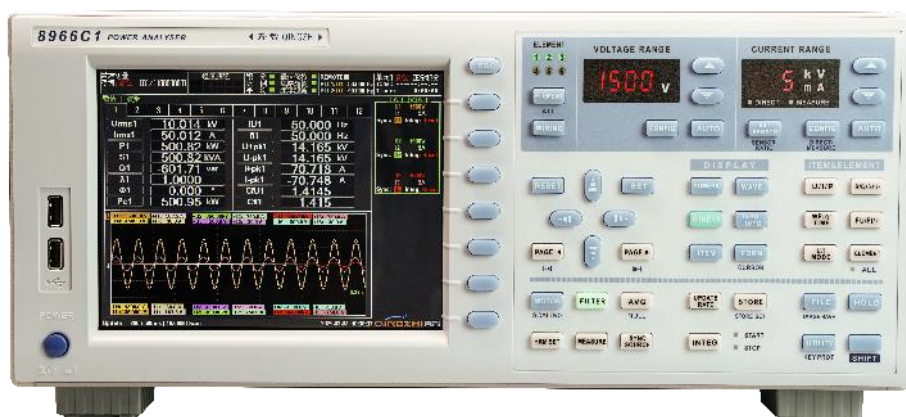


8966C1 高精度数字功率分析仪

SCPI通信说明

版本

- V1.00版



青岛青智仪器有限公司

地址：青岛市高新区宝源路780号联东U谷8号楼东

电话：0532-81920028(多线)

技术热线：(0)13953270323

网址：<https://www.qingzhi.com>

更多详细资料，例如通讯协议，上位机软件，请扫描下方二维码至公司网站技术资料中下载



感谢： 欢迎选择青智仪器有限公司的产品，在本产品使用前请详细阅读本手册，以便于正确使用。

请注意以下事项

- 本手册的版权归青智仪器有限公司所有。在未经本公司书面许可的情况下，严禁以任何形式复制、传递、分发和存储本手册的任何内容。
- 青智仪器有限公司遵循持续发展的策略。因此，青智仪器有限公司保留在不预先通知的情况下，对本手册中描述的任何产品进行修改和改进的权力。
- 本手册的内容可能因为修改和改进而产生未经预告的变更。如有不详之处，请参照本手册提供的信息联系。
- 青智仪器有限公司严格实施ISO9001 质量管理体系。本公司产品虽然在严格的品质管理过程控制下制造、出厂，但如果出现不正常事项或意外之处，请通知本公司代理商、或参照本手册提供的信息联系。
- 在产品使用过程中出现任何不正常事项或意外之处，请参照本手册提供的信息联系。
- “青智QINGZHI” 为青岛青智仪器有限公司注册商标。

目录

1、	编程概述	1
1.1、	消息	1
1.2、	命令	3
1.3、	应答	4
1.4、	数据	5
2、	8966C1 功率计命令集	7
2.1、	SCPI 命令列表	7
2.2、	通讯相关命令 (COMMunicate Group)	14
2.3、	显示相关命令 (DISPlay Group)	15
2.4、	输入相关命令 (INPut Group)	19
2.5、	数值相关命令 (NUMeric Group)	29
2.6、	波形相关命令 (WAVEform Group)	34
2.7、	积分相关命令 (INTEGrate Group)	37
2.8、	谐波相关命令 (HARMonics Group)	42
2.9、	测量相关命令 (MEASure Group)	44
2.10、	高速测量相关命令 (HSPeet Group)	47
2.11、	电机测试相关命令 (MOTor Group)	48
2.12、	系统设置相关命令 (SYSTem Group)	69
2.13、	更新相关命令 (RATE Group)	69
2.14、	保持相关命令 (HOLD Group)	70
2.15、	通用命令 (Common Command Group)	70
附录	71
1、	显示模式说明表	71
2、	数值数据项的预设模式表	71
3、	谐波数值数据项的预设模式表	74
4、	<Function>选项列表	75
5、	电机测试数据类型列表	77

1、编程概述

1.1、消息

消息用于控制器和功率计的通信。由控制器发送到功率计的消息称为命令消息，由功率计发回给控制器的消息称为应答消息。功率计接收到含有查询命令的命令消息则立即返回一个应答消息。对于功率计，一个应答消息对应一个命令消息。

1.1.1、命令消息

命令消息的格式如图 1.1 所述：

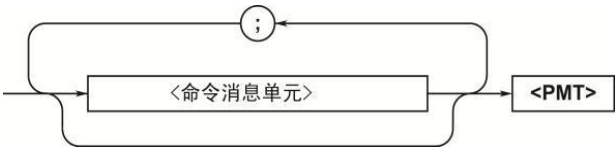


图 1.1 命令消息的格式

(1) 命令消息单元Required

一个命令消息由一个或多个命令消息单元组成，每个命令消息单元对应一个命令，命令消息单元之间用分号“;”分隔，功率计先执行先收到的命令。命令消息单元的语法如图 1.2 所示，示例如图 1.3 所示。命令头指示命令的类型，命令数据则是命令的数据参数，和命令头间以空格间隔，多个命令数据之间则通过逗号分隔，命令头和命令数据示例如图 1.4 所示。

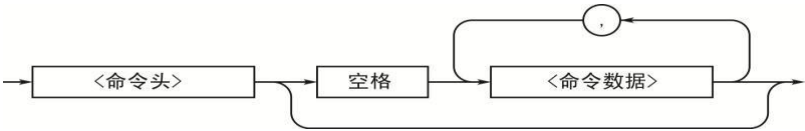


图 1.2 命令消息单元的语法



图 1.3 命令消息单元示例



图 1.4 命令头和命令数据示例

(2) PMT

- PMT 是命令消息的结束符，有如下三种：
- NL（新行）。和 LF（line feed，换行）一样，ASCII 码为“0AH”；
- ^END。紧跟^END 的命令数据是命令消息中的最后一个数据字节；
- NL^END。NL 和 END 消息一起发。

1.1.2、应答消息

应答消息的语法如图 1.5 所示：

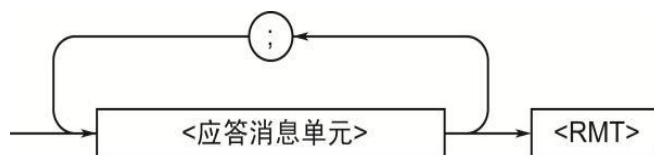


图 1.5 应答消息的语法

(1) 应答消息单元

一个应答消息有一个或多个应答消息单元，每一个应答消息单元对应一个应答，各个应答消息单元以分号分隔，如图 1.6 所示：

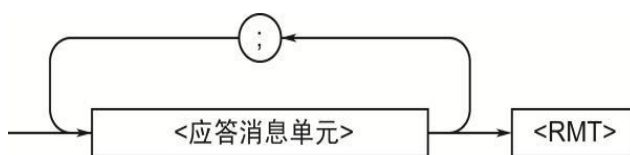


图 1.6 应答消息单元示例

应答消息单元的语法如图 1.7 所示：

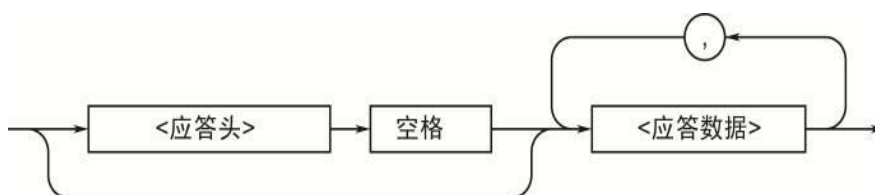


图 1.7 应答消息单元的语法

应答头出现于应答数据之前，应答头和应答数据之间以空格分隔。应答数据包含了应答的内容，多组应答数据之间用逗号“,”分隔如图 1.8 所示：

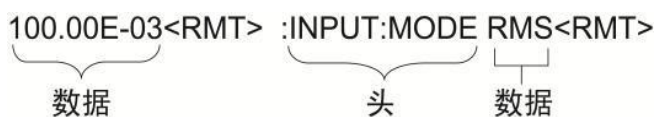


图 1.8 应答头和应答数据的例子

若命令消息里有多个查询请求，功率计将先回应先收到的查询请求。大多数情况下，单个查询返回单个应答，但也有一些查询请求需要返回多个应答；于是会出现这样的情形，第一个应答对应第一个查询，但第 N 个应答不一定对应第 N 个查询。因此，如果用户希望确保能收到每个应答，可使一个命令消息里只有一个查询请求。

(2) RMT

RMT 是一个应答消息结束符，是 NL^END。

1.1.3、注意事项

- (1) 若控制器发送的命令消息不含查询，则可在随后的任意时刻再发送一个命令消息
- (2) 若控制器发送的命令消息含查询，则须在接收完应答消息后才能发送下一个命令消息，否则出错
- (3) 若控制器去接收一个不存在的应答，则产生错误
- (4) 若控制器还没有完成命令消息发送就去接收应答，则产生错误
- (5) 若控制器发送的命令消息不完整，则可能产生错误

1.2、命令

1.2.1、类型

根据命令头格式的不同，控制器发送到功率计的命令可分为三种，如表 1.1 所示：

表1.1 命令头的格式

命令头格式	说明
 通用命令示例: *CLS	IEEE 488.2-1992 定义的命令为通用命令，通用命令头的格式如左所示，通用命令前必须紧跟“*”
 复合命令头示例: :INPut:MODE	部分功率计特有命令须根据其功能，分层次来表示，不同的层次用“:”号隔开，格式和示例如左所示
 单一命令头示例: :HOLD	部分命令的功能是独立的，其命令头格式比较简单，没有层次划分，如左所示

1.2.2、命令的连接

- (1) 不同命令组命令的连接
两个不同命令组命令之间需要用“:”号分隔，“:”号不能省略。
例如，:INTEGrate:MODE NORMal;:INPut:MODE RMS<PMT>。
- (2) 单独命令和其它命令的连接
若单独命令紧跟其它命令，则须在单独命令前加“:”号。
例如，:INTEGrate:MODE NORMal;:HOLD ON<PMT>
- (3) 连接通用命令
通用命令在 IEEE488.2-1992 定义。连接通用命令时，命令之前不需要“:”号
例如，:INTEGrate:MODE NORMal;*CLS;:INTEGrate:TIMer 1,0,0<PMT>

(4) 用<PMT>分隔命令

当用<PMT>分隔两个命令时，将发送两个命令消息。此时，这两个命令即使是同一命令组并具有相同的命令头，也不能省略命令头。

1.2.3、上层查询

在命令组的最高级命令后添加一个“?”，可查询命令组内所有低级查询指令所查询到的信息，例如：INTEGrate?▶▶:INTEGRATE:MODE NORMAL;TIMER 0,0,0;STATE START

上层查询得到的应答信息，也可作为命令消息发回控制器，从而令当时返回的设置信息重新生效。不过需要注意的是，不是所有命令组的信息都会返回，例如对某些上层查询请求，当前没有用到的设置信息就不会返回。

1.2.4、命令头的书写说明

命令头的书写说明：如下所述：

(1) 命令助记符不区分大小写

例如，INPut 可写成 input 或 INPUT

(2) 命令助记符中的小写字母可省略

例如，INPut 也可写为 INPu 或 INP

(3) 命令头后的“?”，用于指示查询功能，用户不可省略“?”

例如，INPut?最短可缩写为 INP?

(4) 若命令助记符后的数值在书写时被省略，则认为该数值为 1

例如，ELEMeNt写成ELEM，即被识别为“ELEMeNt1”

(5) 若写在方框里的命令或参数可省略

例如，[:INPut]:SCALing[:STATe] ON”可被写成“SCAL ON”

但在上层查询命令里，不能省略最后一个方框里的命令或参数

例如，“SCALing?”和“SCALing:STATe?”是不同的

1.3、应答

功率计返回的应答消息有如下两种形式：

由头与数据组成。这类应答消息有命令头，可不加修改就直接作为命令消息使用

例如，:INTEGrate:MODE? ▶▶ :INTEGRATE:MODE NORMAL

仅由数据组成。这类应答消息不能作为命令消息使用，因为没有命令头

例如，:INTEGrate:STATe? ▶▶ RESET

如果需要功率计返回的应答消息不附加命令头，用户可使用“COMMunicate:HEADer”命令来配置功率计。返回的应答头通常省略小写部分和方框内的部分

1.4、数据

1.4.1、概述

数据位于命令头之后，和命令头以空格分隔。数据包含了条件和值，相关声明概述如表1.2 所示：

表 1.2 数据说明

数据	说明	示例
<Decimal>	以十进制表示的值	VT 比的设置▶▶[:INPut]:SCALing:VT 100
<Voltage><Current><Time>	物理量的值	电压量程设置▶▶[:INPut]:VOLTag:e:RANge 150V
<Register>	寄存器值，可用二进制、八进制、十进制、十六进制表示	扩展事件寄存器值▶▶:STATUS:EESE #HFE
<Character data>	预定义的字符串	测量模式选择▶▶[:INPut]:MODE {RMS VMEan DC}
<Boolean>	指示开启或关闭。通常是 ON、OFF 或一个值	将数据设置为保持状态▶▶:HOLD ON
<String data>	任意字符串	返回型号▶▶:SYSTEM:MODEL "8966C1"
<Block data>	包含多个 8 位数值的数据	应答中的测量数据▶▶ #40012ABCDEFGH(IJKL)

1.4.2、乘数和单位符号

用户可用的乘数及其符号见图1.3，乘数符号不区分大小写

表 1.3 乘数及其符号

符号	乘数
EX	10 ¹⁸
PE	10 ¹⁵
T	10 ¹²
G	10 ⁹
MA	10 ⁶
K	10 ³
M	10 ⁻³
U	10 ⁻⁶
N	10 ⁻⁹
P	10 ⁻¹²
F	10 ⁻¹⁵

用户可用的单位符号如表1.4所示：

表 1.4 单位及符号

符号	单位	说明：
V	伏特	电压
A	安培	电流
S	秒	时间

若乘数和单位都省略未写，则默认使用基本单位（V、A、S），功率计返回的应答消息通常使用<NR3>的格式，并且不使用乘数和单位而是使用基本格式。

<Decimal>数值

<Decimal>指示数据是十进制数值。在ANSI X3.42-1975.标准里，用“NR”来表示不同格式的十进制数值，如表1.5 所示：

表 1.5 Decimal 数值

符号	含义	例子
<NR1>	整数	125、- 1、+ 100
<NR2>	定点数字	125.0、-.90、+ 001.
<NR3>	浮点数字	125.0E+0、-9E-1、+.1E4
<NRf>	<NR1>到<NR3>中的任意一种	

相关说明：如下：

- (1) 功率计可接收控制器以NR1~NR3 格式发送的十进制数值；
- (2) 功率计返回的<Decimal>数据数值格式与查询请求中<Decimal>数据的格式一致；
- (3) <NR3>格式中的“+”号可省略，但“-”号不可省略；
- (4) 若输入了超过范围的数据，那么该数据将被识别为范围内与之最接近的值；
- (5) 若数据的有效位数过多，则直接舍弃多余的数据有效位。

<Register> 数据

<Register>指示数据是一个可用二进制、十进制、八进制、十六进制来表示的整数。当一个整数数据的每个位都有特殊含义时，可使用<Register>来表示该数据。

表 1.6 Register 数值

格式	例子
<NRf>	1
#H<十六进制值，包括 0~9 和A~F>	#H0F
#Q<八进制值，由 0~7 组成>	#Q777
#B<二进制值，由 0 和 1 组成>	#B001100

<Register>数值书写不区分大小写，应答消息中的<Register>数值通常使用<NR1>格式。

<Character Data>数据

<Character Data>数据是一个具有特殊含义的字符串，该字符串通常用于表示一个操作或一个功能参数，可供用户选择。<Character Data>数据的书写格式见“第 1.2.4 节命令头的书写说明：”。

<Boolean>数据

<Boolean>数据用于指示 ON 和 OFF 状态，可以是 ON、OFF，也可以是一个整数。

若<Boolean>数据使用整数格式，那么当<Boolean>数据取整后的整数值是 0，则认为是 OFF 状态，否则认为是 ON 状态。应答消息里通常返回 1 来表示 ON，返回 0 来表示 OFF。

<string Data>

<String data>不同于<Character data>，不具有特殊含义，是一个任意的字符串。<String data>必须位于一对单引号或双引号内，如表 1.7 所示。

表 1.7<String data>

格式	例子
<String data>	'ABC' "IEEE488.2-1992"

<Block Data>

<Block data>由 8 个比特位组成，仅用于应答消息。<Block data>的语法如表 1.8 所述。

表 1.8 语法

格式	例子
#N<数据字节数><数据字节序列>	#800000010ABCDEFGHJIJ

对表 1.8 里的格式说明：如下：

- (1) “#N”指示本<Block data>中数据字节数的长度，例如“#800000010ABCDEFGHJIJ”中的 1.8 表“00000010”的长度是 8 个数，而“00000010”是数据字节数
说明：后面的数据字节序列A-J 为 10 个字符数据
- (2) “数据字节数”指示数据中的字节数，并以十进制表示
- (3) “数据字节序列”即实际的数据字节内容，即“ABCDEFGHJIJ”

2、8966C1功率计命令集

2.1、SCPI 命令列表

通讯相关命令 (COMMunicate Group)	功能 (Function)	页码
:COMMunicate?	查询所有通信设置	14
:COMMunicate:HEADer	设置或查询是否在查询应答里返回命令头	14
:COMMunicate:VERBoSe	设置或查询返回的应答信息是使用缩写格式或完整书写格式	14
:COMMunicate:LOCKout	设置或清除本地的按键锁定设置	14
:COMMunicate:REMOte	将此仪器设置为远程或本地模式 (开是远程模式)	14

显示相关命令 (DISPlay Group)	功能 (Function)	页码
:DISPlay?	查询所有显示设置信息	15
:DISPlay:MODE	设置或查询显示模式	15
:DISPlay:NUMeric?	查询所有数值显示设置	15
:DISPlay:NUMeric:FRAMe	设置或查询数值显示的数据部分帧的开/关状态	15
:DISPlay:NUMeric:NORMal?	查询数值显示设置	15
:DISPlay:NUMeric[:NORMal]:ALL?	查询“所有项目”显示模式下数值显示的所有设置	15
:DISPlay:NUMeric[:NORMal]:FORMat	设置或查询数值显示格式	16
:DISPlay:NUMeric[:NORMal]:LIST?	查询列表显示模式中的所有数值显示设置	16
:DISPlay:NUMeric[:NORMal]:MATRix?	查询矩阵显示模式下的所有数值显示设置	16
:DISPlay:NUMeric[:NORMal]:{VAL4 VAL8 VAL16}?	查询 4、8 或 16 项显示模式下的所有数值显示设置	16
:DISPlay:WAVE?	查询所有波形显示设置	16
:DISPlay:WAVE:FORMat	设置或查询波形显示格式	17
:DISPlay:WAVE:TDIV	设置或查询波形显示 TIME/DIV	17
:DISPlay:WAVE:TRIGger?	查询所有波形触发设置	17
:DISPlay:WAVE:TRIGger:LEVel	设置或查询触发级别	17
:DISPlay:WAVE:TRIGger:MODE	设置或查询触发模式	17
:DISPlay:WAVE:TRIGger:SLOPe	设置或查询触发类型	18
:DISPlay:WAVE:TRIGger:SOURce	设置或查询触发源	18
:DISPlay:TREND?	查询所有趋势显示设置	18
:DISPlay:BAR?	查询所有条形图显示设置	18
:DISPlay:VECTor?	查询所有矢量显示设置	18
:DISPlay:HSPeet?	查询所有高速数据捕获显示设置	18

输入相关命令 (INPut Group)	功能 (Function)	页码
:INPut?	查询所有输入元素设置	19
[:INPut]:CFACtor	设置或查询波峰因数	19
[:INPut]:CURREnt?	查询所有电流设置	19
[:INPut]:CURREnt:AUTO?	查询所有单元的电流自动量程开关状态	19
[:INPut]:CURREnt:AUTO[:ALL]	统一设定所有单元的电流自动量程开关状态	19
[:INPut]:CURREnt:AUTO:ELEMent<x>	设置或查询指定单元的电流自动量程开关状态	19
[:INPut]:CURREnt:AUTO:{SIGMA SIGMB SIGMC}	统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)中的所有单元的电流自动量程开/关状态	20
[:INPut]:CURREnt:CONFig?	查询所有单元的电流设置	20
[:INPut]:CURREnt:CONFig[:ALL]	统一设定所有单元的电流设置	20
[:INPut]:CURREnt:CONFig:ELEMent<x>	设置或查询指定单元的电流设置	20
[:INPut]:CURREnt:RANGe?	查询所有单元的电流量程	21

[[:INPut]:CURREnt:RANGe[:ALL]	统一设定所有单元的电流量程	21
[[:INPut]:CURREnt:RANGe:ELEMent<x>	设置或查询指定单元的电流量程	21
[[:INPut]:VOLTage?	查询所有电压设置	21
[[:INPut]:VOLTage:AUTO?	查询所有单元的电压自动量程开关状态	21
[[:INPut]:VOLTage:AUTO[:ALL]	统一设定所有单元的电压自动量程开关状态	22
[[:INPut]:VOLTage:AUTO:ELEMent<x>	设置或查询指定单元的电压自动量程开关状态。	22
[[:INPut]:VOLTage:AUTO:{SIGMA SIGMB SIGMC}	统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)中的所有单元的电压自动量程开/关状态	22
[[:INPut]:VOLTage:CONFig?	查询所有单元的电压设置	22
[[:INPut]:VOLTage:CONFig[:ALL]	统一设定所有单元的电压设置	22
[[:INPut]:VOLTage:CONFig:ELEMent<x>	设置或查询指定单元的电压设置	23
[[:INPut]:VOLTage:RANGe?	查询所有单元的电压量程	23
[[:INPut]:VOLTage:RANGe[:ALL]	统一设定所有单元的电压量程	23
[[:INPut]:VOLTage:RANGe:ELEMent<x>	设置或查询指定单元的电压量程	23
[[:INPut]:SCALing?	查询所有比例设置	23
[[:INPut]:SCALing:STATe	设置或查询比例的开/关状态	24
[[:INPut]:SCALing:{VT CT SFACtor EXT}?	查询所有通道的电压、电流、功率或电流传感器的比例系数	24
[[:INPut]:SCALing:{VT CT SFACtor EXT}[:ALL]	统一设置所有通道的电压、电流、功率或电流传感器的比例系数	24
[[:INPut]:SCALing:{VT CT SFACtor EXT}:ELEMent<x>	设置或查询指定通道的电压、电流、功率或电流传感器的比例系数	24
[[:INPut]:SCALing:{VT CT SFACtor EXT}:{SIGMA SIGMB SIGMC}	统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)的所有通道的电压、电流、功率或电流传感器的比例系数	24
[[:INPut]:FILTer?	查询所有输入滤波设置	24
[[:INPut]:FILTer:LINE?	查询所有通道的线路滤波器	25
[[:INPut]:FILTer[:LINE][:ALL]	统一设置所有通道的线路滤波器	25
[[:INPut]:FILTer[:LINE]:ELEMent<x>	设置或查询指定通道的线路滤波器	25
[[:INPut]:FILTer[:LINE]:{SIGMA SIGMB SIGMC}	统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)的所有通道的线路滤波器	25
[[:INPut]:FILTer[:LINE]:FIR?	查询所有通道的线路滤波器 FIR 类型	25
[[:INPut]:FILTer[:LINE]:FIR[:ALL]	统一设置所有通道的线路滤波器 FIR 类型	26
[[:INPut]:FILTer[:LINE]:FIR:ELEMent<x>	设置或查询指定通道的线路滤波器 FIR 类型	26
[[:INPut]:FILTer[:LINE]:FIR:{SIGMA SIGMB SIGMC}	统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)的所有通道的线路滤波器 FIR 类型	26
[[:INPut]:FILTer[:LINE]:BANK<x>	设置或查询线路滤波器截止频率	26
[[:INPut]:FILTer:FREQuency?	查询所有通道的频率滤波器	26
[[:INPut]:FILTer:FREQuency[:ALL]	统一设置所有通道的频率滤波器	26
[[:INPut]:FILTer:FREQuency:ELEMent<x>	设置或查询指定通道的频率滤波器	27
[[:INPut]:SYNChronize?	查询所有通道的同步源	27
[[:INPut]:SYNChronize[:ALL]	统一设置所有通道的同步源	27
[[:INPut]:SYNChronize:ELEMent<x>	设置或查询指定通道的同步源	27
[[:INPut]:SYNChronize:{SIGMA SIGMB SIGMC}	统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)的所有通道的同步源	27
[[:INPut]:WIRing	设置或查询接线组	28

数值相关命令 (NUMeric Group)	功能 (Function)	页码
:NUMeric?	查询所有数值数据输出设置	29
:NUMeric:FORMat	设置或查询数值数据格式	29
:NUMeric:HOLD	设置或查询数值数据保持状态	29
:NUMeric:NORMal?	查询所有数值数据输出设置	29

:NUMeric[:NORMal]:CLEar	清除数值数据输出项	30
:NUMeric[:NORMal]:DELeTe	删除数值数据输出项	30
:NUMeric[:NORMal]:ITEM<x>	设置或查询指定的数值数据输出项(函数、元素和谐波顺序)	30
:NUMeric[:NORMal]:NUMber	设置或查询传输的数值数据项的个数	31
:NUMeric[:NORMal]:PRESet	预设数值数据输出项模式	31
:NUMeric[:NORMal]:VALue?	查询数值数据	31
:NUMeric:LIST?	查询所有谐波测量数值列表数据输出设置	31
:NUMeric:LIST:CLEar	清除谐波测量数值列表数据输出项目（项目设置为无）	31
:NUMeric:LIST:DELeTe	删除谐波测量数值列表数据输出项	32
:NUMeric:LIST:ITEM<x>	设置或查询指定谐波测量数值列表数据项的输出项	32
:NUMeric:LIST:NUMber	设置或查询传输的数值数据项的个数	32
:NUMeric:LIST:ORDer	设置或查询谐波测量数值列表数据的最大输出谐波阶数	32
:NUMeric:LIST:PRESet	预设谐波测量数值列表数据输出项目模式	32
:NUMeric:LIST:SELEct	设置或查询谐波测量数值列表数据的输出分量	33
:NUMeric:LIST:VALue?	查询谐波测量数值列表数据	33

波形相关命令 (WAVEform Group)	功能 (Function)	页码
:WAVEform?	查询所有波形数据输出设置	34
:WAVEform:MODE	设置和查询波形数据模式	34
:WAVEform:TRACe	设置和查询波形数据绑定	34
:WAVEform:FORMat	设置和查询显示数据的格式	34
:WAVEform:START	设置和查询数据的输出起点	34
:WAVEform:END	设置和查询数据的输出终点	35
:WAVEform:TIME	设置和查询 2048 点波形的通信区间	35
:WAVEform:HOLD	设置或查询所有波形的波形显示数据保持功能	35
:WAVEform:LENGth?	查询指定波形的总点数	36
:WAVEform:SEND?	查询指定的波形显示数据	36
:WAVEform:BYTeorder	设置和查询波形数据(FLOAT 格式)的输出字节顺序	36
:WAVEform:SRATe?	查询当前波形采样率	36

积分相关命令 (INTEGrate Group)	功能 (Function)	页码
:INTEGrate?	查询所有积分设置	37
:INTEGrate:ACAL	设置或查询积分自动校准的开/关状态	37
:INTEGrate:INDependent	设置或查询独立积分的开/关状态	37
:INTEGrate:MODE	设置或查询积分模式	37
:INTEGrate:QMODE?	查询所有通道的电流积分的电流模式	37
:INTEGrate:QMODE[:ALL]	统一设置所有通道的电流积分的电流模式	38
:INTEGrate:QMODE:ELEMeNt<x>	设置或查询指定通道的电流积分的电流模式	38
:INTEGrate:RESet	重置积分	38
:INTEGrate:RTALI:{START END}	设置实时积分模式共同设置所有通道的积分开始或结束时间	38
:INTEGrate:RTIME<x>?	查询实时积分模式的积分开始和结束时间	39
:INTEGrate:RTIME<x>:{START END}	设置或查询实时积分模式的积分开始或结束时间	39
:INTEGrate:START	开始积分	39
:INTEGrate:STATe?	查询积分状态	39
:INTEGrate:STOP	停止积分	40
:INTEGrate:TIMer<x>	设置或查询积分计时器值	40

:INTEGrate:TMALl	统一设置所有通道的积分计时器	40
:INTEGrate:WPTYPE?	查询所有通道的极性(WP+/WP-)的瓦时积分方法	41
:INTEGrate:WPTYPE[:ALL]	统一设置所有通道的极性(WP+/WP-)的瓦时积分方法	41
:INTEGrate:WPTYPE:ELEMENT<x>	设置或查询指定通道的极性(WP+/WP-)的瓦时积分方法	41

谐波相关命令 (HARMonics Group)	功能 (Function)	页码
:HARMonics<x>?	查询所有谐波测量设置	42
:HARMonics:CONFigure?	查询所有通道的谐波测量	42
:HARMonics:CONFigure[:ALL]	统一设置所有通道的谐波测量	42
:HARMonics:CONFigure:ELEMENT<x>	设置或查询指定通道的谐波测量	42
:HARMonics:CONFigure:{SIGMA SIGMB SIGMC}	统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)所有通道的谐波测量	42
:HARMonics<x>:ORDER	设置或查询所分析的最大和最小谐波阶数	43
:HARMonics<x>:PLLSource	设置或查询 PLL 源	43
:HARMonics<x>:THD	设置或查询用于计算 THD (总谐波失真) 的公式	43
:HARMonics<x>:POINT	设置或查询用于谐波测量的 FFT 点数	43

测量相关命令 (MEASure Group)	功能 (Function)	页码
:MEASure?	查询所有测量设置	44
:MEASure:AVERaging?	查询所有平均设置	44
:MEASure:AVERaging:COUNT	设置或查询平均系数	44
:MEASure:AVERaging[:STATe]	设置或查询平均的开/关状态	44
:MEASure:AVERaging:TYPE	设置或查询平均类型	44
:MEASure:MHOLD	设置或查询最大值保持的开/关状态	45
:MEASure:PC?	查询所有 Pc 修正设置	45
:MEASure:PC:IEC	设置或查询 Pc 修正公式	45
:MEASure:PC:P<x>	设置或查询 Pc 修正公式参数	45
:MEASure:PHASe	设置或查询相位差的显示格式	45
:MEASure:SAMPLing	设置或查询采样频率	46
:MEASure:SFORMula	设置或查询用于计算 S (视在功率) 的方程	46
:MEASure:SQFormula	设置或查询用于计算 S (视在功率) 和 Q (无功功率) 的方程。	46
:MEASure:SYNChronize	设置或查询同步测量模式	46

高速测量相关命令 (HSPeet Group)	功能 (Function)	页码
:HSPeet?	查询所有高速测量设置	47
:HSPeet:STATe	设置或查询高速测量的开/关状态	47
:HSPeet:TRIGger	设置或查询高速测量的触发模式	47
:HSPeet:SYNChronize	设置或查询高速同步源	47
:HSPeet[:TiME]:{MIN MAX}	设置或查询高速测量单次测量的最小/最大(超时)时间	47

电机相关命令 (MOTor Group)	功能 (Function)	页码
:MOTor?	查询所有电机设置	48
:MOTor:STARtup?	查询所有电机启动测试设置	48
:MOTor:STARtup:MODE	设置或查询启动测试模式(响应周期)	48
:MOTor:STARtup:TiMer	设置或查询启动测试持续时间	48
:MOTor:STARtup:RTiMe?	查询启动测试已运行时间	48
:MOTor:STARtup:STATe?	查询启动测试状态	49

:MOTor:STARtup:STARt	开始启动测试	49
:MOTor:STARtup:STOP	停止启动测试	49
:MOTor:STARtup:RESet	重置启动测试	49
:MOTor:STARtup:LENGth?	查询启动测试数据采集的数量	49
:MOTor:STARtup:TRACe	设置或查询启动测试的数据绑定	49
:MOTor:STARtup:SEND?	查询启动测试数据	50
:MOTor:RLOCKed?	查询所有电机堵转测试设置	51
:MOTor:RLOCKed:MODE	设置或查询堵转测试模式(响应周期)	51
:MOTor:RLOCKed:TIMer	设置或查询堵转测试持续时间	51
:MOTor:RLOCKed:RTIME?	查询堵转测试已运行时间	51
:MOTor:RLOCKed:STATe?	查询堵转测试状态	51
:MOTor:RLOCKed:STARt	开始堵转测试	51
:MOTor:RLOCKed:STOP	停止堵转测试	52
:MOTor:RLOCKed:RESet	重置堵转测试	52
:MOTor:RLOCKed:LENGth?	查询堵转测试数据采集的数量	52
:MOTor:RLOCKed:TRACe	设置或查询堵转测试的数据绑定	52
:MOTor:RLOCKed:SEND?	查询堵转测试数据	52
:MOTor:VOLStore?	查询所有电机电压存储设置	53
:MOTor:VOLStore:MODE	设置或查询电压存储模式(响应周期)	53
:MOTor:VOLStore:TIMer	设置或查询电压存储持续时间	53
:MOTor:VOLStore:RTIME?	查询电压存储已运行时间	53
:MOTor:VOLStore:STATe?	查询电压存储状态	53
:MOTor:VOLStore:STARt	开始电压存储	53
:MOTor:VOLStore:STOP	停止电压存储	54
:MOTor:VOLStore:RESet	重置电压存储	54
:MOTor:VOLStore:LENGth?	查询电压存储数据采集的数量	54
:MOTor:VOLStore:TRACe	设置或查询电压存储的数据绑定	54
:MOTor:VOLStore:SEND?	查询电压存储数据	54
:MOTor:DATRecord?	查询所有电机数据记录设置	55
:MOTor:DATRecord:MODE	设置或查询数据记录模式(响应周期)	55
:MOTor:DATRecord:TIMer	设置或查询数据记录持续时间	55
:MOTor:DATRecord:RTIME?	查询数据记录已运行时间	55
:MOTor:DATRecord:STATe?	查询数据记录状态	55
:MOTor:DATRecord:STARt	开始数据记录	55
:MOTor:DATRecord:STOP	停止数据记录	56
:MOTor:DATRecord:RESet	重置数据记录	56
:MOTor:DATRecord:LENGth?	查询数据记录数据采集的数量	56
:MOTor:DATRecord:TRACe	设置或查询数据记录的数据绑定	56
:MOTor:DATRecord:SEND?	查询数据记录数据	56
:MOTor:WAVRecord?	查询所有电机录波测试设置	57
:MOTor:WAVRecord:MODE	设置或查询录波测试模式(每秒采集的点数)	57
:MOTor:WAVRecord:TIMer	设置或查询录波测试持续时间	57
:MOTor:WAVRecord:RTIME?	查询录波测试已运行时间	57
:MOTor:WAVRecord:STATe?	查询录波测试状态	57
:MOTor:WAVRecord:STARt	启动录波	58
:MOTor:WAVRecord:STOP	停止录波	58

:MOTor:WAVRecord:RESet	重置录波	58
:MOTor:WAVRecord:LENGth?	查询录波测试数据采集的数量	58
:MOTor:WAVRecord:TRACe	设置或查询录波数据的数据绑定	58
:MOTor:WAVRecord:SEND?	查询录波测试数据	59
:MOTor:TRANSient?	查询所有电机瞬态测试设置	59
:MOTor:TRANSient:MODE	设置或查询瞬态测试模式(响应周期)	59
:MOTor:TRANSient:TIMer	设置或查询瞬态测试持续时间	59
:MOTor:TRANSient:RTIME?	查询瞬态测试已运行时间	60
:MOTor:TRANSient:STATe?	查询瞬态测试状态	60
:MOTor:TRANSient:START	开始瞬态测试	60
:MOTor:TRANSient:STOP	停止瞬态测试	60
:MOTor:TRANSient:RESet	重置瞬态测试	60
:MOTor:TRANSient:LENGth?	查询瞬态测试数据采集的数量	60
:MOTor:TRANSient:TRACe	设置或查询瞬态测试的数据绑定	61
:MOTor:TRANSient:SEND?	查询瞬态测试数据	61
:MOTor<x>?	查询指定电机参数设置	62
:MOTor<x>:SPEed?	查询所有电机转速设置	62
:MOTor<x>:SPEed:TYPE	设置或查询转速信号输入类型	62
:MOTor<x>:SPEed:SCALIng	设置或查询转速计算比例系数	62
:MOTor<x>:SPEed:RANGe?	查询转速信号的输入量程(模拟输入)	63
:MOTor<x>:SPEed:LSCale?	查询所有转速线性比例设置(模拟输入)	63
:MOTor<x>:SPEed:LSCale:KVALue	设置或查询转速线性比例的斜率(模拟输入)	63
:MOTor<x>:SPEed:LSCale:DVALue	设置或查询转速线性比例的偏移量(模拟输入)	63
:MOTor<x>:SPEed:PRANGe	设置或查询转速脉冲输入量程(脉冲输入)	64
:MOTor<x>:SPEed:PULSe	设置或查询转速每转脉冲的个数(脉冲输入)	64
:MOTor<x>:TORQue?	查询所有电机扭矩设置	64
:MOTor<x>:TORQue:TYPE	设置或查询扭矩信号输入类型	64
:MOTor<x>:TORQue:SCALIng	设置或查询扭矩计算比例系数	65
:MOTor<x>:TORQue:RANGe?	查询扭矩信号的电压范围(模拟输入)	65
:MOTor<x>:TORQue:LSCale?	查询所有扭矩线性比例设置(模拟输入)	65
:MOTor<x>:TORQue:LSCale:KVALue	设置或查询扭矩线性比例的斜率	65
:MOTor<x>:TORQue:LSCale:DVALue	设置或查询扭矩线性比例的偏移量	66
:MOTor<x>:TORQue:PRANGe	设置或查询扭矩脉冲输入量程(脉冲输入)	66
:MOTor<x>:TORQue:RATE?	查询所有扭矩信号的额定值设置(脉冲输入)	66
:MOTor<x>:TORQue:RATE:{UPPer LOWer}	设置或查询扭矩信号的额定值和额定频率(脉冲输入)	66
:MOTor<x>:TORQue:SCHool	扭矩校零	67
:MOTor<x>:TORQue:SCHool:FREQ	设置或查询扭矩校零频率	67
:MOTor<x>:FILTer	设置或查询电机线路滤波	67
:MOTor<x>:PM:SCALIng	设置或查询电机机械功率比例系数	67
:MOTor<x>:POLE	设置或查询电机极数	68
:MOTor<x>:SYNChronize	设置或查询转速和扭矩计算的同步源	68
:MOTor<x>:SSPeed	设置或查询电机同步转速计算的频率测量源	68

系统设置相关命令 (SYSTEM Group)		功能 (Function)	页码
:SYSTEM:RESolution		设置或查询数值数据显示分辨率	69
更新相关命令 (RATE Group)		功能 (Function)	页码
:RATE[:RATE]		设置或查询数据更新间隔	69
:RATE:AUTO?		查询数据更新间隔设置为自动时的所有适用设置	69
:RATE:AUTO:TIMEout		设置或查询数据更新间隔设置为自动时的超时时间	69
保持相关命令 (HOLD Group)		功能 (Function)	页码
:HOLD		设置或者查询数据保持状态的开启或关闭	70
通用命令 (Common Command Group)		功能 (Function)	页码
*IDN?		查询仪表型号	70
*CLS		清除所有寄存器	70
*RST		初始化仪表	70

2.2、通讯相关命令 (COMMunicate Group)

:COMMunicate?

- 功能： 查询所有通信设置
- 语法： :COMMunicate?
- 实例： :COMMunicate? ►► :COMMUNICATE:HEADER 1;VERBOSE 1;LOCKOUT 0;REMOTE 0

:COMMunicate:HEADer

- 功能： 设置或查询是否在查询应答里返回命令头
- 语法： :COMMunicate:HEADer {<Boolean>}
:COMMunicate:HEADer?
- 实例： :COMMunicate:HEADer ON ►►
:COMMUNICATE:Header? ►► :COMMUNICATE:HEADER 1

:COMMunicate:VERBose

- 功能： 设置或查询返回的应答信息是使用缩写格式或完整书写格式
- 语法： :COMMunicate:VERBose {<Boolean>}
:COMMunicate:VERBose?
- 实例： :COMMunicate:VERBose ON ►►
:COMMunicate:VERBose? ►► :COMMUNICATE:VERBOSE 1
:COMMunicate:VERBose OFF ►►
:COMMunicate:VERBose? ►► :COMM:VERB 0

:COMMunicate:LOCKout

- 功能： 设置或清除本地的按键锁定设置
- 语法： :COMMunicate:LOCKout {<Boolean>}
:COMMunicate:LOCKout?
- 实例： :COMMunicate:LOCKout OFF ►►
:COMMunicate:LOCKout? ►► :COMMUNICATE:LOCKOUT 0

:COMMunicate:REMOte

- 功能： 将此仪器设置为远程或本地模式（开是远程模式）
- 语法： :COMMunicate:REMOte {<Boolean>}
:COMMunicate:REMOte?
- 实例： :COMMunicate:REMOte OFF ►►
:COMMunicate:REMOte? ►► :COMMUNICATE:REMOTE 0

2.3、显示相关命令 (DISPlay Group)

:DISPlay?

- 功能： 查询所有显示设置信息
- 语法： :DISPlay?
- 实例： :DISPlay? ►► :DISPLAY:BAR:FORMAT TRIAD;ITEM1 I,1;ITEM2 I,2;ITEM3 I,3;MODE NUMERIC;
:DISPLAY:NUMERIC ;FRAME ;NORMAL

:DISPlay:MODE

- 功能： 设置或查询显示模式
- 语法： :DISPlay:MODE {NUMeric|WAVE|TRENd|BAR|VECTor|NWAVE|NTREnd|NBAR|NVECTor|WNUMeric|WTRend|WBAR
|WVECTor|TNUMeric|TWAVE|TBAR|TVECTor|HSPeet}
:DISPlay:MODE?
- 实例： :DISPlay:MODE NUMeric ►►
:DISPlay:MODE? ►► :DISPLAY:MODE NUMERIC
- 说明： 详细参数说明请参考[附录：显示模式说明表](#)

:DISPlay:NUMeric?

- 功能： 查询所有数值显示设置
- 语法： :DISPlay:NUMeric?
- 实例： :DISPlay:NUMeric? ►►

:DISPlay:NUMeric:FRAMe

- 功能： 设置或查询数值显示的数据部分帧的开/关状态
- 语法： :DISPlay:NUMeric:FRAMe {<Boolean>}
:DISPlay:NUMeric:FRAMe?
- 实例： :DISPlay:NUMeric:FRAMe ON ►►
:DISPlay:NUMeric:FRAMe? ►►

:DISPlay:NUMeric:NORMal?

- 功能： 查询数值显示设置
- 语法： :DISPlay:NUMeric:NORMal?
- 实例： :DISPlay:NUMeric:NORMal? ►► :DISPLAY:NUMERIC:NORMAL
- 说明： 返回与当前数值显示模式(:DISPlay:NUMeric[:NORMal]:FORMat)对应的所有设置

:DISPlay:NUMeric[:NORMal]:ALL?

- 功能： 查询所有项目显示模式下数值显示的所有设置
- 语法： :DISPlay:NUMeric[:NORMal]:ALL?
- 实例： :DISPlay:NUMeric:NORMal:ALL? ►►

:DISPlay:NUMeric[:NORMal]:FORMat

功能： 设置或查询数值显示格式

语法： :DISPlay:NUMeric[:NORMal]:FORMat{VAL4|VAL8|VAL16|MATRIX|ALL|SINGLE|DUAL|CUSTom}
:DISPlay:NUMeric[:NORMal]:FORMat?

实例： :DISPlay:NUMeric:NORMal:FORMat VAL4 ►►

:DISPlay:NUMeric:NORMal:FORMat? ►► :DISPlay:NUMeric:NORMal:FORMat VAL4

说明： 对于每种格式，数值数据的显示方式如下：

- {VAL4|VAL8|VAL16}:数值显示项按项目编号顺序显示。(这些选项中的数值表示在单个屏幕/页面上显示的项目数)
- MATRIX(矩阵):所选函数按元素顺序显示
- ALL(全部):所有函数按元素顺序显示
- SINGLE(单列表):通过将数据分成奇偶谐波顺序来列出一个列表显示项
- DUAL(双列表):两个列表显示项按谐波顺序排列
- CUSTom(自定义):显示指定的数值显示项置

:DISPlay:NUMeric[:NORMal]:LIST?

功能： 查询列表显示模式中的所有数值显示设置

语法： :DISPlay:NUMeric[:NORMal]:LIST?

实例： :DISPlay:NUMeric:NORMal:LIST? ►►

:DISPlay:NUMeric[:NORMal]:MATRix?

功能： 查询矩阵显示模式下的所有数值显示设置

语法： :DISPlay:NUMeric[:NORMal]:MATRix?

实例： :DISPlay:NUMeric[:NORMal]:MATRix? ►►

:DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}?

功能： 查询 4、8 或 16 项显示模式下的所有数值显示设置

语法： :DISPlay:NUMeric[:NORMal]:{VAL4|VAL8|VAL16}?

实例： :DISPlay:NUMeric[:NORMal]:VAL4? ►►

:DISPlay:WAVE?

功能： 查询所有波形显示设置

语法： :DISPlay:WAVE?

实例： :DISPlay:WAVE? ►►

:DISPlay:WAVE:FORMat

功能： 设置或查询波形显示格式

语法： :DISPlay:WAVE:FORMat {SINGle|DUAL|TRIad|QUAD|HEXa}

:DISPlay:WAVE:FORMat?

实例： :DISPlay:WAVE:FORMat SINGle ▶▶

:DISPlay:WAVE:FORMat? ▶▶ :DISPlay:WAVE:FORMat SINGle

:DISPlay:WAVE:TDIV

功能： 设置或查询波形显示 TIME/DIV

语法： :DISPlay:WAVE:TDIV {AUTO|<Time>}

:DISPlay:WAVE:TDIV?

<Time> = 0.25、0.5、1、2、4、8、16、32、64、128(MS)

实例： :DISPlay:WAVE:TDIV 0.25MS ▶▶

:DISPlay:WAVE:TDIV? ▶▶ :DISPlay:WAVE:TDIV 2.50E-04

:DISPlay:WAVE:TRIGger?

功能： 查询所有波形触发设置

语法： :DISPlay:WAVE:TRIGger?

实例： :DISPlay:WAVE:TRIGger? ▶▶ :DISPlay:WAVE:TRIGger:LEVEL 0.0;MODE AUTO;SLOPE RISE;SOURCE U1

:DISPlay:WAVE:TRIGger:LEVel

功能： 设置或查询触发级别

语法： :DISPlay:WAVE:TRIGger:LEVel {<NRf>}

:DISPlay:WAVE:TRIGger:LEVel?

<NRf> = -100.0 to 100.0(%)

实例： :DISPlay:WAVE:TRIGger:LEVel 0 ▶▶

:DISPlay:WAVE:TRIGger:LEVel? ▶▶ :DISPlay:WAVE:TRIGger:LEVel 0.0

说明： 根据屏幕上显示的满量程值的百分比设置该值

:DISPlay:WAVE:TRIGger:MODE

功能： 设置或查询触发模式

语法： :DISPlay:WAVE:TRIGger:MODE {AUTO|NORMal|OFF}

:DISPlay:WAVE:TRIGger:MODE?

AUTO(自动)、NORMal(常规)、OFF(关闭)

实例： :DISPlay:WAVE:TRIGger:MODE AUTO ▶▶

:DISPlay:WAVE:TRIGger:MODE? ▶▶ :DISPlay:WAVE:TRIGger:MODE AUTO

说明： 如果启用波形显示并且触发模式设置为自动或常规，则数据更新间隔取决于触发操作

:DISPlay:WAVE:TRIGger:SLOPe

功能： 设置或查询触发类型

语法： :DISPlay:WAVE:TRIGger:SLOPe {RISE|FALL|BOTH}

:DISPlay:WAVE:TRIGger:SLOPe?

RISE(上升沿触发)、FALL(下降沿触发)、BOTH(双沿触发，过零触发)

实例： :DISPlay:WAVE:TRIGger:SLOPe RISE ►►

:DISPlay:WAVE:TRIGger:SLOPe? ►► :DISPLAY:WAVE:TRIGGER:SLOPE RISE

:DISPlay:WAVE:TRIGger:SOURce

功能： 设置或查询触发源

语法： :DISPlay:WAVE:TRIGger:SOURce {U<x>|I<x>}

:DISPlay:WAVE:TRIGger:SOURce?

<x> = 1 to 6 (element)

实例： :DISPlay:WAVE:TRIGger:SOURce U1 ►►

:DISPlay:WAVE:TRIGger:SOURce? ►► :DISPLAY:WAVE:TRIGGER:SOURCE U1

:DISPlay:TREND?

功能： 查询所有趋势显示设置

语法： :DISPlay:TREND?

实例： :DISPlay:TREND? ►►

:DISPlay:BAR?

功能： 查询所有条形图显示设置

语法： :DISPlay:BAR?

实例： :DISPlay:BAR? ►► :DISPLAY:BAR:FORMAT TRIAD

:DISPlay:VECTor?

功能： 查询所有矢量显示设置

语法： :DISPlay:VECTor?

实例： :DISPlay:VECTor? ►►

:DISPlay:HSPeet?

功能： 查询所有高速数据捕获显示设置

语法： :DISPlay:HSPeet?

实例： :DISPlay:HSPeet? ►►

2.4、输入相关命令 (INPut Group)

:INPut?

功能： 查询所有输入元素设置

语法： :INPut?

实例： :INPut? ►►

[[:INPut]:CFACtor

功能： 设置或查询波峰因数

语法： [[:INPut]:CFACtor {<NRf>}

[[:INPut]:CFACtor?

<NRf> = 3,6

实例： :INPut:CFACtor 3 ►►

:INPut:CFACtor? ►► :INPUT:CFACtor 3

[[:INPut]:CURRent?

功能： 查询所有电流设置

语法： [[:INPut]:CURRent?

实例： :INPut:CURRent? ►►

[[:INPut]:CURRent:AUTO?

功能： 查询所有单元的电流自动量程开关状态

语法： [[:INPut]:CURRent:AUTO?

实例： :INPut:CURRent:AUTO? ►► :INPUT:CURRENT:AUTO:ELEMENT1 1;ELEMENT2 1;ELEMENT3 1;ELEMENT4 1;

ELEMENT5 1;ELEMENT6 1

[[:INPut]:CURRent:AUTO[:ALL]

功能： 统一设定所有单元的电流自动量程开关状态

语法： [[:INPut]:CURRent:AUTO[:ALL] {<Boolean>}

实例： :INPut:CURRent:AUTO:ALL OFF ►►

[[:INPut]:CURRent:AUTO:ELEMeNt<x>

功能： 设置或查询指定单元的电流自动量程开关状态

语法： [[:INPut]:CURRent:AUTO:ELEMeNt<x> {<Boolean>}

[[:INPut]:CURRent:AUTO:ELEMeNt<x>?

<x> = 1 to 6 (element)

实例： :INPut:CURRent:AUTO:ELEMeNt1 ON ►►

:INPut:CURRent:AUTO:ELEMeNt1? ►► :INPUT:CURRENT:AUTO:ELEMENT1 1

[[:INPut]:CURRent:AUTO:{SIGMA|SIGMB|SIGMC}]

- 功能:** 统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)中的所有单元的电流自动量程开/关状态
- 语法:** [[:INPut]:CURRent:AUTO:{SIGMA|SIGMB|SIGMC}] {<Boolean>}
- 实例:** [[:INPut]:CURRent:AUTO:SIGMA ON ▶▶
- 说明:** 如果连接系统设置([[:INPut]: WIRing])导致相应的连接单元(ΣA 、 ΣB 或 ΣC)不存在, 则 SIGMA、SIGMB 或 SIGMC 无效

[[:INPut]:CURRent:CONFig?]

- 功能:** 查询所有单元的电流设置
- 语法:** [[:INPut]:CURRent:CONFig?]
- 实例:** :INPut:CURRent:CONFig? ▶▶

[[:INPut]:CURRent:CONFig[:ALL]]

- 功能:** 统一设定所有单元的电流设置
- 语法:** [[:INPut]:CURRent:CONFig[:ALL]] {ALL|<Current>[,<Current>][,<Current>]...}
- ALL = 所有范围都有效
- 50A 量程**
- 当波峰系数设置为 3 时: <Current> = 200mA、500mA、1A、2A、5A、10A、20A、50A
- 当波峰系数设置为 6 时: <Current> = 100mA、250mA、500mA、1A、2.5A、5A、10A、25A
- 5A 量程**
- 当波峰系数设置为 3 时: <Current> = 20mA、50mA、100mA、200mA、500mA、1A、2A、5A
- 当波峰系数设置为 6 时: <Current> = 10mA、25mA、50mA、100mA、250mA、500mA、1A、2.5A
- 实例:** :INPut:CURRent:CONFig:ALL ALL ▶▶
- :INPut:CURRent:CONFig:ALL 10A,5A,2A,1A,500mA,200mA ▶▶
- 说明:** 在参数中, 列出所有需要使能的电流范围。要启用所有范围, 请指定参数"ALL"

[[:INPut]:CURRent:CONFig:ELEMeNt<x>]

- 功能:** 设置或查询指定单元的电流设置
- 语法:** [[:INPut]:CURRent:CONFig:ELEMeNt<x>] {ALL|<Current>[,<Current>][,<Current>]...}
- [[:INPut]:CURRent:CONFig:ELEMeNt<x>?]
- <x> = 1 to 6 (element)
- ALL 和 <Current> = 参考 [[:INPut]:CURRent:CONFig[:ALL]]
- 实例:** :INPut:CURRent:CONFig:ELEMeNt1 ALL ▶▶
- :INPut:CURRent:CONFig:ELEMeNt1? ▶▶ :INPut:CURRent:CONFig:ELEMeNt1 ALL
- :INPut:CURRent:CONFig:ELEMeNt1 10A,5A,2A,1A,500mA,200mA ▶▶
- :INPut:CURRent:CONFig:ELEMeNt1? ▶▶
- :INPut:CURRent:CONFig:ELEMeNt1 1.0E+01,5.0E+00,2.0E+00,1.0E+00,5.0E-01,2.0E-01

[[:INPut]:CURRent:RANGe?

功能: 查询所有单元的电流量程

语法: [[:INPut]:CURRent:RANGe?

实例: :INPut:CURRent:RANGe? ►► :INPUT:CURRENT:RANGE:ELEMENT1 2.0E-01;ELEMENT2 2.0E-01;
ELEMENT3 2.0E-01;ELEMENT4 2.0E-01;ELEMENT5 2.0E-01;ELEMENT6 2.0E-01

[[:INPut]:CURRent:RANGe[:ALL]

功能: 统一设定所有单元的电流量程

语法: [[:INPut]:CURRent:RANGe[:ALL] {<Current>}
<Current> = 参考 [[:INPut]:CURRent:CONFig[:ALL]

实例: :INPut:CURRent:RANGe:ALL 50A ►►

[[:INPut]:CURRent:RANGe:ELEMeNt<x>

功能: 设置或查询指定单元的电流量程

语法: [[:INPut]:CURRent:RANGe:ELEMeNt<x> {<Current>}
[[:INPut]:CURRent:RANGe:ELEMeNt<x>?
<x> = 1 to 6 (element)
<Current> = 参考 [[:INPut]: CURRent:CONFig[:ALL]

实例: :INPut:CURRent:RANGe:ELEMeNt1 50A ►►
:INPut:CURRent:RANGe:ELEMeNt1? ►► :INPUT:CURRENT:RANGE: ELEMENT1 5.0E+01

[[:INPut]:VOLTage?

功能: 查询所有电压设置

语法: [[:INPut]:VOLTage?

实例: :INPut:VOLTage? ►► :INPUT:VOLTAGE:RANGE:ELEMENT1 1.5E+01;ELEMENT2 1.5E+01;ELEMENT3 1.5E+01;
ELEMENT4 1.5E+01;ELEMENT5 1.5E+01;ELEMENT6 1.5E+01;:INPUT:VOLTAGE:AUTO:ALL 1;
:INPUT:VOLTAGE:CONFig:ELEMENT1 ALL;ELEMENT2 ALL;ELEMENT3 ALL;ELEMENT4 ALL;
ELEMENT5 ALL;ELEMENT6 ALL 1

[[:INPut]:VOLTage:AUTO?

功能: 查询所有单元的电压自动量程开关状态

语法: [[:INPut]:VOLTage:AUTO?

实例: :INPut:VOLTage:AUTO? ►► :INPUT:VOLTAGE:AUTO:ELEMENT1 1;ELEMENT2 1;ELEMENT3 1;ELEMENT4 1;
ELEMENT5 1;ELEMENT6 1

[[:INPut]:VOLTage:AUTO[:ALL]

- 功能：** 统一设定所有单元的电压自动量程开关状态
- 语法：** [[:INPut]:VOLTage:AUTO[:ALL] {<Boolean>}
- 实例：** :INPut:VOLTage:AUTO:ALL ON ▶▶

[[:INPut]:VOLTage:AUTO:ELEMeNt<x>

- 功能：** 设置或查询指定单元的电压自动量程开关状态。
- 语法：** [[:INPut]:VOLTage:AUTO:ELEMeNt<x> {<Boolean>}
- [[:INPut]:VOLTage:AUTO:ELEMeNt<x>?
- <x> = 1 to 6 (element)
- 实例：** :INPut:VOLTage:AUTO:ELEMeNt1 ON ▶▶
- :INPut:VOLTage:AUTO:ELEMeNt1? ▶▶ :INPUT:VOLTAGE:AUTO:ELEMENT1 1

[[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB|SIGMC}

- 功能：** 统一设置属于指定接线组(ΣA 、 ΣB 或 ΣC)中的所有单元的电压自动量程开/关状态
- 语法：** [[:INPut]:VOLTage:AUTO:{SIGMA|SIGMB|SIGMC} {<Boolean>}
- 实例：** :INPut:VOLTage:AUTO:SIGMA ON ▶▶
- 说明：** 如果接线组设置([[:INPut]:WIRing])导致相应的连接单元(ΣA 、 ΣB 或 ΣC)不存在，则 SIGMA、SIGMB 或 SIGMC 无效

[[:INPut]:VOLTage:CONFIg?

- 功能：** 查询所有单元的电压设置
- 语法：** [[:INPut]:VOLTage:CONFIg?
- 实例：** :INPut:VOLTage:CONFIg? ▶▶ :INPUT:VOLTAGE:CONFIG:ELEMENT1 ALL;ELEMENT2 ALL;ELEMENT3 ALL;
- ELEMENT4 ALL;ELEMENT5 ALL;ELEMENT6 ALL

[[:INPut]:VOLTage:CONFIg[:ALL]

- 功能：** 统一设定所有单元的电压设置
- 语法：** [[:INPut]:VOLTage:CONFIg[:ALL] {ALL|<Voltage>[,<Voltage>][,<Voltage>]...}?
ALL = 所有范围都有效
当波峰系数设置为 3 时: <Voltage> = 15V、30V、60V、100V、150V、300V、600V、1000V
当波峰系数设置为 6 时: <Voltage> = 7.5V、15V、30V、50V、75V、150V、300V、500V
- 实例：** :INPut:VOLTage:CONFIg:ALL ALL ▶▶
- :INPut:VOLTage:CONFIg:ALL 1000V,300V,100V,30V,15V ▶▶

[[:INPut]:VOLTage:CONFig:ELEMeNt<x>

功能： 设置或查询指定单元的电压设置

语法： [[:INPut]:VOLTage:CONFig:ELEMeNt<x> {ALL|<Voltage>[,<Voltage>][,<Voltage>]...}

[[:INPut]:VOLTage:CONFig:ELEMeNt<x>?

<x> = 1 to 6 (element)

ALL = 所有范围都有效

<Voltage> = 参考 [[:INPut]:VOLTage:CONFig[:ALL]

实例： :INPut:VOLTage:CONFig:ELEMeNt1 ALL ►►

:INPut:VOLTage:CONFig:ELEMeNt1? ►► :INPUT:VOLTAGE:CONFIG:ELEMENT1 ALL

:INPut:VOLTage:CONFig:ELEMeNt1 1000V,300V,100V,30V,15V ►►

:INPut:VOLTage:CONFig:ELEMeNt1? ►► :INPUT:VOLTAGE:CONFIG:ELEMENT1 1.0E+03,

3.0E+02,1.0E+02,3.0E+01,1.5E+01

[[:INPut]:VOLTage:RANGe?

功能： 查询所有单元的电压量程

语法： [[:INPut]:VOLTage:RANGe?

实例： :INPut:VOLTage:RANGe? ►►

[[:INPut]:VOLTage:RANGe[:ALL]

功能： 统一设定所有单元的电压量程

语法： [[:INPut]:VOLTage:RANGe[:ALL] {<Voltage>}}

<Voltage> = 参考 [[:INPut]:VOLTage:CONFig[:ALL]

实例： :INPut:VOLTage:RANGe:ALL 1000V ►►

[[:INPut]:VOLTage:RANGe:ELEMeNt<x>

功能： 设置或查询指定单元的电压量程

语法： [[:INPut]:VOLTage:RANGe: ELEMeNt<x> {<Voltage>}

[[:INPut]:VOLTage:RANGe:ELEMeNt<x>?

<x> = 1 to 6 (element)

<Voltage> = 参考 [[:INPut]:VOLTage:CONFig[:ALL]

实例： :INPut:VOLTage:RANGe: ELEMeNt1 1000V ►►

:INPut:VOLTage:RANGe: ELEMeNt1? ►► :INPUT:VOLTAGE:RANGE: ELEMENT1 1.0E+03

[[:INPut]:SCALIng?

功能： 查询所有比例设置

语法： [[:INPut]:SCALIng?

实例： :INPut:SCALIng? ►►

[[:INPut]:SCALing:STATe

- 功能：** 设置或查询比例的开/关状态
- 语法：** [[:INPut]:SCALing:STATe {<Boolean>}
[[:INPut]:SCALing:STATe?
- 实例：** :INPut:SCALing:STATe ON ►►
:INPut:SCALing:STATe? ►► :INPUT:SCALING:STATE 1

[[:INPut]:SCALing:{VT|CT|SFACtor|EXT}?

- 功能：** 查询所有通道的电压、电流、功率或电流传感器的比例系数
- 语法：** [[:INPut]:SCALing:{VT|CT|SFACtor|EXT}?
[[:INPut]:SCALing:VT? ►► :INPUT:SCALING:VT:ELEMENT1 1.0000,ELEMENT2 1.0000,ELEMENT3 1.0000,
ELEMENT4 1.0000,ELEMENT5 1.0000,ELEMENT6 1.0000

[[:INPut]:SCALing:{VT|CT|SFACtor|EXT}[:ALL]

- 功能：** 统一设置所有通道的电压、电流、功率或电流传感器的比例系数
- 语法：** [[:INPut]:SCALing:{VT|CT|SFACtor|EXT}[:ALL] {<NRf>}
{<NRf>} = 0.0001 to 9999.9999
- 实例：** :INPUT:SCALING:VT:ALL 1 ►►

[[:INPut]:SCALing:{VT|CT|SFACtor|EXT}:ELEMent<x>

- 功能：** 设置或查询指定通道的电压、电流、功率或电流传感器的比例系数
- 语法：** [[:INPut]:SCALing:{VT|CT|SFACtor|EXT}:ELEMent<x> {<NRf>}
<x> = 1 to 6 (element)
<NRf> = 0.0001 to 9999.9999
- 实例：** :INPUT:SCALING:VT:ELEMENT1 0.1 ►►
:INPUT:SCALING:VT:ELEMENT1? ►► :INPUT:SCALING:VT:ELEMENT1 0.1000

[[:INPut]:SCALing:{VT|CT|SFACtor|EXT}:{SIGMA|SIGMB|SIGMC}

- 功能：** 统一设置属于指定接线组（ ΣA 、 ΣB 或 ΣC ）的所有通道的电压、电流、功率或电流传感器的比例系数
- 语法：** [[:INPut]:SCALing:{VT|CT|SFACtor|EXT}:{SIGMA|SIGMB|SIGMC} {<NRf>}
<NRf> = 0.0001 to 9999.9999
- 实例：** :INPUT:SCALING:VT:SIGMA 1 ►►
- 说明：** 如果接线组设置([[:INPut]:WIRing)导致相应的连接单元(ΣA 、 ΣB 或 ΣC)不存在，则 SIGMA、SIGMB 或 SIGMC 无效

[[:INPut]:FILTer?

- 功能：** 查询所有输入滤波设置
- 语法：** [[:INPut]:FILTer?
- 实例：** :INPut:FILTer? ►►

[[:INPut]:FILTer:LINE?

功能： 查询所有通道的线路滤波器

语法： [[:INPut]:FILTer:LINE?

实例： :INPut:FILTer:LINE? ►► :INPUT:FILTER:BANK1 2.00E+00,BANK2 5.50E+00;:FILTER:LING:ELEMENT1 OFF,
ELEMENT2 OFF,ELEMENT3 OFF,ELEMENT4 OFF,ELEMENT5 OFF,ELEMENT6 OFF;:FILTER:FIR:ELEMENT1
BANK1,ELEMENT2 BANK1,ELEMENT3 BANK1,ELEMENT4 BANK1,ELEMENT5 BANK1,ELEMENT6 BANK1

[[:INPut]:FILTer[:LINE][:ALL]

功能： 统一设置所有通道的线路滤波器

语法： [[:INPut]:FILTer[:LINE][:ALL] {OFF|1KHz|FIR}

OFF = 线路滤波器

1KHz = 硬件滤波器

FIR = 软件滤波器

实例： :INPut:FILTer:LINE:ALL FIR ►►

[[:INPut]:FILTer[:LINE]:ELEMeNt<x>

功能： 设置或查询指定通道的线路滤波器

语法： [[:INPut]:FILTer[:LINE]:ELEMeNt<x> {OFF|1KHz|FIR}

[[:INPut]:FILTer[:LINE]:ELEMeNt<x>?

实例： :INPut:FILTer:LINE:ELEMeNt1 1KHz ►►

:INPut:FILTer:LINE:ELEMeNt1 ? ►► :INPUT:FILTER:LING:ELEMENT1 1.0E+03

[[:INPut]:FILTer[:LINE]:{SIGMA|SIGMB|SIGMC}

功能： 统一设置属于指定接线组（ ΣA 、 ΣB 或 ΣC ）的所有通道的线路滤波器

语法： [[:INPut]:FILTer[:LINE]:{SIGMA|SIGMB|SIGMC} {OFF|1KHz|FIR}

实例： :INPut:FILTer:LINE:SIGMA OFF ►►

说明： 如果接线组设置([[:INPut]:WIRing)导致相应的连接单元(ΣA 、 ΣB 或 ΣC)不存在，则 SIGMA、SIGMB 或 SIGMC 无效

[[:INPut]:FILTer[:LINE]:FIR?

功能： 查询所有通道的线路滤波器 FIR 类型

语法： [[:INPut]:FILTer[:LINE]:FIR?

实例： :INPut:FILTer:LINE:FIR? ►► :INPUT:FILTER:FIR:ELEMENT1 BANK1,ELEMENT2 BANK1,ELEMENT3 BANK1,
ELEMENT4 BANK1,ELEMENT5 BANK1,ELEMENT6 BANK1

说明： BANK<x>代表预设值，有 BANK1 和 BANK2 两种选择

[[:INPut]:FILTer[:LINE]:FIR[:ALL]

- 功能：** 统一设置所有通道的线路滤波器 FIR 类型
- 语法：** [[:INPut]:FILTer[:LINE]:FIR[:ALL] {BANK1|BANK2}
- 实例：** :INPut:FILTer:LINE:FIR:ALL BANK1 ►►

[[:INPut]:FILTer[:LINE]:FIR:ELEMeNt<x>

- 功能：** 设置或查询指定通道的线路滤波器 FIR 类型
- 语法：** [[:INPut]:FILTer[:LINE]:FIR:ELEMeNt<x> {BANK1|BANK2}
- 实例：** :INPut:FILTer:LINE:FIR:ELEMeNt1 bank2 ►►
:INPut:FILTer:LINE:FIR:ELEMeNt1? ►► :INPut:FILTer:LINE:FIR:ELEMeNt1 BANK2

[[:INPut]:FILTer[:LINE]:FIR:{SIGMA|SIGMB|SIGMC}

- 功能：** 统一设置属于指定接线组（ ΣA 、 ΣB 或 ΣC ）的所有通道的线路滤波器 FIR 类型
- 语法：** [[:INPut]:FILTer[:LINE]:FIR:{SIGMA|SIGMB|SIGMC} {BANK1|BANK2}
- 实例：** :INPut:FILTer:LINE:FIR:SIGMA BANK1 ►►
- 说明：** 如果接线组设置([[:INPut]:WIRing])导致相应的连接单元(ΣA 、 ΣB 或 ΣC)不存在，则 SIGMA、SIGMB 或 SIGMC 无效

[[:INPut]:FILTer[:LINE]:BANk<x>

- 功能：** 设置或查询线路滤波器截止频率
- 语法：** [[:INPut]:FILTer[:LINE]:BANk<x> {<NRf>}
[[:INPut]:FILTer[:LINE]:BANk<x>?
<x> = 1 or 2
<NRf> = 0.5 to 100.0(KHz)
- 实例：** :INPut:FILTer:LINE:BANk1 1.5 ►►
:INPut:FILTer:LINE:BANk1? ►► :INPut:FILTer:LINE:BANk1 1.50E+00

[[:INPut]:FILTer:FREQuency?

- 功能：** 查询所有通道的频率滤波器
- 语法：** [[:INPut]:FILTer:FREQuency?
- 实例：** :INPut:FILTer:FREQuency? ►► :INPut:FILTer:FREQuency:ELEMeNt1 OFF,ELEMeNt2 OFF,ELEMeNt3 OFF,
ELEMeNt4 OFF,ELEMeNt5 OFF,ELEMeNt6 OFF

[[:INPut]:FILTer:FREQuency[:ALL]

- 功能：** 统一设置所有通道的频率滤波器
- 语法：** [[:INPut]:FILTer:FREQuency[:ALL] {OFF|<NRf>}
<NRf> = 0.001 to 500.00(KHz)
- 实例：** :INPut:FILTer:FREQuency:ALL 10.0 ►►

[[:INPut]:FILTer:FREQuency:ELEMeNt<x>

- 功能：** 设置或查询指定通道的频率滤波器
- 语法：** [[:INPut]:FILTer:FREQuency:ELEMeNt<x> {OFF|<NRf>}
[[:INPut]:FILTer:FREQuency:ELEMeNt<x>?<NRf> = 0.001 to 500.00(KHz)
- 实例：** :INPut:FILTer:FREQuency:ELEMeNt1 10.0 ▶▶
:INPut:FILTer:FREQuency:ELEMeNt1? ▶▶ :INPut:FILTer:FREQuency:ELEMeNt1 1.000E+01

[[:INPut]:SYNChronize?

- 功能：** 查询所有通道的同步源
- 语法：** [[:INPut]:SYNChronize?
- 实例：** :INPut:SYNChronize? ▶▶ :INPut:SYNChronize:ELEMENT1 U1,ELEMENT2 U2,ELEMENT3 U3,ELEMENT4 U4,
ELEMENT5 U5,ELEMENT6 U6

[[:INPut]:SYNChronize[:ALL]

- 功能：** 统一设置所有通道的同步源
- 语法：** [[:INPut]:SYNChronize[:ALL] {U<x>|I<x>|NONE}
<x> = 1 to 6 (element)
NONE = 无同步源
- 实例：** :INPut:SYNChronize:ALL U1 ▶▶

[[:INPut]:SYNChronize:ELEMeNt<x>

- 功能：** 设置或查询指定通道的同步源
- 语法：** [[:INPut]:SYNChronize:ELEMeNt<x> {U<x>|I<x>|NONE}
[[:INPut]:SYNChronize:ELEMeNt<x>?<x> = 1 to 6 (element)
NONE = 无同步源
- 实例：** :INPut:SYNChronize:ELEMeNt1 I1 ▶▶
:INPut:SYNChronize:ELEMeNt1? ▶▶ :INPut:SYNChronize:ELEMeNt1 I1
- 说明：** 设置无同步源时，以数据更新速率为准

[[:INPut]:SYNChronize:{SIGMA|SIGMB|SIGMC}

- 功能：** 统一设置属于指定接线组（ΣA、ΣB 或 ΣC）的所有通道的同步源
- 语法：** [[:INPut]:SYNChronize:{SIGMA|SIGMB|SIGMC} {U<x>|I<x>|NONE}
<x> = 1 to 6 (element)
NONE = 无同步源
- 实例：** :INPut:SYNChronize:SIGMA U1 ▶▶
- 说明：** 如果接线组设置([[:INPut]:WIRing)导致相应的连接单元(ΣA、ΣB 或 ΣC)不存在，则 SIGMA、SIGMB 或 SIGMC 无效

[[:INPut]:WIRing

功能： 设置或查询接线组

语法： `[[:INPut]:WIRing {(P1W2|P1W3|P3W3|P3W4|V3A3)`
`[(P1W2|P1W3|P3W3|P3W4|V3A3)]`
`[(P1W2|P1W3|P3W3|P3W4|V3A3)]`
`[(P1W2|P1W3|P3W3|P3W4|V3A3)]`
`[(P1W2|P1W3|P3W3)]`
`[(P1W2)]`

P1W2 = 单相两线制系统 [1P2W]

P1W3 = 单相三线制系统 [1P3W]

P3W3 = 三相三线制系统 [3P3W]

P3W4 = 三相四线制系统 [3P3W]

V3A3 = 采用三电压、三电流方法的三相三线制系统[3P3W(3V3A)]

`[[:INPut]:WIRing?`

实例： `:INPut:WIRing P3W3,P3W3,P1W2,P1W2 ▶▶`

`:INPut:WIRing ▶▶ :INPUT:WIRING P3W3,P3W3,P1W2,P1W2`

说明： 1、设置指令同时兼容 (**P1W2**、**1P2W**)两种格式，查询时默认为 **P1W2** 格式

2、当设置指令省略后续通道时，后续通道默认设置为 **P1W2**

例如 `:INPut:WIRing P3W3` 只设置 1 和 2 通道为三相三线制，则后续 3、4、5、6 通道默认设置为 **P1W2** 单相两线制

2.5、数值相关命令 (NUMeric Group)

:NUMeric?

功能： 查询所有数值数据输出设置

语法： :NUMeric?

实例： :NUMeric? ►► :NUMERIC:FORMAT ASCII;NORMAL:NUMBER 1000;ITEM1 URMS,1;ITEM2 URMS,1;...(省略)...

:NUMeric:FORMat

功能： 设置或查询数值数据格式

语法： :NUMeric:FORMat {ASCIi|FLOat}

:NUMeric:FORMat?

实例： :NUMeric:FORMat ASCII ►►

:NUMeric:FORMat? ►► :NUMERIC:FORMAT ASCII

说明： **ASCIi**: 除了积分用时—TIME 以<NR1>格式输出外，物理量值均以<NR3>输出。数据项之间以逗号隔开。

FLOat: 在各个数值数据块前可加一个头，例如"#240"或"#3208"。头后面紧跟 IEEE 标准单精度浮点（4 个字节）格式的物理量。数据内的字节顺序是 MSB 在 LSB 前。

:NUMeric:HOLD

功能： 设置或查询数值数据保持状态

语法： :NUMeric:HOLD {<Boolean>}

:NUMeric:HOLD?

实例： :NUMeric:HOLD ON ►►

:NUMeric:HOLD? ►► :NUMERIC:HOLD 1

说明： 如果:**NUMeric:HOLD** 设置为 **ON**，则该时间点的所有数值数据都可以在内部保存，即使屏幕上的数值数据被更新，数值数据也会被保留。

如果:**NUMeric:HOLD** 在之前被设置为 **ON** 后被设置为 **ON**，则清除数值数据，并保留最近的数值数据。当连续检索数值数据时，可以使用此方法 来避免重复将:**NUMeric:HOLD** 设置为 **OFF**。

:NUMeric:NORMal?

功能： 查询所有数值数据输出设置

语法： :NUMeric:NORMal?

实例： :NUMeric:NORMal? ►► :NUMERIC:NORMAL:NUMBER 1000;ITEM1 URMS,1;ITEM2 URMS,1;ITEM3 URMS,1;
...(省略)...;ITEM998 URMS,1;ITEM999 URMS,1;ITEM1000 URMS,1

:NUMeric[:NORMal]:CLEar

功能：清除数值数据输出项

语法：:NUMeric[:NORMal]:CLEar {ALL|<NRf>[,<NRf>]}

ALL = Clear all items

First <NRf> = 1 to 1000 (the number of the first item to clear)

Second <NRf> = 1 to 1000 (the number of the last item to clear)

实例：:NUMeric:NORMal:CLEar ALL ▶▶

说明：如果省略第二个<NRf>，则清除第一个<NRf>指定的输出项和随后的所有输出项(不超过 1000)。

:NUMeric[:NORMal]:DELeTe

功能：删除数值数据输出项

语法：:NUMeric[:NORMal]:DELeTe {<NRf>[,<NRf>]}

First <NRf> = 1 to 1000 (要删除的第一个 ITEMx 的编号)

Second <NRf> = 1 to 1000 (最后要删除的 ITEMx 的编号)

实例：:NUMeric:NORMal:DELeTe 1 ▶▶

(删除 ITEM1 并向前移动 ITEM2 和后续项目)

:NUMeric:NORMal:DELeTe 1,3 ▶▶

(将 ITEM1 删除到 ITEM3，并将 ITEM4 和后续项目向前移动)

说明：被删除的输出项的位置由它们后面的项填充，并将末尾的空部分设置为 NONE。

如果省略第二个<NRf>，则只删除第一个<NRf>指定的输出项。

:NUMeric[:NORMal]:ITEM<x>

功能：设置或查询指定的数值数据输出项(函数、元素和谐波顺序)

语法：:NUMeric[:NORMal]:ITEM<x> {NONE|<Function>[,<Element>][,<Order>]}

:NUMeric[:NORMal]:ITEM<x>?

<x> = 1 to 1000 (item number) NONE = No output item

<Function> = {URMS|IRMS|P|S|Q|...}

<Element> = {<NRf>|SIGMA|SIGMB|SIGMC}(<NRf> = 1 to 6)

<Order> = {TOTal|DC|<NRf>}(<NRf> = 1 to 500)

实例：:NUMeric:NORMal:ITEM1 URMS,1 ▶▶

:NUMeric:NORMal:ITEM1? ▶▶ :NUMeric:NORMal:ITEM1 URMS,1

:NUMeric:NORMal:ITEM1 UK,1,1 ▶▶

:NUMeric:NORMal:ITEM1? ▶▶ :NUMeric:NORMal:ITEM1 UK,1,1

说明：1、如果省略<Element>，则该元素被设置为 1。如果省略<Order>，则顺序设置为 TOTal。

<Element>和<Order>在对不需要它们的函数的响应中被省略。

2、有关<Function>可用选项的信息，请参考[附录：<Function>选项列表](#)

:NUMeric[:NORMal]:NUMBER

功能： 设置或查询传输的数值数据项的个数

语法： :NUMeric[:NORMal]:NUMBER {<NRf>|ALL}

:NUMeric[:NORMal]:NUMBER?

<NRf> = 1 to 1000 (ALL)

实例： :NUMeric:NORMal:NUMBER 100 ►►

:NUMeric:NORMal:NUMBER? ►► :NUMERIC:NORMAL:NUMBER 100

:NUMeric[:NORMal]:PRESet

功能： 预设数值数据输出项模式

语法： :NUMeric[:NORMal]:PRESet {<NRf>}

<NRf> = 1 to 4 (默认值 1)

实例： :NUMeric:NORMal:PRESet 1 ►►

说明： 有关预设输出项的信息请参考[附录：数值数据项的预设模式表](#)

:NUMeric[:NORMal]:VALue?

功能： 查询数值数据

语法： :NUMeric[:NORMal]:VALue? {<NRf>}

<NRf> = 1 to 1000 (item number)

实例： 当指定<NRf>时: :NUMeric:NORMal:VALue? 1 ►► 103.79E+00

当省略<NRf>时: :NUMeric:NORMal:VALue? ►► 103.79E+ 00,1.0143E+00,105.27E+00,...(省略)....,1.428E+00

当:NUMeric:FORMat 设置为 FLOat 时: :NUMeric:NORMal:VALue? ►► #N(n 位字节数)(数据字节序列)

说明： 如果指定了<NRf>, 则只输出指定项的数字数据。

如果省略<NRf>, 则按顺序输出从 1 到:NUMeric[:NORMal]: NUMBER 命令指定的数字的数值数据项。

有关输出的单个数值数据项的格式, 请参考:NUMeric:FORMat

:NUMeric:LIST?

功能： 查询所有谐波测量数值列表数据输出设置

语法： :NUMeric:LIST?

实例： :NUMeric:LIST? ►► :NUMERIC:LIST:NUMBER 64;ORDER 500;SELECT ALL;ITEM1 U,1;ITEM2 I,1;...(省略)...

:NUMeric:LIST:CLEar

功能： 清除谐波测量数值列表数据输出项目 (项目设置为 NONE)

语法： :NUMeric:LIST:CLEar {ALL|<NRf> [, <NRf>]}

ALL = 全部清除 ITEMS

<NRf> = 1 to 64 (清除的项目的编号的起始位置)

实例： :NUMeric:LIST:CLEar ALL ►►

:NUMeric:LIST:DELeTe

- 功能：**删除谐波测量数值列表数据输出项
- 语法：**
:NUMeric:LIST:DELeTe {<NRf>[,<NRf>]}
<NRf> = 1 to 64 (删除的项目的编号的起始位置)
- 实例：**
:NUMeric:LIST:DELeTe 1 ►►
- 说明：**实例中删除 ITEM1 并将 ITEM2 和后续项目向前移动

:NUMeric:LIST:ITEM<x>

- 功能：**设置或查询指定谐波测量数值列表数据项的输出项（函数和元素）
- 语法：**
:NUMeric:LIST:ITEM<x> {NONE|<Function>,<Element>}
:NUMeric:LIST:ITEM<x>?
<Function> = {U|I}
<Element> = 1 to 6
- 实例：**
:NUMeric:LIST:ITEM1 U,1 ►►
:NUMeric:LIST:ITEM1? ►► :NUMERIC:LIST:ITEM1 U,1

:NUMeric:LIST:NUMber

- 功能：**设置或查询传输的数值数据项的个数
- 语法：**
:NUMeric:LIST:NUMber {<NRf>|ALL}
:NUMeric:LIST:NUMber?
<NRf> = 1 to 64 (ALL)
- 实例：**
:NUMeric:LIST:NUMber 10 ►►
:NUMeric:LIST:NUMber? ►► :NUMERIC:LIST:NUMBER 10

:NUMeric:LIST:ORDer

- 功能：**设置或查询谐波测量数值列表数据的最大输出谐波阶数
- 语法：**
:NUMeric:LIST:ORDer {<NRf>|ALL}
<NRf> = 1 to 500 (ALL)
- 实例：**
:NUMeric:LIST:ORDer 100 ►►
:NUMeric:LIST:ORDer? ►► :NUMERIC:LIST:ORDER 100

:NUMeric:LIST:PRESet

- 功能：**预设谐波测量数值列表数据输出项目模式
- 语法：**
:NUMeric:LIST:PRESet <NRf>
<NRf> = 1
- 实例：**
:NUMeric:LIST:PRESet 1 ►►
- 说明：**有关预设输出项的信息，请参考[附录：谐波数值数据项的预设模式表](#)

:NUMeric:LIST:SElect

功能： 设置或查询谐波测量数值列表数据的输出分量

语法： :NUMeric:LIST:SElect {EVEN|ODD|ALL}

:NUMeric:LIST:SElect?

EVEN = 输出 TOTal、DC 和偶次谐波分量

ODD = 输出 TOTal、DC 和奇次谐波分量

ALL = 输出所有谐波分量

实例： :NUMeric:LIST:SElect EVEN ▶▶

:NUMeric:LIST:SElect? ▶▶ :NUMERIC:LIST:SELECT EVEN

:NUMeric:LIST:VALue?

功能： 查询谐波测量数值列表数据

语法： :NUMeric:LIST:VALue? {<NRf>}

<NRf> = 1 to 64 (item number)

实例： 当省略<NRf>时:

:NUMeric:LIST:VALue? ▶▶ 9.985E+00,2.125E-03,1.431E-03,3.039E-04,7.205E-04,3.305E-04,7.065E-04,

5.921E-04,6.138E-04,5.638E-04,3.260E-04,8.984E-04,1.012E-03,4.083E-04,7.593E-04,3.124E-04,...(省略)...

(502×5 = 最多 2510 个数据项)

当:NUMeric:FORMat 设置为 FLOat 时:

:NUMeric:LIST:VALue? ▶▶ #N (n 位字节数)

说明： 单个数字列表数据项最多由 502 个数字数据项组成，顺序如下:TOTal, DC, 一次谐波分量, ...(省略)..., n 次谐波分量

2.6、波形相关命令 (WAVeform Group)

:WAVeform?

功能： 查询所有波形数据输出设置

语法： :WAVeform?

实例： :WAVeform? ►► :WAVEFORM:TRACE U1;FORMAT ASCII;START 0;END 1601;HOLD 0

:WAVeform:MODE

功能： 设置和查询波形数据模式

语法： :WAVeform:MODE {MODE1|MODE2}

实例： :WAVeform:MODE MODE1 ►►

:WAVeform:MODE? ►► :WAVEFORM:MODE MODE1

说明： 1、 **MODE1** 模式采用指定区间内寻找最大值和最小值作为波形数据，最多 2002 个波形数据

2、 **MODE2** 模式采用指定区间内进行抽点作为波形数据，最多 2049 个波形数据

:WAVeform:TRACe

功能： 设置和查询波形数据绑定

语法： :WAVeform:TRACe {U<x>|I<x>}

<x> = 1 to 6

实例： :WAVeform:TRACe U1 ►►

:WAVeform:TRACe? ►► :WAVEFORM:TRACE U1

:WAVeform:FORMat

功能： 设置和查询显示数据的格式

语法： :WAVeform:FORMat {ASCIi|FLOat}

:WAVeform:FORMat?

实例： :WAVeform:FORMat ASCIi ►►

:WAVeform:FORMat? ►► :WAVEFORM:FORMAT ASCII

:WAVeform:STARt

功能： 设置和查询数据的输出起点

语法： :WAVeform:STARt <NRf>

:WAVeform:STARt?

MODE1: <NRf> = 0 to 2001

MODE2: <NRf> = 0 to 2048

实例： :WAVeform:STARt 1 ►►

:WAVeform:STARt? ►► :WAVEFORM:START 1

:WAVeform:END

功能: 设置和查询数据的输出终点

语法: :WAVeform:END <NRf>

:WAVeform:END?

<NRf> 参考:**WAVeform:STARt**

实例: :WAVeform:END 1000 ►►

:WAVeform:END? ►► :WAVEFORM:END 1000

说明: 设置输出终点时需要大于输出起点，如果设置的输出终点小于输出起点，则:WAVeform:SEND?查询时不回复

:WAVeform:TIME

功能: 设置和查询 2048 点波形的通信区间

语法: :WAVeform:TIME <TIM>

:WAVeform:TIME?

<TIM> = 10MS to 200MS (每 10MS 为一个间隔)

实例: :WAVeform:TIME 40MS ►►

:WAVeform:TIME? ►► :WAVEFORM:TIME 40MS

说明: 1、只有当:**WAVeform:MODE** 设置为 **MODE2** 时此命令有效

2、指定<TIM>波形区间内的波形数据

:WAVeform:HOLD

功能: 设置或查询所有波形的波形显示数据保持功能

语法: :WAVeform:HOLD {<Boolean>}

:WAVeform:HOLD?

实例: :WAVeform:HOLD ON ►►

:WAVeform:HOLD? ►► :WAVEFORM:HOLD 1

说明: 1、如果波形显示数据保持功能被打开，则该时间的所有波形数据都可以在内部保存

例如: **:WAVeform:HOLD** 设置为 **ON**，即使在屏幕上显示的波形更新时，波形数据也会保持

以获取 U1 和 I1 在同一时间点的波形显示数据为例:

:WAVeform:HOLD ON(打开波形显示数据保持功能)

:WAVeform:TRACe U1(将:**WAVeform:TRACe** 类型绑定 **U1**，并查询 **U1** 的波形数据)

:WAVeform:SEND?(查询 U1 的波形显示数据)

:WAVeform:TRACe I1(将:**WAVeform:TRACe** 类型绑定 **I1**，并查询 **I1** 的波形数据)

:WAVeform:SEND?(查询 I1 的波形显示数据)

:WAVeform:HOLD OFF(关闭波形显示数据保持功能)

2、如果打开波形显示数据保持功能时重新打开保持功能，波形显示数据被清除，并且内部保持最新的波形数据

当连续检索波形显示数据时，这种方法可以避免重复设置:WAVeform:HOLD OFF

:WAVeform:LENGth?

功能： 查询指定波形的总点数

语法： :WAVeform:LENGth?

实例： **MODE1:** :DISPlay:NUMeric:NORMal? >> :WAVEFORM:LENGTH 2002

MODE2: :DISPlay:NUMeric:NORMal? >> :WAVEFORM:LENGTH 2049

:WAVeform:SEND?

功能： 查询指定的波形显示数据

语法： :WAVeform:SEND?

实例： 当:**WAVeform:FORMat** 设置为 ASCII

:WAVeform:SEND? >> <NR3>,<NR3>,...(省略)...,<NR3>

当:**WAVeform:FORMat** 设置为 FLOAT

:WAVeform:SEND? >> #4? (4 位字节数)(数据字节序列)

说明： 在 **MODE1** 模式下数据顺序为：采样点 1 的最小值,采样点 1 的最大值,采样点 2 的最小值,采样点 2 的最大值,...(省略)...

:WAVeform:BYTeorder

功能： 设置和查询波形数据(FLOAT 格式)的输出字节顺序

语法： :WAVeform:BYTeorder {LSBFirst|MSBFirst}

:WAVeform:BYTeorder?

LSBFirst: 低字节在前

MSBFirst: 高字节在前

实例： :WAVeform:BYTeorder MSBFirst >>

:WAVeform:BYTeorder? >> :WAVEFORM:BYTEORDER MSBFIRST

:WAVeform:SRATe?

功能： 查询当前波形采样率

语法： :WAVeform:SRATe?

实例： :WAVeform:SRATe? >> 800.0E+00

2.7、积分相关命令 (INTEGrate Group)

:INTEGrate?

功能： 查询所有积分设置

语法： :INTEGrate?

实例： :INTEGrate? ►►

:INTEGrate:ACAL

功能： 设置或查询积分自动校准的开/关状态

语法： :INTEGrate:ACAL {<Boolean>}

:INTEGrate:ACAL?

实例： :INTEGrate:ACAL ON ►►

:INTEGrate:ACAL? ►► :INTEGRATE:ACAL 1

:INTEGrate:INDependent

功能： 设置或查询独立积分的开/关状态

语法： :INTEGrate:INDependent {<Boolean>}

:INTEGrate:INDependent?

实例： :INTEGrate:INDependent ON ►►

:INTEGrate:INDependent? ►► :INTEGRATE:INDEPENDENT 1

:INTEGrate:MODE

功能： 设置或查询积分模式

语法： :INTEGrate:MODE {NORMal|CONTInuous| RNORMal|RCONTInuous}

:INTEGrate:MODE?

NORMal = 正常积分模式

CONTInuous = 连续积分模式

RNORMal = 预约积分模式

RCONTInuous = 预约连续积分模式

实例： :INTEGrate:MODE NORMal ►►

:INTEGrate:MODE? ►► :INTEGRATE:MODE NORMAL

:INTEGrate:QMODe?

功能： 查询所有通道的电流积分的电流模式

语法： :INTEGrate:QMODe?

实例： :INTEGrate:QMODe? ►► :INTEGRATE:QMODE:ELEMENT1 RMS;ELEMENT2 RMS;ELEMENT3 RMS;

ELEMENT4 RMS;ELEMENT5 RMS;ELEMENT6 RMS

:INTEGrate:QMODE[:ALL]

- 功能：** 统一设置所有通道的电流积分的电流模式
- 语法：** :INTEGrate:QMODE[:ALL] {RMS|MEAN|DC|RMEAN|AC}
- 实例：** :INTEGrate:QMODE:ALL DC ►►

:INTEGrate:QMODE:ELEMent<x>

- 功能：** 设置或查询指定通道的电流积分的电流模式
- 语法：** :INTEGrate:QMODE:ELEMent<x> {RMS|MEAN|DC|RMEAN|AC}
- :INTEGrate:QMODE:ELEMent<x>?
- <x> = 1 to 6 (element)
- 实例：** :INTEGrate:QMODE:ELEMent1 RMS ►►
- :INTEGrate:QMODE:ELEMent1? ►► :INTEGRATE:QMODE:ELEMENT1 RMS

:INTEGrate:RESet

- 功能：** 重置积分
- 语法：** :INTEGrate:RESet {[<NRf>][,<NRf>] [<NRf>][,<NRf>][,<NRf>][,<NRf>]}
- <NRf> = 1 to 6 (element)
- 实例：** :INTEGrate:RESet ►► (重置所有通道)
- :INTEGrate:RESet 1,2,3 ►► (重置指定的通道)
- 说明：** 1、当独立积分(:INTEGrate:INDependent)设置 ON(1)时，您可以使用参数指定要对哪些通道执行操作
如果省略参数，则将对所有通道执行操作
- 2、当独立积分(:INTEGrate: independent)设置为 OFF(0)时，不能使用该参数指定通道

:INTEGrate:RTALl:{START|END}

- 功能：** 设置预约积分模式共同设置所有通道的积分开始或结束时间
- 语法：** :INTEGrate:RTALl:{START|END} {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}
- {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>} = 2001,1,1,0,0,0 to 2099,12,31,23,59,59
- First <NRf> = 2001 to 2099 (year)
- Second <NRf> = 1 to 12 (month)
- Third <NRf> = 1 to 31 (day)
- Fourth <NRf> = 0 to 23 (hour)
- Fifth <NRf> = 0 to 59 (minute)
- Sixth <NRf> = 0 to 59 (second)
- 实例：** :INTEGrate:RTALl:START 2024,1,1,0,0,0 ►►

:INTEGrate:RTIME<x>?

功能: 查询预约积分模式的积分开始和结束时间

语法: :INTEGrate:RTIME<x>?

<x> = 1 to 6 (element)

实例: :INTEGrate:RTIME1? ►► :INTEGRATE:RTIME1:START 2024,1,1,0,0,0;END 2099,12,31,23,59,59

:INTEGrate:RTIME<x>:{START|END}

功能: 设置或查询预约积分模式的积分开始或结束时间

语法: :INTEGrate:RTIME<x>:{START|END} {<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>}

:INTEGrate:RTIME<x>:{START|END}?

<x> = 1 to 6 (element)

{<NRf>,<NRf>,<NRf>,<NRf>,<NRf>,<NRf>} 参考:INTEGrate:RTALI:{START|END}

实例: :INTEGrate:RTIME1:START 2024,1,1,0,0,0 ►►

:INTEGrate:RTIME1:START? ►► :INTEGRATE:RTIME1:START 2024,1,1,0,0,0

说明: 1、当积分模式:INTEGrate:MODE 设置为预约积分模式 (RNORmal 或 RCONTinuous) 时, 此命令有效

2、当独立积分关闭时, <x>需要使用 1

:INTEGrate:START

功能: 开始积分

语法: :INTEGrate:START {[<NRf>],[<NRf>] [<NRf>],[<NRf>],[<NRf>],[<NRf>]}

<NRf> = 1 to 6 (element)

实例: :INTEGrate:START ►► (所有通道开始积分)

:INTEGrate:START 1,2,3 ►► (所有通道开始积分)

说明: 1、当独立积分 (:INTEGrate:INDePendent) 设置为 ON (1) 时, 可以使用参数指定要对哪些通道执行操作
如果省略参数, 将对所有元素执行操作

2、当独立积分 (:INTEGrate:INDePendent) 设置为 OFF (0) 时, 不能使用参数指定通道

3、当积分打开时, 所有关于积分设置的命令全部无效, 除了:INTEGrate:STOP 命令

:INTEGrate:STATe?

功能: 查询积分状态

语法: :INTEGrate:STATe?

实例: 当独立积分关闭时:

:INTEGrate:STATe? ►► RESET

当独立积分打开时:

:INTEGrate:STATe? ►► RESET,RESET,RESET,RESET,RESET,RESET

说明: 回复类型: RESeT(集成复位)、READy(预约积分)、STARt(正在积分)、STOP(积分停止)、TIMeup(积分超时)

:INTEGrate:STOP

功能: 停止积分

语法: :INTEGrate:STOP [{<NRf>},<NRf>] [,<NRf>],<NRf>},{<NRf>},<NRf>}]

<NRf> = 1 to 6 (element)

实例: :INTEGRATE:STOP (停止对所有通道的积分)

:INTEGRATE:STOP 1,2,3 (停止对指定通道的积分)

说明: 1、当独立积分（:INTEGrate:INDEpendent）设置为 ON（1）时，可以使用参数指定要对哪些元素执行操作

如果省略参数，将对所有元素执行操作

2、当独立积分（:INTEGrate:INDEpendent）设置为 OFF（0）时，不能使用参数指定元素

3、只有当积分打开时，此命令才有效

:INTEGrate:TMer<x>

功能: 设置或查询积分计时器值

语法: :INTEGrate:TMer<x> {<NRf>,<NRf>,<NRf>}

:INTEGrate:TMer<x>?

<x> = 1 to 6 (element)

{<NRf>,<NRf>,<NRf>}=0,0,0 to 99999,59,59

First <NRf> = 0 to 99999 (hours)

Second <NRf> = 0 to 59 (minutes)

Third <NRf> = 0 to 59 (seconds)

实例: :INTEGrate:TMer1 1,0,0 ▶▶

:INTEGrate:TMer1? ▶▶ :INTEGrate:TMer1 1,0,0

说明: 1、当独立积分（:INTEGrate:INDEpendent）设置为 ON（1）时，可以使用<x>参数指定要对哪些通道执行操作

如果省略参数，将对所有元素执行操作

2、当独立积分（:INTEGrate:INDEpendent）设置为 OFF（0）时，可以使用 x = 1 执行操作

:INTEGrate:TMALI

功能: 统一设置所有通道的积分计时器

语法: :INTEGrate:TMALI {<NRf>,<NRf>,<NRf>}

{<NRf>,<NRf>,<NRf>} = 0, 0, 0 to 99999,59,59

First <NRf> = 0 to 10000 (hours)

Second <NRf> = 0 to 59 (minutes)

Third <NRf> = 0 to 59 (seconds)

实例: :INTEGrate:TMALI 2,0,0 ▶▶

说明: 独立积分是否打开都有效

:INTEGrate:WPTYPE?

功能: 查询所有通道的每个极性 (WP+/WP-) 的瓦时积分方法

语法: :INTEGrate:WPTYPE?

实例: :INTEGrate:WPTYPE? ►► :INTEGRATE:WPTYPE:ELEMENT1 CHARGE;ELEMENT2 CHARGE;
ELEMENT3 CHARGE;ELEMENT4 CHARGE;ELEMENT5 CHARGE;ELEMENT6 CHARGE

:INTEGrate:WPTYPE[:ALL]

功能: 统一设置所有通道的每个极性 (WP+/WP-) 的瓦时积分方法

语法: :INTEGrate:WPTYPE[:ALL] {CHARGE|SOLD}

实例: :INTEGrate:WPTYPE:ALL CHARGE ►►

:INTEGrate:WPTYPE:ELEMENT<x>

功能: 设置或查询指定通道的每个极性 (WP+/WP-) 的瓦时积分方法

语法: :INTEGrate:WPTYPE:ELEMENT<x> {CHARGE|SOLD}

:INTEGrate:WPTYPE:ELEMENT<x>?

<x> = 1 to 6 (element)

实例: :INTEGrate:WPTYPE:ELEMENT1 CHARGE ►►

:INTEGrate:WPTYPE:ELEMENT1? ►► :INTEGRATE:WPTYPE:ELEMENT1 CHARGE

2.8、谐波相关命令 (HARMonics Group)

:HARMonics<x>?

功能： 查询所有谐波测量设置

语法： :HARMonics<x>?

<x> = 1 or 2 (谐波测量组)

实例： :HARMonics1? ►► :HARMONICS1:ORDER 0,500;PLLSOURCE U1;POINT 8192;THD FUNDamental;
:HARMONICS1:CONFIGURE:ELEMENT1 1,ELEMENT2 1,ELEMENT3 1,ELEMENT4 1,ELEMENT5 1,ELEMENT6 1

说明： 本仪器共有两组谐波，本别是 Harm1、Harm2

:HARMonics:CONFigure?

功能： 查询所有通道的谐波测量

语法： :HARMonics:CONFigure?

实例： :HARMonics:CONFigure? ►► :HARMONICS1:CONFIGURE:ELEMENT1 1,ELEMENT2 1,ELEMENT3 1,
ELEMENT4 1,ELEMENT5 1,ELEMENT6 1

:HARMonics:CONFigure[:ALL]

功能： 统一设置所有通道的谐波测量

语法： :HARMonics:CONFigure[:ALL] {<NRf>}

<NRf> = 1 (Harm1), 2 (Harm2)

实例： :HARMonics:CONFigure:ALL 1 ►►

:HARMonics:CONFigure:ELEMEnt<x>

功能： 设置或查询指定通道的谐波测量

语法： :HARMonics:CONFigure:ELEMEnt<x> {<NRf>}

:HARMonics:CONFigure:ELEMEnt<x>?

<x> = 1 to 6 (element)

<NRf> = 1 (Harm1), 2 (Harm2)

实例： :HARMonics:CONFigure:ELEMEnt1 1 ►►
:HARMonics:CONFigure:ELEMEnt1? ►► :HARMONICS1:CONFIGURE:ELEMENT1 1

:HARMonics:CONFigure:{SIGMA|SIGMB|SIGMC}

功能： 统一设置属于指定接线组（ ΣA 、 ΣB 或 ΣC ）所有通道的谐波测量

语法： :HARMonics:CONFigure:{SIGMA|SIGMB|SIGMC} {<NRf>}

<NRf> = 1 (Harm1), 2 (Harm2)

实例： :HARMonics:CONFigure:SIGMA 1 ►►

:HARMonics<x>:ORDer

- 功能:** 设置或查询所分析的最大和最小谐波阶数
- 语法:** :HARMonics<x>:ORDer {<NRf>,<NRf>}
:HARMonics<x>:ORDer?
<x> = 1 or 2 (谐波测量组)
First <NRf> = 0 or 1 (分析的最小谐波阶数)
Second <NRf> = 1 to 500 (分析的最大谐波阶数)
- 实例:** :HARMonics1:ORDer 0,500 ►►
:HARMonics1:ORDer? ►► :HARMONICS1:ORDER 0,500

:HARMonics<x>:PLLSource

- 功能:** 设置或查询 PLL 源
- 语法:** :HARMonics<x>:PLLSource {U<x>|I<x>}
:HARMonics<x>:PLLSource?
HARMonics<x>'s <x> = 1 or 2 (谐波测量组)
U<x> and I<x>'s <x> = 1 to 6 (element)
- 实例:** :HARMonics1:PLLSource U1 ►►
:HARMonics1:PLLSource? ►► :HARMONICS1:PLLSOURCE U1

:HARMonics<x>:THD

- 功能:** 设置或查询用于计算 THD (总谐波失真) 的公式
- 语法:** :HARMonics<x>:THD {TOTal|FUNDamental}
:HARMonics<x>:THD?
<x> = 1 or 2 (谐波测量组)
- 实例:** :HARMonics1:THD TOTal ►►
:HARMonics1:THD? ►► :HARMONICS1:THD TOTAL

:HARMonics<x>:POINT

- 功能:** 设置或查询用于谐波测量的 FFT 点数
- 语法:** :HARMonics<x>:POINT {<NRf>}
:HARMonics<x>:POINT?
<x> = 1 (谐波测量组)
- 实例:** :HARMonics1:POINT 1024 ►►
:HARMonics1:POINT? ►► :HARMONICS1:POINT 1024

2.9、测量相关命令 (MEASure Group)

:MEASure?

功能： 查询所有测量设置

语法： :MEASure?

实例： :MEASure? ►► :MEASURE: AVERAGING: STATE 0; TYPE LINEAR; COUNT 8; :MEASURE: PC: IEC 1976;
P1 0.5000; P2 0.5000; :MEASURE: PHASE 180; SAMPLING 200K; SFORMULA RMS; SQFORMULA TYPE1;
SYNCHRONIZE MASTER

:MEASure:AVERaging?

功能： 查询所有平均设置

语法： :MEASure:AVERaging?

实例： :MEASure:AVERaging? ►► :MEASURE: AVERAGING: STATE 0; TYPE LINEAR; COUNT 8

:MEASure:AVERaging:COUNT

功能： 设置或查询平均系数

语法： :MEASure:AVERaging:COUNT {<NRf>}
:MEASure:AVERaging:COUNT?
<NRf> = 2 to 64 (当 TYPE = EXPonent(衰减常数))
<NRf> = 8 to 64 (当 TYPE = LINear(平均个数))

实例： :MEASure:AVERaging:COUNT 2 ►►
:MEASure:AVERaging:COUNT? ►► :MEASURE: AVERAGING: COUNT 2

:MEASure:AVERaging[:STATe]

功能： 设置或查询平均的开/关状态

语法： :MEASure:AVERaging[:STATe] {<Boolean>}
:MEASure:AVERaging[:STATe]?

实例： :MEASure:AVERaging:STATe ON ►►
:MEASure:AVERaging:STATe? ►► :MEASURE: AVERAGING: STATE 1

:MEASure:AVERaging:TYPE

功能： 设置或查询平均类型

语法： :MEASure:AVERaging:TYPE {EXPonent| LINear}
EXPonent(衰减常数)
LINear(平均个数)

实例： :MEASure:AVERaging:TYPE EXPonent ►►
:MEASure:AVERaging:TYPE? ►► :MEASURE: AVERAGING: TYPE EXPONENT

:MEASure:MHOLd

功能: 设置或查询最大值保持的开/关状态

语法: :MEASure:MHOLd {<Boolean>}

:MEASure:MHOLd?

实例: :MEASure:MHOLd ON ►►

:MEASure:MHOLd? ►► :MEASURE:MHOLD 1

:MEASure:PC?

功能: 查询所有 Pc 修正设置

语法: :MEASure:PC?

实例: :MEASure:PC? ►► :MEASURE:PC:IEC 1976;P1 0.5000;P2 0.5000

:MEASure:PC:IEC

功能: 设置或查询 Pc 修正公式

语法: :MEASure:PC:IEC {<NRf>}

:MEASure:PC:IEC?

<NRf> = 1976, 1993

实例: :MEASure:PC:IEC 1976 ►►

:MEASure:PC:IEC? ►► :MEASURE:PC:IEC 1976

:MEASure:PC:P<x>

功能: 设置或查询 Pc 修正公式参数

语法: :MEASure:PC:P<x> {<NRf>}

:MEASure:PC:P<x>?

<x> = 1, 2 (P1, P2)

<NRf> = 0.0001 to 9.9999

实例: :MEASure:PC:P1 0.5 ►►

:MEASure:PC:P1? ►► :MEASURE:PC:P1 0.5000

:MEASure:PHASe

功能: 设置或查询相位差的显示格式

语法: :MEASure:PHASe {<NRf>}

:MEASure:PHASe?

<NRf> = 180, 360

实例: :MEASure:PHASe 180 ►►

:MEASure:PHASe? ►► :MEASURE:PHASE 180

:MEASure:SAMPling

功能： 设置或查询采样频率

语法： :MEASure:SAMPling {<NRf>}

:MEASure:SAMPling?

<NRf> = 100, 200(KSps)

实例： :MEASure:SAMPling 100 ►►

:MEASure:SAMPling? ►► :MEASURE:SAMPLING 100K

:MEASure:SFORMula

功能： 设置或查询用于计算 S（视在功率）的方程

语法： :MEASure:SFORMula {RMS|MEAN|DC|MRMS|RMEAN}

:MEASure:SFORMula?

实例： :MEASure:SFORMula RMS ►►

:MEASure:SFORMula? ►► :MEASURE:SFORMULA RMS

说明： RMS: $S = U_{rms} \cdot I_{rms}$

MEAN: $S = U_{mean} \cdot I_{mean}$

DC: $S = U_{dc} \cdot I_{dc}$

MRMS: $S = U_{mean} \cdot I_{rms}$

RMEAN: $S = U_{rmean} \cdot I_{rmean}$

:MEASure:SQFormula

功能： 设置或查询用于计算 S（视在功率）和 Q（无功功率）的方程。

语法： :MEASure:SQFormula {TYPE1|TYPE2| TYPE3}

:MEASure:SQFormula?

实例： :MEASure:SQFormula TYPE1 ►►

:MEASure:SQFormula? ►► :MEASURE:SQFORMULA TYPE1

:MEASure:SYNChronize

功能： 设置或查询同步测量模式

语法： :MEASure:SYNChronize {MASTer|SLAVe}

:MEASure:SYNChronize?

实例： :MEASure:SYNChronize MASTer ►►

:MEASure:SYNChronize? ►► :MEASURE:SYNCHRONIZE MASTER

2.10、高速测量相关命令 (HSPeed Group)

:HSPeed?

功能： 查询所有高速测量设置

语法： :HSPeed?

实例： :HSPeed? ►► :HSPEED:STATE OFF;TRIGGER RISE;SYNCHRONIZE U1;TIME:MIN 1.0MS;TIME:MAX 50.0MS

:HSPeed:STATe

功能： 设置或查询高速测量的开/关状态

语法： :HSPeed:STATe {<Boolean>}

:HSPeed:STATe?

实例： :HSPeed:STATe ON ►►

:HSPeed:STATe? ►► :HSPEED:STATE 1

:HSPeed:TRIGger

功能： 设置或查询高速测量的触发模式

语法： :HSPeed:TRIGger {RISE|FALL|BOTH}

:HSPeed:TRIGger?

RISE(上升沿触发)、FALL(下降沿触发)、BOTH(双沿触发, 过零触发(ZERO))

实例： :HSPeed:TRIGger RISE ►►

:HSPeed:TRIGger? ►► :HSPEED:TRIGGER RISE

:HSPeed:SYNChronize

功能： 设置或查询高速同步源

语法： :HSPeed:SYNChronize {U<x>|I<x>|NONE}

:HSPeed:SYNChronize?

<x> = 1 to 6 (element)

实例： :HSPeed:SYNChronize U1 ►►

:HSPeed:SYNChronize? ►► :HSPEED:SYNCHRONIZE U1

:HSPeed[:TIme]:{MIN|MAX}

功能： 设置或查询高速测量单次测量的最小/最大(超时)时间

语法： :HSPeed[:TIme]:{MIN|MAX} {<NRf>}

:HSPeed[:TIme]:{MIN|MAX}?

MIN'S <NRf> = 1.0 to 10.0(ms)、MAX'S <NRf> = 10.0 to 999.9(ms)

实例： :HSPeed:TIme:MIN 1 ►►

:HSPeed:TIme:MIN? ►► :HSPEED:TIME:MIN 1.0MS

说明： 此命令是设置单周期测量计算的时间区间

2.11、电机测试相关命令 (MOTor Group)

:MOTor?

功能： 查询所有电机设置

语法： :MOTor?

实例： :MOTor? ►► :MOTOR:STARTUP:MODE 1;TIMER 1;TRACE U,1;:MOTOR:RLOCKED:MODE 1;TIMER 1;
TRACE U,1;:MOTOR:VOLSTORE:MODE 1;TIMER 1;TRACE U,1;:MOTOR:DATRECORD:MODE 1;TIMER 1;TRACE U,1;
:MOTOR:WAVRECORD:MODE 1;TIMER 1;TRACE U,1;:MOTOR:TRANSIENT:MODE 1;TIMER 1;TRACE U,1

:MOTor:STARtup?

功能： 查询所有电机启动测试设置

语法： :MOTor:STARtup?

实例： :MOTor:STARtup? ►► :MOTOR:STARTUP:MODE 1;TIMER 1;TRACE U,1

:MOTor:STARtup:MODE

功能： 设置或查询启动测试(响应周期)

语法： :MOTor:STARtup:MODE {<NRf>}

:MOTor:STARtup:MODE?

<NRf> = 1、2、3、4、5、6、7、8

实例： :MOTor:STARtup:MODE 1 ►►

:MOTor:STARtup:MODE? ►► :MOTOR:STARTUP:MODE 1

:MOTor:STARtup:TIMer

功能： 设置或查询启动测试持续时间

语法： :MOTor:STARtup:TIMer {<NRf>}

:MOTor:STARtup:TIMer?

{<NRf>} = 1 to 100(秒)

实例： :MOTor:STARtup:TIMer 1 ►►

:MOTor:STARtup:TIMer? ►► :MOTOR:STARTUP:TIMER 1

:MOTor:STARtup:RTIME?

功能： 查询启动测试已运行时间

语法： :MOTor:STARtup:RTIME?

实例： :MOTor:STARtup:RTIME? ►► :MOTOR:STARTUP:RTIME 0.0S

说明： 当启动测试复位时:**MOTor:STARtup:RTIME?**默认回复 0.0S(秒)

:MOTor:STARtup:STATe?

- 功能：** 查询启动测试状态
- 语法：** :MOTor:STARtup:STATe?
- 实例：** :MOTor:STARtup:STATe? ►► RESET
- 说明：** 回复类型：RESet(复位)、STARt(运行)、STOP(停止)、END(结束)

:MOTor:STARtup:STARt

- 功能：** 开始启动测试
- 语法：** :MOTor:STARtup:STARt
- 实例：** :MOTor:STARtup:STARt ►►
- 说明：** 开始启动测试时所有启动测试设置命令均无效

:MOTor:STARtup:STOP

- 功能：** 停止启动测试
- 语法：** :MOTor:STARtup:STOP
- 实例：** :MOTor:STARtup:STOP ►►
- 说明：** 只有:MOTor:STARtup:STATe?为 STARt(运行)时，才可以进行停止测试

:MOTor:STARtup:RESet

- 功能：** 重置启动测试
- 语法：** :MOTor:STARtup:RESet
- 实例：** :MOTor:STARtup:RESet ►►
- 说明：** 只有:MOTor:STARtup:STATe?为 STOP(停止)或 END(结束)时，才可以进行重置测试

:MOTor:STARtup:LENGth?

- 功能：** 查询启动测试数据采集的数量
- 语法：** :MOTor:STARtup:LENGth?
- 实例：** :MOTor:STARtup:LENGth? ►► :MOTOR:STARTUP:LENGTH 5000

:MOTor:STARtup:TRACe

- 功能：** 设置或查询启动测试的数据绑定
- 语法：** :MOTor:STARtup:TRACe {<Function>,<Element>}
- :MOTor:STARtup:TRACe?
- <Function> = U、I、P、F、UIMX、IMX、PMX、PMN、TIMX、NIMX、FMX、FMN
- <Element> = 1 to 6 (element)、A、B、C(wiring)、ALL(极值可用)
- 实例：** :MOTor:STARtup:TRACe U,1 ►►
- :MOTor:STARtup:TRACe? ►► :MOTOR:STARTUP:TRACE U,1
- 说明：** 关于<Function>和<Element>的详情请参考[电机测试数据类型列表\(1\)](#)

:MOTor:STARtup:SEND?

功能： 查询启动测试数据

语法： :MOTor:STARtup:SEND? {<NRf1>[,<NRf2>]}

<NRf1> = 1 to 5000(数据发送起始点)

<NRf2> = 1 to 5000(数据发送结束点)

实例： 一、当数据类型为 **Data(数据)**时: (固定回复 **Float** 类型的数据)

步骤 1、:MOTor:STARtup:LENGth? 回复 **length**(当前数据已采集数量)

步骤 2、:MOTor:STARtup:TRACe **U,1**(绑定数据类型 **U1**)

步骤 3、读取启动测试数据 **U1**:

读取**全部数据** :MOTor:STARtup:SEND? ▶▶ **#4(bytes)×length**(从 **1** 到 **length** 个数据依次回送)

读取**部分数据** :MOTor:STARtup:SEND? **1,1000** ▶▶ **#4(bytes)×1000**(从 **1** 到 **1000** 个数据依次回送)

二、当数据类型为 **Extremes(极值)**时: (回复的数据类型与:**NUMeric:FORMat** 命令设置的类型有关)

读取**全部数据** :MOTor:STARtup:TRACe **UIMX,ALL**(绑定数据类型 **UIMX** 的全部数据)

当:NUMeric:FORMat?为 **ASCII** 时: :MOTor:STARtup:SEND? ▶▶ **0.000E+00×9**

当:NUMeric:FORMat?为 **Float** 时: :MOTor:STARtup:SEND? ▶▶ **#4(bytes)×9**

读取**部分数据** :MOTor:STARtup:TRACe **UIMX,1**(绑定数据类型 **UIMX** 的第一通道)

当:NUMeric:FORMat?为 **ASCII** 时: :MOTor:STARtup:SEND? ▶▶ **0.000E+00**

当:NUMeric:FORMat?为 **Float** 时: :MOTor:STARtup:SEND? ▶▶ **#4(bytes)**

说明： 当:MOTor:STARtup:TRACe 设置的数据类型为 **Data** 时，存在<NRf1>及<NRf2>参数项，需满足以下条件：

条件 1、<NRf1>必须小于等于<NRf2>

条件 2、<NRf1>、<NRf2>必须小于等于 **length**

条件 3、<NRf1>和<NRf2>可以省略

若<NRf2>省略，则从第<NRf1>个数据到第 **length** 个数据依次回送

若<NRf1>和<NRf2>都省略，则从第 1 个数据 到 第 **length** 个数据依次回送

注意： 其他电机测试类型（录波测试除外）对数据查询时，查询格式以及回复格式都根据以上标准，后续不再解释说明

:MOTor:RLOCKed?

- 功能：** 查询所有电机堵转测试设置
- 语法：** :MOTor:RLOCKed?
- 实例：** :MOTor:RLOCKed? ►► :MOTOR:RLOCKED:MODE 1;TIMER 1;TRACE U,1

:MOTor:RLOCKed:MODE

- 功能：** 设置或查询堵转测试模式(响应周期)
- 语法：** :MOTor:RLOCKed:MODE {<NRf>}
:MOTor:RLOCKed:MODE?
<NRf> = 1、2、3、4、5、6、7、8
- 实例：** :MOTor:RLOCKed:MODE 1 ►►
:MOTor:RLOCKed:MODE? ►► :MOTOR:RLOCKED:MODE 1

:MOTor:RLOCKed:TIMer

- 功能：** 设置或查询堵转测试持续时间
- 语法：** :MOTor:RLOCKed:TIMer {<NRf>}
:MOTor:RLOCKed:TIMer?
{<NRf>} = 1 to 100(秒)
- 实例：** :MOTor:RLOCKed:TIMer 1 ►►
:MOTor:RLOCKed:TIMer? ►► :MOTOR:RLOCKED:TIMER 1

:MOTor:RLOCKed:RTIME?

- 功能：** 查询堵转测试已运行时间
- 语法：** :MOTor:RLOCKed:RTIME?
- 实例：** :MOTor:RLOCKed:RTIME? ►► :MOTOR:RLOCKED:RTIME 0.0S
- 说明：** 当堵转测试复位时:**MOTor:RLOCKed:RTIME?**默认回复 0.0S(秒)

:MOTor:RLOCKed:STATe?

- 功能：** 查询堵转测试状态
- 语法：** :MOTor:RLOCKed:STATe?
- 实例：** :MOTor:RLOCKed:STATe? ►► RESET
- 说明：** 回复类型: RESet(复位)、START(运行)、STOP(停止)、END(结束)

:MOTor:RLOCKed:START

- 功能：** 开始启动测试
- 语法：** :MOTor:RLOCKed:START
- 实例：** :MOTor:RLOCKed:START ►►
- 说明：** 开始堵转测试时所有堵转测试设置命令均无效

:MOTor:RLOCKed:STOP

- 功能:** 停止堵转测试
- 语法:** :MOTor:RLOCKed:STOP
- 实例:** :MOTor:RLOCKed:STOP ►►
- 说明:** 只有:MOTor:RLOCKed:STATe?为 START(运行)时, 才可以进行停止测试

:MOTor:RLOCKed:RESet

- 功能:** 重置堵转测试
- 语法:** :MOTor:RLOCKed:RESet
- 实例:** :MOTor:RLOCKed:RESet ►►
- 说明:** 只有:MOTor:RLOCKed:STATe?为 STOP(停止)或 END(结束)时, 才可以进行重置测试

:MOTor:RLOCKed:LENGth?

- 功能:** 查询堵转测试数据采集的数量
- 语法:** :MOTor:RLOCKed:LENGth?
- 实例:** :MOTor:RLOCKed:LENGth? ►► :MOTOR:RLOCKED:LENGTH 5000

:MOTor:RLOCKed:TRACe

- 功能:** 设置或查询堵转测试的数据绑定
- 语法:** :MOTor:RLOCKed:TRACe {<Function>,<Element>}
:MOTor:RLOCKed:TRACe?

<Function> = U、I、P、F、UIMX、IMX、PMX、PMN、TIMX、NIMX、FMX、FMN

<Element> = 1 to 6 (element)、A、B、C(wiring)、ALL(极值可用)
- 实例:** :MOTor:RLOCKed:TRACe U,1 ►►
:MOTor:RLOCKed:TRACe? ►► :MOTOR:RLOCKED:TRACE U,1
- 说明:** 关于<Function>和<Element>的详情请参考[电机测试数据类型列表\(2\)](#)

:MOTor:RLOCKed:SEND?

- 功能:** 查询堵转测试数据
- 语法:** :MOTor:RLOCKed:SEND? {<NRf1>[,<NRf2>]}

<NRf1> = 1 to 5000(数据发送起始点)

<NRf2> = 1 to 5000(数据发送结束点)
- 实例:** 当设置:MOTor:RLOCKed:TRACe U,1 时, 当前已经记录 **length** 条数据:
:MOTor:RLOCKed:SEND? ►► #4(bytes)×**length**(从 1 到 **length** 个数据依次回送)
:MOTor:RLOCKed:SEND? 1,1000 ►► #4(bytes)×**1000**(从 1 到 **1000** 个数据依次回送)
- 说明:** 详情请参考:[MOTor:STARtup:SEND?](#)

:MOTor:VOLStore?

功能： 查询所有电机电压存储设置

语法： :MOTor:VOLStore?

实例： :MOTor:VOLStore? ►► :MOTOR:VOLSTORE:MODE 1;TIMER 1;TRACE U,1

:MOTor:VOLStore:MODE

功能： 设置或查询电压存储模式(响应周期)

语法： :MOTor:VOLStore:MODE {<NRf>}

:MOTor:VOLStore:MODE?

<NRf> = 1、2、3、4、5、6、7、8、9、10、12、15、20、25、30、50

实例： :MOTor:VOLStore:MODE 1 ►►

:MOTor:VOLStore:MODE? ►► :MOTOR:VOLSTORE:MODE 1

:MOTor:VOLStore:TIMer

功能： 设置或查询电压存储持续时间

语法： :MOTor:VOLStore:TIMer {<NRf>}

:MOTor:VOLStore:TIMer?

{<NRf>} = 1 to 999(秒)

实例： :MOTor:VOLStore:TIMer 1 ►►

:MOTor:VOLStore:TIMer? ►► :MOTOR:VOLSTORE:TIMER 1

:MOTor:VOLStore:RTIME?

功能： 查询电压存储已运行时间

语法： :MOTor:VOLStore:RTIME?

实例： :MOTor:VOLStore:RTIME? ►► :MOTOR:VOLSTORE:RTIME 0.0S

说明： 当电压存储复位时:**MOTor:VOLStore:RTIME?**默认回复 0.0S(秒)

:MOTor:VOLStore:STATe?

功能： 查询电压存储状态

语法： :MOTor:VOLStore:STATe?

实例： :MOTor:VOLStore:STATe? ►► RESET

说明： 回复类型：RESet(复位)、START(运行)、STOP(停止)、END(结束)

:MOTor:VOLStore:START

功能： 开始电压存储

语法： :MOTor:VOLStore:START

实例： :MOTor:VOLStore:START ►►

说明： 开始电压存储时所有电压存储设置命令均无效

:MOTor:VOLStore:STOP

- 功能：** 停止电压存储
- 语法：** :MOTor:VOLStore:STOP
- 实例：** :MOTor:VOLStore:STOP ►►
- 说明：** 只有:MOTor:VOLStore:STATe?为 START(运行)时，才可以进行停止测试

:MOTor:VOLStore:RESet

- 功能：** 重置电压存储
- 语法：** :MOTor:VOLStore:RESet
- 实例：** :MOTor:VOLStore:RESet ►►
- 说明：** 只有:MOTor:VOLStore:STATe?为 STOP(停止)或 END(结束)时，才可以进行重置测试

:MOTor:VOLStore:LENGth?

- 功能：** 查询电压存储数据采集的数量
- 语法：** :MOTor:VOLStore:LENGth?
- 实例：** :MOTor:VOLStore:LENGth? ►► :MOTOR:VOLSTORE:LENGTH 5000

:MOTor:VOLStore:TRACe

- 功能：** 设置或查询电压存储的数据绑定
- 语法：** :MOTor:VOLStore:TRACe {<Function>,<Element>}
- :MOTor:VOLStore:TRACe?
- <Function> = U、UCF、UF、TIM、UMX、UMN、CFMX、CFMN、NMX、NMN
- <Element> = 1 to 6 (element)、A、B、C(wiring)、ALL(极值可用)
- 实例：** :MOTor:VOLStore:TRACe U,1 ►►
- :MOTor:VOLStore:TRACe? ►► :MOTOR:VOLSTORE:TRACE U,1
- 说明：** 关于<Function>和<Element>的详情请参考[电机测试数据类型列表\(3\)](#)

:MOTor:VOLStore:SEND?

- 功能：** 查询电压存储数据
- 语法：** :MOTor:VOLStore:SEND? {<NRf1>[,<NRf2>]}
- <NRf1> = 1 to 2000(数据发送起始点)
- <NRf2> = 1 to 2000(数据发送结束点)
- 实例：** 当设置:MOTor:VOLStore:TRACe U,1 时，当前已经记录 length 条数据:
- :MOTor:VOLStore:SEND? ►► #4(bytes)×length(从 1 到 length 个数据依次回送)
- :MOTor:VOLStore:SEND? 1,1000 ►► #4(bytes)×1000(从 1 到 1000 个数据依次回送)
- 说明：** 详情请参考:[MOTor:STARtup:SEND?](#)

:MOTor:DATRecord?

- 功能：** 查询所有电机数据记录设置
- 语法：** :MOTor:DATRecord?
- 实例：** :MOTor:DATRecord? ►► :MOTOR:DATRECORD:MODE 1;TIMER 1;TRACE U,1

:MOTor:DATRecord:MODE

- 功能：** 设置或查询数据记录模式(响应周期)
- 语法：** :MOTor:DATRecord:MODE {<NRf>}
:MOTor:DATRecord:MODE?
<NRf> = 1、2、3、4、5、6、7、8、9、10、12、15、20、25、30、50
- 实例：** :MOTor:DATRecord:MODE 1 ►►
:MOTor:DATRecord:MODE? ►► :MOTOR:DATRECORD:MODE 1

:MOTor:DATRecord:TIMer

- 功能：** 设置或查询数据记录持续时间
- 语法：** :MOTor:DATRecord:TIMer {<NRf>}
:MOTor:DATRecord:TIMer?
{<NRf>} = 1 to 99999(秒)
- 实例：** :MOTor:DATRecord:TIMer 1 ►►
:MOTor:DATRecord:TIMer? ►► :MOTOR:DATRECORD:TIMER 1

:MOTor:DATRecord:RTIME?

- 功能：** 查询数据记录已运行时间
- 语法：** :MOTor:DATRecord:RTIME?
- 实例：** :MOTor:DATRecord:RTIME? ►► :MOTOR:DATRECORD:RTIME 0.0S
- 说明：** 当数据记录复位时:**MOTor:DATRecord:RTIME?**默认回复 0.0S(秒)

:MOTor:DATRecord:STATe?

- 功能：** 查询数据记录状态
- 语法：** :MOTor:DATRecord:STATe?
- 实例：** :MOTor:DATRecord:STATe? ►► RESET
- 说明：** 回复类型: RESet(复位)、START(运行)、STOP(停止)、END(结束)

:MOTor:DATRecord:START

- 功能：** 开始数据记录
- 语法：** :MOTor:DATRecord:START
- 实例：** :MOTor:DATRecord:START ►►
- 说明：** 开始数据记录时所有数据记录设置命令均无效

:MOTor:DATRecord:STOP

功能： 停止数据记录

语法： :MOTor:DATRecord:STOP

实例： :MOTor:DATRecord:STOP >>

说明： 只有:MOTor:DATRecord:STATe?为 START(运行)时，才可以进行停止测试

:MOTor:DATRecord:RESet

功能： 重置数据记录

语法： :MOTor:DATRecord:RESet

实例： :MOTor:DATRecord:RESet >>

说明： 只有:MOTor:DATRecord:STATe?为 STOP(停止)或 END(结束)时，才可以进行重置测试

:MOTor:DATRecord:LENGth?

功能： 查询数据记录数据采集的数量

语法： :MOTor:DATRecord:LENGth?

实例： :MOTor:DATRecord:LENGth? >> :MOTOR:DATRECORD:LENGTH 5000

:MOTor:DATRecord:TRACe

功能： 设置或查询数据记录的数据绑定

语法： :MOTor:DATRecord:TRACe {<Function>,<Element>}

:MOTor:DATRecord:TRACe?

<Function> = U、I、P、F、UMA、UMI、IMA、IMI、PMA、PMI、FMA、FMI

<Element> = 1 to 6 (element)、A、B、C(wiring)、ALL(极值可用)

实例： :MOTor:DATRecord:TRACe U,1 >>

:MOTor:DATRecord:TRACe? >> :MOTOR:DATRECORD:TRACE U,1

说明： 关于<Function>和<Element>的详情请参考[电机测试数据类型列表\(4\)](#)

:MOTor:DATRecord:SEND?

功能： 查询数据记录数据

语法： :MOTor:DATRecord:SEND? {<NRf1>[,<NRf2>]}

<NRf1> = 1 to 10000(数据发送起始点)

<NRf2> = 1 to 10000(数据发送结束点)

实例： 当设置:MOTor:DATRecord:TRACe U,1 时，当前已经记录 length 条数据：

:MOTor:DATRecord:SEND? >> #4(bytes)×length(从 1 到 length 个数据依次回送)

:MOTor:DATRecord:SEND? 1,1000 >> #4(bytes)×1000(从 1 到 1000 个数据依次回送)

说明： 详情请参考:[MOTor:STARtup:SEND?](#)

:MOTor:WAVRecord?

功能: 查询所有电机录波测试设置

语法: :MOTor:WAVRecord?

实例: :MOTor:WAVRecord? ►► :MOTOR:WAVRECORD:MODE 25600;TIMER 8;TRACE U1

:MOTor:WAVRecord:MODE

功能: 设置或查询录波测试模式(每秒采集的点数)

语法: :MOTor:WAVRecord:MODE {<NRf>}

:MOTor:WAVRecord:MODE?

<NRf> = 400、800、1600、3200、6400、12800、25600、51200、102400、204800 (每个设置项后面 **00** 可省略)

实例: :MOTor:WAVRecord:MODE 25600 ►►

:MOTor:WAVRecord:MODE? ►► :MOTOR:WAVRECORD:MODE 25600

:MOTor:WAVRecord:TIMer

功能: 设置或查询录波测试持续时间

语法: :MOTor:WAVRecord:TIMer {<NRf>}

:MOTor:WAVRecord:TIMer?

{<NRf>} = 1 to **MAX**(秒)

MAX 的值与:MOTor:WAVRecord:MODE 设置的模式有关, 具体如下:

MAX(MODE) = 512(400)、256(800)、128(1600)、64(3200)、32(6400)、
16(12800)、8(25600)、4(51200)、2(102400)、1(204800)

实例: 当:MOTor:WAVRecord:MODE 设置为 25600 时, :MOTor:WAVRecord:TIMer 最大可设置 **8**(秒)

:MOTor:WAVRecord:TIMer 8 ►►

:MOTor:WAVRecord:TIMer? ►► :MOTOR:WAVRECORD:TIMER 8

说明: 当改变录波模式:MOTor:WAVRecord:MODE 时, 录波测试持续时间默认设置为当前模式可设置的最大值

:MOTor:WAVRecord:RTIME?

功能: 查询录波测试已运行时间

语法: :MOTor:WAVRecord:RTIME?

实例: :MOTor:WAVRecord:RTIME? ►► :MOTOR:WAVRECORD:RTIME 0.0S

说明: 当录波测试复位时:MOTor:WAVRecord:RTIME?默认回复 0.0S

:MOTor:WAVRecord:STATe?

功能: 查询录波测试状态

语法: :MOTor:WAVRecord:STATe?

实例: :MOTor:WAVRecord:STATe? ►► RESET

说明: 回复类型: RESet(录波复位)、STARt(录波启动)、STOP(录波停止)、END(录波结束)

:MOTor:WAVRecord:START

- 功能:** 启动录波
- 语法:** :MOTor:WAVRecord:START
- 实例:** :MOTor:WAVRecord:START ►►
- 说明:** 启动录波时所有录波设置命令均无效

:MOTor:WAVRecord:STOP

- 功能:** 停止录波
- 语法:** :MOTor:WAVRecord:STOP
- 实例:** :MOTor:WAVRecord:STOP ►►
- 说明:** 只有**:MOTor:WAVRecord:STATe?**为 START(录波启动)时, 才可以进行停止录波

:MOTor:WAVRecord:RESet

- 功能:** 重置录波
- 语法:** :MOTor:WAVRecord:RESet
- 实例:** :MOTor:WAVRecord:RESet ►►
- 说明:** 只有**:MOTor:WAVRecord:STATe?**为 STOP(录波停止)、END(录波结束)时, 才可以进行重置录波

:MOTor:WAVRecord:LENGth?

- 功能:** 查询录波测试数据采集的数量
- 语法:** :MOTor:WAVRecord:LENGth?
- 实例:** :MOTor:WAVRecord:LENGth? ►► :MOTOR:WAVRECORD:LENGTH 204800

:MOTor:WAVRecord:TRACe

- 功能:** 设置或查询录波的数据绑定
- 语法:** :MOTor:WAVRecord:TRACe {<Function>,<Element>}
:MOTor:WAVRecord:TRACe?
- <Function> = U、I、UMX、UMN、IMX、IMN、UPK、IPK
<Element> = 1 to 6 (element)、ALL(极值可用)
- 实例:** :MOTor:WAVRecord:TRACe U,1 ►►
:MOTor:WAVRecord:TRACe? ►► :MOTOR:WAVRECORD:TRACE U,1
- 说明:** 关于<Function>和<Element>的详情请参考[电机测试数据类型列表\(5\)](#)

:MOTor:WAVRecord:SEND?

功能： 查询录波测试数据

语法： :MOTor:WAVRecord:SEND? {<NRf1>[,<NRf2>]}

<NRf1> = 1 to 204800(数据发送起始点)

<NRf2> = 1 to 204800(数据发送结束点)

实例： 步骤 1、:MOTor:WAVRecord:LENGth? 回复 **length**(当前数据已采集数量)

步骤 2、:MOTor:WAVRecord:TRACe **U,1**(绑定录波数据 **U1**)

步骤 3、读取录波数据 **U1**: (回送的数据前四个字节为当前通道的电压（电流）的系数)

读取**全部数据** :MOTor:WAVRecord:SEND? ►► **#4(bytes)+#2(bytes)×length**(从 **1** 到 **length** 个数据依次回送)

读取**部分数据** :MOTor:WAVRecord:SEND? **0,2000** ►► **#4(bytes)+#2(bytes)×2000**(从 **0** 到 **2000** 个数据依次回送)

说明： 当:**MOTor:WAVRecord:TRACe** 设置的数据类型为 **Data** 时，存在<NRf1>及<NRf2>参数项，需满足以下条件：

条件 1、<NRf1>必须小于等于<NRf2>

条件 2、<NRf1>、<NRf2>必须小于等于 **length**

条件 3、<NRf1>和<NRf2>可以省略

若<NRf2>省略，则从第<NRf1>个数据到第 **length** 个数据依次回送

若<NRf1>和<NRf2>都省略，则从第 1 个数据 到 第 **length** 个数据依次回送

:MOTor:TRANSient?

功能： 查询所有电机瞬态测试设置

语法： :MOTor:TRANSient?

实例： :MOTor:TRANSient? ►►

:MOTor:TRANSient:MODE

功能： 设置或查询瞬态测试模式(响应周期)

语法： :MOTor:TRANSient:MODE {<NRf>}

:MOTor:TRANSient:MODE?

<NRf> = 1、2、3、4、5、6、7、8

实例： :MOTor:TRANSient:MODE 1 ►►

:MOTor:TRANSient:MODE? ►► :MOTOR:TRANSIENT:MODE 1

:MOTor:TRANSient:TIMer

功能： 设置或查询瞬态测试持续时间

语法： :MOTor:TRANSient:TIMer {<NRf>}

:MOTor:TRANSient:TIMer?

{<NRf>} = 1 to 9999(秒)

实例： :MOTor:TRANSient:TIMer 1 ►►

:MOTor:TRANSient:TIMer? ►► :MOTOR:TRANSIENT:TIMER 1

:MOTor:TRANsient:RTIME?

- 功能：** 查询瞬态测试已运行时间
- 语法：** :MOTor:TRANsient:RTIME?
- 实例：** :MOTor:TRANsient:RTIME? ►► :MOTOR:TRANSIENT:RTIME 0.0S
- 说明：** 当瞬态测试复位时:**MOTor:TRANsient:RTIME?**默认回复 0.0S(秒)

:MOTor:TRANsient:STATe?

- 功能：** 查询瞬态测试状态
- 语法：** :MOTor:TRANsient:STATe?
- 实例：** :MOTor:TRANsient:STATe? ►► RESET
- 说明：** 回复类型：RESet(复位)、STARt(运行)、STOP(停止)、END(结束)

:MOTor:TRANsient:START

- 功能：** 开始瞬态测试
- 语法：** :MOTor:TRANsient:START
- 实例：** :MOTor:TRANsient:START ►►
- 说明：** 开始瞬态测试时所有瞬态测试设置命令均无效

:MOTor:TRANsient:STOP

- 功能：** 停止瞬态测试
- 语法：** :MOTor:TRANsient:STOP
- 实例：** :MOTor:TRANsient:STOP ►►
- 说明：** 只有:**MOTor:TRANsient:STATe?**为 START(运行)时，才可以进行停止测试

:MOTor:TRANsient:RESet

- 功能：** 重置瞬态测试
- 语法：** :MOTor:TRANsient:RESet
- 实例：** :MOTor:TRANsient:RESet ►►
- 说明：** 只有:**MOTor:TRANsient:STATe?**为 STOP(停止)或 END(结束)时，才可以进行重置测试

:MOTor:TRANsient:LENGth?

- 功能：** 查询瞬态测试数据采集的数量
- 语法：** :MOTor:TRANsient:LENGth?
- 实例：** :MOTor:TRANsient:LENGth? ►► :MOTOR:TRANSIENT:LENGTH 5000

:MOTor:TRANsient:TRACe

功能: 设置或查询瞬态测试的数据绑定

语法: :MOTor:TRANsient:TRACe {<Function>,<Element>}

:MOTor:TRANsient:TRACe?

<Function> = U、I、P、F、UMX、UMN、IMX、IMN、PMX、PMN、FMX、FMN

<Element> = 1 to 6 (element)、A、B、C(wiring)、ALL(极值可用)

实例: :MOTor:TRANsient:TRACe U,1 ▶▶

:MOTor:TRANsient:TRACe? ▶▶ :MOTOR:TRANSIENT:TRACE U,1

说明: 关于<Function>和<Element>的详情请参考[电机测试数据类型列表\(6\)](#)

:MOTor:TRANsient:SEND?

功能: 查询瞬态测试数据

语法: :MOTor:TRANsient:SEND? {<NRf1>[,<NRf2>]}

<NRf1> = 1 to 10000(数据发送起始点)

<NRf2> = 1 to 10000(数据发送结束点)

实例: 当设置:MOTor:TRANsient:TRACe U,1 时, 当前已经记录 **length** 条数据:

:MOTor:TRANsient:SEND? ▶▶ **#4**(bytes)×**length**(从 **1** 到 **length** 个数据依次回送)

:MOTor:TRANsient:SEND? 1,1000 ▶▶ **#4**(bytes)×**1000**(从 **1** 到 **1000** 个数据依次回送)

说明: 详情请参考:[MOTor:STARtup:SEND?](#)

:MOTor<x>?

功能: 查询指定电机参数设置

语法: :MOTor<x>?

<x> = 1 or 2 (motor number)(<x>若没有指定，默认为 MOTor1)

实例: :MOTor1? ►► :MOTOR1:SPEED:TYPE PULSE;SCALING 1.000E+00;RANGE 5V;PRANGE 1.000E+04,0.000E+00;
PULSE 60;;MOTOR1:SPEED:LSCALE:KVALUE 1.000E+00;DVALUE 0.000E+00;;MOTOR1:TORQUE:TYPE PULSE;
SCALING 1.000E+00;RANGE 5V;PRANGE 1.000E+04,0.000E+00;;MOTOR1:TORQUE:LSCALE:KVALUE 1.000E+00;
DVALUE 0.000E+00;;MOTOR1:TORQUE:RATE:UPPER 1.000E+04,15.000E+03;LOWER -1.000E+04,5.000E+03;
:MOTOR1:TORQUE:SCHOOL:FREQ 1.000E+02;;MOTOR1:FILTER OFF;;MOTOR1:PM:SCALING 1.000E+00;
:MOTOR1:FILTER OFF;;MOTOR1:SSPEED U1

注意: 1、在仪器支持模拟量输入时，只有一组电机

2、电机参数设置的浮点数请保持在 7 位有效数字及以内，例如：999.9999、9999.999、99999.99、999999.9

:MOTor<x>:SPEEd?

功能: 查询所有电机转速设置

语法: :MOTor<x>:SPEEd?

<x> = 1 or 2 (motor number)

实例: :MOTor1:SPEEd? ►► :MOTOR1:SPEED:TYPE PULSE;SCALING 1.000E+00;RANGE 5V;PRANGE 1.000E+04,
1.000E+00;PULSE 60;;MOTOR1:SPEED:LSCALE:KVALUE 1.000E+00;DVALUE 0.000E+00

:MOTor<x>:SPEEd:TYPE

功能: 设置或查询转速信号输入类型

语法: :MOTor<x>:SPEEd:TYPE {PULSe|ANALog}

:MOTor<x>:SPEEd:TYPE?

<x> = 1 or 2 (motor number)

实例: :MOTor1:SPEEd:TYPE PULSe ►►

:MOTor1:SPEEd:TYPE? ►► :MOTOR1:SPEED:TYPE PULSE

说明: 在仪器支持模拟量输入时此命令设置有效

:MOTor<x>:SPEEd:SCALing

功能: 设置或查询转速计算比例系数

语法: :MOTor<x>:SPEEd:SCALing {<NRf>}

:MOTor<x>:SPEEd:SCALing?

<x> = 1 or 2 (motor number)

<NRf> = 0.0001 to 99999.99

实例: :MOTor1:SPEEd:SCALing 1 ►►

:MOTor1:SPEEd:SCALing? ►► :MOTOR1:SPEED:SCALING 1.000E+00

:MOTor<x>:SPEed:RANGe?

功能: 查询转速信号的输入量程(模拟输入)

语法: :MOTor<x>:SPEed:RANGe?

<x> = 1(motor number)

实例: :MOTor1:SPEed:RANGe? ►► :MOTOR1:SPEED:RANGE 5V

说明: 此命令在仪器支持模拟量输入时有效,模拟量输入量程有 20mA、1V、5V、10V 四种规格

:MOTor<x>:SPEed:LSCale?

功能: 查询所有转速线性比例设置(模拟输入)

语法: :MOTor<x>:SPEed:LSCale?

<x> = 1(motor number)

实例: :MOTor1:SPEed:LSCale? ►► :MOTOR1:SPEED:LSCALE:KVALUE 1.000E+00;DVALUE 0.000E+00

:MOTor<x>:SPEed:LSCale:KVALue

功能: 设置或查询转速线性比例的斜率(模拟输入)

语法: :MOTor<x>:SPEed:LSCale:KVALue {<NRf>}

:MOTor<x>:SPEed:LSCale:KVALue?

<x> = 1(motor number)

<NRf> = 0.0001 to 99999.99

实例: :MOTor1:SPEed:LSCale:KVALue 1.0000 ►►

:MOTor1:SPEed:LSCale:KVALue? ►► :MOTOR1:SPEED:LSCALE:KVALUE 1.000E+00

:MOTor<x>:SPEed:LSCale:DVALue

功能: 设置或查询转速线性比例的偏移量(模拟输入)

语法: :MOTor<x>:SPEed:LSCale:DVALue {<NRf>}

:MOTor<x>:SPEed:LSCale:DVALue?

<x> = 1(motor number)

<NRf> = -99999.99 to 99999.99

实例: :MOTor1:SPEed:LSCale:DVALue 0 ►►

:MOTor1:SPEed:LSCale:DVALue? ►► :MOTOR1:SPEED:LSCALE:DVALUE 0.000E+00

:MOTor<x>:SPEed:PRANge

功能： 设置或查询转速脉冲输入量程(脉冲输入)

语法： :MOTor<x>:SPEed:PRANge {<NRf>,<NRf>}

:MOTor<x>:SPEed:PRANge?

<x> = 1 or 2(motor number)

<NRf> = 0.0 to 10000000.0

The First <NRf> : Upper Limit(上限值)

The Second <NRf> : Lower Limit(下限值)

实例： :MOTor1:SPEed:PRANge 10000.0,0.0 ▶▶

:MOTor1:SPEed:PRANge? ▶▶ :MOTOR1:SPEED:PRANGE 1.000E+04,0.000E+00

:MOTor<x>:SPEed:PULSe

功能： 设置或查询转速每转脉冲的个数(脉冲输入)

语法： :MOTor<x>:SPEed:PULSe {<NRf>}

:MOTor<x>:SPEed:PULSe?

<x> = 1 or 2 (motor number)

<NRf> = 1 to 9999

实例： :MOTor1:SPEed:PULSe 60 ▶▶

:MOTor1:SPEed:PULSe? ▶▶ :MOTOR1:SPEED:PULSE 60

说明： 当 **:MOTor<x>:SPEed:TYPE** 设置为 PULSe(脉冲输入)时，此命令有效

:MOTor<x>:TORQue?

功能： 查询所有电机扭矩设置

语法： :MOTor<x>:TORQue?

<x> = 1 or 2 (motor number)

实例： :MOTor1:TORQue? ▶▶ :MOTOR1:TORQUE:TYPE PULSE;SCALING 1.000E+00;RANGE 5V;PRANGE 1.000E+04,
0.000E+00;;MOTOR1:TORQUE:LSCALE:KVALUE 1.000E+00;DVALUE 0.000E+00;;MOTOR1:TORQUE:RATE:
UPPER 1.000E+04,15.000E+03;LOWER -1.000E+04,5.000E+03;;MOTOR1:TORQUE:SCHOOL:FREQ 1.000E+02

:MOTor<x>:TORQue:TYPE

功能： 设置或查询扭矩信号输入类型

语法： :MOTor<x>:TORQue:TYPE {PULSe|ANALog}

:MOTor<x>:TORQue:TYPE?

<x> = 1 or 2 (motor number)

实例： :MOTor1:TORQue:TYPE PULSe ▶▶

:MOTor1:TORQue:TYPE? ▶▶ :MOTOR1:TORQUE:TYPE PULSE

说明： 在仪器支持模拟量输入时此命令设置有效

:MOTor<x>:TORQue:SCALing

功能： 设置或查询扭矩计算比例系数

语法： :MOTor<x>:TORQue:SCALing {<NRf>}

:MOTor<x>:TORQue:SCALing?

<x> = 1 or 2 (motor number)

<NRf> = 0.0001 to 99999.99

实例： :MOTor1:TORQue:SCALing 1 ►►

:MOTor1:TORQue:SCALing? ►► :MOTOR1:TORQUE:SCALING 1.000E+00

:MOTor<x>:TORQue:RANGe?

功能： 查询扭矩信号的输入量程(模拟输入)

语法： :MOTor<x>:TORQue:RANGe?

<x> = 1(motor number)

实例： :MOTor1:TORQue:RANGe? ►► :MOTOR1:TORQUE:RANGE 5V

说明： 此命令在仪器支持模拟量输入时有效,模拟量输入量程有 20mA、1V、5V、10V 四种规格

:MOTor<x>:TORQue:LSCale?

功能： 查询所有扭矩线性比例设置(模拟输入)

语法： :MOTor<x>:TORQue:LSCale?

<x> = 1(motor number)

实例： :MOTor1:TORQue:LSCale? ►► :MOTOR1:TORQUE:LSCALE:KVALUE 1.000E+00;DVALUE 0.000E+00

:MOTor<x>:TORQue:LSCale:KVALue

功能： 设置或查询扭矩线性比例的斜率

语法： :MOTor<x>:TORQue:LSCale:KVALue {<NRf>}

:MOTor<x>:TORQue:LSCale:KVALue?

<x> = 1(motor number)

<NRf> = 0.0001 to 99999.99

实例： :MOTor1:TORQue:LSCale:KVALue 1.0000 ►►

:MOTor1:TORQue:LSCale:KVALue? ►► :MOTOR1:TORQUE:LSCALE:KVALUE 1.000E+00

:MOTor<x>:TORQue:LSCale:DVALue

功能: 设置或查询扭矩线性比例的偏移量

语法: :MOTor<x>:TORQue:LSCale:DVALue {<NRf>}

:MOTor<x>:TORQue:LSCale:DVALue?

<x> = 1(motor number)

<NRf> = -99999.99 to 99999.99

实例: :MOTor1:TORQue:LSCale:DVALue 0 ▶▶

:MOTor1:TORQue:LSCale:DVALue? ▶▶ :MOTOR1:TORQUE:LSCALE:DVALUE 0.000E+00

:MOTor<x>:TORQue:PRANge

功能: 设置或查询扭矩脉冲输入量程(脉冲输入)

语法: :MOTor<x>:TORQue:PRANge {<NRf>,<NRf>}

:MOTor<x>:TORQue:PRANge?

<x> = 1 or 2(motor number)

<NRf> = -999999.9 to 999999.9

The First <NRf> : Upper Limit(上限值)

The Second <NRf> : Lower Limit(下限值)

实例: :MOTor1:TORQue:PRANge 10000.0,0.0 ▶▶

:MOTor1:TORQue:PRANge? ▶▶ :MOTOR1:TORQUE:PRANGE 1.000E+04,0.000E+00

:MOTor<x>:TORQue:RATE?

功能: 查询所有扭矩信号的额定值设置(脉冲输入)

语法: :MOTor<x>:TORQue:RATE?

<x> = 1 or 2 (motor number)

实例: :MOTor1:TORQue:RATE? ▶▶ :MOTOR1:TORQUE:RATE:UPPER 1.000E+04,15.000E+03;

LOWER -1.000E+04,5.000E+03

:MOTor<x>:TORQue:RATE:{UPPer|LOWer}

功能: 设置或查询扭矩信号的额定值和额定频率(脉冲输入)

语法: :MOTor<x>:TORQue:RATE:{UPPer|LOWer} {<NRf>,<Frequency>}

:MOTor<x>:TORQue:RATE:{UPPer|LOWer}?

<x> = 1 or 2 (motor number)

<NRf> = -999999.9 to 999999.9

<Frequency> = 1Hz to 100MHz(单位 HZ、KHZ、MHZ)

实例: :MOTor1:TORQue:RATE:UPPer 10000.0,15KHZ ▶▶

:MOTor1:TORQue:RATE:UPPer? ▶▶ :MOTOR1:TORQUE:RATE:UPPER 1.000E+04,15.000E+03

:MOTor<x>:TORQue:SCHool

功能: 扭矩校零

语法: :MOTor<x>:TORQue:SCHool {Run}
<x> = 1 or 2 (motor number)

实例: :MOTor1:TORQue:SCHool Run ▶▶

:MOTor<x>:TORQue:SCHool:FREQ

功能: 设置或查询扭矩校零频率

语法: :MOTor<x>:TORQue:SCHool:FREQ {<Frequency>}
:MOTor<x>:TORQue:SCHool:FREQ?
<x> = 1 or 2 (motor number)
<Frequency> = 1HZ to 100KHZ(单位 HZ、KHZ)

实例: :MOTor1:TORQue:SCHool:FREQ 100HZ ▶▶
:MOTor1:TORQue:SCHool:FREQ? ▶▶ :MOTOR1:TORQUE:SCHOOL:FREQ 1.000E+02

:MOTor<x>:FILTer

功能: 设置或查询电机线路滤波

语法: :MOTor<x>:FILTer {OFF|<Frequency>}
:MOTor<x>:FILTer?
<x> = 1 or 2 (motor number)
OFF = 线路滤波器关闭
<Frequency> = 100HZ、1KHZ(线路滤波器的截止频率)

实例: :MOTor1:FILTer OFF ▶▶
:MOTor1:FILTer? ▶▶ :MOTOR1:FILTER OFF

:MOTor<x>:PM:SCALing

功能: 设置或查询电机机械功率比例系数

语法: :MOTor<x>:PM:SCALing {<NRf>}
:MOTor<x>:PM:SCALing?
<x> = 1 or 2 (motor number)
<NRf> = 0.0001 to 99999.9999

实例: :MOTor1:PM:SCALing 1.0000 ▶▶
:MOTor1:PM:SCALing? ▶▶ :MOTOR1:PM:SCALING 1.000E+00

:MOTor<x>:POLE

功能: 设置或查询电机极数

语法: :MOTor<x>:POLE {<NRf>}

:MOTor<x>:POLE?

<x> = 1 or 2 (motor number)

<NRf> = 1 to 99

实例: :MOTor1:POLE 2 ►►

:MOTor1:POLE? ►► :MOTOR1:POLE 2

:MOTor<x>:SYNChronize

功能: 设置或查询转速和扭矩计算的同步源

语法: :MOTor<x>:SYNChronize {NONE|U<x>|I<x>}

:MOTor<x>:SYNChronize?

<x> = 1 or 2 (motor number)

<x> = 1 to 6 (element)

实例: :MOTor1:SYNChronize U1 ►►

:MOTor1:SYNChronize ? ►► :MOTOR1:SYNCHRONIZE U1

:MOTor<x>:SSpeed

功能: 设置或查询电机同步转速计算的频率测量源

语法: :MOTor<x>:SSpeed {U<x>|I<x>}

:MOTor<x>:SSpeed?

<x> = 1 or 2 (motor number)

<x> = 1 to 6 (element)

实例: :MOTor1:SSpeed U1 ►►

:MOTor1:SSpeed? ►► :MOTOR1:SSPEED U1

2.12、系统设置相关命令 (SYSTem Group)

:SYSTem:RESolution

功能： 设置或查询数值数据显示分辨率

语法： :SYSTem:RESolution {4|5}

:SYSTem:RESolution?

实例： :SYSTem:RESolution 4 >>

:SYSTem:RESolution? >> :SYSTEM:RESOLUTION 4

说明： 此命令是设置:WAVeform:SEND?、:NUMeric:LIST:VALue?、:NUMeric[:NORMal]:VALue? 召测 ASCII 小数的位数

例如:

:SYSTem:RESolution 4 >>

:NUMeric:VALue? 1 >> 1.000E+01

:SYSTem:RESolution 5 >>

:NUMeric:VALue? 1 >> 1.0000E+01

2.13、更新相关命令 (RATE Group)

:RATE[:RATE]

功能： 设置或查询数据更新间隔

语法： :RATE[:RATE] {<Time>|AUTO}

:RATE[:RATE]?

<Time> = 50, 100, 200, 500 (ms), 1, 2, 5, 10, 20 (s)

实例： :RATE 500MS >>

:RATE? >> :RATE 500.0E-03

说明： 若不带单位默认为秒(S)，若设置值大于可设置最大值则默认设置为可设置最大值，即 20S

:RATE:AUTO?

功能： 查询数据更新间隔设置为“自动”时的所有适用设置

语法： :RATE:AUTO?

实例： :RATE:AUTO? >> :RATE:AUTO:TIMEOUT 1

:RATE:AUTO:TIMEout

功能： 设置或查询数据更新间隔设置为自动时的超时时间

语法： :RATE:AUTO:TIMEout {<NRf>}

:RATE:AUTO:TIMEout?

<NRf> = 1, 5, 10, 20 (s)?

实例： :RATE:AUTO:TIMEout 1

:RATE:AUTO:TIMEout? >> :RATE:AUTO:TIMEOUT 1

2.14、保持相关命令 (HOLD Group)

:HOLD

功能： 设置或者查询数据保持状态的开启或关闭

语法： :HOLD {<Boolean>}

:HOLD?

实例： :HOLD ON ▶▶

:HOLD? ▶▶ :HOLD 1

2.15、通用命令 (Common Command Group)

*IDN?

功能： 查询仪表型号

语法： *IDN?

实例： *IDN? ▶▶ QINGZHI,8966C1-5A3-50A3,8920071001,F1.01

说明： 返回的信息包括：<生产厂商>、<型号>、<序列号>、<固件版本>

*CLS

功能： 清除所有寄存器

语法： *CLS

实例： *CLS ▶▶

说明： 清除所有 SCPI 通信相关寄存器，包括自定义寄存器

*RST

功能： 初始化仪表

语法： *RST

实例： *RST ▶▶

说明： 所有设置参数除通信参数外均恢复到出厂值

附录

1、显示模式说明表

指令	模式名称
NUMeric	数值显示
WAVE	波形显示
TRENd	趋势显示
BAR	棒图显示
VECTor	矢量显示
NWAVE	数值和波形显示
NTRend	数值和趋势显示
NBAR	数值和条形图显示
NVECTor	数值和矢量显示
WNUMeric	波形和数值显示
WTRend	波形和趋势显示
WBAR	波形和条形图显示
WVECTor	波形和矢量显示
TNUMeric	趋势和数值显示
TWAVE	趋势和波形显示
TBAR	趋势和条形图显示
TVECTor	趋势和矢量显示
HSPeed	高速数据捕获

2、数值数据项的预设模式表

模式 1:

ITEM<x>	<Function>	<Element>
1	URMS	1
2	IRMS	1
3	P	1
4	S	1
5	Q	1
6	LAMBda	1
7	PHI	1
8	FU	1
9	FI	1
10	NONE	
11 to 19	URMS to FI	2
20	NONE	
21 to 29	URMS to FI	3
30	NONE	
31 to 39	URMS to FI	4
40	NONE	
41 to 49	URMS to FI	5
50	NONE	
51 to 59	URMS to FI	6
60	NONE	
61 to 69	URMS to FI	SIGMA
70	NONE	
71 to 79	URMS to FI	SIGMB
80	NONE	
81 to 89	URMS to FI	SIGMC
90	NONE	
91 to 1000	NONE	

模式 2:

ITEM<x>	<Function>	<Element>
1	URMS	1
2	UMN	1
3	UDC	1
4	UAC	1
5	IRMS	1
6	IMN	1
7	IDC	1
8	IAC	1
9	P	1
10	S	1
11	Q	1
12	LAMBda	1
13	PHI	1
14	FU	1
15	FI	1
16 to 30	URMS to FI	2
31 to 45	URMS to FI	3
46 to 60	URMS to FI	4
61 to 75	URMS to FI	5
76 to 90	URMS to FI	6
91 to 105	URMS to FI	SIGMA
106 to 120	URMS to FI	SIGMB
121 to 135	URMS to FI	SIGMC
136 to 1000	NONE	

模式 3:

ITEM<x>	<Function>	<Element>
1	URMS	1
2	UMN	1
3	UDC	1
4	UAC	1
5	IRMS	1
6	IMN	1
7	IDC	1
8	IAC	1
9	P	1
10	S	1
11	Q	1
12	LAMBda	1
13	PHI	1
14	FU	1
15	FI	1
16	UPPeak	1
17	UMPeak	1
18	IPPeak	1
19	IMPeak	1
20	NONE	
21 to 39	URMS to IMPeak	2

40	NONE	
41 to 59	URMS to IMPeak	3
60	NONE	
61 to 79	URMS to IMPeak	4
80	NONE	
81 to 99	URMS to IMPeak	5
100	NONE	
101 to 119	URMS to IMPeak	6
120	NONE	
121 to 139	URMS to IMPeak	SIGMA
140	NONE	
141 to 159	URMS to IMPeak	SIGMB
160	NONE	
161 to 179	URMS to IMPeak	SIGMC
180	NONE	
181 to 1000	NONE	

模式 4:

ITEM<x>	<Function>	<Element>
1	URMS	1
2	UMN	1
3	UDC	1
4	UAC	1
5	IRMS	1
6	IMN	1
7	IDC	1
8	IAC	1
9	P	1
10	S	1
11	Q	1
12	FU	1
13	FI	1
14	TIME	1
15	WH	1
16	WHP	1
17	WHM	1
18	AH	1
19	AHP	1
20	AHM	1
21 to 40	URMS to AHM	2
41 to 60	URMS to AHM	3
61 to 80	URMS to AHM	4
81 to 100	URMS to AHM	5
101 to 120	URMS to AHM	6
121 to 140	URMS to AHM	SIGMA
141 to 160	URMS to AHM	SIGMB
161 to 180	URMS to AHM	SIGMC
181 to 1000	NONE	

3、谐波数值数据项的预设模式表

模式 1:

ITEM<x>	<Function>	<Element>
1	U	1
2	I	1
3	NONE	
4	U	2
5	I	2
6	NONE	
7	U	3
8	I	3
9	NONE	
10	U	4
11	I	4
12	NONE	
13	U	5
14	I	5
15	NONE	
16	U	6
17	I	6
18 to 64	NONE	

4、<Function>选项列表

<Function>	Function Name	<Element>	<Order>
URMS	Urms	需要	不需要
UMN	Umn	需要	不需要
UDC	Udc	需要	不需要
URMN	Urmn	需要	不需要
UAC	Uac	需要	不需要
IRMS	Irms	需要	不需要
IMN	Imn	需要	不需要
IDC	Idc	需要	不需要
IRMN	Irmn	需要	不需要
IAC	Iac	需要	不需要
P	P	需要	不需要
S	S	需要	不需要
Q	Q	需要	不需要
LAMBda	λ	需要	不需要
PHI	φ	需要	不需要
FU	FreqU(fU)	需要	不需要
FI	FreqI(fI)	需要	不需要
UPPeak	U+peak(U+pk)	需要	不需要
UMPeak	U-peak(U-pk)	需要	不需要
IPPeak	I+peak(I+pk)	需要	不需要
IMPeak	I-peak(I-pk)	需要	不需要
CFU	CfU	需要	不需要
CFI	CfI	需要	不需要
PC	Pc	需要	不需要
PPPeak	P+peak(P+pk)	需要	不需要
PMPeak	P-peak(P-pk)	需要	不需要
TIME	Time	需要	不需要
WH	WP	需要	不需要
WHP	WP+	需要	不需要
WHM	WP-	需要	不需要
AH	q	需要	不需要
AHP	q+	需要	不需要
AHM	q-	需要	不需要
WS	WS	需要	不需要
WQ	WQ	需要	不需要
ETA1 to ETA6	η_1 to η_6	不需要	不需要
EV1 to EV8	Event1 to Event8	不需要	不需要
UK	U(k)	需要	需要
IK	I(k)	需要	需要
PK	P(k)	需要	需要

SK	S(k)	需要	需要
QK	Q(k)	需要	需要
LAMBDK	$\lambda(k)$	需要	需要
PHIK	$\phi(k)$	需要	需要
PHIUk	$\phi_U(k)$	需要	需要
PHIk	$\phi_I(k)$	需要	需要
Zk	Z(k)	需要	需要
RSk	R _s (k)	需要	需要
XSk	X _s (k)	需要	需要
RPk	R _p (k)	需要	需要
XPk	X _p (k)	需要	需要
UHDFk	U _{hdf} (k)	需要	需要
IHDFk	I _{hdf} (k)	需要	需要
PHDFk	Ph _{df} (k)	需要	需要
UTHD	U _{thd}	需要	不需要
ITHD	I _{thd}	需要	不需要
PTHD	P _{thd}	需要	不需要
UTHF	U _{thf}	需要	不需要
ITHF	I _{thf}	需要	不需要
UTIF	U _{tif}	需要	不需要
ITIF	I _{tif}	需要	不需要
HVF	h _{vf}	需要	不需要
HCF	h _{cf}	需要	不需要
KFActor	K-factor	需要	不需要
PHI_U1U2	$\phi_{U_i-U_j}$	需要	不需要
PHI_U1U3	$\phi_{U_i-U_k}$	需要	不需要
PHI_U1I1	$\phi_{U_i-I_i}$	需要	不需要
PHI_U2I2	$\phi_{U_j-I_j}$	需要	不需要
PHI_U3I3	$\phi_{U_j-I_k}$	需要	不需要
FPLL1	fPLL1	不需要	不需要
FPLL2	fPLL2	不需要	不需要
SPEed	Speed	不需要	不需要
TORQue	Torque	不需要	不需要
SYNCsp	SyncSp	不需要	不需要
SLIP	Slip	不需要	不需要
PM	P _m	不需要	不需要

5、电机测试数据类型列表

(1)启动测试

Function Name	<Function>	<Element>	Type
电压	U	1-6 A B C	Data
电流	I	1-6 A B C	Data
有功功率	P	1-6 A B C	Data
基准相频率	F	-	Data
电流最大值时的电压值	UIMX	1-6 A B C ALL	Extremes
电流最大值	IMX	1-6 A B C ALL	Extremes
有功功率最大值	PMX	1-6 A B C ALL	Extremes
有功功率最小值	PMN	1-6 A B C ALL	Extremes
电流最大值的产生时间	TIMX	1-6 A B C ALL	Extremes
电流最大值的产生条数	NIMX	1-6 A B C ALL	Extremes
基准相频率最大值	FMX	-	Extremes
基准相频率最小值	FMN	-	Extremes

(2)堵转测试

Function Name	<Function>	<Element>	Type
电压	U	1-6 A B C	Data
电流	I	1-6 A B C	Data
有功功率	P	1-6 A B C	Data
基准相频率	F	-	Data
电流最大值时的电压值	UIMX	1-6 A B C ALL	Extremes
电流最大值	IMX	1-6 A B C ALL	Extremes
有功功率最大值	PMX	1-6 A B C ALL	Extremes
有功功率最小值	PMN	1-6 A B C ALL	Extremes
电流最大值的产生时间	TIMX	1-6 A B C ALL	Extremes
电流最大值的产生条数	NIMX	1-6 A B C ALL	Extremes
基准相频率最大值	FMX	-	Extremes
基准相频率最小值	FMN	-	Extremes

(3)电压存储

Function Name	<Function>	<Element>	Type
电压	U	1-6 A B C	Data
电压峰值因数	UCF	1-6 A B C	Data
电压频率	UF	1-6	Data
每条数据发生的时间	TIM	-	Data
电压最大值	UMX	1-6 A B C ALL	Extremes
电压最小值	UMN	1-6 A B C ALL	Extremes
电压峰值因数最大值	CFMX	1-6 ALL	Extremes
电压峰值因数最小值	CFMN	1-6 ALL	Extremes
电压最大值的产生条数	NMX	1-6 A B C ALL	Extremes
电压最小值的产生条数	NMN	1-6 A B C ALL	Extremes

(4)数据记录

Function Name	<Function>	<Element>	Type
电压	U	1-6 A B C	Data
电流	I	1-6 A B C	Data
有功功率	P	1-6 A B C	Data
基准相频率	F	-	Data
电压最大值	UMX	1-6 A B C ALL	Extremes
电压最小值	UMN	1-6 A B C ALL	Extremes
电流最大值	IMX	1-6 A B C ALL	Extremes
电流最小值	IMN	1-6 A B C ALL	Extremes
有功功率最大值	PMX	1-6 A B C ALL	Extremes
有功功率最小值	PMN	1-6 A B C ALL	Extremes
基准相频率最大值	FMX	-	Extremes
基准相频率最小值	FMN	-	Extremes

(5)录波测试

Function Name	<Function>	<Element>	Type
电压	U	1-6	Data
电流	I	1-6	Data
电压最大值	UMX	1-6 ALL	Extremes
电压最小值	UMN	1-6 ALL	Extremes
电流最大值	IMX	1-6 ALL	Extremes
电流最小值	IMN	1-6 ALL	Extremes
电压峰值最大值	UPK	1-6 ALL	Extremes
电流峰值最大值	IPK	1-6 ALL	Extremes

(6)瞬态测试

Function Name	<Function>	<Element>	Type
电压	U	1-6 A B C	Data
电流	I	1-6 A B C	Data
有功功率	P	1-6 A B C	Data
基准相频率	F	-	Data
电压最大值	UMX	1-6 A B C ALL	Extremes
电压最小值	UMN	1-6 A B C ALL	Extremes
电流最大值	IMX	1-6 A B C ALL	Extremes
电流最小值	IMN	1-6 A B C ALL	Extremes
有功功率最大值	PMX	1-6 A B C ALL	Extremes
有功功率最小值	PMN	1-6 A B C ALL	Extremes
基准相频率最大值	FMX	-	Extremes
基准相频率最小值	FMN	-	Extremes