

# 可編程高精度雙輸出直流電源供應器

PPH-1503D

操作指令

固緯料號：



ISO-9001 CERTIFIED MANUFACTURER

**GW INSTEK**

## 版权声明

这本手册包含所有权信息是受到版权保护的。版权属固纬电子实业股份有限公司拥有。手册的任何章节不得在固纬电子实业股份有限公司未授权之下做出任何之复制、重组或是翻译成其它之语言。

这本手册的所有信息在印制之前已经完全校正过。但因固纬电子实业股份有限公司不断地改善产品质量，固纬电子实业股份有限公司有权在未来修改产品之规格、特性及保养维修步骤，不必事前通知。

# 目 录

指令语法 .....	5
指令列表 .....	8
测量指令 .....	8
显示指令 .....	8
数据格式指令 .....	8
输出控制指令 .....	8
输出指令 .....	9
回读指令 .....	9
状态指令 .....	11
系统指令 .....	12
与系统相关指令 .....	12
IEEE488.2 共同命令 .....	14
指令详解 .....	15
测量指令 .....	15
显示指令 .....	18
数据格式指令 .....	19
输出控制指令 .....	20
输出指令 .....	24
回读指令 .....	27
状态指令 .....	37
系统指令 .....	42
与系统相关指令 .....	48
SCPI 状态模式 .....	51
事件寄存器 .....	52
允许寄存器 .....	53
状态字节寄存器 .....	53
标准事件寄存器 .....	54

---

状态字节寄存器命令 .....	55
标准事件寄存器命令 .....	57
其它状态寄存器命令 .....	58
错误.....	<b>59</b>
错误信息 .....	59
命令错误 .....	59

## 指令语法

PPH-1503D 中所用到的指令都满足 IEEE488.2 和 SCPI 标准。

### SCPI 语言简介

#### 命令格式

SCPI 是一种基于 ASCII 的仪器命令语言，专供测试测量仪器使用。SCPI 命令呈分级结构(树系统)，并分为不同的子系统，每个子系统以不同的根关键字区分。每个命令由一个根关键字和一个或多个层次关键字构成，关键字之间用冒号“:”分隔。命令关键字后面跟随参数，并且关键字和参数之间用“空格”分开。命令行后面添加问号“?”，表示查询功能。

例如：

```
:SYSTem:BEEPer:STATe {0|1|OFF|ON}
```

```
:SYSTem:BEEPer:STATe?
```

SYSTem 是命令的根关键字，BEEPer 和 STATe 分别是第二、第三级关键字，各级关键字之间用“:”分开。“{ }”括起的部分表示参数。命令关键字 SYSTem:BEEPer:STATe 和参数{0|1|OFF|ON}之间用“空格”分开。:SYSTem:BEEPer:STATe?表示查询。

此外，在一些带多个参数的命令中，参数之间通常用逗号“,”分隔，例如：:STATus:QUEue:ENABle (-110:-222, -220)

#### 符号说明

SCPI 命令中约定如下的符号，它们不是命令中的内容，但是通常用于辅助说明命令中的参数。

#### 1. 大括号 { }

大括号括住命令串中参数，例如：{OFF|ON}

## 2. 竖线 |

竖线分隔两个或多个可选的参数。使用命令时，每次只能选择其中一个参数，例如：{ON|OFF}只能选择 ON 或 OFF。

## 3. 方括号 [ ]

方括号中的内容表示可省略的关键字或参数，不管是否省略均被执行。例如：:OUTPut[:STATe] {ON|OFF}，其中[:STATe]可以省略。

## 4. 尖括号 < >

尖括号中的参数必须用一个有效值来替换。例如：

```
:DISPlay:CONTrast< brightness >
```

其中< brightness >要用一个数值来代替，如:DISPlay:CONTrast 1

## 参数类型

命令中有以下几种参数类型，参数的设置方法将根据参数类型而定。

### 1. 布尔型

参数取值为“OFF”、“ON”。例如：

DISPlay:FOCUs {ON|OFF}，“ON”表示开启焦点显示功能，

“OFF”表示关闭焦点

显示功能。

### 2. 连续整型

参数取值为连续的整数，例如：

:DISPlay:CONTrast <brightness>，<brightness>可取值的范围是1~3之间(包括1和3)的整数。

### 3. 连续实型

参数在有效值范围内和精度要求下，可以任意取值。例如：

CURRent {<current>|MINimum|MAXimum}，该命令用于设定当前操作通道的电流值，<current>参数可取当前通道电流设定范围内的任意实数。

### 4. 离散型

参数取值为所列举的值，例如：

\*RCL{0|1|2|3|4|5}，参数只能取 0、1、2、3、4 或 5。

### 5. ASCII 字符串

参数取值为 ASCII 字符的组合。例如：

:MODE <name>命令中，参数<name>是 ASCII 字符的组合。

### 命令缩写

按照 SCPI 语法，大多数命令以大小写字母混合的方式表示，大写字母表示命令的缩写。

所有命令对大小写不敏感，您可以全部采用大写或小写。请注意：若要使用命令缩写形式，必须完整输入命令格式中指定的大写字母，例如：

:MEASure: CURRent?

可缩写成：

:MEAS: CURR?

### 命令终止符

发送到函数发生器的命令串必须以一个<换行>字符结尾。可以将 IEEE-488 EOI 结束或标识信息当作<换行>字符，并用来代替<换行>字符。终止命令串<回车>后跟一个<换行>符也是可行的命令串终止。命令语法的终止总是将当前的 SCPI 命令路径复位到根级。

回读值以 0x0A 终止

## 指令列表

---

### 测量指令

:FETCh[1 2]?	15 页
:FETCh[1 2]:ARRay?	15 页
:READ[1 2] ?	15 页
:READ[1 2]:ARRay?	15 页
:MEASure[1 2] [<function>]?	16 页
:MEASure[1 2]:ARRay[<function>]?	16 页

### 显示指令

:DISPlay:ENABle <b>	18 页
:DISPlay:ENABle?	18 页
DISPlay::BRIGhtness <Nrf>	18 页
DISPlay::BRIGhtness?	18 页

### 数据格式指令

:FORMat[:DATA] <type>	19 页
:FORMat[:DATA]?	19 页
:FORMat:BORDer <name>	19 页
:FORMat:BORDer?	19 页

### 输出控制指令

:OUTPut[1 2][:STATe] <b>	20 页
:OUTPut[1 2][:STATe]?	20 页
BOTHOUTON	20 页
BOTHOUTOFF	20 页

:ROUTe:TERMinals {FRONT REAR}	20 页
:ROUTe:TERMinals?	21 页
:OUTPut[1 2]:RELAy <name>	22 页
:OUTPut[1 2]:RELAy?	22 页
:OUTPut[1 2]:OVP:STATe <b>	22 页
:OUTPut[1 2]:OVP:STATe?	22 页
:OUTPut[1 2]:OVP <value>	22 页
:OUTPut[1 2]:OVP?	23 页

## 输出指令

:[SOURce[1 2]]:CURRent[:LIMit][:VALue] <NRf>	24 页
:[SOURce[1 2]]:CURRent[:LIMit][:VALue]?	24 页
:[SOURce[1 2]]:CURRent[:LIMit]:TYPE <name>	24 页
:[SOURce[1 2]]:CURRent[:LIMit]:TYPE?	24 页
:[SOURce[1 2]]:CURRent[:LIMit]:STATe?	25 页
:[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude] <n>	25 页
:[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]?	25 页
:[SOURce[1 2]]:RESistance[:LEVel][:IMMediate][:AMPLitude] <n>	25 页
:[SOURce[1 2]]:RESistance[:LEVel][:IMMediate][:AMPLitude]?	25 页

## 回读指令

:SENSe[1 2]:FUNCTion <name>	27 页
:SENSe[1 2]:FUNCTion?	27 页
:SENSe[1 2]:NPLCycles <n>	27 页
:SENSe[1 2]:NPLCycles?	27 页
:SENSe[1 2]:AVERage <NRf>	28 页
:SENSe[1 2]:AVERage?	28 页
:SENSe[1 2]:CURRent[:DC]:RANGe[:UPPer] <n>	28 页

:SENSe2:CURRent[:DC]:RANGe[:UPPer] <n>	28 頁
:SENSe[1 2]:CURRent[:DC]:RANGe[:UPPer]?	29 頁
:SENSe[1 2]:CURRent[:DC]:RANGe:AUTO <b>	29 頁
:SENSe[1 2]:CURRent[:DC]:RANGe:AUTO?	29 頁
:SENSe[1 2]:PCURrent:AVERage <NRf>	29 頁
:SENSe[1 2]:PCURrent:AVERage?	30 頁
:SENSe[1 2]:PCURrent:MODE <name>	30 頁
:SENSe[1 2]:PCURrent:MODE?	30 頁
:SENSe[1 2]:PCURrent:TIME:AUTO	30 頁
:SENSe[1 2]:PCURrent:TIME:HIGH <NRf>	31 頁
:SENSe[1 2]:PCURrent:TIME:HIGH?	31 頁
:SENSe[1 2]:PCURrent:TIME:LOW <NRf>	31 頁
:SENSe[1 2]:PCURrent:TIME:LOW?	31 頁
:SENSe[1 2]:PCURrent:TIME:AVERage <NRf>	32 頁
:SENSe[1 2]:PCURrent:TIME:AVERage?	32 頁
:SENSe[1 2]:PCURrent:SYNChronize[:STATe] <b>	32 頁
:SENSe[1 2]:PCURrent:SYNChronize[:STATe]?	33 頁
:SENSe[1 2]:PCURrent:SYNChronize:DELay <NRf>	33 頁
:SENSe[1 2]:PCURrent:SYNChronize:DELay?	33 頁
:SENSe[1 2]:PCURrent:SYNChronize:TLEVel<NRf>	33 頁
:SENSe[1 2]:PCURrent:SYNChronize:TLEVel?	33 頁
:SENSe[1 2]:LINTegration:TIME <NRf>	34 頁
:SENSe[1 2]:LINTegration:TIME?	34 頁
:SENSe[1 2]:LINTegration:TIME:AUTO	34 頁
:SENSe[1 2]:LINTegration:TLEVel <NRf>	34 頁
:SENSe[1 2]:LINTegration:TLEVel?	34 頁
:SENSe[1 2]:LINTegration:TEDGE <name>	35 頁
:SENSe[1 2]:LINTegration:TEDGE?	35 頁

:SENSe[1 2]:LINTegration:TimeOUT <NRF>	35 页
:SENSe[1 2]:LINTegration:TimeOUT?	35 页
:SENSe[1 2]:LINTegration:SEARCh <b>	36 页
:SENSe[1 2]:LINTegration:SEARCh?	36 页
:SENSe[1 2]:LINTegration:FAST <b>	36 页
:SENSe[1 2]:LINTegration:FAST?	36 页

## 状态指令

:STATus:PRESet	37 页
:STATus:OPERation[:EVENT]?	37 页
:STATus:OPERation:CONDition?	37 页
:STATus:OPERation:ENABLE <NRF>	37 页
:STATus:OPERation:ENABLE?	38 页
:STATus:MEASurement[:EVENT]?	38 页
:STATus:MEASurement:ENABLE <NRF>	38 页
:STATus:MEASurement:ENABLE?	38 页
:STATus:MEASurement:CONDition?	39 页
:STATus:QUESTionable[:EVENT]?	39 页
:STATus:QUESTionable:CONDition?	39 页
:STATus:QUESTionable:ENABLE <NRF>	39 页
:STATus:QUESTionable:ENABLE?	39 页
:STATus:QUEue[:NEXT]?	40 页
:STATus:QUEue:ENABLE <list>	40 页
:STATus:QUEue:ENABLE?	40 页
:STATus:QUEue:DISable <list>	40 页
:STATus:QUEue:DISable?	41 页
:STATus:QUEue:CLEar	41 页

## 系统指令

:SYSTem:VERSion?	42 页
:SYSTem:ERRor?	42 页
:SYSTem:CLEar	42 页
:SYSTem:LFRequency?	42 页
:SYSTem:POSetup <name>	42 页
:SYSTem:POSetup?	43 页
:SYSTem:COMMunicate:LAN:DHCP[:STATe] <b>	43 页
:SYSTem:COMMunicate:LAN:DHCP[:STATe]?	44 页
:SYSTem:COMMunicate:LAN:IPADdress <IP 地址>	44 页
:SYSTem:COMMunicate:LAN:IPADdress?	44 页
:SYSTem:COMMunicate:LAN:SMASk <掩码>	44 页
:SYSTem:COMMunicate:LAN:SMASk?	45 页
:SYSTem:COMMunicate:LAN:GATEway <IP 地址>	45 页
:SYSTem:COMMunicate:LAN:GATEway?	45 页
:SYSTem:COMMunicate:LAN:DNS <地址>	45 页
:SYSTem:COMMunicate:LAN:DNS?	46 页
:SYSTem:COMMunicate:LAN:MANualip[:STATe] <b>	46 页
:SYSTem:COMMunicate:LAN:MANualip[:STATe]?	46 页
:SYSTem:COMMunicate:LAN:APPLY	46 页
:SYSTem:REMote	47 页
:SYSTem:BEEPer:STATe {0 1 OFF ON}	47 页
:SYSTem:BEEPer:STATe?	47 页
:SYSTem:LOCal	47 页

## 与系统相关指令

*IDN?	48 页
*RST	48 页

---

*TST?	48 页
*WAI	49 页
*TRG	49 页
*SAV <NRf>	49 页
*RCL <NRf>	49 页

**IEEE488.2 共同命令**

*SRE <允许值>	55 页
*SRE?	56 页
*STB?	56 页
*ESE<允许值>	57 页
*ESE?	57 页
*ESR?	57 页
*CLS	58 页
*OPC	58 页
*OPC?	58 页

## 指令详解

---

说明：命令中[1]代表通道 1，2 代表通道 2；对通道 1 执行命令时可以省略 1，对通道 2 执行命令时必须加 2。

### 测量指令

指令       :FETCh[1|2]?

功能        返回上次回读值

响应时间   最大值 16ms

---

例子        :FETCh2?

返回通道 2 上次回读值

指令        :FETCh:ARRay[1|2]?

功能        返回上次数组读取值

响应时间   最大值 16ms

---

例子        :FETCh:ARRay2?

返回通道 2 上次数组读取值

指令        :READ[1|2]?

功能        触发一个新的读取动作并返回读取值

响应时间   最大值 32ms

---

例子        :READ2?

触发通道 2 一个新的读取动作并返回读取值

指令        :READ:ARRay[1|2]?

功能        触发一个新的读取数组并将它们返回

响应时间 最大值 32ms

---

例子 :READ:ARRay2?  
触发通道 2 一个新的读取数组并将它们返回

指令 :MEASure[1|2] [[:<function>]?

功能 在指定功能函数中执行 “:READ?”

说明 function CURRent[:DC]: 测量电流  
VOLTage[:DC]: 测量电压  
PCURrent: 测量脉冲电流  
DVMeter: 测量 DVM 输入  
LINTegration: 长积分电流测量

在脉冲和长积分测量中，无脉冲的情况下，测试响应时间为溢出时间。

响应时间 最大值 32ms

---

例子 :MEASure2:CURRent?  
指定为通道 2 电流测量模式并读取返回值

指令 :MEASure[1|2]:ARRay[:<function>]?

功能 在指定功能函数中执行 “:READ:ARRay?”

说明 function CURRent[:DC]: 测量电流  
VOLTage[:DC]: 测量电压  
PCURrent: 测量脉冲电流  
DVMeter: 测量 DVM 输入  
LINTegration: 长积分电流测量

在脉冲和长积分测量中，无脉冲的情况下，测试响应时间为溢出时间。

响应时间 最大值 32ms

---

---

例子        :MEASure2:ARRay:PCURrent?  
             指定通道 2 为脉冲电流测量数组模式，并读取返回值

## 显示指令

指令	:DISPlay:ENABle <b>
功能	使能或禁止面板(LCD)显示
说明	b            0/OFF: 禁止面板显示 1/ON: 打开面板显示

---

例子        :DISPlay:ENABle ON  
              打开面板(LCD)显示

指令	:DISPlay:ENABle?
功能	查询面板显示的状态

---

例子        :DISPlay:ENABle?  
              查询面板显示的状态

指令	DISPlay: BRIGhtness < NRf >
功能	屏幕背光亮度设定
说明	NRf        0.33~0.00: 弱 0.66~0.34: 中 1.00~0.67: 强

---

例子        DISPlay: BRIGhtness 0.33  
              设定屏幕背光亮度为弱光

指令	DISPlay: BRIGhtness?
----	----------------------

---

说明        查询面板显示的亮度

---

例子        DISPlay: BRIGhtness?  
              查询面板显示的亮度

## 数据格式指令

指令 :FORMat[:DATA] <type>

功能 设定数据的格式

说明 type     ASCIi: ASCII 格式  
          SREal: IEEE754 单精度格式  
          DREal: IEEE754 双精度格式

---

例子 :FORMat:DATA SREal  
      设定数据格式为 IEEE754 单精度格式

指令 :FORMat[:DATA]?

功能 查询数据的格式

---

例子 :FORMat:DATA?  
      查询数据的格式

指令 :FORMat:BORDER <name>

功能 设定字节顺序

说明 name     NORMal: 正常二进制字节顺序  
          SWAPped: 反序的二进制字节顺序

---

例子 :FORMat:BORDER NORMal  
      设置数据格式为正常二进制字节顺序

指令 :FORMat:BORDER?

功能 查询数据格式的字节顺序

---

例子 :FORMat:BORDER?  
      查询数据格式的字节顺序

## 输出控制指令

指令 :OUTPut[1 | 2][:STATe] <b>

功能 输出的打开与关断

说明 b 0/OFF: 关闭输出  
1/ON: 打开输出

---

例子 :OUTPut:STATe ON  
打开 CH1 输出

指令 :OUTPut[1 | 2][:STATe]?

功能 查询输出状态

---

例子 :OUTPut:STATe?  
查询 CH1 输出状态

指令 BOTHOUTON

功能 打开双通道输出

---

例子 BOTHOUTON  
打开双通道输出

指令 BOTHOUTOFF

功能 关闭双通道输出

---

例子 BOTHOUTOFF  
关闭双通道输出

指令 :ROUTe:TERMinals {FRONt | REAR}

---

功能	前后面板输出的切换
说明	FRONt: 指定为前面板输出 REAR: 指定为后面板输出

---

例子	:ROUte:TERMinals FRONt 指定为前面板输出
----	------------------------------------

指令	:ROUte:TERMinals?
----	-------------------

功能	查询面板输出状态
----	----------

---

例子	:ROUte:TERMinals? 查询面板输出状态
----	-------------------------------

指令 :OUTPut[1|2]:RELAy <name>  
功能 关断或打开外部继电器控制电路  
说明 name ZERO: 关断  
ONE: 打开

---

例子 :OUTPut2:RELAy ONE  
打开通道 2 外部继电器控制电路

指令 :OUTPut[1|2]:RELAy?  
功能 查询继电器电路的状态

---

例子 :OUTPut2:RELAy?  
查询通道 2 继电器电路的状态

指令 :OUTPut[1|2]:OVP:STATe <b>  
功能 设定 OVP 保护状态  
说明 b 0/OFF: 关断 OVP 保护功能  
1/ON: 打开 OVP 保护功能

---

例子 :OUTPut2:OVP:STATe ON  
打开通道 2 OVP 保护功能

指令 :OUTPut[1|2]:OVP:STATe?  
功能 查询 OVP 保护功能状态

---

例子 :OUTPut2:OVP:STATe?  
查询通道 2 OVP 保护功能状态

指令 :OUTPut[1|2]:OVP <value>  
功能 设定 OVP 保护启动电压

---

说明	value	1.00-15.20
----	-------	------------

---

例子	:OUTPut2:OVP 10.05	设定通道 2OVP 保护启动电压为 10.05V
----	--------------------	--------------------------

---

指令	:OUTPut[1 2]:OVP?	
功能	查询 OVP 保护启动电压	

---

例子	:OUTPut2:OVP?	查询通道 2OVP 保护启动电压
----	---------------	------------------

---

## 输出指令

指令 :[SOURce[1|2]]:CURRent[:LIMit][:VALue] <NRf>

功能 设定电流限流电点的值

NRf 0.0000-5.0000

---

例子 :SOURce2:CURRent 1.0005  
设定限通道 2 通道流电为 1.0005A

指令 :[SOURce[1|2]]:CURRent[:LIMit][:VALue]?

功能 查询电流限流电点的值

---

例子 :SOURce2:CURRent?  
查询通道 2 电流限流电点的值

指令 :[SOURce[1|2]]:CURRent[:LIMit]:TYPE <name>

功能 设定电流的限流模式

说明 name LIMit: 一般限流模式  
TRIP: 输出关断模式  
LIMRELAY | LIMITRELAY: 一般限流加外部继电器控制模式  
TRIPRELAY: 输出关断加外部继电器控制模式

---

例子 :SOURce2:CURRent:TYPE LIMITRELAY  
设定通道 2 限流模式为一般限流模式外加外部继电器控制

指令 :[SOURce[1|2]]:CURRent[:LIMit]:TYPE?

功能 查询电流的限流模式

---

例子 :SOURce2:CURRent:TYPE?  
查询通道 2 电流的限流模式

---

指令	: <b>[SOURce[1 2]]:CURRent[:LIMit]:STATe?</b>
功能	查询电流的限流状态，返回 0 表示未限流，1 表示限流

---

例子	: <b>SOURce2:CURRent:STATe?</b> 查询通道 2 电流的限流状态
----	---

指令	: <b>[SOURce[1 2]]:VOLTag[:LEVel][:IMMediate][:AMPLitude] &lt; NRf &gt;</b>
功能	设定输出电压幅值
说明	NRf      0.000-15.000

---

例子	: <b>SOURce2:VOLTag 5.321</b> 设定输出通道 2 电压 5.321V
----	---

指令	: <b>[SOURce[1 2]]:VOLTag[:LEVel][:IMMediate][:AMPLitude]?</b>
功能	查询已设定电压幅值

---

例子	: <b>SOURce2:VOLTag?</b> 查询通道 2 设定电压
----	---

指令	: <b>[SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude] &lt; NRf &gt;</b>
功能	设定电阻值
说明	NRf      0.000-1.000

---

例子	: <b>SOURce: RESistance 1.000</b> 设定电阻值
----	--

指令	: <b>[SOURce]:RESistance[:LEVel][:IMMediate][:AMPLitude]?</b>
功能	查询已设定电阻值

---

例子        :SOURce: RESistance?  
              查询已设定电阻值并返回读值

## 回读指令

指令	:SENSe[1 2]:FUNcTion <name>
功能	选择测量函数(电压/电流/脉冲测量/长积分测量/DVM测量)
说明	name     “VOLTage”: 电压测量 “CURRent”: 电流测量 “PCURrent”: 脉冲测量 “LINTegration”: 长积分测量 “DVMeter”: 外部设备电压测量

---

例子     :SENSe2:FUNcTion “VOLTage”  
          选择通道 2 测量函数为电压模式

指令	:SENSe[1 2]:FUNcTion?
功能	查询测量函数
响应时间	最大值 16ms

---

例子     :SENSe2:FUNcTion?  
          查询通道 2 测量函数

指令	:SENSe[1 2]:NPLCycles <n>
功能	设定针对于电压、电流、DVM 测试的相对于电源周期的整合比例
说明	n           0.01-10.00

---

例子     :SENSe2:NPLCycles 0.10  
          设定通道 2NPLC 的值为 0.1

指令	:SENSe[1 2]:NPLCycles?
功能	查询整合比例

---

例子	:SENSe2:NPLCycles? 查询通道 2 整合比例
指令	:SENSe[1 2]:AVERAge <NRf>
功能	设定用于计算电压、电流、DVM 测量值的平均数的平均周期个数
说明	NRf      1-10
例子	:SENSe2:AVERAge 3 设定通道 2 用于平均的周期个数为 3
指令	:SENSe[1 2]:AVERAge?
功能	查询用于计算电压、电流、DVM 测量值的平均数的平均周期个数
例子	:SENSe2:AVERAge? 查询通道 2 用于计算电压、电流、DVM 测量值的平均数的平均周期个数
指令	:SENSe[1]:CURRent[:DC]:RANGe[:UPPer] <n>
功能	设定电流测量的量程范围
说明	n            0.005: 小量程 (<=0.005) 0.5    中量程 (>=0.005&<=0.5) 5: 大量程(>0.5)
例子	:SENSe:CURRent:RANGe 0.5 设定通道 1 电流测量的量程范围为中量程
指令	:SENSe2:CURRent[:DC]:RANGe[:UPPer] <n>
功能	设定电流测量的量程范围

说明	n	MIN: 小量程 MAX 大量程
例子	:SENSe2:CURRent:RANGe MIN 设定通道 2 电流测量的量程范围为小量程	
指令	:SENSe[1 2]:CURRent[:DC]:RANGe[:UPPer]?	
功能	查询电流测量的量程范围	
说明	已处于"AUTO"设定時，返回的是实际所处的量程 (MAX 或 MIN)，而不是"AUTO"。	
例子	:SENSe2:CURRent:RANGe? 查询通道 2 电流测量的量程范围	
指令	:SENSe[1 2]:CURRent[:DC]:RANGe:AUTO <b>	
功能	使能或禁止电流测量的自动模式	
说明	b	0/OFF: 禁止 1/ON: 使能
例子	:SENSe2:CURRent:RANGe:AUTO ON 使能通道 2 电流测量的自动模式	
指令	:SENSe[1 2]:CURRent[:DC]:RANGe:AUTO?	
功能	查询电流测量之自动模式的使能状态	
例子	:SENSe2:CURRent:RANGe:AUTO? 查询通道 2 电流测量之自动模式的使能状态	
指令	:SENSe[1 2]:PCURRent:AVERAge <NRf>	
功能	设定用于计算脉冲测量值的平均数的平均周期个数	
说明	NRf	1-100 或 1-5000(数字量化输出模式/Pulse current digitization)

---

例子	:SENSe2:PCURrent:AVERage 5 设定通道 2 平均周期个数为 5
指令	:SENSe[1 2]:PCURrent:AVERage?
功能	查询用于计算脉冲测量值的平均数的平均周期个数
例子	:SENSe2:PCURrent:AVERage? 查询通道 2 已设定的平均周期个数
指令	:SENSe[1 2]:PCURrent:MODE <name>
功能	指定脉冲测量的测量模式
说明	name   HIGH: 高脉冲（以触发信号的上升沿为开始） LOW: 低脉冲（以触发信号的下降沿为开始） AVERage: 脉冲平均测量
例子	:SENSe2:PCURrent:MODE HIGH 设定通道 2 为高脉冲测量方式
指令	:SENSe[1 2]:PCURrent:MODE?
功能	查询脉冲测量的测量模式
例子	:SENSe2:PCURrent:MODE? 查询通道 2 脉冲测量的测量模式
指令	:SENSe[1 2]:PCURrent:TIME:AUTO
功能	设定脉冲测量为自动设定积分时间之测量模式
例子	:SENSe2:PCURrent:TIME:AUTO 设定通道 2 脉冲测量为自动设定积分时间之测量模式

---

指令 :SENSe[1|2]:PCURrent:TIME:HIGH <NRf>

功能 设定脉冲测量之高脉冲的积分时间

说明 NRf 33.3~ 833333, 步进为 33.3

---

例子 :SENSe2:PCURrent:TIME:HIGH 0.000233

设定通道 2 高脉冲的积分时间为 233 微秒

---



提示

电流数字化输出模式(Pulse current digitization)时  
IntTime 设定自动变为 33.3 微秒(uS)

---

指令 :SENSe[1|2]:PCURrent:TIME:HIGH?

功能 查询脉冲测量高脉冲的积分时间

---

例子 :SENSe2:PCURrent:TIME:HIGH?

查询通道 2 脉冲测量高脉冲的积分时间

---



提示

电流数字化输出模式(Pulse current digitization)时  
IntTime 回读的将是已取整的 33 微秒(uS)

---

指令 :SENSe[1|2]:PCURrent:TIME:LOW <NRf>

功能 设定脉冲测量低脉冲的积分时间

说明 NRf 33.3-833333, 步进为 33.3

---

例子 :SENSe2:PCURrent:TIME:LOW 0.000233

设定通道 2 低脉冲的积分时间为 233 微秒

---



提示

电流数字化输出模式(Pulse current digitization)时  
IntTime 设定自动变为 33.3 微秒(uS)

---

指令 :SENSe[1|2]:PCURrent:TIME:LOW?

功能 查询低电平脉冲测量的积分时间

---

例子 :SENSe2:PCURrent:TIME:LOW?

查询通道 2 脉冲测量低脉冲的积分时间

---

 提示 电流数字量化输出模式(Pulse current digitization)时  
IntTime 回读的将是已取整的 33 微秒(μS)

指令 :SENSe[1|2]:PCURrent:TIME:AVERAge <NRf>  
功能 设定平均脉冲测量的积分时间  
说明 NRf 33-833333, 步进为 33.3

例子 :SENSe2:PCURrent:TIME:AVERAge 0.000233  
查询通道 2 平均脉冲的积分时间为 233 微秒

 提示 电流数字量化输出模式(Pulse current digitization)时  
IntTime 设定自动变为 33.3 微秒(μS)

指令 :SENSe[1|2]:PCURrent:TIME:AVERAge?  
功能 查询平均脉冲测量的积分时间

例子 :SENSe2:PCURrent:TIME:AVERAge?  
查询通道 2 平均脉冲积分时间

 提示 电流数字量化输出模式(Pulse current digitization)时  
IntTime 回读的将是已取整的 33 微秒(μS)

指令 :SENSe[1|2]:PCURrent:SYNChronize[:STATe] <b>  
功能 设定脉冲电流测量的触发方式选择。  
说明 b 0 /OFF: 数字量化输出模式 (Pulse current  
digitization)  
1/ON: 一般的脉冲电平触发方式

例子 :SENSe2:PCURrent:SYNChronize ON  
设定通道 2 为一般的脉冲电平触发方式

 提示 数字量化输出模式 (Pulse current digitization):  
远程控制时, 一次批量读取数据, 数量范围可在 1~5000  
之间设定, 参照:SENSe[1|2]:PCURrent:AVERAge <NRf>

指令	:SENSe[1 2] [1 2]:PCURrent:SYNChronize[:STATe]?
功能	查询脉冲电流测量的触发方式
例子	:SENSe2:PCURrent:SYNChronize? 查询通道 2 脉冲电流测量的触发方式
指令	:SENSe[1 2]:PCURrent:SYNChronize:DElay <NRf>
功能	设定触发延时的时间
说明	NRf 0-0.1 或 0-5(数字量化输出模式/Pulse current digitization)
例子	\:SENSe2:PCURrent:SYNChronize:DElay 0.05 设定通道 2 触发延时时间为 0.05s
指令	:SENSe[1 2]:PCURrent:SYNChronize:DElay?
功能	查询触发延时的时间
例子	:SENSe2:PCURrent:SYNChronize:DElay? 查询通道 2 触发延时的时间
指令	:SENSe[1 2]:PCURrent:SYNChronize:TLEVel<NRf>
功能	设定触发电平
说明	NRf 0.000-5.000
例子	:SENSe2:PCURrent:SYNChronize:TLEVel 1 设定通道 2 触发电平为 1.000A
指令	:SENSe[1 2]:PCURrent:SYNChronize:TLEVel?
功能	查询已设定的触发电平
例子	:SENSe2:PCURrent:SYNChronize:TLEVel? 查询通道 2 已设定的触发电平

---

指令	:SENSe[1 2]:LINTegration:TIME <NRf>
功能	设定长积分的积分时间
说明	NRf X (电源频率为 50HZ 时 X=0.840~60.000,分辨率为 20mS;为 60HZ 时 X=0.850~60.000,分辨率为 16.7mS)

---

例子	:SENSe2:LINTegration:TIME 1.2 设定通道 2 长积分的时间为 1.2s
----	--

指令	:SENSe[1 2]:LINTegration:TIME?
功能	查询长积分的积分时间

---

例子	:SENSe2:LINTegration:TIME? 查询通道 2 长积分的积分时间
----	---

指令	:SENSe[1 2]:LINTegration:TIME:AUTO
功能	电源内部自动设定长积分的积分时间

---

例子	:SENSe2:LINTegration:TIME:AUTO 设定通道 2 电源内部自动设定长积分的积分时间
----	---

指令	:SENSe[1 2]:LINTegration:TLEVel <NRf>
功能	设定长积分测量的触发电平
说明	NRf 0.000-5.000

---

例子	:SENSe2:LINTegration:TLEVel 1.2 设定通道 2 长积分测量的触发电平为 1.2A
----	--

指令	:SENSe[1 2]:LINTegration:TLEVel?
功能	查询长积分测量的触发电平

---

---

例子	:SENSe2:LINTegration:TLEVel? 查询通道 2 长积分测量的触发电平
指令	:SENSe[1 2]:LINTegration:TEDGE <name>
功能	设定长积分测量开始的触发边沿模式
说明	name    RISING: 上升沿触发 FALLING: 下降沿触发 NEITHER: 无边沿触发

---

例子	:SENSe2:LINTegration:TEDGE RISING 设定通道 2 长积分测量开始的触发边沿模式为上升沿触发
指令	:SENSe[1 2]:LINTegration:TEDGE?
功能	查询长积分测量开始的触发边沿模式

---

例子	:SENSe2:LINTegration:TEDGE? 查询通道 2 长积分测量开始的触发边沿模式
指令	:SENSe[1 2]:LINTegration:TimeOUT <NRf>
功能	设定长积分测量中用于计时溢出的计时时间长度
说明	NRf     1-63

---

例子	:SENSe2:LINTegration:TimeOUT 2 设定通道 2 长积分测量中用于计时溢出的计时时间为 2s
指令	:SENSe[1 2]:LINTegration:TimeOUT?
功能	查询长积分测量中用于计时溢出的计时时间长度

---

例子	:SENSe2:LINTegration:TimeOUT? 查询通道 2 长积分测量中用于计时溢出的计时时间
----	---

---

指令	:SENSe[1 2]:LINTegration:SEARCh <b>
功能	使能或禁止长积分测量中脉冲搜索功能
说明	b            0/OFF: 禁止 1/ON: 使能

---

例子	:SENSe2:LINTegration:SEARCh ON 设定通道 2 使能长积分测量脉冲搜索功能
----	--

指令	:SENSe[1 2]:LINTegration:SEARCh?
功能	查询长积分测量中脉冲搜索功能的使能状态

---

例子	:SENSe2:LINTegration:SEARCh? 查询通道 2 长积分测量中脉冲搜索功能的使能状态
----	--

指令	:SENSe[1 2]:LINTegration:FAST <b>
功能	使能或禁止长积分测量的快速测量模式
说明	b            0/OFF: 禁止 1/ON: 使能

---

例子	:SENSe2:LINTegration:FAST ON 设定通道 2 使能长积分测量的快速测量模式
----	---

指令	:SENSe[1 2]:LINTegration:FAST?
功能	查询长积分测量的快速测量模式之状态

---

例子	:SENSe2:LINTegration:FAST? 查询通道 2 长积分测量的快速测量模式之状态
----	--

## 状态指令

指令 :STATus:PRESet

功能 清除操作事件使能寄存器(Operation event enable register), 测量事件使能寄存器(Measurement event enable register), 以及问题事件使能寄存器(Questionable event enable register),并返回为默认设置状态

---

例子 :STATus:PRESet

指令 :STATus:OPERation[:EVENT]?

功能 读取操作事件寄存器组中之事件记录寄存器

---

例子 :STATus:OPERation?

读取操作事件寄存器组中之事件记录寄存器

指令 :STATus:OPERation:CONDition?

功能 读取操作事件寄存器组中之状态寄存器

---

例子 :STATus:OPERation:CONDition?

读取操作事件寄存器组中之状态寄存器

指令 :STATus:OPERation:ENABle <NRf>

功能 编程使能指定的操作事件寄存器组中事件记录寄存器

说明 NRf 8: CL 使能位

16: CLT 使能位

64: PSS 使能位

---

例子 :STATus:OPERation:ENABle 64

使能 PSS 功能

指令	:STATus:OPERation:ENABLE?
功能	读取操作事件寄存器组中使能寄存器
例子	:STATus:OPERation:ENABLE? 读取操作事件寄存器组中使能寄存器
指令	:STATus:MEASurement[:EVENT]?
功能	读取测量状态寄存器组中之事件记录寄存器
例子	:STATus:MEASurement? 读取测量状态寄存器组中之事件记录寄存器
指令	:STATus:MEASurement:ENABle <NRf>
功能	编程设置测量状态寄存器组中之使能寄存器
说明	NRf      8: 读取溢出 ROF 使能位 16: 脉冲触发溢出 PTT 使能位 32: 读取可用功能 RAV 使能位 512: 缓冲器满使能位 (寄存器是 16 位的, <value>值在 512~1023 肯定有效, 在 1024~ 65535 之間須保證 BF (bit9) 有效的數才可。)
例子	:STATus:MEASurement:ENABle 8 使能读取溢出功能
指令	:STATus:MEASurement:ENABle?
功能	读取测量状态寄存器组中之使能寄存器
例子	:STATus:MEASurement:ENABle? 读取测量状态寄存器组中之使能寄存器

---

指令	:STATus:MEASurement:CONDition?
功能	读取测量状态寄存器组中之状态寄存器
例子	:STATus:MEASurement:CONDition? 读取测量状态寄存器组中之状态寄存器
指令	:STATus:QUEStionable[:EVENT]?
功能	读取问题事件记录寄存器
例子	:STATus:QUEStionable? 读取问题事件记录寄存器
指令	:STATus:QUEStionable:CONDition?
功能	读取问题事件状态寄存器
例子	:STATus:QUEStionable:CONDition? 读取问题事件状态寄存器
指令	:STATus:QUEStionable:ENABle <NRf>
功能	编程设定问题状态寄存器组中的使能寄存器
说明	NRf     256: 校正使能位  寄存器是 16 位的, <value>值在 256~511 肯定有效, 在 512~65535 之間須保證 Cal (bit8) 有效的數才可。
例子	:STATus:QUEStionable:ENABle 256 指定使能 Cal 功能位
指令	:STATus:QUEStionable:ENABle?
功能	读取问题状态寄存器组中的使能寄存器
例子	:STATus:QUEStionable:ENABle? 读取问题状态寄存器组中的使能寄存器

---

指令	:STATus:QUEue[:NEXT]?
功能	读取最近的出错信息
例子	:STATus:QUEue? 读取最近的出错信息
指令	:STATus:QUEue:ENABle <list>
功能	指定出错和状态信息进入出错队列
说明	list 可指定范围 (-440:+900) (-110): 单条出错信息进入出错队列 (-110:-222): 指定范围内的出错信息进入出错队列 (-110:-222, -220): 指定一定范围内的出错信息和某一条一起进入出错队列
例子	:STATus:QUEue:ENABle (-110:-222) 指定(-110:-222)区间内的出错和状态信息进入出错队列
指令	:STATus:QUEue:ENABle?
功能	读取已被使能的信息列表空间
例子	:STATus:QUEue:ENABle? 读取已被使能的信息列表空间
指令	:STATus:QUEue:DISABle <list>
功能	指定消息不被放入出错队列
说明	list 可指定范围 (-440:+900) (-110): 单条出错信息进入出错队列 (-110:-222): 指定范围内的出错信息进入出错队列 (-110:-222, -220): 指定某范围内的出错信息和某一条一起进入出错队列

---

例子 :STATus:QUEue:DISable (-110:-222)  
指定(-110:-222)范围内的出错信息不能进入出错队列

指令 :STATus:QUEue:DISable?

功能 读取未被使能的消息

---

例子 :STATus:QUEue:DISable?

读取未被使能的消息

指令 :STATus:QUEue:CLEar

功能 清空出错队列中的所有消息

---

例子 :STATus:QUEue:CLEar

清空出错队列中的所有消息

## 系统指令

指令 :SYSTem:VERSion?

功能 查询 SCPI 的版本级别

---

例子 :SYSTem:VERSion?

查询 SCPI 的版本级别

指令 :SYSTem:ERRor?

功能 读取并清除错误队列中的错误信息

---

例子 :SYSTem:ERRor?

读取并清除错误队列中的错误信息

指令 :SYSTem:CLEar

功能 清空出错队列

---

例子 :SYSTem:CLEar

清空出错队列

指令 :SYSTem:LFRequency?

功能 查询电源线的频率

---

例子 :SYSTem:LFRequency?

查询电源线的频率

指令 :SYSTem:POSetup <name>

功能 选择开机调用设定模式

说明	name	RST: 机器默认设置 SAV0: 用户已存储设置 0 (OUTPUT 状态为 off) SAV1: 用户已存储设置 1 (OUTPUT 状态为 off) SAV2: 用户已存储设置 2 (OUTPUT 状态为 off) SAV3: 用户已存储设置 3 (OUTPUT 状态为 off) SAV4: 用户已存储设置 4 (OUTPUT 状态为 off) SAV5: 用户已存储设置 5 SAV6: 用户已存储设置 6 SAV7: 用户已存储设置 7 SAV8: 用户已存储设置 8 SAV9: 用户已存储设置 9
----	------	--

例子 :SYSTem:POSetup SAV0  
指定开机调用设定为用户已存储设置 0

指令 :SYSTem:POSetup?  
功能 查询开机所调用的设定模式

例子 :SYSTem:POSetup?  
查询开机所调用的设定模式

指令 :SYSTem:COMMunicate:LAN:DHCP[:STATe] <b>  
功能 设置 DHCP 配置模式的状态  
说明 b 0/OFF: 关闭  
1/ON: 打开

需要执行:SYSTem:COMMunicate:LAN:APPLy 命令，新设置的状态才会生效

例子 :SYSTem:COMMunicate:LAN:DHCP ON  
打开设置 DHCP 配置模式

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

功能 查询服务器开关状态

例子 :SYSTem:COMMunicate:LAN:DHCP?  
查询 DHCP 配置模式的状态

指令 :SYSTem:COMMunicate:LAN:IPADdress <IP 地址>

功能 IP 地址设定

说明 IP 地址 ASCII 字符串，取值范围为 1.0.0.0 至 223.255.255.255（127.nnn.nnn.nnn 除外）。

该命令仅在手动 IP 模式下有效，发送该命令后，需要执行 SYSTem:COMMunicate:LAN:APPLy 命令，新设置的 IP 地址才可生效。

例子 :SYSTem:COMMunicate:LAN:IPADdress 172.131.161.152  
设定 IP 地址为：172.131.161.152

指令 :SYSTem:COMMunicate:LAN:IPADdress?

功能 查询 IP 地址

例子 :SYSTem:COMMunicate:LAN:IPADdress?  
查询 IP 地址

指令 :SYSTem:COMMunicate:LAN:SMASk <掩码>

功能 子网掩码设置

说明 掩码 ASCII 字符串，取值范围为 1.0.0.0 至 255.255.255.255。

发送该命令后，需要  
执:SYSTem:COMMunicate:LAN:APPLY 命令，新设置的  
子网掩码才可生效。

---

例子 :SYSTem:COMM:LAN:SMAS 255.255.255.0  
设置子网掩码为 255.255.255.0

指令 :SYSTem:COMMunicate:LAN:SMASK?

功能 查询子网掩码

---

例子 :SYSTem:COMMunicate:LAN:SMASK?  
查询子网掩码

指令 :SYSTem:COMMunicate:LAN:GATEway <IP 地址>

功能 设置网关 IP 地址

说明 IP 地址 ASCII 字符串，取值范围为 1.0.0.0 至  
223.255.255.255（127.nnn.nnn.nnn 除外）。

发送该命令后，需要  
执:SYSTem:COMMunicate:LAN:APPLY 命令，新设置的  
网关才可生效。

---

例子 :SYSTem:COMMunicate:LAN:GATEway 172.16.3.1  
设置网关为 172.16.3.1

指令 :SYSTem:COMMunicate:LAN:GATEway?

功能 查询网关

---

例子 :SYSTem:COMMunicate:LAN:GATEway?  
查询网关

指令 :SYSTem:COMMunicate:LAN:DNS <IP 地址>

功能 设置网络的 DNS 服务器

说明 IP 地址 ASCII 字符串，取值范围为 1.0.0.0 至 223.255.255.255（127.nnn.nnn.nnn 除外）。

发送该命令后，需要  
执:SYSTem:COMMunicate:LAN:APPLy 命令，新设置的  
DNS 服务器才可生效。

---

例子 :SYSTem:COMMunicate:LAN:DNS 172.16.2.3  
设置网络的 DNS 服务器地址为 172.16.2.3

指令 :SYSTem:COMMunicate:LAN:DNS?

功能 查询网络的 DNS 服务器

---

例子 :SYSTem:COMMunicate:LAN:DNS?  
查询网络的 DNS 服务器

指令 :SYSTem:COMMunicate:LAN:MANualip[:STATe] <b>

功能 手动设置 IP 开关

b 0/OFF: 关闭

1/ON: 打开

---

例子 :SYSTem:COMMunicate:LAN:MANualip ON  
打开手动设置 IP 开关

指令 :SYSTem:COMMunicate:LAN:MANualip[:STATe]?

功能 查询手动设置 IP 开关状态

---

例子 :SYSTem:COMMunicate:LAN:MANualip?  
查询手动设置 IP 开关状态

指令 :SYSTem:COMMunicate:LAN:APPLy

功能 执行该命令，仪器将应用所设置的 LAN 参数

---

---

例子 :SYSTem:COMMunicate:LAN:APPLy  
应用已设置的 LAN 参数

指令 :SYSTem:REMote

功能 设置为远程控制

---

例子 :SYSTem:REMote  
设置为远程控制

指令 :SYSTem:BEEPer:STATe <b>

功能 设置蜂鸣器开关状态

b 0/OFF: 关闭蜂鸣器

1/ON: 打开蜂鸣器

例子 :SYSTem:BEEPer:STATe OFF  
关闭蜂鸣器

指令 :SYSTem:BEEPer:STATe?

功能 查询蜂鸣器开关状态

---

例子 :SYSTem:BEEPer:STATe?  
查询蜂鸣器开关状态

指令 :SYSTem:LOCAl

功能 断开远程控制

---

例子 :SYSTem:LOCAl  
断开远程控制

## 与系统相关指令

指令     \*IDN?

功能     读取仪器的标识<字符串>。

说明     字符串   <字符串>包含有四个逗号分隔的字段，第一个字段是制造商的名称，第二个字段是型号，第三个字段是机器的特定序列号，第四个字段是版本号。

---

例子     \*IDN?

        返回 GW,PPH-1503D,XXXXXXXXX,V0.62

        GW: 制造商名称,

        PPH-1503D: 机器的型号,

        XXXXXXXXX: 机器的序列号,

        V0.62: 软件版本号。

        根据 PPH 类型来描述

指令     \*RST

功能     复位机器, 从存储单元 RST 中调用用户设置。

---

例子     \*RST

        复位机器

指令     \*TST?

功能     自检或检测 RAM。

        返回值   0: 表示没有错误。

                  2: 表示 RAM 有错误。

---

例子     \*TST?

        返回 0 , 如果没有错误。返回 2 , 如果 RAM 有错误。

---

指令	*WAI
功能	等待所有未决操作完成之后，再通过接口执行任何其他命令。

---

例子	*WAI
----	------

---

指令	*TRG
功能	发送总线触发器

---

例子	*TRG 发送总线触发器
----	-----------------

---

指令	*SAV <NRf>
功能	保存现在设置 • 到用户保存设置中
说明	NRf 0: 保存设置的存储单元 SAV0 1: 保存设置的存储单元 SAV1 2: 保存设置的存储单元 SAV2 3: 保存设置的存储单元 SAV3 4: 保存设置的存储单元 SAV4

---

例子	*SAV 3 保存当前设置到存储单元 SAV3 中
----	------------------------------

---

指令	*RCL <NRf>
功能	从寄存器中调用用户已保存设置
说明	0: 保存设置的存储单元 SAV0 1: 保存设置的存储单元 SAV1 2: 保存设置的存储单元 SAV2 3: 保存设置的存储单元 SAV3 4: 保存设置的存储单元 SAV4

---

例子      \*RCL 2

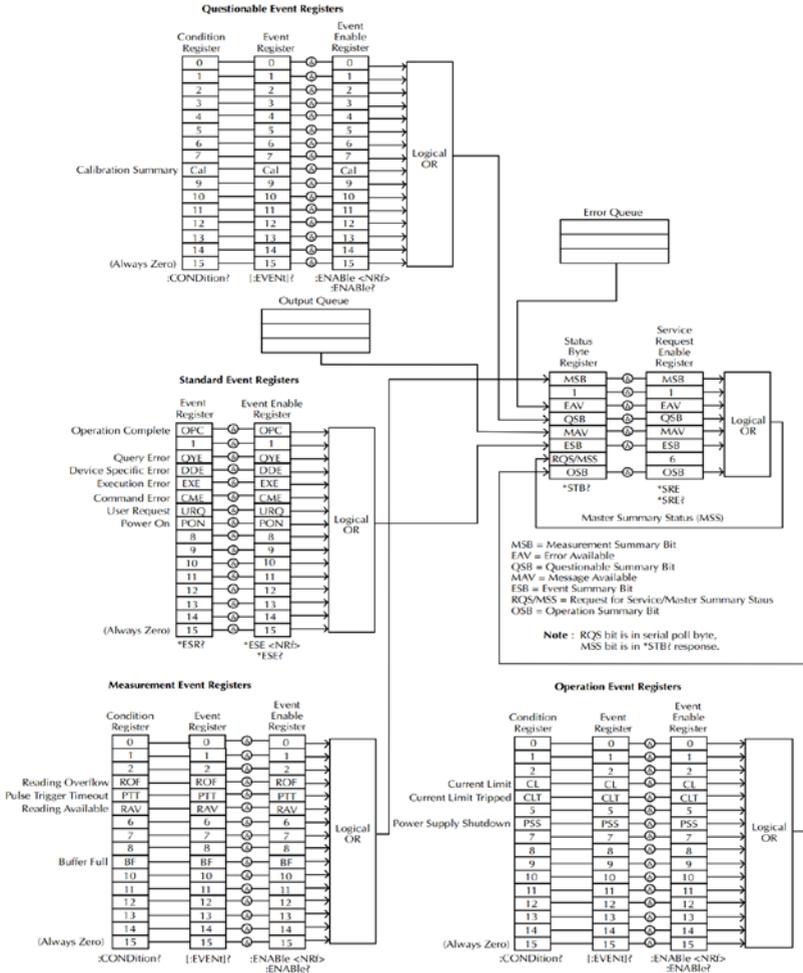
从存储单元 SAV2 中调用用户保存设置

## SCPI 状态模式

---

所有 SCPI 仪器配置都以相同方法提供给状态寄存器。状态系统在三个寄存器组中记录各种仪器状况，这三个寄存器组：状态字节（Status Byte）寄存器、标准事件（Standard Event）寄存器和质疑数据（Questionable Data）寄存器。状态字节寄存器记录了其它寄存器群组中报告的高阶摘要信息。下图就是 SCPI 状态系统图。

SCPI 状态系统



\* note: URQ 表示操作面板上的"Lock" 键被操作過 (從 unlock 進入 lock 或者從 lock 進入 unlock)。

## 事件寄存器

标准事件和质疑数据寄存器都有事件寄存器。事件寄存器是只读寄存器，它反应了仪器中所定义的状况。寄存器中的各个位都是锁存的，只要一个事件位被设定，接下来的状态变更都会被忽略。查询寄存器 (如\*ESR?) 或发送\*CLS 命令，都会自动清除事件寄存器中的各

个位。复位 (\*RST) 或设备清除命令，不会清除事件寄存器中的各个位。查询事件寄存器传回一个十进位的数值，表示寄存器中设备的所有位的二进制加权之和。

## 允许寄存器

允许寄存器定义在对应的事件寄存器中哪些位进行逻辑或操作形成唯一的累加位。允许寄存器即可读也可写。查询允许寄存器不会清除寄存器的值。\*CLS (清除状态) 命令不会清除允许寄存器，但会清除事件寄存器中的各个位。若要设定允许寄存器中的各个位，必须将想设定的位所代表的二进制数，以十进位表表示，写入寄存器。

## 状态字节寄存器

状态字节累加寄存器报告其他状态寄存器的状态。可通过“信息可用位” (第 4 位) 立即报告函数发生器输出缓冲区中待发的数据。从其他一个寄存器组中清除一个事件寄存器，将会清除状态字节条件寄存器中相应的位。读取输出缓冲区中的所有信息，包括任何未决的查询，将清除“信息可用”位。要设置允许寄存器掩码并生成一个 SRQ (服务请求)，必须使用 \*SRE 命令将一个十进制值写入寄存器。

### 位定义 – 状态字节寄存器

位编号	十进制值	定义
0 未使用	1	未使用。返回“0”
1 未使用	2	未使用。返回“0”
2 错误队列	4	存储在“错误队列”中的一个或多个错误。
3 可疑数据	8	在可疑数据寄存器中设置一个或多个位 (这些位必须已启用)
4 信息可用	16	仪器输出缓冲器中可用的数据
5 标准事件	32	在标准事件寄存器中设置一个或多个位 (这些位必须已启用)
6 主累加	64	在状态字节寄存器中设置一个或多个位 (这些位必须已启用)

7 未使用	128	未使用。返回“0”
-------	-----	-----------

出现下列情况时，会清除状态字节条件寄存器：

- 执行 \*CLS 清除状态命令。
- 从其他一个寄存器组中读取事件寄存器（只清除条件寄存器中相应的位）。

出现下列情况时会清除状态字节允许寄存器：

- 执行 \*SRE 0 命令

### 使用 \*STB? 读取状态字节

\*STB? 命令返回的结果只要允许的条件仍然保留就不清除第 6 位。

### 使用 \*OPC 显示输出缓冲器中的信号

一般而言，最好是使用标准事件寄存器的“执行完毕”位（位 0），来表示命令序列已经执行完毕。在执行 \*OPC 命令之后，这个位就会被设为 1。如果在将信息载入仪器输出缓冲器的命令（无论是读取数据或查询数据）之后发送 \*OPC，可以使用执行完毕位来判断什么时候信息可利用。不过，如果在 \*OPC 命令执行（依序）之前有太多信息产生，输出缓冲器会饱和，仪器就会停止取读数。

## 标准事件寄存器

标准事件寄存器组报告下列类型的事件：加电检测、命令语法错误、命令执行错误、自检或校准错误、查询错误或者已执行的 \*OPC 命令，任一或全部状态都可以通过允许寄存器报告给标准事件累加位。要设置允许寄存器掩码，必须使用 \*ESE 命令向寄存器中写入一个十进制的值。

### 位定义—标准事件寄存器

位编号	十进制值	定义
0 操作完成	1	所有 *OPC 之前的命令（包括 *OPC 命令）都已完成，并且重叠的命令也已经完成。

1 未使用	2	未使用, 返回“0”
2 查询错误	4	该仪器试图读取输出缓冲器, 但它是空的。或者在读取上一次查询之前接收到一个新的命令行, 或者输入和输出缓冲区都已满。
3 设备错误	8	出现自检、校准或其它设备特定的错误
4 执行错误	16	出现执行错误
5 命令错误	32	出现命令语法错误
6 未使用	64	未使用, 返回“0”
7 接通电源	128	自从上次读取或清除事件寄存器之后, 一直开关电源

出现下列情况时会清除标准事件寄存器

- 执行 \*CLS 命令
- 使用 \*ESR? 命令查询事件寄存器

出现下列情况时会清除标准事件允许寄存器

- 执行 \*ESE 0 命令

### 状态字节寄存器命令

指令        \*SRE <允许值>

功能        启动状态字节允许寄存器中的位, SRER 寄存器中设定的所有位代表的二进制加权之和。

说明        允许值    一个十进制位值: 0~255

例子        \*SRE 7

             设置 SRER 为 0000 0111

---

指令	*SRE?
功能	查询状态字节允许寄存器。它会传回一个十进制位值，表示寄存器中设定的所有位代表的二进制加权之和，范围是 0~255。
例子	*SRE? 返回 7，因为此时 SRER 被设置为 0000 0111
指令	*STB?
功能	查询状态字节累加寄存器，该命令返回的结果与串行轮询的相同，但“主累加”位（第 6 位）不会被 *STB? 命令清除。返回值范围是 0~255。
例子	*STB? 返回 81，如果 SBR 被设置为 0101 0001

---

## 标准事件寄存器命令

指令 \*ESE<允许值>

功能 设置标准事件寄存器。允许值的范围是 0~255

---

例子 \*ESE 65

设置 ESER 为 0100 0001

指令 \*ESE?

功能 查询标准事件寄存器。它会传回一个十进制位值，表示寄存器中设定的所有位的二进制加权之和。

---

例子 \*ESE?

返回 65，因为 ESER 为 0100 0001

指令 \*ESR?

功能 查询标准事件寄存器。它会传回一个十进制位值。表示寄存器中设定的所有位所代表的二进制加权之和，范围是 0~255。

---

例子 \*ESR?

返回 198，因为 ESER 为 1100 0110

## 其它状态寄存器命令

指令 \*CLS

功能 清除状态字节累加寄存器和所有事件寄存器。

---

例子 \*CLS

清除所有事件寄存器,涵盖 Standard event registers, Operation event registers, Measurement event registers, Questionable event registers.

指令 \*OPC

功能 执行命令之后, 设定标准事件寄存器中的“执行完毕”位。

---

例子 \*OPC

指令 \*OPC?

功能 当询问时, 会传回“1”到输出缓冲器在执行命令之后。

---

例子 \*OPC?

在执行完上一个指令后, 返回“1”到输出缓冲器。

## 错误

---

### 错误信息

- 以先进先出(FIFO)的顺序检索错误。返回的第一个错误即是存储的第一个错误。读取错误时，错误即被清除。
- 如果产生的错误超过 10 个，存储在队列中的最后一个错误(最新错误)会被替换为“Queue overflow”。除非清除队列中的错误，否则不再存储其他错误。如果在读取错误队列时没有出现错误，则仪器将响应“No error”。
- 可以使用：:SYSTem:CLear 命令或开关电源来清除错误队列。当您读取错误队列时，错误也被清除。当仪器复位(使用 \*RST 命令)时，不会清除错误队列。
- 远程接口操作见上述指令介绍

### 命令错误

- 440 Query unterminated after indefinite
- 430 response
- 420 Query deadlocked
- 410 Query unterminated
- 363 Query interrupted
- 350 Input buffer overrun
- 330 Queue overflow
- 314 Self-test failed
- 315 Save/recall memory lost
- 260 Configuration memory lost
- 241 Expression error
- 230 Hardware missing
- 225 Data corrupt or stale
- 224 Out of memory

- 223 Illegal parameter value
- 222 Too much data
- 221 Parameter data out of range
- 220 Settings conflict
- 200 Parameter error
- 178 Execution error
- 171 Expression data not allowed
- 170 Invalid expression
- 161 Expression error
- 160 Invalid block data
- 158 Block data error
- 154 String data not allowed
- 151 String too long
- 150 Invalid string data String data error
- 148 Character data not allowed
- 144 Character data too long
- 141 Invalid character data
- 140 Character data error
- 124 Too many digits
- 123 Exponent too large
- 121 Invalid character in number
- 120 Numeric data error
- 114 Header suffix out of range
- 113 Undefined header
- 112 Program mnemonic too long
- 111 Header separator error
- 110 Command header error
- 109 Missing parameter

---

-108	Parameter not allowed
-105	GET not allowed
-104	Data type error
-103	Invalid separator
-102	Syntax error
-101	Invalid character
-100	Command error
+000	No error
+101	Operation complete
+301	Reading overflow
+302	Pulse trigger detection timeout
+306	Reading available
+310	Buffer full
+320	Current limit event
+321	Current limit tripped event
+409	OTP Error
+410	OVP Error
+438	Date of calibration not set
+440	Gain-aperture correction error
+500	Calibration data invalid
+510	Reading buffer data lost
+511	GPIB address lost
+512	Power-on state lost
+514	DC Calibration data lost
+515	Calibration dates lost
+522	GPIB communication data lost
+610	Questionable calibration
+900	Internal system error