

U2516 系列直流电阻测试仪 编程手册

常州市优高电子科技有限公司

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	中止测试系统命令	8
2.4	初始系统命令	8
2.5	触发系统命令	9
2.6	功能系统命令	1
2.7	测量速度系统命令	4
2.8	数据读取系统命令	4
2.9	比较系统命令	5
2.10	数据存储系统命令	8
2.11	文件存储系统命令	9
2.12	按键系统命令	10
2.13	系统配置系统命令	11
3	出错信息	14
4	编程实例	15
4.1	Visual C++ 6.0 编程实例	17
4.2	Visual Basic 6.0 编程实例	22
4.3	LabVIEW 8.5 编程实例	25

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: Save TYPE 长格式为 STYPE, 缩写为 STYP。

2 命令系统

U2516 系列直流电阻测试仪的命令系统主要包括下列系统命令：

- ◆ 公用命令
- ◆ 显示(DISPlay)系统命令
- ◆ 初始(INITiate)系统命令
- ◆ 触发(TRIGger)系统命令
- ◆ 功能(FUNcTion)系统命令
- ◆ 测量速度(APERture)系统命令
- ◆ 数据读取(FETCh)系统命令
- ◆ 中止测试(ABORt)系统命令
- ◆ 比较(COMParator)系统命令
- ◆ 数据存储(MEMory)系统命令
- ◆ 文件存储(MMEMory)系统命令
- ◆ 按键(KEY)系统命令
- ◆ 系统配置(SYSTem)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?	<厂商>,<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 Tech.,<仪器型号>,<仪器序列号>,<仪器软件版本>

2. *RST (Reset)

复位测量设置，复位触发状态和数据读取(FETCH)状态。

命令语法：*RST

3. *RCL <value>

调出 value 指定的内部设置文件。value 范围： 1~100。

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
MMEMory命令也可以完成调用功能。

4. *SAV <value>

保存设置文件到 value 指定位置。value 范围：1~100。

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17。

例如：*SAV 1 将设置文件保存到内部存储器位置1处
MMEMory命令也可以完成保存功能。

5. *TRG

触发仪器测量并返回测量结果，相当于 TRIG+FETCh 命令。

命令语法：*TRG

注意：该命令需要配合INITiate系统命令使用。

6. *CLS

清除状态寄存器，并且清除输出缓冲区和清除 OPC 位。此外还将清除数据读取 (FETCH)状态和触发状态。

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. *TST?

返回自测试结果, 仪器工作正常时总是返回 0。

查询语法: *TST?

查询返回: {1 | 0}<NL>

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。

DISPlay	:PAGE	<page name>
	:ENABle	ON (1) OFF (0)
	:EASy	ON (1) OFF (0)
	:CCLear	

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)
BCOunter	设定显示页面至: 档计数显示页面(Bin COunter)
MSEtup	设定显示页面至: 测量设置页面(Meas SEtup)
LTABle	设定显示页面至: 极限设置页面(Limit TABle)
SYSTem	设定显示页面至: 系统设置页面(SYSTem)
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 | BCOU | MSET | LTAB | SYST | INFO | FLIS}<NL>

:ENABle 设定测量页面数据显示开关，此命令还将自动关闭全屏显示模式。

:ENABle? 查询当前的测量页面数据显示开关。

命令语法: DISPlay:ENABle { {1 | ON} | {0 | OFF} }

查询语法: DISPlay:ENABle?

查询返回: {1 | 0}<NL>

:EASy 设定简易测试模式开关，打开简易测试模式后仪器自动将显示页面设定为测量显示页面(MEASurement)。:EASy? 查询当前的全屏显示开关。

命令语法: DISPlay:EASy { {1 | ON} | {0 | OFF} }

查询语法: DISPlay:EASy

查询返回: {1 | 0}<NL>

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

:CClear 清除信息显示区域的出错信息。

命令语法: `DISPlay:CClear`

2.3 中止测试系统命令

ABORt 中止测试系统命令用于放弃当前正在进行的一次测量，复位触发状态以及 **FETCh** 状态。

命令语法: `ABORt`

2.4 初始系统命令

INITiate 初始系统命令结合 ***TRG** 命名控制仪器触发。表 2-4-1 是统计系统命令树。

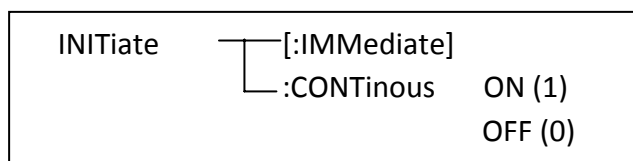


表 2-4-1 初始系统命令树

[:IMMediate] 允许仪器接收一次触发命令(***TRG**)。

命令语法: `INITiate [:IMMediate]`

:CONTinous 设置仪器是否允许触发命名持续有效。设置为 1 时，***TRG** 命令总是触发仪器进行测试，设置为 0 时仪器只有在 `INIT:IMM` 命令后才能接收一次 ***TRG** 触发指令。

命令语法: `INITiate:CONTinous { {1 | ON} | {0 | OFF} }`

查询语法: `INITiate:CONTinous?`

查询返回: `{1 | 0}<NL>`

2.5 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 2-5-1 是 TRIGger 触发系统命令树。

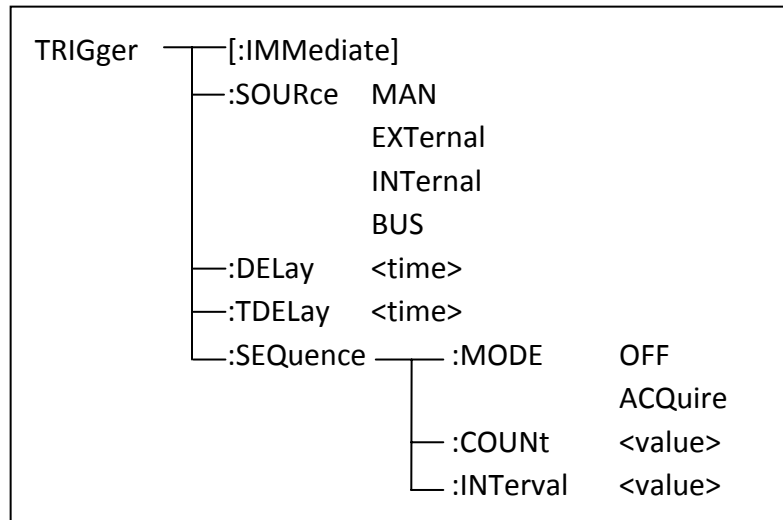


表 2-5-1 触发系统命令树

[:IMMEDIATE] 用于触发一次测量。

命令语法: TRIGger[:IMMEDIATE]

①注意: 此命令只在测量显示页面和档计数页面有效, 在其他页面该命令被忽略。只有触发模式是 **BUS** 时, 该命令才能有效。

: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 | HOLD | EXTernal | INTernal | BUS}

其中, MANual 和 HOLD 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 | HOLD | EXT | INT | BUS}<NL>

:DELay 或者:TDELay 设定触发延时时间。:DELay? 或者:TDELay? 查询当前设定的触发延时时间。触发延时时间范围是 0~60s, 1ms 步进。

命令语法: TRIGger:DELay <time> time 是 NR1, NR2 或 NR3 格式, 可跟时间单位

查询语法: TRIGger:DELay?

查询返回: <NR3><NL>

:SEquence:MODE 用于设定序列测试开关状态，**:SEquence:MODE?**用于查询当前序列测试开关状态。

命令语法: TRIGger:SEquence:MODE {OFF | ACQuire}

其中，OFF 表示关闭序列测试，这时仪器处于正常测试状态，ACQuire 表示序列测试状态，这时仪器以设定的间隔时间采集数据，然后将数据存入序列缓存中，缓冲数据数目由 SEquence:COUNt 设定。

查询语法: TRIGger:SEquence:MODE?

查询返回: {OFF | ACQ}<NL>

:SEquence:COUNt 设定序列测试数据缓冲区数目。**:SEquence:COUNt?**用于查询当前设定的序列测试数据缓冲区数目。

命令语法: TRIGger:SEquence:COUNt <value>

这里: <value>是 NR1 格式，范围是 1~512，无单位。

查询语法: TRIGger:SEquence:COUNt?

查询返回: <NR1><NL>

:SEquence:INTerval 设定序列测试数据采集间隔时间。**:SEquence:INTerval?**用于查询当前设定的序列测试数据采集间隔时间。

命令语法: TRIGger:SEquence:INTerval <value>

这里: <value>可以是 NR1, NR2 或 NR3 格式，范围是 0~999，时间单位。

查询语法: TRIGger:SEquence:INTerval?

查询返回: <NR1><NL>

注: 并非所有仪器支持序列测试模式，详情请参考用户说明书。

2.6 功能系统命令

FUNcTion 功能系统命令主要用于量程, 结果显示模式, 温度显示和温度补偿。
表 2-6-1 是 FUNcTion 系统命令树。

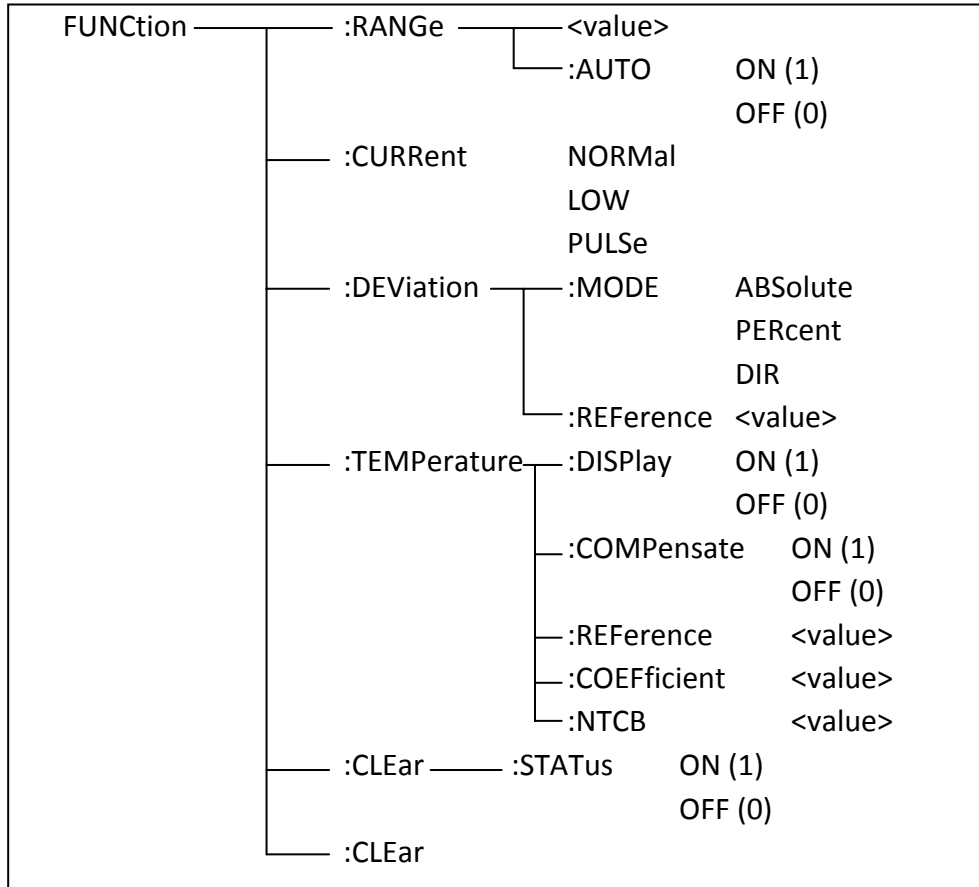


表 2-6-1 功能系统命令树

:RANGe 用于设定仪器的量程。:RANGe?用于查询仪器的量程。

命令语法: FUNcTion:RANGe {10maohm | 1maohm | 100kohm | 10kohm | 1kohm | 100ohm | 10ohm | 1ohm | 100mohm | 10mohm | 1mohm}

查询语法: FUNcTion:RANGe?

查询返回: {10maohm | 1maohm | 100kohm | 10kohm | 1kohm | 100ohm | 10ohm | 1ohm | 100mohm | 10mohm | 1mohm}<NL>

①注意: 不同型号仪器的量程范围有所不同, 请根据仪器说明书获取实际的量程范围。

☞提示: 仪器使用的电阻倍率是 ma(10^6), k(10^3), m(10^{-3}), u(10^{-6}), 单位是 ohm。仪器使用的时间倍率是 m(10^{-3}), 单位是 s

:RANGe:AUTO 用于设定仪器的量程自动选择开关。:RANGe:AUTO?用于查询当前仪器的量程自动选择开关状态。

命令语法: FUNCtion:RANGe:AUTO { {1 | ON} | {0 | OFF} }

查询语法: FUNCtion:RANGe:AUTO?

查询返回: {1 | 0}<NL>

:CURRent 用于设定仪器的电流模式。:CURRent?用于查询当前仪器的电流模式。

命令语法: FUNCtion:CURRent { NORMAl | LOW | PULSe }

这里: HIGH: 普通电流模式; LOW: 低电流模式; PULSe: 脉冲电流模式

查询语法: FUNCtion:CURRent?

查询返回: {NORM | LOW | PULS}<NL>

:DEVIation:MODE 用于设定仪器的测量结果显示模式。:DEVIation:MODE 用于查询当前仪器的测量结果显示模式。

命令语法: FUNCtion:DEVIation:MODE { ABSolute | PERcent | DIR }

这里: ABSolute: 绝对值偏差; PERCent: 百分比偏差; DIR: 实测值

查询语法: FUNCtion:DEVIation:MODE?

查询返回: {ABS | PER | DIR}<NL>

:DEVIation:REFeRence 用于设定仪器的偏差标称值。:DEVIation:REFeRence?用于查询当前仪器的偏差标称值。测量结果显示模式是绝对值和百分比偏差显示的时候需要使用偏差标称值。**该标称值和极限设置的标称值是同一个数据。**

命令语法: FUNCtion:DEVIation:REFeRence <value>

这里: <value>可以是 NR1、NR2 或 NR3 数据格式, 范围是-999M~999M, 电阻单位或无单位。

查询语法: FUNCtion:DEVIation:REFeRence?

查询返回: <NR3><NL>

:TEMPerature:DISPlay 用于设定仪器温度显示开关。:TEMPerature:DISPlay?用于查询当前的温度显示开关状态。

命令语法: FUNCtion:TEMPerature:DISPlay { {1 | ON} | {0 | OFF} }

查询语法: FUNCtion:TEMPerature:DISPlay?

查询返回: {1 | 0}<NL>

:TEMPerature:COMPensate 设定仪器温度补偿开关。:TEMPerature:COMPensate?用于查询当前的温度补偿开关状态。

命令语法: FUNCtion:TEMPerature:COMPensate { {1 | ON} | {0 | OFF} }

支持 NTC 时: FUNCtion:TEMPerature:COMPensate {OFF | NORMal | NTC}

查询语法: FUNCtion:TEMPerature:COMPensate?

查询返回: {1 | 0}<NL> 或者 {OFF | NORM | NTC}<NL>

:TEMPerature:REference 设定仪器参考温度。:TEMPerature:REference?用于查询当前的参考温度。

命令语法: FUNCtion:TEMPerature:REference <value>

这里: <value>可以是 NR1、NR2 或 NR3 格式, 范围是-999~999, 无单位。

查询语法: FUNCtion:TEMPerature:REference?

查询返回: <NR3><NL>

:TEMPerature:COEfficent 设定仪器温度系数。:TEMPerature:COEfficent?用于查询当前的温度系数。

命令语法: FUNCtion:TEMPerature:COEfficent <value>

这里: <value>可以是 NR1、NR2 或 NR3 格式, 温度系数可以是%单位或者 ppm 单位, %单位时范围是-999~999, ppm 单位时范围是 -999000~999000, 如果 value 不带单位, 则取仪器当前温度系数单位。

查询语法: FUNCtion:TEMPerature:COEfficent?

查询返回: <NR3><NL> 返回数据包含单位%或者 ppm

:TEMPerature:NTCB 设定 NTC 热敏电阻材料常数。:TEMPerature:NTCB?用于查询当前设定的热敏电阻材料常数。

命令语法: FUNCtion:TEMPerature:NTCB <value>

这里: <value>可以是 NR1、NR2 或 NR3 格式, 范围是 1000~9999, 无单位。

查询语法: FUNCtion:TEMPerature:NTCB?

查询返回: <NR3><NL>

:CLEar:STATus 用于设定仪器的清零开关。:CLEar:STATus?用于查询当前仪器的清零开关状态。

命令语法: FUNCtion:CLEar { {1 | ON} | {0 | OFF} }

查询语法: FUNCtion:CLEar?

查询返回: {1 | 0}<NL>

:CLEar 用于执行仪器清零操作, 在清零操作前应该正确短路测试夹。

命令语法: FUNCtion:CLEar

①注意: 执行 **FUNCtion:CLEar** 后, 仪器自动打开清零开关。清零命令仅在测量显示分选显示页面有效, 其他页面忽略清零命令。

2.7 测量速度系统命令

APERTure 测量速度命令用于设定仪器的测试速度，以及测试的平均次数。

APERTure?用于查询当前的测试速度和平均次数。

命令语法: APERTure {FAST | MEDium | SLOW}[,<value>]

这里: FAST 表示快速, MEDium 表示中速, SLOW 表示慢速。

<value>是 NR1 数据格式, 范围是 1~255, 无后缀参数。

查询语法: APERTure?

查询返回: {FAST | MED | SLOW},<NR1><NL>

2.8 数据读取系统命令

FETCH 数据读取命令系统用于读取仪器测量结果, 表 2-8-1 是数据读取系统命令树。

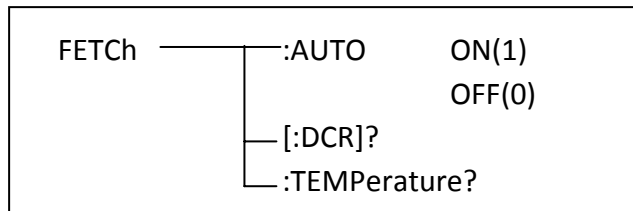


表 2-8-1 数据读取系统命令树

:AUTO 用于设定仪器自动查询功能开关。:AUTO?查询当前的自动查询开关状态。

命令语法: FETCH:AUTO {{1 | ON} | {0 | OFF}}

查询语法: FETCH:AUTO?

查询返回: {1 | 0}<NL>

☞提示: 自动查询功能打开时, 仪器每次测试后自动返回电阻测量结果。

[:DCR]?命令使仪器返回一次直流电阻测量结果。

查询语法: FETCH[:DCR]?

查询返回: <NR3, NR1><NL> 即<结果, 档号><NL>

返回档号的具体内容如下所述: 0 表示不进行比较, 1~4 表示分选所在档, 11 表示下超档, 12 表示上超档。

:TEMPerature?命令使仪器返回温度测量结果。

查询语法: FETCH:TEMPerature?

查询返回: <NR3 ><NL>

2.9 比较系统命令

COMParator 比较系统命令用于设定档比较器功能，包括比较器开关以及极限列表的设定。表 2-9-1 是 COMParator 比较系统命令树。

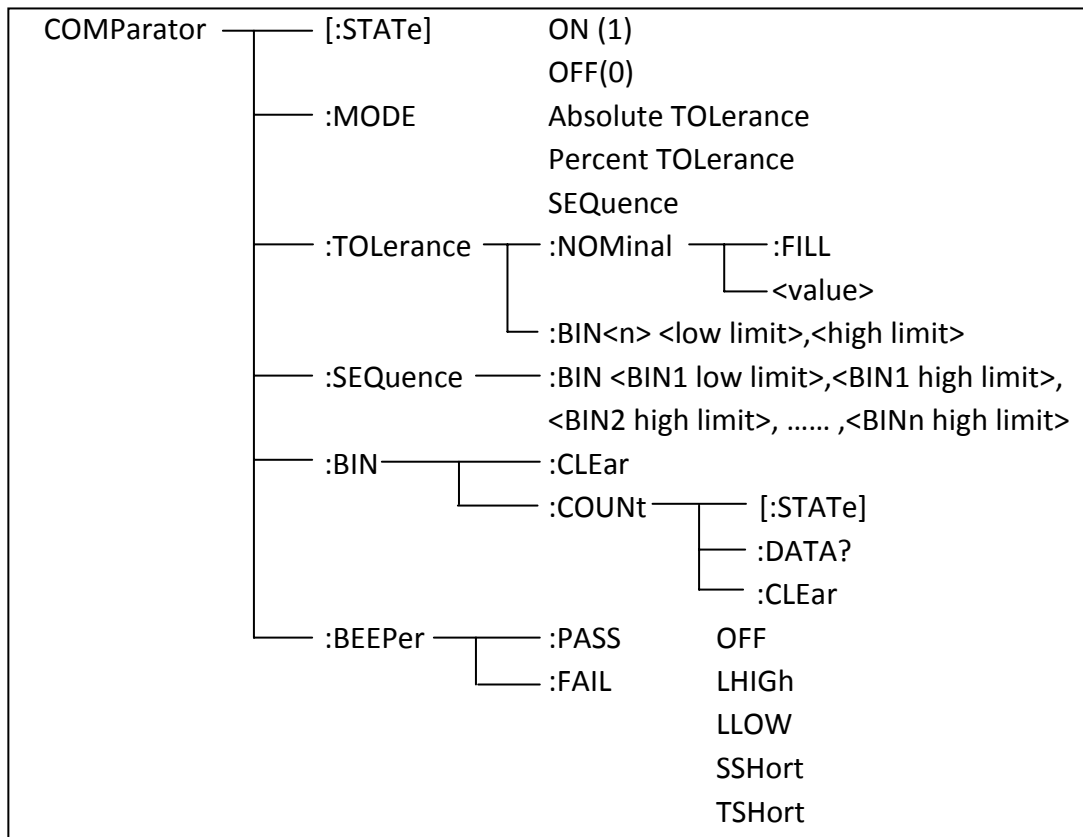


表 2-9-1 比较系统命令树

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

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

查询语法: COMParator[:STATEe]?

查询返回: {1 | 0}<NL>

:MODE 用于设定仪器比较的极限方式。:MODE? 查询仪器当前设定的比较极限方式。

命令语法: COMParator:MODE {ATOLerance | PTOLerance | SEQUence}

这里: ATOLerance 是绝对值容差方式, PTOLerance 是百分比容差方式, SEQUence 是顺序方式。**注意: U2516B 不支持 SEQ 顺序方式。**

查询语法: COMParator:MODE?

查询返回: {ATOL | PTOL | SEQ}<NL>

:TOLerance:NOMinal 用于设定比较功能容差方式的标称值(该值仅用于百分比容差和绝对值容差方式,也用于显示时的百分比和绝对值偏差方式)。

:TOLerance:NOMinal? 返回仪器当前设置的容差方式标称值。

命令语法: **COMParator:TOLerance:NOMinal <value>**

这里: <value>可以是 NR1、NR2 或 NR3 数据格式,范围是-999M~999M,电阻单位或者不加单位。

命令语法: **COMParator:TOLerance:NOMinal:FILL** 使用最近一次的测量结果作为标称值。

查询语法: **COMParator:TOLerance:NOMinal?**

查询返回: **<NR3><NL>**

:TOLerance:BIN<n> 用于设定比较功能容差模式各档上下限极限数值。

:TOLerance:BIN<n>?用于查询当前仪器设定的各档上下限数值。

命令语法: **COMParator:TOLerance:BIN<n><low limit>,<high limit>**

这里: <n> 1 到 4 (NR1): 档号数 (U2516B 只能是 1)

<low limit> 下限数据, NR1, NR2 或 NR3 格式数据

<high limit> 上限数据, NR1, NR2 或 NR3 格式数据

注: 下限数据应小于上限数据, 否则提示出错信息。

百分比容差模式时, 上下限输入是无单位参数, 范围是-999~999。

绝对值容差模式时, 上下限输入是电阻单位, 范围是-999M~999M。

提示: 如果当前是顺序极限模式, 那么该命令极限值写入百分比容差极限中。

查询语法: **COMParator:TOLerance:BIN<n>?**

查询返回: **<low limit>,<high limit><NL>** 返回上下限是 NR3 格式。

:SEQuence:BIN 用于设定比较功能顺序模式上下限数据。**:SEQuence:BIN?**查询仪器当前设定的顺序模式各档上下限数值。

命令语法: **COMParator:SEQuence:BIN <BIN1 low limit>,<BIN1 high limit>,<BIN2 high limit>,...,<BINn high limit>**

这里:<BIN1 low limit> NR1, NR2 或 NR3 数据格式, 档 1 的下限数值

<BIN1 high limit> NR1, NR2 或 NR3 数据格式, 档 1 的上限数值

<BINn high limit> NR1, NR2 或 NR3 数据格式, 档 n 的上限数值

注: 下限应该小于上限, 上下限输入的范围是-999M~999M, 电阻单位或者不加单位。U2516B 只有一档。

查询语法: **COMParator:SEQuence:BIN?**

查询返回: **<BIN1 low limit>,<BIN1 high limit>,<BIN2 high limit>,...,<BINn high limit><NL>**

所有返回的极限数据都是 NR3 格式。

:BIN:CLEar 用于清除极限列表设置各档极限设置数据。

命令语法: COMParator:BIN:CLEar

①注意: 该指令仅清除当前极限方式的极限设置数据(每种极限方式包含独立的极限设置数据)。

:BIN:COUNT[:STATe]用于设定档计数功能开关。:BIN:COUNT[:STATe]?查询当前仪器设定的档计数开关状态。

命令语法: COMParator:BIN:COUNT[:STATe] {{1 | ON} | {0 | OFF}}

查询语法: COMParator:BIN:COUNT[STATe]?

查询返回: {1 | 0}<NL>

:BIN:COUNT:DATA?用于返回档计数结果。

查询语法: COMParator:BIN:COUNT:DATA?

查询返回: <BIN1 count>,<BIN2 count>,...,<BINn count>,<LOW BIN count>,<HIGH BIN count><NL>

这里: BIN1~BINn count 是 1~n 档的档计数结果。

LOW BIN count 是下超档的档计数结果。

HIGH BIN count 是上超档的档计数结果。

:BIN COUNT:CLEar 用于清除所有档计数结果。

命令语法: COMParator:BIN:COUNT:CLEar

:BEEPer:PASS 设定比较合格时的报警方式。:BEEPer:PASS?查询仪器当前设定的比较合格时的报警方式。

命令语法: COMParator:BEEPer:PASS {OFF | LHIGH | LLOW | SSHort | TSHort}

这里: OFF 表示关闭报警; LHIGH 表示长高音报警; LLOW 表示长低音报警; SSHort 表示短音报警; TSHort 表示双短音报警。

查询语法: COMParator:BEEPer:PASS?

查询返回: {OFF | LHIG | LLOW | SSH | TSH}<NL>

:BEEPer:FAIL 设定比较不合格时的报警方式。:BEEPer:FAIL?查询仪器当前设定的比较不合格时的报警方式。

命令语法: COMParator:BEEPer:FAIL {OFF | LHIGH | LLOW | SSHort | TSHort}

这里: OFF 表示关闭报警; LHIGH 表示长高音报警; LLOW 表示长低音报警; SSHort 表示短音报警; TSHort 表示双短音报警。

查询语法: COMParator:BEEPer:FAIL?

查询返回: {OFF | LHIG | LLOW | SSH | TSH}<NL>

2.10 数据存储系统命令

MEMory 数据存储系统命令主要用于管理测量数据记录功能。表 2-10-1 是 MEMory 数据存储系统命令树。

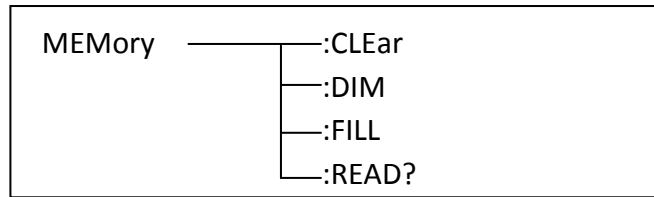


表 2-10-1 数据存储系统命令

:CLEar 用于清除数据记录缓冲区。当数据记录打开时，该命令将缓冲区中的数据写入 USB 存储器中，然后结束数据记录并且清空数据记录缓冲区。当数据记录未打开时，直接清除数据记录缓冲区。

命令语法: MEMory:CLEar DBUF

:DIM 用于设置数据记录缓冲区大小。:DIM?查询当前的数据记录缓冲区大小。

命令语法: MEMory:DIM DBUF,<numeric>

其中: numeric 是 NR1 格式，无单位，范围是 1~255。

查询语法: MEMory:DIM?

查询返回: <NR1><NL>

仪器开机默认的缓冲区大小是 128，只有该命令能够修改缓冲区大小。

:FILL 用于启动数据记录功能。

命令语法: MEMory:FILL DBUF

如果当前数据记录功能以及启用，忽略该命令。

如果当前的保存类型不是 CSV 格式，那么仪器将自动设置为 CSV 格式。

:READ?命令用于查询缓冲区数据。

命令语法: MEMory:READ?

注意: 如果缓冲区中没有数据时返回 0。返回的每一个测试数据和 FETCH:DCR? 返回的数据一样，每个测试数据之间使用换行符("\r\n")隔开。

2.11 文件存储系统命令

Mass MEMory 文件存储系统命令用于文件的保存与加载。表 2-11-1 是 Mass MEMory 系统命令树。

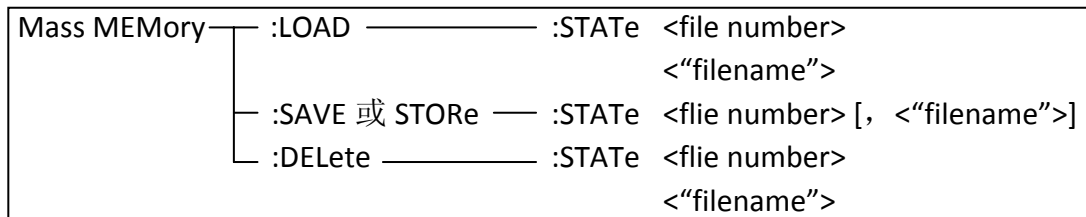


表 2-11-1 文件存储系统命令树

:LOAD:STATe 命令用于加载已保存的文件。

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

<flie number>是文件序号, 范围是 1~100, 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:SAVE 和:DEL 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指定位置文件。

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

<file number> 是文件序号, 范围是 1~100, NR1 数据格式, 无后缀参数。

<“filename”> 是要保存的文件名, 可用 17 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>

<flie number> 是文件序号, 范围是 1~100, 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: MMEMory:DElete:STATe “filename”

按照文件名删除文件, 仅限于内部文件, “filename” 是要删除的文件名称, 可用 17 个以内的 ASCII 字符表示, 注意要用引号将文件名括起来。

2.12 按键系统命令

KEY 按键系统命令等同于前面板上的按键。

KEY:LOCAl	释放按键操作,退出远控（非按键，用于退出远控）。
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:UPPer	上按键
KEY:DOWN	下按键
KEY:LEFT	左按键
KEY:RIGHT	右按键
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:TRIGger	触发按键
KEY:KEYLock	按键锁定/解锁按键
KEY:CLEAr	清零按键
KEY:F<n>	软键，n 为 1~5

2.13 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，handler 接口配置等。表 2-13-1 是系统配置系统命令树。

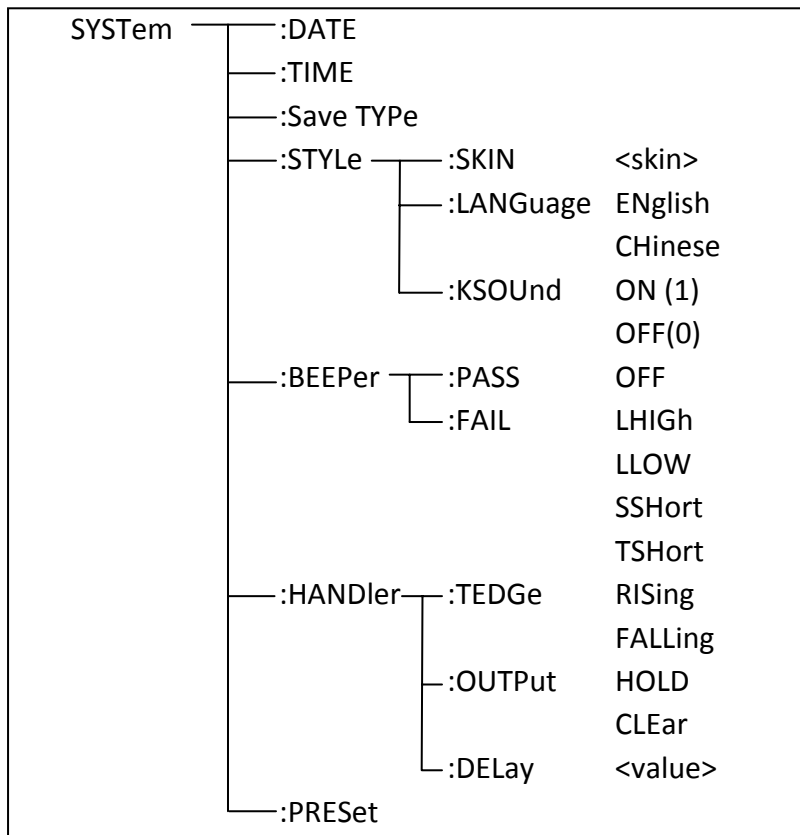


表 2-13-1 系统配置系统命令树

: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>

其中: hour, minute, second 是 NR1 格式。

查询语法: SYSTem:TIME?

查询返回: <NR1>,<NR1>,<NR1><NL>

:Save TYPE 用于设定文件保存类型。:Save TYPE?查询文件保存类型。

命令语法: SYSTem:STYPe {CSV | GIF | BMP | PNG}

查询语法: SYSTem:STYPe?

查询返回: {CSV | GIF | BMP | PNG}<NL>

:STYLE:SKIN 用于设定仪器的显示风格。:STYLE:SKIN?查询当前的显示风格。

命令语法: SYSTem:STYLe:SKIN {GRAY | BLACK | BLUE | CYAN}

其中: GRAY 表示简洁灰; BLACK 表示炫酷黑; BLUE 表示典雅蓝; CYAN 表示深青色。

查询语法: SYSTem:STYLe:SKIN?

查询返回: {GRAY | BLACK | BLUE | CYAN}<NL>

:STYLE:LANGUage 用于设定仪器显示语言。:STYLE:LANGUage?查询当前显示语言。

命令语法: SYSTem:STYLe:LANGUage {ENGLISH | Chinese}

查询语法: SYSTem:STYLe:LANGUage?

查询返回: {EN | CH}<NL>

:STYLE:KSOUnd 用于设定按键声音开关。:STYLE:KSOUnd?查询按键声音开关状态。

命令语法: SYSTem:STYLe:KSOUnd {{1 | ON} | {0 | OFF}}

查询语法: SYSTem:STYLe:KSOUnd?

查询返回: {1 | 0}<NL>

:BEEPer:PASS 设定比较合格时的报警方式。:BEEPer:PASS?查询仪器当前设定的比较合格时的报警方式。

命令语法: SYSTem:BEEPer:PASS {OFF | LHIGH | LLOW | SSHort | TSHort}

这里: OFF 表示关闭报警; LHIGH 表示长高音报警; LLOW 表示长低音报警; SSHort 表示短音报警; TSHort 表示双短音报警。

查询语法: SYSTem:BEEPer:PASS?

查询返回: {OFF | LHIG | LLOW | SSH | TSH}<NL>

:BEEPer:FAIL 设定比较不合格时的报警方式。:BEEPer:FAIL?查询仪器当前设定的比较不合格时的报警方式。

命令语法: SYSTem:BEEPer:FAIL {OFF | LHIGH | LLOW | SSHort | TSHort}

这里: OFF 表示关闭报警; LHIGH 表示长高音报警; LLOW 表示长低音报警; SSHort 表示短音报警; TSHort 表示双短音报警。

查询语法: SYSTem:BEEPer:FAIL?

查询返回: {OFF | LHIG | LLOW | SSH | TSH}<NL>

:HANDler:TEDGe 用于设定 handler 接口触发信号的边沿。:HANDler:TEDGe?查询当前设定的 handler 接口触发边沿。

命令语法: SYSTem:HANDler:TEDGe {RISing | FALLing}

其中: RISing 表示上升沿触发; FALLing 表示下降沿触发。

查询语法: SYSTem:HANDler:TEDGe?

查询返回: {RIS| FALL}<NL>

:HANDler:OUTPut 用于设定 handler 接口信号输出模式。:HANDler:OUTPut?查询当前设定的 handler 接口信号输出模式。

命令语法: SYSTem:HANDler:OUTPut {HOLD | CLEar}

其中: HOLD 表示输出信号保持; CLEar 输出信号延时清除。

查询语法: SYSTem:HANDler:OUTPut?

查询返回: {HOLD| CLE}<NL>

:HANDler:DELay 用于设定 handler 接口信号输出延时时间。:HANDler:DELay?查询当前设定的 handler 接口信号输出延时时间。

命令语法: SYSTem:HANDler:DELay <value>

其中: value 是 NR3 格式, 时间单位或不加单位, 范围是 0~60s, 1ms 步进

查询语法: SYSTem:HANDler:DELay?

查询返回: <NR3><NL>

:PRESet 用于调出初始的系统配置参数, 密码状态除外。

命令语法: SYSTem:PRESet

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数据超出范围等，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 未知命令	无法识别的消息，一般是拼写错误或者仪器不支持的命令 例如: TRG 应该为 TRIG DISP:PAG MEAS 应该为 DISP:PAGE MEAS
Data out of range 数据超出范围	数据超限 例如: TRIG:DEL 66s, 触发延时时间超出范围 0~60s
Invalid parameter 无效参数	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 无效后缀	后缀错误，一般是不匹配的单位 例如:TRIG:DEL 200us, us 是时间不支持的单位。
Data too long 数据太长	数据太长，例如文件名超过 17 个字符
Syntax error 语法错误	系统错误，不符合 SCPI 语法格式，如 DISP 命令后面应该是冒号(:)，如果不是就是系统错误
Missing parameter 参数丢失	未给定命令的参数，或者参数不全
Character data not allowed 非法字符	不允许输入字符的场合
Numeric data not allowed 非法数字	不允许输入数字的场合
Command ignored 命令忽略	命令正确，但是不执行 例如在非总线触发模式时发送 TRIG 命令

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。

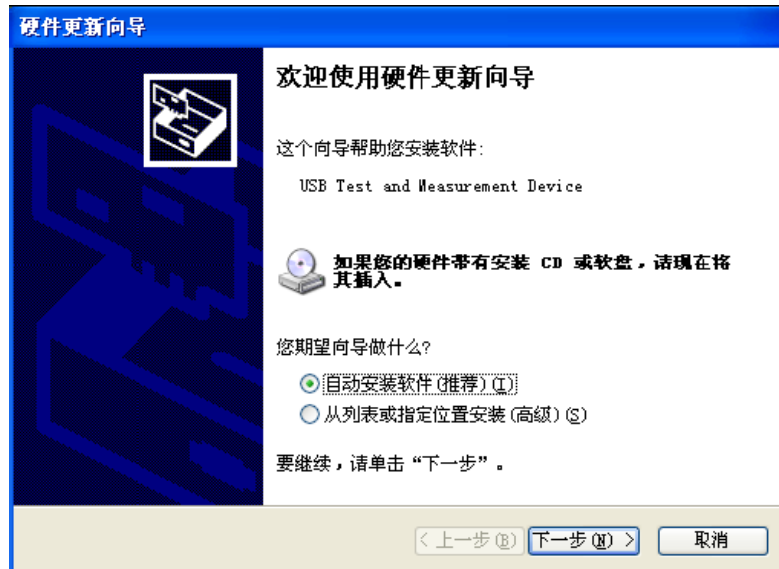


图 4-1

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。

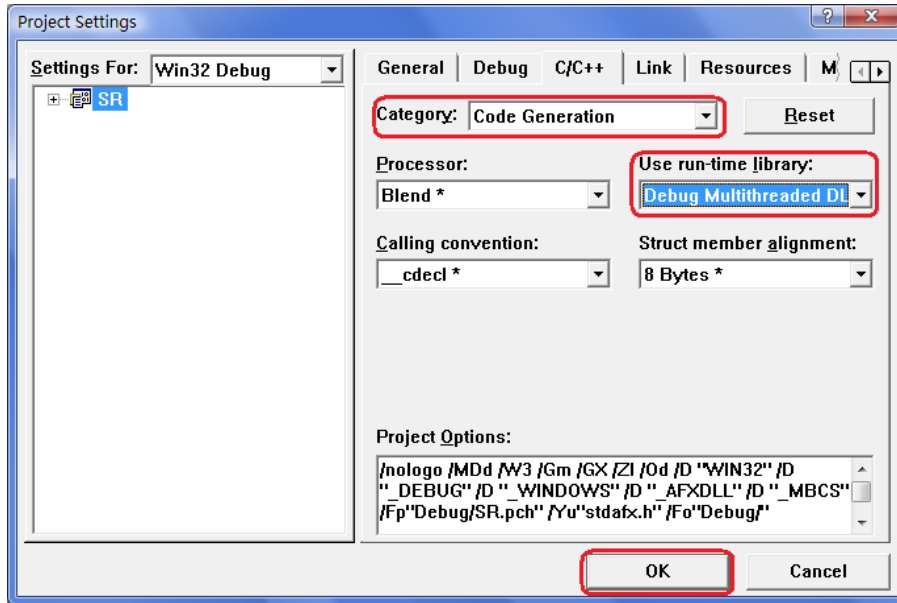


图 4-1-1

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。

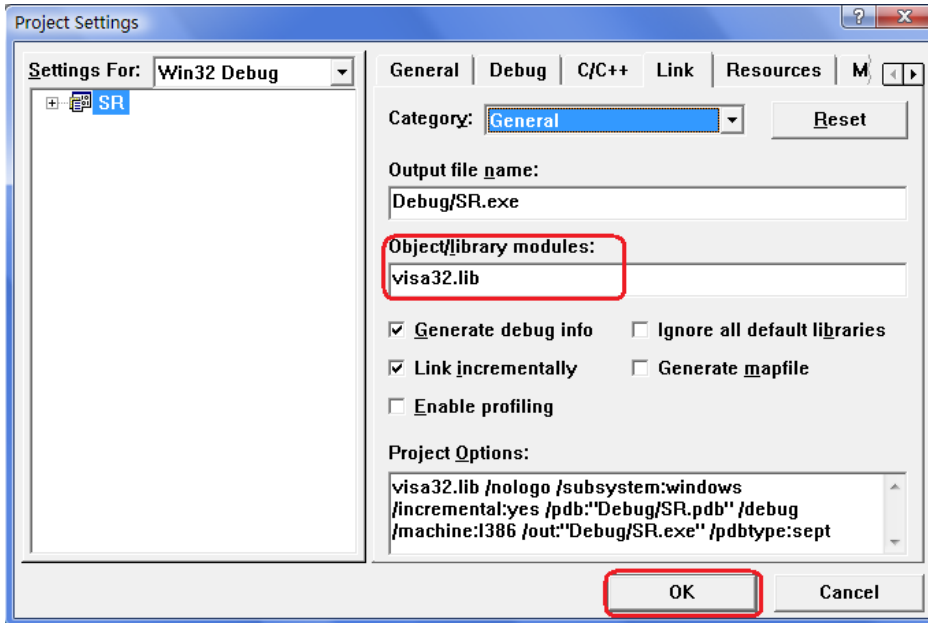


图 4-1-2

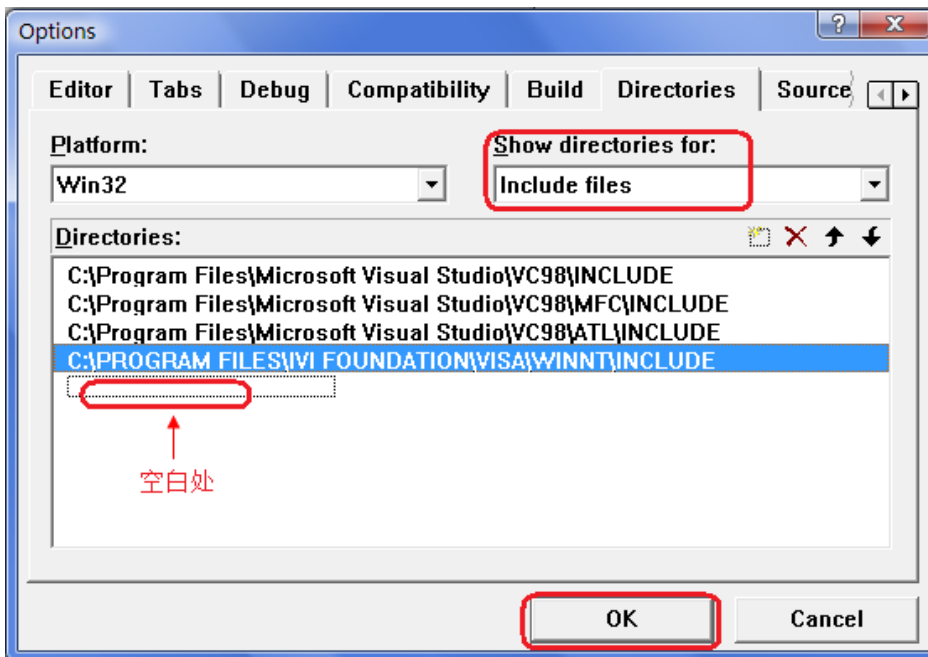


图 4-1-3

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

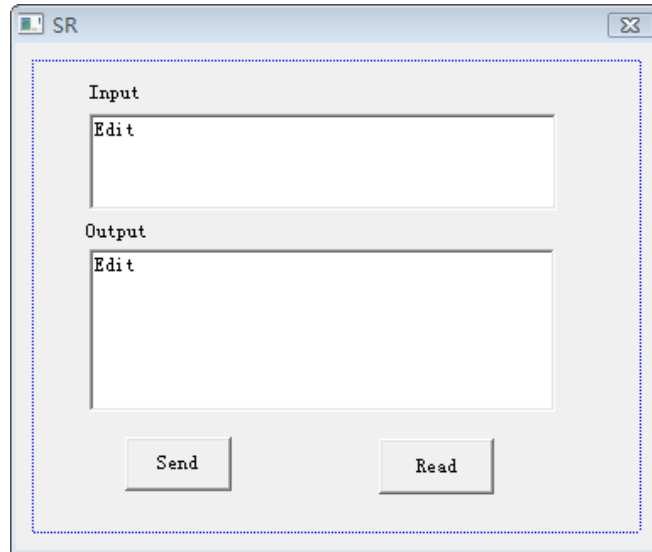


图 4-1-4

(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。

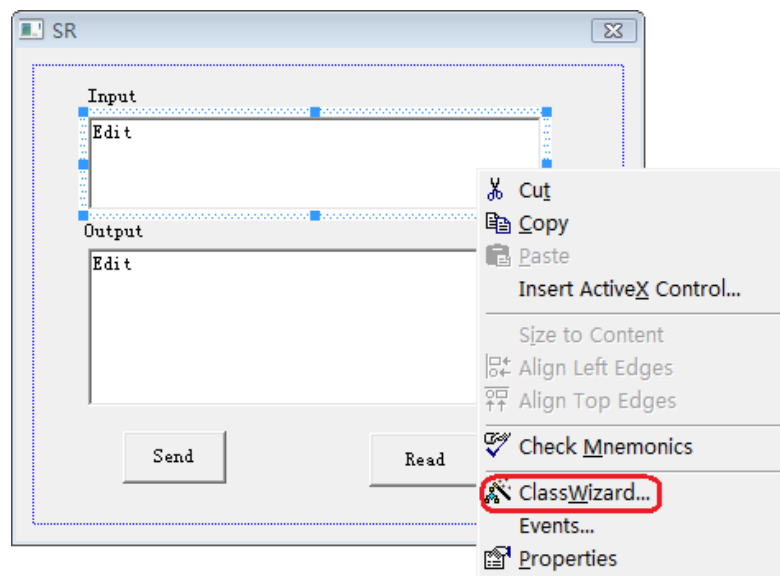


图 4-1-5

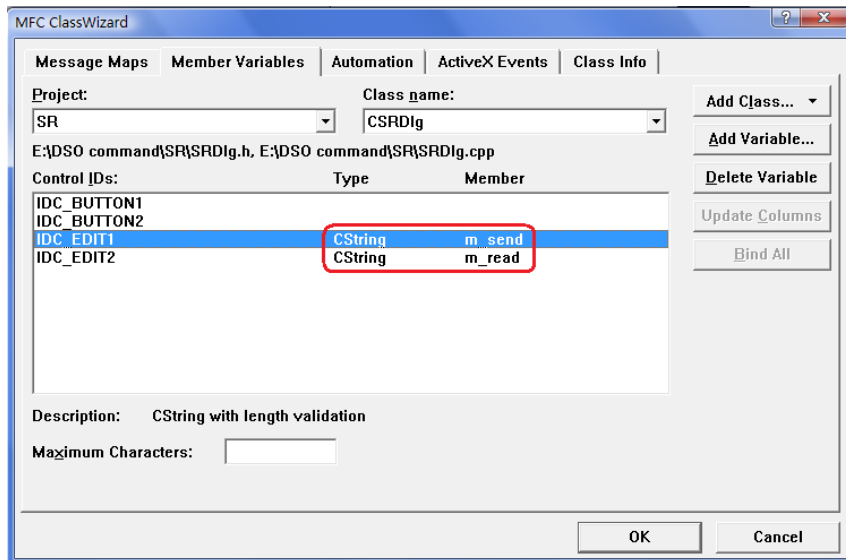


图 4-1-6

(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。

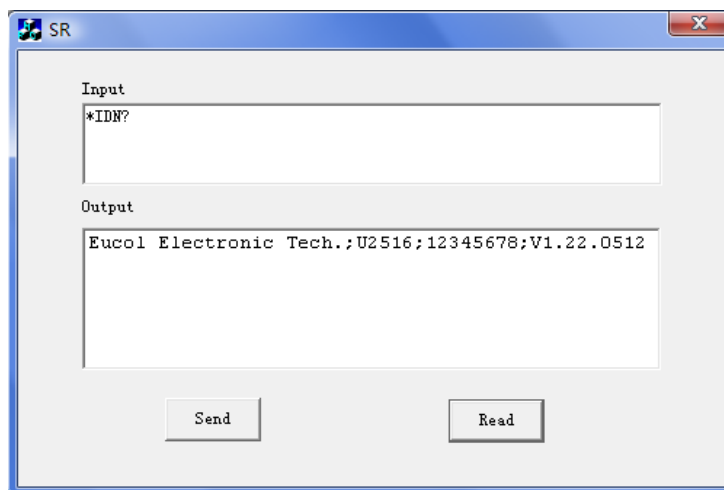


图 4-1-7

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。

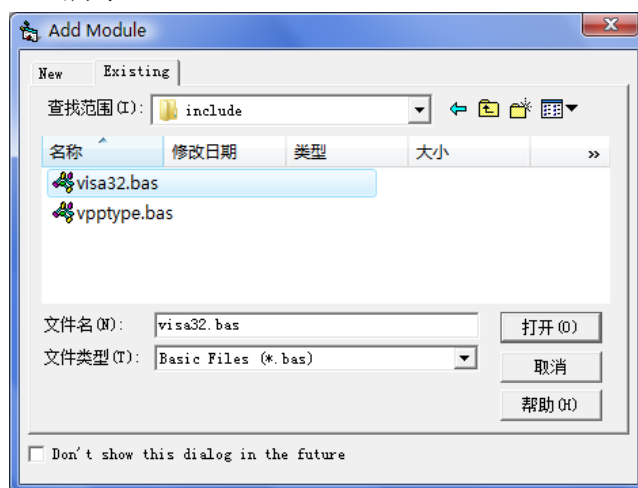


图 4-2-1

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。

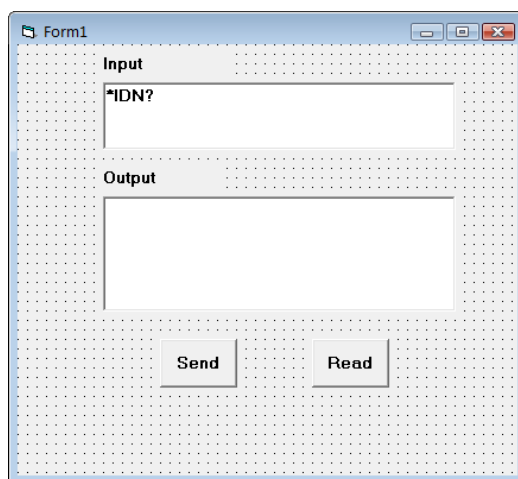


图 4-2-2

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。

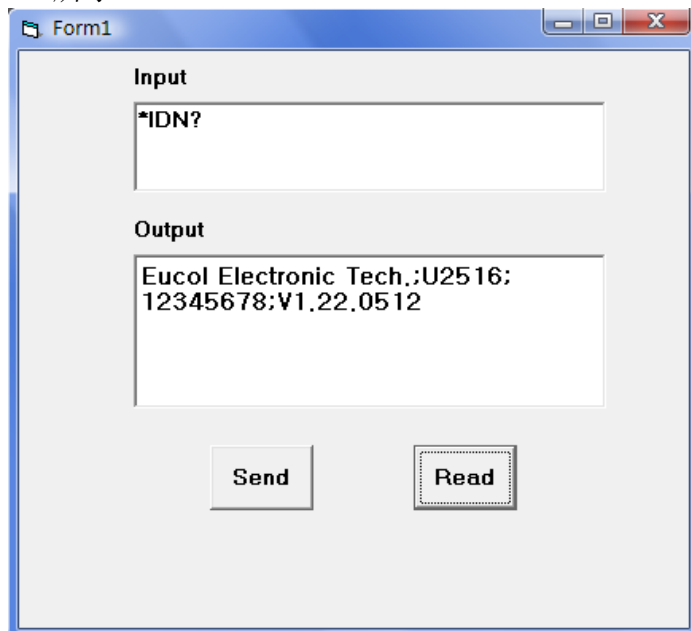


图 4-2-3

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。

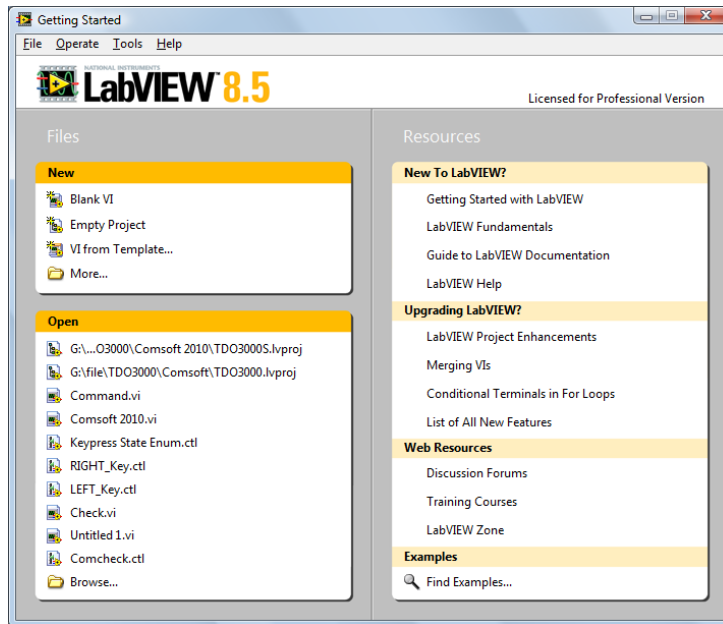


图 4-3-1

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。

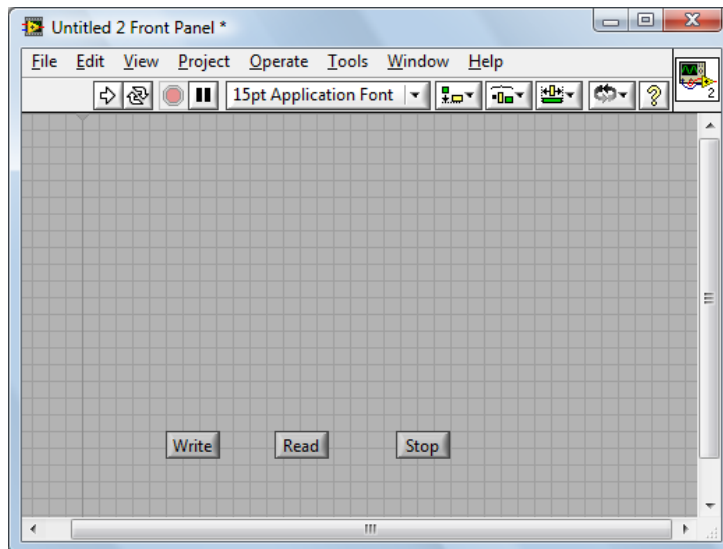


图 4-3-2

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。

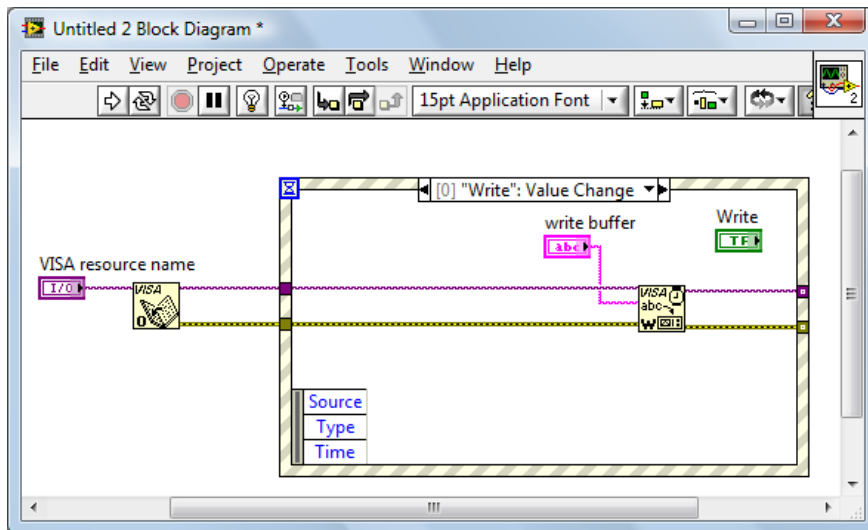


图 4-3-3

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。

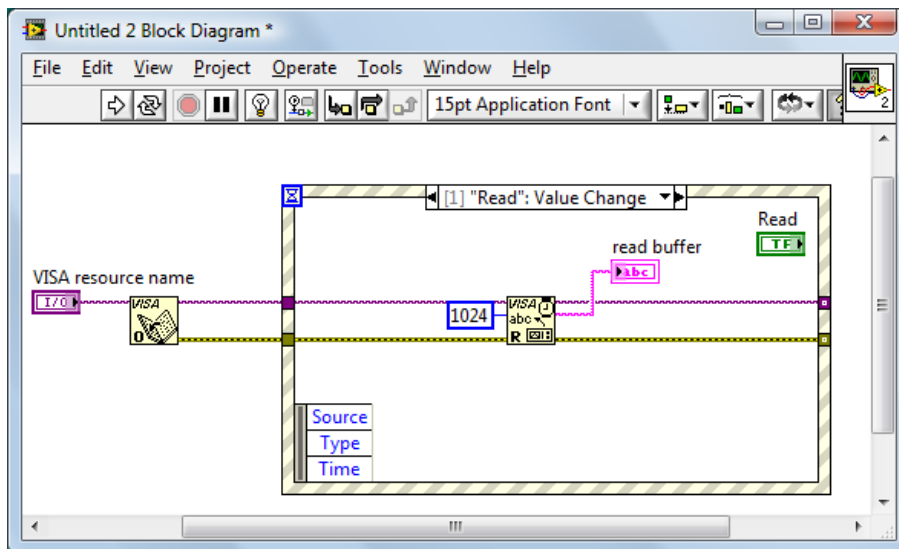


图 4-3-4

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。

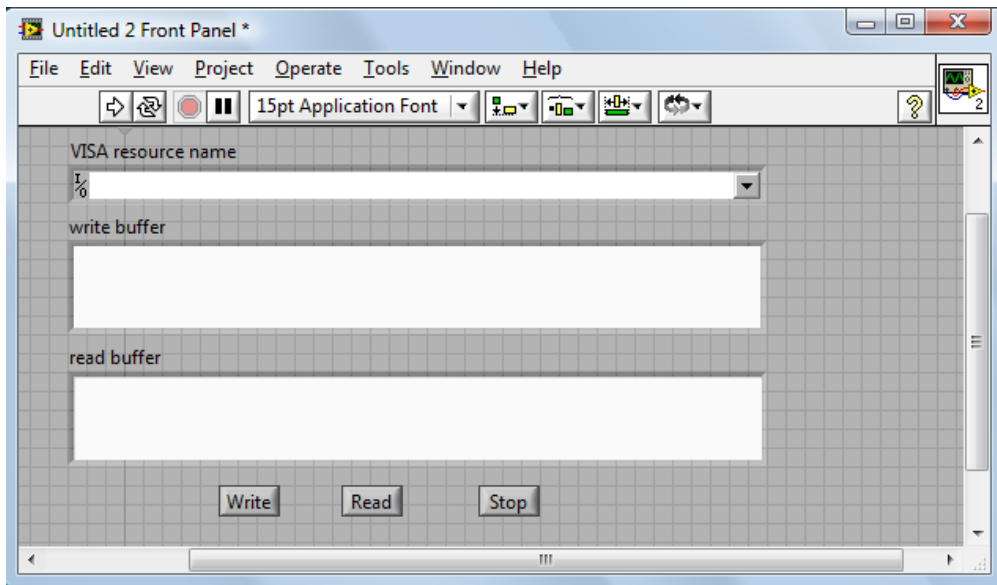


图 4-3-5

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。

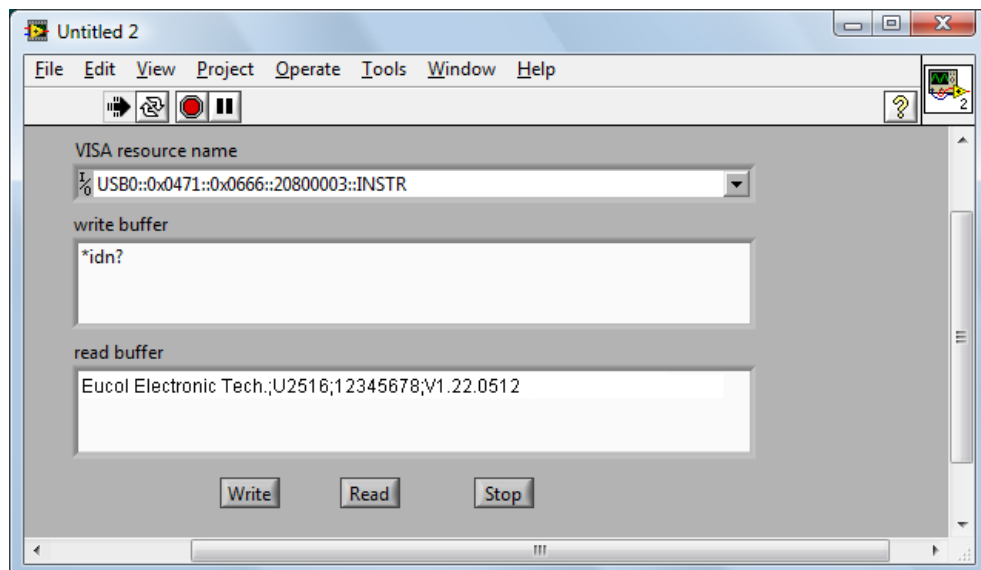


图 4-3-6

23. 按 Stop 键退出程序执行。