

# UTR2832 系列 LCR 数字电桥

## （编程手册）

优利德科技（中国）股份有限公司

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
1.3	单位的倍率 .....	2
2	命令系统 .....	3
2.1	公用命令 .....	4
2.2	显示系统命令 .....	8
2.3	中止测试系统命令 .....	9
2.4	初始系统命令 .....	9
2.5	触发系统命令 .....	10
2.6	功能系统命令 .....	11
2.7	频率系统命令 .....	13
2.8	电压系统命令 .....	13
2.9	测量速度系统命令 .....	13
2.10	源内阻系统命令 .....	13
2.11	数据读取系统命令 .....	14
2.12	校准系统命令 .....	15
2.13	比较系统命令 .....	17
2.14	数据存储系统命令 .....	21
2.15	文件存储系统命令 .....	22
2.16	按键系统命令 .....	23
2.17	系统配置系统命令 .....	24
3	出错信息 .....	26
4	编程实例 .....	27
4.1	Visual C++ 6.0 编程实例 .....	29
4.2	Visual Basic 6.0 编程实例 .....	34
4.3	LabVIEW 8.5 编程实例 .....	37

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, 只能从几个项目中选择一个。
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: IMPedance 缩写为 IMP。

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。

### 1.3 单位的倍率

向仪器发送的命令参数是数据时，可以使用单位及单位的倍率。倍率的定义及符号参见下表：

定    义	符    号	本仪器使用
1E18 (EXA)	EX	ý
1E15 (PETA)	PE	ý
1E12 (TERA)	T	ý
1E9 (GIGA)	G	ý
1E6 (MEGA)	MA*	p
1E3 (KILO)	K	p
1E-3 (MILI)	M	p
1E-6 (MICRO)	U	p
1E-9 (NANO)	N	p
1E-12 (PICO)	P	p
1E-15 (FEMTO)	F	ý
1E-18 (ATTO)	A	ý

\*注：由于本仪器在命令及参数中不区分大小写，因此这里与标准的符号定义有所区别。

---

注意：倍率一般要求和单位一起使用，而不能单独存在。但阻抗类参数的单位已经因仪器设置的测量功能而被隐含定义，这些数据就不需要带单位，而直接使用倍率 ma, k, m, u, n, p 这些倍率。

---

例如标称值单位一般取决于一次参数：

命令 COMP:TOL:NOM 100m

当一次参数是 Z 时，表示 100mW，一次参数是 Ls 时，表示 100mH。

其它可以带单位的数据场合：

单位及倍率	应用
Hz, kHz, mHz	频率单位，命令如 FREQ
V,mV	电压单位，命令如 VOLT,VOLT:LEV
s,ms	时间单位，命令如 TRIG:DElay
mohm,ohm,kohm maohm	电阻单位，源内阻，量程等

## 2 命令系统

UTR2832 系列 LCR 数字电桥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
- ◆ 中止测试(ABORt)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
- ◆ 频率(FREQuency)系统命令
- ◆ 电压(VOLTage)系统命令
- ◆ 测量速度(APERture)系统命令
- ◆ 源内阻(Source RESister)系统命令
- ◆ 数据读取(FETCH)系统命令
- ◆ 校准(CORRection)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。

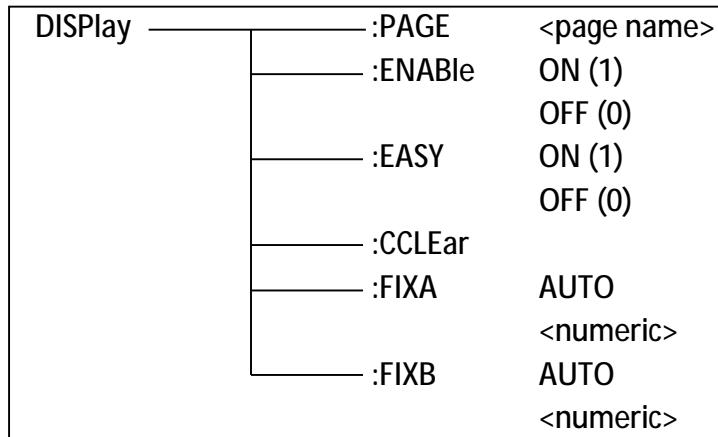


表 2-2-1 显示系统命令树

**: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)
BDISplay	设定显示页面至: 分选显示页面(Bin DISplay)
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 | BDIS | 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**

**:FIXA** 设定第一参数的小数位锁定值。

命令语法: **DISPlay:FIXA {<numeric> | AUTO}**

其中: **numeric** 范围是 0~5, 0 和 **AUTO** 表示自动。

**:FIXB** 设定第二参数的小数位锁定值。

命令语法: **DISPlay:FIXB {<numeric> | AUTO}**

其中: **numeric** 范围是 0~5, 0 和 **AUTO** 表示自动。

## 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** 命令控制仪器触发。表 2-4-1 是统计系统命令树。

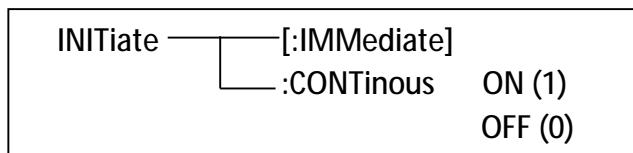


表 2-4-1 初始系统命令树

**[:IMMEDIATE]** 允许仪器接收一次触发命令(**\*TRG**)。

命令语法: **INITiate [:IMMEDIATE]**

**:CONTinuous** 设置仪器是否允许触发命令持续有效。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:CONTinuous {{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。

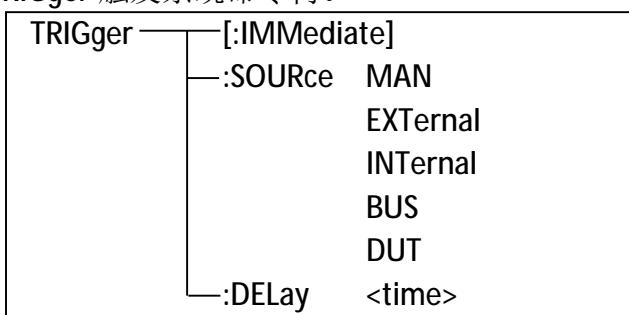


表 2-5-1 触发系统命令树

**[:IMMEDIATE]** 用于触发一次测量。

命令语法: TRIGger[:IMMEDIATE]

i 注意: 此命令只在测量显示页面, 分选页面和列表扫描页面有效, 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, DUT 表示器件触发。

查询语法: TRIGger:SOURce?

查询返回: {MAN | HOLD | EXT | INT | BUS | DUT}<NL>

**:DElay** 设定触发延时时间。:DElay?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。

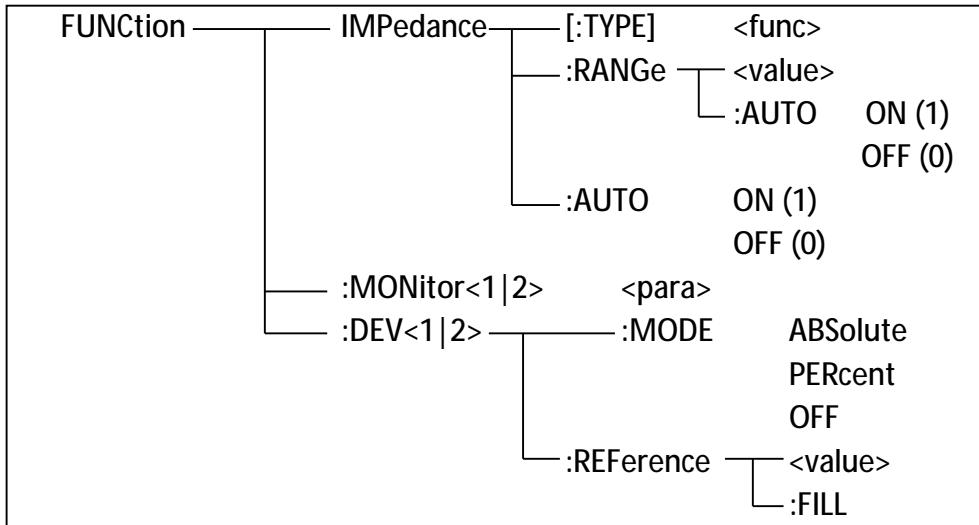
命令语法: TRIGger:DElay <time> time 是 NR1, NR2 或 NR3 格式, 可跟时间单位

查询语法: TRIGger:DElay?

查询返回: <NR3><NL>

## 2.6 功能系统命令

**FUNCTION** 功能系统命令主要用于选择测量参数，设置量程，选择监视参数和设置偏差模式。表 2-6-1 是 **FUNCTION** 系统命令树。



**IMPedance[:TYPE]** 用于设定仪器的测量参数。 **IMPedance[:TYPE]?** 用于查询仪器当前设定的测量参数。

命令语法: **FUNCTION:IMPedance[:TYPE] {CPD | CPRP | CSD | CSRS | LSQ | LSRS | LPQ | LPRP | RX | ZTD | ZTR | GB}**

其中: ZTD 表示  $Z - \theta^\circ$  , ZTR 表示  $Z - \theta \text{ r}$

查询语法: **FUNCTION:IMPedance[:TYPE]?**

查询返回: {CPD|CPRP|CSD|CSRS|LSQ|LSRS|LPQ|LPRP|RX|ZTD |ZTR|GB}<NL>

**IMPedance:RANGE** 用于设定仪器的量程，并锁定量程。 **IMPedance:RANGE?** 用于查询仪器的量程。

命令语法: **FUNCTION:RANGE { 1 | 10 | 30 | 100 | 300 | 1000 | 3000 | 10000 | 100000 | 1000000 | 1ohm | 10ohm | 30ohm | 100ohm | 300ohm | 1kohm | 10kohm | 100kohm | 1maohm}**

查询语法: **FUNCTION:RANGE?**

查询返回: {1|10|30|100|300|1000|3000|10000|100000|1000000}<NL>

i 注意: 不同型号仪器的量程范围有所不同, 实际量程请参考仪器说明书。

**IMPedance:RANGE:AUTO** 用于设定仪器的量程自动选择开关。

**IMPedance: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>

**IMPedance:AUTO** 用于设定自动选择测量参数 (AUTO LCR)。 **IMPedance:AUTO?**

查询当前设定的自动参数开关状态。

命令语法: **FUNCTION:IMPedance:AUTO { {1 | ON} | {0 | OFF}}**

查询语法: **FUNCTION:IMPedance:AUTO?**

查询返回: {1 | 0}<NL>

**:MONitor<n>** 用于设定监视参数。 **:MONitor<n>?** 用于查询当前设定的监视参数。

命令语法: **FUNCTION:MONitor<n> {CS | CP | LS | LP | RS | RP | R | D | Q | Z | RAD | DEG | X | Y | G | B | V | I}**

其中: n 是 1 和 2, 表示监视参数 1 和监视参数 2。

查询语法: **FUNCTION:MONitor<n>?**

查询返回: {CS|CP|LS|LP|RS|RP|R|D|Q|Z|RAD|DEG|X|Y|G|B|V|I}<NL>

**:DEV<n>:MODE** 用于设定仪器的测量结果偏差模式。 **:DEV<n>:MODE** 用于查询当前仪器的测量结果偏差模式。

命令语法: **FUNCTION:DEV<n>:MODE { ABSolute | PERcent | OFF }**

这里: ABSolute 绝对值偏差显示

PERCent 百分比偏差显示

OFF 关闭偏差显示

n 1 或者 2, 表示主参数和副参数

查询语法: **FUNCTION:DEV<n>:MODE?**

查询返回: {ABS | PER | OFF}<NL>

**:DEV<n>:REFerence[:VALue]** 用于设定仪器的偏差标称值。

**:DEV<n>:REFerence[:VALue]?** 用于查询当前仪器的偏差标称值。偏差模式是绝对值偏差和百分比偏差的时候需要使用偏差标称值。

命令语法: **FUNCTION:DEV<n>:REFerence <value>**

这里: <value> 可以是 NR1、NR2 或 NR3 数据格式

n 是 1 或者 2, 表示主参数和副参数的偏差标称值

查询语法: **FUNCTION:DEV<n>:REFerence?**

查询返回: <NR3><NL>

**:DEV<n>:REFerence:FILL** 用于设定仪器的偏差标称值, 该命令将最近一次的测量结果分别输入到主参数和副参数的偏差参考中。

命令语法: **FUNCTION:DEV<n>:REFerence:FILL**

## 2.7 频率系统命令

**FREQuency** 频率系统命令主要用于设定仪器的测量频率。

**FREQuency?** 查询当前设定的测量频率。

命令语法: **FREQuency {<value> | MAX | MIN}**

其中: <value> 可以是 NR1,NR2,NR3 数据格式, 可加频率单位

MIN 设定测量频率为 50Hz

MAX 设定测量频率为 100kHz 或 200kHz (和仪器有关)

查询语法: **FREQuency?**

查询返回: <NR3><NL>

## 2.8 电压系统命令

**VOLTage** 电压系统命令主要用于设定仪器的测量电平。

**VOLTage[:LEVel]?** 查询当前设定的测量电平。

命令语法: **VOLTage[:LEVel] {<value> | MAX | MIN}**

其中: value 可以是 NR1,NR2,NR3 数据格式, 可加电压单位

MIN 设定测量频率为 0.1V

MAX 设定测量频率为 1V 或 2V (和仪器有关)

查询语法: **VOLTage[:LEVel]?**

查询返回: <NR3><NL>

## 2.9 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>

## 2.10 源内阻系统命令

**Source RESister** 源内阻系统命令用于设定仪器的输出内阻, **SRESister?** 查询仪器当前设定的源内阻。

命令语法: **SRESister {30 | 50 | 100}**

查询语法: **SRESister?**

查询返回: {30 | 50 | 100}<NL>

## 2.11 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 2-11-1 是数据读取系统命令树。

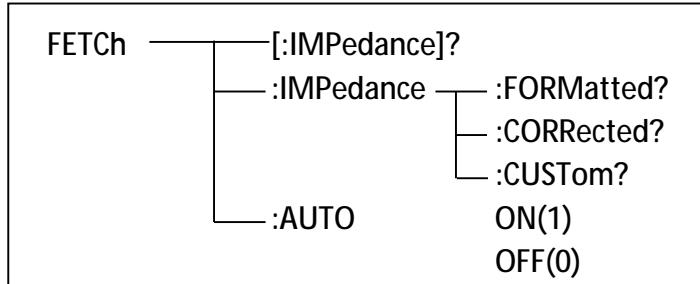


表 2-11-1 数据读取系统命令树

**[:IMPedance]?**查询当前测量参数的测量结果。

查询语法: **FETCh[:IMPedance]?**

查询返回:

在测量显示和分选显示页面, 查询返回的数据输出格式描述如下:

**SN.NNNNNESNN** | **SN.NNNNNESNN** | **SN** | **SN** **NL**  
<DATA A>            <DATA B>    <状态><档号>

其中:

<DATA A>是主参测量数据

<DATA B>是副参测量数据

主副参数使用 12 位 ASCII 码格式 (SN.NNNNNESNN, S: +/-, N: 0 到 9, E: Exponent Sign(指数标志))。

<状态>是测量状态, 0 表示测量正常, 其他表示错误。

<档号>输出分选比较结果

数值	分选结果
0	未比较
1~3	档号 BIN1~3
10	不合格档 OUT
11	辅助档 AUX
其它	非法输出

**:IMPedance:FORMatted?**

查询语法: **FETCh:IMPedance:FORMatted?**

查询返回: 同 **FETCh[:IMPedance]?**

**:IMPedance:CORRected?**查询返回修正的 R-X 类型测量结果，和当前设定的测量参数和测量页无关。

查询语法: **FETCH:IMPedance:CORRected?**

查询返回: <DATA A>,<DATA B><NL>

DATA A 和 DATA B 分别表示主参数和副参数，返回结果如前所述。

**:IMPedance:CUSTom?**查询返回所有测量结果参数。

查询语法: **FETCH:IMPedance:CUSTom?**

查询返回: <DATA A>,<DATA B>,<MON 1>,<MON 2>,<状态>,<档号><NL>

该查询指令和 **FETCH[:IMPedance]?**的区别在于该命令还返回两个监视参数。

**: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>

**F 提示:** 自动查询功能打开时，仪器每次测试后自动返回测量结果，返回格式同 **FETCH[:IMPedance]?**。

## 2.12 校准系统命令

**CORRect** 校准系统命令用于设定开路和短路校准。

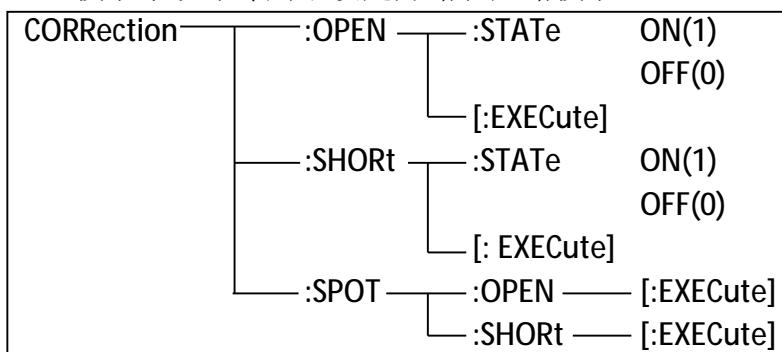


表 2-12-1 校准系统命令树

**:OPEN[:EXECute]** 用于执行开路全频清零，无查询。

命令语法: **CORRect:OPEN[:EXECute]**

**注意:** 清零期间，来自总线的命令被暂停响应（挂起），清零命令完成后，再继续执行后面的命令。

---

说明：可以在需要执行一段时间的命令后，紧跟一个\*OPC?命令，命令完成后返回 1。如 CORR:OPEN:EXEC;\*OPC?

---

**:OPEN:STATe** 用于设置仪器开路修正开关。:OPEN:STATe?用于查询当前仪器设定的短路修正开关状态。

命令语法：CORRection:OPEN:STATe {{1 | ON} | {0 | OFF}}

查询语法：CORRection:OPEN:STATe?

查询返回：{1 | 0}<NL>

**:SHORt[:EXECute]** 用于执行短路全频清零，无查询。

命令语法：CORRection:SHORt[:EXECute]

注意：清零期间，来自总线的命令被暂停响应（挂起），清零命令完成后，再继续执行后面的命令。

---

说明：可以在需要执行一段时间的命令后，紧跟一个\*OPC?命令，命令完成后返回 1。如 CORR:SHOR:EXEC;\*OPC?

---

**:SHORt:STATe** 用于设置仪器短路修正开关。:SHORt:STATe?用于查询当前仪器设定的短路修正开关状态。

命令语法：CORRection:SHORt:STATe {{1 | ON} | {0 | OFF}}

查询语法：CORRection:SHORt:STATe?

查询返回：{1 | 0}<NL>

**:SPOT:OPEN[:EXECute]** 用于执行当前频率点的开路校准。

命令语法：CORRection:SPOT:OPEN[:EXECute]

注意：清零期间，来自总线的命令被暂停响应（挂起），清零命令完成后，再继续执行后面的命令。

**:SPOT:SHORt[:EXECute]** 用于执行当前频率点的短路校准。

命令语法：CORRection:SPOT:SHORt[:EXECute]

注意：清零期间，来自总线的命令被暂停响应（挂起），清零命令完成后，再继续执行后面的命令。

## 2.13 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 2-13-1 是 **COMParator** 比较系统命令树。

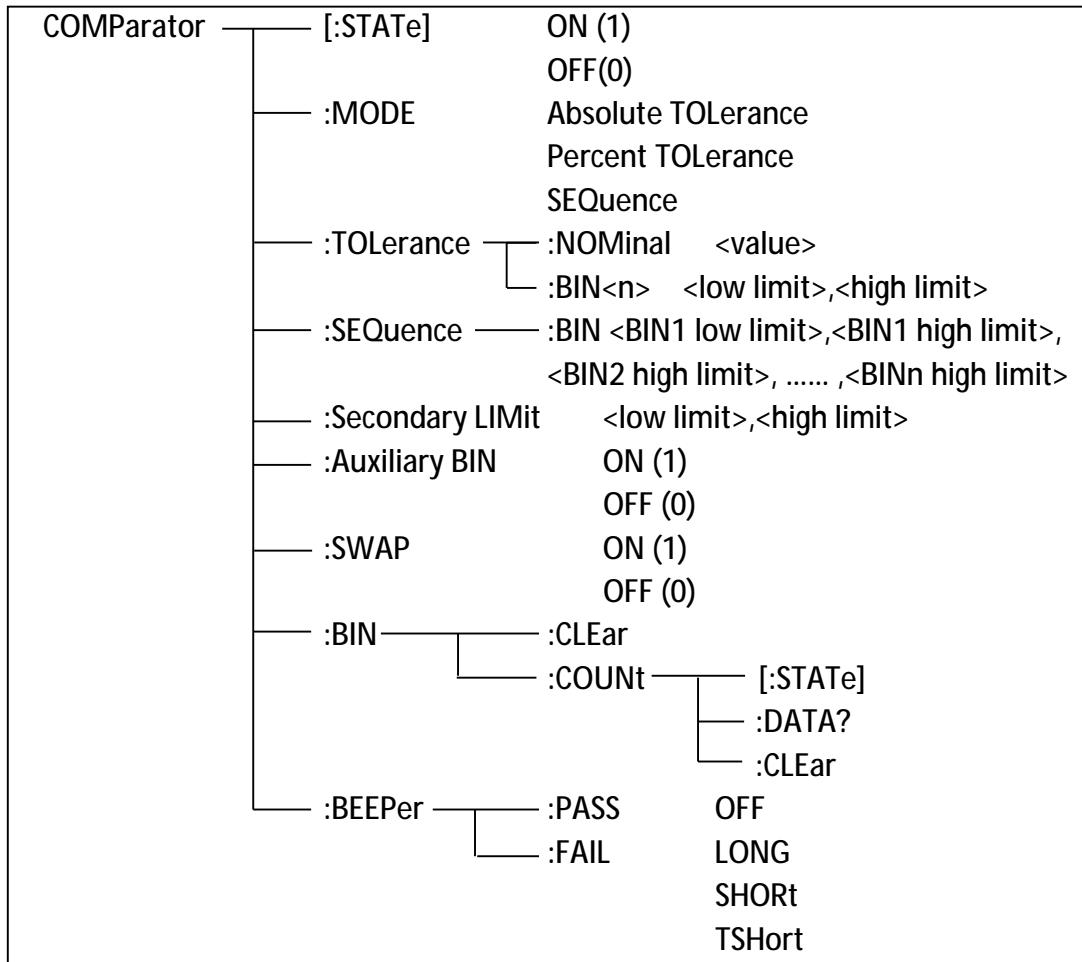


表 2-13-1 比较系统命令树

**[: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 | 0}<NL>

**:MODE** 用于设定仪器比较的极限方式。:MODE?查询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。

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)。

**: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。

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) 是档号数

<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。

**F 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。

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。

**:Secondary LIMit** 用于设定仪器比较功能副参数上下限数值。SLIMit?查询仪器当前副参数上下限数值。

命令语法: COMParator:SLIMit <low limit>,<high limit>

这里:

<low limit> NR1, NR2 或 NR3 格式数据, 副参数下限数值

<high limit> NR1, NR2 或 NR3 格式数据, 副参数上限数值

注: 上限应大于下限, 否则提示出错信息。

查询语法: COMParator:SLIMit?

查询返回: <NR3>,<NR3><NL>

**:Auxiliary BIN** 用于设定档计数辅助档开关。ABIN?查询仪器当前附属档开关状态。

命令语法: COMParator:ABIN {{1 | ON} | {0 | OFF}}

查询语法: COMParator:ABIN?

查询返回: {1 | 0}<NL>

**:SWAP** 用于设定主副参数对调比较模式开关, 例如功能参数 Cp-D, 选择 SWAP 模式为 ON, 则功能参数变为 D-Cp; 此时 1~3 档参数极限设置变为设定 D 的上下限, 2and 档设置 Cp 极限。: SWAP?查询当前仪器设定主副参数对调比较模式开关状态。

命令语法: COMParator:SWAP {{1 | ON} | {0 | OFF}}

查询语法: COMParator:SWAP?

查询返回: {1 | 0}<NL>

**: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

i 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>,<OUT BIN count>,<AUX BIN count><NL>

这里: **BIN1~BINn count** 是 1~n 档的档计数结果。  
**OUT BIN count** 是超差档的档计数结果。  
**AUX BIN count** 是辅助档的档计数结果。

**: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 | LONG | SHORt | TSHort}**

这里: OFF 表示关闭报警; LONG 表示长音报警; SHORt 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 | LONG | SHOR | 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 | LONG | SHORt | TSHort }**

这里: OFF 表示关闭报警; LONG 表示长音报警; SHORt 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 | LONG | SHOR | TSH}<NL>

## 2.14 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 2-14-1 是 **MEMory** 数据存储系统命令树。

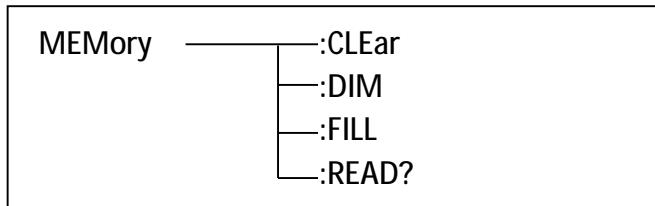


表 2-14-1 数据存储系统命令

**: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?** 返回的数据一样，每组测试数据之间使用换行符 (“\r\n”) 隔开。

## 2.15 文件存储系统命令

Mass **MEM**ory 文件存储系统命令用于文件的保存与加载。表 2-15-1 是 Mass **MEM**ory 系统命令树。

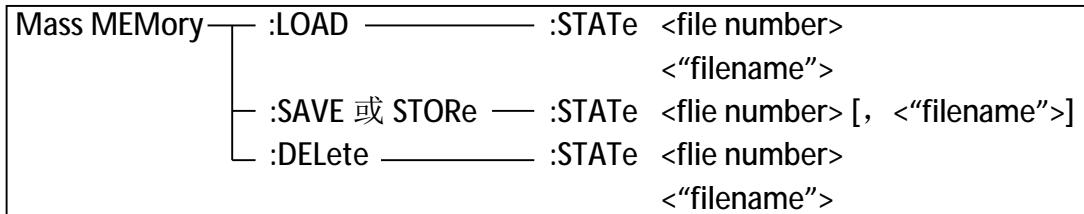


表 2-15-1 文件存储系统命令树

**:LOAD:STATE** 命令用于加载已保存的文件。

命令语法: **MMEM**ory:**LOAD:STATE** <file number>

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

i 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: **MMEM**ory:**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。

i 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: **MMEM**ory:**STORe:STATE** <file 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。

i 注意: 本仪器在覆盖已存在的文件记录时并不会提示。

F 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: **MMEM**ory:**DELETE:STATE** <file number>

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

i 注意: 仪器删除存在的文件记录时并不会提示。

命令语法: **MMEM**ory:**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。

## 2.16 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

## 2.17 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 2-17-1 是系统配置系统命令树。

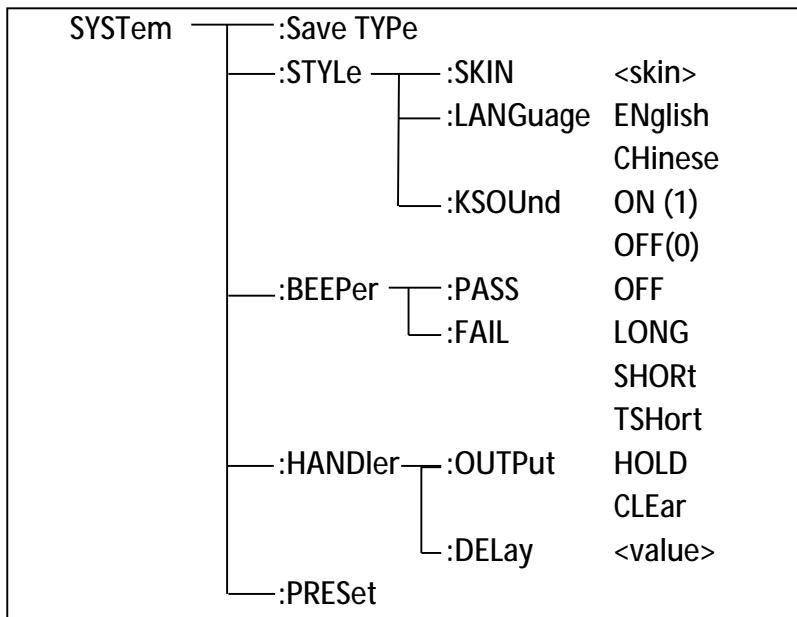


表 2-17-1 系统配置系统命令树

**: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 | LONG | SHORt | TSHort }  
这里: OFF 表示关闭报警; LONG 表示长音报警; SHORt 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 | LONG | SHOR | 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 | LONG | SHORt | TSHort }  
这里: OFF 表示关闭报警; LONG 表示长音报警; SHORt 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 | LONG | SHOR | TSH}<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 表示输出信号延时清除。  
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
<b>Undefined message</b> 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
<b>Data out of range</b> 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
<b>Invalid parameter</b> 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
<b>Invalid suffix</b> 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。
<b>Data too long</b> 数据太长	数据太长，例如文件名超过 17 个字符
<b>Syntax error</b> 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
<b>Missing parameter</b> 参数丢失	未给定命令的参数，或者参数不全
<b>Character data not allowed</b> 非法字符	不允许输入字符的场合
<b>Numeric data not allowed</b> 非法数字	不允许输入数字的场合
<b>Command ignored</b> 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。

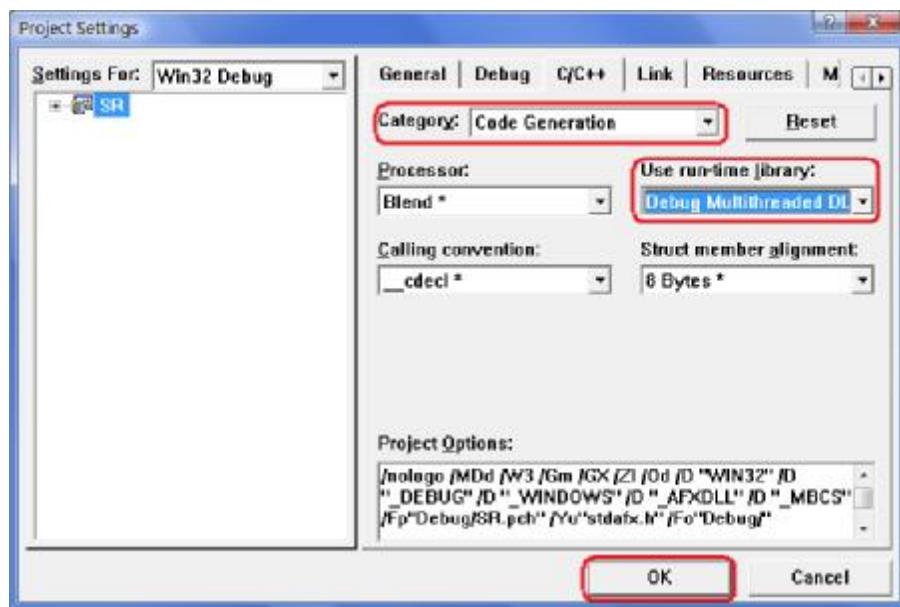


图 4-1-1

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。

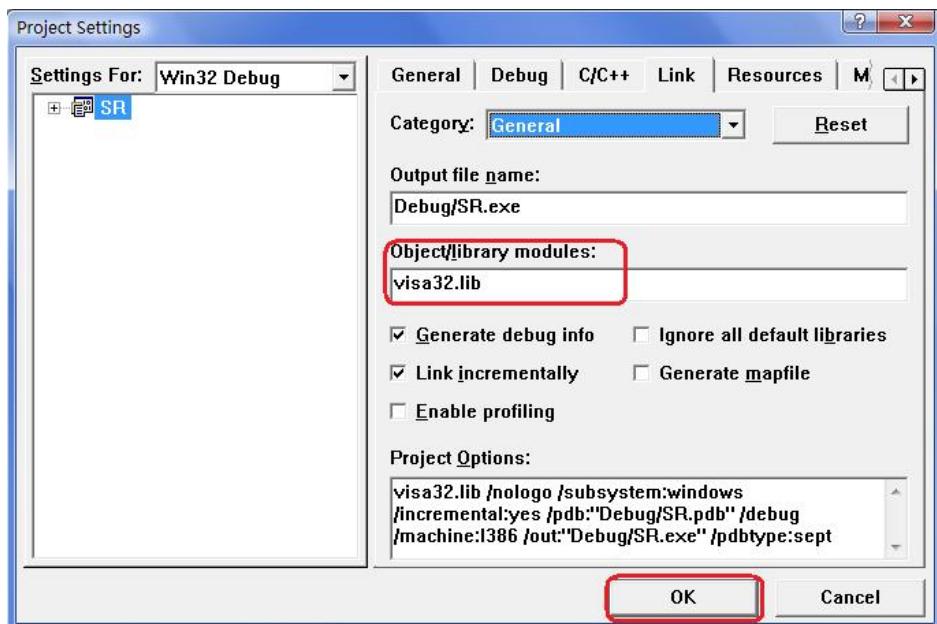


图 4-1-2

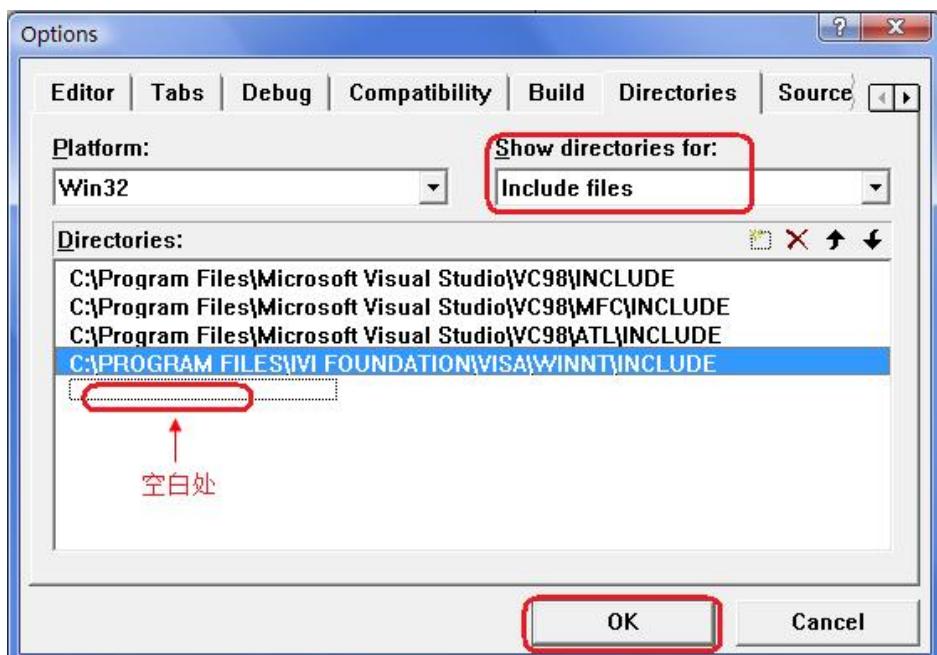


图 4-1-3

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

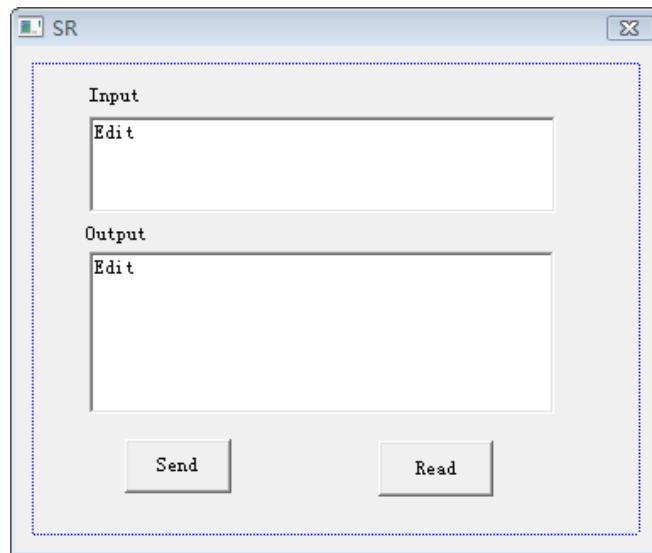


图 4-1-4

- (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。

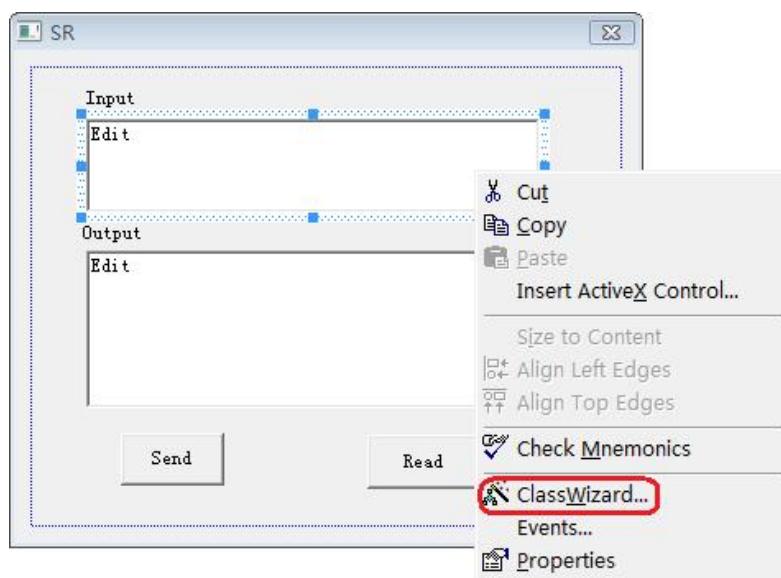


图 4-1-5

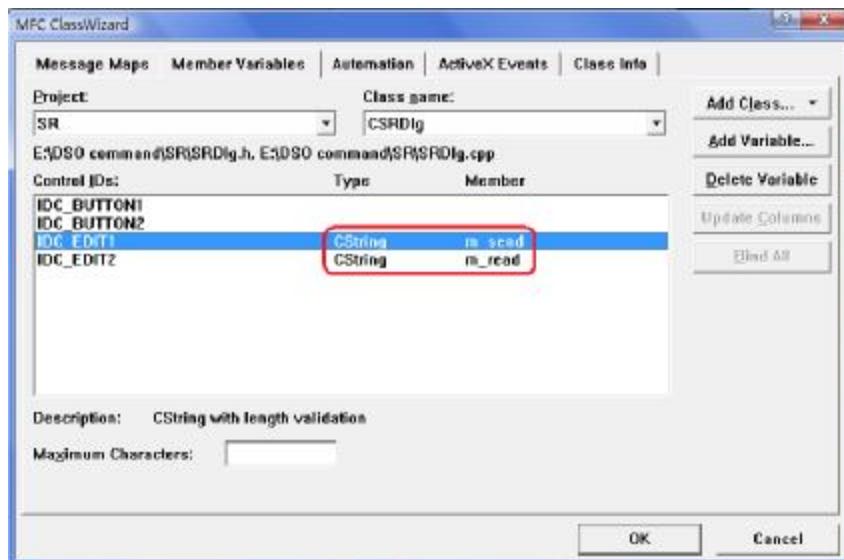


图 4-1-6

(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\_BUFLEN];

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。

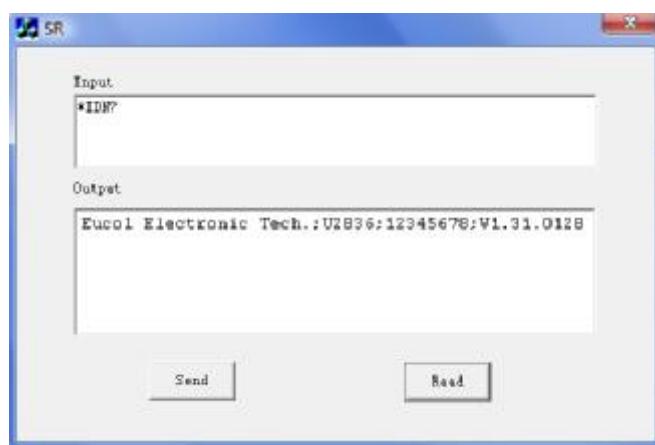


图 4-1-7

## 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。

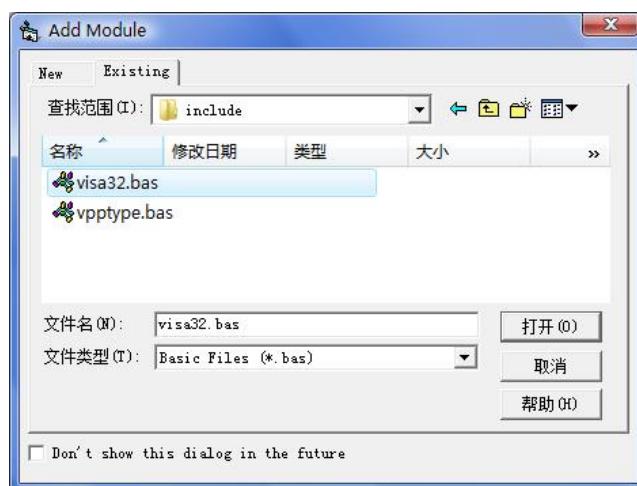


图 4-2-1

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。

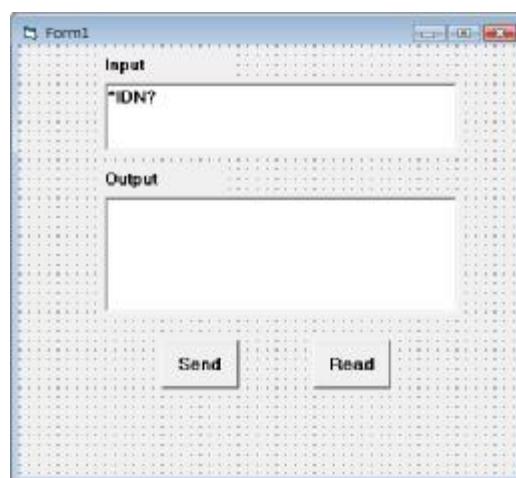


图 4-2-2

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 按键进入编程环境，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。

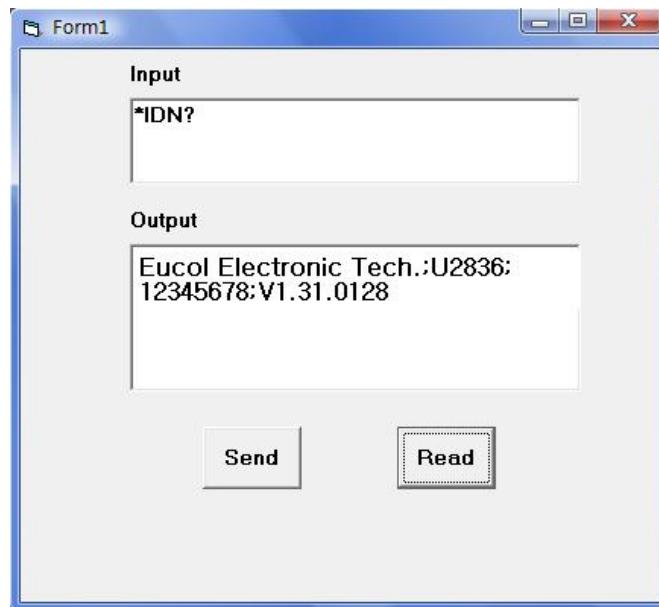


图 4-2-3

### 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。

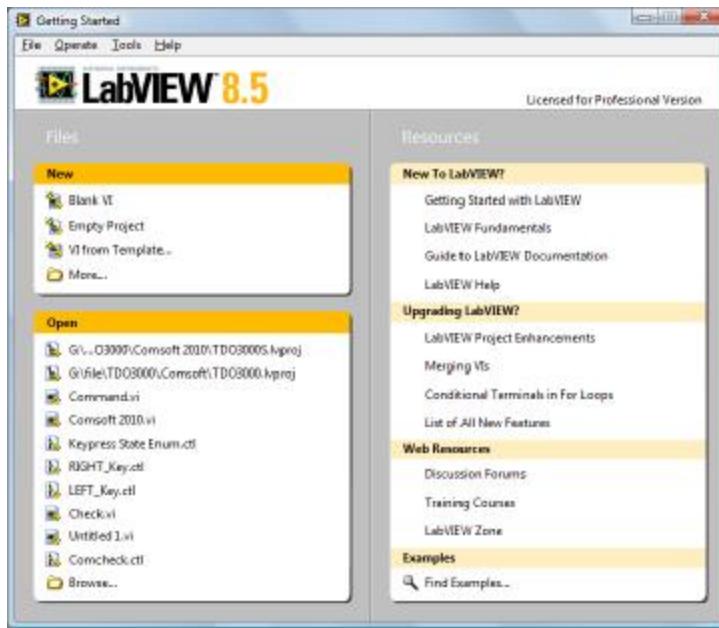


图 4-3-1

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

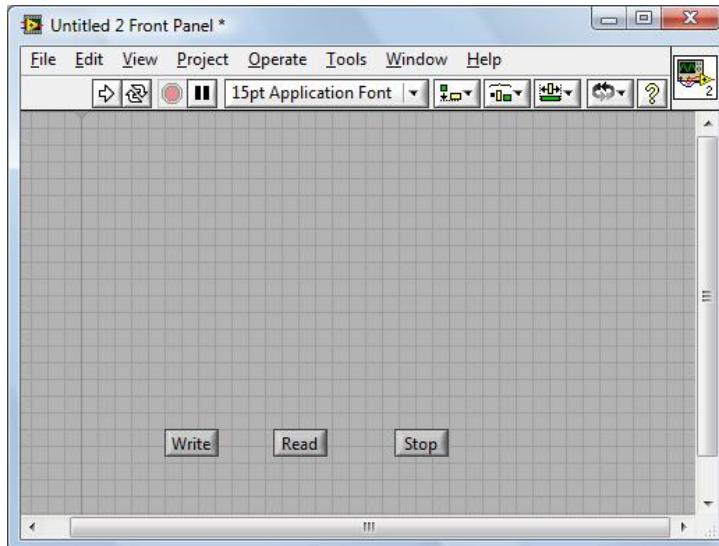


图 4-3-2

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。

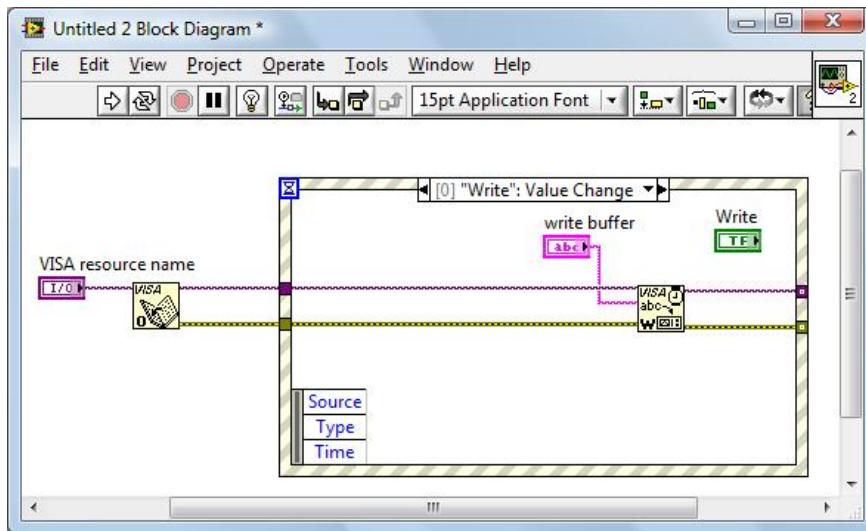


图 4-3-3

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。

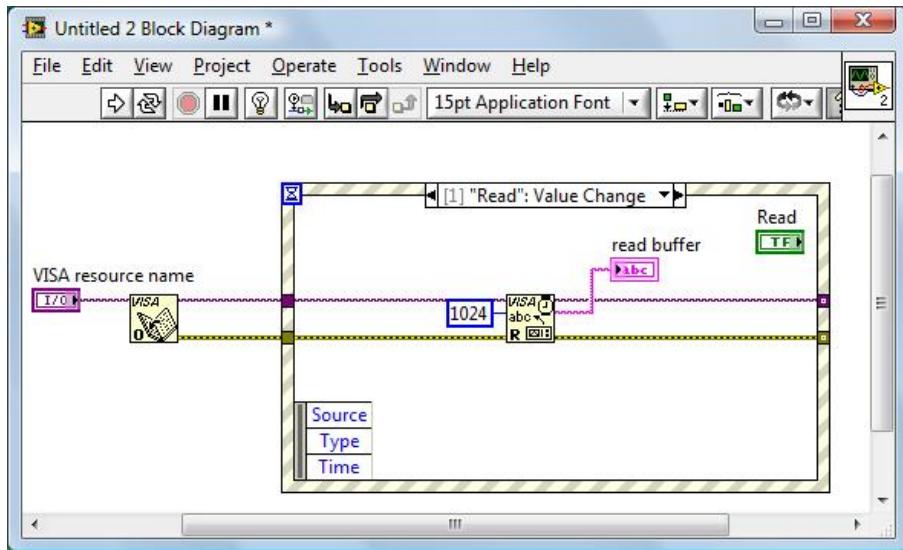


图 4-3-4

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。

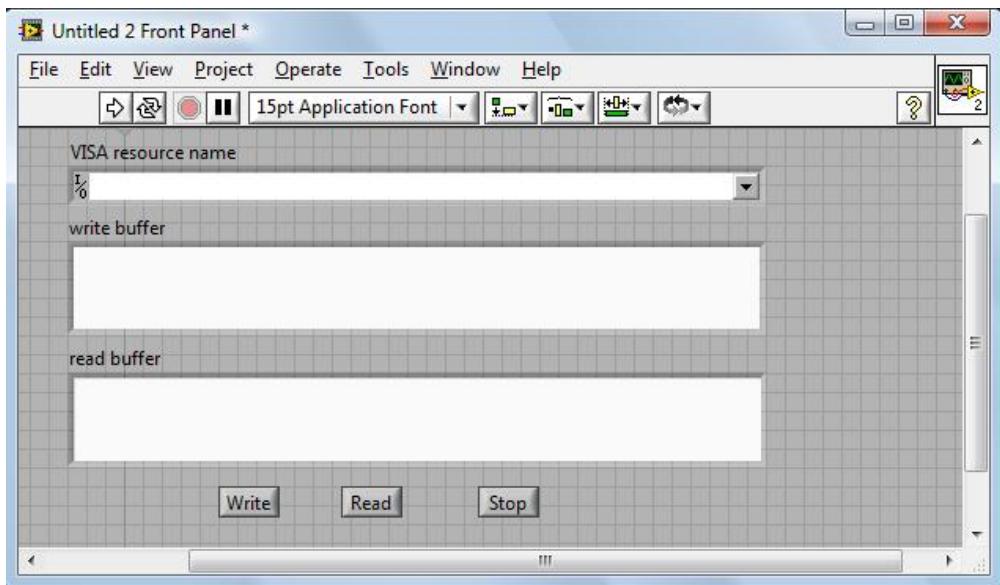


图 4-3-5

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。

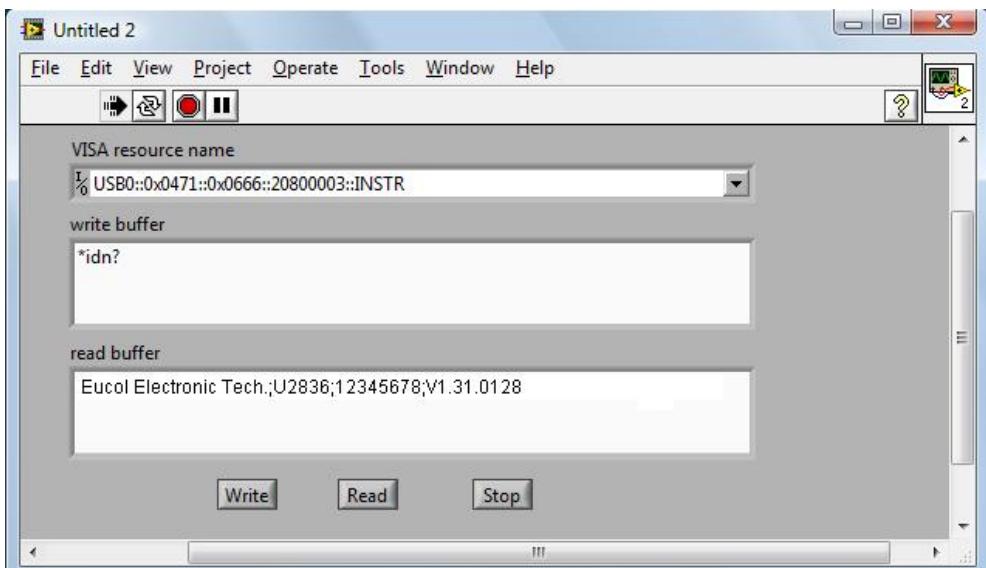


图 4-3-6

23. 按 Stop 键退出程序执行。