SLOPe 或:VIDEO。

当<mode>为:PULSe 或:SLOPe 时, <value>可取+GREaterthan、+LESSthan、+EQUal、-GREaterthan、-LESSthan 或-EQUal;

当<mode>为:VIDEO 时, <value>可取 ODDfield、EVENfield、LINE 或 ALLlines。

返回格式:

查询返回 value 的设置值。

举例:

```
:TRIG:ALT:PULS:MODE +GRE          设置脉冲条件
:TRIG:ALT:PULS:MODE?              查询返回+GREATER THAN
```

8. :TRIGger:ALTernation<mode>:TIME**命令格式:**

```
:TRIGger:ALTernation<mode>:TIME <value>
```

```
:TRIGger:ALTernation<mode>:TIME?
```

功能描述:

该命令用于设置和查询当前操作状态下脉宽触发的脉冲宽度或斜率触发的斜率时间。<mode>可取:SLOPe 或:PULSe, <value>的取值范围为: 20ns~10s。

返回格式:

查询返回 value 的设置值, 单位为 s。

举例:

```
:TRIG:ALT:SLOP:TIME 0.002          设置斜率时间为 2ms
:TRIG:ALT:SLOP:TIME?              查询返回 2.000e-03
```

9. :TRIGger:ALTernation:VIDEO:POLarity**命令格式:**

```
:TRIGger:ALTernation:VIDEO:POLarity {POSitive|NEGative}
```

```
:TRIGger:ALTernation:VIDEO:POLarity?
```

功能描述:

该命令用于设置和查询当前操作状态下视频触发的视频极性。

返回格式:

查询返回 POSITIVE 或 NEGATIVE。

举例:

```
:TRIG:ALT:VIDEO:POL POS      设置视频极性为正极性
:TRIG:ALT:VIDEO:POL?        查询返回 POSITIVE
```

10. :TRIGger:ALTerNation:VIDEO:STANdard**命令格式:**

```
:TRIGger:ALTerNation:VIDEO:STANdard {NTSC|PALSecam}
:TRIGger:ALTerNation:VIDEO:STANdard?
```

功能描述:

该命令用于设置和查询当前操作状态下视频触发的视频标准。

返回格式:

查询返回 NTSC 或 PAL/SECAM。

举例:

```
:TRIG:ALT:VIDEO:STAN NTSC      设置视频标准为 NTSC
:TRIG:ALT:VIDEO:STAN?        查询返回 NTSC
```

11. :TRIGger:ALTerNation:VIDEO:LINE**命令格式:**

```
:TRIGger:ALTerNation:VIDEO:LINE <value>
:TRIGger:ALTerNation:VIDEO:LINE?
```

功能描述:

该命令用于设置和查询当前操作状态下同步的指定行数。NTSC 标准下, <value> 的取值范围为: 1~525; PAL/SECAM 标准下, <value> 的取值范围为: 1~625。

返回格式:

查询返回 value 的设置值。

举例：


```
:TRIG:ALT:VIDEO:LINE 100      设置同步的指定行数为 100
:TRIG:ALT:VIDEO:LINE?        查询返回 100
```

12. :TRIGger:ALTerNation:SLOPe:WINDow

命令格式：

```
:TRIGger:ALTerNation:SLOPe:WINDow <count>
:TRIGger:ALTerNation:SLOPe:WINDow?
```

功能描述：

该命令用于设置和查询当前操作状态下能被  LEVEL 调节的触发电平类型。当斜率条件为 +GREaterthan、+LESSthan 或 + EQUal 时，<count>可取 PA（上升沿 Level A）、PB（上升沿 Level B）或 PAB（上升沿 Level AB）；当斜率条件为 -GREaterthan、-LESSthan 或 -EQUal 时，<count>可取 NA（下降沿 Level A）、NB（下降沿 Level B）或 NAB（下降沿 Level AB）。

返回格式：

查询返回 P_WIN_A、P_WIN_B、P_WIN_AB、N_WIN_A、N_WIN_B 或 N_WIN_AB。

举例：

```
:TRIG:ALT:SLOP:WIND PA      设置触发电平类型为上升沿 Level A
:TRIG:ALT:SLOP:WIND?      查询返回 P_WIN_A
```

13. :TRIGger:ALTerNation:SLOPe:LEVelA

命令格式：

```
:TRIGger:ALTerNation:SLOPe:LEVelA <value>
:TRIGger:ALTerNation:SLOPe:LEVelA?
```

功能描述：

该命令用于设置和查询当前操作状态下斜率触发的触发电平上边界 Level A。<value>的取值范围为：LevelB ~ +6*Scale，Scale 表示当前的垂直档位，单位为 V/div。

返回格式:

查询返回 value 的设置值，单位为 V。

举例:

:TRIG:ALT:SLOP:LEVA 2 设置触发电平上边界 Level A 为 2V
:TRIG:ALT:SLOP:LEVA? 查询返回 2.000e+00

14. :TRIGger:ALTerNation:SLOPe:LEVeIB**命令格式:**

:TRIGger:ALTerNation:SLOPe:LEVeIB <value>
:TRIGger:ALTerNation:SLOPe:LEVeIB?

功能描述:

该命令用于设置和查询当前操作状态下斜率触发的触发电平下边界 Level B。
<value>的取值范围为: $-6 * \text{Scale} \sim \text{LevelA}$, Scale 表示当前的垂直档位，单位为 V/div。

注: Level A (上边界) 的最小值不能小于 Level B (下边界) 的最大值。

返回格式:

查询返回 value 的设置值，单位为 V。

举例:

:TRIG:ALT:SLOP:LEVB -1.5 设置触发电平下边界 Level B 为-1.5V
:TRIG:ALT:SLOP:LEVB ? 查询返回-1.500e+00

15. :TRIGger:ALTerNation<mode>:COUPLing**命令格式:**

:TRIGger:ALTerNation<mode>:COUPLing {DC|AC|HF|LF}
:TRIGger:ALTerNation<mode>:COUPLing?

功能描述:

该命令用于设置和查询耦合方式。其中:

DC (直流): 让信号的所有分量通过;

AC (交流): 阻挡“直流”成分并衰减 10Hz 以下的信号;

HF (高频抑制): 衰减超过 150kHz 的高频分量;

LF（低频抑制）：阻挡直流分量并衰减低于 8kHz 的低频成分；
<mode>可取:EDGE、:PULSe 或:SLOPe。

返回格式：

查询返回 DC 、 AC、 HF 或 LF。

举例：

:TRIG:ALT:EDGE:COUP DC 设置耦合方式为直流
:TRIG:ALT:EDGE:COUP? 查询返回 DC

16. :TRIGger:ALTerNation<mode>:HOLDOff**命令格式：**

:TRIGger:ALTerNation<mode>:HOLDOff <count>
:TRIGger:ALTerNation<mode>:HOLDOff ?

功能描述：

该命令用于设置和查询触发释抑。 <mode> 可取 :EDGE、 :PULSe、 :SLOPe 或:VIDEO。 <count>的取值范围为： 500ns~1.5s。

返回格式：

查询返回 count 的设置值，单位为 s。

举例：

:TRIG:ALT:PULS:HOLD 1 设置触发释抑为 1s
:TRIG:ALT:PULS:HOLD? 查询返回 1.000e+00

17. :TRIGger:ALTerNation<mode>:SENSitivity**命令格式：**

:TRIGger:ALTerNation<mode>:SENSitivity <count>
:TRIGger:ALTerNation<mode>:SLOPe:SENSitivity?

功能描述：

该命令用于设置和查询触发灵敏度。 <mode> 可取:EDGE、 :PULSe、 :SLOPe 或:VIDEO。 <count>的取值范围为： 0.1div ~1div。

返回格式:

查询返回 count 的设置值，单位为 div。

举例:

:TRIG:ALT:EDGE:SENS 0.5 设置触发灵敏度为 0.5div

:TRIG:ALT:EDGE:SLOP:SENS? 查询返回 5.00e-01

STORage命令

STORage 命令用于调出出厂设置。

STORage 命令包括：

- :STORage:FACTory:LOAD

下面将详细介绍每条命令的格式、功能和使用方法。

1. :STORage:FACTory:LOAD

命令格式:

:STORage:FACTory:LOAD

功能描述:

执行该命令，调用系统的出厂设置。

MATH命令

MATH（数学运算）命令用于显示 CH1、CH2 波形相加、相减、相乘以及 FFT 运算的结果。数学运算的结果同样可以通过栅格或游标进行测量。

MATH 命令包括：

- :MATH:DISPlay
- :MATH:OPERate
- :FFT:DISPlay

下面将详细介绍每条命令的格式、功能和使用方法。

1. :MATH:DISPlay

命令格式:

:MATH:DISPlay {ON|OFF}
:MATH:DISPlay?

功能描述:

该命令用于设置和查询数学运算功能的开关状态。

返回格式:

查询返回 ON 或 OFF。

举例:

:MATH:DISP ON 打开数学运算功能
:MATH:DISP? 查询返回 ON

2. :MATH:OPERate

命令格式:

:MATH:OPERate <operate>
:MATH:OPERate?

功能描述:

该命令用于设置和查询当前通道进行的数学运算类型。<operate>可取 A+B、A-B、AB 或 FFT。

返回格式:

查询返回 A+B、A-B、A*B 或 FFT。

举例:

:MATH:OPER A-B 设置运算操作为 A-B
:MATH:OPER? 查询返回 A-B

3. :FFT:DISPlay

命令格式:

:FFT:DISPlay {ON|OFF}

:FFT:DISPlay?

功能描述:

该命令用于设置和查询 FFT 运算功能的开关状态。

返回格式:

查询返回 ON 或 OFF。

举例:

:FFT:DISP ON 打开 FFT 运算功能

:FFT:DISP? 查询返回 ON

CHANnel命令

CHANnel 命令用于对仪器的两个通道进行控制。每个通道都拥有独立的垂直菜单。每个项目都按不同的通道单独设置。

CHANnel 命令包括：

- :CHANnel<n>:BWLimit
- :CHANnel<n>:COUPling
- :CHANnel<n>:DISPlay
- :CHANnel<n>:INVert
- :CHANnel<n>:OFFSet
- :CHANnel<n>:PROBe
- :CHANnel<n>:SCALe
- :CHANnel<n>:FILTer
- :CHANnel<n>:MEMoryDepth?
- :CHANnel<n>:VERNier

下面将详细介绍每条命令的格式、功能和使用方法。

1. :CHANnel<n>:BWLimit

命令格式:

```
:CHANnel<n>:BWLimit {ON|OFF}  
:CHANnel<n>:BWLimit?
```

功能描述:

该命令用于设置和查询带宽限制功能的开关状态。<n>可取 1 或 2。

返回格式:

查询返回 ON 或 OFF。

举例:

```
:CHAN2:BWL OFF      关闭通道 2 的带宽限制  
:CHAN2:BWL?        查询返回 OFF
```

2. :CHANnel<n>:COUPling

命令格式:

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

功能描述:

该命令用于设置和查询通道的耦合方式。DC（直流）表示可通过输入信号的交流和直流分量；AC（交流）表示阻挡输入信号的直流分量；GND（接地）表示断开输入信号。<n>可取 1 或 2。

返回格式:

查询返回 AC、DC 或 GND。

举例:

```
:CHAN2:COUP DC      设置通道 2 为直流耦合方式  
:CHAN2:COUP?       查询返回 DC
```

3. :CHANnel<n>:DISPlay

命令格式:

:CHANnel<n>:DISPlay {ON|OFF}
:CHANnel<n>:DISPlay?

功能描述:

该命令用于设置和查询通道的开关状态。<n>可取 1 或 2。

返回格式:

查询返回 ON 或 OFF。

举例:

:CHAN2:DISP ON 打开通道 2 显示
:CHAN2:DISP? 查询返回 ON

4. :CHANnel<n>:INVert**命令格式:**

:CHANnel<n>:INVert {ON|OFF}
:CHANnel<n>:INVert?

功能描述:

该命令用于设置和查询波形反相功能的开关状态。<n>可取 1 或 2。

返回格式:

查询返回 ON 或 OFF。

举例:

:CHAN2:INV OFF 关闭通道 2 的反相显示
:CHAN2:INV? 查询返回 OFF

5. :CHANnel<n>:OFFSet**命令格式:**

:CHANnel<n>:OFFSet <offset>
:CHANnel<n>:OFFSet?

功能描述:

该命令用于设置和查询波形在垂直方向上的位移。<n>可取 1 或 2。

当 Scale \geq 250mV, <offset>的取值范围为: -40V~+40V;

当 Scale<250mV, <offset>的取值范围为: -2V~+2V。

返回格式:

查询返回 offset 的设置值。

举例:

:CHAN2:OFFS 20 设置通道 2 的垂直位置为 20V

:CHAN2:OFFS? 查询返回 2.000e+01

6. :CHANnel<n>:PROBe

命令格式:

:CHANnel<n>:PROBe <attn>

:CHANnel<n>:PROBe?

功能描述:

该命令用于设置和查询探头衰减因数。<n>可取 1 或 2, <attn>可取 1、5、10、50、100、500 或 1000。

返回格式:

查询返回 attn 的设置值。

举例:

:CHAN2:PROB 10 设置通道 2 探头衰减比例为 10X

:CHAN2:PROB? 查询返回 1.000e+01

7. :CHANnel<n>:SCALE

命令格式:

:CHANnel<n>:SCALE <range>

:CHANnel<n>:SCALE?

功能描述:

该命令用于设置和查询示波器在垂直方向上放大波形的档位。<n>可取 1 或 2。

当 Probe 设置为 1X 时, <range>的取值范围为: 2mV ~ 10V;

当 Probe 设置为 5X 时, <range>的取值范围为: 10mV ~50V;

当 Probe 设置为 10X 时, <range>的取值范围为: 20mV ~ 100V;
当 Probe 设置为 50X 时, <range>的取值范围为: 100mV ~ 500V;
当 Probe 设置为 100X 时, <range>的取值范围为: 200mV ~ 1000V;
当 Probe 设置为 500X 时, <range>的取值范围为: 1V ~ 5000V;
当 Probe 设置为 1000X 时, <range>的取值范围: 2V~ 10000V。

返回格式:

查询返回 range 的设置值, 单位为 V。

举例:

```
:CHAN2:PROB 10      设置探头衰减比为 10X  
:CHAN2:SCAL 20     设置垂直档位为 20V  
:CHAN2:SCAL?       查询返回 2.000e+01
```

8. :CHANnel<n>:FILTer**命令格式:**

```
:CHANnel<n>:FILTer {ON|OFF}  
:CHANnel<n>:FILTer?
```

功能描述:

该命令用于设置和查询数字滤波功能的开关状态。<n>可取 1 或 2。

返回格式:

查询返回 ON 或 OFF。

举例:

```
:CHAN2:FILT OFF    关闭通道 2 的数字滤波功能  
:CHAN2:FILT?      查询返回 OFF
```

9. :CHANnel<n>:MEMoryDepth?**命令格式:**

```
:CHANnel<n>:MEMoryDepth?
```

功能描述:

该命令用于查询通道的存储深度。<n>可取 1 或 2。

长存储时，单通道可存储 1Mpts，双通道可存储 512kpts。
普通存储时，单通道可存储 16kpts，双通道可存储 8kpts。

返回格式：

如：8192 (8kpts)。

10. :CHANnel<n>:VERNier

命令格式：

:CHANnel<n>:VERNier {ON|OFF}

:CHANnel<n>:VERNier?

功能描述：

该命令用于设置和查询档位调节方式。ON 表示微调 (Fine)，OFF 表示粗调 (Coarse)。<n>可取 1 或 2。

返回格式：

查询返回 Coarse 或 Fine。

举例：

:CHAN2:VERN ON 打开微调功能

:CHAN2:VERN? 查询返回 Fine

MEASure命令

MEASure 命令用于示波器最基本的测量操作，这些命令只针对模拟通道的测量，通常以科学计数方式返回所测量的结果。

MEASure 命令包括：

- :MEASure:CLEar
- :MEASure:VPP?
- :MEASure:VMAX?
- :MEASure:VMIN?
- :MEASure:VAMPLitude?
- :MEASure:VTOP?
- :MEASure:VBASe?
- :MEASure:VAverage?
- :MEASure:VRMS?
- :MEASure:OVERshoot?
- :MEASure:PREShoot?
- :MEASure:FREQuency?
- :MEASure:RISetime?
- :MEASure:FALLtime?
- :MEASure:PERiod?
- :MEASure:PWIDth?
- :MEASure:NWIDth?
- :MEASure:PDUTYcycle?
- :MEASure:NDUTYcycle?
- :MEASure:PDELay?
- :MEASure:NDELay?
- :MEASure:TOTal
- :MEASure:SOURce
- :COUNter:VALue?

下面将详细介绍每条命令的格式、功能和使用方法。

1. :MEASure:CLEar

命令格式:

:MEASure:CLEar

功能描述:

该命令用于清除当前测量的参数值。

2. :MEASure:VPP?

命令格式:

:MEASure:VPP? [<source>]

功能描述:

该命令用于查询所测波形的峰峰值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 5.12e+03, 单位为 V。

3. :MEASure:VMAX?

命令格式:

:MEASure:VMAX? [<source>]

功能描述:

该命令用于查询所测波形的最大值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 2.60e+03, 单位为 V。

4. :MEASure:VMIN?

命令格式:

:MEASure:VMIN? [<source>]

功能描述:

该命令用于查询所测波形的最小值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: -2.52e+03, 单位为 V。

5. :MEASure:VAMPlitude?**命令格式:**

:MEASure:VAMPlitude? [<source>]

功能描述:

该命令用于查询所测波形的幅值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 5.07e+03, 单位为 V。

6. :MEASure:VTOP?**命令格式:**

:MEASure:VTOP? [<source>]

功能描述:

该命令用于查询所测波形的顶端值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 2.55e+03, 单位为 V。

7. :MEASure:VBASe?**命令格式:**

:MEASure:VBASe? [<source>]

功能描述:

该命令用于查询所测波形的底端值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: -2.52e+03, 单位为 V。

8. :MEASure:VAverage?

命令格式:

:MEASure:VAverage? [<source>]

功能描述:

该命令用于查询所测波形的平均值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 4.01e+01, 单位为 V。

9. :MEASure:VRMS?

命令格式:

:MEASure:VRMS? [<source>]

功能描述:

该命令用于查询所测波形的均方根值。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 1.78e+03, 单位为 V。

10. :MEASure:OVERshoot?

命令格式:

:MEASure:OVERshoot? [<source>]

功能描述:

该命令用于查询所测波形的过冲。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 1.74e-02。

11. :MEASure:PREShoot?

命令格式:

:MEASure:PREShoot? [<source>]

功能描述:

该命令用于查询所测波形的预冲。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 0.00e+00。

12. :MEASure:FREQuency?**命令格式:**

:MEASure:FREQuency? [<source>]

功能描述:

该命令用于查询所测波形的频率。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 1.00e+03, 单位为 Hz。

13. :MEASure:RISetime?**命令格式:**

:MEASure:RISetime? [<source>]

功能描述:

该命令用于查询所测波形的上升时间。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 3.10e-04, 单位为 s。

14. :MEASure:FALLtime?**命令格式:**

:MEASure:FALLtime? [<source>]

功能描述:

该命令用于测量波形的下降时间。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如：3.10e-04，单位为 s。

15. :MEASure:PERiod?

命令格式：

:MEASure:PERiod? [<source>]

功能描述：

该命令用于查询所测波形的周期。<source>可取 CHANnel1 或 CHANnel2。

返回格式：

如：1.00e-03，单位为 s。

16. :MEASure:PWIDth?

命令格式：

:MEASure:PWIDth? [<source>]

功能描述：

该命令用于查询所测波形的正脉宽。<source>可取 CHANnel1 或 CHANnel2。

返回格式：

如：5.00e-04，单位为 s。

17. :MEASure:NWIDth?

命令格式：

:MEASure:NWIDth? [<source>]

功能描述：

该命令用于查询所测波形的负脉宽。<source>可取 CHANnel1 或 CHANnel2。

返回格式：

如：5.00e-04，单位为 s。

18. :MEASure:PDUTyCycle?

命令格式:

:MEASure:PDUTyCycle? [<source>]

功能描述:

该命令用于查询所测波形的正占空比。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 5.00e-01。

19. :MEASure:NDUTyCycle?

命令格式:

:MEASure:NDUTyCycle? [<source>]

功能描述:

该命令用于查询所测波形的负占空比。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: 5.00e-01。

20. :MEASure:PDELay?

命令格式:

:MEASure:PDELay? [<source>]

功能描述:

该命令用于查询通道 1、2 相对于上升沿的延迟。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: <-1.00e-04。

21. :MEASure:NDELay?

命令格式:

:MEASure:NDElay? [<source>]

功能描述:

该命令用于查询通道 1、2 相对于下降沿的延迟。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

如: <1.00e-04。

22. :MEASure:TOTal

命令格式:

:MEASure:TOTal {ON|OFF}

:MEASure:TOTal?

功能描述:

该命令用于设置和查询全部测量功能的开关状态。

返回格式:

查询返回 ON 或 OFF。

举例:

:MEAS:TOT ON 打开全部测量功能

:MEAS:TOT? 查询返回 ON

23. :MEASure:SOURce

命令格式:

:MEASure:SOURce <source>

:MEASure:SOURce?

功能描述:

该命令用于选择和查询当前测量的通道。<source>可取 CHANnel1 或 CHANnel2。

返回格式:

查询返回 CH1 或 CH2。

举例:

:MEAS:SOUR CHAN1 选择测量 CH1 的波形
:MEAS:SOUR? 查询返回 CH1

24. :COUNter:VALue?**命令格式:**

:COUNter:VALue?

功能描述:

查询频率计的测量结果（频率值，单位为 Hz）。

返回格式:

查询以科学计数形式返回当前测量值。

举例:

:COUNter:VALue? 查询返回 1.000004e+03。

WAVeform命令

WAVeform 命令用于读取示波器屏幕中的波形数据，返回 1024 个（1k）数据。

WAVeform 命令包括：

- :WAVeform:DATA?
- :WAVeform:POINts:MODE

下面将详细介绍每条命令的格式、功能和使用方法。

1. :WAVeform:DATA?

命令格式:

:WAVeform:DATA? [<source>]

功能描述:

该命令用于读取指定源中的波形数据。<source>可取 CHANnel1、CHANnel2、DIGital、MATH 或 FFT。

返回格式:

查询返回 1024 个数据。

2. :WAVeform:POINts:MODE

命令格式:

:WAVeform:POINts:MODE <points_mode>

:WAVeform:POINts:MODE?

功能描述:

该命令用于设置波形点模式。<points_mode>可设置为 NORMal、MAXimum 或 RAW。

:WAVeform:DATA?命令在不同模式下返回的波形点数:

| | NORmal | RAW | | MAX |
|----------|--------|------------|--------------|--|
| | | 普通存储 | 长存储 | |
| MATH | 600 | 600 | 600 | 运行状态时, 与 NORMal 相同, 停 止状态时, 与 RAW 相同。 |
| FFT | 512 | 512 | 512 | |
| CHx | 600 | 8192(8k) | 524288(512k) | |
| 半通道 CHx* | 600 | 16384(16k) | 1048576(1M) | |
| DIGITAL | 600 | 16384(16k) | 1048576(1M) | |

*注: “半通道”是指 CH1、CH2 中只选一个, 且 MATH 关闭时的状态。

返回格式:

查询返回 NORMal、RAW 或 MAXimum。

举例:

:WAV:POIN:MODE NORM

设置波形点模式为 NORMal。

:WAV:POIN:MODE?

查询返回 NORMal。

LA命令

LA 命令用于控制逻辑分析仪模块对数字信号进行的采集、分析操作。

LA 命令包括：

- :LA:DISPlay
- :DIGital<n>:TURN
- :DIGital<n>:POSition
- :LA:THReshold
- :LA:POSition:RESet
- :LA:GROUp
- :LA:GROUp<n>:SIZe

下面将详细介绍每条命令的格式、功能和使用方法。

1. :LA:DISPlay

命令格式:

:LA:DISPlay {ON|OFF}
:LA:DISPlay?

功能描述:

该命令用于设置和查询数字通道的开关状态。

返回格式:

查询返回 ON 或 OFF。

举例:

:LA:DISP ON 打开数字通道
:LA:DISP? 查询返回 ON

2. :DIGital<n>:TURN

命令格式:

:DIGital<n>:TURN {ON|OFF}
:DIGital<n>:TURN?

功能描述:

该命令用于设置和查询数字通道中指定通道的开关状态。<n>的取值范围为 0~15。

返回格式:

查询返回 ON 或 OFF。

举例:

:DIG2:TURN ON 打开数字通道 2
:DIG2:TURN? 查询返回 ON

3. :DIGital<n>:POSition

命令格式:

:DIGital<n>:POSition <position>

:DIGital<n>:POSition?

功能描述:

该命令用于设置和查询指定数字通道波形的显示位置。<n>可取 0~15。

选择大波形时，屏幕可显示 8 个通道的波形，<position>的取值范围为：0~7；

选择小波形时，屏幕可显示 16 个通道的波形，<position>的取值范围为：0~15。

返回格式:

查询返回 position 的设置值。

举例:

:DIG2:POS 3 设置数字通道 2 的波形在 D3 位置处显示

:DIG2:POS? 查询返回 3

4. :LA:THReshold

命令格式:

:LA:THReshold<value>[suffix]

:LA:THReshold?

功能描述:

该命令用于设置和查询数字通道的门限类型。<value>可取 TTL、CMOS、ECL 或 User（用户自定义，User 的设置范围是：-8.00~+8.00）。suffix 可取 V 或 mV，最小的时间间隔为 10mV。

举例:

:LA:THR COMS 设置门限类型为 COMS

:LA:THR? 查询返回 COMS

5. :LA:POSition:RESet

命令格式:

:LA:POSition:RESet

功能描述:

该命令用于重置波形显示位置，即使杂乱的屏幕显示恢复正常。

6. :LA:GROUp

命令格式:

:LA:GROUp<n> {ON|OFF}
:LA:GROUp<n>?

功能描述:

该命令用于设置和查询通道组 GROUp1 (D7~D0) 和 GROUp2(D15~D8) 的开关状态。<n>可取 1 或 2。

返回格式:

查询返回 ON 或 OFF。

举例:

:LA:GROU1 ON 打开数字通道组 D7~D0
:LA:GROU1? 查询返回 ON

7. :LA:GROUp<n>:SIZE

命令格式:

:LA:GROUp<n>:SIZE <size>
:LA:GROUp<n>:SIZE?

功能描述:

该命令用于设置和查询通道组的波形大小。<n>可取 1 或 2。<size>可取 Small 或 Big, 分别代表全屏显示 16 个通道或全屏显示 8 个通道。

返回格式:

查询返回 SMALL 或 BIG。

举例:

:LA:GROU1:SIZE S 设置 GROUp1 的波形大小为小波形
:LA:GROU1:SIZE? 查询返回 SMALL

KEY命令

KEY 命令用于控制 DS1000E, DS1000D 系列数字示波器面板上的按键和旋钮。

KEY 命令包括：

- :KEY:LOCK
- :KEY:RUN
- :KEY:AUTO
- :KEY:CHANnel1
- :KEY:CHANnel2
- :KEY:MATH
- :KEY:REF
- :KEY:F1
- :KEY:F2
- :KEY:F3
- :KEY:F4
- :KEY:F5
- :KEY:MNUoff
- :KEY:MEASure
- :KEY:CURSor
- :KEY:ACQuire
- :KEY:DISPlay
- :KEY:STORage
- :KEY:UTILity
- :KEY:MNU TIME
- :KEY:MNU TRIG
- :KEY:Trig%50
- :KEY:FORCe
- :KEY:V_POS_INC
- :KEY:V_POS_DEC
- :KEY:V_SCALE_INC
- :KEY:V_SCALE_DEC
- :KEY:H_SCALE_INC
- :KEY:H_SCALE_DEC
- :KEY:TRIG_LVL_INC
- :KEY:TRIG_LVL_DEC
- :KEY:H_POS_INC
- :KEY:H_POS_DEC
- :KEY:PROMPT_V
- :KEY:PROMPT_H
- :KEY:FUNcTion
- :KEY:+FUNcTion
- :KEY:-FUNcTion
- :KEY:LA
- :KEY:PROMPT_V_POS
- :KEY:PROMPT_H_POS
- :KEY:PROMPT_TRIG_LVL
- :KEY:OFF

下面将详细介绍每条命令的格式、功能和使用方法。

1. :KEY:LOCK

命令格式:

:KEY:LOCK {ENABLE|DISABLE}
:KEY:LOCK?

功能描述:

该命令用于开启/禁用前面板的按键功能（Force 键除外）。

返回格式:

查询返回 ENABLE 或 DISABLE。

举例:

:KEY:LOCK ENAB 开启前面板的按键功能
:KEY:LOCK? 查询返回 ENABLE

2. :KEY:RUN

命令格式:

:KEY:RUN

功能描述:

该命令用于控制示波器的运行状态，连续发送该命令，示波器将在停止和运行状态间切换。

3. :KEY:AUTO

命令格式:

:KEY:AUTO

功能描述:

该命令用于自动设定示波器的各项控制值，以产生适宜观察的波形显示。

4. :KEY:CHANnel1

命令格式:

:KEY:CHANnel1

功能描述:

该命令用于开启/关闭通道 1 及其菜单显示。连续发送该命令，可切换开关状态。

5. :KEY:CHANnel2**命令格式:**

:KEY:CHANnel2

功能描述:

该命令用于开启/关闭通道 2 及其菜单显示。连续发送该命令，可切换开关状态。

6. :KEY:MATH**命令格式:**

:KEY:MATH

功能描述:

该命令用于开启/关闭 MATH 功能及其菜单显示。连续发送该命令，可切换开关状态。

7. :KEY:REF**命令格式:**

:KEY:REF

功能描述:

该命令用于开启/关闭波形参考功能及其菜单显示。连续发送该命令，可切换开关状态。

8. :KEY:F1**命令格式:**

:KEY:F1

功能描述:

该命令用于控制 F1 按键所对应的菜单选项。若设有下级菜单，通过连续发送该

命令，可以循环选择下级菜单中的其他选项。

9. :KEY:F2

命令格式:

:KEY:F2

功能描述:

该命令用于控制 F2 按键所对应的菜单选项。若设有下级菜单，通过连续发送该命令，可以循环选择下级菜单中的其他选项。

10. :KEY:F3

命令格式:

:KEY:F3

功能描述:

该命令用于控制 F3 按键所对应的菜单选项。若设有下级菜单，通过连续发送该命令，可以循环选择下级菜单中的其他选项。

11. :KEY:F4

命令格式:

:KEY:F4

功能描述:

该命令用于控制 F4 按键所对应的菜单选项。若设有下级菜单，通过连续发送该命令，可以循环选择下级菜单中的其他选项。

12. :KEY:F5

命令格式:

:KEY:F5

功能描述:

该命令用于控制 F5 按键所对应的菜单选项。若设有下级菜单，通过连续发送该

命令，可以循环选择下级菜单中的其他选项。

13. :KEY:MNUOff

命令格式:

:KEY:MNUOff

功能描述:

该命令用于开启/关闭当前菜单显示。连续发送该命令，可切换开关状态。

14. :KEY:MEASure

命令格式:

:KEY:MEASure

功能描述:

该命令用于开启/关闭测量功能及其菜单显示。连续发送该命令，可切换开关状态。

15. :KEY:CURSor

命令格式:

:KEY:CURSor

功能描述:

该命令用于开启/关闭光标测量功能及其菜单显示，连续发送该命令，可在关闭、手动、追踪和自动测量四种光标模式之间进行切换。

16. :KEY:ACQuire

命令格式:

:KEY:ACQuire

功能描述:

该命令用于开启/关闭采样功能及其菜单显示。连续发送该命令，可切换开关状态。

17. :KEY:DISPlay

命令格式:

:KEY:DISPlay

功能描述:

该命令用于开启/关闭显示功能及其菜单显示。连续发送该命令,可切换开关状态。

18. :KEY:STORage

命令格式:

:KEY:STORage

功能描述:

该命令用于开启/关闭存储功能及其菜单显示。连续发送该命令,可切换开关状态。

19. :KEY:UTILity

命令格式:

:KEY:UTILity

功能描述:

该命令用于开启/关闭辅助功能及其菜单显示。连续发送该命令,可切换开关状态。

20. :KEY:MNUTIME

命令格式:

:KEY:MNUTIME

功能描述:

该命令用于开启/关闭水平系统及其菜单显示。连续发送该命令,可切换开关状态。

21. :KEY:MNUTRIG

命令格式:

:KEY:MNUTRIG

功能描述:

该命令用于开启/关闭触发功能及其菜单显示。连续发送该命令,可切换开关状态。

22. :KEY:Trig%50**命令格式:**

:KEY:Trig%50

功能描述:

该命令用于设定触发电平值至信号幅值的垂直中点。

23. :KEY:FORCe**命令格式:**

:KEY:FORCe

功能描述:

该命令用于解除远程控制的锁定状态。

24. :KEY:V_POS_INC**命令格式:**

:KEY:V_POS_INC

功能描述:

该命令以均匀步进增加当前通道的垂直位移。

25. :KEY:V_POS_DEC**命令格式:**

:KEY:V_POS_DEC

功能描述:

该命令以均匀步进减少当前通道的垂直位移。

26. :KEY:V_SCALE_INC

命令格式:

:KEY:V_SCALE_INC

功能描述:

该命令用于增加当前通道的垂直档位。

27. :KEY:V_SCALE_DEC

命令格式:

:KEY:V_SCALE_DEC

功能描述:

该命令用于减少当前通道的垂直档位。

28. :KEY:H_SCALE_INC

命令格式:

:KEY:H_SCALE_INC

功能描述:

该命令以 5-2-1 为步进减少当前通道的水平档位。

29. :KEY:H_SCALE_DEC

命令格式:

:KEY:H_SCALE_DEC

功能描述:

该命令以 1-2-5 为步进增加当前通道的水平档位。

30. :KEY:TRIG_LVL_INC

命令格式:

:KEY:TRIG_LVL_INC

功能描述:

该命令以均匀步进增加触发电平值。

31. :KEY:TRIG_LVL_DEC**命令格式:**

:KEY:TRIG_LVL_DEC

功能描述:

该命令以均匀步进减少触发电平值。

32. :KEY:H_POS_INC**命令格式:**

:KEY:H_POS_INC

功能描述:

该命令以均匀步进增加当前通道的水平位移。

33. :KEY:H_POS_DEC**命令格式:**

:KEY:H_POS_DEC

功能描述:

该命令以均匀步进减少当前通道的水平位移。

34. :KEY:PROMPT_V**命令格式:**

:KEY:PROMPT_V

功能描述:

该命令用于切换垂直档位的粗调/微调方式。粗调以 1-2-5 步进，微调以均匀间隔步进。

35. :KEY:PROMPT_H

命令格式:

:KEY:PROMPT_H

功能描述:

该命令用于开启/关闭延迟扫描功能。连续发送该命令，可切换开关状态。

36. :KEY:FUNCTION

命令格式:

:KEY:FUNCTION

功能描述:

该命令用于按下多功能旋钮。

37. :KEY:+FUNCTION

命令格式:

:KEY:+FUNCTION

功能描述:

该命令用于增加多功能旋钮的位移。

38. :KEY:-FUNCTION

命令格式:

:KEY:-FUNCTION

功能描述:

该命令用于减少多功能旋钮的位移。

39. :KEY:LA

命令格式:

:KEY:LA

功能描述:

该命令用于开启/关闭逻辑分析仪功能。连续发送该命令，可切换开关状态。

40. :KEY:PROMPT_V_POS**命令格式:**

:KEY:PROMPT_V_POS

功能描述:

该命令用于将通道的垂直位移恢复至零点。

41. :KEY:PROMPT_H_POS**命令格式:**

:KEY:PROMPT_H_POS

功能描述:

该命令用于将触发位移（或延迟扫描位移）恢复至零点。

42. :KEY:PROMPT_TRIG_LVL**命令格式:**

:KEY:PROMPT_TRIG_LVL

功能描述:

该命令用于将触发电平恢复至屏幕中心位置。

43. :KEY:OFF**命令格式:**

:KEY:OFF

功能描述:

连续发送该命令，将逐个关闭 CH1、CH2、MATH、REF 和 LA 通道。

其他命令

除了上述用于控制示波器基本操作的命令外，还有一些用于控制系统语言、频率计和蜂鸣器的命令。

其他命令包括：

- :INFO:LANGuage
- :COUNter:ENABle
- :BEEP:ENABle
- :BEEP:ACTion

下面将详细介绍每条命令的格式、功能和使用方法。

1. :INFO:LANGUage

命令格式:

:INFO:LANGUage <lang>
:INFO:LANGUage?

功能描述:

该命令用于设置和查询系统语言。<lang>可取 SIMPlifiedChinese (简体中文)、TRADitionalChinese (繁体中文)、ENGLish (英文)、KORean (韩语)、JAPanese (日语)、FRENch (法语)、GERMan (德语)、RUSSian (俄语)、SPANish (西班牙语) 或 PORTuguese (葡萄牙语)。

返回格式:

查询返回 Simplified Chinese、Traditional Chinese、English、Korean、Japanese、French、German、Russian、Spanish 或 Portuguese。

举例:

:INFO:LANG SIMP 设置系统语言为简体中文
:INFO:LANG? 查询返回 Simplified Chinese

2. :COUNter:ENABle

命令格式:

:COUNter:ENABle {ON|OFF}
:COUNter:ENABle?

功能描述:

该命令用于设置和查询示波器频率计功能的开关状态。

返回格式:

查询返回 ON 或 OFF。

举例:

:COUN:ENAB ON 打开频率计功能
:COUN:ENAB? 查询返回 ON

3. :BEEP:ENABle

命令格式:

:BEEP:ENABle {ON|OFF}

:BEEP:ENABle?

功能描述:

该命令用于设置和查询蜂鸣器的开关状态。

返回格式:

查询返回 ON 或 OFF。

举例:

| | |
|---------------|---------|
| :BEEP:ENAB ON | 打开蜂鸣器 |
| :BEEP:ENAB? | 查询返回 ON |

4. :BEEP:ACTion**命令格式:**

:BEEP:ACTion

功能描述:

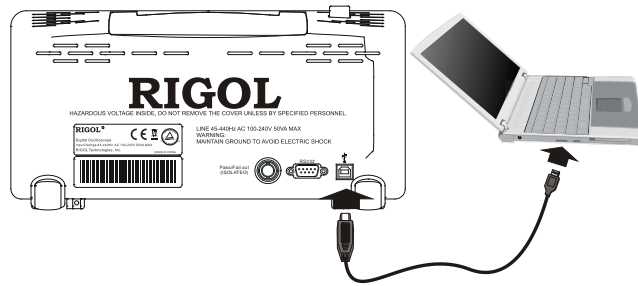
该命令用于产生一次蜂鸣（无论蜂鸣器是否打开）。

第3章 编程实例

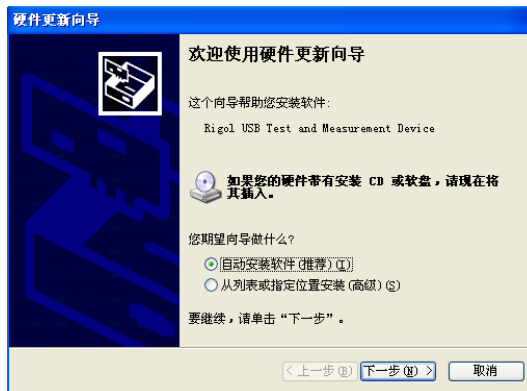
本章例举了在 Visual C++ 6.0、Visual Basic 6.0 以及 LabView 8.2 开发环境中如何使用命令实现示波器常用功能的编程实例。用户可使用 RIGOL 提供的 USB 驱动，也可以基于 VISA 库进行编程。如使用 VISA 编程，请确认您的电脑上已经安装 NI 的 VISA 库（可到 NI 网站 <http://www.ni.com/china> 下载），本文中默认路径为：C:\Program Files\IVI Foundation\VISA。

编程准备

本文应用 DS1000E, DS1000D 示波器的 USB 接口与 PC 通信。请使用 USB 数据线将示波器后面板的 USB Device 接口与 PC 相连，如下图所示。



示波器与 PC 正确连接后，接通仪器电源，此时 PC 上将弹出“硬件更新向导”对话框，请按照向导的提示安装“Rigol USB Test and Measurement Device”（使用 RIGOL USB 驱动时）或“USB Test and Measurement Device”（使用 VISA 时）。



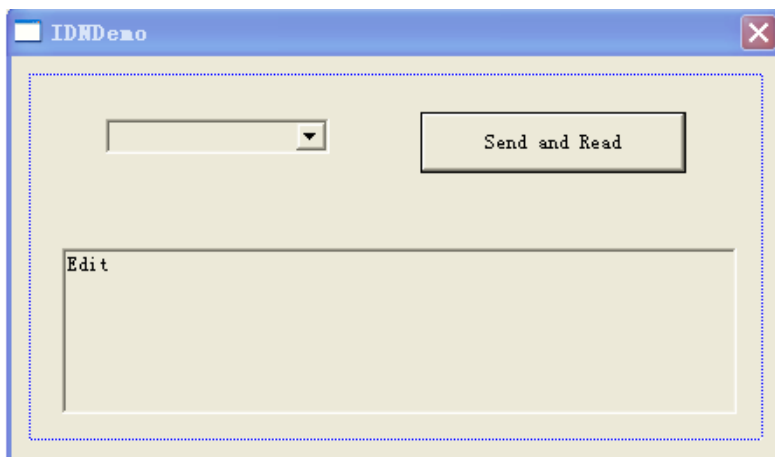
至此，编程准备工作到此结束，下面将详细介绍在 Visual C++ 6.0、Visual Basic 6.0 以及 LabView 8.2 开发环境中的编程实例。

基于RIGOL USB驱动的编程

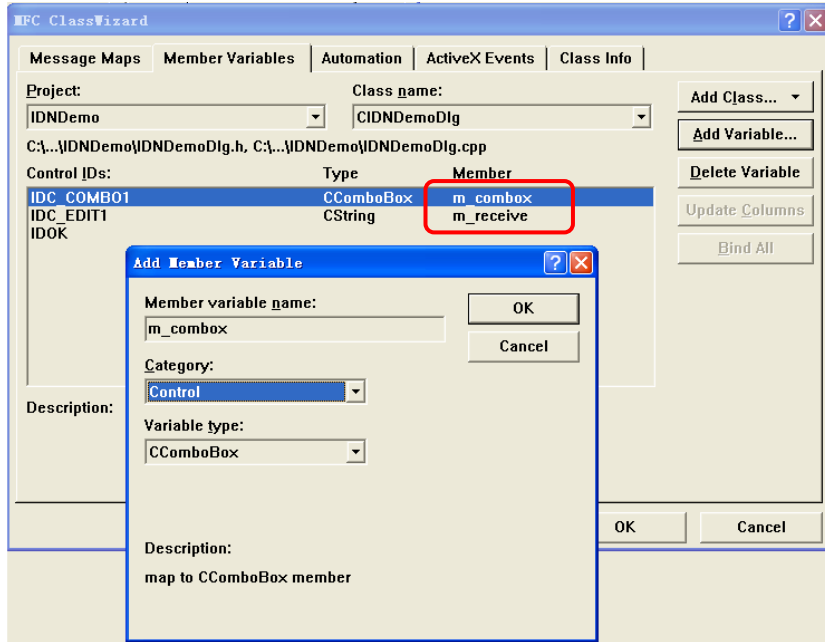
Visual C++ 6.0 编程实例

本例制作一个发送查询命令并返回结果的 Demo。进入 Visual C++6.0 编程环境，按照下列步骤操作：

1. 新建一个基于 MFC 的工程 IDNDemo。
2. 添加控件： Button、 Combo Box 和  Edit Box。
3. 修改控件属性：
 - 1) 将 Button 命名为“Send and Read”，如下图所示。



- 2) 打开 Com Box 属性中的 Data 对话框，手动输入以下两条命令：
*IDN?
:ACQ:TYPE?
- 3) 为 Com Box 和 Edit Box 控件分别添加变量：m_combox 和 m_receive，如下图所示：



注：添加变量 m_combox 时，Category 选择“Control”；添加变量 m_receive 时，Category 选择“Value”。

4. 将 Demo 文件夹中的动态链接库 RigoITMCUsb_UI.dll 拷贝到新建工程的目录下，以便引用。

说明：在 IDNDemo 中，我们只需用到动态链接库中的三个函数：GetTMCDeviceNum, WriteUSB 及 ReadUSB（函数在头文件 RigoIusb.h 中声明）。下面以 GetTMCDeviceNum 函数为例为您介绍是如何调用这三个函数的：

```
HMODULE module = LoadLibrary("RigoITMCUsb_UI.dll");
typedef int (CALLBACK* pGetDeviceNum)();
if(module)
{
    pGetDeviceNum GetDevice
    = (pGetDeviceNum)GetProcAddress(module,"GetTMCDeviceNum");
    int j = GetDevice(); //返回
}
```

5. 添加代码：

进入编程环境，请在头文件中对 void SendToUSB(CString cmd)和 unsigned char* Read_USB(void)进行声明。添加如下代码：

```
//发送设置命令

void SendToUSB(CString cmd)
{
    int temp,j;
    CString cmdstr;
    cmdstr = cmd;
    unsigned char * strInput;

    //调用 RigoITMCUsb_UI.dll 中的 GetTMCDeviceNum 及 WriteUSB 函数
    HMODULE module = LoadLibrary("RigoITMCUsb_UI.dll");

    typedef bool (CALLBACK* pWrite)(unsigned long, unsigned char,unsigned
char,unsigned long,unsigned char*);
    typedef int (CALLBACK* pGetDeviceNum)();

    temp = cmdstr.GetLength();
    strInput = (unsigned char*)cmdstr.GetBuffer(10);

    if(module)
    {
        pGetDeviceNum GetDevice =
(pGetDeviceNum)GetProcAddress(module,"GetTMCDeviceNum");
        pWrite Write = (pWrite)GetProcAddress(module,"WriteUSB");

        j = GetDevice();

        if( j > 0 )
        {
            //向示波器发送命令
            //接口的定义在 Rigolusb.h 中注明
            Write(0,1,1,temp,strInput);
        }
    }
}

//从示波器获取返回值
unsigned char* Read_USB( void )
```

```
{
    unsigned long infoSize;
    //开辟空间，用于存放示波器的返回值
    int len = 100;
    unsigned char* strInput = new unsigned char [len];
    CString str;

    //调用 RigolTMCUsb_UI.dll 中的 ReadUSB 及 WriteUSB 函数
    HMODULE module = LoadLibrary("RigolTMCUsb_UI.dll");

    typedef bool (CALLBACK* pWrite)(unsigned long, unsigned char,unsigned
char,unsigned long,unsigned char*);
    typedef bool (CALLBACK* pRead)(unsigned long,unsigned long*,unsigned
char*);

    if(module)
    {
        pWrite Write = (pWrite)GetProcAddress(module,"WriteUSB");
        pRead Read = (pRead)GetProcAddress(module,"ReadUSB");

        //向示波器发送取值命令，发送此命令后可以从示波器获得数值
        Write(0,2,2,len,0);
        //从示波器中读取返回值
        Read(0,&infoSize,strInput);
    }
    //开辟空间用于存放示波器的有效返回值
    unsigned char *buffer = new unsigned char[infoSize];

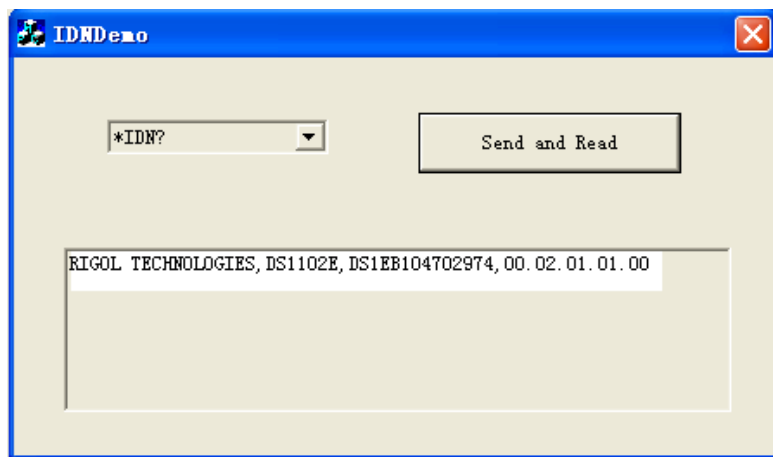
    memcpy(buffer,strInput,infoSize);
    buffer[infoSize] = 0;
    delete []strInput;

    //返回从示波器获得的字符串
    return buffer;
}
```

6. 双击 Button 控件，进入编程环境，添加如下代码：

```
unsigned char* readInfo;  
CString strInput;  
//char* stringTemp;  
CString s,strTemp;  
  
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);  
strInput = (char *) (LPCTSTR)strTemp;  
SendToUSB(strInput);  
readInfo = Read_USB();  
  
//将结果显示出来  
UpdateData (TRUE);  
m_receive = readInfo;  
UpdateData (FALSE);
```

7. 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 正确连接时，在 Combo Box 中选择一条命令如 “*IDN?”，点击 “Send and Read” 按键，将显示示波器返回的结果，如下图所示：

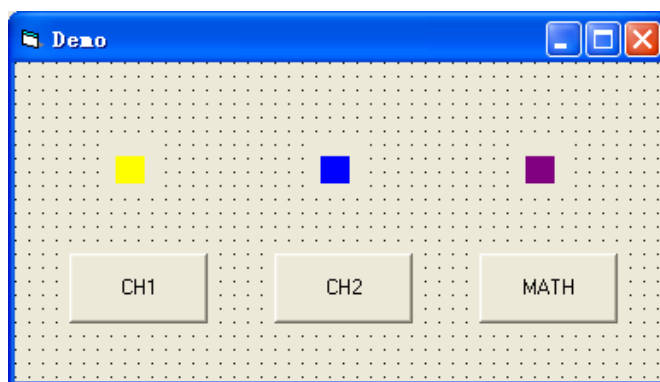


Visual Basic 6.0 编程实例

例 1：通道状态查询及设置

本例制作一个查询和设置通道状态的 Demo。进入 Visual Basic 6.0 编程环境，按照下列步骤操作：

1. 新建一个 Standard EXE 工程，并将 Demo 文件夹中的动态链接库 RigolTMCUsb_UI.dll 拷贝到新建工程的目录下，以便引用。
2. 将 Form1 命名为 Demo。在 Demo 中添加三个 CommandButton 控件（分别代表 CH1、CH2 和 MATH）和三个 Label 控件（分别显示 CH1、CH2 和 MATH 的状态，打开时显示通道颜色，关闭时显示灰色），如下图所示：



3. 进入编程环境，引用 RigolTMCUsb_UI.dll 动态链接库中的函数（**ReadUSB**，**WriteUSB**，**GetTMCDeviceNum**）。在工程代码区域中添加如下代码：

```
Private Declare Function ReadUSB Lib " RigolTMCUsb_UI.dll " (ByVal Index As Long, ByVal IpszLength As Long, ByVal IpszBuffer As Long) As Boolean
```

```
Private Declare Function WriteUSB Lib " RigolTMCUsb_UI.dll " (ByVal Index As Long, ByVal msgid As Long, ByVal tag As Long, ByVal length As Long, ByVal IpszBuffer As Long) As Boolean
```

```
Private Declare Function GetTMCDeviceNum Lib "RigolTMCUsb_UI.dll" () As Long
```

4. 添加子函数，以完成发送命令、获取返回值的功能，代码如下：

'发送设置命令

```
Sub SendToUSB(cmd As String)
    Dim retcode As Boolean
    Dim send_buf(256) As Byte
    Dim temp As Long
    Dim cmdstr As String

    cmdstr = cmd

    temp = Len(cmdstr)
    For i = 0 To temp - 1
        tempStr = Mid(cmdstr, i + 1, 1)
        send_buf(i) = Asc(tempStr)
    Next i
    '向示波器发送命令
    '接口的定义在 Rigolusb.h 中注明
    retcode = WriteUSB(0, 1, 1, Len(cmdstr), VarPtr(send_buf(0)))
End Sub
```

'获取返回值

```
Public Function Read_USB() As String

    Dim retcode As Boolean
    Dim rcv_buffer(256) As Byte
    Dim tmpstr As String
    Dim i, size As Long
    Dim rSize As Long

    rSize = 100
    tmpstr = ""

    '向示波器发送取值命令，发送此命令后可以从示波器获得数值
    retcode = WriteUSB(0, 2, 2, 256, 0)
    '从示波器中读取返回值
    retcode = ReadUSB(0, VarPtr(rSize), VarPtr(rcv_buffer(0)))
```

```
For i = 0 To (rSize - 1)
    tmpstr = tmpstr + Chr(rcv_buffer(i))
Next i

Read_USB = tmpstr
End Function
```

5. 双击 CH1 控件，添加如下代码：

```
Dim i As Long
Dim sendbuf As String
Dim readbuf As String

rSize = 100
sendbuf = ":CHAN1:DISP?"

'初始化 USB 设备
i = GetTMCDeviceNum

'发送询问命令，并读取返回值
Call SendToUSB(sendbuf)
readbuf = Read_USB()

'根据返回值来确定对应通道灯的状态
If readbuf = "ON" Then

' 发送设置命令
sendbuf = ":CHAN1:DISP 0"
Call SendToUSB(sendbuf)

Label1(0).ForeColor = &H808080 '灰色

Else
sendbuf = ":CHAN1:DISP 1"
Call SendToUSB(sendbuf)
Label1(0).ForeColor = &HFFFF& '黄色
End If
```

注：CH2、MATH 与 CH1 代码类似，用户可对部分参数进行修改获得相应代码进行添加，这里不再赘述。

6. 双击 Label 控件，添加如下代码：

```
Dim i As Long
Dim sendbuf As String
Dim readbuf As String

rSize = 100
sendbuf = ":CHAN1:DISP?"

i = GetTMCDiviceNum

Call SendToUSB(sendbuf)
readbuf = Read_USB()

If readbuf = "ON" Then

Label1(0).ForeColor = &HFFFF& '黄色

Else

Label1(0).ForeColor = &H808080 '灰色

End If

sendbuf = ":CHAN2:DISP?"

i = GetTMCDiviceNum

Call SendToUSB(sendbuf)
readbuf = Read_USB()

If readbuf = "ON" Then

Label1(1).ForeColor = &HFF0000 '蓝色

Else

Label1(1).ForeColor = &H808080 '灰色
```

```
End If
```

```
sendbuf = ":MATH:DISP?"
```

```
i = GetTMCDDeviceNum
```

```
Call SendToUSB(sendbuf)
```

```
readbuf = Read_USB()
```

```
If readbuf = "ON" Then
```

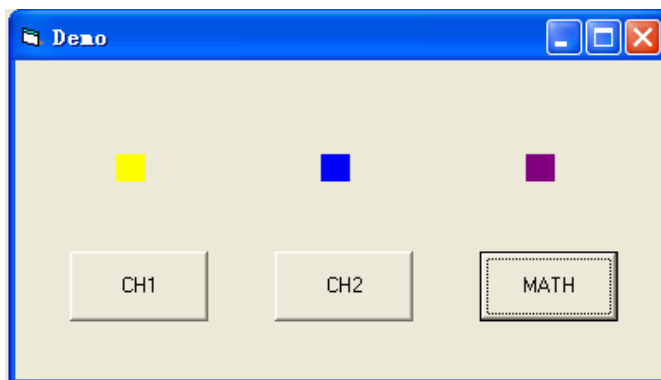
```
Label1(2).ForeColor = &H800080 '紫色
```

```
Else
```

```
Label1(2).ForeColor = &H808080 '灰色
```

```
End If
```

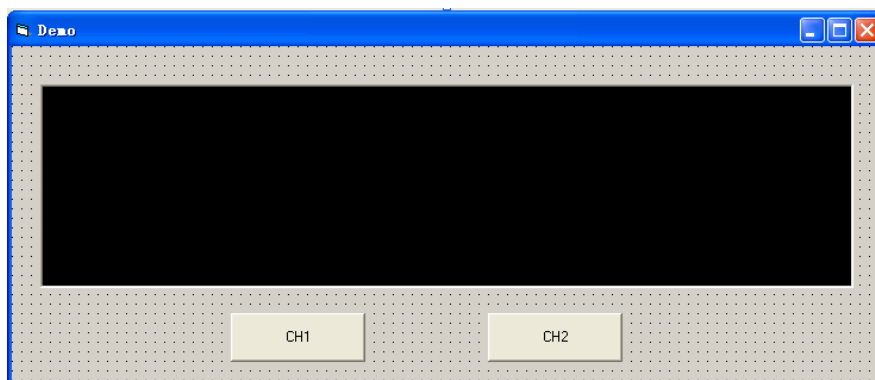
7. 保存、运行整个工程，可得到 Demo 的单个可执行文件。当示波器与 PC 正确相连时，可实现对任意一个通道的开/关控制，如下图所示：



例 2：波形数据读取

本例制作一个读取波形数据的 Demo。进入 Visual Basic 6.0 编程环境，按照下列步骤操作：

1. 新建一个 Standard EXE 工程，并将 Demo 文件夹中的动态链接库 RigoITMCUsb_UI.dll 拷贝到新建工程的目录下，以便引用。
2. 将 Form1 命名为 Demo。在 Demo 中添加两个 CommandButton 控件（分别代表 CH1 和 CH2）和一个 PictureBox 控件（用来显示 CH1 及 CH2 的波形数据），如下图所示：



3. 选择 Project→Add Module，新建一个 Module，添加如下代码，用来声明全局变量 rcv_buffer：

```
Global rcv_buffer(10240) As Byte
```

注：由于示波器返回的数据量较大，用该 buffer 可存储示波器返回的波形数据。

4. 进入编程环境，引用 **RigoITMCUsb_UI.dll** 动态链接库中的函数（**ReadUSB**，**WriteUSB**，**GetTMCDeviceNum**）。在工程代码区域中添加如下代码：

```
Private Declare Function ReadUSB Lib "RigoITMCUsb_UI.dll" (ByVal Index As Long, ByVal lpszLength As Long, ByVal lpszBuffer As Long) As Boolean
```

```
Private Declare Function WriteUSB Lib "RigoITMCUsb_UI.dll" (ByVal Index As Long, ByVal msgid As Long, ByVal tag As Long, ByVal length As Long, ByVal lpszBuffer As Long) As Boolean
```

```
Private Declare Function GetTMCDeviceNum Lib "RigoITMCUsb_UI.dll" () As Long
```

5. 添加子函数，以完成发送命令、获取返回值的功能，代码如下：

'发送设置命令

```
Sub SendToUSB(cmd As String)
```

```
    Dim retcode As Boolean
```

```
    Dim send_buf(256) As Byte
```

```
    Dim temp As Long
```

```
    Dim cmdstr As String
```

```
    cmdstr = cmd
```

```
    temp = Len(cmdstr)
```

```
    For i = 0 To temp - 1
```

```
        tempStr = Mid(cmdstr, i + 1, 1)
```

```
        send_buf(i) = Asc(tempStr)
```

```
    Next i
```

'向示波器发送命令

'接口的定义在 Rigolusb.h 中注明

```
    retcode = WriteUSB(0, 1, 1, Len(cmdstr), VarPtr(send_buf(0)))
```

```
End Sub
```

'获取返回值

```
Public Function Read_USB() As Long
```

```
    Dim retcode As Boolean
```

```
    Dim i, size As Long
```

```
    Dim rSize As Long
```

```
    rSize = 100
```

'向示波器发送取值命令

```
    retcode = WriteUSB(0, 2, 2, 10240, 0)
```

'从示波器中读取返回值

```
    retcode = ReadUSB(0, VarPtr(rSize), VarPtr(rcv_buffer(0)))
```

```
Read_USB = rSize
```

```
End Function
```

6. 双击 CH1 控件，添加如下代码：

```
Dim i As Long
```

```
Dim sendbuf As String
```

```
Dim DataLen As Long
```

```
Dim stepW As Integer
```

```
Dim stepH As Integer
```

```
sendbuf = ":WAV:DATA? CHAN1"
```

```
'初始化 USB 设备
```

```
i = GetTMCDeviceNum
```

```
'发送询问命令，并读取返回值
```

```
Call SendToUSB(sendbuf)
```

```
DataLen = Read_USB()
```

```
stepW = Picture1.Width
```

```
stepH = Picture1.Height / 256
```

```
'清除波形
```

```
Picture1.Cls
```

```
For i = 0 To DataLen - 1
```

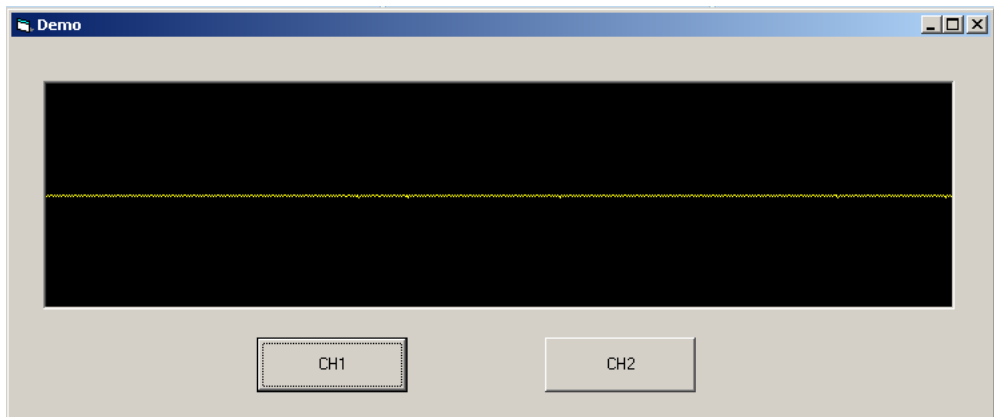
```
'将波形数据点连接，并将波形设置为黄色
```

```
Picture1.Line ((i * stepW / DataLen), (rcv_buffer(i) * stepH)-(((i + 1) * stepW / DataLen), (rcv_buffer((i + 1))) * stepH), &HFFFF&
```

```
Next i
```

注：CH2 与 CH1 代码类似，用户可对部分参数进行修改获得相应代码进行添加，这里不再赘述。

7. 保存、运行整个工程，可得到 Demo 的单个可执行文件。当示波器与 PC 正确相连时，可实现对任一通道波形数据的读取，如下图所示：

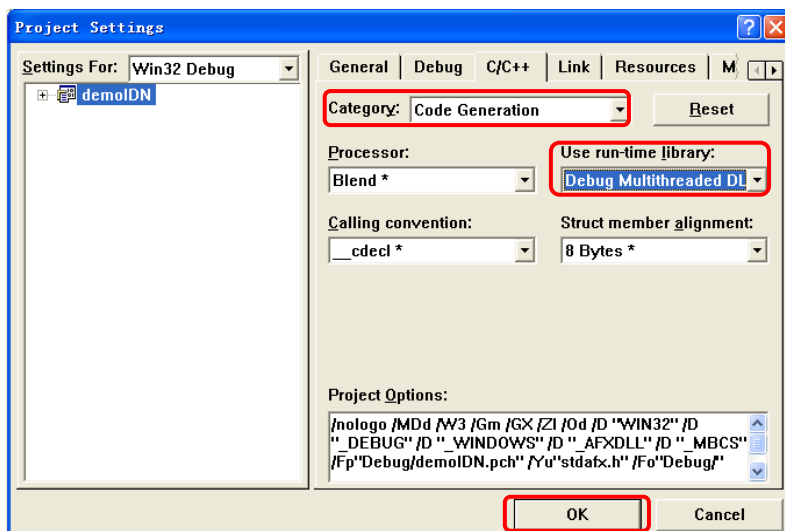


基于VISA库的编程

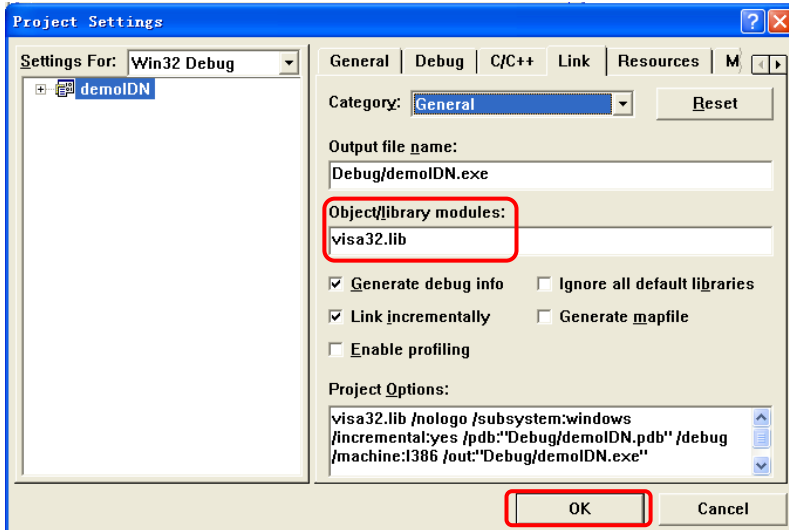
Visual C++ 6.0 编程实例

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 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。



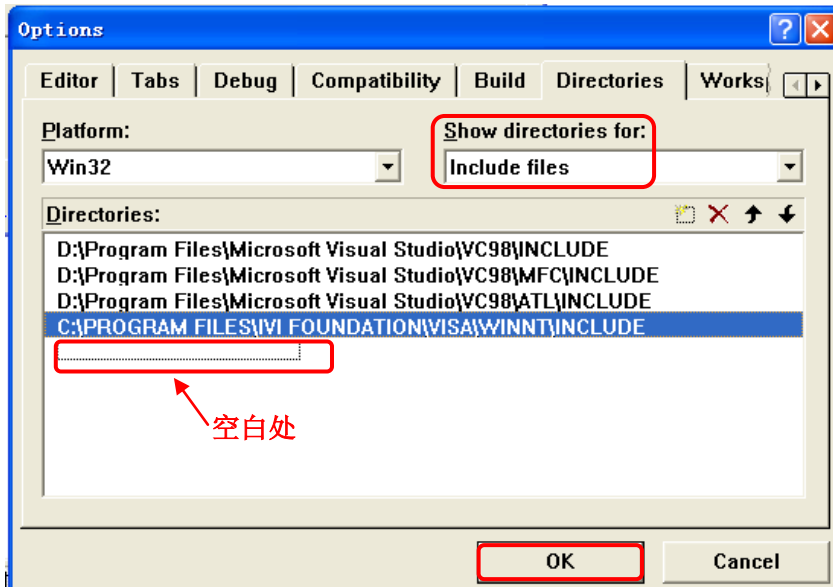
3. 打开 Project->Settings 中的 Link 选项卡，在 Object/library modules 中手动添加 visa32.lib。



4. 打开 Tools→Options 中的 Directories 选项卡。

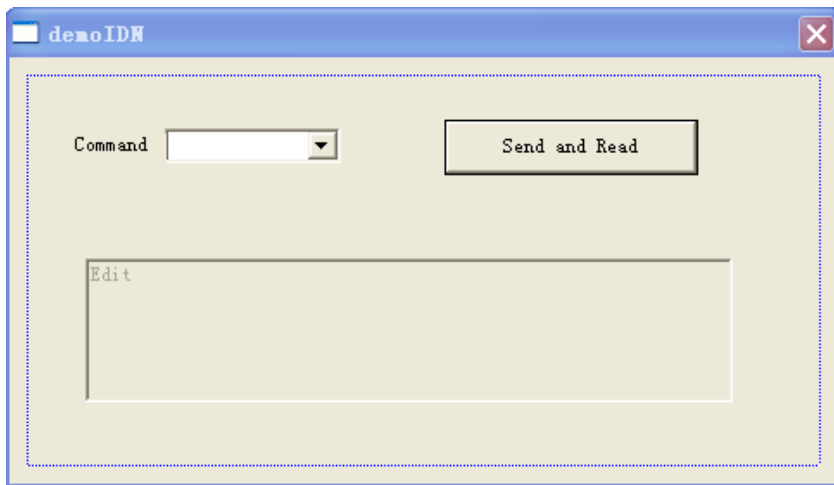
在 Show directories for 中选择 Include files，双击 Directories 选框中的空白处添加 Include 的路径：C:\Program Files\IVI Foundation\VISA\WinNT\include。

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。



注：至此，VISA 库添加完毕。

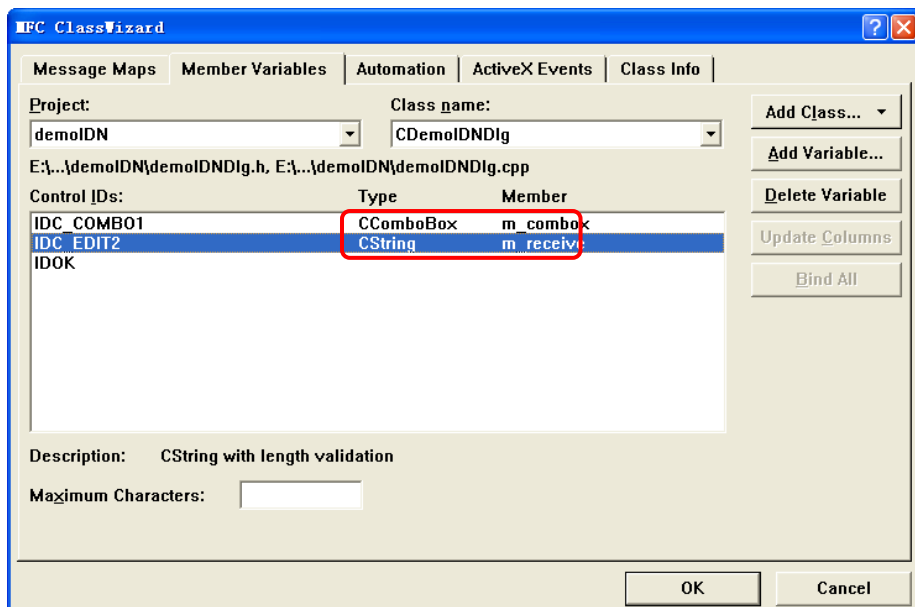
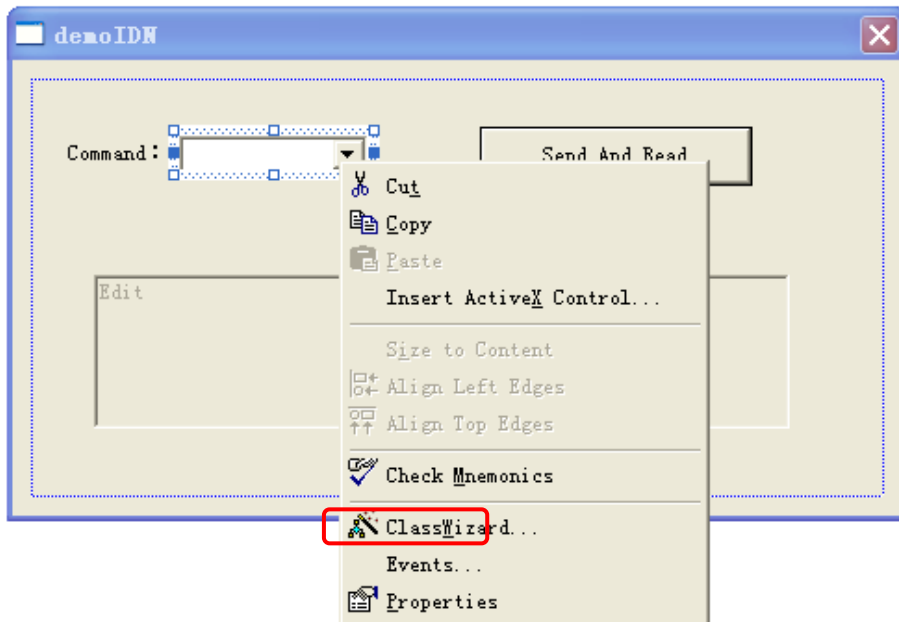
5. 添加控件 Text, Com box, Button 及 Edit, 得到下图所示界面:



6. 修改控件属性。

- 1) 将 Text 命名为 “Command”。
- 2) 打开 Com box 属性中的 Data 项, 手动输入三个命令:
*IDN?
*OPC?
:ACQuire:TYPE?
- 3) 打开 Edit 属性中的 General 项, 选中 Disable。
- 4) 将 Button 命名为 Send and Read。

7. 为 Com 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);

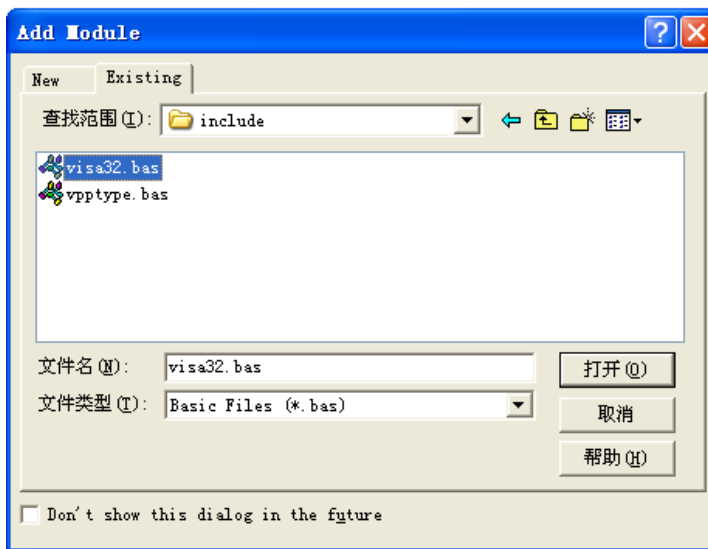
//将结果显示出来
UpdateData (TRUE);
m_receive = buf;
UpdateData (FALSE);
viClose (vi);
viClose (defaultRM);
```

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，选择一条命令如*IDN?，按“Send and Read”按键，将显示示波器返回的结果。

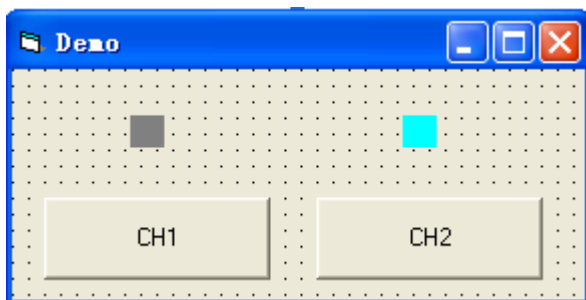
Visual Basic 6.0 编程实例

进入 Visual Basic 6.0 编程环境，按照下列步骤操作：

1. 建立一个 Standard EXE 工程。
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 visa.bas 文件并添加。



3. 在 Demo 中添加如下两个按钮，分别代表 CH1 和 CH2。添加两个 Label: Label1(0), Label1(1)，分别显示 CH1 和 CH2 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 和 CH2 的控制。
以下为 CH1 的代码，其他通道代码类似。

```
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

' 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)
If Left(strRes, 2) = "ON" 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

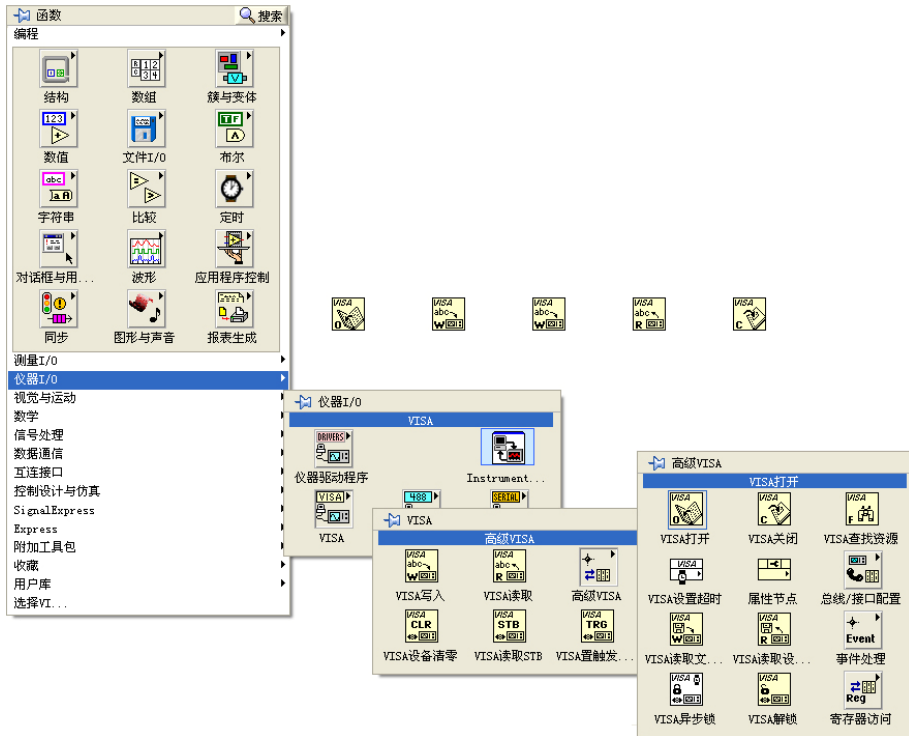
' 关闭资源
Call viClose(vi)
Call viClose(defrm)
```

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。

LabVIEW 8.2 编程实例

进入 LabVIEW 8.2 开发环境，按照下列步骤操作：

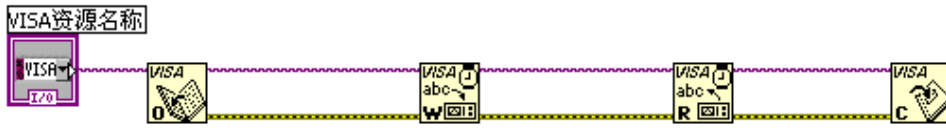
1. 打开 LabVIEW，进入程序框图面板，从仪器 I/O→VISA 中分别添加四个函数：“VISA 打开”，“VISA 读取”，“VISA 写入”以及“VISA 关闭”：



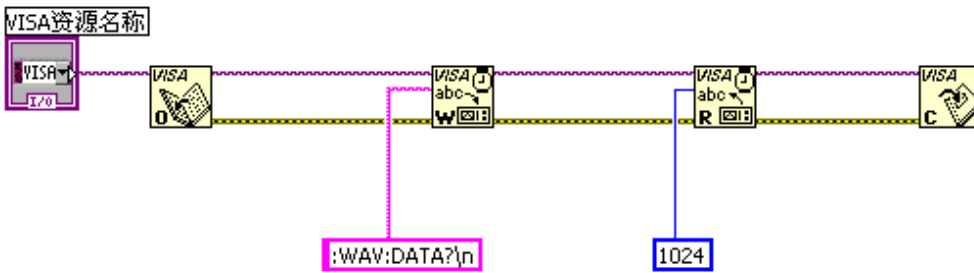
2. 将鼠标移动至“VISA 打开”控件的“VISA 资源名称”项目上，点击右键，选择创建→输入控件。完成后如图所示：



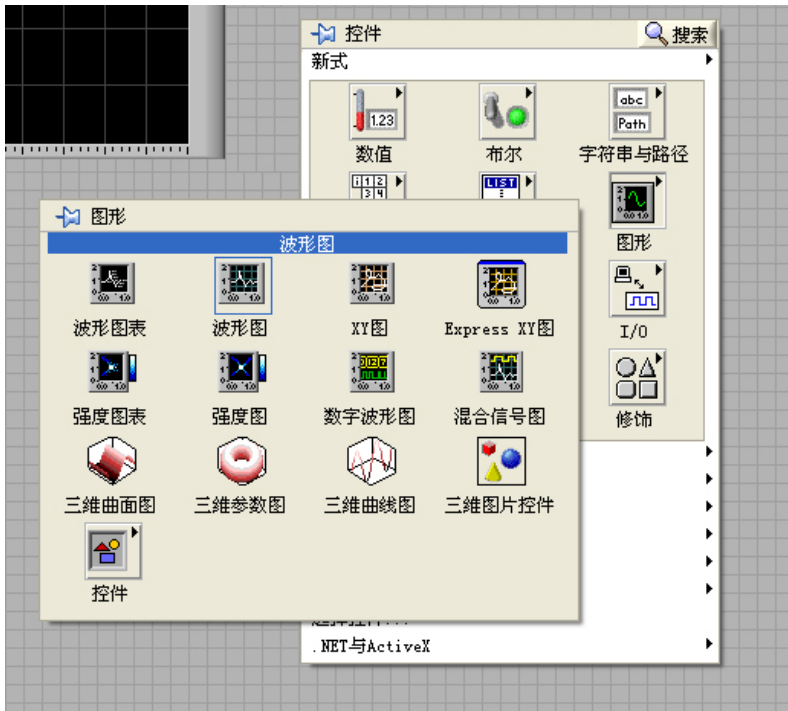
- 3. 将所有函数的“VISA 资源名称”和“VISA 资源名称输出”连接，“错误输出”和“错误输入连接”：



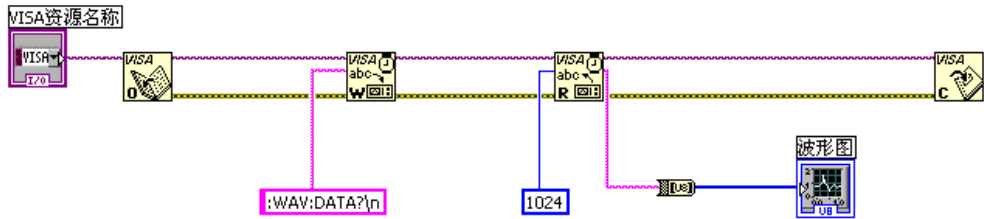
- 4. 在两个“VISA 写入”控件的“写入缓冲区”上添加文本框，写入：“:WAV:DATA?\n”，用来读取屏幕波形数据。



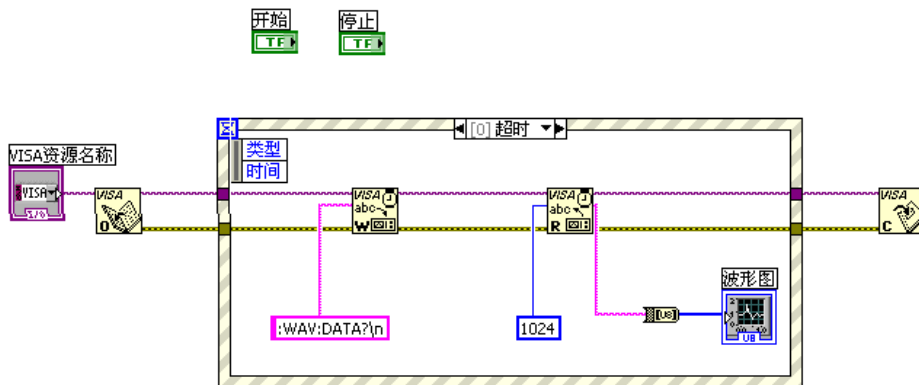
- 5. 打开 LabVIEW 前面板，从新式→图形→波形图中添加一个波形图控件：



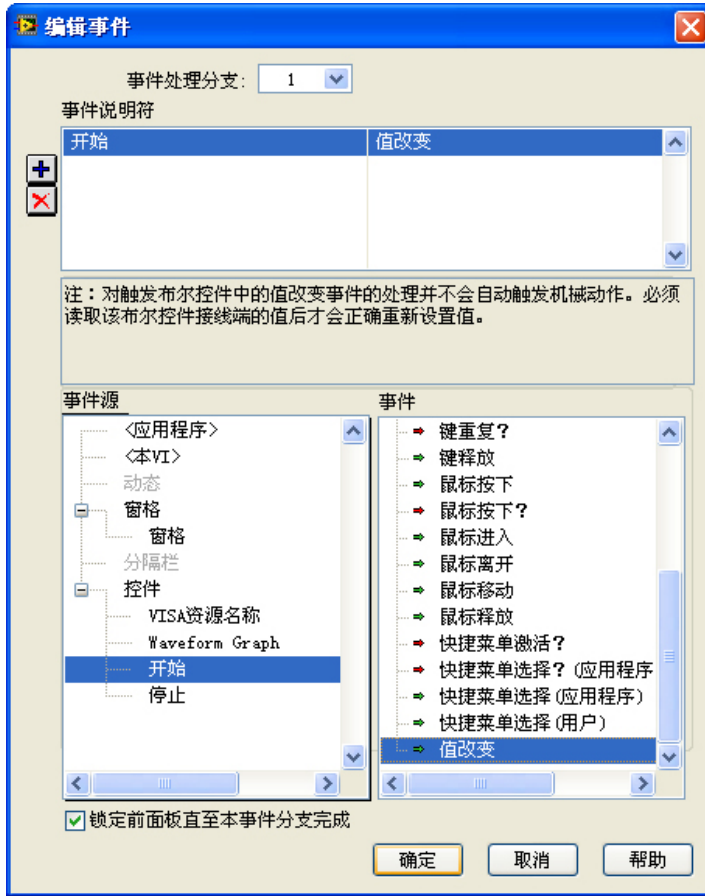
6. 打开 LabVIEW 程序框图，点击右键，在“编程”→“字符串”→“字符串/数组/路径转换”中选择一个“字符串至字节数组字符转换”函数，用其将刚刚创建的波形图与“VISA 读取”函数中的“读取缓冲区”连接起来。



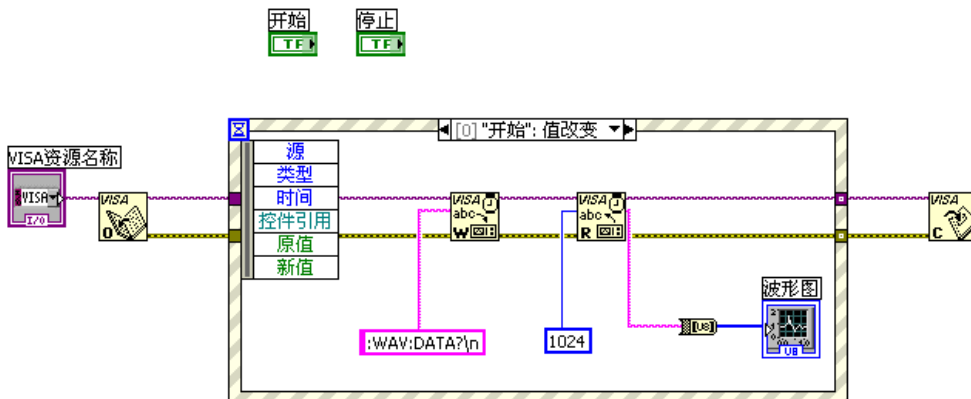
7. 增加一个事件结构和一个 While 循环，并添加两个按钮，一个用来控制开始捕捉波形，一个用来停止程序，完成后如图所示：



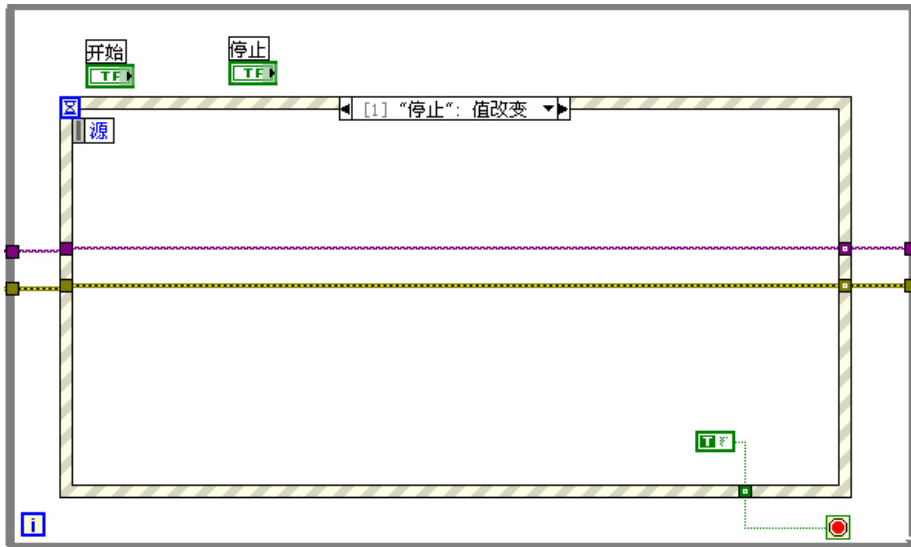
8. 在“选择器标签”上点击右键，选择“编辑事件分支”或者“添加事件分支”，分别为按钮添加事件，“开始”进行波形捕捉，“停止”退出程序：



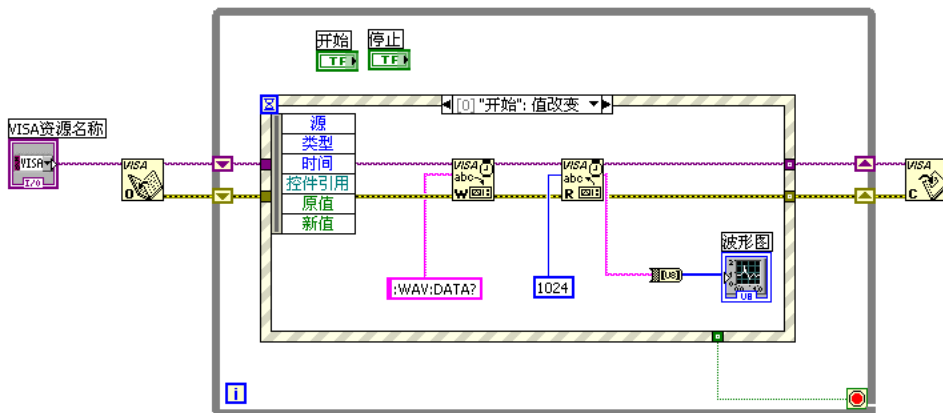
“开始捕捉”事件设置好之后如下图所示：



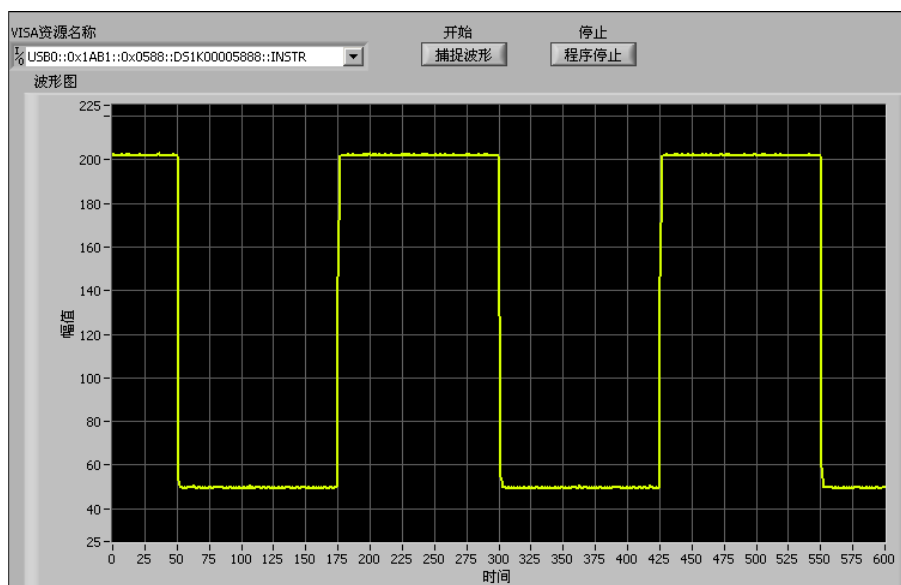
9. 为程序增加一个 While 循环，并增添一个“布尔”→“真常量”将“程序停止”按钮的事件指向 While 退出：



10. 将 While 循环上的 VISA 资源名输入和错误输入更改成“移位寄存器”，程序的制作就此结束：



11. 调整前面板的样式，点击“捕捉波形”后可得到如下界面（示波器已正确连接）：



附录 命令速查 A-Z

*IDN? 2-3

*RST 2-3

A

:AUTO 2-5

:ACQUIRE:TYPE 2-7

:ACQUIRE:MODE 2-7

:ACQUIRE:AVERAGES 2-7

:ACQUIRE:SAMPLINGRATE? 2-8

:ACQUIRE:MEMDEPTH <depth> 2-8

B

:BEEP:ENABLE 2-89

:BEEP:ACTION 2-90

C

:CHANNEL<n>:BWLIMIT 2-53

:CHANNEL<n>:COUPLING 2-53

:CHANNEL<n>:DISPLAY 2-53

:CHANNEL<n>:INVERT 2-54

:CHANNEL<n>:OFFSET 2-54

:CHANNEL<n>:PROBE 2-55

:CHANNEL<n>:SCALE 2-55

:CHANNEL<n>:FILTER 2-56

:CHANNEL<n>:MEMORYDEPTH? 2-56

:CHANNEL<n>:VERNIER 2-57

:COUNTER:ENABLE 2-89

:COUNTER:VALUE? 2-66

D

:DISPLAY:TYPE 2-11

:DISPLAY:GRID 2-11

:DISPLAY:PERSIST 2-11

:DISPLAY:MNUDISPLAY 2-12

:DISPLAY:MNUSTATUS 2-12

:DISPLAY:CLEAR 2-13

:DISPLAY:BRIGHTNESS 2-13

:DISPLAY:INTENSITY 2-13

:DIGITAL<n>:TURN 2-71

:DIGITAL<n>:POSITION 2-71

F

:FFT:DISPLAY 2-50

:FORCETRIG 2-25

H

:HARDCOPY 2-5

I

:INFO:LANGUAGE 2-89

K

:KEY:LOCK 2-78

:KEY:RUN 2-78

:KEY:AUTO 2-78

:KEY:CHANNEL1 2-78

:KEY:CHANNEL2 2-79

:KEY:MATH 2-79

:KEY:REF 2-79

:KEY:F1 2-79

:KEY:F2 2-80

:KEY:F3 2-80

:KEY:F4 2-80

:KEY:F5 2-80

:KEY:MNUIOFF 2-81

:KEY:MEASURE 2-81

:KEY:CURSORS 2-81

:KEY:ACQUIRE 2-81

:KEY:DISPlay 2-82
 :KEY:STORAge 2-82
 :KEY:UTILity 2-82
 :KEY:MNUTIME 2-82
 :KEY:MNUTRIG 2-82
 :KEY:Trig%50 2-83
 :KEY:FORCe 2-83
 :KEY:V_POS_INC 2-83
 :KEY:V_POS_DEC 2-83
 :KEY:V_SCALE_INC 2-84
 :KEY:V_SCALE_DEC 2-84
 :KEY:H_SCALE_INC 2-84
 :KEY:H_SCALE_DEC 2-84
 :KEY:TRIG_LVL_INC 2-84
 :KEY:TRIG_LVL_DEC 2-85
 :KEY:H_POS_INC 2-85
 :KEY:H_POS_DEC 2-85
 :KEY:PROMPT_V 2-85
 :KEY:PROMPT_H 2-86
 :KEY:FUNCTion 2-86
 :KEY:+FUNCTion 2-86
 :KEY:-FUNCTion 2-86
 :KEY:LA 2-86
 :KEY:PROMPT_V_POS 2-87
 :KEY:PROMPT_H_POS 2-87
 :KEY:PROMPT_TRIG_LVL 2-87
 :KEY:OFF 2-87

L

:LA:DISPlay 2-71
 :LA:THReshold 2-72
 :LA:POSition:RESet 2-72
 :LA:GROUp 2-73
 :LA:GROUp<n>:SIZE 2-73

M

:MATH:DISPlay 2-50
 :MATH:OPERate 2-50

:MEASure:CLEar 2-59
 :MEASure:VPP? 2-59
 :MEASure:VMAX? 2-59
 :MEASure:VMIN? 2-59
 :MEASure:VAMplitude? 2-60
 :MEASure:VTOP? 2-60
 :MEASure:VBASe? 2-60
 :MEASure:VAverage? 2-61
 :MEASure:VRMS? 2-61
 :MEASure:OVERshoot? 2-61
 :MEASure:PREShoot? 2-61
 :MEASure:FREQuency? 2-62
 :MEASure:RISetime? 2-62
 :MEASure:FALLtime? 2-62
 :MEASure:PERiod? 2-63
 :MEASure:PWIDth? 2-63
 :MEASure:NWIDth? 2-63
 :MEASure:PDUTYcycle? 2-64
 :MEASure:NDUTYcycle? 2-64
 :MEASure:PDELay? 2-64
 :MEASure:NDELay? 2-64
 :MEASure:TOTal 2-65
 :MEASure:SOURce 2-65

R

:RUN 2-5

S

:STOP 2-5
 :STORAge:FACTory:LOAD 2-48

T

:TIMebase:MODE 2-16
 :TIMebase[:DElayed]:OFFSet 2-16
 :TIMebase[:DElayed]:SCALE 2-17
 :TIMebase:FORMat 2-17
 :TRIGger:MODE 2-22
 :TRIGger<mode>:SOURce 2-22

:TRIGger<mode>:LEVel 2-23
:TRIGger<mode>:SWEep 2-23
:TRIGger<mode>:COUPling 2-24
:TRIGger:HOLDoff 2-24
:TRIGger:STATus? 2-25
:Trig%50 2-25
:TRIGger:EDGE:SLOPe 2-26
:TRIGger:EDGE:SENSitivity 2-26
:TRIGger:PULSe:MODE 2-27
:TRIGger:PULSe:SENSitivity 2-27
:TRIGger:PULSe:WIDTh 2-28
:TRIGger:VIDEO:MODE 2-29
:TRIGger:VIDEO:POLarity 2-29
:TRIGger:VIDEO:STANdard 2-30
:TRIGger:VIDEO:LINE 2-30
:TRIGger:VIDEO:SENSitivity 2-30
:TRIGger:SLOPe:TIME 2-32
:TRIGger:SLOPe:SENSitivity 2-32
:TRIGger:SLOPe:MODE 2-32
:TRIGger:SLOPe:WINDow 2-33
:TRIGger:SLOPe:LEVelA 2-34
:TRIGger:SLOPe:LEVelB
:TRIGger:PATtern:PATtern 2-35
:TRIGger:DURation:PATtern 2-36
:TRIGger:DURation:TIME 2-36
:TRIGger:DURation:QUALifier 2-37
:TRIGger:ALTerNation:SOURce 2-38
:TRIGger:ALTerNation:TYPE 2-38
:TRIGger:ALTerNation:TimeSCALe 2-38
:TRIGger:ALTerNation:TimeOFFSet 2-39
:TRIGger:ALTerNation<mode>:LEVel 2-39
:TRIGger:ALTerNation:EDGE:SLOPe 2-40
:TRIGger:ALTerNation<mode>:MODE 2-40
:TRIGger:ALTerNation<mode>:TIME 2-41
:TRIGger:ALTerNation:VIDEO:POLarity
2-41
:TRIGger:ALTerNation:VIDEO:STANdard
2-42
:TRIGger:ALTerNation:VIDEO:LINE 2-42
:TRIGger:ALTerNation:SLOPe:WINDow
2-43
:TRIGger:ALTerNation:SLOPe:LEVelA 2-43
:TRIGger:ALTerNation:SLOPe:LEVelB 2-44
:TRIGger:ALTerNation<mode>:COUPling
2-44
:TRIGger:ALTerNation<mode>:HOLDoff
2-45
:TRIGger:ALTerNation<mode>:SENSitivity
2-45

W

:WAVEform:DATA? 2-68
:WAVEform:POINts:MODE 2-68