



RIGOL

DP2000系列

可编程线性直流电源

编程手册

2023.01

保证和声明

版权

© 2023 普源精电科技股份有限公司

商标信息

RIGOL®是普源精电科技股份有限公司的英文名称和商标。

声明

- 本公司产品受中国及其他国家和地区的专利（包括已取得的和正在申请的专利）保护。
- 本公司保留改变规格及价格的权利。
- 本手册提供的信息取代以往出版的所有资料。
- 本手册提供的信息如有变更，恕不另行通知。
- 对于本手册可能包含的错误，或因手册所提供的信息及演绎的功能以及因使用本手册而导致的任何偶然或继发的损失，RIGOL 概不负责。
- 未经 RIGOL 事先书面许可，不得影印、复制或改编本手册的任何部分。

产品认证

RIGOL 认证本产品符合中国国家产品标准和行业产品标准及 ISO9001:2015 标准和 ISO14001:2015 标准，并进一步认证本产品符合其他国际标准组织成员的相关标准。

联系我们

如您在使用此产品或本手册的过程中有任何问题或需求，可与 RIGOL 联系：

电子邮箱：service@rigol.com

网址：<http://www.rigol.com>

章	主题	页码
	表格目录.....	VI
1	文档概述.....	1
2	SCPI 简介.....	2
3	SCPI 状态寄存器.....	5
3.1	可疑状态寄存器.....	7
3.2	标准事件寄存器.....	9
3.3	状态字节寄存器.....	9
4	命令系统.....	11
4.1	:ANALyzer 命令子系统.....	11
4.1.1	:ANALyzer:COMMMon:MEASure:TYPE.....	11
4.1.2	:ANALyzer:CURRent:MEASure:TYPE.....	12
4.1.3	:ANALyzer:CURRent:THRE.....	12
4.1.4	:ANALyzer:SAVE:ROUTE.....	13
4.1.5	:ANALyzer:SAVE:STATe.....	14
4.1.6	:ANALyzer:STATe.....	15
4.1.7	:ANALyzer:TYPE.....	15
4.2	:APPLy 命令子系统.....	16
4.2.1	:APPLy.....	16
4.3	IEEE488.2 通用命令.....	17
4.3.1	*CLS.....	17
4.3.2	*ESR?.....	18
4.3.3	*ESE.....	19
4.3.4	*IDN?.....	19
4.3.5	*OPC.....	20
4.3.6	*OPT?.....	21
4.3.7	*PSC.....	21
4.3.8	*RCL.....	22
4.3.9	*RST.....	23
4.3.10	*SAV.....	23
4.3.11	*SRE.....	24
4.3.12	*STB?.....	25
4.3.13	*TRG.....	25
4.3.14	*TST?.....	26

4.3.15	*WAI.....	26
4.4	:INSTRument 命令子系统.....	27
4.4.1	:INSTRument:NSElect.....	27
4.4.2	:INSTRument[:SElect].....	28
4.4.3	:INSTRument[:SELEct].....	28
4.5	:LIC 命令子系统.....	29
4.5.1	:LIC:SET.....	29
4.5.2	:LIC:INSTall.....	30
4.6	:MEASure 命令子系统.....	31
4.6.1	:MEASure[:SCALar]:ALL[:DC]?.....	31
4.6.2	:MEASure[:SCALar]:CURRent[:DC]?.....	31
4.6.3	:MEASure[:SCALar]:CURRent:DATA?.....	32
4.6.4	:MEASure[:SCALar]:POWER[:DC]?.....	33
4.6.5	:MEASure[:SCALar][:VOLTage][:DC]?.....	33
4.6.6	:MEASure[:SCALar][:VOLTage]:DATA?.....	34
4.7	:MEMory 命令子系统.....	35
4.7.1	:MEMory:CATalog?.....	35
4.7.2	:MEMory:CDIRectory.....	36
4.7.3	:MEMory:DElete.....	36
4.7.4	:MEMory:DISK?.....	37
4.7.5	:MEMory:LOAD.....	37
4.7.6	:MEMory:LOCK.....	38
4.7.7	:MEMory:MDIRectory.....	39
4.7.8	:MEMory:STORE.....	39
4.7.9	:MEMory:VALid?.....	40
4.8	:OUTPut 命令子系统.....	40
4.8.1	:OUTPut:CVCC?.....	41
4.8.2	:OUTPut:MODE?.....	42
4.8.3	:OUTPut:OCP:ALAR?.....	42
4.8.4	:OUTPut:OCP:QUES?.....	43
4.8.5	:OUTPut:OCP:CLEar.....	43
4.8.6	:OUTPut:OCP:DELay.....	44
4.8.7	:OUTPut:OCP[:STATe].....	45
4.8.8	:OUTPut:OCP:VALue.....	45
4.8.9	:OUTPut:OVP:ALAR?.....	46
4.8.10	:OUTPut:OVP:QUES?.....	47
4.8.11	:OUTPut:OVP:CLEar.....	47

4.8.12	:OUTPut:OVP[:STATe].....	48
4.8.13	:OUTPut:OVP:VALue.....	49
4.8.14	:OUTPut:PAIR.....	50
4.8.15	:OUTPut[:STATe].....	50
4.8.16	:OUTPut:TRACk[:STATe].....	51
4.9	:SOURce 命令子系统.....	52
4.9.1	[:SOURce[<n>]]:CURRent[:LEVel][:IMMediate][:AMPLitude].....	52
4.9.2	[:SOURce[<n>]]:CURRent[:LEVel] [:IMMediate]:STEP[:INCRement]	53
4.9.3	[:SOURce[<n>]]:CURRent:PROTection:CLEar.....	54
4.9.4	[:SOURce[<n>]]:CURRent:PROTection[:LEVel].....	54
4.9.5	[:SOURce[<n>]]:CURRent:PROTection:STATe.....	55
4.9.6	[:SOURce[<n>]]:CURRent:PROTection:TRIPped?.....	56
4.9.7	[:SOURce[<n>]]:VOLTagE[:LEVel][:IMMediate][:AMPLitude].....	57
4.9.8	[:SOURce[<n>]]:VOLTagE[:LEVel] [:IMMediate]:STEP[:INCRement]	58
4.9.9	[:SOURce[<n>]]:VOLTagE:PROTection:CLEar.....	59
4.9.10	[:SOURce[<n>]]:VOLTagE:PROTection[:LEVel].....	59
4.9.11	[:SOURce[<n>]]:VOLTagE:PROTection:STATe.....	60
4.9.12	[:SOURce[<n>]]:VOLTagE:PROTection:TRIPped?.....	61
4.10	:STATus 命令子系统.....	61
4.10.1	:STATus:OPERation:CONDition?.....	62
4.10.2	:STATus:OPERation:ENABle.....	62
4.10.3	:STATus:OPERation[:EVENT]?.....	63
4.10.4	:STATus:PRESet.....	63
4.10.5	:STATus:QUEStionable:ENABle.....	64
4.10.6	:STATus:QUEStionable[:EVENT]?.....	64
4.10.7	:STATus:QUEStionable:INSTRument:ENABle.....	65
4.10.8	:STATus:QUEStionable:INSTRument[:EVENT]?.....	66
4.10.9	:STATus:QUEStionable:INSTRument:ISUMmary[<n>]:CONDition?.....	66
4.10.10	:STATus:QUEStionable:INSTRument:ISUMmary[<n>]:ENABle.....	67
4.10.11	:STATus:QUEStionable:INSTRument:ISUMmary[<n>][:EVENT]?.....	68
4.11	:SYSTem 命令子系统.....	69
4.11.1	:SYSTem:BEEPer:IMMediate.....	69
4.11.2	:SYSTem:BEEPer[:STATe].....	69
4.11.3	:SYSTem:BRIGhtness.....	70
4.11.4	:SYSTem:COMMunicate:LAN.....	70

4.11.4.1	:SYSTem:COMMunicate:LAN:APPLY.....	71
4.11.4.2	:SYSTem:COMMunicate:LAN:AUTOip[:STATe].....	71
4.11.4.3	:SYSTem:COMMunicate:LAN:DHCP[:STATe].....	72
4.11.4.4	:SYSTem:COMMunicate:LAN:DNS.....	73
4.11.4.5	:SYSTem:COMMunicate:LAN:IPADdress.....	73
4.11.4.6	:SYSTem:COMMunicate:LAN:GATEway.....	74
4.11.4.7	:SYSTem:COMMunicate:LAN:MAC?.....	75
4.11.4.8	:SYSTem:COMMunicate:LAN:MANualip[:STATe].....	76
4.11.4.9	:SYSTem:COMMunicate:LAN:SMASk.....	76
4.11.5	:SYSTem:COMMunicate:RLSTate.....	77
4.11.6	:SYSTem:COMMunicate:RS232.....	78
4.11.6.1	:SYSTem:COMMunicate:RS232:BAUD.....	78
4.11.6.2	:SYSTem:COMMunicate:RS232:DBIT.....	78
4.11.6.3	:SYSTem:COMMunicate:RS232:PBIT.....	79
4.11.6.4	:SYSTem:COMMunicate:RS232:SBIT.....	80
4.11.7	:SYSTem:ERRor[:NEXT]?.....	80
4.11.8	:SYSTem:KLOCK:STATe.....	81
4.11.9	:SYSTem:LANGuage:TYPE.....	81
4.11.10	:SYSTem:LOCAl.....	82
4.11.11	:SYSTem:POWEron.....	83
4.11.12	:SYSTem:PRINt?.....	83
4.11.13	:SYSTem:REMote.....	84
4.11.14	:SYSTem:RWLock.....	84
4.11.15	:SYSTem:SAMPLing.....	85
4.11.16	:SYSTem:SAVer.....	86
4.11.17	:SYSTem:SENSe.....	87
4.11.18	:SYSTem:SYNC[:STATe].....	87
4.11.19	:SYSTem:TMODE.....	88
4.11.20	:SYSTem:TLOCK.....	89
4.11.21	:SYSTem:VERSion?.....	89
4.12	:TIMEr 命令子系统.....	90
4.12.1	:TIMEr:CYCLEs.....	90
4.12.2	:TIMEr:CHANnel.....	91
4.12.3	:TIMEr:ENDState.....	91
4.12.4	:TIMEr:GROUPs:NUM?.....	92
4.12.5	TIMEr:GROUP:INDEx.....	93
4.12.6	:TIMEr:GROUP:PARAMeter.....	93

4.12.7	:TIMER:GROUP:DELeT...	94
4.12.8	:TIMER:RUN.....	95
4.12.9	:TIMER[:STATe].....	96
4.12.10	:TIMER:TEMPLet:CONSTRUct.....	96
4.12.11	:TIMER:TEMPLet:FALLRate.....	97
4.12.12	:TIMER:TEMPLet:INTERval.....	98
4.12.13	:TIMER:TEMPLet:INVERt.....	98
4.12.14	:TIMER:TEMPLet:MAXValue.....	99
4.12.15	:TIMER:TEMPLet:MINValue.....	99
4.12.16	:TIMER:TEMPLet:OBJect.....	100
4.12.17	:TIMER:TEMPLet:PERIod.....	101
4.12.18	:TIMER:TEMPLet:POINTs.....	102
4.12.19	:TIMER:TEMPLet:RISERate.....	102
4.12.20	:TIMER:TEMPLet:SELeCt.....	103
4.12.21	:TIMER:TEMPLet:SYMMetry.....	104
4.12.22	:TIMER:TEMPLet:WIDTh.....	104
4.12.23	:TIMER:TEMPLet:STAIr.....	105
4.12.24	:TIMER:TRIG.....	105
4.13	:TRIGger 命令子系统.....	106
4.13.1	:TRIGger:IN[:ENABLe].....	106
4.13.2	:TRIGger:IN:IMMEdiate.....	107
4.13.3	:TRIGger:IN:RESPonse.....	107
4.13.4	:TRIGger:IN:SOURce.....	108
4.13.5	:TRIGger:IN:TYPE.....	109
4.13.6	:TRIGger:OUT:POLARity.....	110
4.13.7	:TRIGger:OUT:SOURce.....	110
4.13.8	:TRIGger:OUT[:ENABLe].....	111
5	编程实例.....	112
5.1	编程准备.....	112
5.2	LabVIEW 编程实例.....	112
5.3	Visual Basic 编程实例.....	117
5.4	Visual C++ 编程实例.....	118

表格目录

表 3.1 可疑状态寄存器中位的定义及其二进制加权值对应的十进制	7
表 3.2 通道可疑状态寄存器中位的定义及其二进制加权值对应的十进制	8
表 3.3 通道可疑状态 SUMMARY 寄存器中位的定义及其二进制加权值对应的十进制	8
表 3.4 标准事件寄存器中位的定义及其二进制加权值对应的十进制	9
表 3.5 状态字节寄存器中位的定义及其二进制加权值对应的十进制	10
表 4.9 DP2031 各通道电压/电流可设范围和默认值	16
表 4.35 过压/过流保护可设范围和默认值	41

1 文档概述

手册指导用户如何使用 SCPI 命令通过远程接口编程控制 DP2000 系列直流电源。DP2000 系列可通过 USB、LAN 和 RS232 接口与计算机进行通信。

提示

本手册的最新版本可登录 RIGOL 网址 (<http://www.rigol.com>) 进行下载。

文档编号

PGH09002-1110

软件版本

00.00.01

软件升级可能更改或增加产品功能，请关注 RIGOL 网站获取最新版本手册或联系 RIGOL 升级软件。

文档格式的约定

1. 按键

用图标表示前面板按键，如  表示“Utility”按键。

2. 菜单

用“菜单文字（加粗）+字符底纹”表示一个菜单选项，如 **系统设置** 表示操作界面上的“系统设置”选项，点击 **系统设置** 进入“系统设置”的功能配置菜单。

3. 操作步骤

用箭头“>”表示下一步操作，如  > **存储** 表示点击  后，再点击 **存储** 功能键。

文档内容的约定

DP2000 系列电源包含以下型号。如无特殊说明，本手册以 DP2031 为例说明 DP2000 系列及其基本操作。

型号	通道数	输出通道电压/电流
DP2031	3	档位 1: 32 V/3 A、32 V/3 A、6 V/5 A 档位 2 ^[1] : 32 V/2 A、32 V/2 A、6 V/10 A (选件)

注[1]: DP2000 系列电源的 CH3 分两个档位，一个是 6 V*5 A，另外一个为 6 V*10 A (选件)，如果档位切换到 6 V*10 A 档位，CH1 和 CH2 立刻会降档为 32 V*2 A。

2 SCPI 简介

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

命令格式

命令通常以冒号 ":" 开始。关键字之间用冒号 ":" 分隔, 关键字后面跟随可选的参数设置。命令行后面添加问号 "?", 表示对此功能进行查询。命令关键字和第一个参数之间以空格分开。

例如:

```
:ANALyzer:TYPE <type>
```

```
:ANALyzer:TYPE?
```

:ANALyzer 是命令的根关键字, TYPE 是第二级关键字。命令行以冒号 ":" 开始, 同时用冒号 ":" 将各级关键字分开, <type> 表示可设置的参数。问号 "?" 表示查询。命令关键字:ANALyzer:TYPE 和参数 <type> 之间用空格分开。

在一些带参数的命令中, 通常用逗号 "," 分隔多个参数, 例如:

```
:TRIGger:IN:TYPE <d>,<type>
```

符号说明

以下符号不随命令发送。

1. 大括号 {}

在命令的定义中, 大括号内包含一个或多个参数, 这些参数可以省略或多次引用。在命令参数变量的取值范围说明中, 参数变量的取值之间通常用竖线 "|" 分隔, 代表必须选择其中一个取值。

2. 竖线 |

竖线用于分隔多个参数选项, 使用命令时必须选择其中一个参数。

3. 方括号 []

方括号中的内容是可省略的。

4. 三角括号 <>

三角括号中的参数必须用一个有效值来替换。

参数类型

1. 布尔型 (Bool)

参数取值为 ON、OFF、1 或 0。例如:

```
:SYSTem:BEEPer <bool>
```

```
:SYSTem:BEEPer?
```

其中: <bool>可设置为: {{1|ON}}{0|OFF}}。查询返回 1 或 0。

2. 离散型 (Discrete)

参数取值为所列举的选项。例如:

```
:ANALyzer:TYPE <type>
```

```
:ANALyzer:TYPE?
```

其中:

- <type>可设置为: COM|CURR。
- 查询返回: COM 或 CURR。

3. 整型 (Integer)

除非另有说明, 参数在有效值范围内可以是任意整数 (NR1 格式)。



注意

此时请不要设置参数为小数格式, 否则将出现异常。

例如:

```
TIMEr:GROUP:INDEx <val>
```

```
TIMEr:GROUP:INDEx
```

其中: <val>可设置为: 1 至 512 之间的整数。查询返回 1 至 512 之间的整数。

4. 实型 (Real)

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

```
:TIMEr:TEMPlet:INTERval <time>
```

```
:TIMEr:TEMPlet:INTERval?
```

其中: <time>可设置为: 0.01 (即 0.01 s) 至 3600 (即 3600 s) 之间的实数。查询以小数格式返回一个实数。

5. ASCII 字符串 (ASCII String)

参数取值为 ASCII 字符的组合。例如:

```
:MEMory:DELeTe <filename>
```

其中: <filename>可设置为: NEW.RSF

命令缩写

所有命令对大小写不敏感, 你可以全部采用大写或小写。但是如果缩写, 必须输入命令格式中的所有大写字母。例如:

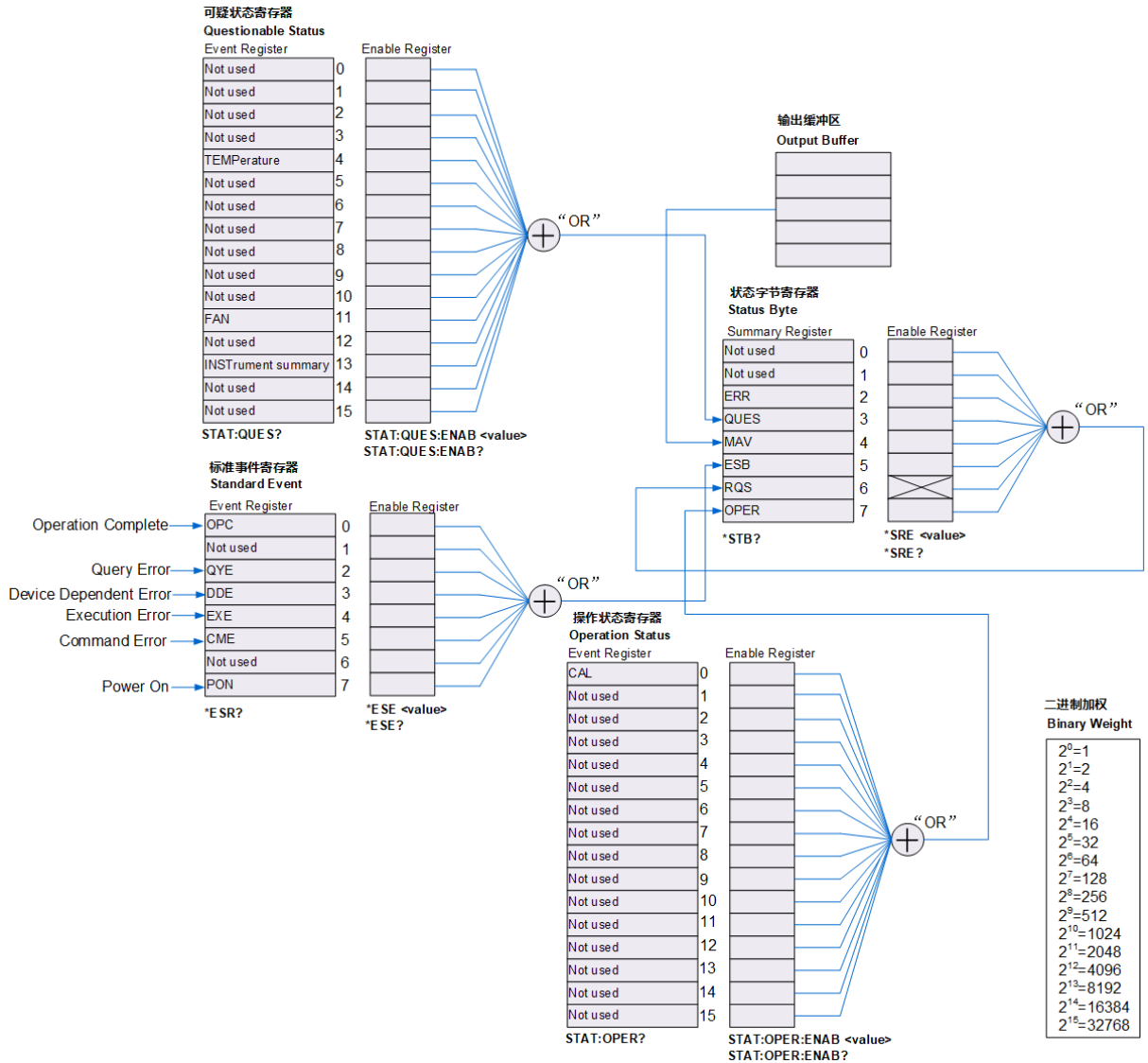
:SYSTem:BRIGhtness?

可缩写成:

:SYST:BRIG?

3 SCPI 状态寄存器

所有的 SCPI 仪器以同样的方式执行状态寄存器操作。状态系统将各种仪器状态记录到如下三个寄存器组中：状态字节寄存器，标准事件寄存器，和可疑状态寄存器组。状态字节寄存器用来记录高级汇总信息，这些信息由其它寄存器组报告。SCPI 状态系统如下图所示。



事件寄存器

事件寄存器是只读寄存器，用来报告电源内部已定义的一些状态。事件寄存器中的位均被锁存，只要一个事件位被置位，随后的状态（该位所表示的事件的状态）改变均被忽略。查询事件寄存器（如 **ESR?* 或 *:STATus:QUESTionable[:EVENT]?* 命令）或者发送清除状态命令（**CLS*）将自动清除事件寄存器中的位，但复位命令 **RST* 不清除事件寄存器中的位。查询事件寄存器将返回一个十进制值，该值与此寄存器中所有位的二进制加权和相对应。

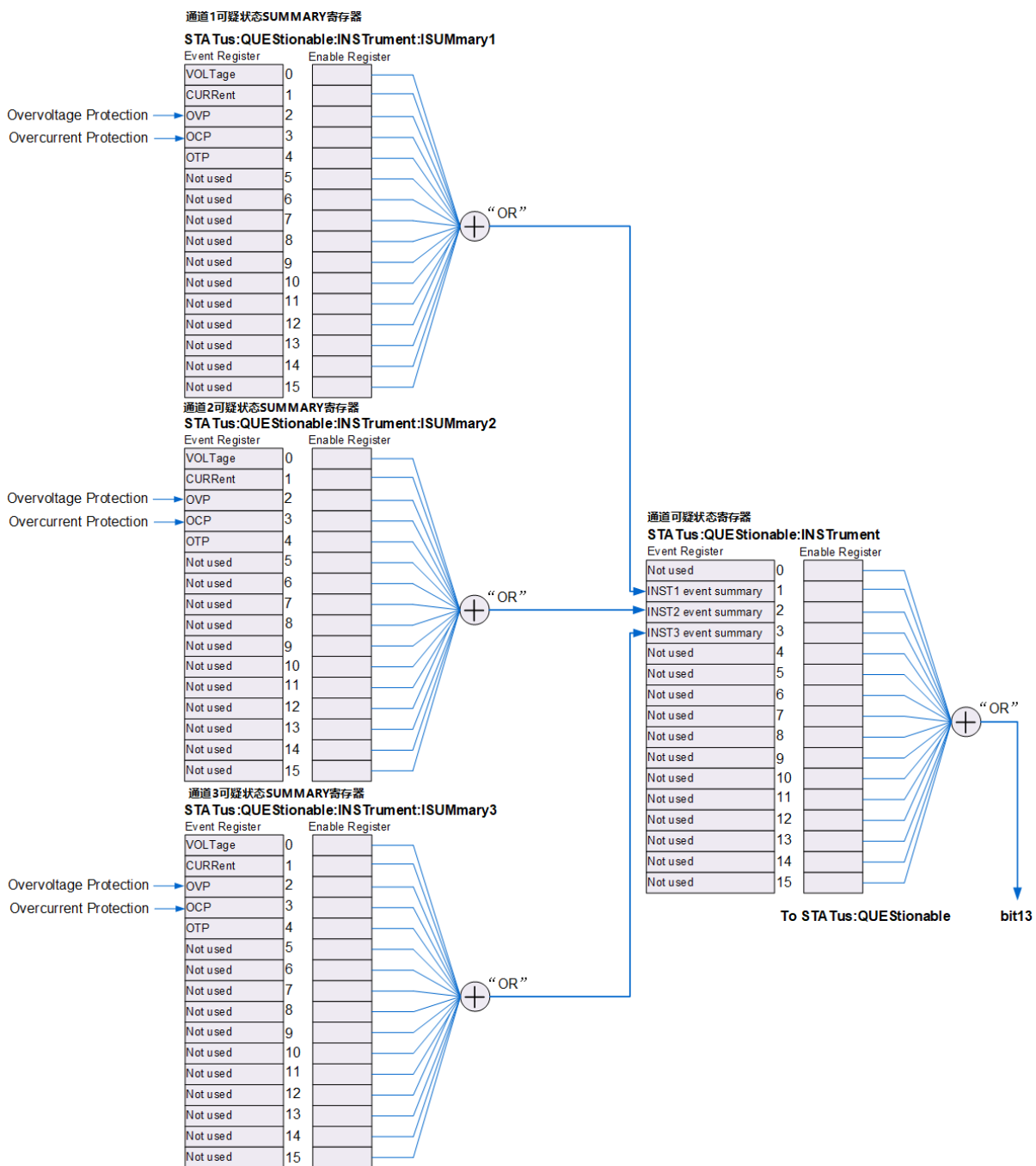
使能寄存器

使能寄存器是可读可写寄存器，用来指定向下一级报告哪些状态信息。查询使能寄存器和发送清除状态命令 **CLS* 均不能清除使能寄存器中的位（但清除状态命令 **CLS* 可以清除事件寄存器

中的位)。要能使使能寄存器中的位，您必须写入一个十进制值，该值与此寄存器中要使能的位的二进制加权和相对应。

多逻辑输出

此部分仅适用于多通道型号。为例，电源的三逻辑输出包括一个通道可疑状态寄存器和三个独立的通道可疑状态 SUMMARY 寄存器（分别对应三个通道的逻辑输出）。通道可疑状态 SUMMARY 寄存器向通道可疑状态寄存器报告各通道的状态，通道可疑状态寄存器继而向可疑状态寄存器的 bit13 (ISUM 位) 报告通道状态。



3.1 可疑状态寄存器

通道可疑状态寄存器指示哪个通道发生了可疑事件，而对于每一个特定的输出来说，通道可疑状态 SUMMARY 寄存器是一个伪可疑状态寄存器。

可疑状态寄存器提供关于电源可疑状态的信息。bit4 (TEMPerature) 报告过温状态，bit11 (FAN) 报告风扇故障状态，bit13 (INSTrument summary) 汇总三个输出通道中任意通道的可疑输出状态。发送 `:STATus:QUEStionable[:EVENT]?` 命令可读取该寄存器。要使用 bit13，您必须先使能您想要用 bit13 来汇总其信息的寄存器。发送 `:STATus:QUEStionable:INSTrument:ENABle` 命令使能通道可疑状态寄存器。然后发送 `:STATus:QUEStionable:INSTrument:ISUMmary[<n>]:ENABle` 命令使能相应的通道可疑状态 SUMMARY 寄存器。多通道型号可疑状态寄存器中位的定义及其二进制加权值对应的十进制如表下所示。

表 3.1 可疑状态寄存器中位的定义及其二进制加权值对应的十进制

位		十进制值	定义
0-3	Not used	0	始终为 0
4	TEMPerature	16	过温
5-10	Not used	0	始终为 0
11	FAN	2048	风扇故障
12	Not used	0	始终为 0
13	INSTrument summary	8192	通道可疑状态寄存器和通道可疑状态 SUMMARY 寄存器组的汇总信息
14-15	Not used	0	始终为 0

通道可疑状态寄存器

通道可疑状态寄存器提供关于三个通道中任何一个通道的可疑状态的信息。bit1 (INST1 event summary)、bit2 (INST2 event summary) 和 bit3 (INST3 event summary) 分别报告 CH1、CH2 和 CH3 的可疑状态的信息。发送 `:STATus:QUEStionable:INSTrument[:EVENT]?` 命令可读取该寄存器。要使用通道可疑状态寄存器，必须使能通道可疑状态 SUMMARY 寄存器。

您可以发送 `:STATus:QUEStionable:INSTrument:ISUMmary[<n>]:ENABle` 命令使能相应的通道可疑状态 SUMMARY 寄存器。通道可疑状态寄存器中位的定义及其二进制加权值对应的十进制如下表所示。

表 3.2 通道可疑状态寄存器中位的定义及其二进制加权值对应的十进制

位		十进制值	定义
0	Not used	0	始终为 0
1	INST1 event summary	2	通道 1 事件汇总信息
2	INST2 event summary	4	通道 2 事件汇总信息
3	INST3 event summary	8	通道 3 事件汇总信息
4-15	Not used	0	始终为 0

通道可疑状态 SUMMARY 寄存器

3 个通道可疑状态 SUMMARY 寄存器分别对应 3 个通道。通道可疑状态 SUMMARY 寄存器提供关于通道电压、电流控制，过压、过流情况的信息。当电压变得不可控时，bit0 (VOLTage) 被置位；当电流变得不可控时，bit1 (CURRent) 被置位。发送 `:STATus:QUEStionable:INSTrument:ISUMmary[<n>]::EVENTt?` 命令可读取相应通道的通道可疑状态 SUMMARY 寄存器。通道可疑状态 SUMMARY 寄存器中位的定义及其二进制加权值对应的十进制如下表所示。

表 3.3 通道可疑状态 SUMMARY 寄存器中位的定义及其二进制加权值对应的十进制

位		十进制值	定义
0	VOLTage	1	电源工作于恒流模式，电压不可控
1	CURRent	2	电源工作于恒压模式，电流不可控
2	OVP	4	发生过压保护
3	OCP	8	发生过流保护
4	OTP	16	发生过温保护
4-15	Not used	0	始终为 0

发送 `:STATus:QUEStionable:INSTrument:ISUMmary[<n>]::CONDition?` 命令可以查询相应通道当前的工作模式（恒压或恒流模式），bit0 为真表示相应通道工作于恒流模式，bit1 为真表示相应通道工作于恒压模式，bit0 和 bit1 均为真表示相应通道的输出电流和电压均不受控制，bit0 和 bit1 均为假表示电源输出关闭。

3.2 标准事件寄存器

标准事件寄存器报告如下类型的仪器事件：上电检测、命令语法错误、命令执行错误、自检或校准错误、查询错误或者操作完成。所有这些事件或者其中任何一个都可以通过使能寄存器报告给状态字节寄存器的 bit5 (ESB, Event Summary Bit)。要设置使能寄存器掩码，您需要使用 **ESE* 命令向该寄存器写入一个十进制值。标准事件寄存器中位的定义及其二进制加权值对应的十进制如下表所示。

一个错误状态（标准事件寄存器的 bit2、3、4 或 5）记录电源错误队列中的一个或多个错误，您可以发送 *:SYSTEM:ERROR[:NEXT]?* 命令读取错误队列。

表 3.4 标准事件寄存器中位的定义及其二进制加权值对应的十进制

位		十进制值	定义
0	OPC	1	操作完成。之前的所有命令，包括 <i>*OPC</i> 命令，都已被执行。
1	Not used	0	始终为 0。
2	QYE	4	查询错误。电源试图读取输出缓冲器但它却是空的；或者前一条查询命令还未被读取时，又接收到一条新的命令；或者输入和输出缓冲器都已满。
3	DDE	8	设备错误。自检或校准发生错误。
4	EXE	16	执行错误。包括触发忽略，初始化忽略，设置冲突，数据超限，数据过长，无效参数值。
5	CME	32	命令错误。命令语法发生错误。
6	Not used	0	始终为 0。
7	PON	128	上电检测。从上一次事件寄存器被读取或清除后，先关闭电源，然后再打开电源。

3.3 状态字节寄存器

状态字节寄存器报告其它状态寄存器的状态信息。查询电源输出缓冲区中正等待查询的数据将立刻通过状态字节寄存器的 bit4 (MAV, Message Available Bit) 报告。状态字节寄存器的 SUMMARY 寄存器中的位不被锁存。清除事件寄存器将会清除状态字节寄存器的 SUMMARY 寄存器中的相应位。读取输出缓冲器中的所有信息，包括任何未完成的查询，将会清除 bit4 (MAV, Message Available Bit)。状态字节寄存器中位的定义及其二进制加权值对应的十进制如下表所示。

表 3.5 状态字节寄存器中位的定义及其二进制加权值对应的十进制

位		十进制值	定义
0-1	Not used	0	始终为 0。
2	ERR	4	错误队列中存储了一个或多个错误。
3	QUES	8	可疑状态寄存器中的一个或多个位被置位（使能寄存器中的位必须被使能）。
4	MAV	16	电源输出缓冲区中的数据可用。
5	ESB	32	标准事件寄存器中的一个或多个位被置位（使能寄存器中的位必须被使能）。
6	RQS	64	电源正请求服务。
7	OPER	128	操作事件寄存器中的一位或者多位被置位（使能寄存器的位必须被使能）。

4 命令系统

本章以字母 A-Z 的顺序逐条介绍本仪器支持各命令的格式、功能、参数以及使用说明等。



说明

- 如无特殊说明，本手册中的说明均以 DP2031 型号为例。
- 对于参数设置命令（如时间、电压、电流等），仪器均无法识别随参数发送的单位，仅能识别数字，并以默认单位设置该参数。如无特殊说明，有关各参数类型的默认单位如下表所示。

参数类型	默认单位
时间	秒 (s)
电压	伏特 (V)
电流	安培 (A)
功率	瓦特 (W)

4.1 :ANALyzer 命令子系统

:ANALyzer 命令用于设置分析器参数、执行分析并查询分析结果。

4.1.1 :ANALyzer:COMMOn:MEASure:TYPE

命令格式

```
:ANALyzer:COMMOn:MEASure:TYPE <ch>[,<ch>[,<ch>]]
```

```
:ANALyzer:COMMOn:MEASure:TYPE?
```

功能描述

设置或查询通用分析的分析对象。

参数

名称	类型	范围	默认值
<ch>	离散型	{CH1_V CH1_C CH1_P CH2_V CH2_C CH2_P CH3_V CH3_C CH3_P}	-

说明

您可以选择通道 1 至通道 3 中“电流”、“电压”或“功率” 9 项中的任意 1 至 3 项进行分析。

返回格式

查询返回 CH1_V、CH1_C、CH1_P、CH2_V、CH2_C、CH2_P、CH3_V、CH3_C 或 CH3_P 中的 1 至 3 项。

举例

```
:ANALyzer:COMMon:MEASure:TYPE CH1_V,CH2_P /*设置通用分析的分析对象为通道
1 的电压和通道 2 的功率*/
:ANALyzer:COMMon:MEASure:TYPE? /*查询通用分析的分析对象, 返回
CH1_V,CH2_P*/
```

4.1.2 :ANALyzer:CURRent:MEASure:TYPE

命令格式

```
:ANALyzer:CURRent:MEASure:TYPE <ch>[,<ch>]
```

```
:ANALyzer:CURRent:MEASure:TYPE?
```

功能描述

设置或查询脉冲电流的分析对象。

参数

名称	类型	范围	默认值
<ch>	离散型	{CH1 CH2}	-

说明

无。

返回格式

查询返回 CH1、CH2 或 CH1,CH2。

举例

```
:ANALyzer:CURRent:MEASure:TYPE CH1,CH2 /*设置脉冲电流的分析对象为通道 1 和
通道 2*/
:ANALyzer:CURRent:MEASure:TYPE? /*查询脉冲电流的分析对象, 返回 CH1,CH2*/
```

4.1.3 :ANALyzer:CURRent:THRE

命令格式

```
:ANALyzer:CURRent:THRE <ch>,<type>,<bool>,<val>
```

```
:ANALyzer:CURRent:THRE? <ch>,<type>
```

功能描述

设置或查询脉冲电流分析的正脉冲阈值和负脉冲阈值。

参数

名称	类型	范围	默认值
<ch>	离散型	{CH1 CH2}	-
<type>	离散型	{UP LOW}	-
<bool>	布尔型	{0 1 ON OFF}	OFF
<val>	实型	0 至 3.15A	正脉冲: 1A 负脉冲: 0.1A

说明

- **UP**: 设置正脉冲阈值<val>, 进行脉冲电流分析时, 分析器将记录高于此阈值的脉冲数量和最近一个高于此阈值脉冲的脉宽。
- **LOW**: 设置负脉冲阈值<val>, 进行脉冲电流分析时, 分析器将记录低于此阈值的脉冲数量和最近一个低于此阈值脉冲的脉宽。

返回格式

查询返回对应通道脉冲电流分析的上下阈值分析的开关状态和具体设置值, 如返回 1,1.0000。

举例

```
:ANALyzer:CURRent:THRE CH1,UP,ON,1 /*设置脉冲电流分析的通道 1 正脉冲阈值为 1A*/
:ANALyzer:CURRent:THRE? CH1,UP /*查询脉冲电流分析的通道 1 正脉冲阈值, 返回 1,1.0000*/
```

4.1.4 :ANALyzer:SAVE:ROUTE

命令格式

```
:ANALyzer:SAVE:ROUTE <dest>
```

```
:ANALyzer:SAVE:ROUTE?
```

功能描述

设置或查询录制的日志文件存储路径。

参数

名称	类型	范围	默认值
<dest>	ASCII 字符串	有效存储路径	-

说明

参数<dest>为内/外部存储器中的指定路径，格式为<route>.ROF，如 C:/RA.ROF。其中<route>表示文件路径，可包括中文、英文字符和数字；.ROF 为录制的日志文件名称后缀，不可省略。

返回格式

查询返回当前录制的日志文件的保存路径，如 C:/RA.ROF。

举例

```
:ANALyzer:SAVE:ROUTe C:/RA.ROF /*设置当前日志文件的保存路径为 C:/RA.ROF*/
:ANALyzer:SAVE:ROUTe? /*查询当前日志文件的保存路径，返回 C:/RA.ROF*/
```

4.1.5 :ANALyzer:SAVE:STATe

命令格式

```
:ANALyzer:SAVE:STATe <bool>
```

```
:ANALyzer:SAVE:STATe?
```

功能描述

设置或查询日志保存状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{1 0 ON OFF}	OFF

说明

- 当日志保存打开时，打开分析器仪器以当前采样率实时存储采集到的波形数据。
- 至少录制 1 个点才可以保存录制文件。
- 关闭分析器时，本次录制结束，仪器自动将录制文件存储至当前设置的保存路径。

返回格式

查询返回 0 或 1。

举例

```
:ANALyzer:SAVE:STATe 1 /*打开日志录制*/
:ANALyzer:SAVE:STATe? /*查询当前录制器的状态，返回 1*/
```

4.1.6 :ANALyzer:STATe

命令格式

```
:ANALyzer:STATe <bool>
```

```
:ANALyzer:STATe?
```

功能描述

设置或查询分析器的运行状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1ON OFF}	OFF

说明

无。

返回格式

查询返回 0 或 1。

举例

```
:ANALyzer:STATe ON /*打开分析器*/
:ANALyzer:STATe? /*查询分析器的状态, 返回 1*/
```

4.1.7 :ANALyzer:TYPE

命令格式

```
:ANALyzer:TYPE <type>
```

```
:ANALyzer:TYPE?
```

功能描述

设置或查询分析器的分析项目。

参数

名称	类型	范围	默认值
<type>	离散型	{COM CURR}	COM

说明

发送该命令可选择分析项目为通用分析（COM）或脉冲电流分析（CURR）。

返回格式

查询返回 COM 或 CURR。

举例

```
:ANALyzer:TYPE CURR /*设置分析器分析项目为脉冲电流*/
:ANALyzer:TYPE? /*查询分析器分析项目，返回 CURR*/
```

4.2 :APPLY 命令子系统

:APPLY 命令是通过远程接口对电源进行编程的最直接的方法。对于多通道型号，在一个命令中可以同时选择指定通道并设置电压和电流值。只要设置值在指定型号的相应通道的参数可设范围之内，执行该命令后，输出电压和电流立即改变为设置值。DP2031 各通道对应的电压/电流可设范围和默认值如下表所示。

表 4.9 DP2031 各通道电压/电流可设范围和默认值

通道 (档位)			电压/电流可设范围	电压/电流默认值
DP2031	Range1	CH1	0~32 V/0~3 A	0 V/0.1 A
		CH2	0~32 V/0~3 A	0 V/0.1 A
		CH3	0~6 V/0~5 A	0 V/0.1 A
	Range2 (选件)	CH1	0~32 V/0~2 A	0 V/0.1 A
		CH2)	0~32 V/0~2 A	0 V/0.1 A
		CH3	0~6 V/0~10 A	0 V/0.1 A

4.2.1 :APPLY

命令格式

```
:APPLy [<source>[,< volt>|< app>[,< curr>|< app>]]]
```

```
:APPLy? [<source>[,< option>]]
```

功能描述

选择指定通道为当前通道并设置该通道的电压和电流值。

查询指定通道当前设置的电压电流值。

参数

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

名称	类型	范围	默认值
<volt>	实型	请参考表 4.9: DP2031 各通道电压/电流可设范围和默认值	
<curr>	实型	请参考表 4.9: DP2031 各通道电压/电流可设范围和默认值	
<option>	离散型	{CURR VOLT}	-
<app>	离散型	{MINimum MAXimum DEF}	-

说明

- 查询命令中，参数<source>用来选择要查询的通道，若省略，则表示查询当前通道。
- 参数<volt>和<curr>分别表示指定通道的电压和电流值，若仅指定一个参数值，默认它为电压设置值；若两个参数均省略，则表示选择通道，与*:INSTrument[:SELEct]*命令功能相同。
- 选择参数 MINimum、MAXimum 或 DEF 时，设置指定通道的电流/电压值为该通道可设范围的最小值、最大值或默认值。不同型号各通道（档位）的电压/电流可设范围和默认值请参考表 4.9: DP2031 各通道电压/电流可设范围和默认值。
- 参数<option>用来选择查询指定通道的电压或电流值，若省略，则同时查询指定通道的电压和电流值。

返回格式

返回一个字符串。

- 若仅指定参数<source>，返回指定通道的通道名称、额定电压/电流值、电压设置值和电流设置值，如 CH1:32V/3A,5.000,1.0000。
- 若省略所有参数，返回当前通道的电压设置值和电流设置值，如 5.000,1.0000。

举例

```
:APPL CH1,5,1 /*设置 CH1 的电压和电流值分别为 5V 和 1A*/
:APPL? CH1 /*查询 CH1 的电压和电流设置值，返回 CH1:32V/3A,5.000,1.0000*/
```

4.3 IEEE488.2 通用命令

4.3.1 *CLS

命令格式

*CLS

功能描述

清除所有事件寄存器。

参数

无。

说明

- 您也可以发送查询事件寄存器的命令 (`:STATus:QUEStionable[:EVENT]?` 或 `*ESR?`) 清除相应事件寄存器。
- 发送复位命令 (`*RST`) 或设备清除命令不能清除事件寄存器。

返回格式

无。

4.3.2 *ESR?**命令格式**

`*ESR?`

功能描述

查询标准事件寄存器的事件寄存器，同时清除此寄存器的位。

参数

无。

说明

- 执行该命令，返回一个十进制值（该值与此寄存器中所有位的二进制加权和相对应），同时清除此寄存器的状态。标准事件寄存器中位的定义及其对应的十进制值请参考 [标准事件寄存器中位的定义及其二进制加权值对应的十进制值](#)。

例如，若当前仪器中发生查询错误和执行错误，那么标准事件寄存器的事件寄存器中的 bit2（查询错误位）和 bit4（执行错误位）被置位，则执行该命令的返回结果为 20（根据： $2^2+2^4=20$ ）。

- 标准事件寄存器的事件寄存器中的位被锁存，读取该寄存器将清除该寄存器。您也可以使用命令 `*CLS` 清除此寄存器。

返回格式

返回一个十进制值，该值与标准事件寄存器的事件寄存器中所有位的二进制加权和相对应，如 20。

举例

```
*ESR? /*查询标准事件寄存器的事件寄存器并清除，返回 20*/
```

4.3.3 *ESE

命令格式

***ESE** <enable_value>

***ESE?**

功能描述

使能或查询标准事件寄存器的使能寄存器中的位。

参数

名称	类型	范围	默认值
<enable_value>	字符型	见下文 说明	-

说明

- 参数 <enable_value> 是一个十进制值，该值与标准事件寄存器的使能寄存器中要使能的位的二进制加权和相对应。标准事件寄存器中位的定义及其对应的十进制值请参考 [标准事件寄存器中位的定义及其二进制加权和对应的十进制值](#)。

例如，要使能标准事件寄存器的使能寄存器中的 bit2（查询错误）和 bit4（执行错误），参数 <enable_value> 需设置为 20（根据： $2^2+2^4=20$ ）。

- 使能标准事件寄存器的使能寄存器中的位后，系统向状态字节寄存器报告相应位的状态。
- 参数 <enable_value> 设为 0 时，执行该命令可以清除标准事件寄存器的使能寄存器。
- 您也可以发送 ***PSC** 命令（*PSC 1）在下次仪器上电时清除标准事件寄存器的使能寄存器。

返回格式

返回一个十进制值，该值与标准事件寄存器的使能寄存器中使能的位的二进制加权和相对应，如 20。

举例

```
*ESE 20 /*使能标准事件寄存器的使能寄存器中的 bit2（查询错误）和 bit4（执行错误）*/
*ESE? /*查询当前标准事件使能寄存器中使能的位，返回 20*/
```

4.3.4 *IDN?

命令格式

***IDN?**

功能描述

查询仪器的 ID 字符串。

参数

无

说明

无。

返回格式

返回仪器的 ID 字符串，格式为 Rigol Technologies,<model>,<serial number>,<software version>。

- <model>: 仪器型号
- <serial number>: 仪器序列号
- <software version>: 仪器软件版本

4.3.5 *OPC

命令格式

*OPC

*OPC?

功能描述

使能标准事件寄存器的事件寄存器中的 bit0 位 (OPC, “完成操作” 位)。

查询之前的所有命令是否已被执行。

参数

无。

说明

- 操作完成是指之前的所有命令，包括*OPC 命令，都已被执行。
- 发送查询命令*OPC?并查看结果可以保证同步。
- 编程设置仪器配置（通过执行命令串）时，将该命令作为最后一条命令可以确定何时命令队列已全部被执行（命令队列全部被执行后，标准事件寄存器的事件寄存器的 bit0 (OPC, “完成操作” 位) 被置位)。
- 若在加载电源输出缓冲区信息（查询数据）的命令之后发送*OPC 命令，则可以通过“OPC” 位来确定何时此信息可用。

返回格式

若之前的所有命令已被执行，返回+1。

举例

```
*OPC /*完成当前操作后置位标准事件寄存器 bit0 (OPC, “完成操作”位) */
*OPC? /*查询当前操作是否完成, 返回+1*/
```

4.3.6 *OPT?**命令格式**

***OPT?**

功能描述

查询选件的安装状态。

参数

无。

说明

- DP2000 系列选件包括第 3 通道 10A 大电流量程和 7.5 kSa/s 快速电流采样选件。
- 如需使用可选功能，请购买相应选件并正确安装 (*:LIC:SET*)。

返回格式

返回选件的安装状态，不同选件之间用逗号 “,” 隔开。若选件已安装，则返回选件名称；若未安装，则无返回值。

类别	返回值
提供 7.5 kSa/s 快速电流采样选件	DP2000-HADC
提供第 3 通道 10 A 大电流量程	DP2000-10A

例如：返回 DP2000-HADC,DP2000-10A 表示已安装对应的 2 个选件。

举例

```
*OPT? /*查询选件的安装状态, 返回 DP2000-HADC,DP2000-10A*/
```

4.3.7 *PSC**命令格式**

***PSC <bool>**

***PSC?**

功能描述

启用或禁用上电时清除状态字节和标准事件寄存器的使能寄存器。

查询上电时清除状态字节和标准事件寄存器的使能寄存器功能的状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1}	0

说明

- *PSC 1 命令表示上电时清除状态字节和标准事件寄存器的使能寄存器；*PSC 0 命令表示上电时状态字节和标准事件寄存器的使能寄存器不受影响。
- 您也可以分别发送 *SRE 命令 (*SRE 0) 或 *ESE 命令 (*ESE 0) 清除状态字节和标准事件寄存器的使能寄存器。

返回格式

查询返回 0 或 1。

举例

```
*PSC 1 /*启用通电时清除状态字节和标准事件寄存器的使能寄存器*/
*PSC? /*查询通电时状态清除设置，返回 1*/
```

4.3.8

*RCL

命令格式

*RCL <n>

功能描述

调用内部存储器中已存储的仪器状态。

参数

名称	类型	范围	默认值
<n>	离散型	{0 1 2 3 4 5 6 7 8 9}	-

说明

- 该命令调用电源内部存储器中的仪器状态，选择编号 0 至 9 可以分别调用命名为 RIGOL0.RSF~RIGOL9.RSF 的仪器状态。
- 仅当内部存储器中已存有指定的状态文件时，该命令有效。

- 您也可以发送 `.MEMory[:STATe]:LOAD` 命令调用内部存储器中已存储的仪器状态。

返回格式

无。

举例

```
*RCL 5 /*调用存储在内部存储器中文件名为 RIGOL5.RSF 的状态文件*/
```

4.3.9 *RST

命令格式

```
*RST
```

功能描述

将电源恢复至出厂默认状态。

参数

无

说明

执行命令立即恢复至出厂默认状态，不询问是否恢复。

返回格式

无

4.3.10 *SAV

命令格式

```
*SAV <n>
```

功能描述

以指定的文件名 (RIGOLn.RSF) 将当前的仪器状态保存到内部存储器中。

参数

名称	类型	范围	默认值
<n>	离散型	{0 1 2 3 4 5 6 7 8 9}	-

说明

- 若存储位置已存有同名状态文件，该命令将直接存储当前仪器状态到内部存储器（直接覆盖原文件）。若指定的存储位置已存储的状态文件被锁定（*:MEMory:LOCK*），该命令无效（不会直接覆盖原文件）。
- 您可以发送 *:MEMory:STORe* 命令将电源当前的状态保存到内部或外部存储器中。

返回格式

无。

举例

```
*SAV 5 /*将当前的仪器状态命名为 RIGOL5.RSF 并存储到电源内部存储器中*/
```

4.3.11

*SRE

命令格式

**SRE <enable_value>*

**SRE?*

功能描述

使能状态字节寄存器的使能寄存器中的位。

查询当前状态字节寄存器的使能寄存器中使能的位。

参数

名称	类型	范围	默认值
<enable_value>	离散型	见下文 <i>说明</i>	-

说明

- 参数 <enable_value> 是一个十进制值，该值与状态字节寄存器的使能寄存器中要使能的位的二进制加权和相对应。状态字节寄存器中位的定义及其对应的十进制值请参考 *状态字节寄存器*。
例如，要使能状态字节寄存器的使能寄存器中的 bit3 (QUES) 和 bit4 (MAV)，参数 <enable_value> 需设置为 24 (根据： $2^3+2^4=24$)。
- 使能后，系统通过状态字节寄存器的 bit6 (请求服务位) 发送服务请求。
- 参数 <enable_value> 设为 0 时，执行该命令可以清除状态字节寄存器的使能寄存器。您也可以发送 **PSC* 命令 (**PSC 1*) 在下次仪器上电时清除状态字节寄存器的使能寄存器。

返回格式

返回一个十进制值，该值与状态字节使能寄存器中使能的位的二进制加权和相对应，如+24。

举例

```
*SRE 24 /*使能状态字节寄存器的使能寄存器中的 bit3 (QUES) 和 bit4 (MAV)，启用服务请求*/  
*SRE? /*查询当前状态字节寄存器的使能寄存器中使能的位，返回+24*/
```

4.3.12 *STB?

命令格式

***STB?**

功能描述

查询当前状态字节寄存器的 SUMMARY 寄存器。

参数

无。

说明

执行该命令，返回一个十进制值（该值与此寄存器中所有位的二进制加权和相对应），但不清除此寄存器。状态字节寄存器中位的定义及其对应的十进制值请参考 [状态字节寄存器中位的定义及其二进制加权值对应的十进制值](#)。例如，若当前仪器中产生可疑状态且发送服务请求中断，那么状态字节寄存器的 SUMMARY 寄存器中的 bit3 (QUES) 和 bit6 (RQS) 被置位，则执行该命令的返回结果为+72（根据： $2^3+2^6=72$ ）。

返回格式

返回一个十进制值，该值与状态字节寄存器的 SUMMARY 寄存器中所有位的二进制加权和相对应，如+72。

举例

```
*STB? /*查询当前状态字节寄存器的 SUMMARY 寄存器，返回+72*/
```

4.3.13 *TRG

命令格式

***TRG**

功能描述

产生一次触发动作。

参数

无。

说明

- 该命令仅适用于已选择“总线（软件）触发”的触发系统。
- 选择“总线（软件）触发”后，发送该命令将触发电源，经过指定的延时时间后产生一次触发动作。

返回格式

无。

举例

```
*TRG /*产生一次触发动作*/
```

4.3.14 *TST?

命令格式

```
*TST?
```

功能描述

查询仪器的自检结果。

参数

无。

说明

电源在开机时执行自检操作，该命令查询自检结果。

返回格式

查询仪器的自检结果，通过返回+0，失败返回+1。

举例

```
无
```

4.3.15 *WAI

命令格式

```
*WAI
```

功能描述

将仪器配置为等待所有未完成操作完成之后，再执行任何其他命令。

参数

无。

说明

当前选择“BUS”（总线触发，即软件触发）时，发送该命令可以保证同步。执行该命令后，仪器将等待所有未完成操作完成之后，再执行任何其他命令。

返回格式

无。

举例

```
*WAI /*将仪器配置为等待所有未完成操作完成之后，再执行任何其他命令*/
```

4.4 :INSTRument 命令子系统

:INSTRument 命令用于选中将要进行操作的通道，或查询当前选中的通道。

4.4.1 :INSTRument:NSElect

命令格式

```
:INSTRument:NSElect <n>
```

```
:INSTRument:NSElect?
```

功能描述

选中将要进行操作的通道或查询当前选中的通道。

参数

名称	类型	范围	默认值
<n>	离散型	{1 2 3}	1

说明

- “1”代表CH1，“2”代表CH2，“3”代表CH3。
- 该命令用数字代替:INSTRument[:SElect]命令中的通道标识符，功能与:INSTRument[:SElect]和:INSTRument[:SElect]两条命令功能相同。

返回格式

返回 1、2 或 3，分别代表 CH1，CH2 和 CH3。

举例

```
:INST:NSEL 2 /*选择 CH2 为当前通道*/
:INST:NSEL? /*查询当前选中的通道，返回 2*/
```

4.4.2 :INSTRument[:SElect]

命令格式

```
:INSTRument[:SElect] <source>
```

```
:INSTRument[:SElect]?
```

功能描述

选中将要进行操作的通道或查询当前选中的通道。

参数

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

说明

该命令与 `:INSTRument:NSElect` 和 `INSTRument[:SElect]` 两条命令功能相同。

返回格式

返回查询通道名称和该通道额定电压/电流，如返回 CH1:32V/3A、CH2:32V/3A。

举例

```
:INST CH2 /*选择 CH2 为当前通道*/
:INST? /*查询当前选中的通道，返回 CH2:32V/3A*/
```

4.4.3 :INSTRument[:SElect]

命令格式

```
:INSTRument[:SElect] <source>
```

```
:INSTRument[:SElect]?
```

功能描述

选中将要进行操作的通道或查询当前选中的通道。

参数

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

说明

该命令与 `:INSTRument:NSElect` 和 `INSTRument[:SElect]` 两条命令功能相同。

返回格式

返回查询通道名称和该通道额定电压/电流，如返回 CH1:32V/3A、CH2:32V/3A。

举例

```
:INST CH2 /*选择 CH2 为当前通道*/
:INST? /*查询当前选中的通道，返回 CH2:32V/3A*/
```

4.5 :LIC 命令子系统

:LIC 命令用于安装选件。

4.5.1 :LIC:SET

命令格式

```
:LIC:SET <license>
```


功能描述

安装选件。

参数

名称	类型	范围	默认值
<license>	ASCII 字符串	见下文 说明	-

说明

- 选件安装需使用选件授权码。<license>即选件授权码（每台仪器对应一个）是一段长度固定的文本。
- 欲获得选件授权码，首先请订购所需选件以获得密匙，然后使用密匙按照如下步骤生成选件授权码。
 - 登录 RIGOL 官网 (www.rigol.com) 后，单击 **服务中心** > **服务** > **产品授权码注册**，进入软件授权码注册界面。
 - 在软件授权码注册界面中输入正确的密匙、仪器序列号（点击  > **帮助** > **关于仪器**，即可获得序列号）和验证码，点击 **生成** 键即可获取选件授权文件下载链接。如需使用选件授权文件，请点击下载文件并将其下载至 U 盘根目录下。
- 您可以用命令 ***OPT?** 查询选件安装情况。

返回格式

无。

举例

无。

4.5.2 :LIC:INSTall**命令格式**

```
:LIC:INSTall <license>
```


功能描述

安装选件。

参数

名称	类型	范围	默认值
<license>	ASCII 字符串	见下文 说明	-

说明

- 选件安装需使用选件授权码。<license>即选件授权码（每台仪器对应一个）是一段长度固定的文本。
- 欲获得选件授权码，首先请订购所需选件以获得密匙，然后使用密匙按照如下步骤生成选件授权码。
 - 登录 RIGOL 官网 (www.rigol.com) 后，单击 **服务中心** > **服务** > **产品授权码注册**，进入软件授权码注册界面。
 - 在软件授权码注册界面中输入正确的密匙、仪器序列号（点击  > **帮助** > **关于仪器**，即可获取序列号）和验证码，点击 **生成** 键即可获取选件授权文件下载链接。如需使用选件授权文件，请点击下载文件并将其下载至 U 盘根目录下。
- 您可以用命令 `*OPT?` 查询选件安装情况。

返回格式

无。

举例

无。

4.6 :MEASure 命令子系统

:MEASure 命令用于查询指定通道输出端子上测得的电压、电流、输出功率值。

4.6.1 :MEASure[:SCALar]:ALL[:DC]?

命令格式

```
:MEASure[:SCALar]:ALL[:DC]? [<source>]
```

功能描述

查询在指定通道输出端子上测得的电压、电流和功率值。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3 SERies PARAllel}	-

说明

- **SERies:** 查询当前串联回路的总电压、电流、功率。
- **PARAllel:** 查询当前并联回路的总电压、电流、功率。

若省略参数<source>, 则查询在当前选中通道输出端子上测得的电压、电流和功率值。

您也可以发送:*:MEASure[:SCALar][:VOLTage][:DC]?*、*:MEASure[:SCALar]:CURRent[:DC]?*或:*:MEASure[:SCALar]:POWER[:DC]?* 命令分别查询指定通道输出端子上测得的电压、电流或功率值。

返回格式

返回在指定通道输出端子上测得的电压、电流和功率值, 三个数值之间以逗号分开, 如 2.0000,0.0500,0.100。

举例

```
:MEAS:ALL? CH1 /*查询在 CH1 输出端子上测得的电压、电流和功率值, 返回 2.0000,0.0500,0.100*/
```

4.6.2 :MEASure[:SCALar]:CURRent[:DC]?

命令格式

```
:MEASure[:SCALar]:CURRent[:DC]? [<source>]
```

功能描述

查询在指定通道输出端子上测得的电流值。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3 ALL SERies PARallel}	-

说明

- **SERies**: 查询当前串联回路的总电流。
- **PARallel**: 查询当前并联回路的总电流。

若省略参数<source>, 则查询在当前选中通道输出端子上测得的电流值。

您也可以发送:*MEASure[:SCALar]:ALL[:DC]?* 命令同时查询指定通道输出端子上测得的电压、电流和功率值。

返回格式

返回在指定通道输出端子上测得的电流值, 如 0.0500。

举例

```
:MEAS:CURR? CH1 /*查询在 CH1 输出端子上测得的电流值, 返回 0.0500*/
```

4.6.3 :MEASure[:SCALar]:CURRent:DATA?**命令格式**

```
:MEASure[:SCALar]:CURRent:DATA? [<source>]
```

功能描述

查询在指定通道输出端子上测得的电流数据组。

参数

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

说明

若省略参数<source>, 则查询在当前选中通道输出端子上测得的电流数据组。

返回格式

查询返回一个以#开始的字符串。

例如: #90000046070.047698,0.047938,0.048428,0.048170,..., 其中#9000004607 为数据块头; 0.047698,0.047938,0.048428,0.048170,...为具体的电流数据。

- 数据块头用于描述数据流长度信息，格式为#NX...X。例如：#9000004607，其中 N 为 9，表示其后跟随的 9 个数据用于描述数据流的长度信息。其后的 000004607 用于描述数据流长度信息（4607 个字节）。
- 每组电流参数之间以逗号 “,” 隔开。例如：0.047698,0.047938,...表示第 1 组电流值为 0.047698 A；第 2 组电流值为 0.047938 A。

举例

```
:MEAS:CURR:DATA? CH1 /*查询在 CH1 输出端子上测得的 512 组电流值*/
```

4.6.4 :MEASure[:SCALar]:POWER[:DC]?

命令格式

```
:MEASure[:SCALar]:POWER[:DC]? [<source>]
```

功能描述

查询在指定通道输出端子上测得的功率值。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3 ALL SERies PARallel}	-

说明

- **SERies**: 查询当前串联回路的总功率。
- **PARallel**: 查询当前并联回路的总功率。

若省略参数 <source>，则查询在当前选中通道输出端子上测得的功率值。

您也可以发送发送 `:MEASure[:SCALar]:ALL[:DC]?` 命令同时查询在指定通道输出端子上测得的电压、电流和功率值。

返回格式

返回在指定通道输出端子上测得的功率值，如 0.100。

举例

```
:MEAS:POWE? CH1 /*查询在 CH1 输出端子上测得的功率值，返回 0.100*/
```

4.6.5 :MEASure[:SCALar][:VOLTage][:DC]?

命令格式

```
:MEASure[:SCALar][:VOLTage][:DC]? [<source>]
```

功能描述

查询在指定通道输出端子上测得的电压值。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3 ALL SERies PARallel}	-

说明

- **SERies**: 查询当前串联回路的总电压。
- **PARallel**: 查询当前并联回路的总电压。

若省略参数 <source> , 则查询在当前选中通道输出端子上测得的电压值。

您也可以发送发送 `:MEASure[:SCALar]:ALL[:DC]?` 命令同时查询在指定通道输出端子上测得的电压、电流和功率值。

返回格式

返回在指定通道输出端子上测得的电压值, 如 2.0000。

举例

```
:MEAS? CH1 /*查询在 CH1 输出端子上测得的电压值, 返回 2.0000*/
```

4.6.6 :MEASure[:SCALar][:VOLTage]:DATA?

命令格式

```
:MEASure[:SCALar][:VOLTage]:DATA? [<source>]
```

功能描述

查询在指定通道输出端子上测得的电压数据组。

参数

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

说明

若省略参数 <source> , 则查询在当前选中通道输出端子上测得的电压数据组。

返回格式

查询返回一个以#开始的字符串。

例如：#90000035830.5763,0.5764,0.5767,0.5765,0.5765,...，其中#9000003583 为数据块头；0.5763,0.5764,0.5767,0.5765,0.5765,...为具体的电压数据。

- 数据块头用于描述数据流长度信息，格式为#NX...X。例如：#9000003583，其中 N 为 9，表示其后跟随的 9 个数据用于描述数据流的长度信息。其后的 000003583 用于描述数据流长度信息（3583 个字节）。
- 每组电压参数之间以逗号“,” 隔开。例如：0.5763,0.5764,...表示第 1 组电压值为 0.5763 V；第 2 组电压值为 0.5764 V。

举例

```
:MEAS:DATA? CH1 /*查询在 CH1 输出端子上测得的 512 组电压值*/
```

4.7 :MEMory 命令子系统

:MEMory 命令用于将文件保存到内/外部存储器中的指定位置，删除、读取、锁定及解锁内部存储器中的指定文件等操作。本系列电源支持将如下五种类型的文件保存到内部存储器。

1. **状态文件 (RSF)**：存储当前系统的状态，包括各个通道的电压、电流、OVP、OCP、跟踪功能状态以及系统参数。
2. **任意波参数文件 (RTF)**：存储已编辑的任意波参数（每组参数的电压、电流和时间值）。
3. **位图文件 (BMP)**：存储的截图文件。
4. **录制文件 (ROF)**：日志录制打开期间，执行分析器，仪器以当前采样率实时存储采集到的波形数据。
5. **校准文件 (CLF)**：存储校准参数。

4.7.1 :MEMory:CATalog?

命令格式

```
:MEMory:CATalog?
```

功能描述

查询当前路径下的文件和文件夹列表。

参数

无。

说明

C 盘不支持存储文件夹。

返回格式

查询返回当前路径下的文件及文件夹名称，各名称之间使用逗号隔开，如 RIGOL0.BMP,cc.RSF，其中 RIGOL0.BMP 表示位图文件，cc.RSF 表示状态文件。

举例

```
:MEMory:CATalog? /*查询返回指定路径下的文件名, 返回 RIGOL0.BMP,cc.RSF*/
```

4.7.2 :MEMory:CDIRectory

命令格式

```
:MEMory:CDIRectory <directory_name>
```

```
:MEMory:CDIRectory?
```

功能描述

设置或查询当前工作路径。

参数

名称	类型	范围	默认值
<directory_name>	ASCII 字符串	见下文 说明	-

说明

- 参数 <directory_name> 必须为有效路径。
- 外部存储器 (D 盘、E 盘) 有效路径包括外部存储器 (D:/、E:/) 及外部存储器下已存储的文件夹 (如 D:/RIGOL)。

返回格式

查询返回当前工作路径, 如 C:/。

举例

```
:MEMory:CDIRectory C:/ /*设置当前工作路径是 C 盘*/
:MEMory:CDIRectory? /*查询当前工作路径, 并返回 C:/*/
```

4.7.3 :MEMory:DELeTe

命令格式

```
:MEMory:DELeTe <filename>
```

功能描述

删除当前路径下的文件和空文件夹。

参数

名称	类型	范围	默认值
<filename>	ASCII 字符串	见下文 说明	-

说明

- 参数 <filename> 为要删除的文件的文件名（文件名称以文件类型为后缀，如 STA.RSF）或空文件夹的名称。
- 仅当当前操作路径下存在指定的文件或空文件夹时，该命令有效。
- 若删除的文件夹中有内容，则无法删除，屏幕会弹出错误提示；若删除的文件被锁定（*:MEMory:LOCK*），该命令无效。

返回格式

无。

举例

```
:MEM:DEL NEW.RSF /*删除当前路径下名称为 NEW 的状态文件*/
```

4.7.4 :MEMory:DISK?

命令格式

```
:MEMory:DISK?
```

功能描述

查询当前有哪几个可用的外部存储器盘。

参数

无。

说明

无。

返回格式

返回可用盘符，如返回 D:/, E:/, 若没有可用外部存储器，返回 NONE。

举例

```
:MEM:DISK? /*查询当前可用的外部存储器盘符，返回 D:/, E:/*
```

4.7.5 :MEMory:LOAD

命令格式

```
:MEMory:LOAD <filename>
```

功能描述

读取当前工作路径下指定的已存文件，包括状态文件（.RSF）和任意波参数文件（.RTF）。

参数

名称	类型	范围	默认值
<filename>	ASCII 字符串	当前路径下已存文件的文件名（带后缀）	-

说明

- 仅当指定的存储位置已存该文件时，该命令有效。
- 您也可以使用 **RCL* 命令读取内部存储器中指定的状态文件。

返回格式

无。

举例

```
:MEM:LOAD NEW.RSF /*读取当前工作路径下名称为 NEW.RSF 的文件*/
```

4.7.6 :MEMory:LOCK

命令格式

```
:MEMory:LOCK <filename>,<bool>
```

```
:MEMory:LOCK? <filename>
```

功能描述

锁定或解锁 C 盘中指定的已存在文件，或查询 C 盘中指定的已存在文件是否已锁定。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	0
<filename>	ASCII 字符串	见下文 <i>说明</i>	-

说明

- <filename>为 C 盘中存在的文件名，包含后缀。
- 仅 C 盘文件支持锁定，只有存在该文件时，该命令有效。
- 已锁定的文件仅允许读取操作，不允许复制和删除操作。

返回格式

返回 0 或 1。

举例

```
:MEM:LOCK NEW.RSF,ON /*锁定 C 盘中名字为 NEW.RSF 的文件*/
:MEM:LOCK? NEW.RSF /*查询 C 盘中名字为 NEW.RSF 的文件是否被锁定, 返回 1*/
```

4.7.7 :MEMory:MDIRectory**命令格式**

```
:MEMory:MDIRectory <folder_name>
```

功能描述

在当前路径下创建一个文件夹。

参数

名称	类型	范围	默认值
<folder_name>	ASCII 字符串	见下文 说明	-

说明

- <folder_name>参数为新文件夹名称，最长为 255 个字符，可为中文、英文字符或数字（1 个中文字符按 2 个字符算）。
- C 盘不支持创建文件夹。当前工作路径为 C 盘时发送该命令，会弹出错误提示。

返回格式

无。

举例

```
:MEMory:MDIRectory NEW /*在当前路径下创建一个名称为 NEW 的文件夹*/
```

4.7.8 :MEMory:STORe**命令格式**

```
:MEMory:STORe <filename>
```

功能描述

在当前操作路径下以指定的文件名保存文件，包括仪器状态文件（.RSF）和任意波参数文件（.RTF）。

参数

名称	类型	范围	默认值
<filename>	ASCII 字符串	见下文 说明	-

说明

- <filename> 参数为带后缀 (.RSF 或 .RTF) 的文件名称，最长为 125 个字符，可为中文、英文字符或数字（1 个中文字符按 2 个字符算）。
- 若当前操作路径下已存有文件，该命令将直接覆盖原文件。若指定存储位置的已存文件被锁定 (:MEMory:LOCK)，该命令无效（不会直接覆盖原文件）。
- 录制（日志）文件的保存路径由 :ANALyzer:SAVE:ROUte 命令设置。
- 您也可以发送 *SAV 命令将当前的仪器状态保存到内部存储器中。

返回格式

无。

举例

```
:MEM:STOR NEW.RSF /*将文件保存到当前操作路径下，文件名为 NEW.RSF*/
```

4.7.9 :MEMory:VALid?

命令格式

```
:MEMory:VALid? <filename>
```

功能描述

查询当前工作路径下是否存在该文件。

参数

名称	类型	范围	默认值
<filename>	ASCII 字符串	见下文 说明	-

说明

<filename> 为带后缀的文件名，可以包括中文、英文字符或数字。

返回格式

返回 0 或 1。

举例

```
:MEM:STOR NEW.RSF /*在前工作路径下存储名字为 NEW 的状态文件*/
:MEM:VAL? NEW.RSF /*查询当前工作路径下是否存在名字为 NEW 的状态文件，返回 1*/
```

4.8 :OUTPut 命令子系统

:OUTPut 命令用于打开和关闭通道的输出、过压/过流保护功能、跟踪功能和 Sense 功能，查询通道的输出模式，设置和查询通道的过压/过流保护的相关信息。DP2031 各通道对应的过压/过流保护可设范围和默认值如下表所示。

表 4.35 过压/过流保护可设范围和默认值

通道			OVP/OCP 可设范围	OVP/OCP 默认值
DP2031	Range1	CH1(32V/3A)	1mV 至 35.2V/1mA 至 3.3A	35.2V/3.3A
		CH2(32V/3A)	1mV 至 35.2V/1mA 至 3.3A	35.2V/3.3A
		CH3(6V/5A)	1mV 至 6.6V/1mA 至 5.5A	6.6V/5.5A
	Range2 (选件)	CH1(32V/2A)	1mV 至 35.2V/1mA 至 2.2A	35.2V/2.2A
		CH2(32V/2A)	1mV 至 35.2V/1mA 至 2.2A	35.2V/2.2A
		CH3(6V/10A)	1mV 至 6.6V/1mA 至 11A	6.6V/11A

4.8.1 :OUTPut:CVCC?

命令格式

```
:OUTPut:CVCC? [<source>]
```

功能描述

查询指定通道输出模式（恒压/恒流/临界）。

参数

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

说明

- 本系列电源有三种输出模式：恒压输出（CV）、恒流输出（CC）和临界模式（UR）。在 CV 模式下，输出电压等于电压设置值，输出电流由负载决定；在 CC 模式下，输出电流等于电流设置值，输出电压由负载决定；UR 模式是介于 CV 和 CC 模式之间的临界模式。
- 若省略参数 <source>，表示查询当前通道的输出模式。

返回格式

返回 CV、CC 或 UR。

举例

```
:OUTPut:CVCC? CH1 /*查询通道 CH1 当前的输出模式，返回 CV*/
```

4.8.2 :OUTPut:MODE?

命令格式

```
:OUTPut:MODE? [<source>]
```

功能描述

查询指定通道输出模式（恒压/恒流/临界）。

参数

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

说明

- 本系列电源有三种输出模式：恒压输出（CV）、恒流输出（CC）和临界模式（UR）。在 CV 模式下，输出电压等于电压设置值，输出电流由负载决定；在 CC 模式下，输出电流等于电流设置值，输出电压由负载决定；UR 模式是介于 CV 和 CC 模式之间的临界模式。
- 若省略参数 <source>，表示查询当前通道的输出模式。

返回格式

返回 CV、CC 或 UR。

举例

```
:OUTP:MODE? CH1 /*查询通道 CH1 当前的输出模式，返回 CV*/
```

4.8.3 :OUTPut:OCP:ALAR?

命令格式

```
:OUTPut:OCP:ALAR? [<source>]
```

功能描述

查询指定通道是否发生过流保护（OCP）。

参数

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

说明

- 过流保护（OCP）是指当通道的实际输出电流超出过流保护值时，输出自动关闭。

- 您可以发送 `:OUTPut:OCP:CLEar` 命令清除指定通道上已发生的 OCP。
- 若省略参数 `<source>`，表示查询当前通道的输出模式。

返回格式

查询返回 0 或 1。

举例

```
:OUTPut:OCP:ALAR? CH1 /*查询通道 1 是否发生过过流保护*/
```

4.8.4 :OUTPut:OCP:QUES?

命令格式

```
:OUTPut:OCP:QUES? [<source>]
```

功能描述

查询指定通道当前是否发生过过流保护（OCP）。

参数

名称	类型	范围	默认值
<code><source></code>	离散型	{CH1 CH2 CH3}	-

说明

- 过流保护（OCP）是指当通道的实际输出电流超出过流保护值时，输出自动关闭。
- 您可以发送 `:OUTPut:OCP:CLEar` 命令清除指定通道上已发生的 OCP。
- 若省略参数 `<source>`，表示查询当前通道的输出模式。

返回格式

返回 1 或 0。

举例

```
:OUTP:OCP:QUES? CH1 /*查询 CH1 当前是否发生过 OCP*/
```

4.8.5 :OUTPut:OCP:CLEar

命令格式

```
:OUTPut:OCP:CLEar [<source>]
```

功能描述

清除指定通道上已发生的过流保护（OCP）的标识。

参数

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

说明

- 执行该命令前，必须保证引起指定通道上过流保护产生的原因已被清除（您可以减小输出电流值到过流保护值以下或者增大过流保护值到输出电流值以上）。
- 若省略参数<source>，表示清除当前通道上已发生的 OCP 的标识。
- 您也可以发送 `[:SOURce[<n>]]:CURRent:PROtection:CLEar` 命令清除指定通道上已发生的 OCP 标识并打开相应通道的输出。
- 您可以发送 `:OUTPut:OCP:ALAR?` 或 `:OUTPut:OCP:QUES?` 命令查询指定通道当前是否已发生 OCP。

返回格式

无。

举例

```
:OUTP:OCP:QUES? CH1 /*查询 CH1 当前是否发生过 OCP，返回 YES*/
:OUTP:OCP:CLE CH1 /*清除 CH1 上已发生的 OCP 的标识*/
:OUTP:OCP:QUES? CH1 /*查询 CH1 当前是否发生 OCP，返回 NO*/
```

4.8.6 :OUTPut:OCP:DElay

命令格式

```
:OUTPut:OCP:DElay [<source>,{<value>|<lim>}]
```

```
:OUTPut:OCP:DElay? [<source>][,<lim>]
```

功能描述

设置或查询指定通道的过流保护（OCP）延时开启时间，默认单位毫秒。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3}	-
<value>	实型	0 至 1000 ms	10 ms
<lim>	离散型	{MINimum MAXimum}	-

说明

- 若省略参数<source>，表示对当前通道执行相应操作。

- 选择“MINimum”、“MAXimum”，表示设定过流保护延时的最小值、最大值。

返回格式

查询返回过流保护延时开启时间，如 200ms。

举例

```
:OUTPut:OCP:DElay CH1 200 /*设置通道 1 过流保护延时开启时间为 200 ms*/
:OUTPut:OCP:DElay? CH1 /*查询通道 1 过流保护延时开启时间，返回 200ms*/
```

4.8.7 :OUTPut:OCP[:STATe]

命令格式

```
:OUTPut:OCP[:STATe] [<source>,<bool>
```

```
:OUTPut:OCP[:STATe]? [<source>]
```

功能描述

设置或查询指定通道的过流保护（OCP）功能的状态。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3}	-
<bool>	布尔型	{0 1 ON OFF}	0

说明

- 打开过流保护功能时，若输出电流超出当前设置的过流保护限值（`:OUTPut:OCP:VALue`），输出自动关闭。您可以发送`:OUTPut:OCP:QUES?`或`:OUTPut:OCP:ALAR?`命令查询指定通道当前是否发生过 OCP。
- 若省略参数<source>，表示对当前通道执行相应操作。

返回格式

返回 0 或 1。

举例

```
:OUTP:OCP CH1,1 /*打开 CH1 的 OCP 功能*/
:OUTP:OCP? CH1 /*查询 CH1 的 OCP 功能的状态，返回 1*/
```

4.8.8 :OUTPut:OCP:VALue

命令格式

```
:OUTPut:OCP:VALue [<source>,<value>|<lim>}
```

```
:OUTPut:OCP:VALue? [<source>],<lim>]
```

:OUTPut:OCP:VALue? [<lim>]

功能描述

设置或查询指定通道的过流保护（OCP）值。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3}	-
<value>	实型	请参考表 4.35: 过压/过流保护可设范围和默认值	
<lim>	离散型	{MINimum MAXimum}	-

说明

- 打开过流保护功能时，若输出电流超出当前设置的过流保护值，输出自动关闭。您可以发送 `:OUTPut:OCP:QUES?` 或 `:OUTPut:OCP:ALAR?` 命令查询指定通道当前是否发生过 OCP。
- 若省略参数 <source>，表示对当前通道执行相应操作。
- “MINimum”、“MAXimum”，表示设定过流保护值的最小值、最大值。
- 您也可以发送 `[[:SOURce[<n>]]:CURRent:PROTection[:LEVel]]` 命令设置指定通道的 OCP 值。

返回格式

返回过流保护值，如 5.0000。

举例

```
:OUTP:OCP:VAL CH1,5 /*设置 CH1 的 OCP 值为 5A*/
:OUTP:OCP:VAL? CH1 /*查询 CH1 当前的 OCP 值，返回 5.0000*/
```

4.8.9 :OUTPut:OVP:ALAR?

命令格式

:OUTPut:OVP:ALAR? [<source>]

功能描述

设置或查询指定通道是否发生过压保护（OVP）。

参数

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

说明

- 过压保护（OVP）是指当通道的实际输出电压超出过压保护值时，输出自动关闭。
- 若省略参数<source>，表示查询当前通道是否发生过 OVP。
- 您可以发送 `:OUTPut:OVP:CLEar` 命令清除指定通道上已发生的 OVP。

返回格式

返回 1 或 0。

举例

```
:OUTP:OVP:ALAR? CH1 /*查询 CH1 当前是否发生过 OVP*/
```

4.8.10 :OUTPut:OVP:QUES?**命令格式**

```
:OUTPut:OVP:QUES? [<source>]
```

功能描述

查询指定通道当前是否发生过过压保护（OVP）。

参数

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

说明

- 过压保护（OVP）是指当通道的实际输出电压超出过压保护值时，输出自动关闭。
- 若省略参数<source>，表示查询当前通道是否发生过 OVP。
- 您可以发送 `:OUTPut:OVP:CLEar` 命令清除指定通道上已发生的 OVP。

返回格式

返回 1 或 0。

举例

```
:OUTP:OVP:QUES? CH1 /*查询 CH1 当前是否发生过 OVP*/
```

4.8.11 :OUTPut:OVP:CLEar**命令格式**

```
:OUTPut:OVP:CLEar [<source>]
```

功能描述

清除指定通道上已发生的过压保护（OVP）的标识。

参数

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

说明

- 执行该命令前，必须保证引起指定通道上过压保护产生的原因已被清除（您可以减小输出电压值到过压保护值以下或者增大过压保护值到输出电压值以上）。
- 若省略参数<source>，表示清除当前通道上已发生的 OVP 的标识。
- 您也可以发送 `[:SOURce[<n>]]:VOLTage:PROtection:CLEar` 命令清除指定通道上已发生的 OVP 标识并打开相应通道的输出。
- 您可以发送 `:OUTPut:OVP:QUES?` 或 `:OUTPut:OVP:ALAR?` 命令查询指定通道当前是否已发生 OVP。

返回格式

无。

举例

```
:OUTP:OVP:CLE CH1 /*清除 CH1 上已发生的 OVP 的标识*/
```

4.8.12 :OUTPut:OVP[:STATe]

命令格式

```
:OUTPut:OVP[:STATe] [<source>,<bool>
```

```
:OUTPut:OVP[:STATe]? [<source>
```

功能描述

打开或关闭指定通道的过压保护（OVP）功能，查询指定通道过压保护（OVP）功能的状态。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3}	-
<bool>	布尔型	{0 1 ON OFF}	0

说明

- 打开过压保护功能时，若输出电压超出当前设置的过压保护值（`:OUTPut:OVP:VALue`），输出自动关闭。您可以发送 `:OUTPut:OVP:QUES?` 或 `:OUTPut:OVP:ALAR?` 命令查询指定通道当前是否发生过 OVP。
- 若省略参数<source>，表示对当前通道执行相应操作。

返回格式

返回 1 或 0。

举例

```
:OUTP:OVP CH1,1 /*打开 CH1 的 OVP 功能*/
:OUTP:OVP? CH1 /*查询 CH1 的 OVP 功能的状态, 返回 1*/
```

4.8.13 :OUTPut:OVP:VALue

命令格式

```
:OUTPut:OVP:VALue [<source>,<value>|<lim>}
:OUTPut:OVP:VALue? [<source>],<lim>]
:OUTPut:OVP:VALue? [<lim>]
```

功能描述

设置或查询指定通道的过压保护 (OVP) 值。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3}	-
<value>	实型	请参考表 4.35: 过压/过流保护可设范围和默认值	
<lim>	离散型	{MINimum MAXimum}	-

说明

- 打开过压保护功能时，若输出电压超出当前设置的过压保护值，输出自动关闭。您可以发送 `:OUTPut:OVP:QUES?` 或 `:OUTPut:OVP:ALAR?` 命令查询指定通道当前是否发生过 OVP。
- 若省略参数 <source>，表示设置或查询当前通道的过压保护限值。
- “MINimum”、“MAXimum”，表示设定过压保护值的最小值、最大值。
- 您也可以发送 `[[:SOURce[<n>]]:VOLTage:PROtection[:LEVel]` 命令设置指定通道的 OVP 值。

返回格式

返回过压保护值，如 8.800。

举例

```
:OUTP:OVP:VAL CH1,8.8 /*设置 CH1 的 OVP 值为 8.8 V*/
:OUTP:OVP:VAL? CH1 /*查询 CH1 的 OVP 值, 返回 8.800*/
```

4.8.14 :OUTPut:PAIR

命令格式

```
:OUTPut:PAIR <type>
```

```
:OUTPut:PAIR?
```

功能描述

设置或查询通道连接方式。

参数

名称	类型	范围	默认值
<type>	离散型	{OFF PARAllel SERies}	OFF

说明

- **OFF**: 设置通道 1 和通道 2 工作在独立模式。
- **PARAllel**: 设置通道 1 和通道 2 工作在内部并联模式。
- **SERies**: 设置通道 1 和通道 2 工作在内部串联模式。

返回格式

查询返回通道连接方式，如 SERIES。

举例

```
:OUTPut:PAIR SERies /*设置通道 1 通道 2 内部串联*/
:OUTPut:PAIR? /*查询通道连接方式，返回 SERIES*/
```

4.8.15 :OUTPut[:STATe]

命令格式

```
:OUTPut[:STATe] [<source>,<bool>
```

```
:OUTPut[:STATe]? [<source>
```

功能描述

设置或查询指定通道的输出状态。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3 ALL}	-
<bool>	布尔型	{0 1 ON OFF}	0

说明

- 打开通道输出之前，请确保当前的设置不会对所连接的设备造成影响。
- 若省略参数<source>，表示对当前通道执行相应操作。

返回格式

返回 1 或 0。

举例

```
:OUTP CH1,ON /*打开 CH1 的输出*/
:OUTP? CH1 /*查询 CH1 当前的输出状态，返回 1*/
```

4.8.16 :OUTPut:TRACk[:STATe]**命令格式**

```
:OUTPut:TRACk[:STATe] <bool>
```

```
:OUTPut:TRACk[:STATe]?
```

功能描述

设置或查询跟踪功能的开关状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	0

说明

- 对于支持跟踪功能的两个通道（通道 1，通道 2），当打开跟踪功能时，修改任意一个通道的通道设置（包括电压电流设置值、过压过流设置值、过压过流开关状态），另一个通道的设置会随之改变。
- 跟踪功能默认为关闭状态，常用于为运算放大器或其它电路提供对称的电压。
- 跟踪功能只跟踪电压电流设置值，与实际输出无关。

返回格式

返回 1 或 0。

举例

```
:OUTP:TRAC ON /*打开跟踪功能*/
:OUTP:TRAC? /*查询跟踪功能的状态，返回 1*/
```

4.9 :SOURce 命令子系统

:SOURce 命令用于设置指定通道的电压、电流、OVP 和 OCP 的值。:APPLY 命令是通过远程接口对电源进行编程的最直接的方法，而:SOURce 命令可以用来改变单个参数，具有更多的灵活性。

4.9.1 [:SOURce[<n>]]:CURRent[:LEVel][:IMMediate][:AMPLitude]

命令格式

```
[:SOURce[<n>]]:CURRent[:LEVel][:IMMediate][:AMPLitude]{<current>|<lim>|<amp>}
```

```
[:SOURce[<n>]]:CURRent[:LEVel][:IMMediate][:AMPLitude]?[<lim>]
```

功能描述

设置或查询指定通道的电流值。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-
<current>	实型	0 至指定通道最大电流值	0.1 A
<lim>	离散型	{MINimum MAXimum DEFault}	-
<amp>	离散型	{UP DOWN}	-

说明

- 若省略[:SOURce[<n>]]或[<n>]，表示设置当前选中通道的相应参数。
- 选择参数<current>时，直接设置指定通道的电流值；选择参数 MINimum、MAXimum 或 DEFault 时，设置指定通道的电流值为该通道电流可设范围的最小值、最大值或默认值；选择参数 UP 或 DOWN 时，根据命令 `[:SOURce[<n>]]:CURRent[:LEVel][:IMMediate]:STEP[:INCRement]` 中设置的步进值增大或减小电流值。
- 您也可以发送 `:APPLY` 命令设置指定通道的电流值。

返回格式

返回指定通道的电流设置值，如 1.5000。

举例

```
:CURR 1.5 /*设置当前通道的电流为 1.5A*/
:CURR? /*查询当前通道的电流，返回 1.5000*/
```

4.9.2 [:SOURce[<n>]]:CURRent[:LEVel] [:IMMediate]:STEP[:INCRement]

命令格式

```
[:SOURce[<n>]]:CURRent[:LEVel][:IMMediate]:STEP[:INCRement]{<numeric value>|<def>}
```

```
[:SOURce[<n>]]:CURRent[:LEVel][:IMMediate]:STEP[:INCRement]?[<def>]
```

功能描述

设置或查询指定通道电流变化的步进值。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-
<numeric value>	实型	0 至指定通道最大电流值	见说明
<def>	离散型	{DEFault}	-

说明

- 若省略[:SOURce[<n>]]或[<n>]，表示设置当前选中通道的相应参数。
- 参数<numeric value>为指定的步进值，DEFault 表示默认步进值。参数<numeric value>的默认值如下表所示。

通道	默认步进值
CH1	0.0001A
CH2	0.0001A
CH3	0.001A

- 当设置[:SOURce[<n>]]:CURRent[:LEVel][:IMMediate]:AMPLitude]命令中的参数为 UP 或 DOWN 时，执行该命令，电流设置值将根据该命令中设置的步进值增大或减小。

返回格式

返回指定通道电流变化的步进值，如 0.1000。

举例

```
:CURR:STEP 0.1 /*设置当前通道电流变化的步进值为 0.1A*/
:CURR:STEP? /*查询当前通道电流变化的步进值，返回 0.1000*/
```

4.9.3 [:SOURce[<n>]]:CURRent:PROTection:CLEar

命令格式

```
[:SOURce[<n>]]:CURRent:PROTection:CLEar
```

功能描述

清除指定通道上已发生的过流保护（OCP）电路和标识并打开相应通道的输出。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-

说明

- 您可以发送 `[:SOURce[<n>]]:CURRent:PROTection:TRIPped?` 命令查询指定通道是否已发生过 OCP。
- 执行该命令前，必须保证引起指定通道上 OCP 产生的原因已被清除（您可以减小输出电流值到 OCP 值以下或者增大 OCP 值到输出电流值以上），否则该命令无效。执行该命令将清除指定通道上已发生的 OCP 电路和标识并打开相应通道的输出。
- 若省略 `[:SOURce[<n>]]` 或 `[<n>]`，表示清除当前通道上已发生的 OCP 电路和标识。
- 您可以发送 `:OUTPut:OCP:CLEar` 命令仅清除指定通道上已发生的 OCP 的标识。

返回格式

无。

举例

```
:CURR:PROT:TRIP? /*查询当前通道是否已发生 OCP, 返回 1*/
:CURR:PROT:CLE /*清除当前通道上已发生的 OCP 电路和标识*/
:CURR:PROT:TRIP? /*查询当前通道是否已发生 OCP, 返回 0*/
```

4.9.4 [:SOURce[<n>]]:CURRent:PROTection[:LEVel]

命令格式

```
[:SOURce[<n>]]:CURRent:PROTection[:LEVel] {<current>|<lim>}
```

```
[:SOURce[<n>]]:CURRent:PROTection[:LEVel]? [<lim>]
```

功能描述

设置或查询指定通道的过流保护（OCP）值。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-

名称	类型	范围	默认值
<current>	实型	请参考表 4.35: 过压/过流保护可设范围和默认值	
<lim>	离散型	{MINimum MAXimum}	-

说明

- 打开指定通道的 OCP 功能 (`[:SOURce[<n>]]:CURRent:PROTection:STATe`) 时, 若输出电流超出当前设置的 OCP 值, 输出自动关闭。您可以发送 `[:SOURce[<n>]]:CURRent:PROTection:TRIPped?` 命令查询指定通道当前是否已发生 OCP。
- 若省略 `[:SOURce[<n>]]` 或 `[<n>]`, 表示设置当前选中通道的相应参数。
- 您也可以发送 `:OUTPut:OCP:VALue` 命令设置指定通道的 OCP 值。

返回格式

返回指定通道的过流保护值, 如 2.0000。

举例

```
:CURR:PROT 2 /*设置当前通道的过流保护值为 2 A*/
:CURR:PROT? /*查询当前通道的过流保护值, 返回 2.0000*/
```

4.9.5 `[:SOURce[<n>]]:CURRent:PROTection:STATe`

命令格式

```
[:SOURce[<n>]]:CURRent:PROTection:STATe <bool>
```

```
[:SOURce[<n>]]:CURRent:PROTection:STATe?
```

功能描述

设置或查询指定通道的过流保护 (OCP) 功能状态。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-
<bool>	布尔型	{ON OFF 1 0}	0

说明

- 打开指定通道的 OCP 功能时, 若输出电流超出当前设置的 OCP 值, 输出自动关闭。您可以发送 `[:SOURce[<n>]]:CURRent:PROTection:TRIPped?` 命令查询指定通道当前是否已发生 OCP。
- 若省略 `[:SOURce[<n>]]` 或 `[<n>]`, 表示设置当前选中通道的相应参数。

- 您也可以发送 `:OUTPut:OCP[:STATe]` 命令打开或关闭指定通道的过流保护（OCP）功能。
- 您可以发送 `[:SOURce[<n>]]:CURRent:PROTection[:LEVel]` 命令查询指定通道当前的 OCP 值。

返回格式

返回 0 或 1。

举例

```
:CURR:PROT:STAT ON /*打开当前通道的 OCP 功能*/
:CURR:PROT:STAT? /*查询当前通道的 OCP 功能状态, 返回 1*/
```

4.9.6 [:SOURce[<n>]]:CURRent:PROTection:TRIPped?

命令格式

```
[:SOURce[<n>]]:CURRent:PROTection:TRIPped?
```

功能描述

查询指定通道当前是否已发生过流保护（OCP）。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-

说明

- 过流保护（OCP）是指当通道的实际输出电流超出 OCP 值时，输出自动关闭。
- 若省略 `[:SOURce[<n>]]` 或 `[<n>]`，表示查询当前通道是否已发生 OCP。
- 您也可以发送 `:OUTPut:OCP:ALAR?` 或 `:OUTPut:OCP:QUES?` 命令查询指定通道当前是否已发生 OCP。
- 您可以发送 `[:SOURce[<n>]]:CURRent:PROTection:CLEar` 命令清除指定通道上已发生的 OCP 标识。

返回格式

返回 1 或 0。

举例

```
:CURR:PROT:TRIP? /*查询当前通道是否已发生 OCP, 返回 1*/
```


4.9.7 [:SOURce[<n>]]:VOLTage[:LEVel][:IMMediate] [:AMPLitude]

命令格式

```
[:SOURce[<n>]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]{<voltage>|<lim>|<amp>}
```

```
[:SOURce[<n>]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]?[<lim>]
```

功能描述

设置或查询指定通道的电压值。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-
<voltage>	实型	请参考表 4.35: 过压/过流保护可设范围和默认值	
<lim>	离散型	{MINimum MAXimum DEFault}	-
<amp>	离散型	{UP DOWN}	-

说明

- 若省略[:SOURce[<n>]]或[<n>]，表示设置当前选中通道的相应参数。
- 选择参数<voltage>时，直接设置指定通道的电压值；选择参数 MINimum、MAXimum 或 DEFault 时，设置指定通道的电压值为该通道电压可设范围的最小值、最大值或默认值；选择参数 UP 或 DOWN 时，根据命令 `[:SOURce[<n>]]:VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]` 中设置的步进值增大或减小电压值。
- 您也可以发送 `.APPLY` 命令设置指定通道的电压值和电流值。

返回格式

返回指定通道的电压设置值，如 7.500。

举例

```
:VOLT 7.5 /*设置当前通道的电压值为 7.5V*/
:VOLT? /*查询当前通道的电压设置值，返回 7.500*/
```

4.9.8 [:SOURce[<n>]]:VOLTage[:LEVel] [:IMMediate]:STEP[:INCRement]

命令格式

```
[:SOURce[<n>]]:VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]{<voltage>|<def>}
```

```
[:SOURce[<n>]]:VOLTage[:LEVel][:IMMediate]:STEP[:INCRement]?[<def>]
```

功能描述

设置或查询指定通道电压变化的步进值。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-
<voltage>	实型	0 至指定通道最大电压值	请参考 说明
<def>	离散型	{DEFault}	-

说明

- 若省略[:SOURce[<n>]]或[<n>]，表示设置当前选中通道的相应参数。
- 参数<voltage>为指定的步进值，DEFault 表示默认步进值。参数<voltage>的默认值如下表所示：

通道	默认步进值
CH1	0.001 V
CH2	0.001 V
CH3	0.001 V

- 当[:SOURce[<n>]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]命令中的参数为 UP 或 DOWN 时，执行命令，电压值将根据本命令中设置的步进值增大或减小。

返回格式

返回指定通道电压变化的步进，如 0.100。

举例

```
:VOLT:STEP 0.1 /*设置当前通道电压变化的步进值为 0.1V*/
:VOLT:STEP? /*查询当前通道电压变化的步进值，返回 0.100*/
```

4.9.9 [:SOURce[<n>]]:VOLTage:PROTection:CLEar

命令格式

```
[:SOURce[<n>]]:VOLTage:PROTection:CLEar
```

功能描述

清除指定通道上已发生的过压保护（OVP）电路和标识并打开相应通道的输出。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-

说明

- 您可以发送 `[:SOURce[<n>]]:VOLTage:PROTection:TRIPped?` 命令查询指定通道是否已发生 OVP。
- 执行该命令前，必须保证引起指定通道上 OVP 产生的原因已被清除（您可以减小输出电压值到 OVP 值以下或者增大 OVP 值到输出电压值以上），否则该命令无效。执行该命令将清除指定通道上已发生的 OVP 电路和标识并打开相应通道的输出。
- 若省略 `[:SOURce[<n>]]` 或 `[<n>]`，表示清除当前通道上已发生的 OVP 电路和标识。
- 您可以发送 `:OUTPut:OVP:CLEar` 命令仅清除指定通道上已发生的 OVP 的标识。

返回格式

无。

举例

```
:VOLT:PROT:TRIP? /*查询当前通道是否已发生 OVP, 返回 1*/
:VOLT:PROT:CLE /*清除当前通道上已发生的 OVP 电路和标识*/
:VOLT:PROT:TRIP? /*查询当前通道是否已发生 OVP, 返回 0*/
```

4.9.10 [:SOURce[<n>]]:VOLTage:PROTection[:LEVel]

命令格式

```
[:SOURce[<n>]]:VOLTage:PROTection[:LEVel] {<voltage>|<lim>}
```

```
[:SOURce[<n>]]:VOLTage:PROTection[:LEVel]? [<lim>]
```

功能描述

设置或查询指定通道的过压保护（OVP）值。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-

名称	类型	范围	默认值
<voltage>	实型	请参考表 4.35: 过压/过流保护可设范围和默认值	
<lim>	离散型	{MINimum MAXimum}	-

说明

- 打开指定通道的 OVP 功能 (`[:SOURce[<n>]]:VOLTage:PROTection:STATe`) 时, 若输出电压超出当前设置的 OVP 值, 输出自动关闭。您可以发送 `[:SOURce[<n>]]:VOLTage:PROTection:TRIPped?` 命令查询指定通道当前是否已发生 OVP。
- 若省略 `[:SOURce[<n>]]` 或 `[<n>]`, 表示设置当前选中通道的相应参数。
- 您也可以发送 `:OUTPut:OVP:VALue` 命令设置指定通道的 OVP 值。

返回格式

返回指定通道的过压保护值, 如 8.800。

举例

```
:VOLT:PROT 8.8 /*设置当前通道的过压保护值为 8.8V*/
:VOLT:PROT? /*查询当前通道的过压保护值, 返回 8.800*/
```

4.9.11 `[:SOURce[<n>]]:VOLTage:PROTection:STATe`

命令格式

```
[:SOURce[<n>]]:VOLTage:PROTection:STATe <bool>
```

```
[:SOURce[<n>]]:VOLTage:PROTection:STATe?
```

功能描述

设置或查询指定通道的过压保护 (OVP) 功能状态。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-
<bool>	布尔型	{0 1 ON OFF}	0

说明

- 打开指定通道的 OVP 功能 (`[:SOURce[<n>]]:VOLTage:PROTection:STATe`) 时, 若输出电压超出当前设置的 OVP 值, 输出自动关闭。您可以发送 `[:SOURce[<n>]]:VOLTage:PROTection:TRIPped?` 命令查询指定通道当前是否已发生 OVP。
- 若省略 `[:SOURce [<n>]]` 或 `[<n>]`, 表示设置当前选中通道的相应参数。

- 您也可以发送 `:OUTPut:OVP[:STATe]` 命令打开或关闭指定通道的过压保护（OVP）功能。
- 您可以发送 `[:SOURce[<n>]]:VOLTage:PROTection[:LEVel]` 命令查询指定通道当前的 OVP 值。

返回格式

返回 1 或 0。

举例

```
:VOLT:PROT:STAT ON /*打开当前通道的 OVP 功能*/
:VOLT:PROT:STAT? /*查询当前通道的 OVP 功能状态, 返回 1*/
```

4.9.12 [:SOURce[<n>]]:VOLTage:PROTection:TRIPped?

命令格式

```
[:SOURce[<n>]]:VOLTage:PROTection:TRIPped?
```

功能描述

查询指定通道当前是否已发生过压保护（OVP）。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-

说明

- 过压保护（OVP）是指当通道的实际输出电压超出 OVP 值时，输出自动关闭。
- 若省略 `[:SOURce[<n>]]` 或 `[<n>]`，表示查询当前通道是否已发生 OVP。
- 您也可以发送 `:OUTPut:OVP:ALAR?` 或 `:OUTPut:OVP:QUES?` 命令查询指定通道当前是否已发生 OVP。
- 您可以发送 `[:SOURce[<n>]]:VOLTage:PROTection:CLEar` 命令清除指定通道上已发生的 OVP 电路和标识。

返回格式

返回 1 或 0。

举例

```
:VOLT:PROT:TRIP? /*查询当前通道是否已发生 OVP, 返回 1*/
```

4.10 :STATus 命令子系统

`:STATus` 命令用于设置、查询可疑状态寄存器和操作状态寄存器。

4.10.1 :STATus:OPERation:CONDition?

命令格式

```
:STATus:OPERation:CONDition?
```

功能描述

查询操作状态寄存器的条件寄存器。

参数

无。

说明

无。

返回格式

如果仪器在校准过程中则返回+1，否则返回+0。

举例

```
:STATus:OPERation:CONDition? /* 查询操作寄存器的条件寄存器，返回+0*/
```

4.10.2 :STATus:OPERation:ENABLE

命令格式

```
:STATus:OPERation:ENABLE <value>
```

```
:STATus:OPERation:ENABLE?
```

功能描述

使能操作状态寄存器的使能寄存器中的位。

查询操作状态寄存器的使能寄存器中使能的位。

参数

名称	类型	范围	默认值
<value>	整型	见下文 说明	-

说明

- 参数 <value> 是一个十进制值，该值与操作状态寄存器的使能寄存器中要使能的位的二进制加权和相对应。
- 参数 <value> 设为 0 时，执行该命令可以清除操作状态寄存器的使能寄存器。

返回格式

返回一个十进制值，该值与操作状态寄存器的使能寄存器中使能的位的二进制加权和相对应，如+16。

举例

```
STATus:OPERation:ENABle 16 /*使能寄存器的第 4 位*/
:STATus:OPERation:ENABle? /*查询使能的位，返回+16*/
```

4.10.3 :STATus:OPERation[:EVENT]?**命令格式**

```
:STATus:OPERation[:EVENT]?
```

功能描述

查询操作状态寄存器的事件寄存器。

参数

无。

说明

执行该命令，返回一个十进制值（该值与此寄存器中所有位的二进制加权和相对应），同时清除此寄存器的状态。

返回格式

查询返回一个十进制值，该值与操作状态寄存器的事件寄存器中所有位的二进制加权和相对应，如+17。

举例

```
:STAT:OPER? /*查询操作状态寄存器的事件寄存器的值*/
```

4.10.4 :STATus:PRESet**命令格式**

```
:STATus:PRESet
```

功能描述

将可疑状态寄存器组的使能寄存器设置为上电时的默认值。

参数

无。

说明

无。

返回格式

无。

举例

```
:STATus:PRESet /*将可疑状态寄存器组的使能寄存器设置为上电时的默认值*/
```

4.10.5 :STATus:QUESTIONable:ENABLE**命令格式**

```
:STATus:QUESTIONable:ENABLE <enable value>
```

```
:STATus:QUESTIONable:ENABLE?
```

功能描述

使能可疑状态寄存器的使能寄存器中的位。

查询可疑状态寄存器的使能寄存器中使能的位。

参数

名称	类型	范围	默认值
<enable value>	整型	见下文 说明	-

说明

- 参数<enable value>是一个十进制值，该值与可疑状态寄存器的使能寄存器中要使能的位的二进制加权和相对应。
- 使能可疑状态寄存器的使能寄存器中的位后，系统向状态字节寄存器报告相应位的状态。
- 参数<enable value>设为 0 时，执行该命令可以清除可疑状态寄存器的使能寄存器。

返回格式

返回一个十进制值，该值与可疑状态寄存器的使能寄存器中使能的位的二进制加权和相对应，如+17。

举例

```
:STAT:QUES:ENAB 17 /*使能可疑状态寄存器的使能寄存器中的 bit0 和 bit4*/
:STAT:QUES:ENAB? /*查询可疑状态寄存器的使能寄存器中使能的位，返回+17*/
```

4.10.6 :STATus:QUESTIONable[:EVENT]?**命令格式**

```
:STATus:QUESTIONable[:EVENT]?
```

功能描述

查询可疑状态寄存器的事件寄存器。

参数

无。

说明

- 执行该命令，返回一个十进制值（该值与此寄存器中所有位的二进制加权和相对应），同时清除此寄存器的状态。
- 可疑状态寄存器的事件寄存器中的位被锁存，读取该寄存器将清除该寄存器。您也可以使用命令 **CLS* 清除此寄存器

返回格式

返回一个十进制值，该值与可疑状态寄存器的事件寄存器中所有位的二进制加权和相对应，如 +17。

举例

```
:STAT:QUES? /*查询可疑状态寄存器的事件寄存器，返回+17*/
```

4.10.7 :STATus:QUESTionable:INSTrument:ENABle

命令格式

```
:STATus:QUESTionable:INSTrument:ENABle <enable value>
```

```
:STATus:QUESTionable:INSTrument:ENABle?
```

功能描述

使能通道可疑状态寄存器的使能寄存器中的位。

查询通道可疑状态寄存器的使能寄存器中使能的位。

参数

名称	类型	范围	默认值
<enable value>	整型	见下文 <i>说明</i>	-

说明

- 参数 <enable value> 是一个十进制值，该值与通道可疑状态寄存器的使能寄存器中要使能的位的二进制加权和相对应。
- 使能通道可疑状态寄存器的使能寄存器中的位后，系统向可疑状态寄存器报告相应位的状态。
- 参数 <enable value> 设为 0 时，执行该命令可以清除通道可疑状态寄存器的使能寄存器。

返回格式

返回一个十进制值，该值与通道可疑状态寄存器的使能寄存器中使能的位的二进制加权和相对应，如 +14。

举例

```
:STAT:QUES:INST:ENAB 14 /*使能通道可疑状态寄存器的使能寄存器中的 bit1、bit2
和 bit3 (INST(n) event summary, 通道(n) 事件 SUMMARY 位, n 为 1、2、3) */
:STAT:QUES:INST:ENAB? /*查询通道可疑状态寄存器的使能寄存器中使能的位, 返回
+14*/
```

4.10.8 :STATus:QUESTionable:INSTrument[:EVENT]?**命令格式**

```
:STATus:QUESTionable:INSTrument[:EVENT]?
```

功能描述

查询通道可疑状态寄存器的事件寄存器。

参数

无。

说明

- 执行该命令，返回一个十进制值（该值与此寄存器中所有位的二进制加权和相对应），同时清除此寄存器的状态。
- 通道可疑状态寄存器的事件寄存器中的位被锁存，读取该寄存器将清除该寄存器。您也可以使用命令 **CLS* 清除该寄存器。

返回格式

返回一个十进制值，该值与通道可疑状态寄存器的事件寄存器中所有位的二进制加权和相对应，如+10。

举例

```
:STAT:QUES:INST? /*查询通道可疑状态寄存器的事件寄存器, 返回+10*/
```

4.10.9 :STATus:QUESTionable:INSTrument:ISUMmary[<n>]:CONDition?**命令格式**

```
:STATus:QUESTionable:INSTrument:ISUMmary[<n>]:CONDition?
```

功能描述

查询指定通道的输出状态。

参数

名称	类型	范围	默认值
<n>	整型	{1 2 3}	-

说明

- 若省略参数[<n>]，表示查询当前通道的输出状态。
- 执行该命令，返回+0、+1、+2 或+3，各返回值的含义见下表。

返回值	含义
+0	通道输出关闭
+1	通道输出状态为 CC (恒流) 模式
+2	通道输出状态为 CV (恒压) 模式
+3	通道输出状态为 UR (临界) 模式

返回格式

返回+0、+1、+2 或+3。

举例

```
:STAT:QUES:INST:ISUM1:COND? /*查询通道 1 输出状态，返回+1*/
```

4.10.10 :STATus:QUESTionable:INSTrument:ISUMmary[<n>]:ENABle

命令格式

```
:STATus:QUESTionable:INSTrument:ISUMmary[<n>]:ENABle <enable value>
```

```
:STATus:QUESTionable:INSTrument:ISUMmary[<n>]:ENABle?
```

功能描述

使能指定通道可疑状态 SUMMARY 寄存器的使能寄存器中的位。

查询指定通道可疑状态 SUMMARY 寄存器的使能寄存器中使能的位。

参数

名称	类型	范围	默认值
<n>	离散型	{1 2 3}	-
<enable value>	整型	见下文 说明	-

说明

- 该命令仅适用于多通道型号，多通道型号有多个通道可疑状态 SUMMARY 寄存器，参数 [<n>] 为 1、2、3 时分别代表通道 1、通道 2、通道 3 可疑状态 SUMMARY 寄存器。若省略参数 [<n>]，表示查询当前通道可疑状态 SUMMARY 寄存器的使能寄存器。

- 参数<enable value>是一个十进制值，该值与通道可疑状态 SUMMARY 寄存器的使能寄存器中要使能的位的二进制加权和相对应。
- 使能指定通道可疑状态 SUMMARY 寄存器的使能寄存器中的位后，系统向通道可疑状态寄存器报告相应位的状态。
- 参数<enable value>设为 0 时，执行该命令可以清除通道可疑状态 SUMMARY 寄存器的使能寄存器。

返回格式

返回一个十进制值，该值与通道可疑状态 SUMMARY 寄存器的使能寄存器中使能的位的二进制加权和相对应，如+9。

举例

```
:STAT:QUES:INST:ISUM1:ENAB 9 /*使能通道 1 可疑状态 SUMMARY 寄存器的使能寄存器中的 bit0 (Voltage, 恒流模式) 和 bit3 (OCP, 过流保护) */
:STAT:QUES:INST:ISUM1:ENAB? /*查询通道 1 可疑状态 SUMMARY 寄存器的使能寄存器中使能的位, 返回+9*/
```

4.10.11 :STATus:QUESTionable:INSTrument:ISUMmary[<n>][:EVENT]?

命令格式

```
:STATus:QUESTionable:INSTrument:ISUMmary[<n>][:EVENT]?
```

功能描述

查询指定通道可疑状态 SUMMARY 寄存器的事件寄存器。

参数

名称	类型	范围	默认值
<n>	离散型	{1 2 3}	-

说明

- 多通道型号有多个通道可疑状态 SUMMARY 寄存器，参数[<n>]为 1、2、3 时分别代表通道 1、通道 2、通道 3 可疑状态 SUMMARY 寄存器。若省略参数[<n>]，表示查询当前通道可疑状态 SUMMARY 寄存器的事件寄存器。
- 执行该命令，返回一个十进制值（该值与此寄存器中所有位的二进制加权和相对应），同时清除此寄存器的状态。
- 通道可疑状态 SUMMARY 寄存器的事件寄存器中的位被锁存，读取该寄存器将清除该寄存器。您也可以使用命令 *CLS 清除此寄存器。

返回格式

返回一个十进制值，该值与指定通道可疑状态 SUMMARY 寄存器的事件寄存器中所有位的二进制加权和相对应，如+1。

举例

```
:STAT:QUES:INST:ISUM1? /* 查询通道 1 可疑仪器 SUMMARY 寄存器的事件寄存器，返回+1 */
```

4.11 :SYSTem 命令子系统

:SYSTem 命令用于对电源进行系统设置、输出设置和接口设置等操作。

4.11.1 :SYSTem:BEEPer:IMMediate

命令格式

```
:SYSTem:BEEPer:IMMediate
```

功能描述

发送该命令，蜂鸣器立即蜂鸣一次。

参数

无。

说明

无。

返回格式

无。

举例

无。

4.11.2 :SYSTem:BEEPer[:STATe]

命令格式

```
:SYSTem:BEEPer[:STATe] <bool>
```

```
:SYSTem:BEEPer[:STATe]?
```

功能描述

设置或查询蜂鸣器是否开启。

参数

名称	类型	范围	默认值
<bool>	布尔型	{ON OFF 1 0}	OFF 0

说明

蜂鸣器打开时，当触摸屏幕、前面板操作或远程操作产生错误时，仪器会发出提示声音。

返回格式

返回 0 或 1。

举例

```
:SYST:BEEP ON /*打开蜂鸣器*/
:SYST:BEEP? /*查询蜂鸣器的开关状态，返回 1*/
```

4.11.3 :SYSTem:BRIGhtness

命令格式

```
:SYSTem:BRIGhtness {<brightness>|<lim>}
```

```
:SYSTem:BRIGhtness? [<lim>]
```

功能描述

设置或查询仪器 LCD 的显示亮度。

参数

名称	类型	范围	默认值
<brightness>	整型	1 至 100	50 (出厂值)
<lim>	离散型	{MINimum MAXimum}	-

说明

MINimum 和 MAXimum 表示可以设置的最小或最大亮度值。

返回格式

返回一个整数，如 60。

举例

```
:SYST:BRIG 60 /*设置屏幕的亮度为 60%*/
:SYST:BRIG? /*查询屏幕的亮度，返回 60*/
```

4.11.4 :SYSTem:COMMunicate:LAN

4.11.4.1 :SYSTem:COMMunicate:LAN:APPLy

命令格式

```
:SYSTem:COMMunicate:LAN:APPLy
```

功能描述

应用当前设置的网络参数。

参数

无。

说明

设置完 LAN 参数后，必须执行此条命令，新的设置才会生效。

返回格式

无。

举例

```
:SYST:COMM:LAN:APPL /*应用当前设置的网络参数*/
```

4.11.4.2 :SYSTem:COMMunicate:LAN:AUTOip[:STATe]

命令格式

```
:SYSTem:COMMunicate:LAN:AUTOip[:STATe] <bool>
```

```
:SYSTem:COMMunicate:LAN:AUTOip[:STATe]?
```

功能描述

设置或查询 Auto IP（自动 IP）配置模式的状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	ON 1

说明

- 使用 LAN 接口前，请使用网线将仪器连接至计算机或计算机所在的网络。
- 仪器提供 3 种 IP 配置模式：DHCP（动态配置）、Auto IP（自动配置）和 Manual IP（手动配置）。
- 自动 IP 配置模式下，仪器根据当前网络配置自动获取从 169.254.0.1 到 169.254.255.254 的 IP 地址和子网掩码 255.255.0.0。

- 三种 IP 配置模式均设为“打开”时，参数配置的优先级从高到低依次为“动态配置”、“自动设置”、“手动设置”，因此，欲启用自动配置模式，需将“动态配置 (`:SYSTem:COMMunicate:LAN:DHCP[:STATe]`)”设为“关闭”。
- 三种 IP 配置模式不能同时设为“关闭”。
- 发送该命令后，必须执行 `:SYSTem:COMMunicate:LAN:APPLy` 命令应用当前设置的网络参数，新的设置才会生效。

返回格式

返回 1 或 0。

举例

```
:SYST:COMM:LAN:AUTO ON /*打开自动 IP 配置模式*/
:SYST:COMM:LAN:AUTO? /*查询自动 IP 配置模式的状态，返回 1*/
```

4.11.4.3 :SYSTem:COMMunicate:LAN:DHCP[:STATe]

命令格式

```
:SYSTem:COMMunicate:LAN:DHCP[:STATe] <bool>
:SYSTem:COMMunicate:LAN:DHCP[:STATe]?
```

功能描述

设置或查询 DHCP（动态配置）模式是否开启。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	ON 1

说明

- DHCP 模式下，由当前网络中的 DHCP 服务器向仪器分配 IP 地址等网络参数。
- 三种 IP 配置模式均设为“打开”时，参数配置的优先级从高到低依次为“动态配置”、“自动设置”、“手动设置”。
- 三种 IP 配置模式不能同时设为“关闭”。
- 发送该命令后，需要执行 `:SYSTem:COMMunicate:LAN:APPLy` 命令，新设置的状态才会生效。

返回格式

查询返回 1 或 0。

举例

```
:SYST:COMM:LAN:DHCP ON /*打开 DHCP 模式*/
:SYST:COMM:LAN:DHCP? /*查询 DHCP 模式的状态, 返回 1*/
```

4.11.4.4 :SYSTem:COMMunicate:LAN:DNS

命令格式

```
:SYSTem:COMMunicate:LAN:DNS <dns>
```

```
:SYSTem:COMMunicate:LAN:DNS?
```

功能描述

设置或查询 DNS (Domain Name Service, 域名服务器) 地址。

参数

名称	类型	范围	默认值
<dns>	ASCII 字符串	见下文 说明	-

说明

- 该命令仅当手动 IP 配置模式打开(`:SYSTem:COMMunicate:LAN:MANualip[:STATe]`)时有效。
- 参数<dns>格式为 nnn.nnn.nnn.nnn, 第一个 nnn 的范围为 1 至 223 (127 除外), 其他三个 nnn 的范围为 0 至 255。
- 建议向您的网络管理员咨询一个可用的地址。
- 发送该命令后, 必须执行 `:SYSTem:COMMunicate:LAN:APPLy` 命令应用当前设置的网络参数, 新的设置才会生效。

返回格式

返回 DNS 地址, 如 172.16.3.2。

举例

```
:SYST:COMM:LAN:DNS 172.16.3.2 /*设置 DNS 地址为 172.16.3.2*/
:SYST:COMM:LAN:DNS? /*查询当前的 DNS 地址, 返回 172.16.3.2*/
```

4.11.4.5 :SYSTem:COMMunicate:LAN:IPADdress

命令格式

```
:SYSTem:COMMunicate:LAN:IPADdress <ip>
```

```
:SYSTem:COMMunicate:LAN:IPADdress?
```

功能描述

设置或查询 IP 地址。

参数

名称	类型	范围	默认值
<ip>	ASCII 字符串	见下文 <i>说明</i>	-

说明

- 该命令仅当手动 IP 配置模式打开(*:SYSTem:COMMunicate:LAN:MANualip[:STATe]*) 时有效。
- 参数<ip>的格式为 nnn.nnn.nnn.nnn，第一个 nnn 的范围为 1 至 223 (127 除外)，其他三个 nnn 的范围为 0 至 255。
- 建议向您的网络管理员咨询一个可用的地址。
- 发送该命令后，必须执行 *:SYSTem:COMMunicate:LAN:APPLy* 命令应用当前设置的网络参数，新的设置才会生效。

返回格式

返回 IP 地址，如 172.16.3.128。

举例

```
:SYST:COMM:LAN:IPAD 172.16.3.128 /*设置 IP 地址*/
:SYST:COMM:LAN:IPAD? /*查询当前的 IP 地址，返回 172.16.3.128*/
```

4.11.4.6 :SYSTem:COMMunicate:LAN:GATEway**命令格式**

```
:SYSTem:COMMunicate:LAN:GATEway <gateway>
```

```
:SYSTem:COMMunicate:LAN:GATEway?
```

功能描述

设置或查询默认网关。

参数

名称	类型	范围	默认值
<gateway>	ASCII 字符串	见下文 <i>说明</i>	-

说明

- 该命令仅当手动 IP 配置模式打开
(`:SYSTem:COMMunicate:LAN:MANualip[:STATe]`) 时有效。
- 参数 <gateway> 的格式为 nnn.nnn.nnn.nnn，第一个 nnn 的范围为 1 至 223 (127 除外)，其他三个 nnn 的范围为 0 至 255。
- 建议向您的网络管理员咨询一个可用的网关地址。
- 发送该命令后，必须执行 `:SYSTem:COMMunicate:LAN:APPLY` 命令应用当前设置的网络参数，新的设置才会生效。

返回格式

返回默认网关，如 172.16.3.1。

举例

```
:SYST:COMM:LAN:GATE 172.16.3.1 /*设置默认网关*/  
:SYST:COMM:LAN:GATE? /*查询当前的默认网关，返回 172.16.3.1*/
```

4.11.4.7 :SYSTem:COMMunicate:LAN:MAC?

命令格式

```
:SYSTem:COMMunicate:LAN:MAC?
```

功能描述

查询 MAC 地址。

参数

无。

说明

MAC(Media Access Control)地址，也称为硬件地址，用于定义网络设备的位置。对于一台电源，MAC 地址总是唯一的。为仪器分配 IP 地址时，总是通过 MAC 地址来识别仪器。MAC 地址 (48 位，即 6 个字节) 通常以十六进制表示，如：00-2A-A0-AA-E0-56。

返回格式

返回 MAC 地址，如 00-2A-A0-AA-E0-56。

举例

无。

4.11.4.8 :SYSTem:COMMunicate:LAN:MANualip[:STATe]

命令格式

```
:SYSTem:COMMunicate:LAN:MANualip[:STATe] <bool>
:SYSTem:COMMunicate:LAN:MANualip[:STATe]?
```

功能描述

设置或查询手动 IP 配置模式的状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	0

说明

- 手动 IP 配置模式下，由用户自定义 IP 地址等网络参数。
- 三种 IP 配置模式均设为“打开”时，参数配置的优先级从高到低依次为“动态配置”、“自动设置”、“手动设置”，因此，欲启用手动配置模式，需将“动态配置 (:SYSTem:COMMunicate:LAN:DHCP[:STATe])”和“自动设置 (:SYSTem:COMMunicate:LAN:AUTOip[:STATe])”设为“关闭”。
- 三种 IP 配置模式不能同时设为“关闭”。
- 发送该命令后，必须执行 :SYSTem:COMMunicate:LAN:APPLY 命令应用当前设置的网络参数，新的设置才会生效。

返回格式

返回 1 或 0。

举例

```
:SYST:COMM:LAN:MAN ON /*打开手动 IP 配置模式*/
:SYST:COMM:LAN:MAN? /*查询手动 IP 配置模式的状态，返回 1*/
```

4.11.4.9 :SYSTem:COMMunicate:LAN:SMASK

命令格式

```
:SYSTem:COMMunicate:LAN:SMASK <submask>
:SYSTem:COMMunicate:LAN:SMASK?
```

功能描述

设置或查询子网掩码。

参数

名称	类型	范围	默认值
<submask>	ASCII 字符串	见下文 <i>说明</i>	-

说明

- 该命令仅当手动 IP 配置模式打开 (`:SYSTem:COMMunicate:LAN:MANualip[:STATe]`) 时有效。
- 参数<submask>的格式为 nnn.nnn.nnn.nnn，其中 nnn 的范围为 0 至 255。
- 建议向您的网络管理员咨询一个可用的子网掩码。
- 发送该命令后，必须执行 `:SYSTem:COMMunicate:LAN:APPLy` 命令应用当前设置的网络参数，新的设置才会生效。

返回格式

返回子网掩码，如 255.255.255.0。

举例

```
:SYST:COMM:LAN:SMAS 255.255.255.0 /*设置子网掩码*/
:SYST:COMM:LAN:SMAS? /*查询当前的子网掩码，返回 255.255.255.0*/
```

4.11.5 :SYSTem:COMMunicate:RLSTate

命令格式

```
:SYSTem:COMMunicate:RLSTate <mode>
:SYSTem:COMMunicate:RLSTate?
```

功能描述


将电源设置为远程、本地模式或者远程锁定模式。

查询电源连接状态。

参数

名称	类型	范围	默认值
<mode>	离散型	{LOCAL REMote RWLock}	LOCAL

说明

- **LOCAL**: 本地模式。
- **REMote**: 远程模式，除通道开关按键和  按键外的前面板按键和触摸屏均可不用，且此模式下开关按键只能关闭通道输出而不能打开通道输出。

- **RWLock**: 远程锁定模式，除通道开关按键外的前面板按键和触摸屏均可不用，且此模式下开关按键只能关闭通道输出而不能打开通道输出。只能使用编程命令退出锁定模式。

返回格式

查询返回 LOCAL、REMOTE 或者 RWLock。

举例

```
:SYSTem:COMMunicate:RLState REMote /*电源设置为远程模式*/
:SYSTem:COMMunicate:RLState? /*查询电源连接状态，并返回 REMote*/
```

4.11.6 :SYSTem:COMMunicate:RS232

4.11.6.1 :SYSTem:COMMunicate:RS232:BAUD

命令格式

```
:SYSTem:COMMunicate:RS232:BAUD <baud>
:SYSTem:COMMunicate:RS232:BAUD?
```

功能描述

设置或查询 RS232 接口的波特率，单位为 Baud。

参数

名称	类型	范围	默认值
<baud>	离散型	{9600 19200 38400 57600 115200}	9600

说明

使用 RS232 接口前，请先使用 RS232 电缆将 RS232 接口与计算机或数据终端设备（DTE）连接，再设置与计算机或终端设备相匹配的接口参数（波特率、校验位等）。

返回格式

返回 RS232 接口的波特率，如 19200。

举例

```
:SYST:COMM:RS232:BAUD 19200 /*设置 RS232 接口的波特率为 19200*/
:SYST:COMM:RS232:BAUD? /*查询 RS232 接口的波特率，返回 19200*/
```

4.11.6.2 :SYSTem:COMMunicate:RS232:DBIT

命令格式

```
:SYSTem:COMMunicate:RS232:DBIT <databit>
```

:SYSTem:COMMunicate:RS232:DBIT?

功能描述

设置或查询 RS232 接口的数据位数。

参数

名称	类型	范围	默认值
<databit>	离散型	{7 8}	8

说明

无。

返回格式

返回 7 或 8。

举例

```
:SYST:COMM:RS232:DBIT 8 /*设置 RS232 接口的数据位数为 8*/
:SYST:COMM:RS232:DBIT? /*查询 RS232 接口的数据位数, 返回 8*/
```

4.11.6.3 :SYSTem:COMMunicate:RS232:PBIT

命令格式

:SYSTem:COMMunicate:RS232:PBIT <bit>

:SYSTem:COMMunicate:RS232:PBIT?

功能描述

设置或查询 RS232 校验方式。

参数

名称	类型	范围	默认值
<bit>	离散型	{NONE ODD EVEN}	NONE

说明

选择参数 NONE、ODD 或 EVEN 分别表示设置校验方式为“无”、“奇校验”或“偶校验”。

返回格式

返回 NONE、ODD 或 EVEN。

举例

```
:SYST:COMM:RS232:PBIT ODD /*设置校验方式为奇校验*/
:SYST:COMM:RS232:PBIT? /*查询当前的校验方式, 返回 ODD*/
```

4.11.6.4 :SYSTem:COMMunicate:RS232:SBIT

命令格式

```
:SYSTem:COMMunicate:RS232:SBIT <n>
```

```
:SYSTem:COMMunicate:RS232:SBIT?
```

功能描述

设置或查询 RS232 停止位数。

参数

名称	类型	范围	默认值
<n>	离散型	{1 2}	1

说明

无。

返回格式

返回 1 或 2。

举例

```
:SYST:COMM:RS232:SBIT 2 /*设置停止位数为 2*/
:SYST:COMM:RS232:SBIT? /*查询当前的停止位数，返回 2*/
```

4.11.7 :SYSTem:ERRor[:NEXT]?

命令格式

```
:SYSTem:ERRor[:NEXT]?
```

功能描述

查询并清除错误队列中的错误消息。

参数

无。

说明

每检测到一个错误，电源将会蜂鸣一声。错误队列最多存储 20 个错误，如果超过 20，队列中的最后一个将被替换为“-350”，即“队列溢出”。在错误从队伍中删除之前，将不会存储其他错误。

当查询错误队列中的错误消息时，按照先进先出的顺序检索错误并清除。

当电源关闭或发送命令 **CLS* 后，错误队列将被清除，但 **RST* 命令不能清除错误队列。

返回格式

返回错误消息的编号和内容，如-113,"Undefined header; keyword cannot be found"。如果没有错误将返回 0, "No error"。

举例

无

4.11.8 :SYSTem:KLOCK:STATe

命令格式

```
:SYSTem:KLOCK:STATe <bool>
```

```
:SYSTem:KLOCK:STATe?
```




功能描述

设置或查询远程锁定状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	OFF

说明

- 仪器处于远程锁定状态时，除各通道对应的输出开关 、、电源开关  之外，其他前面板按键不可用。且此时输出开关键只能关闭通道输出，不能打开通道输出。
- 您也可以发送 `:SYSTem:RWLock` 命令打开或关闭远程锁定。

返回格式

返回 1 或 0。

举例

```
:SYST:KLOC:STAT ON /*打开远程锁定*/
:SYST:KLOC:STAT? /*查询远程锁定状态，返回 1*/
```

4.11.9 :SYSTem:LANGUage:TYPE

命令格式

```
:SYSTem:LANGUage:TYPE <type>
```

```
:SYSTem:LANGUage:TYPE?
```

功能描述

设置或查询系统语言类型。

参数

名称	类型	范围	默认值
<type>	离散型	{EN CH DE ES FR}	-

说明

无。

返回格式

返回 ENGLISH、CHINESE、GERMAN、SPANISH 或 FRENCH。

举例

```
:SYST:LANG:TYPE EN /*设置系统语言为英文*/
:SYST:LANG:TYPE? /*查询系统语言的类型，返回 ENGLISH*/
```

4.11.10 :SYSTem:LOCal**命令格式**

:SYSTem:LOCal





功能描述

使电源从远程模式返回到本地模式。

参数

无。

说明

- 仪器处于远程模式时，除各通道对应的输出开关 、、电源开关  和  之外，其他前面板按键不可用。且开关只能用于关闭通道输出，不能打开通道。执行该命令使电源从远程模式返回到本地模式，此时，可以实现所有前面板操作。
- 您可以发送 *:SYSTem:REMOte* 命令使电源从本地模式返回到远程模式。

返回格式

无。

举例

无。

4.11.11 :SYSTem:POWEron

命令格式

```
:SYSTem:POWEron <poweron>
```

```
:SYSTem:POWEron?
```

功能描述

设置或查询系统的上电模式。

参数

名称	类型	范围	默认值
<poweron>	离散型	{DEFault LAST}	DEFault

说明

- 选择“DEFault”参数，重新开机上电后，系统恢复为出厂默认设置。
- 选择“LAST”参数，重新开机上电后，系统恢复为上次断电前的设置。

返回格式

返回 DEF 或 LAST。

举例

```
:SYST:POWE LAST /*设置系统上电时使用上次断电前的设置*/
:SYST:POWE? /*查询系统上电时使用的设置，返回 LAST*/
```

4.11.12 :SYSTem:PRINt?

命令格式

```
:SYSTem:PRINt?
```

功能描述

查询返回当前显示图像的位图数据流。

参数

无。

说明

无。

返回格式

查询返回 bmp 格式的屏幕截图的十六进制数据流。

举例

无。

4.11.13 :SYSTem:REMOte**命令格式**

:SYSTem:REMOte





功能描述

使电源从本地模式返回到远程模式。

参数

无。

说明

- 仪器处于本地模式时，所有前面板按键允许操作。执行该命令使电源从本地模式返回到远程模式，此时，除各通道对应的输出开关 、、电源开关  和  之外，其它前面板按键不可用。
- 您可以发送 `:SYSTem:LOCal` 命令使电源从远程模式返回到本地模式。

返回格式

无。

举例

无。

4.11.14 :SYSTem:RWLock**命令格式**

:SYSTem:RWLock [:STATe] [<bool>]




功能描述

打开或关闭远程锁定。

参数

名称	类型	范围	默认值
<bool>	布尔型	{1 0 ON OFF}	0 OFF

说明

- 仪器处于远程锁定状态时，除各通道对应的输出开关 、 和  电源开关键之外，前面板其它按键不可用。且此时输出开关只能关闭通道输出，不能打开。
- 若省略参数 <bool>，表示打开远程锁定。
- 您也可以发送 `:SYSTem:KLOCK:STATe` 命令打开或关闭远程锁定。

返回格式

返回 1 或 0。

举例

```
:SYST:RWL ON /*打开远程锁定*/
```

4.11.15 :SYSTem:SAMPLing

命令格式

```
:SYSTem:SAMPLing <mode>
```

```
:SYSTem:SAMPLing?
```

功能描述

设置或查询采样模式。

参数

名称	类型	范围	默认值
<mode>	离散型	{AUTO HIGH LOW}	AUTO

说明

- DP2000 系列电源通道 1 和通道 2 可以切换选择“大电流”、“小电流”或者“自动”三种电流采样模式，默认选择“自动”。
 - **大电流模式：**仪器电流采样通道固定返回大电流采集值。
 - **小电流模式：**电流可设置上限 0.5 A，回读上限值为 11 mA，当输出超过 11 mA 限制值时，用户界面输出电流显示为“-.-.-.-”。当电源输出小于 11 mA 的电流时，仪器电流采样通道返回小电流采集值。

- **自动模式:** 当电源输出大于等于 11 mA 的电流时, 仪器电流采样通道固定返回大电流采集值; 当电源输出小于 11 mA 的电流时, 仪器电流采样通道返回小电流采集值。
- 采样模式的切换针对通道 1 和通道 2 设置, 通道 3 无小电流采样模式, 所以不可以选择采样模式。
- 机器工作在内部并联模式时, 电流采样模式固定为大电流。使用外部并联需要手动设置电流采样模式为大电流模式。

返回格式

查询返回 AUTO、HIGH 或者 LOW。

举例

```
:SYSTem:SAMPLing HIGH /*设置电流采样模式为大电流模式*/
:SYSTem:SAMPLing? /*查询采样模式, 返回 HIGH*/
```

4.11.16 :SYSTem:SAVer

命令格式

```
:SYSTem:SAVer <bool>
```

```
:SYSTem:SAVer?
```

功能描述

设置或查询屏幕保护功能的开关。

参数

名称	类型	范围	默认值
<bool>	布尔型	{1 0 ON OFF}	1

说明

无。

返回格式

返回 1 或 0。

举例

```
:SYST:SAV ON /*打开屏幕保护功能*/
:SYST:SAV? /*查询当前屏幕保护功能的状态, 返回 1*/
```

4.11.17 :SYSTem:SENSe

命令格式

```
:SYSTem:SENSe <source>,<bool>
```

```
:SYSTem:SENSe? <source>
```

功能描述

设置或查询通道的 Sense 功能的开关状态。

参数

名称	类型	范围	默认值
<source>	离散型	{CH1 CH2 CH3 ALL}	-
<bool>	布尔型	{1 0 ON OFF}	OFF

说明

在大电流输出情况下，为确保负载获得准确的电压，DP2000 系列电源提供 Sense（远地敏感）工作模式。在该模式下，测量的是负载端电压而不是电源输出端电压，因此可以自动补偿负载引线引起的压降。

返回格式

返回 1 或 0。

举例

```
:SYSTem:SENS CH1,ON /*打开通道 1 的 Sense 功能*/
:SYSTem:SENS? CH1 /*查询通道 1 的 Sense 功能的开关状态，返回 1*/
```

4.11.18 :SYSTem:SYNC[:STATe]

命令格式

```
:SYSTem:SYNC[:STATe] <bool>
```

```
:SYSTem:SYNC[:STATe]?
```

功能描述

设置或查询开关同步是否打开。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	OFF

说明

- 开关同步打开时，对于通道 1 和通道 2，修改任意一个通道的开关状态时，另一通道的开关状态将随之改变。
- 只有在通道跟踪功能 (`:OUTPut:TRACk[:STATe]`) 打开时，才能使用此命令实现开关同步。

返回格式

返回 1 或 0。

举例

```
:SYST:SYNC ON /*打开开关同步*/
:SYST:SYNC? /*查询开关同步是否已打开，返回 1*/
```

4.11.19 :SYSTem:TMODe

命令格式

```
:SYSTem:TMODe <trackmode>
```

```
:SYSTem:TMODe?
```

功能描述

设置或查询跟踪功能的状态。

参数

名称	类型	范围	默认值
<trackmode>	离散型	{SYNC INDE}	INDE

说明

此电源指定通道 (CH1 和 CH2) 支持跟踪功能，您可以根据需要选择所需的跟踪方式。此命令与 `:OUTPut:TRACk[:STATe]` 命令功能相同。

- **SYNC:** 通道跟踪打开，对于支持跟踪功能的两个通道（仅限于同一台仪器），修改任意一个通道的设置（包括电压电流设置值、过压过流设置值、过压过流开关状态），另一个通道的设置会随之改变。
- **INDE:** 通道跟踪关闭，对于支持跟踪功能的两个通道（仅限于同一台仪器），修改任意一个通道的设置，另一个通道不受影响。

返回格式

返回 SYNCHRONOUS 或 INDEPENDENT。

举例

```
:SYST:TMOD SYNC /*设置跟踪方式为打开*/
:SYST:TMOD? /*查询当前的跟踪状态, 返回 SYNCHRONOUS*/
```

4.11.20 :SYSTem:TLOCK**命令格式**

```
:SYSTem:TLOCK <bool>
```

```
:SYSTem:TLOCK?
```

功能描述

设置或查询触屏板按键是否被锁定。

参数

名称	类型	范围	默认值
<bool>	布尔型	{1 0 ON OFF}	0

说明

触屏板按键被锁定后, 触屏功能不可用。

返回格式

返回 1 或 0。

举例

```
:SYSTem:TLOCK ON /*打开触屏版按键锁定*/
:SYSTem:TLOCK? /*查询触屏版按键锁定状态, 返回 1*/
```

4.11.21 :SYSTem:VERsion?**命令格式**

```
:SYSTem:VERsion?
```

功能描述

查询当前系统的 SCPI 版本号。

参数

无。

说明

无。

返回格式

返回一个字符串（系统中 SCPI 版本号），格式为 YYYY.V，其中，YYYY 表示版本的年份，V 表示该年度的版次，如 1999.0。

举例

```
:SYST:VERS? /*查询当前系统的 SCPI 版本号，返回 1999.0*/
```

4.12 :TIMER 命令子系统

:TIMER 命令用于设置任意波形发生器的参数和打开或关闭任意波形发生器。

4.12.1 :TIMER:CYCLES

命令格式

```
:TIMER:CYCLES <cycle>[,<value>]
```

```
:TIMER:CYCLES?
```

功能描述

设置或查询任意波形发生器当前选择通道对应的循环数。

参数

名称	类型	范围	默认值
<cycle>	离散型	{N I}	N
<value>	整型	1 至 99999	1

说明

- 循环数定义为电源根据预设的电压/电流完成定时输出的循环次数。您可以设置循环数为无限 (I) 或者指定值 (N, <value>)。
- 定时输出的总组数=输出组数×循环数，其中，输出组数可由 `:TIMER:GROUPS:NUM?` 命令查询。
- 电源在完成总组数次输出后终止定时功能。此时，电源的状态由 `:TIMER:ENDState` 命令的设置决定。

返回格式

查询返回 I 或 N, <value>，如 N,20。

举例

```
:TIME:CYCLE N,20 /*设置任意波形发生器的循环数为 20*/
:TIME:CYCLE? /*查询当前任意波形发生器的循环数，返回 N,20*/
```

4.12.2 :TIMER:CHANNEL**命令格式**

```
:TIMER:CHANNEL <ch>
```

```
:TIMER:CHANNEL?
```

功能描述

设置或查询当前编辑的通道。

参数

名称	类型	范围	默认值
<ch>	离散型	{CH1 CH2 CH3}	-

说明

无。

返回格式

查询返回 CH1、CH2 或 CH3。

举例

```
:TIMER:CHANNEL CH2 /*设置当前编辑的通道为 CH2*/
:TIMER:CHANNEL? /*查询当前编辑的通道，返回 CH2*/
```

4.12.3 :TIMER:ENDState**命令格式**

```
:TIMER:ENDState <end>
```

```
:TIMER:ENDState?
```

功能描述

设置或查询任意波形发生器的终止状态。

参数

名称	类型	范围	默认值
<end>	离散型	{OFF LAST}	OFF

说明

终止状态是指，当循环数为有限的数值时，仪器完成总组数次电压/电流输出之后所处的状态。

- **输出关闭 (OFF)**：完成输出后，仪器自动关闭输出。
- **最后一组 (LAST)**：完成输出后，仪器停留在最后一组的输出状态。

定时输出的总组数=输出组数×循环数，其中，输出组数可通过 `:TIMER:GROUPs:NUM?` 命令查询，循环数由 `:TIMER:CYCLEs` 命令设置。

返回格式

查询返回 OFF 或 LAST。

举例

```
:TIME:ENDS LAST /*设置任意波形发生器的终止状态为“最后一组”*/  
:TIME:ENDS? /*查询当前任意波形发生器的终止状态，返回 LAST*/
```

4.12.4 :TIMER:GROUPs:NUM?

命令格式

```
:TIMER:GROUPs:NUM?
```

功能描述

查询当前通道波表的输出组数。

参数

无。

说明

- 输出组数定义为，电源在每个循环中输出的预设电压/电流的组数。
- 定时输出的总组数=输出组数×循环数，其中，循环数由 `:TIMER:CYCLEs` 命令设置。
- 电源在完成总组数次输出后终止定时功能。此时，电源的状态由 `:TIMER:ENDState` 命令的设置决定。

返回格式

查询返回 1 至 512 之间的整数，如 25。

举例

```
:TIME:GROUP:NUM? /*查询当前通道波表的输出组数，返回 25*/
```

4.12.5 TIMEr:GROUP:INDEX

命令格式

TIMEr:GROUP:INDEX <val>

TIMEr:GROUP:INDEX?

功能描述

设置或查询当前通道波表的操作位置。

参数

名称	类型	范围	默认值
<val>	整型	1 至 512	-

说明

当设置的操作位置行无数据时，操作位置自动跳转至当前波表最后一组数据。

返回格式

查询返回当前通道波表的操作位置，如 25。

举例

```
TIMEr:GROUP:INDEX 25 /*设置当前通道波表的操作位置为第 25 组/
TIMEr:GROUP:INDEX? /*查询返回 25*/
```

4.12.6 :TIMEr:GROUP:PARAMeter

命令格式

:TIMEr:GROUP:PARAMeter <volt>,<curr>,<time>

:TIMEr:GROUP:PARAMeter? [<groupcount>]

功能描述

在当前通道波表的当前编辑位置插入一组数据。

查询当前通道波表编辑位置的波形参数。

参数

名称	类型	范围	默认值
<volt>	实型	当前通道的电压范围	-
<curr>	实型	当前通道的电流范围	-
<time>	实型	最大 3600 s	-

名称	类型	范围	默认值
<groupcount>	整型	1 至 512	1

说明

- <volt>、<curr>和<time>分别为该组参数中的电压、电流和时间，单位分别为 V、A、s。
- 从 `TIMER:GROUP:INDEX` 设置的位置开始查询<groupcount>个数据，若数据不足，则停止输出。

返回格式

查询返回一个以#开始的字符串。

例如：#90000000431,0.500,1.0000,1.000;2,5.500,2.0000,1.000;; 其中，#9000000043 为数据块头；1,0.500,1.0000,1.000;2,5.500,2.0000,1.000;为具体的任意波参数。

- 数据块头用于描述数据流长度信息，格式为#NX...X。例如：#9000000043，其中 N 为 9，表示其后跟随的 9 个数据用于描述数据流的长度信息。其后的 000000043 用于描述数据流长度信息（43 个字节）。
- 每组定时参数的格式均为“序号,电压,电流,定时时间”，多组参数之间以分号“;”隔开。例如：1,0.500,1.0000,1.000;2,5.500,2.0000,1.000;表示共两组定时参数；第 1 组定时参数的序号为 1，电压值为 0.5V，电流值为 1A，定时时间为 1s；第 2 组定时参数的序号为 2，电压值为 5.5V，电流值为 2A，定时时间为 1s。

举例

```
:TIMER:GROUP:INDEX 1 /*设置当前通道波表的操作位置为第 1 组*/
:TIME:GROUP:PARA 0.5,1,1 /*设置当前通道波表的当前编辑位置波形参数为 0.5V,
1A, 1s*/
:TIMER:GROUP:INDEX 2 /*设置当前通道波表的操作位置为第 2 组*/
:TIME:GROUP:PARA 5.5,2,1 /*设置当前通道波表的当前编辑位置波形参数为 5.5V,
2A, 1s*/
:TIMER:GROUP:INDEX 1 /*设置当前通道波表的操作位置为第 1 组*/
:TIME:GROUP:PARA? 2 /*查询从第 1 组起的 2 组定时参数，返回
#90000000431,0.500,1.0000,1.000;2,5.500,2.0000,1.000*/
```

4.12.7 :TIMER:GROUP:DELeTe

命令格式

```
:TIMER:GROUP:DELeTe [<groupcount>]
```

功能描述

从当前通道波表的编辑位置向下删除波表数据。

参数

名称	类型	范围	默认值
<groupcount>	整型	1 至 512	1

说明

<groupcount> 为删除的组数，当不填时默认为 1 组。

返回格式

无。

举例

```
:TIMER:GROUP:INDEX 25 /*设置当前通道波表的操作位置为第 25 组*/
:TIMER:GROUP:DELETE 2 /*从当前通道波表删除第 25、26 组波形参数*/
```

4.12.8 :TIMER:RUN**命令格式**

```
:TIMER:RUN <run>
```

```
:TIMER:RUN?
```

功能描述

设置或查询任意波形发生器的运行方式。

参数

名称	类型	范围	默认值
<run>	离散型	{CONTINUE SINGLE}	CONTINUE

说明

- **CONTINUE:** 连续运行，每次打开任意波输出，仪器将按照当前设置的组数和循环数连续输出。
- **SINGLE:** 单次运行，每次打开任意波输出，仪器将按照当前构建的波表顺序输出一组数据。

返回格式

查询返回 CONTINUE 或 SINGLE。

举例

```
:TIMER:RUN SINGLE /*设置任意波形发生器的运行方式为单次*/
:TIMER:RUN? /*查询任意波形发生器的运行方式，返回 SINGLE*/
```

4.12.9 :TIMER[:STATe]

命令格式

```
:TIMER[:STATe] <bool>
```

```
:TIMER[:STATe]?
```

功能描述

设置或查询任意波形发生器开关状态。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	0 OFF

说明

- 打开任意波形发生器会改变通道输出状态，打开前请确认输出状态改变不会对与电源相连接的设备造成影响。
- 打开任意波形发生器且打开对应通道时，任意波输出才生效。
- 若是任意波形发生器的运行方式 (:TIMER:RUN) 为连续运行，打开通道输出和任意波发生器时 (:TIME ON)，任意波形发生器按照当前设置的组数和循环数连续输出；如果此时任意波发生器的触发方式 (:TIMER:TRIG) 为总线触发，还需发送命令 *TRG 才能触发输出。
- 若是任意波形发生器的运行方式 (:TIMER:RUN) 为单次运行，打开通道输出和任意波发生器 (:TIME ON) 后，每次发送:TIME ON，仪器都会按照当前波表顺序输出一组数据；如果此时任意波发生器的触发方式 (:TIMER:TRIG) 为总线触发，打开通道输出和任意波发生器后，每次发送命令 *TRG 会输出一组数据。
- 打开任意波形发生器期间，不可修改参数。

返回格式

查询返回 1 或 0。

举例

```
:TIME ON /*打开任意波形输出*/
:TIME? /*查询任意波形输出的状态，返回 1*/
```

4.12.10 :TIMER:TEMPlet:CONSTRUCT

命令格式

```
:TIMER:TEMPlet:CONSTRUCT
```


功能描述

发送该命令，仪器根据当前所选的模板和设置的参数构建任意波参数。

参数

无。

说明

无。

返回格式

无。

举例

```
:TIMer:TEMPlet:CONSTRUCT /*仪器根据当前所选的模板和设置的参数构建任意波参数*/
```

4.12.11 :TIMer:TEMPlet:FALLRate

命令格式

```
:TIMer:TEMPlet:FALLRate <value>
```

```
:TIMer:TEMPlet:FALLRate?
```

功能描述

设置或查询 ExpFall 的下降指数。

参数

名称	类型	范围	默认值
<value>	整型	0 至 10	0

说明

当前所选模板为 ExpFall (*:TIMer:TEMPlet:SElect*) 时，由于指数函数自身的特点，构建的定时参数将不能达到最小值。构建的定时参数可达到的范围与当前设置的下降指数有关，下降指数越大，定时参数可达到的范围越大。

返回格式

查询返回 0 至 10 之间的整数，如 5。

举例

```
:TIME:TEMP:FALLR 5 /*设置 ExpFall 的下降指数为 5*/
:TIME:TEMP:FALLR? /*查询当前设置的 ExpFall 的下降指数，返回 5*/
```

4.12.12 :TIMER:TEMPlet:INTERval

命令格式

```
:TIMER:TEMPlet:INTERval <value>
```

```
:TIMER:TEMPlet:INTERval?
```

功能描述

设置或查询时间间隔。

参数

名称	类型	范围	默认值
<value>	实型	0.001s 至 3600s	1s

说明

- 时间间隔是指输出由当前所选模板构建的每一组定时参数时所持续的时间。
- Stair Up、Stair Dn、Stair UpDn 和 Pulse 模板类型不支持此参数。

返回格式

查询返回范围之间的实数，如 15.000。

举例

```
:TIME:TEMP:INTE 15 /*设置时间间隔为 15 s*/
:TIME:TEMP:INTE? /*查询当前设置的时间间隔，返回 15.000*/
```

4.12.13 :TIMER:TEMPlet:INVErt

命令格式

```
:TIMER:TEMPlet:INVErt <bool>
```

```
:TIMER:TEMPlet:INVErt?
```

功能描述

设置或查询当前所选模板的反相功能是否打开。

参数

名称	类型	范围	默认值
<bool>	布尔型	{0 1 ON OFF}	OFF

说明

- 打开反相功能时，仪器先将已确定的波形翻转，然后构建任意波参数。

- 仅 Sine、Pulse 和 Ramp 模板类型支持反相功能。

返回格式

查询返回 1 或 0。

举例

```
:TIME:TEMP:INVE ON /*打开当前所选模板的反相功能*/
:TIME:TEMP:INVE? /*查询当前所选模板是否打开反相, 返回 1*/
```

4.12.14 :TIMER:TEMPlet:MAXValue

命令格式

```
:TIMER:TEMPlet:MAXValue {<value>|MINimum|MAXimum}
:TIMER:TEMPlet:MAXValue? [MINimum|MAXimum]
```

功能描述

设置或查询当前所选模板的最大电压或电流值。

参数

名称	类型	范围	默认值
<value>	实型	当前所选通道的电压范围或电流范围	1V/1A

说明

- 当编辑对象 (:TIMER:TEMPlet:OBJECT) 为电压时, 此命令设置的是最大电压值; 当编辑对象为电流时, 此命令设置的是最大电流值。
- 当模板类型为 Pulse 时, 该命令用于设置或查询高电平值。
- “MINimum”、“MAXimum” 分别代表最大电流/电压允许设置的最大值和最小值。

返回格式

返回当前所选模板的最大电压值或最大电流值, 如 5.000 或 5.3000。

举例

```
:TIME:TEMP:OBJ V,2 /*设置当前编辑对象为电压, 并设置电流值为 2A*/
:TIME:TEMP:MAXV 5 /*设置当前所选模板的最大电压为 5V*/
:TIME:TEMP:MAXV? /*查询当前所选模板的最大电压, 返回 5.000*/
```

4.12.15 :TIMER:TEMPlet:MINValue

命令格式

```
:TIMER:TEMPlet:MINValue {<value>|MINimum|MAXimum}
:TIMER:TEMPlet:MINValue? [MINimum|MAXimum]
```

功能描述

设置或查询当前所选模板的最小电压或电流值。

参数

名称	类型	范围	默认值
<value>	实型	当前所选通道的电压范围或电流范围	0

说明

- 当编辑对象 (*:TIMER:TEMPlet:OBJECT*) 为电压时, 此命令设置的是最小电压值; 当编辑对象为电流时, 此命令设置的是最小电流值。
- 当模板类型为 Pulse 时, 该命令用于设置或查询低电平值。
- “MINimum”、“MAXimum” 分别代表最小电流/电压允许设置的最大值和最小值。

返回格式

查询返回当前所选模板的最小电压值或最小电流值, 如 0.200 或 0.1000。

举例

```
:TIME:TEMP:OBJ C,1.5 /*设置当前编辑对象为电流, 并设置电压值为 1.5V*/
:TIME:TEMP:MINV 0.1 /*设置当前所选模板的最小电流为 0.1A*/
:TIME:TEMP:MINV? /*查询当前所选模板的最小电流, 返回 0.1000*/
```

4.12.16 :TIMER:TEMPlet:OBJECT

命令格式

```
:TIMER:TEMPlet:OBJECT <obj> [, <value> | MINimum | MAXimum]
```

```
:TIMER:TEMPlet:OBJECT? [MINimum | MAXimum]
```

功能描述

设置或查询当前所选的模板编辑的对象及其对应的电流或电压值。

参数

名称	类型	范围	默认值
<obj>	离散型	{V C}	V
<value>	实型	当前所选通道的电压范围或电流范围	0

说明

- <obj> 选择 V 即选择编辑对象为电压，<value> 用于设置恒定电流值，范围为当前所选通道的电流范围。此时您可发送 `:TIMEr:TEMPlet:MAXValue` 和 `:TIMEr:TEMPlet:MINValue` 命令编辑电压的最大、最小值。
- <obj> 选择 C 即选择编辑对象为电流，<value> 用于设置恒定电压值，范围为当前所选通道的电压范围。此时您可发送 `:TIMEr:TEMPlet:MAXValue` 和 `:TIMEr:TEMPlet:MINValue` 命令编辑电流的最大、最小值。

返回格式

查询返回当前所选的编辑对象及相应的电流或电压值，两者之间以逗号隔开，如 V,2.0000。其中，V 表示当前所选的编辑对象为电压，2.0000 表示设置的固定电流值为 2A。

举例

```
:TIME:TEMP:OBJ V,2 /*选择模板编辑的对象为电压，并设置恒定电流值为 2A*/
:TIME:TEMP:OBJ? /*查询当前所选的编辑对象及其对应的恒定电流或电压值，返回
V,2.0000*/
```

4.12.17 :TIMEr:TEMPlet:PERIOD

命令格式

```
:TIMEr:TEMPlet:PERIOD <value>
```

```
:TIMEr:TEMPlet:PERIOD?
```

功能描述

设置或查询波形的周期。

参数

名称	类型	范围	默认值
<value>	实型	0.001s 至 3600s	见下文 说明

说明

- <value> 为一个波形周期持续的时间，除 ExpRise 和 ExpFall，其他所有波形都需要设置周期。
- Pulse 波形默认周期为 2s，其他需要设置周期时间的波形默认周期为 50s。
- 当前所选模板一个周期的总点数由所设波形周期时间和时间间隔 (`:TIMEr:TEMPlet:INTERVAL`) 决定，周期点数=波形周期/时间间隔。

返回格式

返回 0.001 至 3600 之间的实数，如 15.000。

举例

```
:TIME:TEMP:PERI 15 /*设置当前选择模板类型的周期为 15s*/
:TIME:TEMP:PERI? /*查询当前选择模板类型的周期, 返回 15.000*/
```

4.12.18 :TIMER:TEMPlet:POINTS**命令格式**

```
:TIMER:TEMPlet:POINTS <value>
```

```
:TIMER:TEMPlet:POINTS?
```

功能描述

设置或查询总点数。

参数

名称	类型	范围	默认值
<value>	整型	1 至 512	50

说明

- 总点数是指，使用当前所选模板构建的定时参数的组数。
- 当总点数（记作 **P**）与当前的输出组数（记作 **G**，`:TIMER:GROUPs:NUM?`）不同时，由模板构建 **P** 组参数，构建后，输出组数自动变为 **P**。

返回格式

查询返回 1 至 512 之间的整数，如 80。

举例

```
:TIME:TEMP:POINT 80 /*设置总点数为 80*/
:TIME:TEMP:POINT? /*查询当前设置的总点数, 返回 80*/
```

4.12.19 :TIMER:TEMPlet:RISERate**命令格式**

```
:TIMER:TEMPlet:RISERate <value>
```

```
:TIMER:TEMPlet:RISERate?
```

功能描述

设置或查询 ExpRise 的上升指数。

参数

名称	类型	范围	默认值
<value>	整型	0 至 10	0

说明

当前所选模板为 ExpRise (`:TIMER:TEMPlet:SElect`) 时, 由于指数函数自身的特点, 构建的定时参数将不能达到最大值。构建的定时参数可达到的范围与当前设置的上升指数有关, 上升指数越大, 定时参数可达到的范围越大。

返回格式

查询返回 0 至 10 之间的整数, 如 5。

举例

```
:TIME:TEMP:RISER 5 /*设置 ExpRise 的上升指数为 5*/
:TIME:TEMP:RISER? /*查询当前设置的 ExpRise 的上升指数, 返回 5*/
```

4.12.20 :TIMER:TEMPlet:SElect

命令格式

```
:TIMER:TEMPlet:SElect <temp>
:TIMER:TEMPlet:SElect?
```

功能描述

设置或查询模板类型。

参数

名称	类型	范围	默认值
<temp>	离散型	{SINE PULSE RAMP UP DN UPDN RISE FALL}	SINE

说明

无。

返回格式

查询返回 SINE、PULSE、RAMP、UP、DN、UPDN、RISE 或 FALL。

举例

```
:TIME:TEMP:SEL UP /*选择模板类型为阶梯上升 (UP) */
:TIME:TEMP:SEL? /*查询当前所选的模板类型, 返回 UP*/
```

4.12.21 :TIMER:TEMPlet:SYMMetry

命令格式

```
:TIMER:TEMPlet:SYMMetry <value>
```

```
:TIMER:TEMPlet:SYMMetry?
```

功能描述

设置或查询 RAMP 的对称性。

参数

名称	类型	范围	默认值
<value>	整型	0 至 100	50

说明

对称性是指一个周期内上升沿所持续的时间占整个周期的百分比。

返回格式

查询返回 RAMP 的对称性，如 60%。

举例

```
:TIME:TEMP:SYMM 60 /*设置 RAMP 的对称性为 60%*/
:TIME:TEMP:SYMM? /*查询当前设置的 RAMP 的对称性，返回 60%*/
```

4.12.22 :TIMER:TEMPlet:WIDTH

命令格式

```
:TIMER:TEMPlet:WIDTH <value>
```

```
:TIMER:TEMPlet:WIDTH?
```

功能描述

设置或查询 Pulse 的正脉宽。

参数

名称	类型	范围	默认值
<value>	实型	最大 3600s	1s

说明

- 脉宽即一个周期内高电平所持续的时间。

- <value>的实际可设置范围与当前设置的周期 (*:TIMER:TEMPlet:PERIOD*) 有关, 正脉宽不能大于周期。

返回格式

查询返回一个实数, 如 14.000。

举例

```
:TIME:TEMP:WIDT 14 /*设置 Pulse 的脉宽为 14s*/
:TIME:TEMP:WIDT? /*查询当前设置的 Pulse 的脉宽, 返回 14.000*/
```

4.12.23 :TIMER:TEMPlet:STAIR

命令格式

```
:TIMER:TEMPlet:STAIR <val>
:TIMER:TEMPlet:STAIR?
```

功能描述

设置或查询 StairUp、StairDn、StairUpDn 波形最大值与最小值之间跨越的阶梯数。

参数

名称	类型	范围	默认值
<val>	整型	3 至 99999	25

说明

要想构建一个完整周期的 StairUp、StairDn 或 StairUpDn 波形, <val>取值应小于构建波形的组数 (*:TIMER:TEMPlet:POINTS*)。

返回格式

查询返回波形最大值与最小值之间跨越的阶梯数, 如 30。

举例

```
:TIME:TEMP:SEL UP /*选择模板类型为阶梯上升 (UP)*/
:TIMER:TEMPlet:STAIR 30 /*设置 StairUp 波形最大值与最小值之间跨越的阶梯数为 30*/
:TIMER:TEMPlet:STAIR? /*查询返回 30*/
```

4.12.24 :TIMER:TRIG

命令格式

```
:TIMER:TRIG <trig>
:TIMER:TRIG?
```

功能描述

设置或查询触发方式。

参数

名称	类型	范围	默认值
<trig>	离散型	{MANual BUS}	MANual

说明

触发方式是指打开任意波输出的方式，包括“手动触发”和“总线触发”。

- **手动触发 (MANual)**：选择手动触发时，打开通道输出 (`:OUTPut[:STATe]`) 和任意波形发生器 (`:TIMer[:STATe]`) 即可根据设置的运行方式 (`:TIMer:RUN`) 输出任意波。
- **总线触发 (BUS)**：选择总线触发时，打开任意波形发生器 (`:TIMer[:STATe]`) 后会等待触发信号，在通道输出打开时接收到命令 `*TRG` 即可根据设置的运行方式 (`:TIMer:RUN`) 输出任意波。

返回格式

查询返回 MANUAL 或 BUS。

举例

```
:TIMer:TRIG BUS /*设置任意波形发生器触发方式为总线触发*/
:TIMer:TRIG? /*查询任意波形发生器触发方式，返回 BUS*/
```

4.13 :TRIGger 命令子系统

`:TRIGger` 命令用于打开和关闭触发器、设置和查询触发条件、设置和查询指定数据线的受控源和控制源。

4.13.1 :TRIGger:IN[:ENABLE]

命令格式

```
:TRIGger:IN[:ENABLE] <d>,<bool>
```

```
:TRIGger:IN[:ENABLE]? <d>
```

功能描述

设置或查询指定数据线的触发输入功能的状态。

参数

名称	类型	范围	默认值
<d>	离散型	{D1 D2 D3 D4}	D1
<bool>	布尔型	{1 ON 0 OFF}	0 OFF

说明

使能触发输入功能后，当该数据线上的输入信号满足设置的触发类型（*:TRIGger:IN:TYPE*）时，指定的受控源（*:TRIGger:IN:SOURce*）按照输出响应（*:TRIGger:IN:RESPonse*）中的设置关闭输出、打开输出或翻转输出状态。

返回格式

查询返回 1 或 0。

举例

```
:TRIG:IN D1,ON /*使能 D1 的触发输入功能*/
:TRIG:IN? D1 /*查询 D1 的触发输入功能状态，返回 1*/
```

4.13.2 :TRIGger:IN:IMMEdiate

命令格式

```
:TRIGger:IN:IMMEdiate
```

功能描述

初始化触发系统。模拟硬件立即产生触发信号。

参数

无。

说明

无。

返回格式

无。

举例

无。

4.13.3 :TRIGger:IN:RESPonse

命令格式

```
:TRIGger:IN:RESPonse <d>,<res>
```

```
:TRIGger:IN:RESPonse? <d>
```

功能描述

设置或查询指定数据线触发输入的输出响应。

参数

名称	类型	范围	默认值
<d>	离散型	{D1 D2 D3 D4}	D1
<res>	离散型	{ON OFF ALTER}	OFF

说明

- **输出打开 (ON)** : 当指定数据线上的输入信号满足设置的触发类型 (*:TRIGger:IN:TYPE*) 时, 打开当前选择的受控源 (*:TRIGger:IN[:ENABLEj]*) 的通道输出。
- **输出关闭 (OFF)** : 当指定数据线上的输入信号满足设置的触发类型 (*:TRIGger:IN:TYPE*) 时, 关闭当前选择的受控源的通道输出。
- **输出翻转 (ALTER)** : 当指定数据线上的输入信号满足设置的触发类型 (*:TRIGger:IN:TYPE*) 时, 翻转当前选择的受控源的通道输出状态, 即当前通道输出状态为打开, 则关闭通道; 当前通道输出状态为关闭, 则打开通道。

返回格式

查询返回 ON、OFF 或 ALTER。

举例

```
:TRIG:IN:RESP D1,ON /*设置 D1 触发输入的输出响应为输出打开*/
:TRIG:IN:RESP? D1 /*查询 D1 触发输入的输出响应, 返回 ON*/
```

4.13.4 :TRIGger:IN:SOURce

命令格式

```
:TRIGger:IN:SOURce <d>,<ch>
```

```
:TRIGger:IN:SOURce? <d>
```

功能描述

设置或查询指定数据线触发输入的受控源。

参数

名称	类型	范围	默认值
<d>	离散型	{D1 D2 D3 D4}	D1

名称	类型	范围	默认值
<ch>	ASCII 字符型	{CH1[,CH2[,CH3]]}	CH1

说明

您可以选择 CH1、CH2 和 CH3 中的任意一个或多个通道作为受控源，若省略<ch>参数，则表示选择 CH1 为指定数据线触发输入的受控源。

返回格式

查询返回所选受控源的名称，若当前所选受控源包含多个通道，则通道之间以逗号分隔，如 CH1 或 CH1,CH2。

举例

```
:TRIG:IN:SOUR D1,CH1,CH2 /*设置 D1 触发输入的受控源为 CH1 和 CH2*/
:TRIG:IN:SOUR? D1 /*查询 D1 触发输入的受控源，返回 CH1,CH2*/
```

4.13.5 :TRIGger:IN:TYPE

命令格式

```
:TRIGger:IN:TYPE <d>,<type>
```

```
:TRIGger:IN:TYPE? <d>
```

功能描述

设置或查询指定数据线触发输入的触发类型。

参数

名称	类型	范围	默认值
<d>	离散型	{D1 D2 D3 D4}	D1
<type>	离散型	{RISE FALL HIGH LOW}	RISE

说明

- 您可以选择在输入信号的上升沿 (RISE)、下降沿 (FALL)、高电平 (HIGH) 或低电平 (LOW) 处触发。
- 对于输入信号，高电平: 2.5V~3.3V; 低电平: 0V~0.8V; 噪声容限: 0.4V。

返回格式

查询返回 RISE、FALL、HIGH 或 LOW。

举例

```
:TRIG:IN:TYPE D1,FALL /*设置 D1 触发输入的触发类型为下降沿*/
:TRIG:IN:TYPE? D1 /*查询 D1 触发输入的触发类型，返回 FALL*/
```

4.13.6 :TRIGger:OUT:POLARity

命令格式

```
:TRIGger:OUT:POLARity <d>,<pol>
```

```
:TRIGger:OUT:POLARity? <d>
```

功能描述

设置或查询指定数据线触发输出的输出响应。

参数

名称	类型	范围	默认值
<d>	离散型	{D1 D2 D3 D4}	D1
<pol>	离散型	{POSitive NEGative}	POSitive

说明

- **高电平 (POSitive)**：当打开控制源输出时，对应数据线输出 3.3 V 高电平。
- **低电平 (NEGative)**：当打开控制源输出时，对应数据线输出低电平（CMOS 电平）。

返回格式

查询返回 POSITIVE 或 NEGATIVE。

举例

```
:TRIG:OUT:POLA D1,NEGative /*设置 D1 的触发输出的输出信号为低电平*/
:TRIG:OUT:POLA? D1 /*查询 D1 的触发输出的输出信号电平，返回 NEGATIVE*/
```

4.13.7 :TRIGger:OUT:SOURce

命令格式

```
:TRIGger:OUT:SOURce <d>,<source>
```

```
:TRIGger:OUT:SOURce? <d>
```

功能描述

设置或查询指定数据线触发输出的控制源。

参数

名称	类型	范围	默认值
<d>	离散型	{D1 D2 D3 D4}	D1

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

说明

您可以选择 CH1、CH2 或 CH3 中的任意一个通道作为触发输出的控制源。

返回格式

查询返回所选控制源的名称，如 CH1。

举例

```
:TRIG:OUT:SOUR D1,CH1 /*设置 D1 触发输出的控制源为 CH1*/
:TRIG:OUT:SOUR? D1 /*查询 D1 触发输出的控制源，返回 CH1*/
```

4.13.8 :TRIGger:OUT[:ENABLE]

命令格式

```
:TRIGger:OUT[:ENABLE] <d>,<bool>
```

```
:TRIGger:OUT[:ENABLE]? <d>
```

功能描述

设置或查询指定数据线的触发输出功能。

参数

名称	类型	范围	默认值
<d>	离散型	{D1 D2 D3 D4}	D1
<bool>	布尔型	{0 1 ON OFF}	0 OFF

说明

启用触发输出功能后，当指定的控制源（*:TRIGger:OUT:SOURce*）的输出打开时，指定的数据线将按照输出设置（*:TRIGger:OUT:POLArity*）输出高/低电平信号。

返回格式

查询返回 0 或 1。

举例

```
:TRIG:OUT D1,ON /*启用 D1 的触发输出功能*/
:TRIG:OUT? D1 /*查询 D1 的触发输出功能状态，返回 1*/
```

5 编程实例

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

5.1 编程准备

编程之前，您需要做如下准备工作：

您可以登陆 RIGOL 官网 (<http://www.rigol.com>) 下载该软件，然后按照指导进行安装。安装 Ultra Sigma 后，NI-VISA 库已自动安装完成。本文中默认安装路径为 C:\Program Files\IVI Foundation\VISA。

本文应用仪器的 USB 接口与 PC 通信。请使用 USB 数据线将后面板的 USB Device 接口与 PC 相连。仪器与 PC 正确连接后，接通仪器电源并开机。

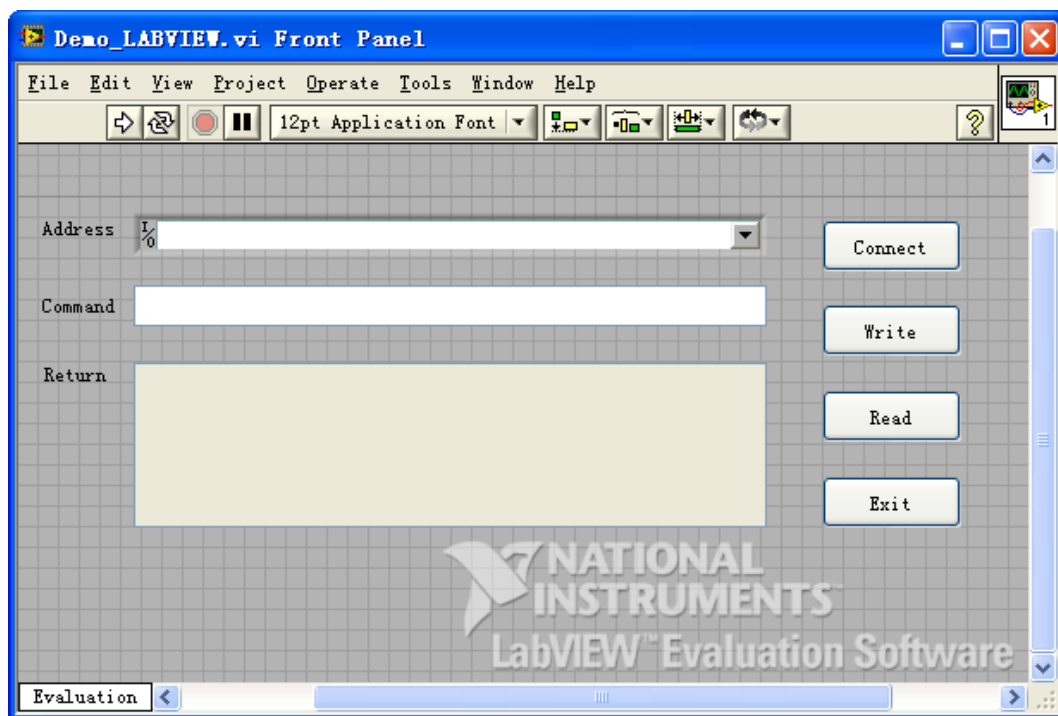
下面将详细介绍在 LabVIEW、Visual Basic 和 Visual C++ 开发环境中的编程实例。

5.2 LabVIEW 编程实例

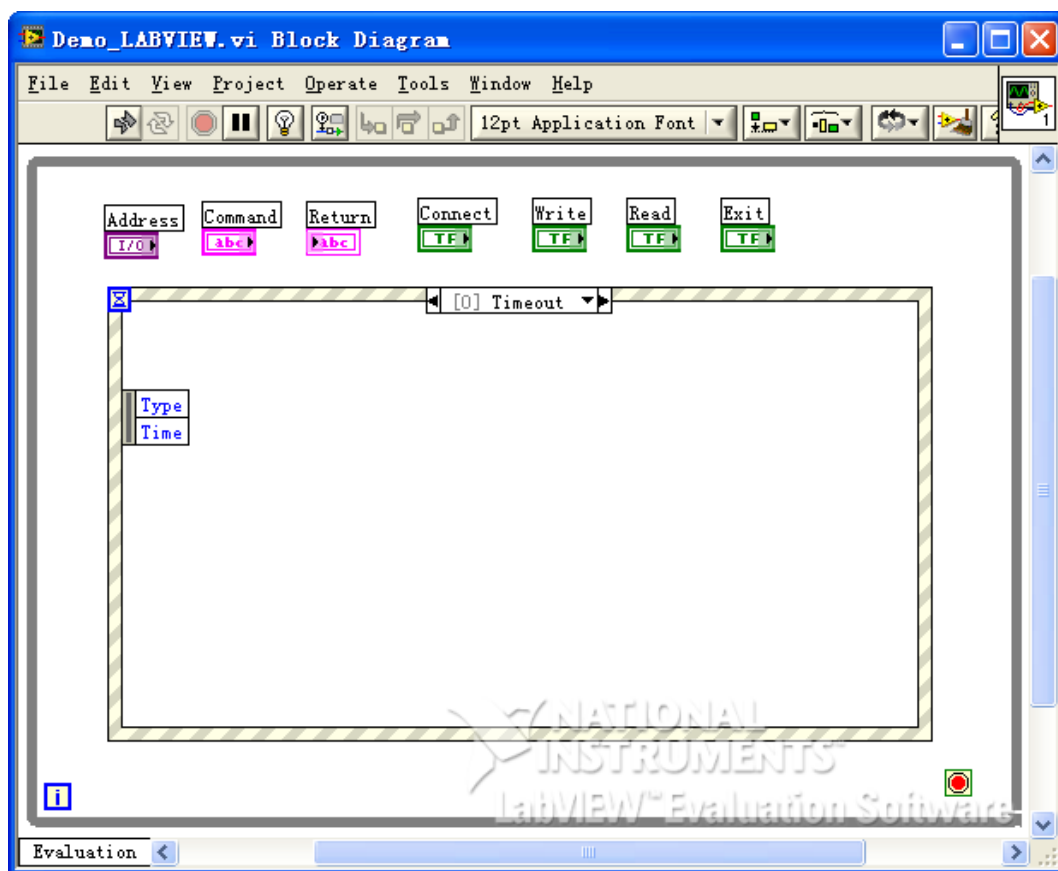
本例使用的程序： LabVIEW 2009

本例实现的功能： 查找仪器地址、连接仪器、发送命令并读取返回值。

1. 运行 LabVIEW，新建一个 VI 文件，命名为 Demo_LABVIEW。
2. 在前面板界面添加控件，包括地址栏、命令栏和返回值栏以及连接、写入、读取和退出按钮。

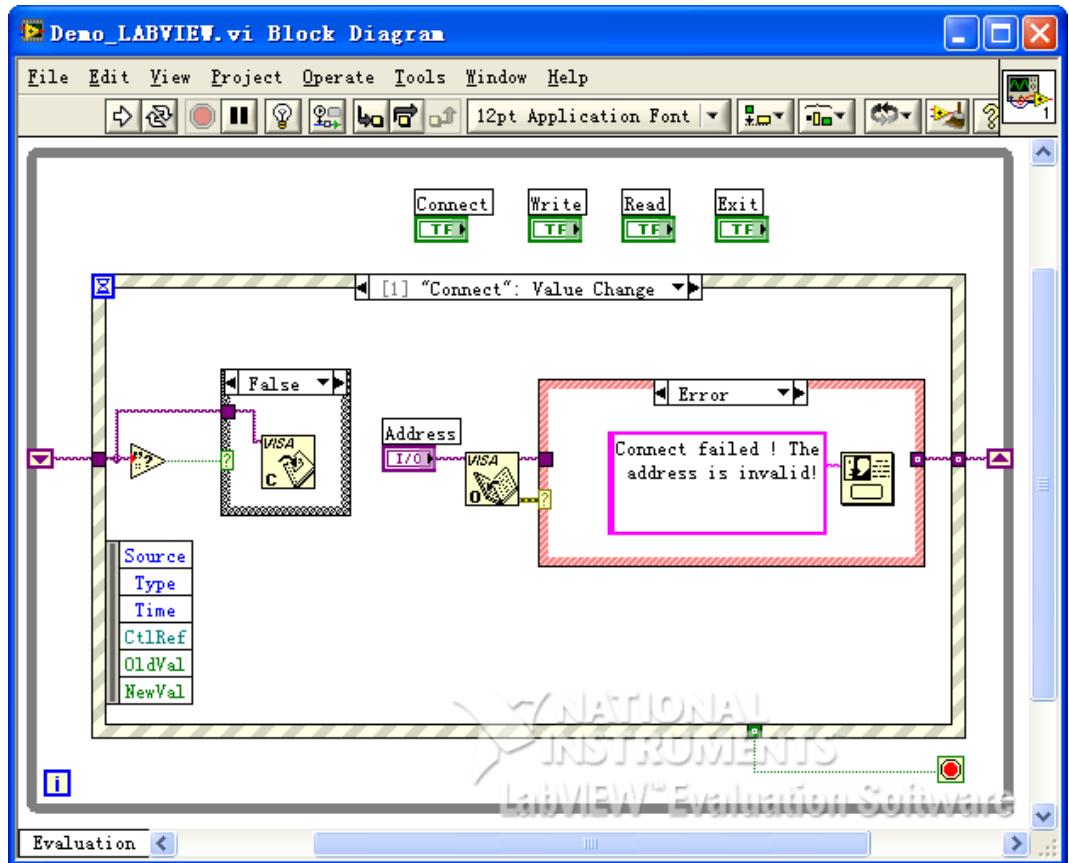


3. 点击 Window 菜单下的 Show Block Diagram, 创建事件结构。

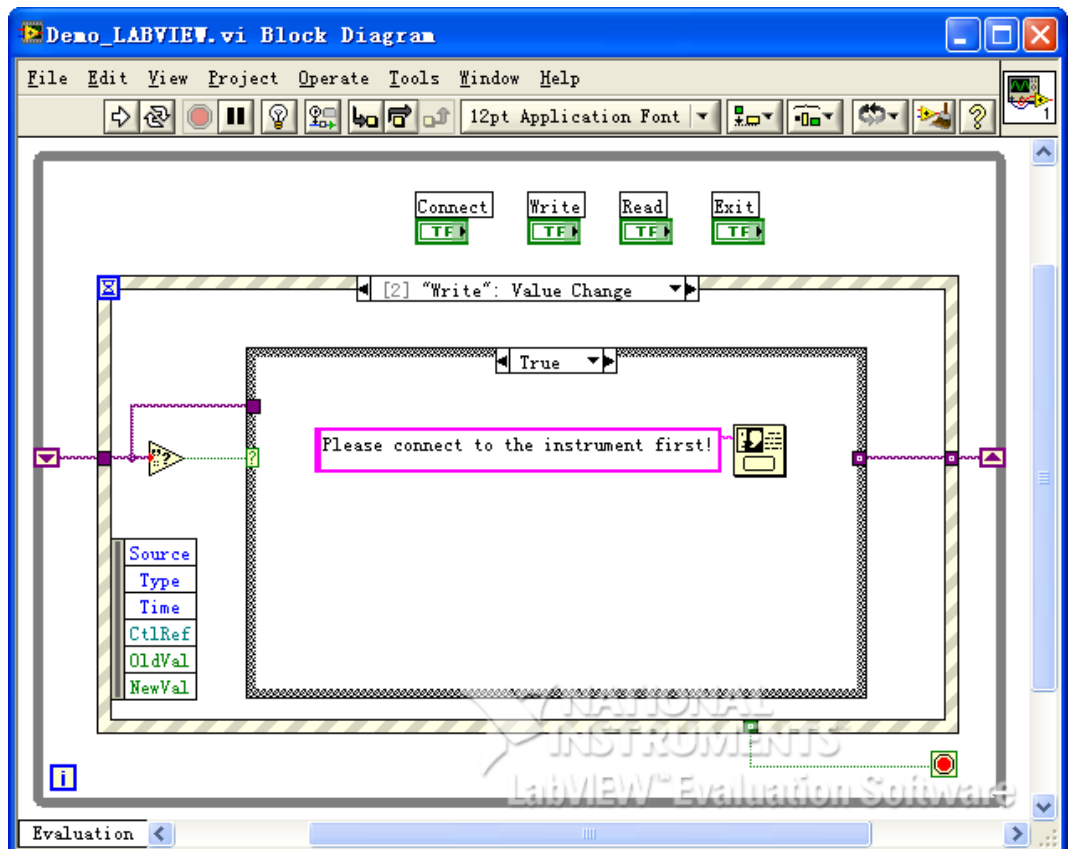


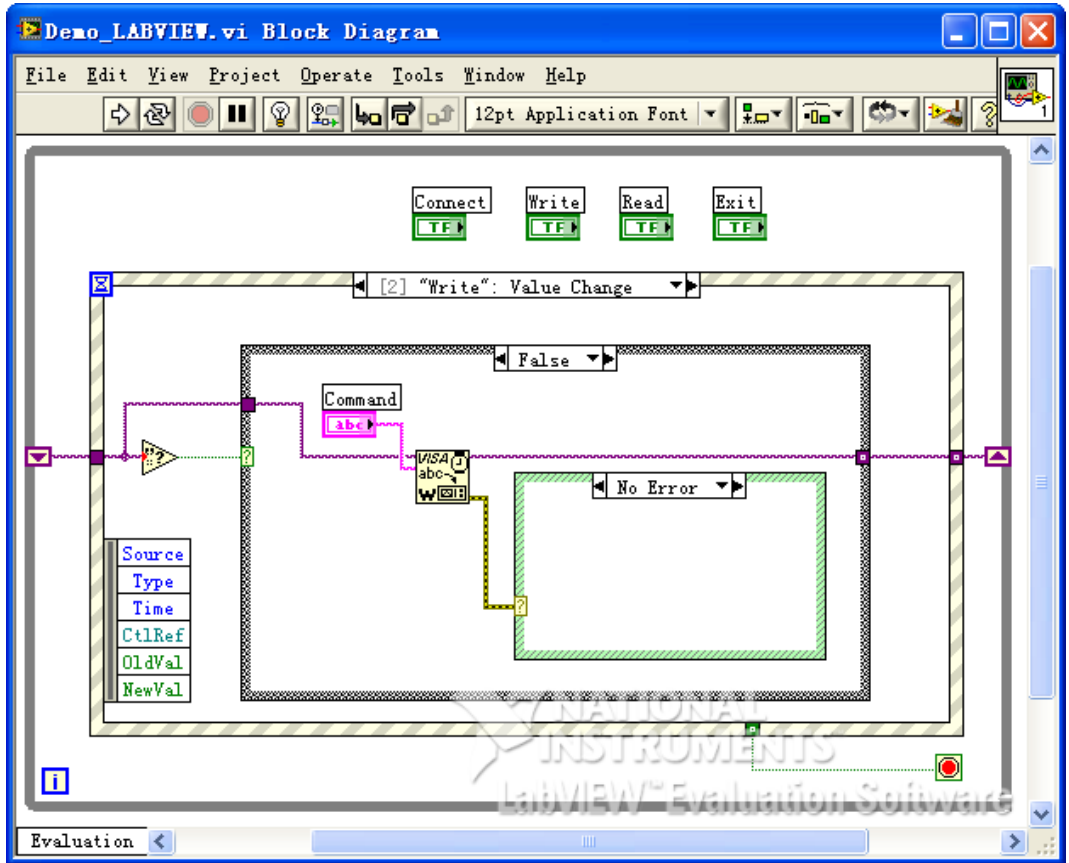
4. 添加事件, 包括连接仪器、写操作、读操作和退出。

a. 连接仪器 (包括出错处理) :

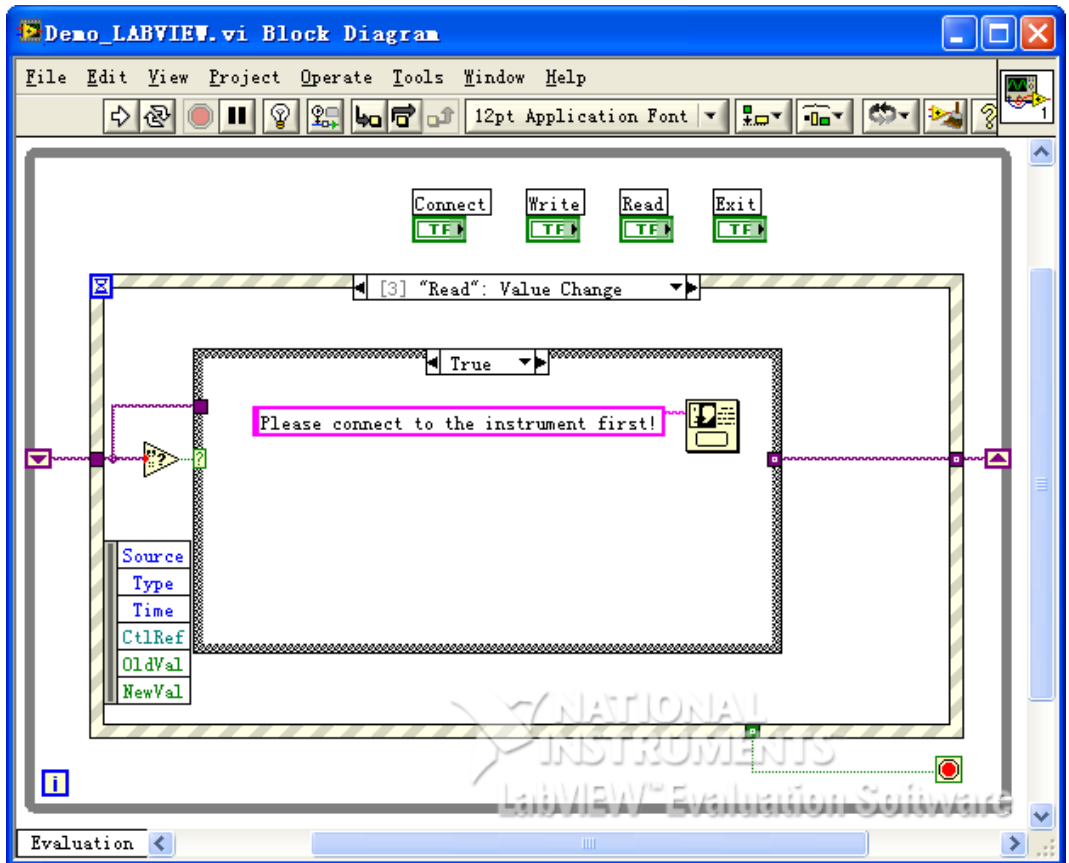


b. 写操作（包括出错判断）：

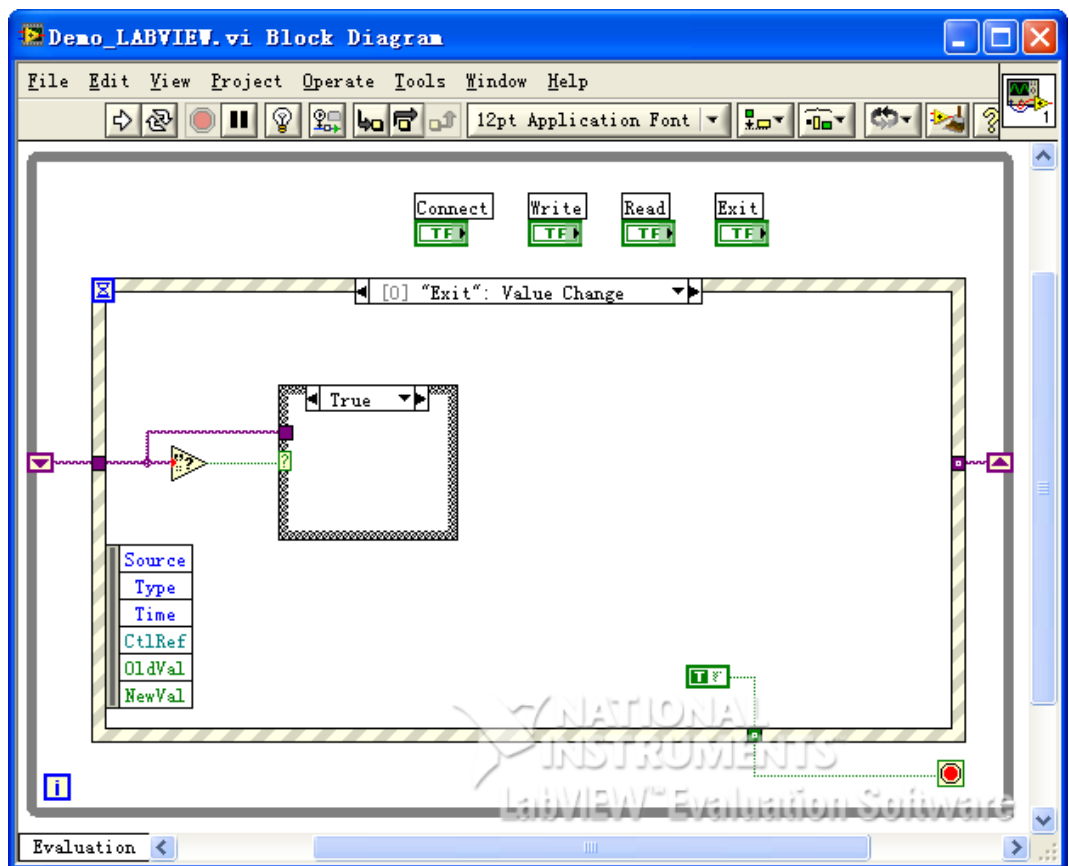




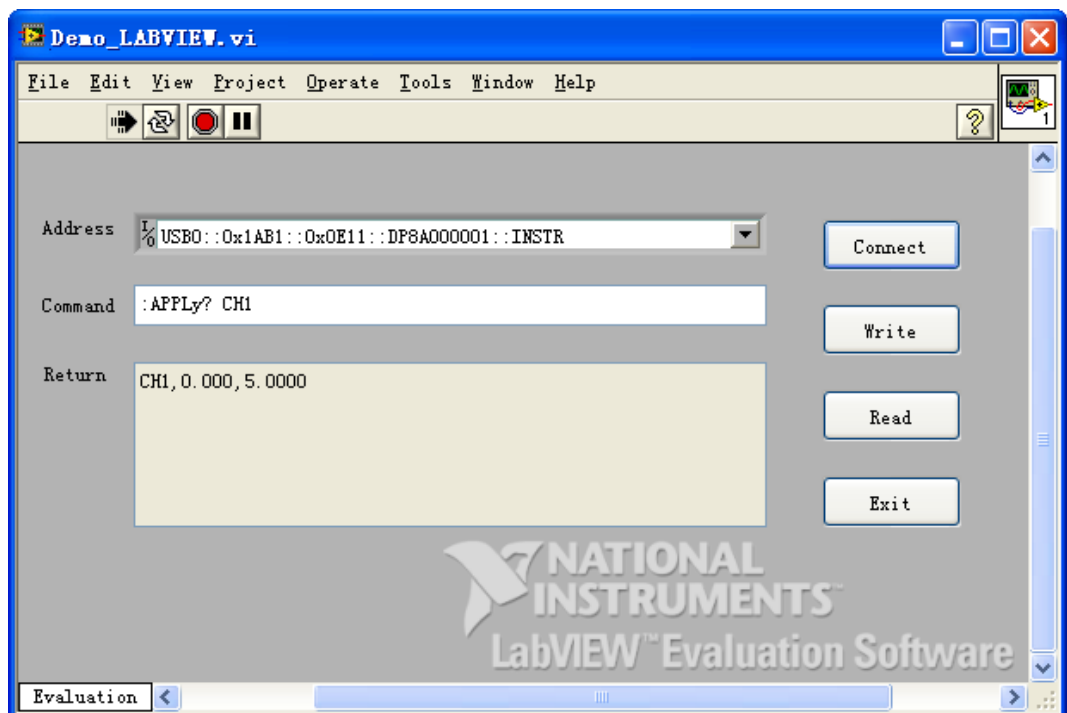
c. 读操作（包括出错处理）：



d. 退出:



5. 运行程序，出现如下图所示界面。点击 Address 下拉框选择 VISA 资源名称，点击 Connect 连接仪器，在 Command 文本框中输入命令，点击 Write 写入仪器。若为查询命令，点击 Read，Return 文本框显示返回值。



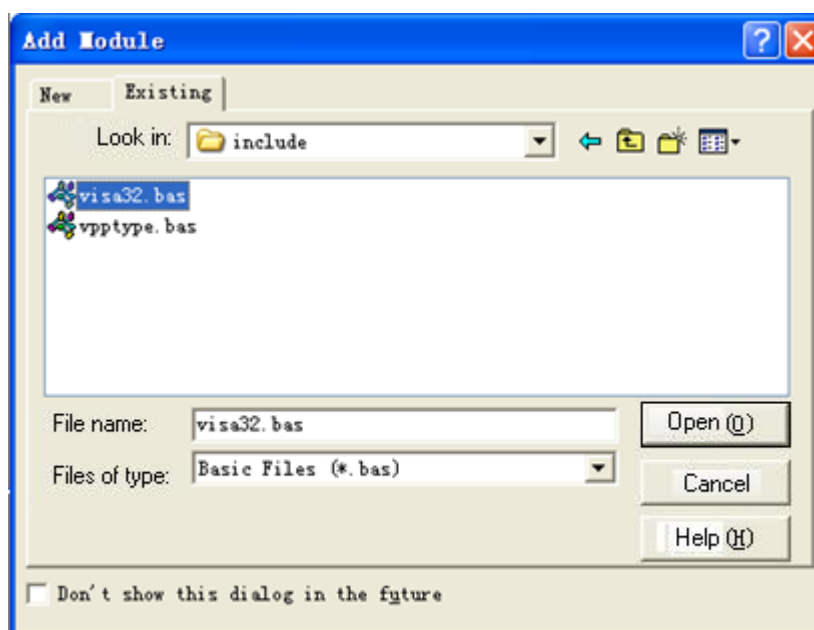
5.3 Visual Basic 编程实例

本例使用的程序： Visual Basic 6.0

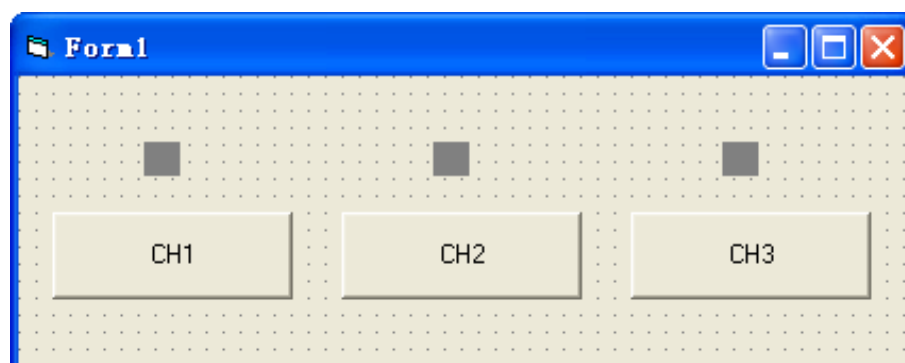
本例实现的功能： 打开电源的三个通道并显示通道对应的颜色。

进入 Visual Basic 6.0 编程环境，按照下列步骤操作：

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



3. 添加三个 CommandButton 控件分别表示 CH1、CH2 和 CH3，添加三个 Text 控件分别表示三个通道的状态（默认显示灰色，通道打开时显示通道对应的颜色），分别为 Label1(0)、Label1(1)和 Label1(2)。布局如下图所示。



4. 打开 Project > Project1 Properties 中的 General 选项卡，在 Startup Object 下拉框中选择 Form1。
5. 双击 CH1 按钮进入编程环境，添加如下代码，即可实现对 CH1~CH3 的控制。以下为 CH1 的代码，其它通道代码类似。

```

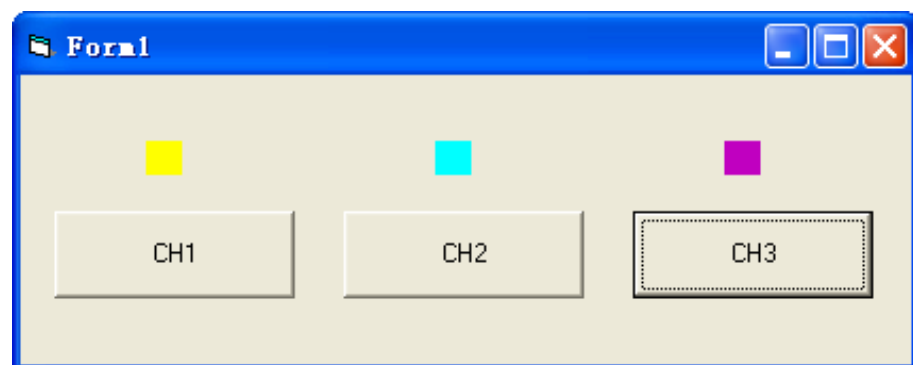
Dim defrm As Long
Dim vi As Long
Dim strRes As String * 200
Dim list As Long
Dim nmatches As Long
Dim matches As String * 200 '保留获取设备号
Dim s32Disp As Integer
' 获得 visa 的 usb 资源
Call viOpenDefaultRM(defrm)
Call viFindRsrc(defrm, "USB?* ", list, nmatches, matches)
' 打开设备
Call viOpen(defrm, matches, 0, 0, vi)
' 发送询问 CH1 状态命令
Call viVPrintf(vi, ":CHAN1:DISP?" + Chr$(10), 0)
' 获取 CH1 状态
Call viVScanf(vi, "%t", strRes)
s32Disp = CInt(strRes)
If (s32Disp = 1) Then
' 发送设置命令
Call viVPrintf(vi, ":CHAN1:DISP 0" + Chr$(10), 0)
Label1(0).ForeColor = &H808080 '灰色
Else
Call viVPrintf(vi, ":CHAN1:DISP 1" + Chr$(10), 0)
Label1(0).ForeColor = &HFFFF& '黄色
End If
' 关闭资源
Call viClose(vi)
Call viClose(defrm)

```

6. 运行结果

- a. 点击“CH1”打开 CH1，CH1 按钮上方显示黄色；
- b. 点击“CH2”打开 CH2，CH2 按钮上方显示蓝色；
- c. 点击“CH3”打开 CH3，CH3 按钮上方显示玫红色。

运行结果如下图所示：



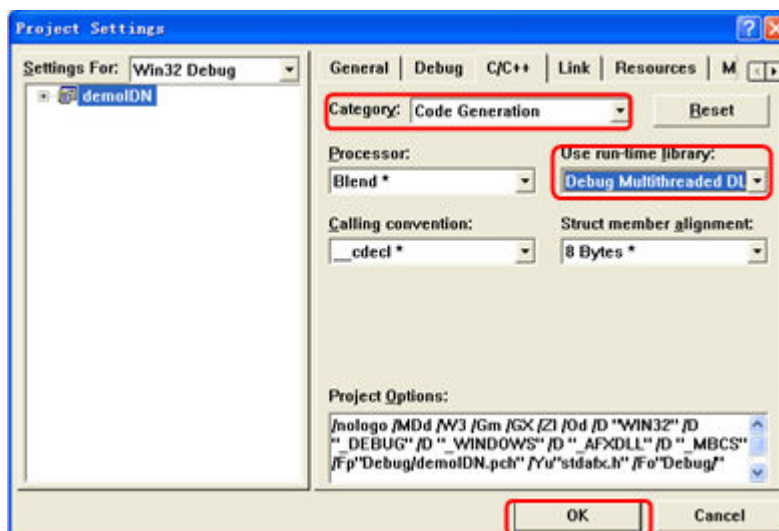
5.4 Visual C++ 编程实例

本例使用的程序：Visual C++6.0

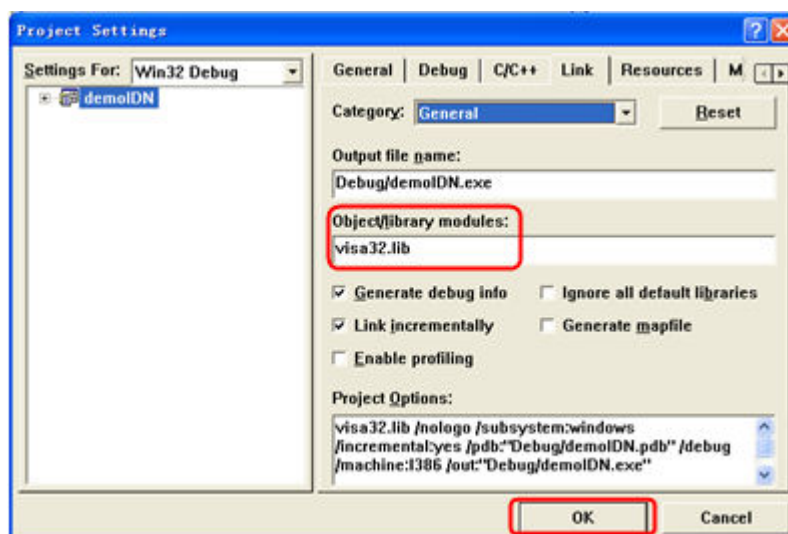
本例实现的功能：查找仪器地址、连接仪器、发送命令并读取返回值。

进入 Visual C++6.0 编程环境，按照下列步骤操作：

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



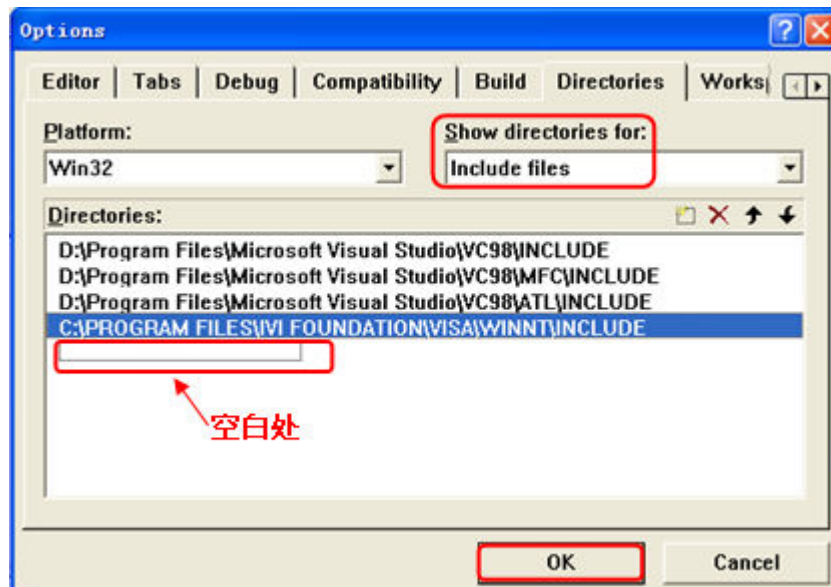
3. 打开 Project > Settings 中的 Link 选项卡，在 Object/library modules 中手动添加 visa32.lib。



4. 打开 Tools > Options 中的 Directories 选项卡。

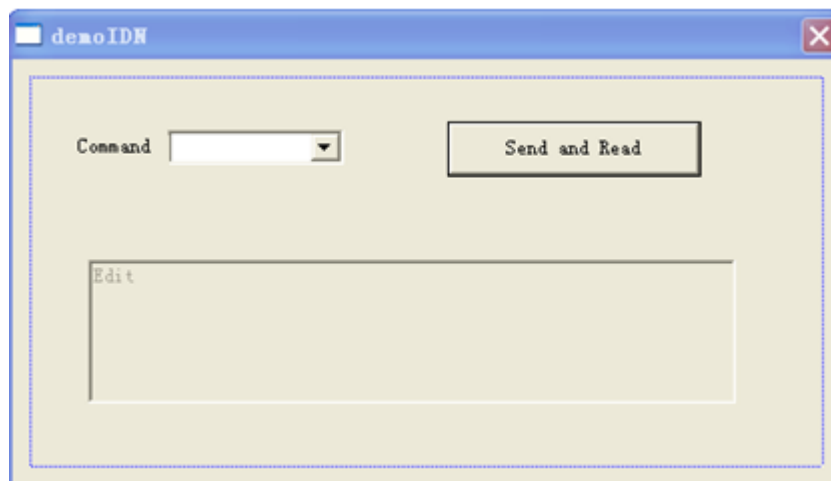
在 Show directories for 中选择 Include files，双击 Directories 选框中的空白处添加 Include 的路径：C:\Program Files\IVI Foundation\VISA\WinNT\include。

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



注：至此，VISA 库添加完毕。

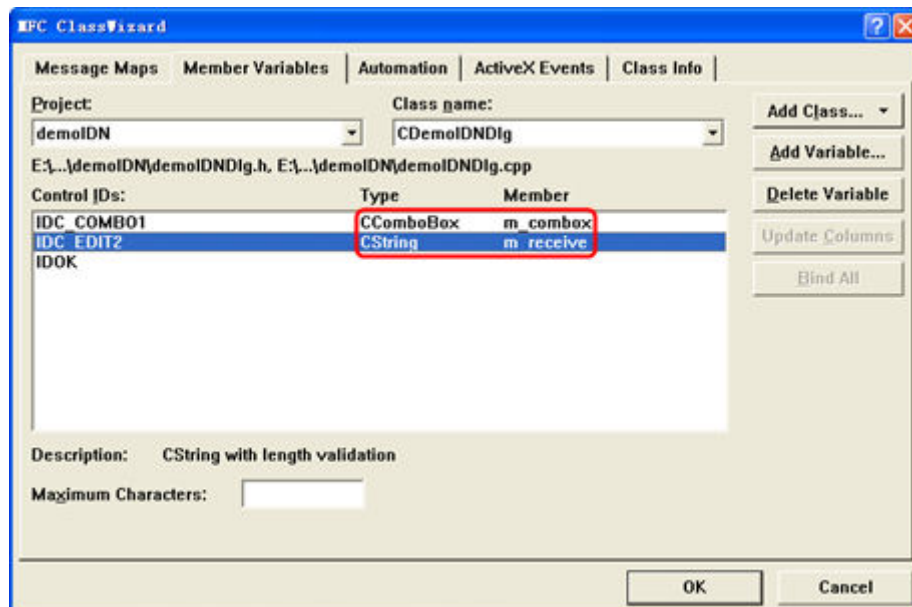
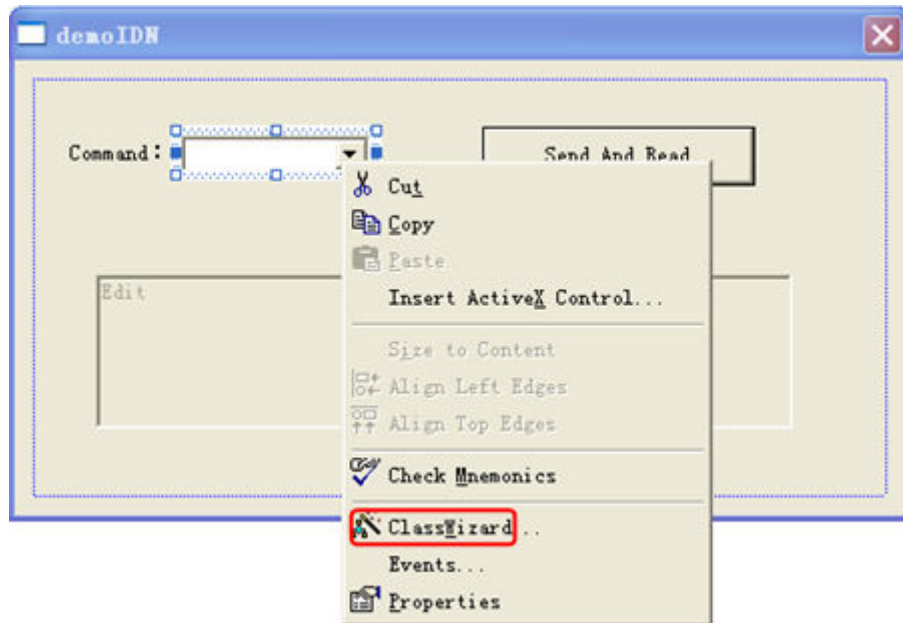
5. 添加 Text、Combo Box、Button 和 Edit Box 控件。布局如下所示：



6. 修改控件属性。

- a. 将 Text 命名为 “Command” 。
- b. 打开 Combo Box 属性中的 Data 项，手动输入命令：*IDN?
- c. 打开 Edit Box 属性中的 General 项，选中 Disabled。
- d. 将 Button 命名为 Send and Read。

7. 为 Combo Box 和 Edit 控件分别添加变量 m_combox 和 m_receive。



8. 添加代码。

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

```
ViSession defaultRM, vi;
char buf [256] = {0};
CString s, strTemp;
char* stringTemp;

ViChar buffer [VI_FIND_BUFLen];
ViRsrc matches=buffer;
ViUInt32 nmatches;
ViFindList list;

viOpenDefaultRM (&defaultRM);
//获取 visa 的 USB 资源
```

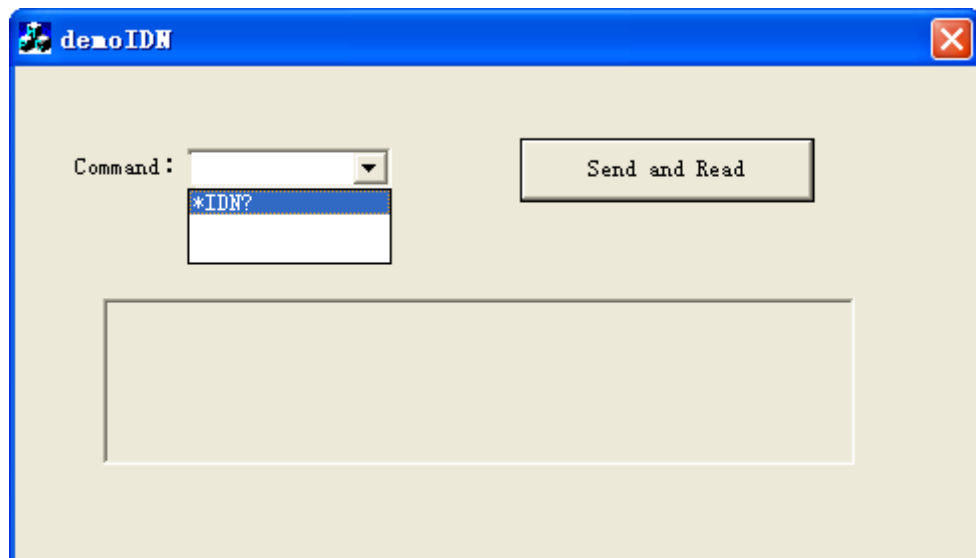
```
viFindRsrc(defaultRM, "USB?*",&list,&nmatches, matches);
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);

//发送接收到的命令
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);
strTemp = strTemp + "\n";
stringTemp = (char*)(LPCTSTR)strTemp;
viPrintf (vi,stringTemp);

//读取结果
viScanf (vi, "%t\n", &buf);

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

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



苏州总部

名称：普源精电科技股份有限公司
地址：中国苏州市高新区科灵路8号
电话：0512-66706688
邮箱：info-cn@rigol.com

北京研发总部

名称：北京普源精电科技有限公司
地址：北京市海淀区丰豪东路9号院4号楼
邮箱：info-cn@rigol.com



RIGOL 服务与支持热线: 4006 200 002

RIGOL 官网网址: www.rigol.com

版权所有 © 2023 普源精电科技股份有限公司