

U9800 系列脉冲式线圈测试仪 编程手册

优高电子科技

1	命令介绍	1
1.1	符号的约定和说明	1
1.2	命令及参数的缩写	1
2	命令系统	2
2.1	公用命令	3
2.2	显示系统命令	7
2.3	脉冲电压系统命令	9
2.4	采样系统命令	11
2.5	统计系统命令	12
2.6	触发系统命令	13
2.7	比较系统命令	14
2.8	标准波形系统命令	18
2.9	多绕组系统命令(仅用于两通道和四通道仪器).....	19
2.10	测量系统命令	21
2.11	波形读取系统命令	22
2.12	FETCH 系统命令.....	24
2.13	ABORt 系统命令	25
2.14	保存/调出系统命令	26
2.15	按键系统命令	27
2.16	系统配置系统命令	28
3	出错信息	30
4	编程实例	31
4.1	Visual C++ 6.0 编程实例	33
4.2	Visual Basic 6.0 编程实例	38
4.3	LabVIEW 8.5 编程实例	41

编程手册的目的是使您利用我们现有的指令对仪器进行编程操作。主要的内容包括符号的约定和说明、命令及参数的缩写、主要命令的介绍和索引。

您可以通过这些指令控制仪器工作或者进行二次开发。

1 命令介绍

1.1 符号的约定和说明

冒号: 代表命令的层次, 表示进入命令的下一层。

问号? 表示查询命令的执行状态。

分号; 表示开始多重命令。

星号* 星号后的命令是公用命令。

逗号, 逗号是多参数的分隔符。

空格 空格是命令和参数的分隔符。

尖括号<> 尖括号包含的字符表示程序代码参数。

方括号[] 方括号表示包含的项目是可选的。

大括号{} 大括号表示当包含几个项目时, 只能从几个项目选择一个。

NR1 整数, 例如: 12。

NR2 定点数, 例如: 12.3

NR3 浮点数, 例如: 2.000000e-03。

NL 表示换行符, ASCII 码是 10, 是字符输入输出的结束符。

注: 每个命令串后面必须加上 NL(ASCII 码是 10)作为命令结束符。

1.2 命令及参数的缩写

为了方便记忆和书写长格式的命令或参数, 我们根据下面的规则, 对长格式的命令或参数进行缩写。

如果命令或参数的字符数小于等于 4 个时, 不缩写。

例如: TYPE 缩写为 TYPE。

如果命令或参数的字符数大于 4 个时, 分两种情况考虑:

1. 如果第 4 个字符是元音, 则缩写取前 3 个字符。

2. 如果第 4 个字符不是元音, 则缩写取前 4 个字符。

例如: POSition 缩写为 POS。

DISPlay 缩写为 DISP。

如果命令或参数是由两个及以上单词组成的, 先按照取用第一个单词的首字符和最后一个单词的全部的原则, 取出长格式; 然后再对长格式使用上面的缩写规则进行缩写。

例如: Impulse VOLTage 长格式为 IVOLTage, 缩写为 IVOL。

2 命令系统

U9800 系列脉冲式线圈测试仪的命令系统主要包括下列系统命令：

- ◆ 公用命令
- ◆ 显示系统命令
- ◆ 脉冲电压系统命令
- ◆ 采样系统命令
- ◆ 统计系统命令
- ◆ 触发系统命令
- ◆ 比较系统命令
- ◆ 标准波形系统命令
- ◆ 多绕组系统命令
- ◆ 测量系统命令
- ◆ 波形读取系统命令
- ◆ FETCh 系统命令
- ◆ ABORt 系统命令
- ◆ 保存/调出系统命令
- ◆ 按键系统命令
- ◆ 系统配置系统命令

2.1 公用命令

公用命令是由 IEEE488.2-1987 标准定义的，也是本仪器命令系统中最基本的命令。它既可以与其它的指令组成指令集，也可以单独完成特定的功能。

本仪器的指令系统中用到的公用指令如表 2-1-1 所示。

命令	查询	查询返回
N/A	*IDN?	Eucol Electronic Technology Co.,Ltd.,<仪器型号>,<仪器序列号>,<仪器软件版本>
*RST	N/A	N/A
*RCL <value>	N/A	N/A
*SAV <value>	N/A	N/A
*TRG	N/A	N/A
*CLS	N/A	N/A
*ESE <0-255>	*ESE?	事件状态使能寄存器
N/A	*ESR?	事件状态寄存器
*OPC	*OPC?	返回1
*SRE <0-255>	*SRE?	服务请求使能寄存器
N/A	*STB?	服务请求寄存器
N/A	*LRN?	返回仪器设置参数

表 2-1-1

1. *IDN?

返回仪器信息，包括公司名称、型号、产品序列号、软件版本号。

查询语法：*IDN?

查询返回：Eucol Electronic Technology Co.,Ltd.,<仪器型号>,<仪器序列号>,<仪器软件版本>

2. *RST (Reset)

使仪器处于稳定的状态，这里直接使仪器调用出厂设置。

命令语法：*RST

3. *RCL <value>

调出 value 指定的内部设置文件。value 范围：单通道仪器 1~300，双通道仪器 1~150，四通道仪器 1~120。

命令语法：*RCL <value>

例如：*RCL 1 调出内部存储器位置1处的设置文件

4. *SAV <value>

保存设置文件到 value 指定位置。value 范围：单通道仪器 1~300，双通道仪器 1~150，四通道仪器 1~120。

命令语法：*SAV <value>[,"name"], name是文件名称，需要使用双引号括起，最长的文件名称长度是20。

例如：*SAV 1 将设置文件保存到内部存储器位置1处

5. *TRG

产生一次触发信号。该触发仅用于测量显示页面，并且是总线触发模式。

命令语法：*TRG

6. *CLS

清除状态寄存器，并且清除输出缓冲区和清除 OPC 位。

命令语法：*CLS

7. *ESE <0-255>

设定事件状态使能寄存器。状态使能寄存器控制状态寄存器中被使能的相应位，当状态使能寄存器中的相应位置为 1 时，使能状态寄存器的相应位。

ESE（事件状态使能寄存器）

PON		CME	EXE		QYE		OPC
-----	--	-----	-----	--	-----	--	-----

事件说明

位	名称	描述	当置为1时，使能
7	PON	上电	当由断变为通时，事件发生
5	CME	命令错误	当检测到一个命令错误时，事件发生
4	EXE	执行错误	当检测到一个执行错误时，事件发生
2	QYE	输出数据丢失	输出缓冲中有数据，但是又有命令时，事件发生
0	OPC	操作完成	当一个操作完成时，事件发生

命令语法：*ESE <0-255>

查询语法：*ESE?

查询返回：<NR1><NL> 返回事件状态使能寄存器值

8. *ESR?

对应ESE中的状态位,返回所有已使能的相应位状态信息,同时清除ESR寄存器。

ESR (事件状态寄存器)

位	名称	描述	当置为 1 时,指明
7	PON	上电	已上电
5	CME	命令错误	已检测到一个命令错误
4	EXE	执行错误	已检测到一个执行错误
2	QYE	输出数据丢失	已检测到输出数据丢失
0	OPC	操作完成	操作已完成

查询语法: *ESR?

查询返回: <NR1><NL> 对应 ESE 中的状态位,返回状态信息

9. *OPC

在所有命令解析完成后,将 ESR 寄存器中的 OPC 置位。

命令语法: *OPC

查询语法: *OPC?

查询返回: <1><NL>

注: 接口必须等待*OPC?返回后才能执行下一步的操作。

10. *SRE <0-255>

设定服务请求使能寄存器。当置 1 时,使能状态字节寄存器中的相应位。

SRE (服务请求使能寄存器)

		ESB	MAV				
--	--	-----	-----	--	--	--	--

事件说明

位	名称	描述	当置为1时,使能
5	ESB	事件状态位	ESR中使能的事件产生时产生中断
4	MAV	有效信息	输出缓冲区中数据有效时产生中断

命令语法: *SRE <0-255>

查询语法: *SRE?

查询返回: <NR1><NL> 返回服务请求使能寄存器

11. *STB?

查询返回服务请求寄存器。

	RQS	ESB	MAV				
--	-----	-----	-----	--	--	--	--

事件说明

位	名称	描述	当置为1时, 指明
6	RQS	服务请求	点名中用来表明是否是服务请求源
5	ESB	事件状态位	ESR中使能的事件产生
4	MAV	有效信息	输出缓冲区中数据有效

查询语法: *STB?

查询返回: <NR1><NL> 返回服务请求寄存器值

12. *LRN?

查询返回仪器设置参数。

查询语法: *LRN?

查询返回: <data block><NL>

data block 数据格式:

1. #8xxxxxxx 后面紧跟二进制数据, 没有空格, xxxxxxxx 是数据长度。
2. \$8xxxxxxx 后面紧跟字符串数据, 没有空格, xxxxxxxx 是数据长度。

#和\$后的 8 表示后面跟着的 8 个 ASCII 码数据是随后数据的长度。

字符串格式就是将二进制数据转换成字符串数据, 每一个二进制数据转换成 2 个 ASCII 字符数据 (即一个有效数据由两个字节组成), 这两个字符分别代表二进制数据对应的十六进制数据的高 4 位和低 4 位, 其中高 4 位在先, 低 4 位在后, 例如 0x01 这个数据就转换为“0”和“1”两个字符。

注: 使用 **SYSTEM:FORMat {ASCIi | BIN}**命令设置数据返回格式。

2.2 显示系统命令

显示系统 (DISPlay) 命令主要用于设定仪器的显示页面。表 2-2-1 是 DISPlay 系统命令的命令树结构。

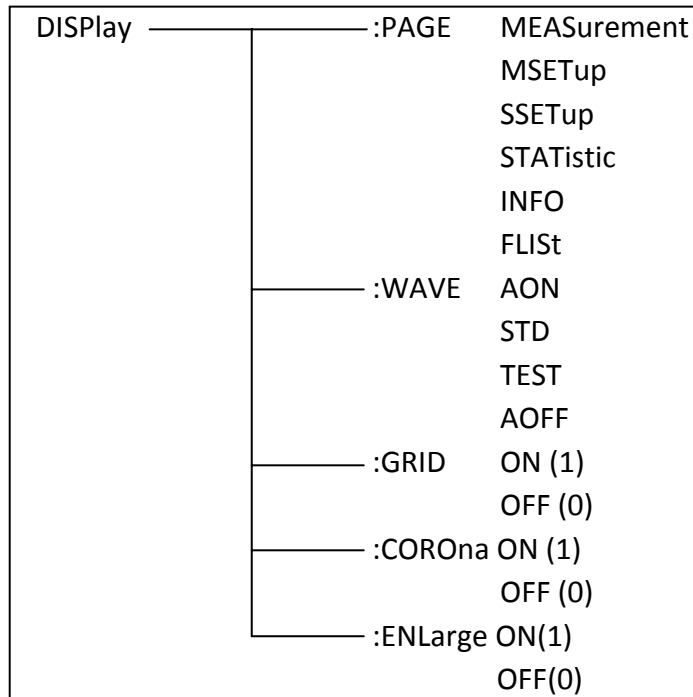


表 2-2-1 显示系统命令树

:PAGE 设定仪器的显示页面，:PAGE? 查询当前 LCD 屏上显示的页面。

命令语法: DISPlay:PAGE <page name>

<page name>具体如下:

MEASurement	设定显示页面至: 测量显示页面(MEASurement)
MSETup	设定显示页面至: 测量设置页面(Meas SETup)
SSETup	设定显示页面至: 系统设置页面(System SETup)
STATistic	设定显示页面至: 统计显示页面(STATistics)
INFO	设定显示页面至: 系统信息显示页面(INFO)
FLISt	设定显示页面至: 文件列表显示页面(File LISt)

查询语法: DISPlay:PAGE?

查询返回: {MEAS | MSET | SSET | STAT | INFO | FLIS}<NL>

Ⓛ注意: 如果正在测量之中, 则不会响应此命令, 即命令被忽略!

:WAVEform 设定波形显示模式, :WAVEform? 查询当前的波形显示模式。

命令语法: DISPlay:WAVEform {AON | STD | TSET | AOFF}

这里:

- ON 表示在屏幕上同时显示标准波形和测试波形。
- STD 表示在屏幕上仅显示标准波形。
- TSET 表示在屏幕上仅显示测试波形。
- AOFF 表示不显示任何波形。

查询语法: DISPLAY:WAVEform?

查询返回: {AON | STD | TSET | AOFF}<NL>

:GRID 设定网格显示开关, :GRID?查询当前的网格显示开关。

命令语法: DISPLAY:GRID { {1 | ON} | {0 | OFF} }

查询语法: DISPLAY:GRID?

查询返回: {{1 | ON} | {0 | OFF}}<NL>

:COROna 设定电晕显示开关, :COROna?查询当前的电晕显示开关。

命令语法: DISPLAY:COROna { {1 | ON} | {0 | OFF} }

查询语法: DISPLAY:COROna?

查询返回: {{1 | ON} | {0 | OFF}}<NL>

:ENLarge 设定波形放大显示开关, :ENLarge?查询当前的波形放大显示开关。

命令语法: DISPLAY:ENLarge { {1 | ON} | {0 | OFF} }

查询语法: DISPLAY:ENLarge?

查询返回: {{1 | ON} | {0 | OFF}}<NL>

注: 字符 **1** 和 **ON** 等价, 字符 **0** 和 **OFF** 等价。以下均同, 不再说明。

2.3 脉冲电压系统命令

Impulse VOLTage 脉冲电压系统命令主要设置与脉冲电压相关的各个参数，包括脉冲电压值，测试脉冲次数，消磁脉冲系数，脉冲电压自动调整和延时等。

表 2-3-1 是 Impulse VOLTage 系统命令树。

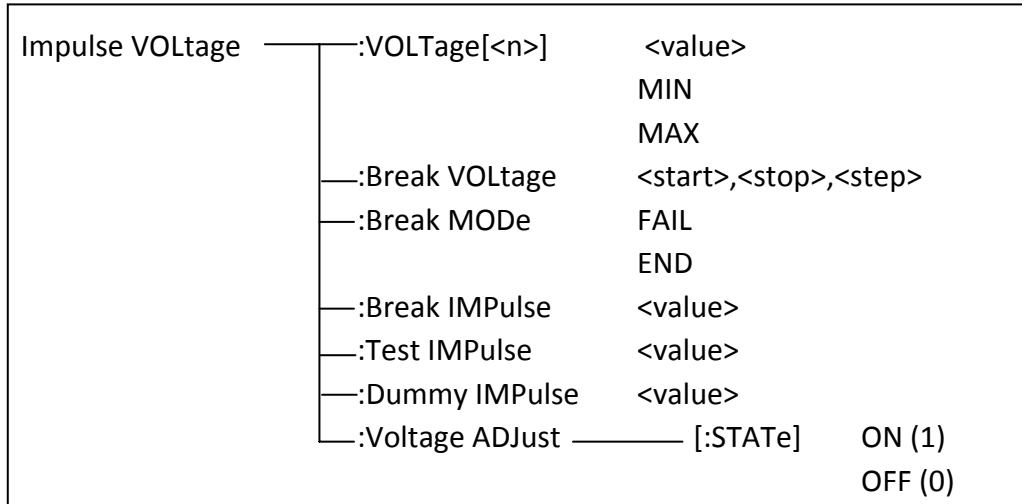


表 2-3-1 脉冲电压系统命令树

:VOLTage 设定脉冲电压值。:VOLTage? 查询仪器当前的脉冲电压值。

命令语法: IVOLTage:VOLTage {<value> | MAX | MIN}

这里: <value> 可以是 NR1,NR2 或 NR3 格式或再加 KV,V 后缀的参数, 设定的脉冲电压值应该在 100~3000V(U98x3)或者 100~5000V(U98x5)之间。

MIN 设定脉冲电压值为 100V

MAX 设定脉冲电压值为 3000V(U98x3)或者 5000V(U98x5)

查询语法: IVOLTage:VOLTage?

查询返回: <NR1><NL> 返回仪器当前的脉冲电压值。

对于多绕组仪器, 还可以使用 IVOLTage:VOLTage<n>命令来设定指定绕组的脉冲电压值。两通道仪器 n 为 1~3, 四通道仪器 n 是 1~4。在多绕组多标准测试时, 可以使用 IVOLTage:VOLTage<n>来设定多绕组的脉冲电压, 其他时候还是使用 IVOLTage:VOLTage 命令设置脉冲电压。

:Break VOLTage 设定破坏测试电压参数。:Break VOLTage?查询破坏测试电压参数。

命令语法: IVOLTage:BVOLTage <start>,<stop>,<step>

该命令分别设定破坏测试的起始电压，终止电压和步进电压，都是 NR1 格式。其中起始电压和终止电压的范围和脉冲电压范围一致，但是需要使终止电压>=起始电压。步进电压的范围是 1~50。

查询语法: IVOLTage:BVOLTage?

查询返回: <start>,<stop>,<step><NL>

:Break MODE 设定破坏测试的停止模式，可以设置停止模式为失败停止（FAIL）或者结束（END）停止。:Break MODE? 查询破坏测试停止模式。

命令语法: IVOLTage:BMODE {FAIL | END}

查询语法: IVOLTage:BMODE?

查询返回: {FAIL | END}<NL>

:Break IMPulse 设定破坏测试的脉冲次数。:Break IMPulse? 查询当前破坏测试设定的脉冲次数。有效的脉冲次数范围是 1~8。

命令语法: IVOLTage:BIMPulse {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8}

查询语法: IVOLTage:BIMPulse?

查询返回: {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8}<NL>

:Test IMPulse 设定测试脉冲次数。:Test IMPulse? 查询当前设定的测试脉冲次数。有效的测试脉冲次数范围是 1~32。

命令语法: IVOLTage:TIMPulse <value> value 范围是 1~32，NR1 格式

查询语法: IVOLTage:TIMPulse?

查询返回: <NR1><NL>

:Dummy IMPulse 设定测试脉冲次数。:Dummy IMPulse? 查询当前设定的测试脉冲次数。有效的测试脉冲次数范围是 0~8。

命令语法: IVOLTage:DIMPulse <value> value 范围是 0~8，NR1 格式

查询语法: IVOLTage:DIMPulse?

查询返回: <NR1><NL>

:Voltage ADJust 设定电压自动调整开关。:Voltage ADJust? 查询当前设定的电压自动调整开关状态。

命令语法: IVOLTage:VADJust {{1 | ON} | {0 | OFF}}

查询语法: IVOLTage:VADJust?

查询返回: {{1 | ON} | {0 | OFF}}<NL>

2.4 采样系统命令

SAMPlE 采样系统命令主要用于设置采样率, 时基和采样模式等参数。表 2-4-1 是 SAMPlE 系统命令树。

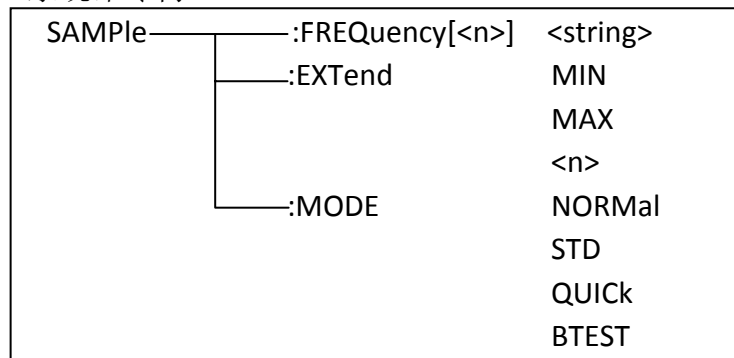


表 2-4-1 采样系统命令树

:FREQUency 用于设定仪器采样率。:FREQUency?用于查询仪器采样率。

命令语法: SAMPlE:FREQUency {100Msa/s | 50Msa/s | 25Msa/s | 10Msa/s | 5Msa/s | 2.5Msa/s | 1Msa/s | 500Ksa/s | 250Ksa/s | 250Ksa/s | 100Ksa/s}

查询语法: SAMPlE:FREQUency?

查询返回: {100Msa/s | 50Msa/s | 25Msa/s | 10Msa/s | 5Msa/s | 2.5Msa/s | 1Msa/s | 500Ksa/s | 250Ksa/s | 250Ksa/s | 100Ksa/s}<NL>

对于多绕组仪器, 还可以使用 SAMPlE:FREQUency<n>命令来设定指定绕组的采样率。两通道仪器 n 为 1~3, 四通道仪器 n 是 1~4。在多绕组多标准测试时, 可以使用 SAMPlE:FREQUency<n>来设定多绕组的采样率, 其他时候还是使用 SAMPlE:FREQUency 命令设置采样率。

提醒:

1. 被测件测试中, 如果是快速测试模式, 则修改快速测试采样率, 否则命令被忽略。
2. 标准波形测试中, 手动模式直接修改采样率, 循环模式忽略命令, 自动模式采集了标准波形时修改采样率, 否则忽略命令。

:EXTend 用于设定仪器时基缩放功能。:EXTend?查询仪器当前的时基缩放值。

命令语法: SAMPlE:EXTend {MIN | MAX | <n>}

其中 n 是 0~3, NR1 格式, MIN 表示不拉伸显示, MAX 表示最大拉伸显示。

查询语法: SAMPlE:EXTend?

查询返回: {0 | 1 | 2 | 3}<NL>

:MODE 用于设定仪器的采集模式。:MODE?用于查询当前仪器的采集模式。

命令语法: **SAMPlE:MODE {NORMAl | STD | QUICK | BTESt}**

其中 **NORMAl** 表示普通采集模式; **STD** 表示标准采集模式, 该命令使仪器进入手动标准测试模式; **QUICK** 表示快速测试模式; **BTESt** 表示破坏测试模式。

查询语法: **SAMPlE:MODE?**

查询返回: **{NORMAl | STD | QUICK | BTES}<NL>**

2.5 统计系统命令

STATistic 统计系统用于设定统计功能的打开或关闭, 以及统计数据的清除或查询。表 2-5-1 是统计系统命令树。

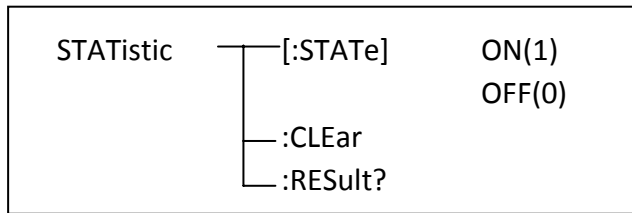


表 2-5-1 统计系统命令树

[:STATe] 用于设定仪器统计功能开关。[:STATe]?查询当前的统计功能状态。

命令语法: **STATistic[:STATe] {{1 | ON} | {0 | OFF}}**

查询语法: **STATistic[:STATe]?**

查询返回: **{{1 | ON} | {0 | OFF}}<NL>**

:CLEar 清除统计数据。

命令语法: **STATistic:CLEar**

:RESult? 查询统计结果。只返回当前绕组的统计结果, 如需其他绕组的统计结果, 先切换绕组。切换绕组指令请参考 **Multi WINDing 多绕组系统命令**。

查询返回:

<NR1>,<NR1>,<NR1>,<NR1>,<NR1>,<NR1>,<NR1>,<NR1>,<NR1>,<NR1><NL>

分别表示总的统计数, 总的合格数, 面积统计数, 面积合格数, 面积差统计数, 面积差合格数, 电晕统计数, 电晕合格数, 相位差统计数, 相位差合格数。

2.6 触发系统命令

TRIGger 触发系统命令用于设定仪器的触发模式，触发延时或触发一次测量。表 2-6-1 是 TRIGger 触发系统命令树。

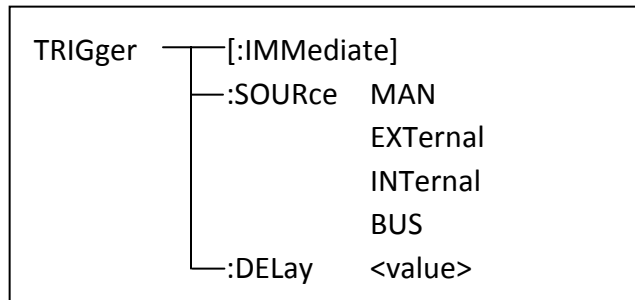


表 2-6-1 触发系统命令树

[:IMMEDIATE] 用于触发一次测量。

命令语法: TRIGger[:IMMEDIATE]

①注意: 此命令只在测量页面有效, 在其他页面该命令被忽略。正在测试中时此命令也被忽略。

:SOURce 用于设定仪器的触发模式, :SOURce?查询仪器当前的触发模式。

命令语法: TRIGger:SOURce {MANual | EXTernal | INTernal | BUS}

其中, MANual 表示手动触发, EXTernal 表示外部触发, INTernal 表示内部触发, BUS 表示总线触发。

查询语法: TRIGger:SOURce?

查询返回: {MAN | EXT | INT | BUS}<NL>

:DELay 设定内部触发模式时两次触发间的延时时间。:DELay?查询当前设定的延时时间。延时时间范围是 0~60s, 1ms 步进。

命令语法: TRIGger:DELay <value> value 是 NR3 格式

查询语法: TRIGger:DELay?

查询返回: <NR3><NL>

2.7 比较系统命令

COMParator 比较系统命令用于比较条件设置，包括面积、面积差、电晕和相位差的条件设置。表 2-7-1 是 COMParator 比较系统命令树。

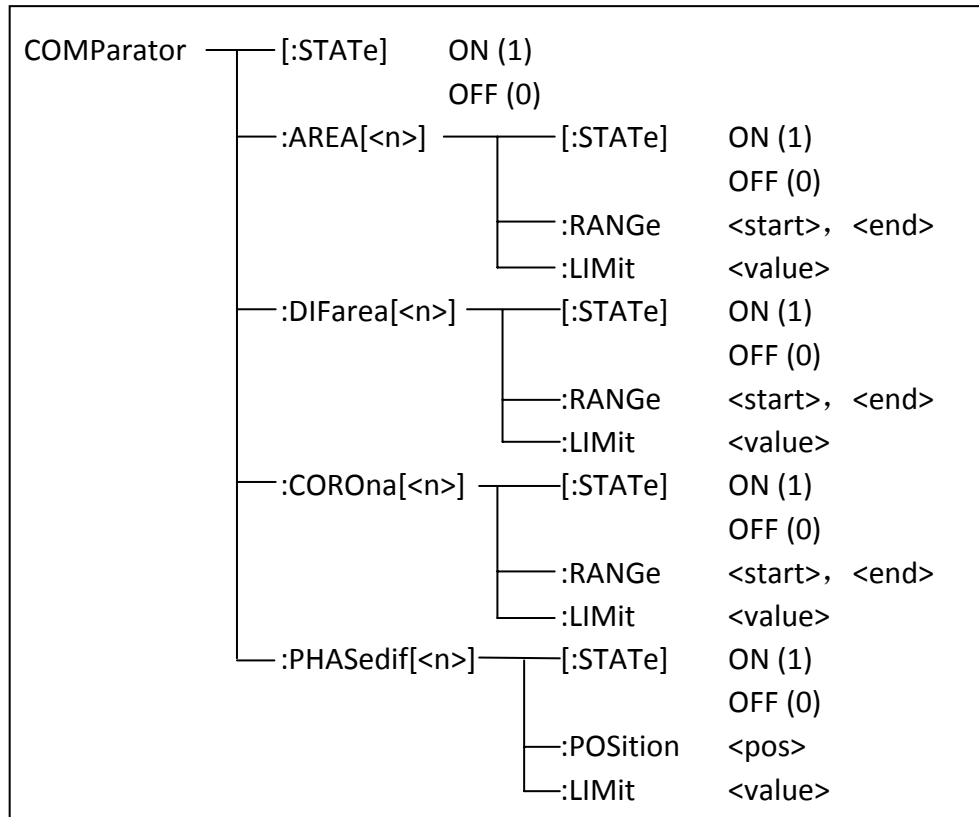


表 2-7-1 比较系统命令树

[:STATE] 用于设定仪器比较功能开关。[:STATE]? 查询当前的比较功能状态。

命令语法: COMParator[:STATE] {{1 | ON} | {0 | OFF}}

查询语法: COMParator[:STATE]?

查询返回: {{1 | ON} | {0 | OFF}}<NL>

:AREA[:STATE] 用于设定面积比较法的开关。 :AREA[:STATE]? 查询仪器当前面积比较法的开关情况。

命令语法: COMParator:AREA[:STATE] {{1 | ON} | {0 | OFF}}

查询语法: COMParator:AREA[:STATE]?

查询返回: {{1 | ON} | {0 | OFF}}<NL>

:AREA:RANGe 用于设定面积比较法的范围。 **:AREA:RANGe?** 返回仪器当前面积比较法的范围。

命令语法: **COMParator:AREA:RANGe <start>,<end>**

这里:

<start> 面积比较法的起点, NR1 数据, 范围(1~6500), 无后缀参数。

<end> 面积比较法的终点, NR1 数据, 范围(1~6500), 无后缀参数。

①注意: 终点数据不能小于起点数据, 否则会提示出错信息。

查询语法: **COMParator:AREA:RANGe?**

查询返回: **<start>,<end><NL>** start,end 是 NR1 数据格式

:AREA:LIMit 用于设定面积比较法的差值。 **:AREA:LIMit?** 返回仪器当前设置的面积比较法的差值。

命令语法: **COMParator:AREA:LIMit <value>** value 范围 0.1~99.9

这里: **<value>**可以是 NR1、NR2 或 NR3 数据格式, 无后缀参数。

①注意: <value>值仅是设定的百分号前的数据, 如设定 2.5%, 那则只需输入 2.5 即可。

查询语法: **COMParator:AREA:LIMit?**

查询返回: **<NR3><NL>**

:DIFarea[:STATe] 用于设定面积差比较法的开关。 **:DIFarea[:STATe]?** 查询仪器当前面积差比较法的开关情况。

命令语法: **COMParator:DIFarea[:STATe] {{1 | ON} | {0 | OFF}}**

查询语法: **COMParator:DIFarea[:STATe]?**

查询返回: **{{1 | ON} | {0 | OFF}}<NL>**

:DIFarea:RANGe 用于设定面积差比较法的范围。 **:DIFarea:RANGe?** 返回仪器当前面积差比较法的范围。

命令语法: **COMParator:DIFarea:RANGe <start>,<end>**

这里:

<start> 面积差比较法的起点, NR1 数据, 范围(1~6500), 无后缀参数。

<end> 面积差比较法的终点, NR1 数据, 范围(1~6500), 无后缀参数。

①注意: 终点数据不能小于起点数据, 否则会提示出错信息。

查询语法: **COMParator:DIFarea:RANGe?**

查询返回: **<start>,<end><NL>** start,end 是 NR1 数据格式

:DIFarea:LIMit 用于设定面积差比较法的差值。 **:DIFarea:LIMit?** 返回仪器当前设置的面积差比较法的差值。

命令语法: **COMParator:DIFarea:LIMit <value>** value 范围 0.1~99.9

这里: <value>可以是 NR1、NR2 或 NR3 数据格式, 无后缀参数。

①注意: <value>值仅是设定的百分号前的数据, 如设定 2.5%, 那则只需输入 2.5 即可。

查询语法: **COMParator:DIFarea:LIMit?**

查询返回: <NR3><NL>

:COROna[:STATe] 用于设定电晕比较法的开关。 **:COROna[:STATe]?** 查询仪器当前电晕比较法的开关情况。

命令语法: **COMParator:COROna[:STATe] {{1 | ON} | {0 | OFF}}**

查询语法: **COMParator:COROna[:STATe]?**

查询返回: <NR1><NL>

:COROna:RANGe 用于设定电晕比较法的范围。 **:COROna:RANGe?** 返回仪器当前电晕比较法的范围。

命令语法: **COMParator:COROna:RANGe <start>, <end>**

这里:

<start> 面积比较法的起点, NR1 数据, 范围(1~6500), 无后缀参数。

<end> 面积比较法的终点, NR1 数据, 范围(1~6500), 无后缀参数。

①注意: 终点数据不能小于起点数据, 否则会提示出错信息。

查询语法: **COMParator:COROna:RANGe?**

查询返回: <start>,<end><NL> start,end 是 NR1 数据格式。

:COROna:LIMit 用于设定电晕比较法的差值。 **:COROna:LIMit?** 返回仪器当前设置的电晕比较法的差值。

命令语法: **COMParator:COROna:LIMit <value>**

这里: <value> 是 NR1 数据格式, 范围(1~255), 无后缀参数。

查询语法: **COMParator:COROna:LIMit?**

查询返回: <NR1><NL>

:PHASedif[:STATe] 用于设定相位差比较法的开关。 **:PHASedif[:STATe]?** 查询仪器当前相位差比较法的开关情况。

命令语法: **COMParator:PHASedif[:STATe] {{1 | ON} | {0 | OFF}}**

查询语法: COMParator:PHASediff[:STATe]?

查询返回: {{1 | ON} | {0 | OFF}}<NL>

:PHASedif:POSition 用于设定使用波形的哪一个过零点计算相位差。

:PHASedif:POSition? 查询仪器当前设定的过零点位置。

命令语法: COMParator:PHASediff:POSition <value>

这里:<value> 是过零点位置, NR1 数据格式, 范围(2~20), 无后缀参数。

查询语法: COMParator:PHASediff:POSition?

查询返回: <NR1><NL>

:PHASedif:LIMit 用于设定相位差比较法的差值。 :PHASedif:LIMit? 查询仪器当前设置的相位差比较法的差值。

命令语法: COMParator:PHASedif:LIMit <value>

这里: <value>可以是 NR1、NR2 或 NR3 数据格式, 无后缀参数。

①注意: <value>值仅是设定的百分号前的数据, 如设定 2.5%, 那则只需输入 2.5 即可。

查询语法: COMParator:PHASedif:LIMit?

查询返回: <NR3><NL>

对于多绕组仪器, 还可以使用 COMParator:AREA<n>, COMParator:DIFarea<n>, COMParator:COROna<n>和 COMParator:PHASedif<n>命令来设定指定绕组的比较参数。两通道仪器 n 为 1~3, 四通道仪器 n 是 1~4。在多绕组多标准测试时, 可以使用这些命令来设定多绕组的比较参数, 其他时候还是使用不带<n>的命令设置比较参数。

2.8 标准波形系统命令

Standard WAVEform 标准波形系统命令主要用于加载标准波形数据。表 2-8-1 是 Standard WAVEform 标准波形系统命令树。

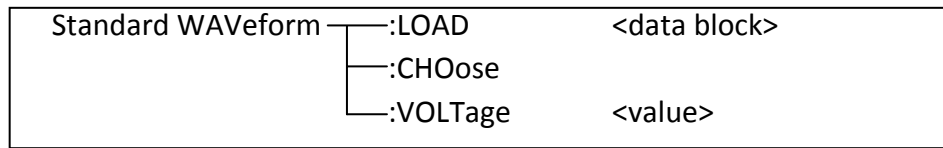


表 2-8-1 标准波形系统命令树

:LOAD 用于从上位机加载标准波形数据到仪器。

命令语法: SWAVEform:LOAD <data block>

data block 数据格式:

1. #8xxxxxxx 后面紧跟二进制数据, 没有空格, xxxxxxxx 是数据长度。
2. \$8xxxxxxx 后面紧跟字符串数据, 没有空格, xxxxxxxx 是数据长度。

#和\$后的 8 表示后面的 8 个 ASCII 码数据是随后波形数据的点数。

字符串格式就是将二进制数据转换成字符串数据, 每一个二进制数据转换成 2 个 ASCII 字符数据 (即一个波形点由两个字节组成), 这两个字符分别代表二进制数据对应的十六进制数据的高 4 位和低 4 位, 其中高 4 位在先, 低 4 位在后, 例如 0x01 这个数据就转换为“0”和“1”两个字符。

注: data block 数据使用 WAVEform 系统命令获取。

使用 WAVEform:FORMat {AScii | BIN}命令设置数据读取格式。

:CHOose 用于在标准波形测量过程中选择所需要的标准波形。

命令语法: SWAVEform:CHOose

①注意:

1. 该命令只在测量页面有效, 在其他页面上命令被忽略。
2. 该命令只有在标准波形采集时有效, 其他时候忽略。
3. 标准采集时, 只要采集到了波形就可以使用该命令完成标准采集。

:VOLTage 命令设定对应于加载标准波形的脉冲电压控制值。由于标准采集可以使用自动调整操作, 只要自动调整打开, 那么实际产生的脉冲电压值和设定的脉冲电压值就会不同, 因此获取标准时需要同时获取该电压值, 同理加载标准波形时一起加载该电压值。

命令语法: SWAVEform:VOLTage <volt>

volt 是 NR1 格式, 可以带 kV 和 V 单位。

查询语法: SWAVEform:VOLTage?

查询返回: <NR1><NL> 如果不存在标准, 则返回当前设定的脉冲电压值。

2.9 多绕组系统命令(仅用于两通道和四通道仪器)

Multi WINDing 多绕组系统命令用于设置多通道仪器的测试线圈类型,工作模式,标准模式和测试线圈等参数。表 2-9-1 是多绕组系统命令树。

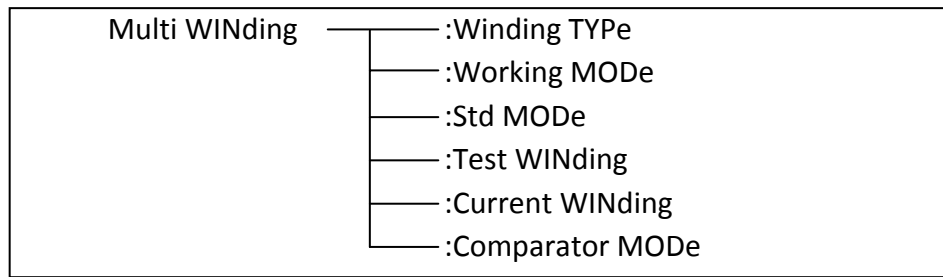


表 2-9-1 多绕组系统命令树

:Winding TYPE 用于设定测试线圈的类型。:Winding TYPE?用于查询当前设定的测试线圈类型。

命令语法: MWINDing:WTYPE {1COIL | 2COIL | 3COIL | 4COIL | 3P3W | 3P4W}

其中, 1COIL 表示单绕组, 2COIL 表示两绕组, 3COIL 表示三绕组, 4COIL 表示四绕组, 3P3W 表示三相三线, 3P4W 表示三相四线。

注: 两通道仪器只能设置 1COILE 和 2COILE。

查询语法: MWINDing:WTYPE?

查询返回: {1COIL | 2COIL | 3COIL | 4COIL | 3P3W | 3P4W}<NL>

:Working MODE 用于设定仪器工作模式。:Working MODE?查询当前工作模式。

命令语法: MWINDing:WMODE {NORMal | BALance}

其中, NORMal 表示普通工作模式, BALance 表示对比测试模式。

查询语法: MWINDing:WMODE?

查询返回: {NORM | BAL}<NL>

:Std MODE 用于设定标准模式。:Std MODE?用于查询当前的标准模式。

命令语法: MWINDing:SMODE {Single STD | Multi STD}

其中, Single STD 表示单标准模式, Multi STD 表示多标准模式。

查询语法: MWINDing:SMODE?

查询返回: {SSTD | MSTD}<NL>

:Test WINDing 用于设定测试线圈。:Test WINDing?用于查询当前的测试线圈。

命令语法: MWINDing:TWINDing <string>

查询语法: MWINDing:TWINDing?

查询返回: <string><NL>

设定和查询中的<string>如下:

对比测试: 2COIL: AG-BG, CG-DG, AB-CD (后两个只用于四通道仪器)

3COIL: AG-BG-CG, AB-BC-AC

4COIL: AG-BG-CG-DG

三相三线: AB-BC-AC

三相四线: AG-BG-CG

普通测试: 单绕组: AG, BG, CG, DG (后两个只用于两通道仪器)

两绕组: AG-BG, CG-DG, AB-CD (后两个只用于四通道仪器)

三绕组: AG-BG-CG, AB-BC-AC

四绕组: AG-BG-CG-DG

三相三线: AB-BC-AC

三相四线: AG-BG-CG

:Current WINDing 用于切换当前显示绕组。:Current WINDing?查询当前绕组。

命令语法: MWINDing:CWINDing <string>

查询语法: MWINDing:CWINDing?

查询返回: <string><NL>

设定和查询中的<string>如下:

对比测试: 2COIL: AG-BG, CG-DG, AB-CD (后两个只用于四 HV 端子仪器)

3COIL: AG-BG, BG-CG, CG -AG / AB-BC, BC-AC, AC -AB

4COIL: AG-BG, BG-CG, CG-DG, DG -AG

三相三线: AB-BC, BC-AC, AC -AB

三相四线: AG-BG, BG-CG, CG -AG

普通测试: 单绕组: AG/BG/CG/DG

两绕组: AG, BG / CG, DG / AB, CD

三绕组: AG, BG, CG / AB, BC, AC

四绕组: AG, BG, CG, DG

三相三线: AB, BC, AC

三相四线: AG, BG, CG

:Comparator MODE 用于设定比较模式。:Comparator MODE 用于查询比较模式。

命令语法: MWINDing:CMODE {PUBLic | PRIVate}

其中, PUBLic 表示公用标记比较模式, PRIVate 表示独立比较模式。比较模式用于多绕组和多标准测试, 这时可以为每个绕组设定不同的比较参数。

查询语法: MWINDing:CMODE?

查询返回: {PUBL | PRIV}<NL>

2.10 测量系统命令

MEASure 测量系统命令用于设定电压、频率和时间测量的范围。图 2-10-1 是 MEASure 测量系统命令树。

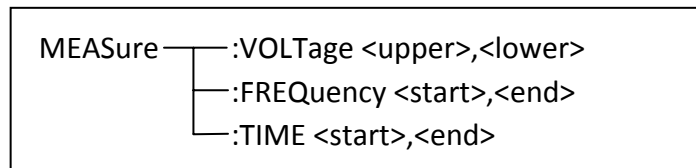


表 2-10-1 测量系统命令树

:VOLTage 用于设定电压测量范围。:VOLTage? 返回仪器当前电压测量范围。

命令语法: MEASure:VOLTage <upper>,<lower>

这里: upper 和 lower 是 NR1 数据格式, 无后缀参数, 范围是-120~120。

Ⓛ注意: 下限数据不能小于上限数据, 否则会提示出错信息。

查询语法: MEASure:VOLTage?

查询返回: <upper>,<lower><NL> upper 和 lower 是 NR1 格式

:FREQuency 用于设定频率测量范围。:FREQuency?返回当前频率测量范围。

命令语法: MEASure:FREQuency <start>,<end>

这里: start 和 end 是 NR1 数据格式, 无后缀参数, 范围是 1~650。

Ⓛ注意: 终点数据不能小于起点数据, 否则会提示出错信息。

☞提醒: 频率测量和时间测量的范围是一样的, 所以改变了频率测量的范围也就改变了时 间测量的范围。

查询语法: MEASure:FREQuency?

查询返回: <start>,<end><NL> start 和 end 是 NR1 数据格式。

:TIME 用于设定时间测量范围。:TIME? 返回当前设定的时间测量范围。

命令语法: MEASure:TIME <star>,<end>

这里: start 和 end 是 NR1 数据格式, 无后缀参数, 范围是 1~650。

Ⓛ注意: 终点数据不能小于起点数据, 否则会提示出错信息。

☞提醒: 时间测量和频率测量的范围是一样的, 所以改变了时间测量的范围也就改变了频 率测量的范围。

查询语法: MEASure:TIME?

查询返回: <start>,<end><NL> start 和 end 是 NR1 数据格式。

2.11 波形读取系统命令

WAVeform 波形读取系统命令用于读取测试波形和标准波形数据以及相关参数。表 2-11-1 是波形读取系统命令树。

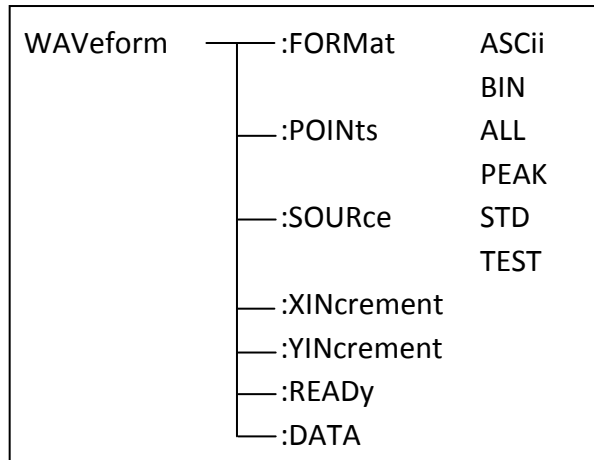


表 2-11-1 波形读取系统命令树

:FORMat 用于设置返回波形数据的格式。:FORMat?查询当前设定的波形数据的返回格式。AScii 表示字符串格式，BIN 表示二进制串格式。

命令语法: WAVeform:FORMat {AScii | BIN}

查询语法: WAVeform:FORMat?

查询返回: {ASC | BIN}<NL>

:POINts 用于设定返回数据的长度。:POINts?用于查询返回数据的长度。

命令语法: WAVeform:POINts {ALL | PEAK}

其中，ALL 表示返回所有数据，即 6500 点，PEAK 返回峰值包络数据，即 650 点。实际返回的点数的 6520 点和 652 点。

查询语法: WAVeform:POINts?

查询返回: {ALL | PEAK}<NL>

:SOURce 用于设定 WAVeform:DATA?返回波形数据的源。:SOURce?查询当前返回波形数据的源。

命令语法: WAVeform:SOURce {STD | TEST}

其中，STD 表示返回标准波形数据，TEST 表示返回测试波形数据。

查询语法: WAVeform:SOURce?

查询返回: {STD | TEST}<NL>

:XINcrement?用于查询返回数据点与点之间的时间间隔值。

查询语法: `WAVeform:XINcrement?`

查询返回: `<NR3><NL>`

:YINcrement?用于查询返回波形垂直方向电压间隔值，即垂直档位值/25。

查询语法: `WAVeform:YINcrement?`

查询返回: `<NR3><NL>`

:READY?用于查询是否可以读取波形数据。只要不是测试中，或者已经采集了数据，就会返回 1；测试中时，总是返回 0，所以可以在一次采集开始后使用 `WAVeform:READY?`来判别是否可以读取数据。

查询语法: `WAVeform:READY?`

查询返回: `{0 | 1}<NL>`

:DATA?用于读取波形数据。

返回的是标准数据还是测试数据则由 `WAVeform:SOURce` 命令设置。

返回波形数据的格式由 `WAVeform:FORMat` 命令设置。

查询语法: `WAVeform:DATA?`

查询返回: `<data block><NL>`

data block 数据格式:

AScii 格式: `#8xxxxxxxx` 后面紧跟二进制数据，没有空格，`xxxxxxxx` 是数据长度。

BIN 格式: `$8xxxxxxxx` 后面紧跟字符串数据，没有空格，`xxxxxxxx` 是数据长度。

#和\$后的 8 表示后面的 8 个 ASCII 码数据是随后波形数据的点数。

字符串格式就是将二进制数据转换成字符串数据，每一个二进制数据转换成 2 个 ASCII 字符数据（即即一个波形点由两个字节组成），这两个字符分别代表二进制数据对应的十六进制数据的高 4 位和低 4 位，其中高 4 位在先，低 4 位在后，例如 `0x01` 这个数据就转换为“0”和“1”两个字符。

返回测试波形：破坏测试返回破坏测试波形；快速测试返回快速测试波形；普通测试返回当前线圈测试波形；对比测试返回的是线圈对中的第一个线圈测试波形。

返回标准波形：标准测试菜单时，返回采集的标准波形数据；对比测试时返回线圈对中的第二个线圈测试波形；单标准时返回标准数据；多标准时返回当前线圈的标准数据。

多通道仪器需要返回其他绕组的波形数据，需要先切换当前绕组，请参考

MWINDing:CWINDing 命令。

2.12 FETCh 系统命令

FETCh 系统命令主要用于输出波形数据和比较结果数据,还包括了输出电压,频率和时间测量结果。表 2-12-1 是 FETCh 系统命令树。

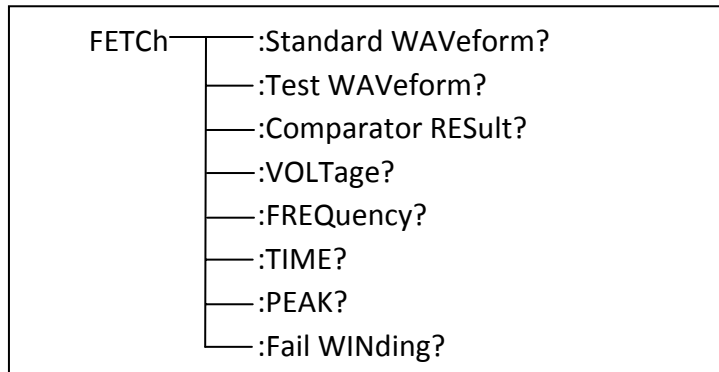


表 2-12-1 FETCh 系统命令树

:Standard WAVEform?命令用于输出仪器当前的标准波形数据。

查询语法: FETCh SWAVEform?

该命令等同于 WAVEform:DATA?(数据源是标准波形)。

:Test WAVEform?命令用于输出仪器最近一次的测试波形数据。

查询语法: FETCh TWAVEform?

该命令等同于 WAVEform:DATA?(数据源是测试波形)。

:Comparator RESult?用于返回比较结果。

查询语法: FETCh CREsult?

查询返回:

1. 单通道仪器: <NR1>,<NR3>,<NR3>,<NR1>,<NR3>, 第一个 NR1 表示总的判别结果, 1 表示通过, 0 表示失败, 2 表示没有比较(没有测试波形或者比较器未打开等), 2 时后面的数据不再返回; 后面四个数据表示每个比较项的比较结果, 如果该比较项未做比较, 则返回+9.90000E+37, 电晕则是返回 9999。
2. 多绕组仪器返回所有测试线圈的比较结果: 格式同单绕组, 第一个数据是总的判别结果, 然后跟着每个线圈的四个比较项的结果。

:VOLTage?输出仪器电压测量范围内的电压结果, 范围设定参考 MEASure 系统命令。

查询语法: FETCh:VOLTage?

查询返回: <NR1><NL> ⚡提醒: 返回的电压值以 V 为单位。

:FREQuency? 输出仪器频率测量范围内的频率结果, 范围设定参考 MEASure 系统命令。

查询语法: FETCh:FREQuency?

查询返回: <NR3><NL>

⚡提醒: 返回的频率值以 Hz 为单位。如果设定的范围为零, 即起点和终点重合, 那么将返回+9.90000E+37。

:TIME? 输出仪器时间测量范围内的时间结果, 范围设定参考 MEASure 系统命令。

查询语法: FETCh:TIME?

查询返回: <NR3><NL>

⚡提醒: 返回的时间值以 s 为单位。

:PEAK? 用于查询波形电压峰值数据。

查询语法: FETCh:PEAK?

查询返回: <NR3><NL>

⚡提醒: 存在测试波形, 返回测试波形的峰值, 否则返回标准波形的峰值, 如果都不存在就返回+9.90000E+37。

:Fail WINDing? 用于查询判别比较失败的绕组名称。(仅用于多绕组仪器)

查询语法: FETCh:FWINDing?

查询返回: <string><NL>

⚡提醒: 没有采集波形或者没有打开比较时, 返回“OFF”; 所有绕组都判别通过时返回“PASS”; 有绕组判别失败时, 则返回“FAIL,” , 然后就是失败绕组的名称, 多个失败绕组的名称使用, 分割。

2.13 ABORt 系统命令

ABORt 系统命令用于放弃当前正在进行的一次测量。

命令语法: ABORt

2.14 保存/调出系统命令

Mass MEMory 保存/调出系统命令用于文件的保存与加载。表 2-14-1 是 Mass MEMory 系统命令树。

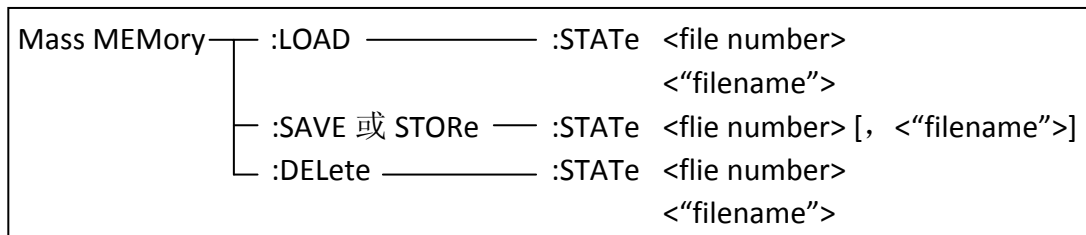


表 2-14-1 保存/调出系统命令树

①注意:在测试过程中, Mass MEMory 系统命令被忽略。

:LOAD:STATe 命令用于加载已保存的文件。

命令语法: MMEMory:LOAD:STATe <flie number>

<file number>是文件序号, 范围是 1~720 (四通道仪器) /1~750 (两通道仪器) /1~900 (单通道仪器), NR1 数据格式, 无后缀参数。

- ①注意: 1. 如果加载的文件不存在, 仪器将显示“没有数据加载”报警信息。
2. 如果指定的文件号超出范围, 仪器将显示“数据超出范围”的报警信息。此项同样适用于下面两个命令。

命令语法: MMEMory:LOAD:STATe <“filename”>

该命令直接通过输入的文件名“filename”来查找文件并进行加载。

①注意: 使用文件名加载文件仅限于内部文件。

:SAVE:STATe 或 **STORE:STATe** 命令用于保存当前仪器的设置到一个文件。

命令语法: MMEMory:STORE:STATe <flie number> [, <“filename”>]

<file number> 是文件序号, 范围是 1~720 (四通道仪器) /1~750 (两通道仪器) /1~900 (单通道仪器), NR1 数据格式, 无后缀参数。

<“filename”> 是要保存的文件名, 可用 20 个以内的 ASCII 字符表示, 注意要用引号将文件名括起来。如果未指定文件名, 仪器以缺省的<Unnamed>命名。

①注意: 本仪器在覆盖已存在的文件记录时并不会提示。

☞提醒: 总线指定的文件名总是被原样引用, 因此通过总线可以得到一些仪器面板上不能输入的字符, 如一些特殊符号等。

:DElete:STATe 命令用于删除一个文件。

命令语法: **MMEMory:DELeTe:STATe** <file number>

<file number>是文件序号, 范围是 1~720 (四通道仪器) /1~750 (两通道仪器) /1~900 (单通道仪器), NR1 数据格式, 无后缀参数。

①**注意:** 仪器删除存在的文件记录时并不会提示。

命令语法: **Mass MEMory:DELeTe:STATe** "filename"

按照文件名删除文件, "filename" 是要删除文件名称, 可用 20 个以内的 ASCII 字符表示, 注意要用引号将文件名括起来。

①**注意:** 使用文件名删除文件仅限于内部文件。

2.15 按键系统命令

KEY 按键系统命令等同于前面板上的按键和旋钮。

KEY:LOCAl	释放按键操作,退出远控。
KEY:MEASure	测量显示按键
KEY:SETUp	测量设置按键
KEY:SYSTem	系统设置按键
KEY:FILE	文件按键
KEY:UPPer	上按键
KEY:DOWN	下按键
KEY:LEFT	左按键
KEY:RIGHT	右按键
KEY:ADD	旋钮正转
KEY:SUB	旋钮反转
KEY:NUM<n>	数字键, n 为 0~9
KEY:DOT	小数点按键
KEY:SIGN	负号按键
KEY:ENTer	回车键
KEY:BACKsapce	退格键
KEY:ESC	取消键
KEY:START	启动按键
KEY:STOP	停止按键
KEY:SAVE	保存按键
KEY:F<n>	软键, n 为 1~6

2.16 系统配置系统命令

SYSTem 系统配置系统命令用于设置实时时钟，保存类型，上传/下载设置文件等。表 2-16-1 是系统配置系统命令树。

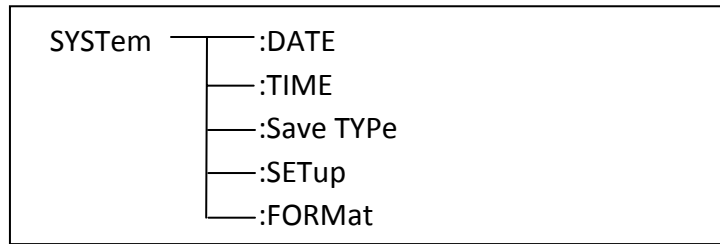


表 2-12-1 系统配置系统命令树

:DATE 用于设定年月日。:DATE?查询当前年月日。

命令语法: SYSTem:DATE <year>,<month>,<day> year,month,day 是 NR1 格式。

其中 month 也可以是字符串格式: {JANuary | FEBruary | MARch | APRil | MAY | JUNE | JULy | AUGust | SEPtember | OCTober | NOVember | DECember}

查询语法: SYSTem:DATE?

查询返回: <NR1>,<NR1>,<NR1><NL>

:TIME 设定时分秒。:TIME?查询时分秒。

命令语法: SYSTem:TIME <hour>,<minute>,<second> hour, minute, second 是 NR1 格式。

查询语法: SYSTem:TIME?

查询返回: <NR1>,<NR1>,<NR1><NL>

:Save TYPE 用于设定 **SAVE** 按键保存文件类型。:Save TYPE?查询保存文件类型。

命令语法: SYSTem:STYPE {CSV | GIF | BMP8 | BMP24 | PNG}

查询语法: SYSTem:STYPE?

查询返回: {CSV | GIF | BMP8 | BMP24 | PNG}<NL>

:FORMat 用于设定设置数据返回的格式。:FORMat?查询返回设置数据的格式。

命令语法: SYSTem:FORMat {ASCii | BIN}

其中, ASCii 表示字符串格式, BIN 表示二进制格式。

查询语法: SYSTem:FORMat?

查询返回: {ASCii | BIN}<NL>

:SETup 用于下载设置参数（不含标准波形数据）。**:SETup?**查询返回设置参数（不含标准波形数据）。

命令语法: **SYSTem:SETup <data block>**

其中 **data block** 是由 **SYSTem:SETup?**读取的设置参数。

查询语法: **SYSTem:SETup?** 该命令等同于 ***LRN?**

查询返回: **<data block><NL>**

data block 数据格式:

1. **#8xxxxxxxx** 后面紧跟二进制数据，没有空格，**xxxxxxxx** 是数据长度。

2. **\$8xxxxxxxx** 后面紧跟字符串数据，没有空格，**xxxxxxxx** 是数据长度。

#和**\$**后的 **8** 表示后面的 **8** 个 **ASCII** 码数据是随后数据的长度。

字符串格式就是将二进制数据转换成字符串数据，每一个二进制数据转换成 **2** 个 **ASCII** 字符数据（即一个有效数据由两个字节组成），这两个字符分别代表二进制数据对应的十六进制数据的高 **4** 位和低 **4** 位，其中高 **4** 位在先，低 **4** 位在后，例如 **0x01** 这个数据就转换为“**0**”和“**1**”两个字符。

注：使用 **SYSTem:FORMat {ASCIi | BIN}**命令设置数据返回格式。

3 出错信息

发给仪器的总线命令中，可能包含错误的命令或语法，或者不正确的参数。本仪器对命令串边分析边执行，如果遇到错误，则显示出错信息并中止命令分析，因此，在一个命令串中，错误之后的内容将被仪器忽略。如果提示的消息是命令忽略或者触发忽略等，那么仪器还是会继续执行分析程序。

下表是总线上常见的错误消息，出错消息将在信息提示区域显示。

错误信息	说明
Undefined message	未知消息，既无法识别的消息。 例如: TRG 应该为 TRIG DISP:PAG MEAS 应该为 DISP:PAGE MEAS
Data out of range	数据超限 例如: IVOLT 5500, 脉冲电压值超出范围
Invalid parameter	无法识别的命令参数。 例如: TRIG:SOUR INTER, INTER 为不支持的触发模式
Invalid suffix	后缀错误，一般是不匹配的单位。 例如:TRIG:DEL 200us, us 是电压不支持的单位。
Data too long	数据太长，例如文件名超过 20 个字符
Syntax error	系统错误，如 DISP 命令后面应该是冒号(:)，如果不是就是系统错误。
Trigger ignored	触发忽略。在测试过程中的触发信息都将被忽略。
Command ignored	命令忽略。例如在测试进行过程中，DISP:PAGE MSET 命令被忽略。

4 编程实例

本章给出了基于VISA (Virtual Instrument Software Architecture)库,在Visual C++ 6.0、 Visual Basic 6.0和LabVIEW 8.5软件开发环境中,通过USB接口传输命令实现仪器基本功能的编程实例,分别实现发送命令和读取返回数据。

VISA是一个用来与各种仪器总线进行通讯的高级应用编程接口(API)。在这里我们所说的VISA是指NI-VISA。NI-VISA是美国国家仪器有限公司根据VISA标准编写的应用程序接口。您可以通过使用NI-VISA,在GPIB、RS232、USB 和LAN等接口上,实现仪器与PC的通信。用户无需了解接口总线如何工作,就可以对仪器进行控制。如果您要了解有关NI-VISA API的具体信息,请参考《NI-VISA 用户手册》与《NI-VISA程序员参考手册》。

一个典型的VISA应用分为以下几个步骤:

- (1) 为已有的资源建立会话
- (2) 对资源做一些配置(如波特率)
- (3) 关闭会话资源

编程前的准备工作

如果没有安装NI-VISA，请先到NI的网站(<http://www.ni.com/china>)下载。我们采用的安装路径是C:\Program Files\IVI Foundation\VISA。

用USB数据线，一端连接仪器后面板上的USB DEVICE接口，一端连接PC机的USB接口。

然后,打开仪器的电源开关，PC机弹出硬件更新向导对话框，请根据安装向导的提示安装“USB Test and Measurement Device”软件。如图4-1所示。

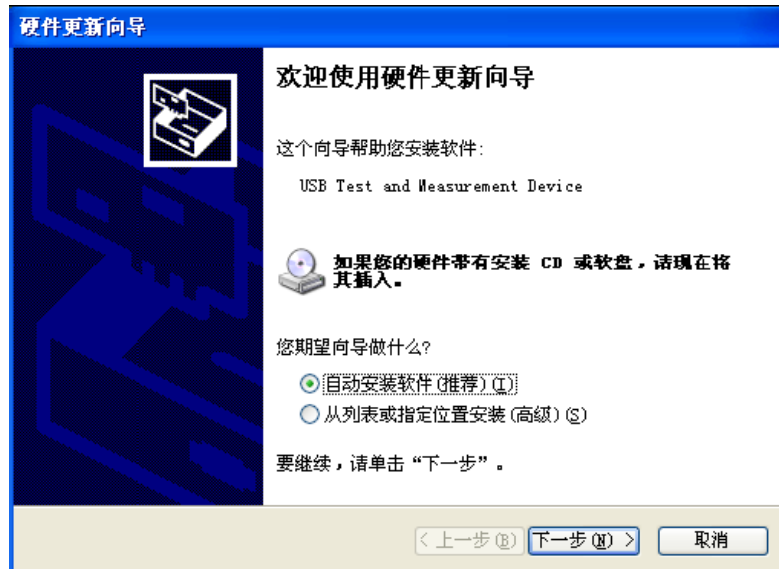


图 4-1

4.1 Visual C++ 6.0 编程实例

请运行 Visual C++ 6.0 软件，并按如下步骤进行操作：

1. 创建一个基于 MFC 的工程。
2. 打开Project->Settings中的[C/C++]选项卡，在Category框中选择Code Generation，在Use run-time library框中选择Debug Multithreaded DLL。点OK关闭对话框。如图4-1-1所示。

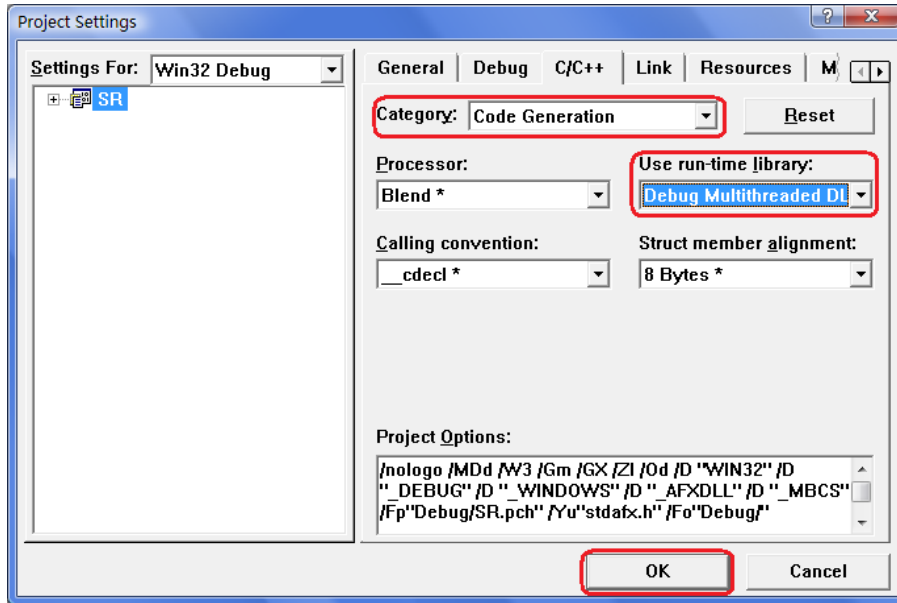


图 4-1-1

3. 打开Project->Settings中的[Link]选项卡，在Object/library modules框中手动添加visa32.lib。点OK关闭对话框。如图4-1-2所示。
4. 打开Tools->Options 中的[Directories]选项卡。

在Show directories for框中选择Include files，双击Directories框中的空白处添加Include的路径：C:\Program Files\IVI Foundation\VISA\WinNT\include，如图4-1-3所示；

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

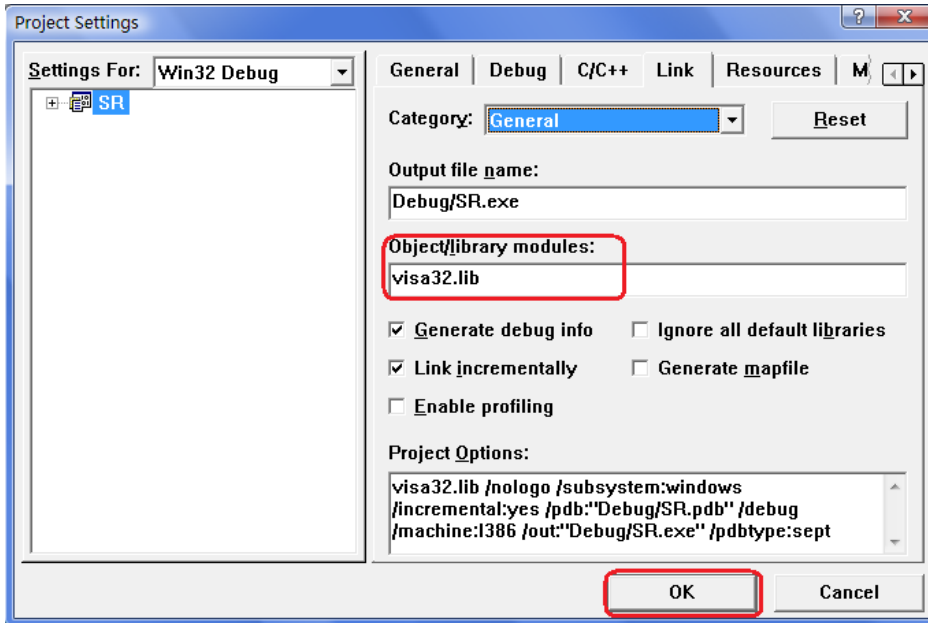


图 4-1-2

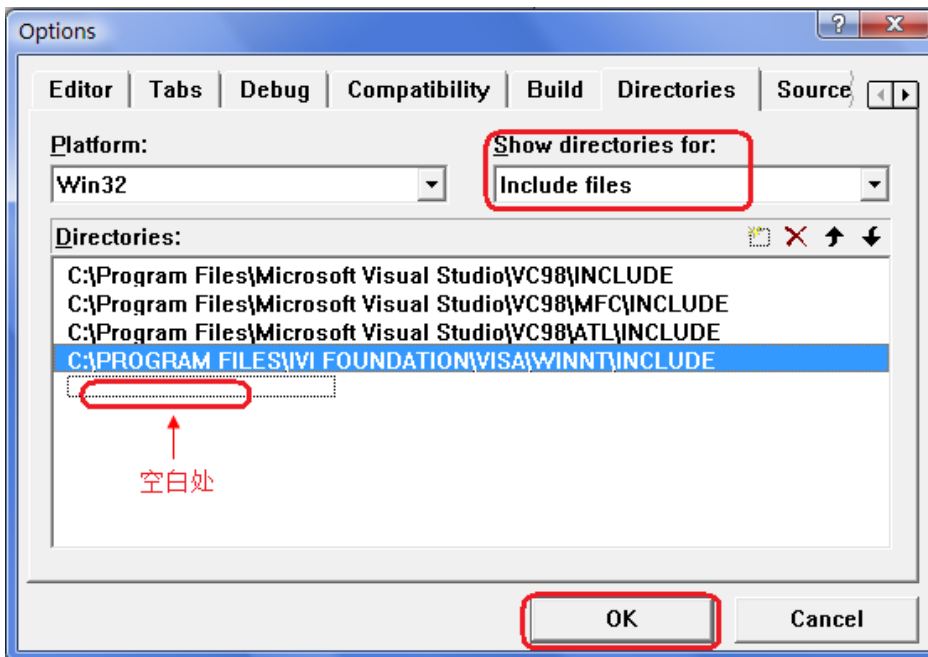


图 4-1-3

5. 添加控件 Static Text、Edit 和 Button。如图 4-1-4 所示。

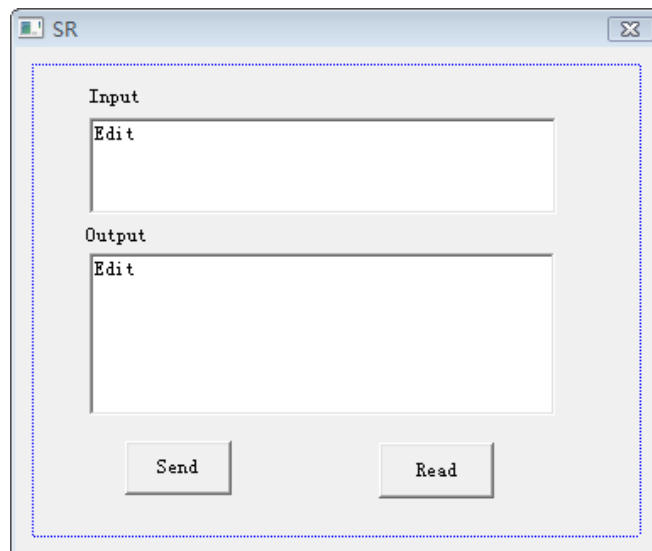


图 4-1-4

(1) 添加两个 Static Text 控件，分别命名为 Input 和 Output。

(2) 添加两个 Edit 控件，分别添加变量 m_send 和 m_read。如图 4-1-5 和 4-1-6 所示。

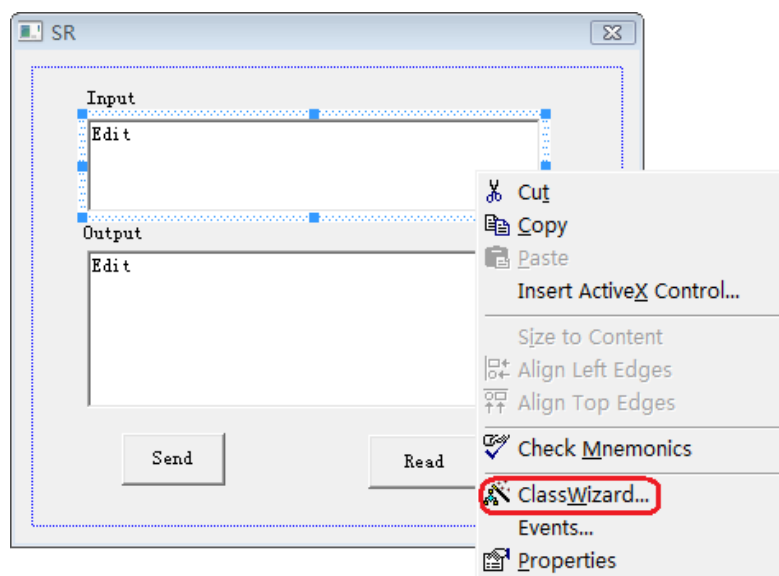


图 4-1-5

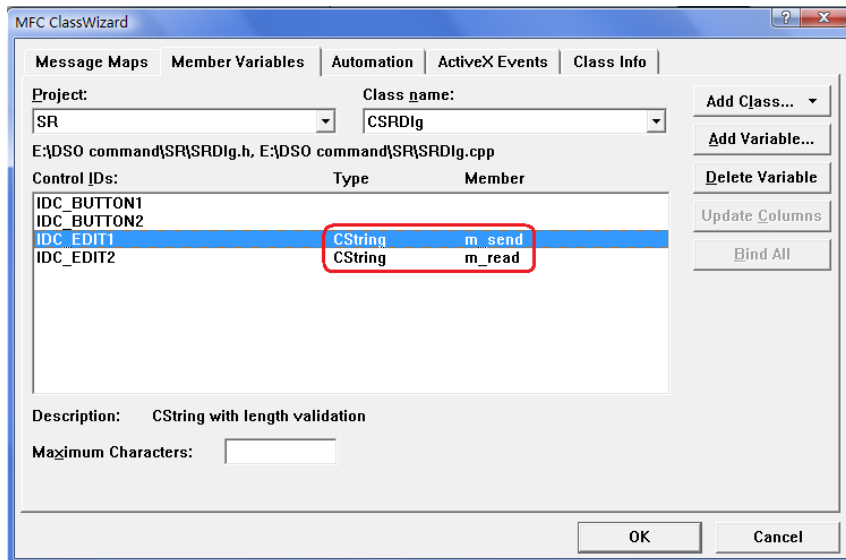


图 4-1-6

(3) 添加两个 Button 控件，分别命名为 Send 和 Read。

6. 双击“Send”按键，进入编程环境。

(1) 在头文件处添加“#include “visa.h””,进行声明。

(2) 在头文件后，添加如下代码，定义相关的变量。

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

(3) 在::CSRDlg(CWnd* pParent /*=NULL*/)
: CDialog(CSRDlg::IDD, pParent)里，令m_send = _T(“*IDN?\n”);

(4) 在::OnInitDialog()里，添加如下代码。

```
viOpenDefaultRM (&defaultRM);
//获取visa的USB资源
viFindRsrc(defaultRM, "USB?* ", &list,&nmatches, matches);
```

```
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);
```

(5) 在”Send”按键处，添加如下代码。

```
//发送接收到的命令
```

```
UpdateData (TRUE);
```

```
strTemp = m_send + "\n";
```

```
stringTemp = (char *) (LPCTSTR) strTemp;
```

```
viPrintf (vi,stringTemp);
```

(6) 在”Read”按键处，添加如下代码。

```
//读取结果
```

```
viScanf (vi, "%t\n", &buf);
```

```
//将结果显示出来
```

```
m_read = buf;
```

```
UpdateData (FALSE);
```

(7) 在::OnQueryDragIcon()下，添加如下代码。

```
//关闭资源
```

```
viClose (vi);
```

```
viClose (defaultRM);
```

7. 保存、编译并运行程序，得到可执行文件。当仪器与PC机成功相连时，在Input下编辑框中输入一条命令，如*IDN? (*IDN?为默认输入命令)，按“Send”按键，再按“Read”按键，仪器返回的结果将显示在Output下编辑框中。如图4-1-7所示。

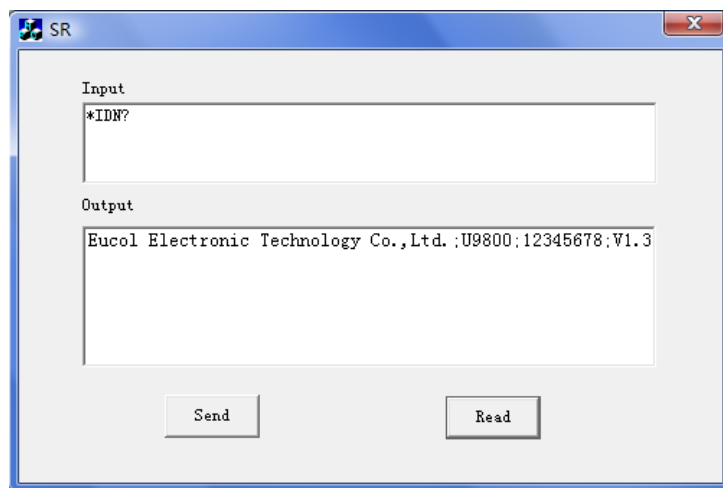


图 4-1-7

4.2 Visual Basic 6.0 编程实例

请运行 Visual Basic 6.0 软件，并按如下步骤进行操作：

1. 建立一个Standard EXE工程。
2. 打开 Project->Add Module 的 **Existing** 选项卡，打开 NI-VISA 安装路径下 C:\Program Files\IVI Foundation\VISA\WinNT\include 文件夹中的 **visa32.bas** 文件并添加。如图 4-2-1 所示。

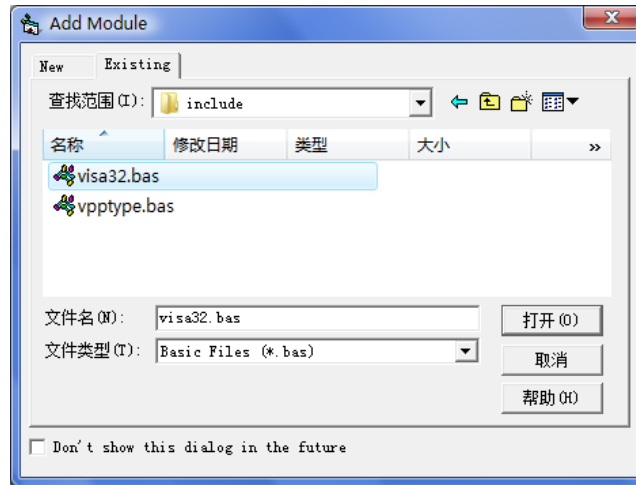


图 4-2-1

3. 添加名称分别为 Input 和 Output 的 Label、两个 TextBox 以及名称分别为 Send 和 Read 的 CommandButton。并把 Input 下的 TextBox 的属性中的 Text 设置为 *IDN? 的初始状态。如图 4-2-2 所示。

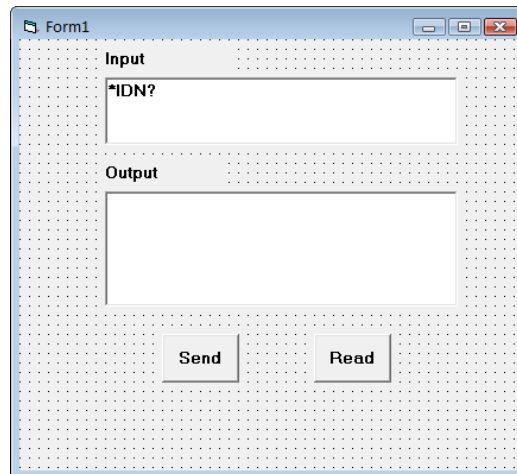


图 4-2-2

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 Send 按钮进入编程环境，添加如下代码。

```
Dim defrm As Long
```

```
Dim vi As Long
```

```
Dim list As Long
```

```
Dim nmatches As Long
```

```
Dim matches As String * 200 '保留获取设备号
```

```
Dim strRes As String * 200
```

```
Private Sub Cmd_Read_Click()
```

```
' 获取命令返回状态
```

```
Call viVScanf(vi, "%t", strRes)
```

```
Txt_output.Text = strRes
```

```
End Sub
```

```
Private Sub Cmd_Send_Click()
```

```
' 发送询问命令内容
```

```
Call viVPrintf(vi, Txt_input.Text + Chr$(10), 0)
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
' 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)
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
' 关闭资源
```

```
Call viClose(vi)
```

```
Call viClose(defrm)
```

End Sub

6. 保存并运行整个工程，得到单个可执行文件。当仪器与 PC 机成功相连时，在 Input 下文本编辑框中输入一条命令，如*IDN? (*IDN?为默认输入命令)，按“Send”按键，再按“Read”按键，仪器返回的结果将显示在 Output 下文本编辑框中。如图 4-2-3 所示。

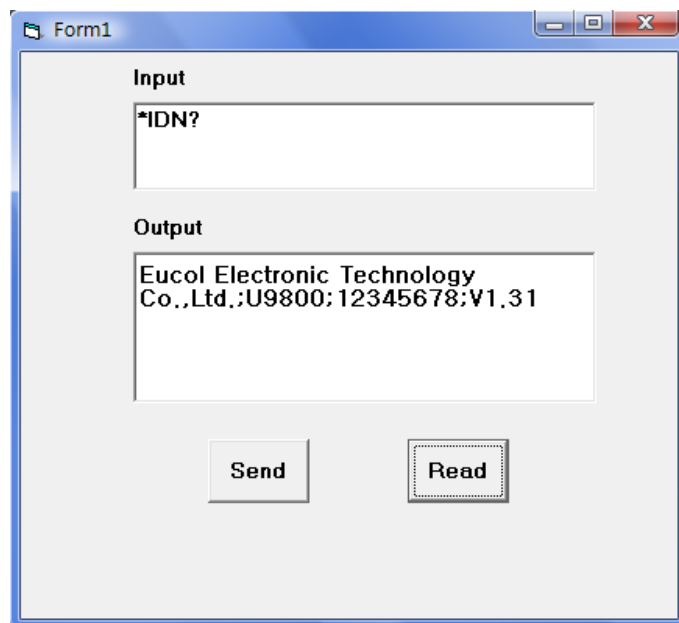


图 3-2-3

4.3 LabVIEW 8.5 编程实例

运行 LabVIEW 8.5 软件，并按如下步骤进行操作：

1. 进入 Getting Started 画面。选择 New>>Blank VI，新建一个 VI。如图 4-3-1 所示。

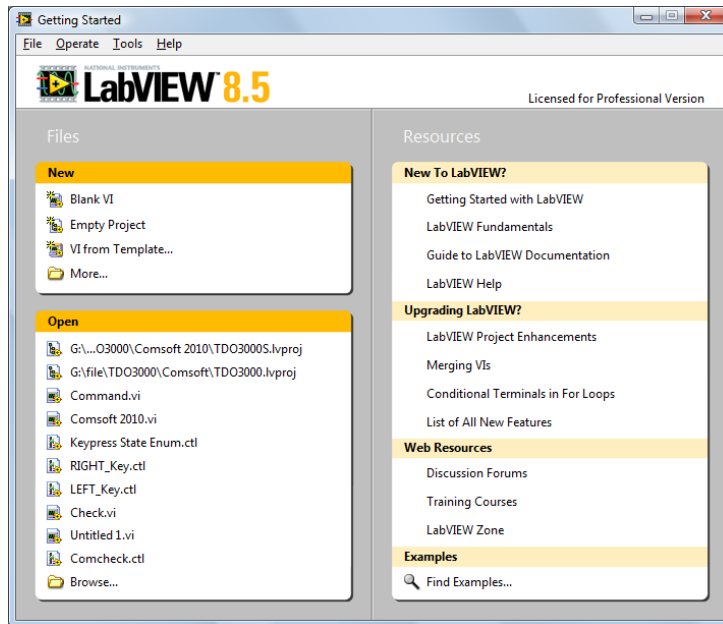


图 4-3-1

2. 从前面板选择 Controls>>Modern>>Boolean>>OK Button 放置 3 个按钮分别定义为 Write，Read 和 Stop。如图 4-3-2 所示。

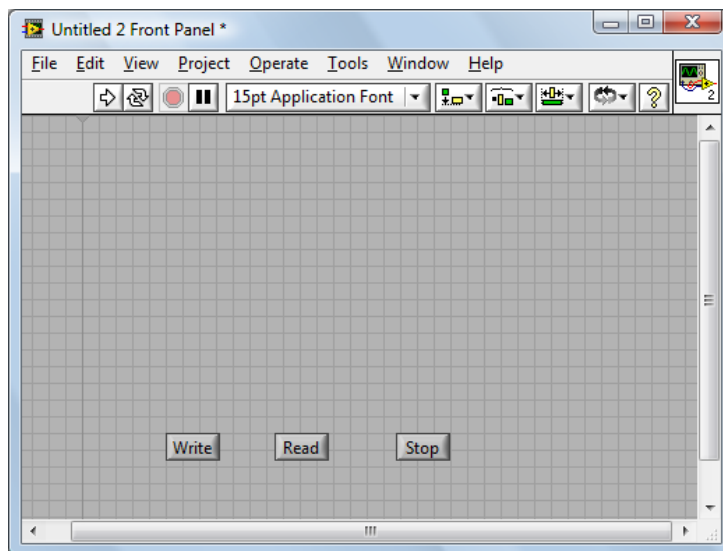


图 4-3-2

3. 从程序框图的函数选板选择 Functions>>Programming>>Structures>>Event Structure 来放置一个 Event Structure 事件框。
4. 从程序框图右击事件框选择 Add Event Case..., 为 3 个按键分别添加 Value Changed 事件。将按键接线端分别放入各自的事件框。
5. 选择 Write 按键 Value Change 事件框, 从函数选板选择 Functions>>Instrument I/O>>VISA>>VISA Write 来放置一个 VISA Write 函数到 Write 按键的 Value Change 事件框中。
6. 从函数选板选择 Functions>>Instrument I/O>>VISA>>VISA Advanced>>VISA Open 来放置一个 VISA Open 函数到事件框的左侧。
7. 右键 VISA Open 的 VISA resource name 接线端, 在快捷键菜单中选择 Create>>Control 来创建一个输入控件 VISA resource name。
8. 连线 VISA Open 的 VISA resource out 接线端和事件框内 VISA Write 函数的 VISA resource name 接线端。连线 error out 和 error in 接线端。
9. 右键 VISA Write 的 write buffer 接线端, 在快捷键菜单中选择 Create>>Control 来创建一个输入控件 write buffer。如图 4-3-3 所示。

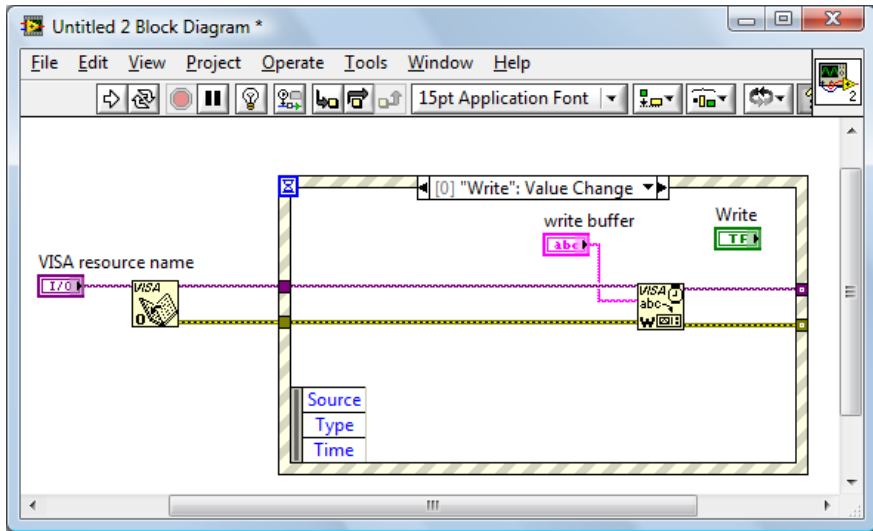


图 4-3-3

10. 选择 Read 按键 Value Change 事件框, 从函数选板选择 Functions>>Instrument I/O>>VISA>>VISA Read 来放置一个 VISA Read 函数到 Read 按键的 Value Change 事件框中。
11. 右键 VISA Read 的 read buffer 接线端, 在快捷键菜单中选择 Create>>Indicator 来创建一个显示控件 read buffer.
12. 右键 VISA Read 的 byte count 接线端, 在快捷键菜单中选择 Create>>Constant 来创建一个常量。并将常量设为 1024.
13. 连线 error 接线端, 连线 VISA resource name 接线端。如图 4-3-4 所示。

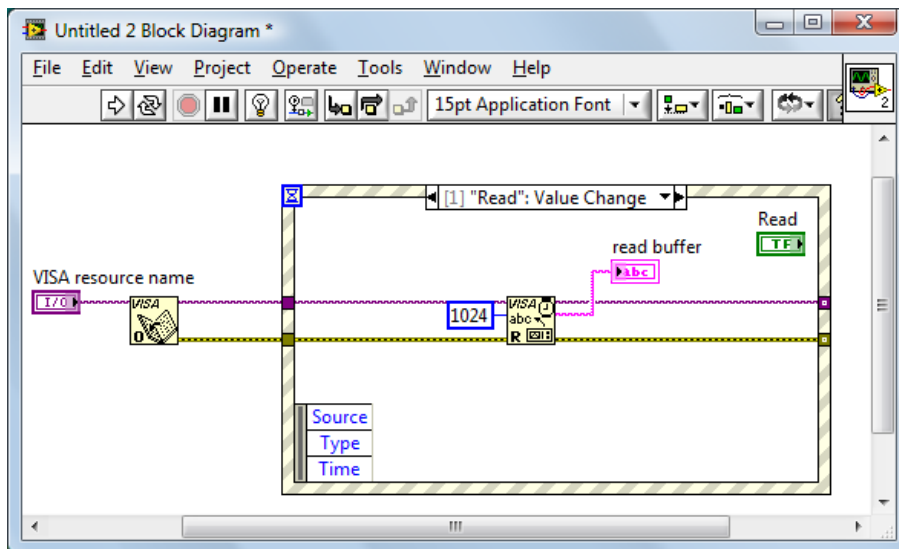


图 4-3-4

14. 选择 Stop 按键 Value Change 事件框，从函数选板选择 Functions>>Instrument I/O>>VISA>>VISA Advanced>>VISA Close 来放置一个 VISA Close 函数到 Stop 按键的 Value Change 事件框中。
15. 连线 error 接线端，连线 VISA resource name 接线端。
16. 从函数选板选择 Functions>>Programming>>Structures>>While Loop 来放置一个 While Loop 结构到事件框的周围。
17. 从函数选板选择 Functions>>Programming>>Boolean>>True Constant 来放置一个 True Constant 常量到 Stop 事件框的内。连接 True Constant 常量到 While Loop 循环结构的停止接线端。
18. 从函数选板选择 Functions>>Programming>>Dialog & User Interface>>Simple Error Handler 来放置一个 Simple Error Handler 函数。连线 error 接线端。
19. 右键 While Loop 循环结构与 error 连线的接线端 Loop Tunnel，在快捷键菜单中选择 Replace with Shift Register 来创建循环移位寄存器对来替代循环隧道。同样也将 VISA resource name 的循环隧道替换成移位寄存器对。
20. 在前面板，调整控件的位置排列如图 4-3-5 所示。

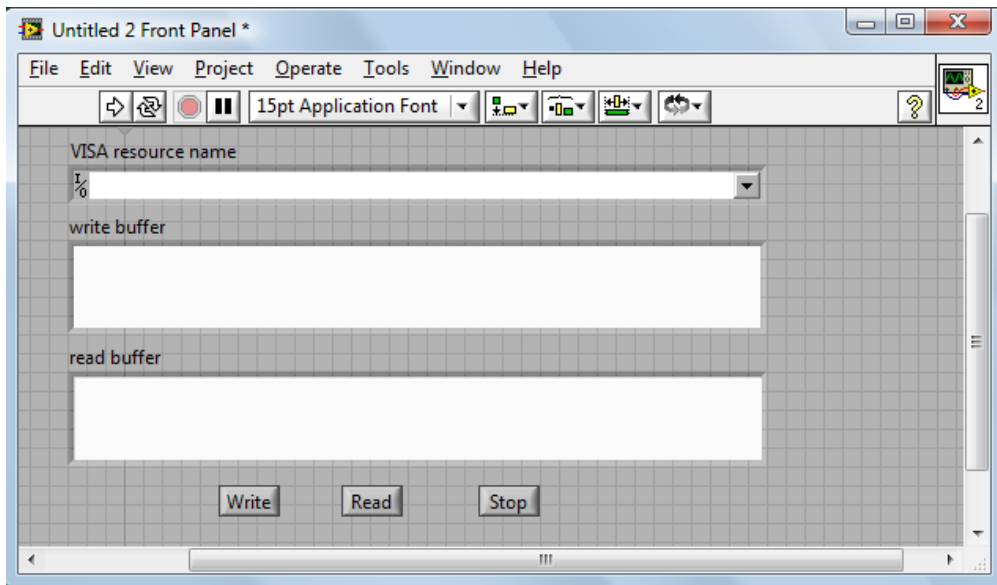


图 4-3-5

21. 保存当前 VI。在运行本 VI 前从 VISA resource name 下拉框中选择正确的 VISA 资源名。
22. 运行本 VI，在 write buffer 中输入你想要发送的命令或查询，如：*idn?。点击 Write 键发送命令。然后按 Read 键读取返回信息。执行结果如图 4-3-6 所示。

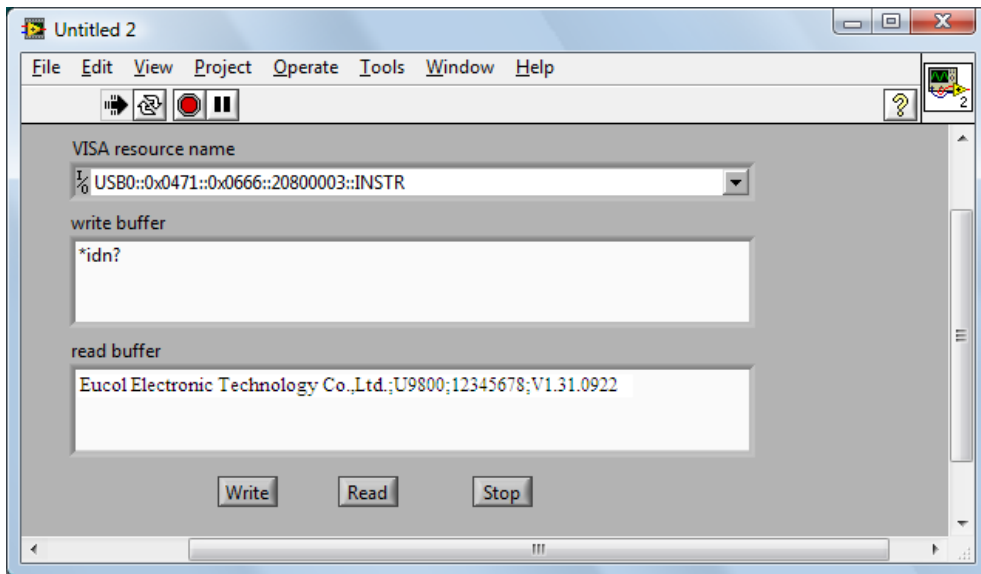


图 4-3-6

23. 按 Stop 键退出程序执行。