UDP5000多通道系列可编程数控电源

SCPI编程手册

REV 2

2025.11

UNI-T®

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

V1.03.1114

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。

目录

第一章 SCPI 指令简介.....4

1.1 指令格式 4

1.2 参数说明 4

1.3 简写规则 4

1.4 数据返回 4

第二章 通信接口及通信设置.....5

第三章 SCPI 指令详解.....5

3.1 指令列表 5

3.2 指令解析 9

第一章 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IEEE754标准中浮点运算规则、ISO646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。

1.1 指令格式

编程命令由关键字、分隔符、参数域和结束符等部分构成，以下面的命令为例：

:VOLTage:LEVel 25

VOLTage、LEVel 是命令关键字，“:”和空格为分隔符，“25”为参数（部分命令具有多个参数，参数间以”,”分隔），命令后的换行符或回车符为命令结束符。

为了方便描述，后续出现的各种符号采用如下约定：

•方括号“[]”表示可选的关键字或参数可省略。

•花括号“{}”表示命令串中的参数选项。

•尖括号“<>”表示必须提供一个数值参数。

•垂直线“|”用于分隔多个可选参数的选项。

•结束符：换行符<LF>（0x0A）或回车符<CR>（0x0D）

1.2 参数说明

编程参数的数据类型有数值型、字符型、布尔型等多种类型。不论哪种类型，均以ASCII码串表示，详见表。

符号	意义	范例
<NR1>	整形	123, 0123
<NR2>	定点浮点数	123., 12.3, 0.123, .123
<NR3>	浮点数	123, 12.3, 123E+3
<NRF>	可能是<NR1>、<NR2>或<NR3>	
<Boolean>	布尔数据	0|1|ON|OFF

1.3 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。

1.4 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。

1.5 多通道通讯

UDP5000 多通道系列对特定通道访问时，需要在指令前增加”**:ADDR#**”,其中#**取值[0,N-1],N 为通道数**

如:获取三通道中的通道 2 的输出电压

发送:ADDR2:MEAS:VOLT?

返回:5.0000

若**未指定通道编号**,则默认访问多通道电源的**通道 0**

第二章 通信接口及通信设置

详细说明请参考《UDP5000系列数控电源说明书》的“第三章”。

第三章 SCPI 指令详解

3.1 指令列表

测量指令	指令描述
[:ADDR#]:MEASure:VOLTage?	测量当前输出电压
[:ADDR#]:MEASure:CURRent?	测量当前输出电流
[:ADDR#]:MEASure:POWEr?	测量当前输出功率
[:ADDR#]:MEASure:ALL?	测量当前输出电压、电流和功率

并联测量指令	指令描述
[:ADDR#]:MEASure:PARALLEL:VOLTage?	测量并联电压
[:ADDR#]:MEASure:PARALLEL:CURRent?	测量并联总电流
[:ADDR#]:MEASure:PARALLEL:POWEr?	测量并联总功率
[:ADDR#]:MEASure:PARALLEL:ALL?	测量并联当前输出电压、电流和功率

串联测量指令	指令描述
[:ADDR#]:MEASure:SERIES:VOLTage?	测量串联总电压
[:ADDR#]:MEASure:SERIES:CURRent?	测量串联电流
[:ADDR#]:MEASure:SERIES:POWEr?	测量串联总功率
[:ADDR#]:MEASure:SERIES:ALL?	测量串联当前输出电压、电流和功率

输出设置指令	指令描述
[:ADDR#][:SOURce]:VOLTage	设置输出电压值
[:ADDR#][:SOURce]:CURRent	设置输出电流值
[:ADDR#][:SOURce]:VOLTage:PROTection	设置过压保护功能的保护值
[:ADDR#][:SOURce]:CURRent:PROTection	设置过流保护功能的保护值
[:ADDR#][:SOURce]:VOLTage:PROTection:STATe	打开或关闭过压保护功能
[:ADDR#][:SOURce:]CURRent:PROTection:STATe	打开或关闭过流保护功能
[:ADDR#][:SOURce:]VOLTage:PROTection:TRIPed?	查询过压保护 OVP 的触发状态
[:ADDR#][:SOURce:]VOLTage:PROTection:CLEar	清除过压保护 OVP 的触发状态
[:ADDR#][:SOURce:]CURRent:PROTection:TRIPed?	查询过流保护 OCP 的触发状态
[:ADDR#][:SOURce:]CURRent:PROTection:CLEar	清除过流保护 OCP 的触发状态
[:ADDR#][:SOURce:]VOLTage:STEP	设置设定电压步进值
[:ADDR#][:SOURce:]VOLTage:UP	向上步进电压设定值
[:ADDR#][:SOURce:]VOLTage:DOWN	向下步进电压设定值
[:ADDR#][:SOURce:]CURRent:STEP	设置设定电流步进值
[:ADDR#][:SOURce:]CURRent:UP	向上步进电流设定值
[:ADDR#][:SOURce:]CURRent:DOWN	向下步进电流设定值
[:ADDR#][:SOURce:]VOLTage:SLEW:RISing	设置电压的上升斜率
[:ADDR#][:SOURce:]VOLTage:SLEW:FALLing	设置电压的下降斜率
[:ADDR#][:SOURce:]CURRent:SLEW:RISing	设置电流的上升斜率
[:ADDR#][:SOURce:]CURRent:SLEW:FALLing	设置电流的下降斜率
[:ADDR#][:SOURce:]RESistance	设置电源输出内阻
[:ADDR#]:OUTPut:MODE	设置电源的斜率工作模式
[:ADDR#]:OUTPut[:STATe]	打开或关闭电源输出
[:ADDR#]:OUTPut:CVCC?	查询输出恒压恒流状态
[:ADDR#]:OUTPut:OVP:VALue	设置过压保护功能的保护值
[:ADDR#]:OUTPut:OCP:VALue	设置过流保护功能的保护值
[:ADDR#]:OUTPut:OVP[:STATe]	打开或关闭过压保护功能
[:ADDR#]:OUTPut:OCP[:STATe]	打开或关闭过流保护功能
[:ADDR#]:OUTPut:OVP:TRIPed?	查询过压保护 OVP 的触发状态
[:ADDR#]:OUTPut:OVP:CLEar	清除过压保护 OVP 的触发状态
[:ADDR#]:OUTPut:OCP:TRIPed?	查询过流保护 OCP 的触发状态
[:ADDR#]:OUTPut:OCP:CLEar	清除过压保护 OCP 的触发状态

列表模式指令	指令描述
[:ADDR#]:LISTout[:STATe]	设置 List 列表模式开关与查询列表模式状态
[:ADDR#]:LISTout:BASE	设置 List 列表模式的基础参数
[:ADDR#]:LISTout:PARAmeter	设置 List 列表模式的组参数

延时器指令	指令描述
[:ADDR#]:DELAY[:STATe]	启动/关闭延时器
[:ADDR#]:DELAY:STARt	设置延时器的起始组
[:ADDR#]:DELAY:GROUPs	设置延时器的组数
[:ADDR#]:DELAY:CYCLEs	设置延时器的循环次数
[:ADDR#]:DELAY:ENDState	设置延时器的停止状态
[:ADDR#]:DELAY:STOP	设置延时器的停止条件
[:ADDR#]:DELAY:PARAmeter	设置延时器的组参数

预设值指令	指令描述
[:ADDR#]:PRESet#[:APPLy]	应用 PRESET 预设值的一组里的参数
[:ADDR#]:PRESet#:SET:VOLTage	设置 PRESET 预设值里的一组电压参数
[:ADDR#]:PRESet#:SET:CURRent	设置 PRESET 预设值里的一组电流参数
[:ADDR#]:PRESet#:SET:OVP	设置 PRESET 预设值里的一组 OVP 参数
[:ADDR#]:PRESet#:SET:OCP	设置 PRESET 预设值里的一组 OCP 参数

系统指令	指令描述
[:ADDR#]:SYSTem:GROUps	设置电源用户组
[:ADDR#]:SYSTem:REMote	设置电源为远程控制模式
[:ADDR#]:SYSTem:LOCal	设置电源为面板控制模式
[:ADDR#]:SYSTem:LOCK	控制面板上锁
[:ADDR#]:SYSTem:UNLOCK	控制面板解锁
[:ADDR#]:SYSTem:BEEPer:TEST	测试蜂鸣器,响一声(蜂鸣器必须开启)
[:ADDR#]:SYSTem:BEEPer[:STATe]	打开或关闭操作蜂鸣器提示音
[:ADDR#]:SYSTem:BRIGhtness	设置电源的屏幕亮度
[:ADDR#]:SYSTem:LANGuage	设置系统语言

电源输出设置	指令描述
[:ADDR#]:SYSTem:POWER:POWERDown[:STATe]	设置电源掉电检测开关
[:ADDR#]:SYSTem:POWER:MODE	设置电源工作模式
[:ADDR#]:SYSTem:POWER:ID	设置电源 ID
[:ADDR#]:SYSTem:POWER:POWEROut	设置电源上电保持开关
[:ADDR#]:SYSTem:POWER:OVPDelay	设置电源 OVP 过压保护延迟
[:ADDR#]:SYSTem:POWER:OCPDelay	设置电源 OCP 过流保护延迟
[:ADDR#]:SYSTem:POWER:ELOAD[:STATe]	设置电源泄放负载开关

网络设置	指令描述
[:ADDR#]:SYSTem:COMMunicate:LAN:APPLy	保存并应用已设置的网络参数
[:ADDR#]:SYSTem:COMMunicate:LAN:DHCP[:STATe]	打开/关闭网络 DHCP 自动分配地址功能
[:ADDR#]:SYSTem:COMMunicate:LAN:IPADdress	设置/获取网络 LAN 口的 IP 地址
[:ADDR#]:SYSTem:COMMunicate:LAN:SMASK	设置/获取网络 LAN 口的子网掩码
[:ADDR#]:SYSTem:COMMunicate:LAN:GATEway	设置/获取网络 LAN 口的网关地址

SCPI 功能指令	指令描述
[:ADDR#]:SYSTem:ERRor[:NEXT]?	获取 SCPI 的错误代码和字符串
[:ADDR#]:SYSTem:ERRor:COUNt?	获取 SCPI 的错误代码和字符串
[:ADDR#]:SYSTem:VERSion?	获取 SCPI 的版本号
[:ADDR#]:STATus:QUEStionable[:EVENt]?	查询 SCPI 的 QUSE 状态事件寄存器
[:ADDR#]:STATus:QUEStionable:CONDition?	查询 SCPI 的 QUSE 状态寄存器,返回当前状态
[:ADDR#]:STATus:QUEStionable:ENABle	设置 SCPI 的 QUSE 状态事件使能寄存器
***IDN?**	查询主机仪器信息
:ADDR#:IDN?	查询仪器信息
***STB?**	查询主机状态字节事件寄存器
:ADDR#:STB?	查询特定通道状态字节事件寄存器
***SRE**	设置主机状态字节事件使能寄存器
:ADDR#:SRE	设置特定通道状态字节事件使能寄存器
***ESR?**	查询主机事件寄存器
:ADDR#:ESR?	查询特定通道事件寄存器
***ESE**	设置主机事件使能寄存器
:ADDR#:ESE	设置特定通道事件使能寄存器

3.2 指令解析

指令	[:ADDR#]:MEASure:VOLTage?
功能	测量当前输出电压
语法	:MEASure:VOLTage?
范例	:MEASure:VOLTage? -> <OutVoltage>

指令	:MEASure:CURRent?
功能	测量当前输出电流
语法	:MEASure:CURRent?
范例	:MEASure:CURRent? -> <OutCurrent>

指令	:MEASure:POWEr?
功能	测量当前输出功率
语法	:MEASure:POWEr?
范例	:MEASure:POWEr? -> <OutPower>

指令	:MEASure:ALL?
功能	测量当前输出电压、电流和功率
语法	:MEASure:ALL?
范例	:MEASure:ALL? -> <OutVoltage>,<OutCurrent>,<OutPower>

指令	:MEASure:PARALLEL:VOLTage?
功能	测量并联电压
语法	:MEASure:PARALLEL:VOLTage?
范例	:MEASure:PARALLEL:VOLTage? -> <OutVoltage>

指令	:MEASure:PARALLEL:CURRent?
功能	测量并联总电流
语法	:MEASure:PARALLEL:CURRent?
范例	:MEASure:PARALLEL:CURRent? -> <OutCurrent>

指令	:MEASure:PARALLEL:POWEr?
功能	测量并联总功率
语法	:MEASure:PARALLEL:POWEr?
范例	:MEASure:PARALLEL:POWEr? -> <OutPower>
指令	:MEASure:PARALLEL:ALL?
功能	测量并联当前输出电压、电流和功率
语法	:MEASure:PARALLEL:ALL?
范例	:MEASure:PARALLEL:ALL? -> <OutVoltage>,<OutCurrent>,<OutPower>

指令	:MEASure:SERIES:VOLTage?
功能	测量串联总电压
语法	:MEASure:SERIES:VOLTage?
范例	:MEASure:SERIES:VOLTage? -> <OutVoltage>

指令	:MEASure:SERIES:CURRent?
功能	测量串联电流
语法	:MEASure:SERIES:CURRent?
范例	:MEASure:SERIES:CURRent? -> <OutCurrent>

指令	:MEASure:SERIES:POWEr?
功能	测量串联总功率
语法	:MEASure:SERIES:POWEr?
范例	:MEASure:SERIES:POWEr? -> <OutPower>

指令	:MEASure:SERIES:ALL?
功能	测量串联当前输出电压、电流和功率
语法	:MEASure:SERIES:ALL?
范例	:MEASure:SERIES:ALL? -> <OutVoltage>,<OutCurrent>,<OutPower>

指令	[:SOURce]:VOLTage
功能	设置输出电压值
语法	[:SOURce]:VOLTage {<Value>|MINimum|MAXimum}
	[:SOURce]:VOLTage?
	<Value>=输出电压设定值,单位:V
范例	:VOLTage <Value>
	:VOLTage? -> <Value>

指令	[:SOURce]:VOLTage:PROTection
功能	设置过压保护功能保护值
语法	[:SOURce]:VOLTage:PROTection {<Value>|MINimum|MAXimum}
	[:SOURce]:VOLTage:PROTection?
	<Value>=过压保护功能保护值,单位:V
范例	:VOLTage:PROTection <Value>
	:VOLTage:PROTection? -> <Value>

指令	[:SOURce]:VOLTage:PROTection:STATe
功能	设置过压保护功能开关
语法	[:SOURce]:VOLTage:PROTection:STATe {<Boolean>}

	[:SOURce]:VOLTage:PROTection:STATe?
	<Boolean>=ON|OFF|0|1
范例	:VOLTage:PROTection:STATe <Boolean>
	:VOLTage:PROTection:STATe? -> <Boolean>

指令	[:SOURce:]VOLTage:PROTection:TRIPed?
功能	查询过压保护状态触发状态
语法	[:SOURce:]VOLTage:PROTection:TRIPed? <State>
	<State>=1:已触发 0:未触发
范例	:VOLTage:PROTection:TRIPed? -> <State>

指令	[:SOURce:]VOLTage:PROTection:CLEar
功能	清除过压保护状态触发状态
语法	[:SOURce:]VOLTage:PROTection:CLEar

指令	[:SOURce]:CURRent
功能	设置输出电流值
语法	[:SOURce]:CURRent {<Value>|MINimum|MAXimum}
	[:SOURce]:CURRent ?
	<Value>=输出电流设定值,单位:A
范例	:CURRent <Value>
	:CURRent? -> <Value>

指令	[:SOURce]:CURRent:PROTection
功能	设置过流保护功能保护值
语法	[:SOURce]:CURRent:PROTection {<Value>|MINimum|MAXimum}
	[:SOURce]:CURRent:PROTection?
	<Value>=过流保护功能保护值,单位:A
范例	:CURRent:PROTection <Value>
	:CURRent:PROTection? -> <Value>

指令	[:SOURce]:CURRent:PROTection:STATe
功能	设置过流保护功能开关
语法	[:SOURce]:CURRent:PROTection:STATe {<Boolean>}
	[:SOURce]:CURRent:PROTection:STATe?
	<Boolean>=ON|OFF|0|1
范例	:CURRent:PROTection:STATe <Boolean>
	:CURRent:PROTection:STATe? -> <Boolean>

指令	[:SOURce:]CURRent:PROTection:TRIPed?

功能	查询过流保护状态触发状态
语法	[:SOURce:]CURRent:PROTection:TRIPed? <State>
	<State>=1:已触发 0:未触发
范例	:CURRent:PROTection:TRIPed? -> <State>

指令	[:SOURce:]CURRent:PROTection:CLEar
功能	清除过流保护状态触发状态
语法	[:SOURce:]CURRent:PROTection:CLEar

指令	[:SOURce:]VOLTage:STEP
功能	设置电压步进值
语法	[:SOURce:]VOLTage:STEP {<Value>}
	[:SOURce:]VOLTage:STEP?
	<Value>=电压步进值,单位:V
范例	:VOLTage:STEP <Value>
	:VOLTage:STEP? -> <Value>

指令	[:SOURce:]VOLTage:UP
功能	按照步进值提高设置电压
语法	[:SOURce:]VOLTage:UP

指令	[:SOURce:]VOLTage:DOWN
功能	按照步进值降低设置电压
语法	[:SOURce:]VOLTage:DOWN

指令	[:SOURce:]CURRent:STEP
功能	设置电流步进值
语法	[:SOURce:]CURRent:STEP {<Value>}
	[:SOURce:]CURRent:STEP?
	<Value>=电流步进值,单位:A
范例	:CURRent:STEP <Value>
	:CURRent:STEP? -> <Value>

指令	[:SOURce:]CURRent:UP
功能	按照步进值提高设置电流
语法	[:SOURce:]CURRent:UP

指令	[:SOURce:]CURRent:DOWN
功能	按照步进值降低设置电流
语法	[:SOURce:]CURRent:DOWN

指令	[:SOURce:]VOLTage:SLEW:RISing
功能	设置电压的上升斜率
语法	[:SOURce]:VOLTage:SLEW:RISing {<Value>}
	[:SOURce]:VOLTage:SLEW:RISing?
	<Value>=电压斜率值,单位:V/s
范例	:VOLTage:SLEW:RISing <Value>
	:VOLTage:SLEW:RISing? -> <Value>

指令	VOLTage:SLEW:FALLing
功能	设置电压的下降斜率
语法	[:SOURce]:VOLTage:SLEW:FALLing {<Value>}
	[:SOURce]:VOLTage:SLEW:FALLing ?
	<Value>=电压斜率值,单位:V/s
范例	:VOLTage:SLEW:FALLing <Value>
	:VOLTage:SLEW:FALLing? -> <Value>

指令	[:SOURce:]CURRent:SLEW:RISing
功能	设置电流的上升斜率
语法	[:SOURce]:CURRent:SLEW:RISing {<Value>}
	[:SOURce]:CURRent:SLEW:RISing?
	<Value>=电流斜率值,单位:A/s
范例	:CURRent:SLEW:RISing <Value>
	:CURRent:SLEW:RISing? -> <Value>

指令	[:SOURce:]CURRent:SLEW:FALLing
功能	设置电流的下降斜率
语法	[:SOURce]:CURRent:SLEW:FALLing {<Value>}
	[:SOURce]:CURRent:SLEW:FALLing?
	<Value>=电流斜率值,单位:A/s
范例	:CURRent:SLEW:FALLing <Value>
	:CURRent:SLEW:FALLing? -> <Value>

指令	:OUTPut:MODE
功能	设置电源的斜率工作模式
语法	:OUTPut:MODE {<Mode>}
	:OUTPut:MODE?
	<Mode>:
	VHS:恒压 CV 高速优先;IHS:恒流 CC 高速优先
	VSR:恒压 CV 斜率优先;ISR:恒流 CC 斜率优先

范例	:OUTPut:MODE <Mode>
	:OUTPut:MODE? -> <Mode>

指令	[:SOURce:]RESistance
功能	设置电源输出内阻
语法	[:SOURce:]RESistance {<Value>}
	[:SOURce:]RESistance?
	<Value>=电源输出内阻,单位:欧姆Ω,范围:[0,1]Ω
	PS:电源输出内阻仅在电源模式为普通模式下才生效
范例	:RESistance <Value>
	:RESistance? -> <Value>

指令	:OUTPut[:STATe]
功能	打开或关闭电源输出
语法	:OUTPut[:STATe] {<Boolean>}
	:OUTPut[:STATe]?
	<Boolean>=ON|OFF|0|1
范例	:OUTPut ON
	:OUTPut? -> <Boolean>

指令	:OUTPut:CVCC?
功能	查询电源当前输出状态
语法	:OUTPut:CVCC? {CC|CV}
	CC=恒流模式 CV=恒压模式
范例	:OUTPut:CVCC? -> <CC|CV>

指令	:OUTPut:OVP:VALue
功能	设置过压保护功能保护值
语法	:OUTPut:OVP:VALue {<Value>|MINimum|MAXimum}
	:OUTPut:OVP:VALue?
	<Value>=过压保护功能保护值,单位:V
范例	:OUTPut:OVP:VALue <Value>
	:OUTPut:OVP:VALue? -> <Value>

指令	:OUTPut:OVP:TRIPed?
功能	查询过压保护状态触发状态
语法	:OUTPut:OVP:TRIPed? -><State>
	<State>=1:已触发 0:未触发
范例	:OUTPut:OVP:TRIPed? -> <State>

指令	:OUTPut:OVP:CLEar
功能	清除过压保护状态触发状态
语法	:OUTPut:OVP:CLEar

指令	:OUTPut:OCP:VALue
功能	设置过流保护功能保护值
语法	:OUTPut:OCP:VALue {<Value>|MINimum|MAXimum}
	:OUTPut:OCP:VALue?
	<Value>=过流保护功能保护值,单位:A
范例	:OUTPut:OCP:VALue <Value>
	:OUTPut:OCP:VALue? -> <Value>

指令	:OUTPut:OCP[:STATe]
功能	设置过流保护功能开关
语法	:OUTPut:OCP[:STATe] {<Boolean>}
	:OUTPut:OCP[:STATe]?
	<Boolean>=ON|OFF|0|1
范例	:OUTPut:OCP <Boolean>
	:OUTPut:OCP? -> <Boolean>

指令	:OUTPut:OCP:TRIPed?
功能	查询过流保护状态触发状态
语法	:OUTPut:OCP:TRIPed? -><State>
	<State>=1:已触发 0:未触发
范例	:OUTPut:OCP:TRIPed? -> <State>

指令	:OUTPut:OCP:CLEar
功能	清除过流保护状态触发状态
语法	:OUTPut:OCP:CLEar

指令	:LISTout[:STATe]
功能	设置 List 列表模式开关与查询列表模式状态
语法	:LISTout[:STATe] {<Boolean>}
	:LISTout[:STATe]? ->{<State>,<time>,<curGroup>,<endGroup>,<remainCycle>,<endState>}
	<Boolean>=ON|OFF|0|1
	<State>=ON|OFF|COMPLETED|PAUSED
	<time>=当前组剩余时间,单位:秒 s
	<curGroup>=当前组
	<endGroup>=终止组
	<remainCycle>=剩余次数
	<endState>=终止状态,OFF:关闭输出;LAST:维持输出
范例	:LISTout <Boolean>
	:LISTout? -> ON,0.1,000,009,00000,OFF

指令	:LISTout:BASE
功能	设置 List 列表模式的基础参数
语法	:LISTout:BASE {<Start>,<Groups>,<Cycle>,<endState>}
	:LISTout:BASE?->{<Start>,<Groups>,<Cycle>,<endState>}
	<Start>=起始组
	<Groups>=组数
	<Cycle>=循环次数,0 为无限循环
	<endState>=终止状态,OFF:关闭输出;LAST:维持输出
范例	:LISTout:BASE 0,10,3,OFF
	:LISTout:BASE?->0,10,3,OFF 设置组 0 为起始点,运行 10 组数据,循环 3 次,运行完毕后关闭电源输出

指令	:LISTout:PARAmeter
功能	设置 List 列表模式的组参数
语法	:LISTout:PARAmeter {<No>,<Volt>,<Curr>,<Time>}
	:LISTout:PARAmeter?->{<No>,<Volt>,<Curr>,<Time>}
	<No>=组序号
	<Volt>=电压,单位:V
	<Curr>=电流,单位:A
	<Time>=时长,单位:s
范例	:LISTout:PARAmeter <No>,<Volt>,<Curr>,<Time>
	:LISTout:PARAmeter? 0,2 ->#226000,10.000,12.000, 100.0;#226001,20.000,07.539, 2.0;
	#226 代表 26 占用 2 个字节,数据段有 26 个数据{000,10.000,12.000, 100.0;},表示序号 0 电压 10V,电流 12A,100 秒,具体格式可参考第一章 数据返回小节

指令	:DELAY[:STATe]
功能	设置延时器开关与查询延时器状态
语法	:DELAY[:STATe] {<Boolean>}
	:DELAY[:STATe]? ->{<State>,<time>,<curGroup>,<endGroup>,<remainCycle>,<endState>}
	<Boolean>=ON|OFF|0|1
	<State>=ON|OFF|FAILED|COMPLETED
	<time>=当前组剩余时间,单位:秒 s
	<curGroup>=当前组
	<endGroup>=终止组
	<remainCycle>=剩余次数
	<endState>=终止状态,OFF:关闭输出;LAST:维持输出;ON:打开输出
范例	:DELAY <Boolean>
	:DELAY? -> ON, 0.4,010,016,99999,OFF

指令	:DELAY:STARt
功能	设置延时器运行起始组
语法	:DELAY:STARt {<Start>}
	:DELAY:STARt?->{<Start>}
	<Start>=延时器运行起始组
范例	:DELAY:STARt 0
	:DELAY:STARt?->0

指令	:DELAY:GROUPs
功能	设置延时器运行组数
语法	:DELAY:GROUPs {<Groups>}
	:DELAY:GROUPs?->{<Groups>}
	<Groups>=延时器运行组数
范例	:DELAY:GROUPs 64
	:DELAY:GROUPs?->64

指令	:DELAY:CYCLEs
功能	设置延时器循环次数
语法	:DELAY:CYCLEs {<Cycles>}
	:DELAY:CYCLEs?->{<Cycles>}
	<Cycles>=延时器运行组数,0 代表无限循环
范例	:DELAY:CYCLEs 0
	:DELAY:CYCLEs?->0

指令	:DELAY:ENDState
功能	设置延时器的停止状态
语法	:DELAY:ENDState {<endState>}
	:DELAY:ENDState?->{<endState>}
	<endState>=终止状态,OFF:关闭输出;LAST:维持输出 ON:打开输出
范例	:DELAY:ENDState OFF
	:DELAY:ENDState?->OFF

指令	:DELAY:STOP
功能	设置延时器的停止条件
语法	:DELAY:STOP {NONE|<V|>V|<C|>C|<P|>P [,<Value>]}
	:DELAY:STOP?->{NONE|<V|>V|<C|>C|<P|>P [,<Value>]}
	<Value>=条件参数,可省略,省略时不改变数值
范例	:DELAY:STOP >V,15.000
	:DELAY:STOP?->V,15.000

指令	:DELAY:PARAmeter
功能	设置延时器的组参数
语法	:DELAY:PARAmeter {<No>,<Boolean>,<Time>}
	:DELAY:PARAmeter?->{<No>,<Boolean>,<Time>}
	<No>=组序号
	<Boolean>=输出状态,ON:打开输出;OFF:关闭输出
	<Time>=时长,单位 s,最小分辨率 1s
范例	:DELAY:PARAmeter 0,ON,1.0
	:DELAY:PARAmeter? 0,1 ->#215000,OFF 1.0;
	#215 代表 15 占用 2 个字节,数据段有 15 个数据{000,OFF 1.0;}, 表示序号 0 打开输出,1.0 秒,具体格式可参考第一章 数据返回小节

指令	:PRESet#[:APPLy]
功能	应用 PRESET 预设值的一组里的参数
语法	:PRESet#[:APPLy]　#取值[0,2]
范例	:PRESet2

指令	:PRESet#:SET:VOLTage
功能	设置 PRESET 预设值里的一组电压参数
语法	:PRESet#:SET:VOLTage {<Voltage>}
	:PRESet#:SET:VOLTage?->{<Voltage>}
	#取值[0,2];<Voltage>=电压
范例	:PRESet1:SET:VOLTage 5
	:PRESet1:SET:VOLTage?->5.000

指令	:PRESet#:SET:CURRent
功能	设置 PRESET 预设值里的一组电流参数
语法	:PRESet#:SET:CURRent {<Current>}
	:PRESet#:SET:CURRent?->{<Current>}
	#取值[0,2];<Current>=电流
范例	:PRESet2:SET:CURRent 5
	:PRESet2:SET:CURRent?->5.000

指令	:PRESet#:SET:OVP
功能	设置 PRESET 预设值里的一组 OVP 参数
语法	:PRESet#:SET:OVP {<Boolean>,<Voltage>}
	:PRESet#:SET:OVP?->{<Boolean>,<Voltage>}
	#取值[0,2];<Boolean>:ON|OFF|0|1,<Voltage>=电压
范例	:PRESet0:SET:OVP ON,40
	:PRESet0:SET:OVP?->ON,40.000

指令	:PRESet#:SET:OCP
功能	设置 PRESET 预设值里的一组 OCP 参数
语法	:PRESet#:SET:OCP {<Boolean>,<Current>}
	:PRESet#:SET:OCP?->{<Boolean>,<Current>}
	#取值[0,2];<Boolean>:ON|OFF|0|1,<Current>=电流
范例	:PRESet1:SET:OCP ON,5
	:PRESet1:SET:OCP?->ON,5.000

指令	:SYSTem:GROUps
功能	设置电源用户组
语法	:SYSTem:GROUps {<Groups>}
	:SYSTem:GROUps?->{<Groups>}
	<Groups>=电源用户组,取值[0,31]
范例	:SYSTem:GROUps 28
	:SYSTem:GROUps?->28

指令	:SYSTem:REMote
功能	设置电源为远程控制模式,锁定按键,手动解锁后则返回面板控制模式

指令	:SYSTem:LOCal
功能	设置电源为面板控制模式

指令	:SYSTem:LOCK
功能	设置控制面板上锁

指令	:SYSTem:UNLOCK
功能	设置控制面板解锁

指令	:SYSTem:BEEPer:TEST
功能	测试蜂鸣器,发送该指令蜂鸣器响一声,仅当开启蜂鸣器后有效

指令	:SYSTem:BEEPer[:STATe]
功能	打开或关闭蜂鸣器提示音
语法	:SYSTem:BEEPer[:STATe] {<Boolean>}
	:SYSTem:BEEPer[:STATe]?->{<Boolean>}
	<Boolean>=ON:打开提示音 OFF:关闭提示音
范例	:SYSTem:BEEPer:STATe ON
	:SYSTem:BEEPer:STATe?->ON

指令	:SYSTem:BRIGhtness
功能	设置显示屏的背光亮度
语法	:SYSTem:BRIGhtness {<Value>}
	:SYSTem:BRIGhtness?->{<Value>}
	<Value>=屏幕亮度,取值范围:20~100
范例	:SYSTem:BRIGhtness 50
	:SYSTem:BRIGhtness?->50

指令	:SYSTem:LANGuage
功能	设置系统语言
语法	:SYSTem:LANGuage {<Language>}
	:SYSTem:LANGuage?->{<Language>}
	<Language>=系统语言,EN:英文;CH:中文
范例	:SYSTem:LANGuage CH
	:SYSTem:LANGuage?->CH

指令	:SYSTem:POWER:MODE
功能	设置电源工作模式
语法	:SYSTem:POWER:MODE {<Mode>}
	:SYSTem:POWER:MODE?-> {<Mode>}
	<Mode>=Normal:普通模式 EXT_V:外部控制
	PARAMaster:并联主机;PARASlave:并联从机
	SERMaster:串联主机;SERSlave:串联从机
	SYNMaster:协同主机;SYNSlave:协同从机
范例	:SYSTem:POWER:MODE Normal
	:SYSTem:POWER:MODE?-> Normal

指令	:SYSTem:POWER:ID
功能	设置电源 ID
语法	:SYSTem:POWER:ID {<DeviceId>}
	:SYSTem:POWER:ID?-> {<DeviceId>}
	<DeviceId>:设备 ID, 串联模式:取值范围[1,1];并联模式:取值范围[1,2]
范例	:SYSTem:POWER:ID 1
	:SYSTem:POWER:ID? ->1

指令	:SYSTem:POWER:POWEROut
功能	设置电源上电保持开关
语法	:SYSTem:POWER:POWEROut {<P-Out>}
	:SYSTem:POWER:POWEROut?-> {<P-Out>}
	{<P-Out>}: KEEP:上电保持上次断电前输出状态 OFF:上电无输出
范例	:SYSTem:POWER:POWEROut KEEP
	:SYSTem:POWER:POWEROut?->KEEP

指令	:SYSTem:POWER:OVPDelay
功能	设置电源 OVP 过压保护触发延迟
语法	:SYSTem:POWER:OVPDelay {<delayTime>}
	:SYSTem:POWER:OVPDelay?-> {<delayTime>}
	{<delayTime>}:　保护触发延迟时间,单位:ms 毫秒
范例	:SYSTem:POWER:OVPDelay 10
	:SYSTem:POWER:OVPDelay?->10

指令	:SYSTem:POWER:OCPDelay
功能	设置电源 OCP 过流保护触发延迟
语法	:SYSTem:POWER:OCPDelay {<delayTime>}
	:SYSTem:POWER:OCPDelay?-> {<delayTime>}
	{<delayTime>}:　保护触发延迟时间,单位:ms 毫秒
范例	:SYSTem:POWER:OCPDelay 10
	:SYSTem:POWER:OCPDelay?->10

指令	:SYSTem:POWER:ELOAD[:STATe]
功能	设置电源泄放负载开关
语法	:SYSTem:POWER:ELOAD[:STATe] {<Boolean>}
	:SYSTem:POWER:ELOAD[:STATe]?->{<Boolean>}
	<Boolean>=ON:打开泄放负载 OFF:关闭泄放负载 AUTO:自动开关负载
范例	:SYSTem:POWER:ELOAD ON
	:SYSTem:POWER:ELOAD?->ON

指令	:SYSTem:POWER:POWERDown[:STATe]
功能	设置掉电检测功能开关
语法	:SYSTem:POWER:POWERDown[:STATe] {<Boolean>}
	:SYSTem:POWER:POWERDown[:STATe]?->{<Boolean>}
	<Boolean>=ON:打开掉电检测 OFF:关闭掉电检测
范例	:SYSTem:POWER:POWERDown ON
	:SYSTem:POWER:POWERDown?->ON

指令	:SYSTem:COMMunicate:LAN:APPLy
功能	保存并应用已设置的网络参数
	注意:网络参数设置后不会立即生效,必须执行该指令存储后生效
语法	:SYSTem:COMMunicate:LAN:APPLy

指令	:SYSTem:COMMunicate:LAN:DHCP[:STATe]
功能	打开或关闭网络的 DHCP 地址分配功能
语法	:SYSTem:COMMunicate:LAN:DHCP[:STATe] {<Boolean>}
	:SYSTem:COMMunicate:LAN:DHCP[:STATe]?->{<Boolean>}
	<Boolean>=ON:打开 DHCP OFF:关闭 DHCP
范例	:SYSTem:COMMunicate:LAN:DHCP ON
	:SYSTem:COMMunicate:LAN:APPLy
	:SYSTem:COMMunicate:LAN:DHCP?->ON

指令	:SYSTem:COMMunicate:LAN:IPADdress
功能	设置 LAN 网络接口的 IP 地址/获取当前 IP 地址
语法	:SYSTem:COMMunicate:LAN:IPADdress {<Address>}
	:SYSTem:COMMunicate:LAN:IPADdress?->{<Address>}
	<Address>=以"x.x.x.x"为格式的 IP 地址
范例	:SYSTem:COMMunicate:LAN:IPADdress "192.168.1.100"
	:SYSTem:COMMunicate:LAN:APPLy
	:SYSTem:COMMunicate:LAN:IPADdress?->192.168.1.100

指令	:SYSTem:COMMunicate:LAN:SMASK
功能	设置 LAN 网络接口的子网掩码/获取当前子网掩码
语法	:SYSTem:COMMunicate:LAN:SMASK {<Address>}
	:SYSTem:COMMunicate:LAN:SMASK?->{<Address>}
	<Address>=以"x.x.x.x"为格式的子网掩码
范例	:SYSTem:COMMunicate:LAN:SMASK "255.255.255.0"
	:SYSTem:COMMunicate:LAN:APPLy
	:SYSTem:COMMunicate:LAN:SMASK?->255.255.255.0

指令	:SYSTem:COMMunicate:LAN:GATEway
功能	设置 LAN 网络接口的网关地址/获取当前网关地址
语法	:SYSTem:COMMunicate:LAN:GATEway {<Address>}
	:SYSTem:COMMunicate:LAN:GATEway?->{<Address>}
	<Address>=以"x.x.x.x"为格式的网关地址
范例	:SYSTem:COMMunicate:LAN:GATEway "192.168.1.1"
	:SYSTem:COMMunicate:LAN:APPLy

	:SYSTem:COMMunicate:LAN:GATEway?->192.168.1.1

指令	:SYSTem:ERRor[:NEXT]?
功能	获取 SCPI 的错误
语法	:SYSTem:ERRor[:NEXT]?->{<errorNum>,<errorString>}
	<errorNum>=错误代码
	<errorString>=错误字符串
范例	:SYSTem:ERRor[:NEXT]?
	->0,"No error"

指令	:SYSTem:ERRor:COUNt?
功能	获取 SCPI 的错误代码和字符串
语法	:SYSTem:ERRor:COUNt?->{<errorCount>}
	<errorCount>=错误队列个数
范例	:SYSTem:ERRor:COUNt?
	->0

指令	:SYSTem:VERSion?
功能	获取 SCPI 的版本号
语法	:SYSTem:VERSion?->{<Version>}
	<Version>=SCPI 版本号
范例	:SYSTem:VERSion?
	1999

指令	*IDN?
功能	查询仪器信息
语法	*IDN? ->{<IDN1>,<IDN2>,<IDN3>,<IDN4>}
	<IDN1>:制造商 <IDN2>:仪器型号
	<IDN3>:仪器序列号 <软件版本>
范例	*IDN?
	->Unitrend,UDP5888,0000000000000,1.03.1114

指令	ADDR#:IDN?
功能	查询单独通道的仪器信息
语法	*IDN? ->{<IDN1>,<IDN2>,<IDN3>,<IDN4>}
	<IDN1>:制造商 <IDN2>:仪器型号
	<IDN3>:仪器序列号 <软件版本>
范例	ADDR1:IDN?
	->Unitrend,UDP5888_1,0000000000000,1.03.1114

状态字节寄存器 STB

状态位寄存器记录其他寄存器的触发状态,当寄存器&使能寄存器不为 0 时,STB 对应的位将被置位,该寄存器不会锁存,其随事件改变而改变。

Bit	十 进制		Definition	描述
0		R01	Not Used	
1	2	PRO	Protection Event Flag	
2	4	QMA	Error/Event queue message available	错误事件消息队列可用
3	8	QES	Questionable status	QUES 状态寄存器触发
4	16	MAV	Message Available	
5	32	ESR	Standard Event Status Register	标准状态寄存器触发
6	64	SRQ	Service Request	
7		OPS	Not Used	

***STB**

功能　查询状态字节事件寄存器,每次读取后自动清除标志位,以十进制返回

语法　*STB?

范例　*STB? -> 4

说明　若返回值为 4，则指示状态字节寄存器被置位 Bit 2；这意味着错误队列非空，也即产生了错误;

具体请查看状态字节寄存器 STB 的描述

***SRE**

功能　设置 SCPI 状态字节事件使能寄存器

若 SRE 寄存器&STB 寄存器不为 0,则 STB 寄存器的 SRQ 位为 1;

语法　*SRE <Value>

*SRE?-><Value>

<Value>=状态字节使能寄存器值

范例　*SRE 128

*SRE?->128

事件寄存器 ESR

事件寄存器记录 SCPI 及电源运行的错误事件,当事件发生后,寄存器会锁存事件,仅当查询该寄存器或发送*CLS 清除命令才会使该寄存器清零。

Bit	十进制		Definition	描述
0		OPC	Operation complete	默认为 0
1		Not Used		默认为 0
2	4	QER	Query Error	查询错误
3	8	DER	Device Dependent Error	设备发生错误,如机器自检异常
4	16	EER	Execution Error (e.g. range error)	指令执行错误
5	32	CER	Command error (e.g. syntax error)	指令错误,如语法错误
6		Not Used		默认为 0
7	128	PON	Power On	电源重新上电

指令	*STB?
	:ADDR#:STB?
功能	查询状态字节事件寄存器,每次读取后自动清除标志位,以十进制返回
语法	*STB?
范例	*STB? -> 4
	若返回值为 4，则指示状态字节寄存器被置位 Bit 2；这意味着错误队列非空，也即产生了错误; 具体请查看状态字节寄存器 STB 的描述

指令	*SRE
	:ADDR#:SRE?
功能	设置 SCPI 状态字节事件使能寄存器 若 SRE 寄存器&STB 寄存器不为 0,则 STB 寄存器的 SRQ 位为 1;
语法	*SRE <Value>
	*SRE?-><Value>
	<Value>=状态字节使能寄存器值
范例	*SRE 128　设置 Bit7,当有 PON(PowerOn)事件后,STB 的 ESR 置位
	*SRE?->128

查询状态寄存器 QUES

查询状态寄存器提供电源的工作状态信息,比如恒流、恒压工作模式、过温保护 OTP、过压过流保护等状态变化。

Bit	十进制		Definition	描述
0	1	CV	CV Mode	工作于恒压模式 CV
1	2	CC	CC Mode	工作于恒流模式 CC
2	4	FAN Error	FAN Error	散热风扇故障(FAN Error)
3	8	Not Used	Not Used	
4	16	OTP	Over Temperature	过温保护(OTP)
5	32	PFC_Hot	PFC_Hot	PFC 模组过热
6	64	MOS_Hot	MOS_Hot	功率管过热
7	128	OPP	Over Power	过功率保护
8	256	OSP	Over Sense	线损补偿过大
9	512	OVP	Over Voltage	过压保护
10	1024	OCP	Over Current	过流保护
11	2048	Font OCP	Font Over Current	前输出过流保护
12	4096	Volt Not Calib	No Voltage CalibPoints	电压未校准
13	8192	Curr Not Calib	No Current CalibPoints	电流未校准

指令	[:ADDR#]:STATus:QUEStionable[:EVENt]?
功能	查询 SCPI 的 QUSE 状态事件寄存器,每次读取后自动清除标志位,以十进制返回
语法	:STATus:QUEStionable[:EVENt]?
范例	:STATus:QUES? -> 512
说明	若返回值为 512，则指示 SCPI 的 QUSE 状态事件寄存器被置位 Bit 9, 这意味着发生了 OVP 过压保护事件; 具体请查看查询状态寄存器 QUES 的描述

指令	[:ADDR#]:STATus:QUEStionable:CONDition?
功能	查询 SCPI 的 QUSE 状态寄存器,返回当前状态
语法	:STATus:QUEStionable:CONDition?
范例	:STATus:QUES:COND? -> 1
说明	若返回值为 1，则指示 SCPI 的 QUSE 状态寄存器被置位 Bit 0；这意味着当前电源工作于 CV 恒压状态; 具体请查看查询状态寄存器 QUES 的描述

指令	[:ADDR#]:STATus:QUEStionable:ENABle
功能	设置 SCPI 的 QUSE 状态事件使能寄存器 若 QUES 状态事件寄存器&QUES 状态事件使能寄存器不为 0,则 STB 寄存器的 QUES 位为 1
语法	:STATus:QUEStionable:ENABle <Value>
	:STATus:QUEStionable:ENABle?-><Value>
	<Value>=QUES 时间使能寄存器值
范例	:STATus:QUEStionable:ENABle 512
	:STATus:QUEStionable:ENABle?->512
说明	设置当 OVP(Bit9 512)触发时,置位 STB 寄存器的 QUES 位

优利德

优利德科技(中国)股份有限公司

地址：广东省东莞市松山湖园区工业北一路 6 号

电话：(86-769) 8572 3888

邮编：523 808

http://www.uni-trend.com.cn

产品执行标准：GBT 4793.1-2007