

UDP3305S系列可编程直流电源

编程手册

REV 1.3
2023.03

UNI-T®

保证和声明

版权

2023 优利德中国科技有限公司

商标信息

UNI-T 是优利德中国科技有限公司的注册商标。

文档编号

软件版本

1.10

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

声明

- 本公司产品受中国及其它国家和地区的专利（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，**UNI-T** 概不负责。
- 未经 **UNI-T** 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

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

一、SCPI 指令简介

SCPI (Standard Commands for Programmable Instruments, 即可编程仪器标准命令集) 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。本节简介 SCPI 命令的格式、符号、参数和缩写规则。

指令格式

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。命令行通常以冒号“:”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。命令关键字和第一个参数之间以空格分开。命令字符串必须以一个 <换行> (<NL>) 字符结尾。命令行后面添加问号“?”通常表示对此功能进行查询。

符号说明

下面四种符号不是 SCPI 命令中的内容, 不随命令发送, 但是通常用于辅助说明命令中的参数。

- **大括号 {}**
大括号中通常包含多个可选参数, 发送命令时必须选择其中一个参数。
如: INSTRument[:SELEct] {CH1|CH2|CH3|SER|PARA} 命令。
- **竖线 |**
竖线用于分隔多个参数选项, 发送命令时必须选择其中一个参数。
如: INSTRument[:SELEct] {CH1|CH2|CH3|SER|PARA} 命令。
- **方括号 []**
方括号中的内容 (命令关键字) 是可省略的。如果省略参数, 仪器将该参数设置为默认值。例如: 对于: INSTRument[:SELEct] {CH1|CH2|CH3|SER|PARA} 命令, [:SELEct] 是可省略的。
- **三角括号 <>**
三角括号中的参数必须用一个有效值来替换。例如: :LISTout:TEMPlet:POINTs <point> 命令的一个实例如下: :LISTout:TEMPlet:POINTs 20

参数说明

本手册介绍的命令中所含的参数可以分为以下 5 种类型: 布尔型、整型、实型、离散型、ASCII 字符串。

- **布尔型**
参数取值为“ON” (1) 或“OFF” (0)。例如: :LISTout[:STATe] {0|1|OFF|ON}
- **整型**
除非另有说明, 参数在有效值范围内可以取任意整数值。注意: 此时, 请不要设置参数为小数格式, 否则将出现异常。
- **实型**

除非另有说明，参数在有效值范围内可以取任意值。

- **离散型**

参数只能取指定的几个数值或字符。例如：`:SOURce:Mode {NORMal|SER|PARA}`命令的参数只能为 `NORMal`、`SER`、`PARA`。

- **ASCII 字符串**

字符串参数实际上可包含所有 ASCII 字符集。字符串必须以配对的引号开始和结尾；可以用单引号或双引号。引号分隔符也可以作为字符串的一部分，只需键入两次并且不在中间添加任何字符，例如设置 IP：`SYST:COMM:LAN:IPAD "192.168.1.10"`。

简写规则

所有命令对大小写都能识别，可以全部采用大写或小写。如果要缩写，必须输完命令格式中的所有大写字母。

数据返回

数据返回分为单个数据和批量数据返回，单个数据返回相对应的参数类型，其中实型返回用科学计数法表示，e 前部分小数点后面保留三位数据，e 部分保留三位数据；批量数据返回必须符合 IEEE 488.2 #格式的字符串数据，其格式：`'#'+ 长度所占的字符位数 [固定为一个字符] + 有效数据长度的 ASCII 值 + 有效数据 + 结束符['\n']`，例如 `#3123xxxxxxxxxxxxxxxxxxxx\n` 表示的具有 123 个字节有效批量数据返回格式，其中‘3’表示“123”占 3 个字符位。

注意：当返回数据为无效数据均用*表示。

二、建立通信

本仪器可以通过USB、LAN、RS232接口与计算机之间进行通信，命令结束符为“\n”，即十六进制数“0x0A”。

使用 USB 接口

本仪器作为 USB-TMC 设备，VISA 编程资源描述符类似为：

“USB0::0x0483::0x5740::UDP51183557335E::INSTR”。

使用 LAN 接口

本仪器支持VXI和SOCKET编程模式，SOCKET模式的端口号为5025，VISA编程资源描述符分别类似为：

“TCPIP0::192.168.10.142::INSTR”；

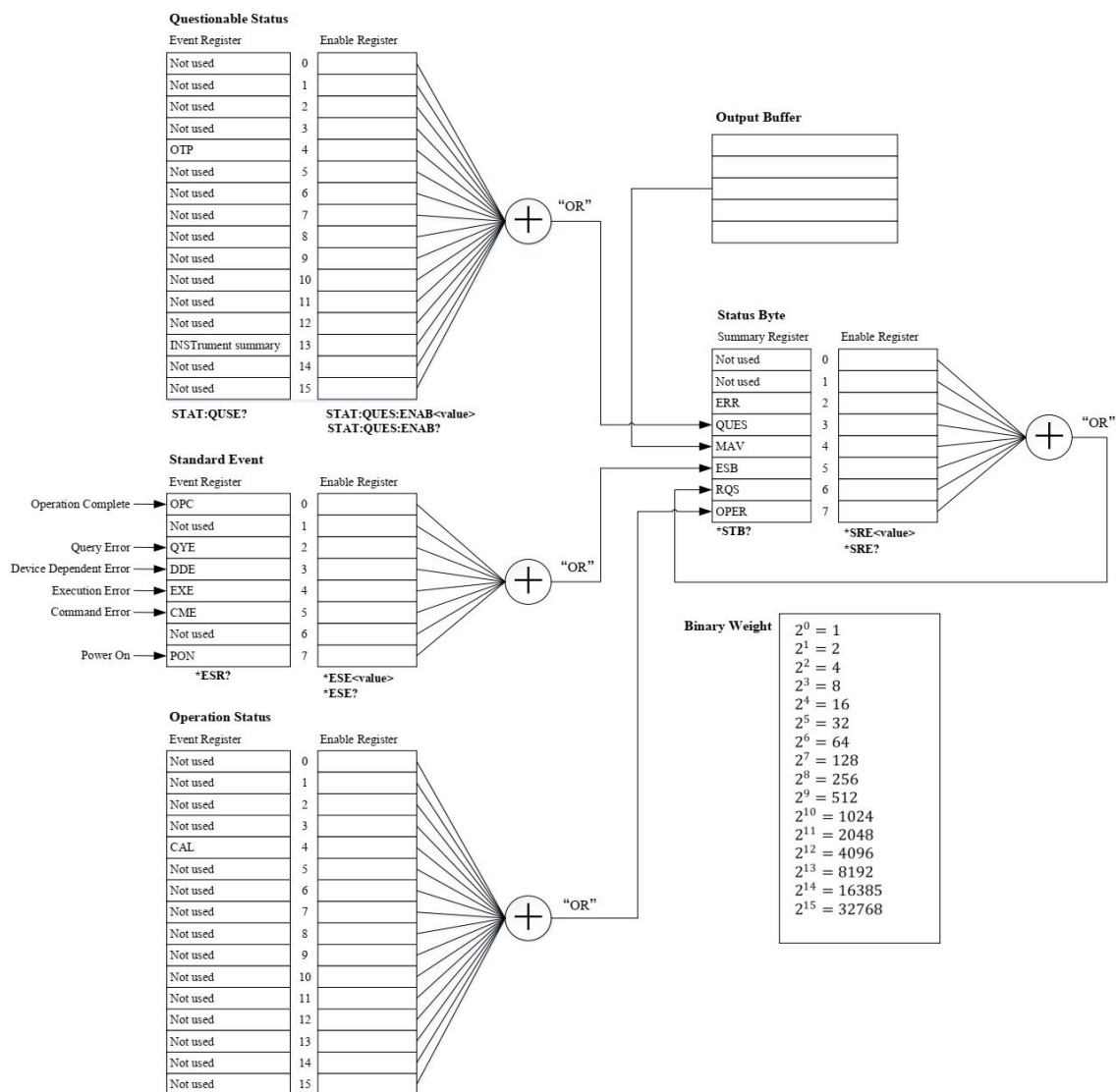
“TCPIP0::192.168.10.142::5025::SOCKET”。

使用 RS232 接口

请使用交叉类型RS232连接线，并根据实际情况选择恰当的波特率。本仪器支持4800、7200、9600、14400、19200、38400、57600、115200、128000多种波特率，请在产品的系统设置内进行设置。

三、SCPI 状态寄存器

标准寄存器



可疑寄存器的定义如下：

Bit	定义	十进制值	含义
4	OTP	16	过温保护
13	INSTrument summary	8192	通道可疑寄存器和通道 SUMMARY 寄存器的汇总
其它	Not used	-	未使用，始终为 0

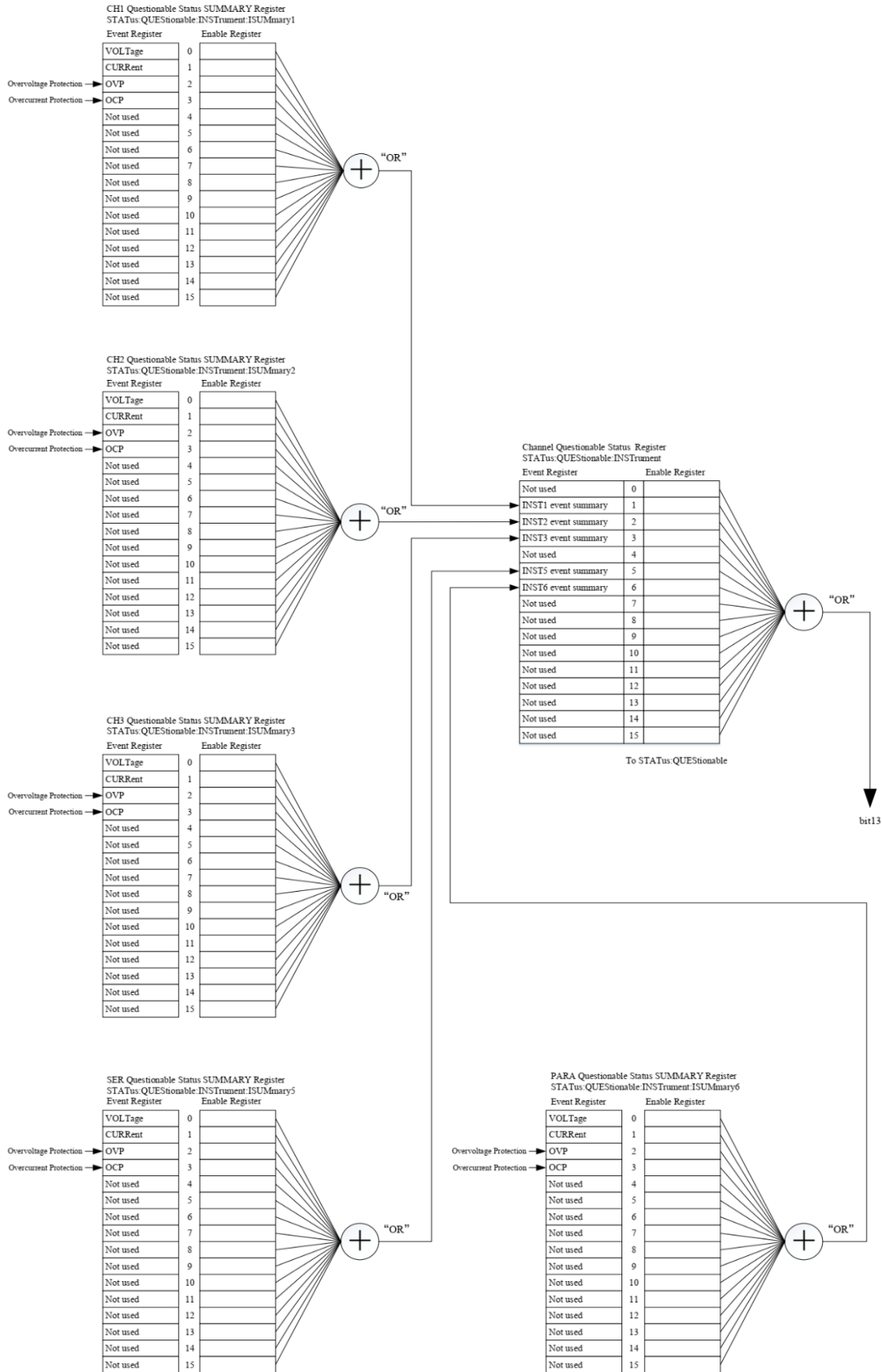
标准事件寄存器的定义如下：

Bit	定义	十进制值	含义
0	OPC	1	操作完成，*OPC 指令执行完成
1	Not used	2	未使用，始终为 0
2	QYE	4	查询出错
3	DDE	8	设备错误
4	EXE	16	执行错误
5	CME	32	命令错误
6	Not used	64	未使用，始终为 0
7	PON	128	上电检测

状态字节的定义如下：

Bit	定义	十进制值	含义
0	Not used	1	未使用，始终为 0
1	Not used	2	未使用，始终为 0
2	ERR	4	错误输出队列中含有消息
3	QUES	8	可疑寄存器上报了事件
4	MAV	16	输出缓冲区中有数据可用
5	ESB	32	标准事件寄存器上报了事件
6	RSQ	64	请求服务
7	OPER	128	操作寄存器上报了事件

通道可疑寄存器



各通道SUMMARY寄存器的定义如下：

Bit	定义	十进制值	含义
0	VOLTage	1	电压不可控，电源处于恒流模式
1	CURRent	2	电流不可控，电源处于恒压模式
2	OVP	4	过压保护
3	OCP	8	过流保护
其它	Not used	-	未使用，始终为 0

通道可疑寄存器的定义如下：

Bit	定义	十进制值	含义
0	Not used	1	未使用，始终为 0
1	INST1 event summary	2	CH1 独立通道的事件汇总
2	INST2 event summary	4	CH2 独立通道的事件汇总
3	INST3 event summary	8	CH3 独立通道的事件汇总
4	Not used	16	未使用，始终为 0
5	INST5 event summary	32	SER 串联通道的事件信息
6	INST6 event summary	64	PARA 并联通道的事件信息
其它	Not used	-	未使用，始终为 0

四、SCPI 命令详解

1. IEEE488 通用命令

***CLS**

格式: *CLS

功能: 设置标准事件寄存器的使能寄存器的值。

***ESE**

格式: *ESE <enable value>

功能: 设置标准事件寄存器的使能寄存器的值。

***ESE?**

格式: *ESE?

功能: 查询标准事件寄存器的使能寄存器的值。

***ESR?**

格式: *ESR?

功能: 查询标准事件寄存器的事件寄存器的值。

***IDN?**

格式: *IDN?

功能: 查询设备基本信息, 厂商、型号、序列号、软件版本号。

***OPC**

格式: *OPC

功能: 执行完成本条指令后, 标准事件寄存器中的“OPC”位被置位。

***OPC?**

格式: *OPC?

功能: 所有指令执行完成后, 返回“1”。

***RST**

格式: *RST

功能: 将设备恢复至默认状态。

***SRE**

格式: *SRE <enable value>

功能: 设置状态寄存器的使能寄存器的值。

***SRE?**

格式: *SRE?

功能: 查询状态寄存器的使能寄存器的值。

***STB?**

格式: *STB?

功能: 查询状态字节寄存器。

***TST?**

格式: *TST?

功能: 查询仪器自检结果。仪器正常时返回 0，异常时返回 1。

***WAI**

格式: *WAI

功能: 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。

***SAV**

格式: *SAV {1|2|3|4|5|6|7|8|9|10}

功能: 保存状态设置数据到内部存储器的指定位置。

说明: 等价于命令 “:MEMory[:STATe]:STORe STA, {1|2|3|4|5|6|7|8|9|10}”

举例: *SAV, 1

***RCL**

格式: *RCL {1|2|3|4|5|6|7|8|9|10}

功能: 从内部存储器的指定位置加载状态设置数据。

说明: 等价于命令 “:MEMory[:STATe]:LOAD STA, {1|2|3|4|5|6|7|8|9|10}”

举例: *RCL, 1

2. APPLY

:APPLY

格式: :APPLY [CH1|CH2|CH3|SER|PARA],[<volt>|MINimum|MAXimum], [<curr>|MINimum|MAXimum]

功能: 设置指定通道为当前通道，同时可设置该通道的电压、电流设置值。

说明: 只有在普通模式时才能选择 “CH1、CH2”，只有在串联模式时才能选择 “SER”，只有在并联模式时才能选择 “PARA”。同时省略 “volt” 和 “curr” 参数时，只设置指定通道为当前通道，不改变通道的电压、电流设置值。

举例: :APPLY CH1,15.00V, 2.000A

:APPLy?

格式: **:APPLy?** [CH1|CH2|CH3|SER|PARA],[CURRent|VOLTage]

功能: 查询某通道的电压、电流设置值。

说明: 如果省略通道参数，则默认指定当前通道为需要查询的通道。如果指定“CURRent”，则返回电流设置值；如果指定“VOLTage”，则返回电压设置值；如果不指定，则同时返回电压、电流设置值。

举例: :APPLy? CH1, VOLT

返回: CH1, 15.00

3. INSTRument

:INSTRument[:SELEct]

:INSTRument[:SELEct]

格式: **:INSTRument[:SELEct]** {CH1|CH2|CH3|SER|PARA}

:INSTRument[:SELEct] {CH1|CH2|CH3|SER|PARA}

功能: 设置指定通道为当前通道。

说明: 只有在普通模式时才能选择“CH1、CH2”，只有在串联模式时才能选择“SER”，只有在并联模式时才能选择“PARA”。

设置电源的模式请参见命令“:SOURce:Mode {NORMal|SER|PARA}”

举例: :INSTRument:SELE Ch3

:INSTRument[:SELEct] ?

:INSTRument[:SELEct] ?

格式: **:INSTRument[:SELEct] ?**

:INSTRument[:SELEct] ?

功能: 查询当前通道。

说明: 返回“CH1|CH2|CH3|SER|PARA”。

举例: :INSTRument:SELE?

返回: CH1

:INSTRument:NSELEct

格式: **:INSTRument:NSELEct** {1|2|3|5|6}

功能: 设置指定通道为当前通道。

说明: 1 对应 CH1，2 对应 CH2，3 对应 CH3，5 对应 SER，6 对应 PARA。只有在普通模式时才能选择“1、2”，只有在串联模式时才能选择“5”，只有在并联模式时才能选择“6”。

举例: :INSTRument:NSELEct 3

:INSTRument:NSELEct?

格式: **:INSTRument:NSELEct?**

功能: 查询当前通道。

说明: 返回 “1|2|3|5|6”。

举例: :INSTrument:NSElect?

返回: 3

4. SOURce

:SOURce:Mode

格式: :SOURce:Mode {NORMAl|SER|PARA}

功能: 设置电源的工作模式。

说明: NORMAl 设置为普通独立模式, SER 设置为串联模式, PARA 设置为并联模式。电源切换模式时需要一段时间, 在此期间执行一些与电源工作模式相关的命令会导致命令执行失败; 因此, 切换电源工作模式后, 至少间隔 500 毫秒后再执行新的命令。

举例: :SOURce:Mode SER

:SOURce:Mode?

格式: :SOURce:Mode?

功能: 查询电源当前的工作模式。

说明: 返回 “NORMAl|SER|PARA”。

举例: :SOURce:Mode?

返回: SER

[:SOURce#]:VOLTage[:LEVel][:IMMediate][:AMPLitude]

格式: [:SOURce#]:VOLTage[:LEVel][:IMMediate][:AMPLitude] {<vol>|MINimum|MAXimum}

功能: 设置某通道的电压设置值, 并设置该通道为当前通道。

说明: #的可选值为 “1|2|3|5|6”, 1 对应 CH1, 2 对应 CH2, 3 对应 CH3, 5 对应 SER, 6 对应 PARA。如果省略 [:SOURce#]或#, 则默认值为 CH1。只有在普通模式时才能选择 “1、2”, 只有在串联模式时才能选择 “5”, 只有在并联模式时才能选择 “6”。

举例: :SOURce1:VOLTage 25.00

[:SOURce#]:VOLTage[:LEVel][:IMMediate][:AMPLitude]?

格式: [:SOURce#]:VOLTage[:LEVel][:IMMediate][:AMPLitude]?

功能: 查询某通道的电压设置值。

说明: #的可选值为 “1|2|3|5|6”, 1 对应 CH1, 2 对应 CH2, 3 对应 CH3, 5 对应 SER, 6 对应 PARA。如果省略 [:SOURce#]或#, 则默认值为 CH1。

举例: :SOURce1:VOLTage?

返回: 25.00

[:SOURce#]:VOLTage:PROTection[:LEVel]

格式: [:SOURce#]:VOLTage:PROTection[:LEVel] {<vol>|MINimum|MAXimum}

功能: 设置某通道的过压保护值, 并设置该通道为当前通道。

说明: #的可选值为“1|2|3|5|6”, 1 对应 CH1, 2 对应 CH2, 3 对应 CH3, 5 对应 SER, 6 对应 PARA。如果省略[:SOURce#]或#, 则默认值为 CH1。只有在普通模式时才能选择“1、2”, 只有在串联模式时才能选择“5”, 只有在并联模式时才能选择“6”。

举例: :SOURce1:VOLTage:PROTection 30.00

[:SOURce#]:VOLTage:PROTection[:LEVel]?

格式: [:SOURce#]:VOLTage:PROTection[:LEVel]?

功能: 查询某通道的过压保护值。

说明: #的可选值为“1|2|3|5|6”, 1 对应 CH1, 2 对应 CH2, 3 对应 CH3, 5 对应 SER, 6 对应 PARA。如果省略[:SOURce#]或#, 则默认值为 CH1。

举例: :SOURce1:VOLTage:PROTection?

返回: 30.00

[:SOURce#]:VOLTage:PROTection:STATe

格式: [:SOURce#]:VOLTage:PROTection:STATe {0|1|OFF|ON}

功能: 设置某通道的过压保护开关, 并设置该通道为当前通道。

说明: #的可选值为“1|2|3|5|6”, 1 对应 CH1, 2 对应 CH2, 3 对应 CH3, 5 对应 SER, 6 对应 PARA。如果省略[:SOURce#]或#, 则默认值为 CH1。只有在普通模式时才能选择“1、2”, 只有在串联模式时才能选择“5”, 只有在并联模式时才能选择“6”。

举例: :SOURce1:VOLTage:PROTection:STATe 1

[:SOURce#]:VOLTage:PROTection:STATe?

格式: [:SOURce#]:VOLTage:PROTection:STATe?

功能: 查询某通道的过压保护开关。

说明: #的可选值为“1|2|3|5|6”, 1 对应 CH1, 2 对应 CH2, 3 对应 CH3, 5 对应 SER, 6 对应 PARA。如果省略[:SOURce#]或#, 则默认值为 CH1。返回“ON | OFF”

举例: :SOURce1:VOLTage:PROTection:STATe?

返回: ON

[:SOURce#]:CURRent [:LEVel][:IMMEDIATE][:AMPLitude]

格式: [:SOURce#]:CURRent [:LEVel][:IMMEDIATE][:AMPLitude] {<curr>|MINimum|MAXimum}

功能: 设置某通道的电流设置值, 并设置该通道为当前通道。

说明: #的可选值为“1|2|3|5|6”, 1 对应 CH1, 2 对应 CH2, 3 对应 CH3, 5 对应 SER, 6 对应 PARA。如果省略[:SOURce#]或#, 则默认值为 CH1。只有在普通模式时才能选择“1、2”, 只有在串联模式时才能选择“5”, 只有在并联模式时才能选择“6”。

举例: :SOURce1:CURRent 5.000

[:SOURce#]:CURRent [:LEVel][:IMMEDIATE][:AMPLitude]?

格式: [:SOURce#]:CURRent [:LEVel][:IMMEDIATE][:AMPLitude]?

功能: 查询某通道的电流设置值。

说明: #的可选值为“1|2|3|5|6”，1 对应 CH1，2 对应 CH2，3 对应 CH3，5 对应 SER，6 对应 PARA。如果省略[:SOURce#]或#，则默认值为 CH1。

举例: :SOURce1:CURRent?

返回: 5.000

[:SOURce#]:CURRent:PROTection[:LEVel]

格式: [:SOURce#]:CURRent:PROTection[:LEVel] {<curr>|MINimum|MAXimum}

功能: 设置某通道的过流保护值，并设置该通道为当前通道。

说明: #的可选值为“1|2|3|5|6”，1 对应 CH1，2 对应 CH2，3 对应 CH3，5 对应 SER，6 对应 PARA。如果省略[:SOURce#]或#，则默认值为 CH1。只有在普通模式时才能选择“1、2”，只有在串联模式时才能选择“5”，只有在并联模式时才能选择“6”。

举例: :SOURce1:CURRent:PROTection 5.000

[:SOURce#]:CURRent:PROTection[:LEVel]?

格式: [:SOURce#]:CURRent:PROTection[:LEVel]?

功能: 查询某通道的过流保护值。

说明: #的可选值为“1|2|3|5|6”，1 对应 CH1，2 对应 CH2，3 对应 CH3，5 对应 SER，6 对应 PARA。如果省略[:SOURce#]或#，则默认值为 CH1。

举例: :SOURce1:CURRent:PROTection?

返回: 5.000

[:SOURce#]:CURRent:PROTection:STATe

格式: [:SOURce#]:CURRent:PROTection:STATe {0|1|OFF|ON}

功能: 设置某通道的过流保护开关，并设置该通道为当前通道。

说明: #的可选值为“1|2|3|5|6”，1 对应 CH1，2 对应 CH2，3 对应 CH3，5 对应 SER，6 对应 PARA。如果省略[:SOURce#]或#，则默认值为 CH1。只有在普通模式时才能选择“1、2”，只有在串联模式时才能选择“5”，只有在并联模式时才能选择“6”。

举例: :SOURce1:CURRent:PROTection:STATe 1

[:SOURce#]:CURRent:PROTection:STATe?

格式: [:SOURce#]:CURRent:PROTection:STATe?

功能: 查询某通道的过流保护开关。

说明: #的可选值为“1|2|3|5|6”，1 对应 CH1，2 对应 CH2，3 对应 CH3，5 对应 SER，6 对应 PARA。如果省略[:SOURce#]或#，则默认值为 CH1。返回“ON | OFF”。

举例: :SOURce1:CURRent:PROTection:STATe?

返回: ON

5. OUTPut

:OUTPut[:STATe]

格式: **:OUTPut[:STATe]** [CH1|CH2|CH3|SER|PARA|ALL] {0|1|OFF|ON}

功能: 打开或关闭某通道的输出开关，并设置该通道为当前通道。

说明: 如果省略通道参数，则默认指定当前通道为需要控制的通道。只有在普通模式时才能选择“CH1、CH2”，只有在串联模式时才能选择“SER”，只有在并联模式时才能选择“PARA”。

举例: :OUTPut:STATe CH1, ON

:OUTPut[:STATe]?

格式: **:OUTPut[:STATe]?** [CH1|CH2|CH3|SER|PARA]

功能: 查询某通道的输出开关状态。

说明: 如果省略通道参数，则默认指定当前通道为需要查询的通道。返回“ON | OFF”。

举例: :OUTPut:STATe? CH1

返回: ON

:OUTPut:CVCC?

格式: **:OUTPut:CVCC?** [CH1|CH2|CH3|SER|PARA]

功能: 查询某通道的恒压、恒流状态。

说明: 如果省略通道参数，则默认指定当前通道为需要查询的通道。返回“CV | CC”。

举例: :OUTPut:CVCC? CH1

返回: CV

:OUTPut:OVP:VALue

格式: **:OUTPut:OVP:VALue** [CH1|CH2|CH3|SER|PARA] {<vol>|MINimum|MAXimum}

功能: 设置某通道的过压保护值，并设置该通道为当前通道。

说明: 如果省略通道参数，则默认指定当前通道为需要控制的通道。只有在普通模式时才能选择“CH1、CH2”，只有在串联模式时才能选择“SER”，只有在并联模式时才能选择“PARA”。

举例: :OUTPut:OVP:VALue CH1, 5

:OUTPut:OVP:VALue?

格式: **:OUTPut:OVP:VALue?** [CH1|CH2|CH3|SER|PARA]

功能: 查询某通道的过压保护值。

说明: 如果省略通道参数，则默认指定当前通道为需要查询的通道。

举例: :OUTPut:OVP:VALue? CH1

返回: 5.00

:OUTPut:OVP[:STATe]

格式: **:OUTPut:OVP[:STATe]** [CH1|CH2|CH3|SER|PARA] {0|1|OFF|ON}

功能: 设置某通道的过压保护开关，并设置该通道为当前通道。

说明: 如果省略通道参数, 则默认指定当前通道为需要控制的通道。只有在普通模式时才能选择“CH1、CH2”, 只有在串联模式时才能选择“SER”, 只有在并联模式时才能选择“PARA”。

举例: :OUTPut:OVP:STATe CH1, ON

:OUTPut:OVP[:STATe]?

格式: :OUTPut:OVP[:STATe]? [CH1|CH2|CH3|SER|PARA]

功能: 查询某通道的过压保护开关。

说明: 如果省略通道参数, 则默认指定当前通道为需要查询的通道。返回“ON | OFF”

举例: :OUTPut:OVP:STATe? CH1

返回: ON

:OUTPut:OCP:VALue

格式: :OUTPut:OCP:VALue [CH1|CH2|CH3|SER|PARA,] {<curr>|MINimum|MAXimum}

功能: 设置某通道的过流保护值, 并设置该通道为当前通道。

说明: 如果省略通道参数, 则默认指定当前通道为需要控制的通道。只有在普通模式时才能选择“CH1、CH2”, 只有在串联模式时才能选择“SER”, 只有在并联模式时才能选择“PARA”。

举例: :OUTPut:OCP:VALue CH1, 5.1

:OUTPut:OCP:VALue?

格式: :OUTPut:OCP:VALue? [CH1|CH2|CH3|SER|PARA]

功能: 查询某通道的过流保护值。

说明: 如果省略通道参数, 则默认指定当前通道为需要查询的通道。

举例: :OUTPut:OCP:VALue? CH1

返回: 5.100

:OUTPut:OCP[:STATe]

格式: :OUTPut:OCP[:STATe] [CH1|CH2|CH3|SER|PARA,] {0|1|OFF|ON}

功能: 设置某通道的过流保护开关, 并设置该通道为当前通道。

说明: 如果省略通道参数, 则默认指定当前通道为需要控制的通道。只有在普通模式时才能选择“CH1、CH2”, 只有在串联模式时才能选择“SER”, 只有在并联模式时才能选择“PARA”。

举例: :OUTPut:OCP:STATe CH1, ON

:OUTPut:OCP[:STATe]?

格式: :OUTPut:OCP[:STATe]? [CH1|CH2|CH3|SER|PARA]

功能: 查询某通道的过流保护开关。

说明: 如果省略通道参数, 则默认指定当前通道为需要查询的通道。返回“ON | OFF”

举例: :OUTPut:OCP:STATe? CH1

返回: ON

6. MEASure

:MEASure:ALL[:DC]?

格式: :MEASure:ALL[:DC]? [CH1|CH2|CH3|SER|PARA]

功能: 查询指定通道的输出端子上的实际电压、电流、功率值。

说明: 如果省略通道参数, 则默认指定当前通道为需要查询的通道。

举例: :MEASure:ALL? CH1

返回: 05.10,0.089,00.45

:MEASure[:VOLTage][:DC]?

格式: :MEASure[:VOLTage][:DC]? [CH1|CH2|CH3|SER|PARA]

功能: 查询指定通道的输出端子上的实际电压值。

说明: 如果省略通道参数, 则默认指定当前通道为需要查询的通道。

举例: :MEASure:VOLTage? CH1

返回: 05.10

:MEASure:CURRent[:DC]?

格式: :MEASure:CURRent[:DC]? [CH1|CH2|CH3|SER|PARA]

功能: 查询指定通道的输出端子上的实际电流值。

说明: 如果省略通道参数, 则默认指定当前通道为需要查询的通道。

举例: :MEASure:CURRent? CH1

返回: 0.089

:MEASure:POWER[:DC]?

格式: :MEASure:POWER[:DC]? [CH1|CH2|CH3|SER|PARA]

功能: 查询指定通道的输出端子上的实际功率值。

说明: 如果省略通道参数, 则默认指定当前通道为需要查询的通道。

举例: :MEASure:POWER? CH1

返回: 00.45

7. LISTout

:LISTout[:STATe]

格式: :LISTout[:STATe] {0|1|OFF|ON}

功能: 启动或停止当前通道的列表输出模式。

说明: 可通过“INSTrument”章节的指令设置和查询当前通道。

举例: :LISTout:STATe ON

:LISTout[:STATe]?

格式: :LISTout[:STATe]?

功能: 查询当前通道的列表输出模式的状态。

说明: 返回“状态(ON|PAUSED|ERROR|OFF), 剩余时间, 当前点, 终止点, 剩余循环次数, 终止状态(OFF|LAST)”。

举例: :LISTout[:STATE]?

返回: ON,1,0,7,2047,0,OFF

:LISTout:BASE

格式: :LISTout:BASE <StrN>, <Grpn>, <CycN>, {OFF|LAST}

功能: 设置当前通道的列表输出模式的基本参数。

说明: StrN: 输出起始组号, 范围 0-2047, StrN + Grpn 不能大于 2048。

Grpn: 输出组数, 范围 1-2048, StrN + Grpn 不能大于 2048。

CycN: 循环次数, 范围 1-99999。

OFF|LAST: 停止状态, 关闭输出或保持最后的输出状态。

注意: 列表输出模式处于运行状态时, 无法设置参数。

举例: :LISTout:BASE 1,100,1,OFF

:LISTout:BASE?

格式: :LISTout:BASE?

功能: 查询当前通道的列表输出模式的基本参数。

说明: 返回“ StrN, Grpn, CycN, {OFF|LAST}”

StrN: 输出起始组号, 范围 0-2047。

Grpn: 输出组数, 范围 1-2048。

CycN: 循环次数, 范围 1-99999。

OFF|LAST: 停止状态, 关闭输出或保持最后的输出状态。

举例: :LISTout:BASE?

返回: 1,100,1,OFF

:LISTout:PARAMeter

格式: :LISTout:PARAMeter <index>, <volt>, <curr>, <time>

功能: 设置当前通道的列表输出模式的组参数。

说明: index: 需要设置参数的组的组序号, 范围 0-2047。

volt: 该组的输出电压, 单位为伏。

curr: 该组的输出电流, 单位为安。

time: 该组的输出时长, 单位为秒。

注意: 列表输出模式处于运行状态时, 无法设置参数。

举例: :LISTout:PARAMeter 0, 10.00, 3.00, 1.5

:LISTout:PARAMeter?

格式: :LISTout:PARAMeter? <index> [,<count>]

功能: 查询当前通道的列表输出模式的组参数。

说明: index: 从此序号的组开始查询设置参数, 范围 0-2047。

count: 需要查询的组的个数, 范围 1-10。省略此参数时, 默认查询 1 组。

返回格式: 以数据块格式返回数据, 数据段为各组的序号、电压、电流、时间。如: “#2190,10.000,3.000,1.5;” , #219 表示数据段有 19 个数据—— “0,10.000,3.000,1.5;” , 数据内容显示组号 0, 电压 10.000 , 电流 3.000, 时间 1.5。

举例: :LISTout:PARAMeter? 0

返回: #2190,10.000,3.000,1.5;

:LISTout:TEMPlet:SElect

格式: :LISTout:TEMPlet:SElect {SINE|PULSE|RAMP|UP|DN|UPDN|RISE|FALL}

功能: 为当前通道的列表输出模式选择模板类型。

说明: SINE: 正弦模板;

PULSE: 脉冲模板;

RAMP: 斜坡模板;

UP: 阶梯上升模板;

DN: 阶梯下降模板;

UPDN: 上下阶梯模板;

RISE: 指数上升模板;

FALL: 指数下降模板。

举例: :LISTout:TEMPlet:SElect SINE

:LISTout:TEMPlet:SElect?

格式: :LISTout:TEMPlet:SElect?

功能: 查询当前通道的列表输出模式的模板类型。

说明: 返回{SINE|PULSE|RAMP|UP|DN|UPDN|RISE|FALL}。

举例: :LISTout:PARAMeter?

返回: SINE

:LISTout:TEMPlet:OBJect

格式: :LISTout:TEMPlet:OBJect {V|C}

功能: 设置当前通道的列表输出模式的模板的构建对象。

说明: V: 电压; C: 电流;

举例: :LISTout:TEMPlet:OBJect V

:LISTout:TEMPlet:OBJect?

格式: :LISTout:TEMPlet:OBJect? {V|C}

功能: 查询当前通道的列表输出模式的模板的构建对象。

说明: 返回{V|C}, V: 电压; C: 电流。

举例: :LISTout:TEMPlet:OBJect?

返回: V

:LISTout:TEMPlet:START

格式: :LISTout:TEMPlet:START < index>

功能: 设置当前通道的列表输出模式的模板的构建起始组号, 以此组为第一个模板构建点。

说明: index, 起始构建组, 范围 0 至 (2048 减去需要构建的组数)。

举例: :LISTout:TEMPlet:START 0

:LISTout:TEMPlet:START?

格式: :LISTout:TEMPlet:START?

功能: 查询当前通道的列表输出模式的模板的构建起始组号, 此组为第一个模板构建点。

说明: 返回<index>, 起始构建组, 范围 0 至 (2048 减去需要构建的组数)。

举例: :LISTout:TEMPlet:START?

返回: 0

:LISTout:TEMPlet:POINTS

格式: :LISTout:TEMPlet:POINTS <point>

功能: 设置当前通道的列表输出模式的模板的构建组数。

说明: point, 构建组数, 从起始组开始构建、共构建 point 指定的组数, 不改变其他组的设置值。脉冲模板最少构建 2 个组, 其他模板最少构建 10 个组。

举例: :LISTout:TEMPlet:POINTS 50

:LISTout:TEMPlet:POINTS?

格式: :LISTout:TEMPlet:POINTS?

功能: 查询当前通道的列表输出模式的模板的构建组数。

说明: 返回< point >, 构建组数。

举例: :LISTout:TEMPlet:POINTS?

返回: 50

:LISTout:TEMPlet:MAXValue

格式: :LISTout:TEMPlet:MAXValue {<value>|MINimum|MAXimum}

功能: 设置当前通道的列表输出模式的模板的构建最大值。

说明: value: 实型数值, 指定一个数值。

MINimum: 0。

MAXimum: 各通道的最大输出值。

举例: :LISTout:TEMPlet:MAXValue 5.55

:LISTout:TEMPlet:MAXValue?

格式: :LISTout:TEMPlet:MAXValue?

功能: 查询当前通道的列表输出模式的模板的构建最大值。

说明: 返回< value >, 实型数值。

举例: :LISTout:TEMPlet:MAXValue?

返回: 5.55

:LISTout:TEMPlet:MINValue

格式: :LISTout:TEMPlet:MINValue {<value>|MINimum|MAXimum}

功能: 设置当前通道的列表输出模式的模板的构建最小值。

说明: value: 实型数值, 指定一个数值。

MINimum: 0。

MAXimum: 各通道的最大输出值。

举例: :LISTout:TEMPlet:MINValue 1.11

:LISTout:TEMPlet:MINValue?

格式: :LISTout:TEMPlet:MINValue?

功能: 查询当前通道的列表输出模式的模板的构建最小值。

说明: 返回< value >, 实型数值。

举例: :LISTout:TEMPlet:MINValue?

返回: 1.11

:LISTout:TEMPlet:INTERval

格式: :LISTout:TEMPlet:INTERval <value>

功能: 设置当前通道的列表输出模式的模板的构建间隔时间。

说明: value: 浮点数值, 单位为秒, 指定组参数的持续时间, 范围: 0.1——9999.9。

举例: :LISTout:TEMPlet:INTERval 1.0

:LISTout:TEMPlet:INTERval?

格式: :LISTout:TEMPlet:INTERval?

功能: 查询当前通道的列表输出模式的模板的构建间隔时间。

说明: 返回< value >, 浮点数值, 单位为秒。

举例: :LISTout:TEMPlet:INTERval?

返回: 1.0

:LISTout:TEMPlet:INVErt

格式: :LISTout:TEMPlet:INVErt {0|1|OFF|ON}

功能: 设置当前通道的列表输出模式的模板的构建反相开关。

说明: 只有正弦、脉冲、斜坡模板具有反相功能, 在这些模板时才能使用此命令。

举例: :LISTout:TEMPlet:INVErt ON

:LISTout:TEMPlet:INVErt?

格式: :LISTout:TEMPlet:INVErt?

功能: 查询当前通道的列表输出模式的模板的构建反相开关。

说明: 返回{OFF|ON }。

举例: :LISTout:TEMPlet:INVErt?

返回: ON

:LISTout:TEMPlet:WIDTh

格式: :LISTout:TEMPlet:WIDTh <value>

功能: 设置当前通道的列表输出模式的脉冲模板的脉冲宽度。

说明: value: 浮点数值, 单位为秒, 指定脉冲宽度, 范围: 0.1—(周期-0.1)。只有在脉冲模板时才能使用此命令。

举例: :LISTout:TEMPlet:WIDTh 5.0

:LISTout:TEMPlet:WIDTh?

格式: :LISTout:TEMPlet:WIDTh?

功能: 查询当前通道的列表输出模式的脉冲模板的脉冲宽度。

说明: 返回<value>, 浮点数值, 单位为秒。

举例: :LISTout:TEMPlet:WIDTh?

返回: 5.0

:LISTout:TEMPlet:PERIod

格式: :LISTout:TEMPlet:PERIod <value>

功能: 设置当前通道的列表输出模式的脉冲模板的周期。

说明: value: 浮点数值, 单位为秒, 指定脉冲的周期, 范围: (脉冲宽度 + 0.1) ——9999.9。只有在脉冲模板时才能使用此命令。

举例: :LISTout:TEMPlet:PERIod 10.

:LISTout:TEMPlet:PERIod?

格式: :LISTout:TEMPlet:PERIod?

功能: 查询当前通道的列表输出模式的脉冲模板的周期。

说明: 返回<value>, 浮点数值, 单位为秒。

举例: :LISTout:TEMPlet:PERIod?

返回: 10.0

:LISTout:TEMPlet:SYMMetry

格式: :LISTout:TEMPlet:SYMMetry <value>

功能: 设置当前通道的列表输出模式的斜坡模板的对称性。

说明: value: 整型数值, 指定斜坡的对称性, 范围: 0——100。只有在斜坡模板时才能使用此命令。

举例: :LISTout:TEMPlet:SYMMetry 50

:LISTout:TEMPlet:SYMMetry?

格式: :LISTout:TEMPlet:SYMMetry?

功能: 查询当前通道的列表输出模式的斜坡模板的对称性。

说明: 返回<value>, 整型数值。

举例： :LISTout:TEMPlEt:SYMMetry?

返回： 50

:LISTout:TEMPlEt:EXPRate

格式： :LISTout:TEMPlEt:EXPRate <value>

功能： 设置当前通道的列表输出模式的指数模板的指数。

说明： value： 整型数值， 范围： 0——10。在指数上升模板时，使用此命令设置上升指数；在指数下降模板时，使用此命令设置下降指数；其他模板下不可使用此命令。

举例： :LISTout:TEMPlEt:EXPRate 5

:LISTout:TEMPlEt:EXPRate?

格式： :LISTout:TEMPlEt:EXPRate?

功能： 查询当前通道的列表输出模式的斜坡模板的对称性。

说明： 返回<value>， 整型数值。

举例： :LISTout:TEMPlEt:EXPRate?

返回： 5

:LISTout:TEMPlEt:CONSTRUCT

格式： :LISTout:TEMPlEt:CONSTRUCT

功能： 启动构建列表组参数。

说明： 仪器接收到此命令后，按已设置的模板参数开始构建列表输出的组参数。

举例： :LISTout:TEMPlEt:CONSTRUCT

8. DELAY

:DELAY[:STATe]

格式： :DELAY[:STATe] {0|1|OFF|ON}

功能： 启动或停止当前通道的延时器。

说明： 可通过“INSTrument”章节的指令设置和查询当前通道。

举例： :DELAY:STATe ON

:DELAY[:STATe]?

格式： :DELAY[:STATe]?

功能： 查询当前通道的延时器状态。

说明： 返回“运行状态(ON|OFF)， 剩余时间， 当前点， 终止点， 剩余循环次数， 停止状态(OFF|LAST|ON)”。

举例： :DELAY[:STATe]?

返回： ON,1,12,2047,0,OFF

:DELAY:START

格式： :DELAY:START <StrN>

功能：设置当前通道的延时器输出起始组号。

说明：StrN: 输出起始组号，整形数值，范围 0-2047，（起始组号 + 输出组数）不能大于 2048。

注意：延时器处于运行状态时，无法设置参数。

举例： :DELAY:START 0

:DELAY:START?

格式： :DELAY:START?

功能：查询当前通道的延时器输出起始组号。

说明：返回“StrN”，整形数值。

举例： :DELAY:START?

返回： 0

:DELAY:GROUPs

格式： :DELAY:GROUPs <Grpn>

功能：设置当前通道的延时器的输出组数。

说明：Grpn: 输出组数，整形数值，范围 1-2048，（起始组号 + 输出组数）不能大于 2048。

注意：延时器处于运行状态时，无法设置参数。

举例： :DELAY:GROUPs 100

:DELAY:GROUPs?

格式： :DELAY:GROUPs?

功能：查询当前通道的延时器的输出组数。

说明：返回“Grpn”，整形数值。

举例： :DELAY:GROUPs?

返回： 100

:DELAY:CYCLEs

格式： :DELAY:CYCLEs <CycN>

功能：设置当前通道的延时器的循环次数。

说明：CycN: 循环次数，整形数值，范围 1-99999。

注意：延时器处于运行状态时，无法设置参数。

举例： :DELAY:CYCLEs 1

:DELAY:CYCLEs?

格式： :DELAY:CYCLEs?

功能：查询当前通道的延时器的循环次数。

说明：返回“CycN: 循环次数，整形数值。

举例： :DELAY:CYCLEs?

返回： 1

:DELAY:ENDState

格式: **:DELAY:ENDState** { ON|OFF|LAST}

功能: 设置当前通道的定时器的停止状态。

说明: ON: 定时器停止时, 打开输出。

OFF: 定时器停止时, 关闭输出。

LAST: 定时器停止时, 保持最后的输出状态。

注意: 定时器处于运行状态时, 无法设置参数。

举例: :DELAY:ENDState OFF

:DELAY:ENDState?

格式: **:DELAY:ENDState?**

功能: 查询当前通道的定时器的停止状态。

说明: 返回 " { ON|OFF|LAST }"

ON: 定时器停止时, 打开输出。

OFF: 定时器停止时, 关闭输出。

LAST: 定时器停止时, 保持最后的输出状态。

举例: :DELAY:ENDState?

返回: OFF

:DELAY:STOP

格式: **:DELAY:STOP** {NONE|<V>>V|<C>>C|<P>>P} [,<value>]

功能: 设置当前通道的定时器的停止条件。

说明: 定时器运行时, 满足设置的停止条件后、定时器将自动停止。省略<value>参数时, 只设置判断条件、不改变原有判断值。

注意: 定时器处于运行状态时, 无法设置参数。

举例: :DELAY:STOP >V, 10.00

:DELAY:STOP?

格式: **:DELAY:STOP?**

功能: 查询当前通道的定时器的停止条件。

说明: 返回" (NONE|<V>>V|<C>>C|<P>>P [,<value>])", 返回的第一个参数为 "NONE" 时, 将没有第二个参数。

举例: :DELAY:STOP?

返回: >V,10.000

:DELAY:PARAMeter

格式: **:DELAY:PARAMeter** <index>, {ON|OFF}, <time>

功能: 设置当前通道的定时器的组参数。

说明: index: 需要设置参数的组的组序号, 范围 0-2047。

ON|OFF: 该组的输出状态。

time: 该组的运行时长, 单位为秒。

注意：延时器处于运行状态时，无法设置参数。

举例：:DELAY:PARAMeter 0, ON, 10

:DELAY:PARAMeter?

格式：**:DELAY:PARAMeter?** <index> [,<count>]

功能：查询当前通道的延时器的组参数。

说明：index：从此序号的组开始查询设置参数，范围 0-2047。

count：需要查询的组的个数，范围 1-10。省略此参数时，默认查询 1 组。

返回格式：以数据块格式返回数据，数据段为各组的序号、输出开关状态、时间。如：“#190,ON,1.0;”，#19 表示数据段有 9 个数据——“0,ON,1.0;”，数据内容显示组号 0，输出打开，时间 1.0 秒。

举例：:DELAY:PARAMeter? 0

返回：#190,ON,1.0;

:DELAY:GENerate:STAT

格式：**:DELAY:GENerate:STAT** < index>, <point>, {01P|10P}

功能：按状态生成模式自动生成延时器的组参数。

说明：仪器接收到此命令后，按状态生成模式生成延时器的组参数，未指定的组参数不受影响。

index：第一个点的组号，从此点开始生成；

point：需要生成的点数；

01P|10P：指定按 01 码或 10 码生成；

注意：延时器处于运行状态时，不可使用此命令。

举例：:DELAY:GENerate:STAT 0, 10, 01P

:DELAY:GENerate:FIX

格式：**:DELAY:GENerate:FIX**< index>, <point>, <time_on>, <time_off>

功能：按固定时间生成模式自动生成延时器的组参数。

说明：仪器接收到此命令后，按固定时间生成模式生成延时器的组参数，未指定的组参数不受影响。

index：第一个点的组号，从此点开始生成；

point：需要生成的点数；

time_on：输出状态为打开的组时，需要运行的时长；

time_off：输出状态为关闭的组时，需要运行的时长；

注意：延时器处于运行状态时，不可使用此命令。

举例：:DELAY:GENerate:FIX 0, 10, 5, 10

:DELAY:GENerate:INC

格式：**:DELAY:GENerate:INC** < index>, <point>, <time_base>, <time_step>

功能：按单调上升生成模式自动生成延时器的组参数。

说明：仪器接收到此命令后，按单调上升生成模式生成延时器的组参数，未指定的组参数不受影响。

index：第一个点的组号，从此点开始生成；

point：需要生成的点数；

time_base: 时间基值;

time_step: 步进值;

注意: 延时器处于运行状态时, 不可使用此命令。

举例: :DELAY:GENerate:INC 0, 10, 10, 2

:DELAY:GENerate:DEC

格式: **:DELAY:GENerate:DEC** < index>, <point>, <time_base>, <time_step>

功能: 按单调下降生成模式自动生成延时器的组参数。

说明: 仪器接收到此命令后, 按单调下降生成模式生成延时器的组参数, 未指定的组参数不受影响。

index: 第一个点的组号, 从此点开始生成;

point: 需要生成的点数;

time_base: 时间基值;

time_step: 步进值;

注意: 延时器处于运行状态时, 不可使用此命令。

举例: :DELAY:GENerate:DEC 0, 10, 100, 1

:DELAY:GENerate?

格式: **:DELAY:GENerate?**

功能: 查询当前通道的延时器的自动生成参数。

说明: 返回: "STAT, <index>, <point>, {0|1|10P}" 或
"FIX, <index>, <point>, <time_on>, <time_off>" 或
"{INC|DEC}, <index>, <point>, <time_base>, <time_step>"

举例: :DELAY:GENerate?

返回: DEC,0,10,100,1

9. MONItor

:MONItor[:STATe]

格式: **:MONItor[:STATe]** {0|1|OFF|ON}

功能: 启动或停止当前通道的监测器。

说明: 可通过 "INSTrument" 章节的指令设置和查询当前通道。

举例: :MONItor:STATe ON

:MONItor[:STATe]?

格式: **:MONItor[:STATe]?**

功能: 查询当前通道的监测器状态。

说明: 返回 "{ON|OFF}"。

举例: :MONItor:STATe?

返回: ON

:MONItor:VOLTage

格式: **:MONItor:VOLTage** {<V>>V|NONE} [,<volt>|MINimum|MAXimum]

功能: 设置当前通道的监测条件中的电压条件。

说明: 省略第二个参数时, 只设置判断条件、不改变原有的判断值。

<V: 启用电压判断, 当电压小于判断值时返回 “真”;

>V: 启用电压判断, 当电压大于判断值时返回 “真”;

NONE: 禁用电压判断, 不可以同时禁用电压判断、电流判断、功率判断;

<volt>: 实型数值, 电压判断值。

举例: :MONItor:VOLTage >V, 15.58

:MONItor:VOLTage?

格式: **:MONItor:VOLTage?**

功能: 查询当前通道的监测条件中的电压条件。

说明: 返回 “{<V>>V|NONE}, <volt>” ; <volt>: 实型数值, 电压判断值。

举例: :MONItor:VOLTage?

返回: >V,15.58

:MONItor:CURRent

格式: **:MONItor:CURRent** {<C>>C|NONE} [,<curr>|MINimum|MAXimum]

功能: 设置当前通道的监测条件中的电流条件。

说明: 省略第二个参数时, 只设置判断条件、不改变原有的判断值。

<C: 启用电流判断, 当电流小于判断值时返回 “真”;

>C: 启用电流判断, 当电流大于判断值时返回 “真”;

NONE: 禁用电流判断, 不可以同时禁用电压判断、电流判断、功率判断;

<curr>: 实型数值, 电流判断值。

举例: :MONItor:CURRent >C, 3.555

:MONItor:CURRent?

格式: **:MONItor:CURRent?**

功能: 查询当前通道的监测条件中的电流条件。

说明: 返回 “{<C>>C|NONE}, <curr>” ; <curr>: 实型数值, 电流判断值。

举例: :MONItor:CURRent?

返回: >C,3.555

:MONItor:POWER

格式: **:MONItor:POWER** {<P>>P|NONE} [,<watt>|MINimum|MAXimum]

功能: 设置当前通道的监测条件中的功率条件。

说明: 省略第二个参数时, 只设置判断条件、不改变原有的判断值。

<P: 启用功率判断, 当功率小于判断值时返回 “真”;

>P: 启用功率判断, 当功率大于判断值时返回 “真”;

NONE: 禁用功率判断, 不可以同时禁用电压判断、电流判断、功率判断;

<watt>: 实型数值, 功率判断值。

举例: :MONItor:POWER >P, 60.00

:MONItor:POWER?

格式: :MONItor:POWER?

功能: 查询当前通道的监测条件中的功率条件。

说明: 返回 “{<P|>P|NONE}, <watt>” ; <watt>: 实型数值, 功率判断值。

举例: :MONItor:POWER?

返回: >P,60.00

:MONItor:LOGic

格式: :MONItor:LOGic {1|2}, {AND|OR}

功能: 设置当前通道的监测条件中的逻辑符号。

说明: {1|2}: 1 指定设置第一个逻辑符号; 2 指定设置第二个逻辑符号;

AND: 设置指定的逻辑符号为逻辑 “与”;

OR: 设置指定的逻辑符号为逻辑 “或”。

举例: :MONItor:LOGic 1, AND

:MONItor:LOGic 2, OR

:MONItor:LOGic?

格式: :MONItor:LOGic? {1|2}

功能: 查询当前通道的监测条件中的逻辑符号。

说明: {1|2}: 1 指定查询第一个逻辑符号; 2 指定查询第二个逻辑符号;

返回: “{AND|OR} ”。

举例: :MONItor:LOGic? 1

返回: AND

:MONItor:LOGic? 2

返回: OR

:MONItor:STOPway

格式: :MONItor:STOPway {OUTOFF|MSG|BEEPER}, {ON|OFF}

功能: 设置当前通道的监测器的停止处理方式。

说明: OUTOFF: 设置是否关闭输出, ON: 关闭输出, OFF: 不关闭输出;

MSG: 设置是否弹出消息提示, ON: 弹出消息, OFF: 不弹出消息;

BEEPER: 设置是否发出蜂鸣声音, ON: 开启蜂鸣音, OFF: 关闭蜂鸣音。

举例: :MONItor:STOPway OUTOFF, ON 监测条件满足时, 关闭输出

:MONItor:STOPway MSG, OFF 监测条件满足时, 不弹出消息提示

:MONItor:STOPway BEEPER, ON 监测条件满足时, 蜂鸣器发出声音

:MONItor:STOPway?

格式: :MONItor:STOPway?

功能: 查询当前通道的监测器的停止处理方式。

说明: 返回: "OutputOff:{ON|OFF}, Msg:{ON|OFF}, Beep:{ON|OFF}" 。

举例: :MONItor:STOPway?

返回: OutputOff:ON,Msg:OFF,Beep:ON

10. TRIGger

:TRIGger:IN[:ENABLE]

格式: :TRIGger:IN[:ENABLE] {D0|D1|D2|D3},{0|1|OFF|ON}

功能: 启动或停止输入模式触发器; 如果触发器当前处于输出模式, 触发器会切换到输入模式。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

举例: :TRIGger:IN:ENABLE D0, OFF 停止 IO1 的触发器, IO1 作为输入模式

:TRIGger:IN:ENABLE D0, ON 启用 IO1 的触发器, IO1 作为输入模式

:TRIGger:IN[:ENABLE]?

格式: :TRIGger:IN[:ENABLE]? {D0|D1|D2|D3}

功能: 查询输入模式触发器开启状态。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

当相应 IO 处于输出模式时, 一定返回 "OFF" ;

当相应 IO 处于输入模式时, 根据触发器的开启状态返回 "OFF" 或 "ON" 。

举例: :TRIGger:IN:ENABLE? D0

返回: ON

:TRIGger:IN:SOURce

格式: :TRIGger:IN:SOURce {D0|D1|D2|D3}, {CH1|CH2|CH3|SER|PARA}[, {CH1|CH2|CH3|SER|PARA}] [, {CH1|CH2|CH3|SER|PARA}]

功能: 设置输入模式触发器的受控源, 当输入 IO 产生触发信号时, 受控源按设置的响应方式执行动作。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

CH1: 通道一; CH2: 通道二; CH3: 通道三; SER: 串联通道; PARA: 并联通道。

后 3 个参数中, CH1\CH2 不可与 SER\PARA 同时出现, SER 与 PARA 也不可同时出现。

举例: :TRIGger:IN:SOURce D0, CH1, CH2, CH3 设置 IO1 的受控源为 CH1 和 CH2 和 CH3

:TRIGger:IN:SOURce D0, CH1, CH3 设置 IO1 的受控源为 CH1 和 CH3

:TRIGger:IN:SOURce D0, CH3, SER 设置 IO1 的受控源为 CH3 和 SER(串联通道)

:TRIGger:IN:SOURce D0, PARA 设置 IO1 的受控源为 PARA(并联通道)

:TRIGger:IN:SOURce?

格式: :TRIGger:IN:SOURce? {D0|D1|D2|D3}

功能: 查询输入模式触发器的受控源。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

返回所有受控通道的组合, 如: “CH2,CH3”, 返回了两个受控通道, 通道二和通道三。

CH1: 通道一; CH2: 通道二; CH3: 通道三; SER: 串联通道; PARA: 并联通道。

举例: :TRIGger:IN:SOURce? d0

返回: CH2,CH3

:TRIGger:IN:TYPE

格式: :TRIGger:IN:TYPE {D0|D1|D2|D3},{RISE|FALL|HIGH|LOW}

功能: 设置输入模式触发器的触发信号的类型。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

RISE: 上升沿信号;

FALL: 下降沿信号;

HIGH: 高电平信号;

LOW: 低电平信号。

举例: :TRIGger:IN:TYPE D0, FALL 设置 IO1 的触发输入信号类型为下降沿

:TRIGger:IN:TYPE?

格式: :TRIGger:IN:TYPE? {D0|D1|D2|D3}

功能: 查询输入模式触发器的触发信号的类型。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

返回 “{RISE|FALL|HIGH|LOW}”。

RISE: 上升沿信号;

FALL: 下降沿信号;

HIGH: 高电平信号;

LOW: 低电平信号。

举例: :TRIGger:IN:TYPE? D0 查询 IO1 的触发输入信号的类型

返回: FALL 返回结果: 下降沿

:TRIGger:IN:SENSitivity

格式: :TRIGger:IN:SENSitivity {D0|D1|D2|D3},{LOW|MID|HIGH}

功能: 设置输入模式触发器的触发信号的灵敏度。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

LOW: 低灵敏度;

MID: 中灵敏度;

HIGH: 高灵敏度;

举例: :TRIGger:IN:SENSitivity D0, MID 设置 IO1 的触发输入信号灵敏度为中等灵敏度

:TRIGger:IN:SENSitivity?

格式: :TRIGger:IN:SENSitivity? {D0|D1|D2|D3}

功能: 查询输入模式触发器的触发信号的灵敏度。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

返回 "{LOW|MID|HIGH}"。

LOW: 低灵敏度;

MID: 中灵敏度;

HIGH: 高灵敏度;

举例: :TRIGger:IN:SENSitivity? D0 查询 IO1 的触发输入信号的灵敏度

返回: MID 返回结果: 中等灵敏度

:TRIGger:IN:RESPonse

格式: :TRIGger:IN:RESPonse {D0|D1|D2|D3},{ON|OFF|ALTER}

功能: 设置输入模式触发器的触发响应, 输入触发事情产生时、触发器执行触发响应。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

ON: 输入触发事情产生时、触发器打开受控通道的输出;

OFF: 输入触发事情产生时、触发器关闭受控通道的输出;

ALTER: 输入触发事情产生时、触发器翻转受控通道的输出。

举例: :TRIGger:IN:RESPonse D0, OFF 设置 IO1 的输入触发响应为关闭输出

:TRIGger:IN:RESPonse?

格式: :TRIGger:IN:RESPonse? {D0|D1|D2|D3}

功能: 查询输入模式触发器的触发响应。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

返回 "{ON|OFF|ALTER}", 代表输入触发事情产生时、触发器执行的触发响应。

ON: 输入触发事情产生时、触发器打开受控通道的输出;

OFF: 输入触发事情产生时、触发器关闭受控通道的输出;

ALTER: 输入触发事情产生时、触发器翻转受控通道的输出。

举例: :TRIGger:IN:RESPonse? D0 查询 IO1 的触发响应

返回: OFF 返回结果: 输入触发事情产生时、触发器关闭受控通道的输出

:TRIGger:OUT[:ENABLE]

格式: :TRIGger:OUT[:ENABLE] {D0|D1|D2|D3},{0|1|OFF|ON}

功能: 启动或停止输出模式触发器; 如果触发器当前处于输入模式, 触发器会切换到输出模式。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

举例: :TRIGger:OUT:ENABLE D0, OFF 停止 IO1 的触发器, IO1 作为输出模式

:TRIGger:OUT:ENABLE D0, ON 启用 IO1 的触发器, IO1 作为输出模式

:TRIGger:OUT[:ENABLE]?

格式: :TRIGger:OUT[:ENABLE]? {D0|D1|D2|D3}

功能: 查询输出模式触发器开启状态。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

当相应 IO 处于输入模式时, 一定返回 "OFF";

当相应 IO 处于输出模式时，根据触发器的开启状态返回“OFF”或“ON”。

举例： :TRIGger:OUT:ENABLE? D0

返回： ON

:TRIGger:OUT:SOURce

格式： :TRIGger:OUT:SOURce {D0|D1|D2|D3},{CH1|CH2|CH3|SER|PARA}

功能： 设置输出模式触发器的控制源，当控制源满足触发条件时，IO 按设置的输出信号输出响应的信号。

说明： D0|D1|D2|D3：选择输出 IO，D0-IO1，D1-IO2，D2-IO3，D3-IO4。

只能选择一个通道作为控制源。

CH1：通道一；CH2：通道二；CH3：通道三；SER：串联通道；PARA：并联通道。

举例： :TRIGger:OUT:SOURce D0, CH1 设置 CH1 为 IO1 的控制通道

:TRIGger:OUT:SOURce?

格式： :TRIGger:OUT:SOURce? {D0|D1|D2|D3},

功能： 查询输出模式触发器的控制源。

说明： D0|D1|D2|D3：选择输出 IO，D0-IO1，D1-IO2，D2-IO3，D3-IO4。

返回： {CH1|CH2|CH3|SER|PARA}。

CH1：通道一；CH2：通道二；CH3：通道三；SER：串联通道；PARA：并联通道。

举例： :TRIGger:OUT:SOURce? D0 查询 IO1 的控制通道

返回： CH1 返回结果： CH1 为 IO1 的控制通道

:TRIGger:OUT:CONDition

格式： :TRIGger:OUT:CONDition {D0|D1|D2|D3},{AUTO|OUTOFF|OUTON|>V|<V|=V|>C|<C|=C|>P|<P|=P},[<value>]

功能： 设置输出模式触发器的触发条件，当控制源满足触发条件时，IO 按设置的输出信号输出响应的信号。

说明： D0|D1|D2|D3：选择输出 IO，D0-IO1，D1-IO2，D2-IO3，D3-IO4。

AUTO：自动触发，此时仪器总是自动产生触发条件；不可带有<value>参数；

OUTOFF：控制源关闭输出时，产生触发条件；不可带有<value>参数；

OUTON：控制源打开输出时，产生触发条件；不可带有<value>参数；

>V：控制源电压大于判断值时，产生触发条件；不可省略<value>参数；

<V：控制源电压小于判断值时，产生触发条件；不可省略<value>参数；

=V：控制源电压等于判断值时，产生触发条件；不可省略<value>参数；

>C：控制源电流大于判断值时，产生触发条件；不可省略<value>参数；

<C：控制源电流小于判断值时，产生触发条件；不可省略<value>参数；

=C：控制源电流等于判断值时，产生触发条件；不可省略<value>参数；

>P：控制源功率大于判断值时，产生触发条件；不可省略<value>参数；

<P：控制源功率小于判断值时，产生触发条件；不可省略<value>参数；

=P：控制源功率等于判断值时，产生触发条件；不可省略<value>参数。

举例： :TRIGger:OUT:CONDition D0,>V,30.00 设置 IO1 的触发条件为：电压大于 30.00 伏

:TRIGger:OUT:CONDition

格式: :TRIGger:OUT:CONDition {D0|D1|D2|D3}

功能: 查询输出模式触发器的触发条件。

说明: D0|D1|D2|D3: 选择输出 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

返回: "{AUTO|OUTOFF|OUTON}>V|<V|=V|>C|<C|=C|>P|<P|=P} [,<value>]"。

返回 "AUTO|OUTOFF|OUTON" 时, 不带有<value>参数;

返回 ">V|<V|=V|>C|<C|=C|>P|<P|=P" 时, 带有<value>参数。

举例: :TRIGger:OUT:CONDition? D0 查询 IO1 的输出触发条件

返回: >V,30.00 返回结果: 触发条件为-电压大于 30.00 伏

:TRIGger:OUT:POLArity

格式: :TRIGger:OUT:POLArity {D0|D1|D2|D3},{POSItive|NEGAtive}

功能: 设置输出触发器的输出信号的极性。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

POSItive: 输出正极性信号;

NEGAtive: 输出负极性信号;

举例: :TRIGger:OUT:POLArity D0, POSItive 设置 IO1 的输出信号为正极性

:TRIGger:OUT:POLArity?

格式: :TRIGger:OUT:POLArity? {D0|D1|D2|D3}

功能: 查询输出触发器的输出信号的极性。

说明: D0|D1|D2|D3: 选择触发 IO, D0-IO1, D1-IO2, D2-IO3, D3-IO4。

返回: "{POSITIVE|NEGATIVE}"。

POSITIVE: 输出信号为正极性;

NEGATIVE: 输出信号为负极性。

举例: :TRIGger:OUT:POLArity? D0 查询 IO1 的输出信号极性

返回: POSITIVE 返回结果: IO1 的输出信号为正极性

11. PRESet

:PRESet#[[:APPLY]

格式: :PRESet#[[:APPLY]

功能: 将指定组的预设值参数应用到输出设置参数。

说明: #的可选值为 "1|2|3|4|5"、不可省略, 对应相应的预设值组。

举例: :PRESet1:APPLY

:PRESet#:SET:VOLTage

格式: :PRESet#:SET:VOLTage {CH1|CH2|CH3|SER|PARA}, {<volt>|MINimum|MAXimum}

功能: 设置预设值组的电压值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。

举例: :PRESet1:SET:VOLTage CH1, 5.00

:PRESet#:SET:VOLTage?

格式: :PRESet#:SET:VOLTage? {CH1|CH2|CH3|SER|PARA}

功能: 查询预设值组的电压值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。返回电压的实型数值。

举例: :PRESet1:SET:VOLTage? CH1

返回: 05.00

:PRESet#:SET:CURRent

格式: :PRESet#:SET:CURRent {CH1|CH2|CH3|SER|PARA}, {<curr>|MINimum|MAXimum}

功能: 设置预设值组的电流值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。

举例: :PRESet1:SET:CURRent CH1, 1.258

:PRESet#:SET:CURRent?

格式: :PRESet#:SET:CURRent? {CH1|CH2|CH3|SER|PARA}

功能: 查询预设值组的电流值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。返回电流的实型数值。

举例: :PRESet1:SET:CURRent? CH1

返回: 1.258

:PRESet#:SET:OVP

格式: :PRESet#:SET:OVP {CH1|CH2|CH3|SER|PARA}, {0|1|OFF|ON} [, {<volt>|MINimum|MAXimum}]

功能: 设置预设值组的过压保护开关、过压保护值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。省略{<volt>|MINimum|MAXimum}参数时, 只设置过压开关、不修改保护值。

举例: :PRESet1:SET:OVP CH1, OFF

:PRESet1:SET:OVP CH1, ON

:PRESet1:SET:OVP CH1, OFF, 20.00

:PRESet1:SET:OVP CH1, ON, 15.00

:PRESet#:SET:OVP?

格式: :PRESet#:SET:OVP? {CH1|CH2|CH3|SER|PARA}

功能: 查询预设值组的过压保护开关、过压保护值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。返回 “{OFF|ON}, <value>”。

OFF|ON: 过压保护的开关状态;

<value>: 实型数值, 过压保护值。

举例: :PRESet1:SET:OVP? CH1

返回: ON,15.000

:PRESet#:SET:OCP

格式: **:PRESet#:SET:OCP** {CH1|CH2|CH3|SER|PARA}, {0|1|OFF|ON} [,<curr>|MINimum|MAXimum]}

功能: 设置预设值组的过流保护开关、过流保护值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。省略<curr>|MINimum|MAXimum}参数时, 只设置过流开关、不修改保护值。

举例: :PRESet1:SET:OCP CH1, OFF

:PRESet1:SET:OCP CH1, ON

:PRESet1:SET:OCP CH1, OFF, 2.000

:PRESet1:SET:OCP CH1, ON, 1.500

:PRESet#:SET:OCP?

格式: **:PRESet#:SET:OCP?** {CH1|CH2|CH3|SER|PARA}

功能: 查询预设值组的过流保护开关、过流保护值。

说明: #的可选值为“1|2|3|4|5”、不可省略, 对应相应的预设值组。返回“{OFF|ON}, <value>”。

OFF|ON: 过流保护的开关状态;

<value>: 实型数值, 过流保护值。

举例: :PRESet1:SET:OCP? CH1

返回: ON,1.500

12. Memory

:MEMory[:STATe]:STORe

格式: **:MEMory[:STATe]:STORe** {STA|LST|DLY}, {1|2|3|4|5|6|7|8|9|10}

功能: 保存相应的设置数据到内部存储器的指定位置。

说明: “STA”为常规的状态设置数据, “LST”为列表输出的设置数据, “DLY”为延时器的设置数据。

举例: :MEM:STOR STA, 1

:MEMory[:STATe]:LOAD

格式: **:MEMory[:STATe]:LOAD** {STA|LST|DLY}, {1|2|3|4|5|6|7|8|9|10}

功能: 从内部存储器的指定位置加载设置数据。

说明: “STA”为常规的状态设置数据, “LST”为列表输出的设置数据, “DLY”为延时器的设置数据。

举例: :MEM:LOAD STA, 1

:MEMory[:STATe]:DELeTe

格式: **:MEMory[:STATe]:DELeTe** {STA|LST|DLY|REC}, {1|2|3|4|5|6|7|8|9|10}

功能: 删除内部存储器的指定位置保存的数据。

说明：“STA”为常规的状态设置数据，“LST”为列表输出的设置数据，“DLY”为定时器的设置数据，“REC”为录制器文件。

举例：:MEM:DEL STA, 1

:MEMory[:STATe]:VALid?

格式：:MEMory[:STATe]:VALid? {STA|LST|DLY|REC}, {1|2|3|4|5|6|7|8|9|10}. RETURN:(YES|NO)

功能：检查内部存储器的指定位置是否保存有数据。

说明：“STA”为常规的状态设置数据，“LST”为列表输出的设置数据，“DLY”为定时器的设置数据，“REC”为录制器文件。

举例：:MEM:VAL? STA, 1

返回：NO

13. MMemory

:MMEMory:DISK?

格式：:MMEMory:DISK? RETURN:(NULL | D:\)

功能：查询外部存储器的盘符。

说明：当外部存储器可用时，返回“D:\”，当外部存储器不可用时返回“NULL”。

举例：:MMEMory:DISK?

返回：D:\

:MMEMory:CDIRectory

格式：:MMEMory:CDIRectory "<D:\Dir_Name>"

功能：设置当前文件操作的操作路径。

说明：路径长度不要超过 200 字节。

举例：:MMEMory:CDIRectory "D:\Dir1"

:MMEMory:CDIRectory?

格式：:MMEMory:CDIRectory? RETURN:(NULL | D:\ Dir_Name)

功能：查询当前文件操作的操作路径。

说明：无。

举例：:MMEMory:CDIRectory?

返回：D:\ Dir_Name

:MMEMory:CATalog?

格式：:MMEMory:CATalog? RETURN:(NULL | dir1, dir2, file1)

功能：查询当前操作路径下可用的文件夹和文件。

说明：只返回文件名为纯英文字符的文件夹和文件，并且只识别以下文件：
状态设置文件(.sta)，列表设置文件(.lst)，定时器设置文件(.dly)。

举例: :MMEMory:CATalog?

返回: dir1,file.lst

:MMEMory:STORE

格式: :MMEMory:STORE "<FileName.sta>" 可用扩展名: (.sta | .lst | .dly)

功能: 保存相应的设置数据到当前操作路径下的指定文件。

说明: 文件扩展名决定了将要保存的设置数据的类型。

状态设置文件(.sta), 列表设置文件(.lst), 延时器设置文件(.dly)。

举例: :MMEMory:STORE "FileName.sta"

:MMEMory:LOAD

格式: :MMEMory:LOAD "<FileName.sta>" 可用扩展名: (.sta | .lst | .dly)

功能: 从当前操作路径下的指定文件加载相应的设置数据。

说明: 文件扩展名决定了将要加载的设置数据的类型。

状态设置文件(.sta), 列表设置文件(.lst), 延时器设置文件(.dly)。

举例: :MMEMory:LOAD "FileName.sta"

:MMEMory:DElete

格式: :MMEMory:DElete "<FileName.sta>" 可用扩展名: (.sta | .lst | .dly)

功能: 删除当前操作路径下的指定文件。

说明: 无。

举例: :MMEMory:DEL "FileName.sta"

14. STATus

:STATus:PRESet

格式: :STATus:PRESet

功能: 将所有可疑寄存器的使能寄存器和操作寄存器的使能寄存器恢复成默认状态。

说明: 无...

举例: 无...

:STATus:OPERation[:EVENT]?

格式: :STATus:OPERation[:EVENT]?

功能: 查询操作寄存器的事件寄存器。

说明: 无...

举例: 无...

:STATus:OPERation:CONDition?

格式: :STATus:OPERation:CONDition?

功能: 查询操作寄存器的条件寄存器。

说明: 无...

举例: 无...

:STATus:OPERation:ENABLE

格式: :STATus:OPERation:ENABLE

功能: 设置操作寄存器的使能寄存器。

说明: 无...

举例: 无...

:STATus:OPERation:ENABLE?

格式: :STATus:OPERation:ENABLE?

功能: 查询操作寄存器的使能寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable[:EVENT]?

格式: :STATus:QUEStionable[:EVENT]?

功能: 查询可疑寄存器的事件寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:CONDition?

格式: :STATus:QUEStionable:CONDition?

功能: 查询可疑寄存器的条件寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:ENABLE

格式: :STATus:QUEStionable:ENABLE

功能: 设置可疑寄存器的使能寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:ENABLE?

格式: :STATus:QUEStionable:ENABLE?

功能: 查询可疑寄存器的使能寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:INSTrument[:EVENT]?

格式: :STATus:QUEStionable:INSTrument[:EVENT]?

功能: 查询通道可疑寄存器的事件寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:INSTrument:CONDition?

格式: :STATus:QUEStionable:INSTrument:CONDition?

功能: 查询通道可疑寄存器的条件寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:INSTrument:ENABle

格式: :STATus:QUEStionable:INSTrument:ENABle

功能: 设置通道可疑寄存器的使能寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:INSTrument:ENABle?

格式: :STATus:QUEStionable:INSTrument:ENABle?

功能: 查询通道可疑寄存器的使能寄存器。

说明: 无...

举例: 无...

:STATus:QUEStionable:INSTrument:ISUMmary#[[:EVENT]?

格式: :STATus:QUEStionable:INSTrument:ISUMmary#[[:EVENT]? (1 <= # <=6, 可省略)

功能: 查询某通道可疑寄存器的事件寄存器。

说明: #号选择通道, "1" : 通道一; "2" : 通道二; "3" : 通道三; "5" : 串联通道; "6" : 并联通道。

省略此参数时, 则默认选择当前通道为指定的通道。

举例: :STATus:QUEStionable:INSTrument:ISUMmary1? 查询通道一的事件

:STATus:QUEStionable:INSTrument:ISUMmary? 查询当前通道的事件

:STATus:QUEStionable:INSTrument:ISUMmary#:CONDition?

格式: :STATus:QUEStionable:INSTrument:ISUMmary#:CONDition? (1 <= # <=6, 可省略)

功能: 查询某通道可疑寄存器的条件寄存器。

说明: #号选择通道, "1" : 通道一; "2" : 通道二; "3" : 通道三; "5" : 串联通道; "6" : 并联通道。

省略此参数时, 则默认选择当前通道为指定的通道。

举例: :STATus:QUEStionable:INSTrument:ISUMmary1:COND? 查询通道一的状态

:STATus:QUESTIONable:INSTRument:ISUMmary#:ENABLE

格式: :STATus:QUESTIONable:INSTRument:ISUMmary#:ENABLE (1 <= # <=6, 可省略)

功能: 设置某通道可疑寄存器的使能寄存器。

说明: #号选择通道, "1": 通道一; "2": 通道二; "3": 通道三; "5": 串联通道; "6": 并联通道。
省略此参数时, 则默认选择当前通道为指定的通道。

举例: 无...

:STATus:QUESTIONable:INSTRument:ISUMmary#:ENABLE?

格式: :STATus:QUESTIONable:INSTRument:ISUMmary#:ENABLE? (1 <= # <=6, 可省略)

功能: 查询某通道可疑寄存器的使能寄存器。

说明: #号选择通道, "1": 通道一; "2": 通道二; "3": 通道三; "5": 串联通道; "6": 并联通道。
省略此参数时, 则默认选择当前通道为指定的通道。

举例: 无...

15. SYSTEM

:SYSTEM:VERSION?

格式: :SYSTEM:VERSION?

功能: 查询当前系统采用的 SCPI 标准的版本号。

说明: 无...

举例: 无...

:SYSTEM:ERROR[:NEXT]?

格式: :SYSTEM:ERROR[:NEXT]?

功能: 读取一条错误消息。

说明: 无...

举例: 无...

:SYSTEM:ERROR:COUNT?

格式: :SYSTEM:ERROR:COUNT?

功能: 查询尚未读出的错误消息的条数。

说明: 无...

举例: 无...

:SYSTEM:BEEPper[:STATE]

格式: :SYSTEM:BEEPper[:STATE] {0|1|OFF|ON}

功能: 打开或关闭操作蜂鸣器提示音。

举例: :SYSTEM:BEEPper:STATE OFF 关闭蜂鸣提示音

:SYSTem:BEEPer[:STATe]?

格式: **:SYSTem:BEEPer[:STATe]?**

功能: 查询操作蜂鸣器提示音的开关状态。

说明: 返回 "{ON|OFF}" 。

举例: **:SYSTem:BEEPer:STATe?** 查询蜂鸣提示音开关状态
 返回: OFF 返回结果: 蜂鸣提示音为关闭状态

:SYSTem:BRIGhtness

格式: **:SYSTem:BRIGhtness** <value>

功能: 设置 LCD 的背光亮度。

说明: value, 整型数值, 范围 1 至 100。

举例: **:SYSTem:BRIGhtness 80** 设置 LCD 的背光亮度为 80%

:SYSTem:BRIGhtness?

格式: **:SYSTem:BRIGhtness?**

功能: 查询 LCD 的背光亮度。

说明: 返回< value >, 整型数值, 范围 1 至 100。

举例: **:SYSTem:BRIGhtness?** 查询 LCD 的背光亮度
 返回: 80 返回结果: LCD 的背光亮度 80%

:SYSTem:COMMunicate:LAN:APPLY

格式: **:SYSTem:COMMunicate:LAN:APPLY**

功能: 应用已设置的网络参数。

说明: 使用 **":SYSTem:COMMunicate:LAN:xxxx"** (xxxx 代表 LAN 下的其它网络设置命令)命令设置网络参数后, 相关设置参数不会立即生效、也不会被存入设备的非易失性存储器, 必须执行 **":SYSTem:COMMunicate:LAN:APPLY"** 命令使参数生效和永久存储。可以执行多条命令修改多个参数后, 共同执行一条 **":SYSTem:COMMunicate:LAN:APPLY"** 命令。

举例: **:SYSTem:COMMunicate:LAN:DHCP:STATe ON** 打开 DHCP 功能
 :SYSTem:COMMunicate:LAN:APPLY 应用已设置的参数

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

格式: **:SYSTem:COMMunicate:LAN:DHCP[:STATe]** {0|1|OFF|ON}

功能: 打开或关闭网络的 DHCP 功能。

说明: 使用此命令设置网络参数后, 设置参数不会立即生效、也不会被存入设备的非易失性存储器; 必须使用 **":SYSTem:COMMunicate:LAN:APPLY"** 命令使参数生效和永久存储。可以执行多条命令修改多个参数后, 共同执行一条 **":SYSTem:COMMunicate:LAN:APPLY"** 命令。

举例: **:SYSTem:COMMunicate:LAN:DHCP:STATe ON** 打开 DHCP 功能
 :SYSTem:COMMunicate:LAN:APPLY 应用已设置的参数

:SYSTem:COMMunicate:LAN:DHCP[:STATE]?

格式: **:SYSTem:COMMunicate:LAN:DHCP[:STATE]?**

功能: 查询网络的 DHCP 功能的开关状态。

说明: 在设置过网络参数后、并且在执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令以前, 使用此命令查询到的结果是设置的临时参数, 反应的是执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令时使用到的参数数据。否则, 使用此命令查询到的结果是设备正在运行使用的参数。
返回 “{ON|OFF}”。

举例: :SYSTem:COMMunicate:LAN:DHCP:STATE?

返回: ON

:SYSTem:COMMunicate:LAN:IPADdress

格式: **:SYSTem:COMMunicate:LAN:IPADdress** "x.x.x.x"

功能: 设置设备的网络 IP 地址。

说明: 使用此命令设置网络参数后, 设置参数不会立即生效、也不会被存入设备的非易失性存储器; 必须使用 “:SYSTem:COMMunicate:LAN:APPLY” 命令使参数生效和永久存储。可以执行多条命令修改多个参数后, 共同执行一条 “:SYSTem:COMMunicate:LAN:APPLY” 命令。

举例: :SYSTem:COMMunicate:LAN:IPADdress "192.168.10.142" 设置 IP 为 192.168.10.142
:SYSTem:COMMunicate:LAN:APPLY 应用已设置的参数

:SYSTem:COMMunicate:LAN:IPADdress?

格式: **:SYSTem:COMMunicate:LAN:IPADdress?**

功能: 查询设备的网络 IP 地址。

说明: 在设置过网络参数后、并且在执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令以前, 使用此命令查询到的结果是设置的临时参数, 反应的是执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令时使用到的参数数据。否则, 使用此命令查询到的结果是设备正在运行使用的参数。
返回类似: "192.168.10.142"。

举例: :SYSTem:COMMunicate:LAN:IPADdress?

返回: "192.168.10.142"

:SYSTem:COMMunicate:LAN:SMASK

格式: **:SYSTem:COMMunicate:LAN:SMASK** "x.x.x.x"

功能: 设置设备的网络子网掩码。

说明: 使用此命令设置网络参数后, 设置参数不会立即生效、也不会被存入设备的非易失性存储器; 必须使用 “:SYSTem:COMMunicate:LAN:APPLY” 命令使参数生效和永久存储。可以执行多条命令修改多个参数后, 共同执行一条 “:SYSTem:COMMunicate:LAN:APPLY” 命令。

举例: :SYSTem:COMMunicate:LAN:SMASK "255.255.255.0" 设置子网掩码为 255.255.255.0
:SYSTem:COMMunicate:LAN:APPLY 应用已设置的参数

:SYSTem:COMMunicate:LAN:SMASK?

格式: **:SYSTem:COMMunicate:LAN:SMASK?**

功能: 查询设备的网络子网掩码。

说明: 在设置过网络参数后、并且在执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令以前, 使用此命令查询到的结果是设置的临时参数, 反应的是执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令时使用到的参数数据。否则, 使用此命令查询到的结果是设备正在运行使用的参数。

返回类似: "255.255.255.0" 。

举例: :SYSTem:COMMunicate:LAN:SMASK?

返回: "255.255.255.0"

:SYSTem:COMMunicate:LAN:GATEway

格式: :SYSTem:COMMunicate:LAN:GATEway "x.x.x.x"

功能: 设置设备的网络网关。

说明: 使用此命令设置网络参数后, 设置参数不会立即生效、也不会被存入设备的非易失性存储器; 必须使用 “:SYSTem:COMMunicate:LAN:APPLY” 命令使参数生效和永久存储。可以执行多条命令修改多个参数后, 共同执行一条 “:SYSTem:COMMunicate:LAN:APPLY” 命令。

举例: :SYSTem:COMMunicate:LAN:GATEway "192.168.10.1" 设置子网掩码为 192.168.10.1
:SYSTem:COMMunicate:LAN:APPLY 应用已设置的参数

:SYSTem:COMMunicate:LAN:GATEway?

格式: :SYSTem:COMMunicate:LAN:GATEway?

功能: 查询设备的网络网关。

说明: 在设置过网络参数后、并且在执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令以前, 使用此命令查询到的结果是设置的临时参数, 反应的是执行 “:SYSTem:COMMunicate:LAN:APPLY” 命令时使用到的参数数据。否则, 使用此命令查询到的结果是设备正在运行使用的参数。

返回类似: "192.168.10.1" 。

举例: :SYSTem:COMMunicate:LAN:GATEway?

返回: "192.168.10.1"

:SYSTem:COMMunicate:RS232:BAUD

格式: :SYSTem:COMMunicate:RS232:BAUD {4800|7200|9600|14400|19200|38400|57600|115200|128000}

功能: 设置 RS232 接口的波特率。

举例: :SYSTem:COMMunicate:RS232:BAUD 9600 设置 RS232 的波特率为 9600

:SYSTem:COMMunicate:RS232:BAUD?

格式: :SYSTem:COMMunicate:RS232:BAUD?

功能: 查询 RS232 接口的波特率。

说明: 返回 "{4800|7200|9600|14400|19200|38400|57600|115200|128000}"

举例: :SYSTem:COMMunicate:RS232:BAUD? 查询 RS232 的波特率为 9600
返回: 9600 返回结果: 波特率为 9600

:SYSTEM:LOCK

格式: :SYSTEM:LOCK {0|1|OFF|ON}

功能: 打开或关闭键盘锁，锁定后长按键盘的“LOCK”键可解锁。

说明: 锁定时，键盘的“ON/OFF”键仍有效。

举例: :SYSTEM:LOCK 锁定键盘。

:SYSTEM:LOCK?

格式: :SYSTEM:LOCK?

功能: 查询键盘锁的状态。

说明: 返回“{ON|OFF}”。

举例: :SYSTEM:LOCK? 查询键盘锁定状态

 返回: OFF 返回结果: 键盘没有锁定

:SYSTEM:KLOCK:STATe

:SYSTEM:RWLock[:STATe]

格式: :SYSTEM:KLOCK:STATe [0|1|OFF|ON]

 :SYSTEM:RWLock[:STATe] [0|1|OFF|ON]

功能: 打开或关闭远程锁，锁定后只能通过远程命令解锁或重启设备解锁。

说明: 省略参数时，默认是加锁。锁定时，键盘的“ON/OFF”键仍有效。

举例: :SYSTEM:KLOCK:STAT ON 远程锁定

 :SYSTEM:RWLock 远程锁定

:SYSTEM:KLOCK:STATe?

:SYSTEM:RWLock[:STATe]?

格式: :SYSTEM:KLOCK:STATe?

 :SYSTEM:RWLock[:STATe]?

功能: 查询远程锁的状态。

说明: 返回“{ON|OFF}”。

举例: :SYSTEM:KLOCK:STAT? 查询远程锁的开启状态

 返回: OFF 返回结果: 远程锁没有打开