

# SMM3000X 系列

## 数字源表

编程手册

CN01A

深圳市鼎阳科技股份有限公司  
SIGLENT TECHNOLOGIES CO.,LTD



## 目录

<b>1 编程概述 .....</b>	<b>10</b>
1.1 建立通信 .....	10
1.1.1 NI-VISA 的安装 .....	10
1.1.2 连接仪器 .....	13
1.2 远程控制的实现 .....	14
1.2.1 用户自定义程序 .....	14
1.2.2 通过 NI-MAX 发送 SCPI 命令 .....	14
1.2.3 通过 Telnet 发送 SCPI 命令 .....	14
1.2.4 通过 Socket 发送 SCPI .....	15
<b>2 SCPI 语言简介 .....</b>	<b>16</b>
2.1 有关命令和查询 .....	16
2.2 描述 .....	16
2.3 用法 .....	16
2.4 命令符号 .....	16
2.5 数字后缀 .....	16
2.6 数据输出格式 .....	17
<b>3 公共命令 .....</b>	<b>20</b>
3.1 *IDN? .....	20
3.2 *CAL? .....	20
3.3 *CLS .....	20
3.4 *ESE .....	21
3.5 *ESR? .....	21
3.6 *OPC .....	21
3.7 *RCL .....	22
3.8 *rst .....	22
3.9 *SAV .....	22
3.10 *SRE .....	22
3.11 *STB? .....	23
3.12 *TRG .....	23

3.13 *TST?	23
3.14 *WAI	24
<b>4 Subsystem Commands 子系统命令</b>	<b>25</b>
4.1 CALCulate Subsystem 计算子系统	25
4.1.1 :CALCulate:CLIMits:CLEAR:AUTO	25
4.1.2 :CALCulate:CLIMits:CLEAR:AUTO:DELay	25
4.1.3 :CALCulate:CLIMits:CLEAR[:IMMediate]	26
4.1.4 :CALCulate:CLIMits:<FAIL PASS>:DIGital[:DATA]	26
4.1.5 :CALCulate:CLIMits:MODE	26
4.1.6 :CALCulate:CLIMits:STATe	27
4.1.7 :CALCulate:CLIMits:STATe:ANY?	27
4.1.8 :CALCulate:CLIMits:UPDate	27
4.1.9 :CALCulate:DATA?	28
4.1.10 :CALCulate:DATA:LATest?	29
4.1.11 :CALCulate:DIGital:BIT	29
4.1.12 :CALCulate:DIGital:<BUSY EOT SOT>	30
4.1.13 :CALCulate:FEED	30
4.1.14 :CALCulate:LIMit:COMPliance:DIGital[:DATA]	31
4.1.15 :CALCulate:LIMit:COMPliance:FAIL	31
4.1.16 :CALCulate:LIMit:FAIL?	32
4.1.17 :CALCulate:LIMit:FUNCTION	32
4.1.18 :CALCulate:LIMit:<LOWER UPPER>	32
4.1.19 :CALCulate:LIMit:<LOWER UPPER>:DIGital[:DATA]	33
4.1.20 :CALCulate:LIMit:PASS:DIGital[:DATA]	33
4.1.21 :CALCulate:LIMit:STATe	33
4.1.22 :CALCulate:MATH:DATA?	34
4.1.23 :CALCulate:MATH:DATA:LATest?	35
4.1.24 :CALCulate:MATH[:EXPRESSION]:CATalog?	35
4.1.25 :CALCulate:MATH[:EXPRESSION]][:DEFine]	35
4.1.26 :CALCulate:MATH[:EXPRESSION]:DElete:ALL	38
4.1.27 :CALCulate:MATH[:EXPRESSION]:DElete[:SELected]	38
4.1.28 :CALCulate:MATH[:EXPRESSION]:NAME	38
4.1.29 :CALCulate:MATH:STATe	39
4.1.30 :CALCulate:MATH:UNITs	39

4.1.31	:CALCulate:OFFSet.....	40
4.1.32	:CALCulate:OFFSet:ACQuire.....	40
4.1.33	:CALCulate:OFFSet:STATe.....	40
4.2	DISPlay Subsystem 显示子系统 .....	41
4.2.1	:DISPlay:DIGits .....	41
4.3	FETCh Subsystem.....	41
4.3.1	:FETCh:ARRay? .....	41
4.3.2	:FETCh:ARRay:<CURRent RESistance SOURce STATus TIME VOLTage>? .....	42
4.3.3	:FETCh[:SCALar]?.....	43
4.3.4	:FETCh[:SCALar]:<CURRent RESistance SOURce STATus TIME VOLTage>? .....	44
4.4	FORMat Subsystem.....	45
4.4.1	:FORMat:BORDer .....	45
4.4.2	:FORMat[:DATA] .....	45
4.4.3	:FORMat:DIGital .....	46
4.4.4	:FORMat:ELEMents:CALCulate .....	46
4.4.5	:FORMat:ELEMents:SENSe .....	47
4.4.6	:FORMat:SREGister .....	47
4.5	HCOPy Subsystem.....	48
4.5.1	:HCOPy:SDUMp:DATA? .....	48
4.5.2	:HCOPy:SDUMp:FORMat .....	48
4.6	MEASure Subsystem .....	49
4.6.1	:MEASure?.....	49
4.6.2	:MEASure:<CURRent RESistance VOLTage>? .....	49
4.7	MMEMemory Subsystem .....	50
4.7.1	:MMEMemory:CATalog? .....	50
4.7.2	:MMEMemory:CDIRectory.....	51
4.7.3	:MMEMemory:COPY .....	51
4.7.4	:MMEMemory:DElete.....	52
4.7.5	:MMEMemory:LOAD:MACRo.....	52
4.7.6	:MMEMemory:LOAD:STATe .....	52
4.7.7	:MMEMemory:MDIRectory .....	53
4.7.8	:MMEMemory:MOVE .....	53
4.7.9	:MMEMemory:RDIRectory .....	53
4.7.10	:MMEMemory:STORe:DATA<:LIMit MATH :SENSel[:ALL]> .....	54

4.7.11 :MMEMemory:STORe:MACRo .....	54
4.7.12 :MMEMemory:STORe:STATe .....	55
4.7.13 :MMEMemory:STORe:TRACe .....	55
4.8 OUTPut Subsystem .....	56
4.8.1 :OUTPut:FILTter:AUTO .....	56
4.8.2 :OUTPut:FILTter[:LPAsS]:FREQuency .....	56
4.8.3 :OUTPut:FILTter[:LPAsS][:STATe] .....	57
4.8.4 :OUTPut:FILTter[:LPAsS]:TCONstant .....	57
4.8.5 :OUTPut:HCApacitance[:STATe] .....	58
4.8.6 :OUTPut:LOW .....	58
4.8.7 :OUTPut:OFF:AUTO .....	58
4.8.8 :OUTPut:OFF:MODE .....	59
4.8.9 :OUTPut:ON:AUTO .....	59
4.8.10 :OUTPut:PROTection[:STATe] .....	60
4.8.11 :OUTPut:RECall .....	60
4.8.12 :OUTPut:SAVE .....	61
4.8.13 :OUTPut[:STATe] .....	61
4.9 READ Subsystem .....	61
4.9.1 :READ:ARRay? .....	61
4.9.2 :READ:ARRay:<CURRent RESistance SOURce STATus TIME VOLTage>? .....	62
4.9.3 :READ[:SCALar]? .....	63
4.9.4 :READ[:SCALar]:<CURRent RESistance SOURce STATus TIME VOLTage>? .....	64
4.10 SENSe Subsystem .....	64
4.10.1 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:APERture .....	64
4.10.2 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:APERture:AUTO .....	65
4.10.3 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:NPLCycles .....	66
4.10.4 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:NPLCycles:AUTO .....	66
4.10.5 :SENSe:<CURRent[:DC] VOLTage[:DC]>:PROTection[:LEVel][:BOTH] .....	67
4.10.6 :SENSe:<CURRent[:DC] VOLTage[:DC]>:PROTection[:LEVel]:NEGative .....	67
4.10.7 :SENSe:<CURRent[:DC] VOLTage[:DC]>:PROTection[:LEVel]:POSitive .....	68
4.10.8 :SENSe:<CURRent[:DC] VOLTage[:DC]>:PROTection:TRIPped? .....	68
4.10.9 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:RANGE:AUTO .....	69
4.10.10 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:RANGE:AUTO:LLIMit .....	69
4.10.11 :SENSe:<CURRent[:DC] VOLTage[:DC]>:RANGE:AUTO:MODE .....	70

4.10.12 :SENSe:<CURRent[:DC] VOLTage[:DC]>:RANGE:AUTO:THReShold .....	71
4.10.13 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:RANGE:AUTO:ULIMit.....	71
4.10.14 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:RANGE[:UPPer] .....	72
4.10.15 :SENSe:DATA? .....	73
4.10.16 :SENSe:DATA:LATest? .....	73
4.10.17 :SENSe:FUNCTION:OFF .....	74
4.10.18 :SENSe:FUNCTION:OFF:ALL .....	74
4.10.19 :SENSe:FUNCTION:OFF:COUNT? .....	74
4.10.20 :SENSe:FUNCTION[:ON] .....	75
4.10.21 :SENSe:FUNCTION[:ON]:ALL.....	75
4.10.22 :SENSe:FUNCTION[:ON]:COUNT? .....	75
4.10.23 :SENSe:FUNCTION:STATe?.....	76
4.10.24 :SENSe:REMote .....	76
4.10.25 :SENSe:RESistance:MODE .....	76
4.10.26 :SENSe:RESistance:OCOMPensated.....	77
4.10.27 :SENSe:TOUTput:SIGNAl .....	78
4.10.28 :SENSe:TOUTput[:STATe] .....	78
4.10.29 :SENSe:WAIT:AUTO .....	78
4.10.30 :SENSe:WAIT:GAIN.....	79
4.10.31 :SENSe:WAIT:OFFSet .....	79
4.10.32 :SENSe:WAIT[:STATe] .....	80
4.11 SOURce Subsystem.....	80
4.11.1 [:SOURce]:<CURRent VOLTage>:<CENTer SPAN>.....	80
4.11.2 [:SOURce]:<CURRent VOLTage>[:LEVel][:IMMediate][:AMPLitude] .....	81
4.11.3 [:SOURce]:<CURRent VOLTage>[:LEVel]:TRIGgered[:AMPLitude] .....	81
4.11.4 [:SOURce]:<CURRent VOLTage>:MODE .....	82
4.11.5 [:SOURce]:<CURRent VOLTage>:POINts.....	82
4.11.6 [:SOURce]:<CURRent VOLTage>:RANGE .....	83
4.11.7 [:SOURce]:<CURRent VOLTage>:RANGE:AUTO .....	83
4.11.8 [:SOURce]:<CURRent VOLTage>:RANGE:AUTO:LLIMit.....	84
4.11.9 [:SOURce]:<CURRent VOLTage>:RANGE:RPRiority.....	84
4.11.10 [:SOURce]:<CURRent VOLTage>:<START STOP> .....	84
4.11.11 [:SOURce]:<CURRent VOLTage>:STEP .....	85
4.11.12 [:SOURce]:<CURRent VOLTage>:TRANsient:SPEEd.....	86

4.11.13 [:SOURce]:DIGital:DATA.....	86
4.11.14 [:SOURce]:DIGital:EXternal:FUNCTION.....	86
4.11.15 [:SOURce]:DIGital:EXternal:POLarity.....	87
4.11.16 [:SOURce]:DIGital:EXternal:TOUTput[:EDGE]:POSition .....	87
4.11.17 [:SOURce]:DIGital:EXternal:TOUTput[:EDGE]:WIDTh .....	88
4.11.18 [:SOURce]:DIGital:EXternal:TOUTput:TYPE .....	88
4.11.19 [:SOURce]:DIGital:INternal:TOUTput[:EDGE]:Position .....	88
4.11.20 [:SOURce]:FUNCTION:MODE.....	89
4.11.21 [:SOURce]:FUNCTION[:SHAPe] .....	89
4.11.22 [:SOURce]:FUNCTION:TRIGgered:CONTinuous.....	90
4.11.23 [:SOURce]:LIST:<CURRent VOLTage>.....	90
4.11.24 [:SOURce]:LIST:<CURRent VOLTage>:APPend .....	91
4.11.25 [:SOURce]:LIST:<CURRent VOLTage>:POINts? .....	91
4.11.26 [:SOURce]:LIST:<CURRent VOLTage>:STARt.....	92
4.11.27 [:SOURce]:PULSe:DELay .....	92
4.11.28 [:SOURce]:PULSe:WIDTh .....	92
4.11.29 [:SOURce]:SWEep:DIRection .....	93
4.11.30 [:SOURce]:SWEep:POINts .....	93
4.11.31 [:SOURce]:SWEep:RANGing.....	94
4.11.32 [:SOURce]:SWEep:SPACing .....	94
4.11.33 [:SOURce]:SWEep:STAir .....	95
4.11.34 [:SOURce]:TOUTput:SIGNAl.....	95
4.11.35 [:SOURce]:TOUTput[:STATe] .....	96
4.11.36 [:SOURce]:WAIT:AUTO.....	96
4.11.37 [:SOURce]:WAIT:GAIN .....	96
4.11.38 [:SOURce]:WAIT:OFFSet.....	97
4.11.39 [:SOURce]:WAIT[:STATe] .....	97
4.12 STATus Subsystem.....	98
4.12.1 :STATus:<MEASurement OPERation QUESTIONable>:CONDITION? .....	98
4.12.2 :STATus:<MEASurement OPERation QUESTIONable>:ENABLE.....	100
4.12.3 :STATus:<MEASurement OPERation QUESTIONable>[:EVENT]? .....	101
4.12.4 :STATus:<MEASurement OPERation QUESTIONable>:NTRansition .....	101
4.12.5 :STATus:<MEASurement OPERation QUESTIONable>:PTRansition .....	102
4.12.6 :STATus:PRESet.....	102

4.12.7 :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:CONDition?	102
4.12.8 :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:ENABLE	103
4.12.9 :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>[:EVENT]?	104
4.12.10 :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:NTRansition	104
4.12.11 :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:PTRansition	105
4.13 SYSTem Subsystem	106
4.13.1 :SYSTem:BEEPer[:IMMediate]	106
4.13.2 :SYSTem:BEEPer:STATe	106
4.13.3 :SYSTem:COMMUnicatE:GPIB[:SELF]:ADDResS	106
4.13.4 :SYSTem:COMMUnicatE:LAN:ADDResS	107
4.13.5 :SYSTem:COMMUnicatE:LAN:DHCP	107
4.13.6 :SYSTem:COMMUnicatE:LAN:DNS	107
4.13.7 :SYSTem:COMMUnicatE:LAN:<GATE GATeway>	108
4.13.8 :SYSTem:COMMUnicatE:LAN:<HNAMe HOSTname>	108
4.13.9 :SYSTem:COMMUnicatE:LAN:MAC?	109
4.13.10 :SYSTem:COMMUnicatE:LAN:SMASK	109
4.13.11 :SYSTem:DATA:QUANTITY?	109
4.13.12 :SYSTem:DATE	110
4.13.13 :SYSTem:ERROr:ALL?	110
4.13.14 :SYSTem:ERROr:CODE:ALL?	111
4.13.15 :SYSTem:ERROr:CODE[:NEXT]?	111
4.13.16 :SYSTem:ERROr:COUNt?	111
4.13.17 :SYSTem:ERROr[:NEXT]?	111
4.13.18 :SYSTem:FAN:MODE	112
4.13.19 :SYSTem:GROUp[:DEFInE]	112
4.13.20 :SYSTem:GROUp:RESet	113
4.13.21 :SYSTem:INTerlock:TRIPped?	113
4.13.22 :SYSTem:LANGUage	113
4.13.23 :SYSTem:LFRequency	114
4.13.24 :SYSTem:PON	114

4.13.25 :SYSTem:PRESet.....	114
4.13.26 :SYSTem:TIME .....	115
4.13.27 :SYSTem:TIME:TIMer:COUNT? .....	115
4.13.28 :SYSTem:TIME:TIMer:COUNT:RESet:AUTO .....	115
4.13.29 :SYSTem:TIME:TIMer:COUNT:RESet[:IMMEDIATE] .....	116
4.13.30 :SYSTem:VERSion? .....	116
4.14 TRACe Subsystem .....	116
4.14.1 :TRACe:CLEar.....	116
4.14.2 :TRACe:DATA? .....	116
4.14.3 :TRACe:FEED.....	117
4.14.4 :TRACe:FEED:CONTrol .....	118
4.14.5 :TRACe:FREE? .....	118
4.14.6 :TRACe:POINts .....	118
4.14.7 :TRACe:POINts:ACTual?.....	119
4.14.8 :TRACe:STATistic:DATA?.....	119
4.14.9 :TRACe:STATistic:FORMAT.....	119
4.14.10 :TRACe:TStamp:FORMAT.....	120
4.15 TRIGger Subsystem .....	120
4.15.1 :ABORT<:ACQirel:TRANSient[:ALL]> .....	120
4.15.2 :ARM<:ACQirel:TRANSient[:ALL]>[:IMMEDIATE] .....	121
4.15.3 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:BYPass .....	121
4.15.4 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:COUNT .....	122
4.15.5 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:DELAY .....	122
4.15.6 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:SOURce:LAN .....	123
4.15.7 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:SOURce[:SIGNAl] .....	123
4.15.8 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:TIMer .....	124
4.15.9 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:TOUTput:SIGNAL.....	124
4.15.10 :ARM<:ACQirel:TRANSient[:ALL]>[:LAYer]:TOUTput:STATE.....	125
4.15.11 :IDLE<:ACQirel:TRANSient[:ALL]>?.....	125
4.15.12 :INITiate[:IMMEDIATE]<:ACQirel:TRANSient[:ALL]> .....	126
4.15.13 :TRIGger<:ACQirel:TRANSient[:ALL]>:BYPass .....	126
4.15.14 :TRIGger<:ACQirel:TRANSient[:ALL]>:COUNT .....	127
4.15.15 :TRIGger<:ACQirel:TRANSient[:ALL]>:DELAY .....	127
4.15.16 :TRIGger<:ACQirel:TRANSient[:ALL]>[:IMMEDIATE] .....	128

---

4.15.17 :TRIGger<:ACQuire[:TRANsient[:ALL]]>:SOURce:LAN.....	128
4.15.18 :TRIGger<:ACQuire[:TRANsient[:ALL]]>:SOURce[:SIGNAl] .....	128
4.15.19 :TRIGger<:ACQuire[:TRANsient[:ALL]]>:TImer .....	129
4.15.20 :TRIGger<:ACQuire[:TRANsient[:ALL]]>:TOUTput:SIGNAl .....	130
4.15.21 :TRIGger<:ACQuire[:TRANsient[:ALL]]>:TOUTput[:STATe] .....	130
<b>5 使用现有程序 .....</b>	<b>132</b>
5.1 SMM3000X 支持的常规命令 .....	132
5.2 SMM3000X 部分支持的常规命令 .....	141
5.3 SMM3000X 不支持的常规命令 .....	144

# 1 编程概述

用户可以通过使用源表的 USB、LAN 或 GPIB 端口，并结合 NI-VISA 和程序语言，远程控制源表。基于 LAN 端口，SMM 支持 VXI-11、Sockets 和 Telnet 通信协议。本节介绍了如何建立 SMM 系列数字源表和计算机之间的通信，同时介绍如何远程控制源表。

## 1.1 建立通信

### 1.1.1 NI-VISA 的安装

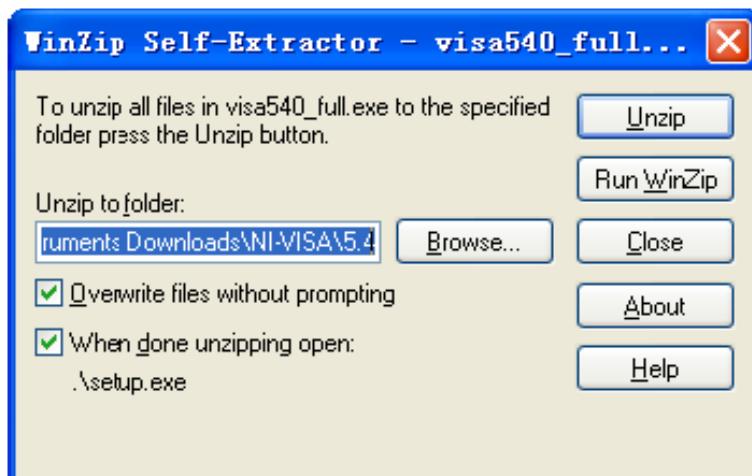
在编程之前，请确保正确安装 NI-VISA 软件的最新版本。

NI-VISA 是用于计算机与设备之间通信的通信库。NI 软件有两种有效 VISA 安装包：完整版和运行引擎版（Run-Time Engine）。完整版包括 NI 设备驱动和 NI MAX 工具，其中 NI MAX 是用于控制设备的用户界面。虽然驱动和 NI MAX 都很有用，但是它们不用于远程控制。运行引擎版（Run-Time Engine）是一个比完整版更小的文件，它主要用于远程控制。

你可以在 NI 官网上下载最新的 NI-VISA 运行引擎或完整版。它们的安装步骤基本相同。

按照下列步骤安装 NI-VISA（示例使用 NI-VISA5.4 完整版）：

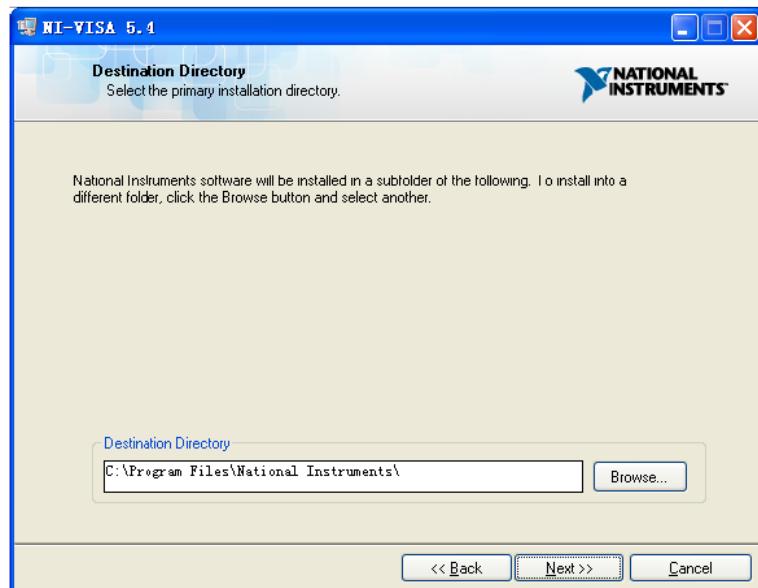
- 下载合适版本的 NI-VISA（推荐运行引擎版）
- 双击 visa540\_full.exe，弹出对话框如下：



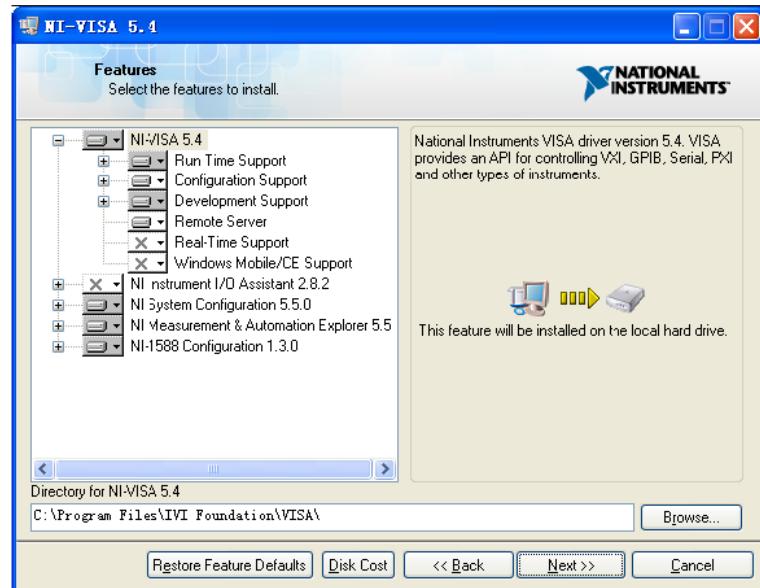
- 点击 Unzip 解压文件，当解压完成后，安装程序将自动执行。若你的计算机需要安装.NET Framework4，则在安装过程会自动安装.NET Framework4。



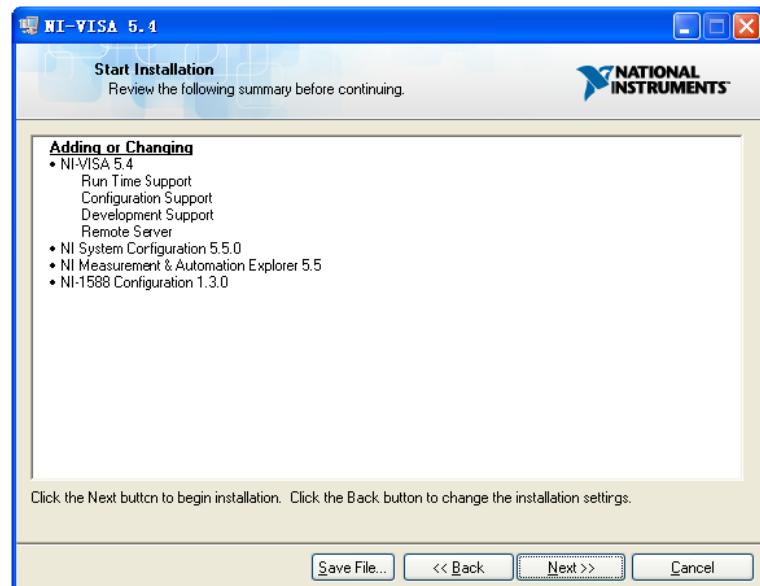
- d. NI-VISA 安装对话框如上图所示，点击 Next 开始安装过程。



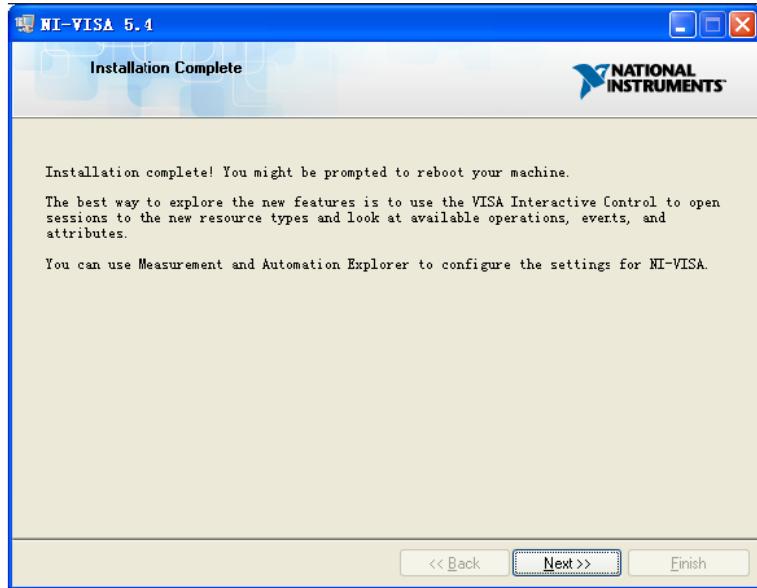
- e. 设置安装路径，默认路径为“C:\Program Files\National Instruments\”。你也可以修改安装路径。点击 Next，对话框如下图所示。



- f. 点击 Next 两次，在许可协议对话框下，选择 “I accept the above 2 License Agreement (s.)” 并点击 Next，对话框如下图显示：



- g. 点击 Next 开始安装：

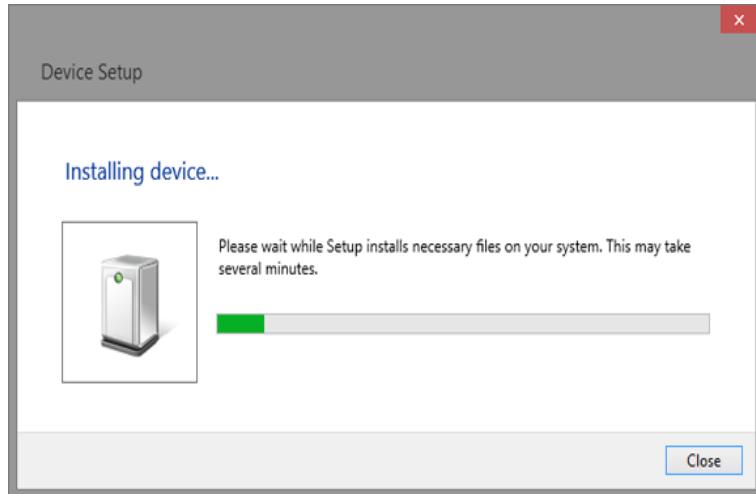


h. 安装完成后，重启电脑。

### 1.1.2 连接仪器

根据具体源表机型，源表能够通过 USB、LAN 端口或 GPIB（选配）接口连接计算机。

使用 USB 线将源表的 USB Device 端口和计算机的 USBHost 端口连接起来。假设你的计算机已经启动，打开源表后，桌面将弹出“设备安装”界面，并自动安装设备驱动，如下图所示：



等待安装完成，然后进行下一步。

## 1.2 远程控制的实现

### 1.2.1 用户自定义程序

用户可通过计算机发送 SCPI 命令实现编程和控制源表。相关内容，请查阅“编程示例”中的介绍。

### 1.2.2 通过 NI-MAX 发送 SCPI 命令

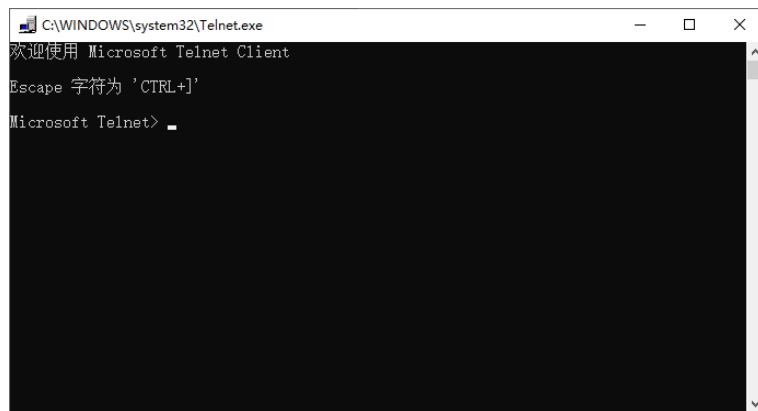
NI-MAX 是由 NI 公司创建和维护的程序。它为 VXI、LAN、USB、GPIB 和串行通信提供基础的远程控制接口。用户可以通过 NI-MAX 发送 SCPI 命令远程控制源表。

### 1.2.3 通过 Telnet 发送 SCPI 命令

Telnet 提供一种通过 LAN 端口与源表通信的方式。Telnet 协议支持从计算机向源表发送 SCPI 命令，该方式类似于通过 USB 与源表通信。发送和接受信息是交互的：一次只能发送一个命令。Windows 操作系统使用命令提示符样式接口作为 Telnet 客户端。

步骤如下：

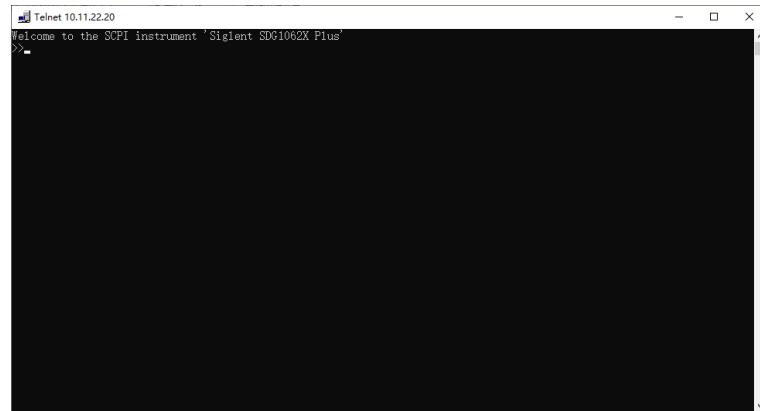
- 1 在计算机桌面，点击开始>所有程序>附件>命令提示符；
- 2 在命令提示符窗口，输入 Telnet；
- 3 按下 Enter 键。将弹出 Telnet 显示窗；



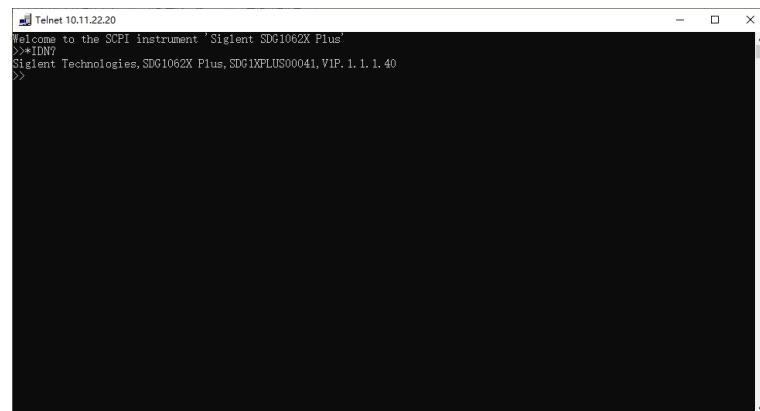
- 4 在 Telnet 命令行，键入：

```
open XXX.XXX.XXX.XXX 5024
```

其中 XXX.XXX.XXX.XXX 是指设备的 IP 地址；5024 是端口。通信成功后你会看到和下面类似的响应内容：



- 5 在“>>”提示符后，输入 SCPI 命令例如 \*IDN?。该命令将返回公司名、机器型号、序列号和固件版本号。



- 6 同时按下 **Ctrl+]** 键将退出 SCPI 会话。  
7 通过在提示符处键入 *quit* 或关闭 Telnet 窗口来关闭设备的连接并退出 Telnet。

#### 1.2.4 通过 Socket 发送 SCPI

Socket 接口可以在不安装其他库的情况下通过 LAN 端口控制 SMM 系列产品。它可以减少编程的复杂度。

<b>SOCKET 地址</b>	IP 地址+端口号
<b>IP 地址</b>	SMM IP 地址
<b>端口号</b>	5025

## 2 SCPI 语言简介

### 2.1 有关命令和查询

本节列出并描述仪器识别的远程控制命令和查询语句。所有命令和查询都可以在本地或远程状态下执行。

本文对每个包含有语法以及其他信息的命令和查询都列举出一些例子。在“命令语法”和“查询语法”中，本文给出了该命令的长格式和短格式。SCPI 命令头部可分为表示命令、查询或命令和查询。可根据 SCPI 命令头部后面跟着问号 (?) 来识别查询操作，例如获取信息。

### 2.2 描述

在描述中给出了执行功能的简要说明。接下来是命令的正式语法的表示，其中命令头部使用大小写字符和从大写字符派生出来的缩写形式。在适用的情况下，查询的语法以其响应的格式给出。

### 2.3 用法

本手册列出的命令和查询可以用在 S MM 3 0 0 0 X 系列的机器上。

### 2.4 命令符号

下面是用在命令中的符号：

<> 角括号包含用于占位的字符，其中分为两种类型：标题路径和命令参数。

:= 冒号后面跟着等号，它用于分隔占位符和占位符的描述，占位符的描述是指用于替换命令中占位符的参数类型和范围。

{ } 花括号列出了可供选择的参数，至少要选择一个参数。

[ ] 方括号包含可选项。

... 省略号表示其左侧和右侧可以重复多次。

### 2.5 数字后缀

本手册列出的部分 SCPI 指令中会带有数字后缀[c]、[i]、[j]、[m]或[n]，一般伴随着命令头出现，它们被用于区别通道、内部总线、节点、限值测试和 GPIO 引脚的参数设置，当忽略此数字后缀时，会自动设置为整数 1，例如：指令 SOURce:VOLTage 3 等价于指令 SOURce1:VOLTage 3，均为对通道 1 设

置电压 3V。各个数字后缀的定义具体如下：

- [c]: 用于表示选中的通道 1 或 2。对于源表的双通道型号，可选整数 1 或 2。
- [i]: 用于表示选中的内部总线 1 或 2。可选整数 1 或 2。
- [j]: 用于表示选中的节点 1 或 2。可选整数 1 或 2。
- [m]: 用于表示选中的限值测试的索引 1-12。可选整数 1-12。
- [n]: 用于表示选中的 GPIO 引脚号 1-12。可选整数 1-12。

## 2.6 数据输出格式

数据格式：

在返回测量、计算数据结果时，可以通过选择数据格式 (format) 返回指定格式的数据，参考指令 :FORMat [:DATA] 。

可设置的数据格式如下：

ASCII, IEEE 754 单精度浮点数, IEEE 754 双精度浮点数。

当返回无效数据时，ASCII 格式会返回值+9.910000E+37，浮点数格式会返回 NaN。

当返回数据为正无穷值时，ASCII 格式会返回值+9.900000E+37，浮点数格式会返回+infinity。

当返回数据为负无穷值时，ASCII 格式会返回值-9.900000E+37，浮点数格式会返回-infinity。

浮点数格式数据返回时可以指定字节顺序低到高（默认）或高到低，参考指令 :FORMat:BORDer。

状态数据 (Status)：

在返回测量、计算数据结果时，可以通过选择数据格式 (format) 返回状态数据 (Status)，参考指令 :FORMat:ELEMents:SENSe 和 :FORMat:ELEMents:CALCulate，状态数据返回一个十进制数，为二进制数的各比特位的加权和。

例如，当前源表设置为电流源，且处于过电压状态，则 Bit0 和 Bit3 置 1，返回 9。

下表列出各比特位定义的状态：

比特位	十进制值	描述	定义
0	1	源模式	0 表示电压源，1 表示电流源
1	2	合规性状态位 0	00 表示没有进入合规性状态，01、10、11 均表示进入合规性状态（达到 limit 值）
2	4	合规性状态位 1	均表示进入合规性状态（达到 limit 值）
3	8	过电压状态	1 表示处于过电压状态
4	16	过电流状态	1 表示处于过电流状态

5	32	过温状态	1 表示处于过温状态
6-12		未使用	返回任意
13	8192	测量超量程	1 表示测量超出量程范围
14	16384	偏置补偿	1 表示打开偏置补偿
15		未使用	返回任意
16-20	65536- 2031616	复合限值测试结果	NO.1-12 各项限值测试结果的具体比特位 定义见下表

Sorting 模式的复合限值测试的限值测试结果的比特位定义如下：

比特位 (高位 Bit20-低位 Bit16)	定义
0 0 0 0 1	NO.1 限值测试通过/合规性检查失败
0 0 0 1 0	NO.2 限值测试通过/合规性检查失败
0 0 0 1 1	NO.3 限值测试通过/合规性检查失败
0 0 1 0 0	NO.4 限值测试通过/合规性检查失败
0 0 1 0 1	NO.5 限值测试通过/合规性检查失败
0 0 1 1 0	NO.6 限值测试通过/合规性检查失败
0 0 1 1 1	NO.7 限值测试通过/合规性检查失败
0 1 0 0 0	NO.8 限值测试通过/合规性检查失败
0 1 0 0 1	NO.9 限值测试通过/合规性检查失败
0 1 0 1 0	NO.10 限值测试通过/合规性检查失败
0 1 0 1 1	NO.11 限值测试通过/合规性检查失败
0 1 1 0 0	NO.12 限值测试通过/合规性检查失败
1 1 1 1 1	所有限值测试失败且合规性检查通过

Grading 模式的复合限值测试的限值测试结果的比特位定义如下：

比特位 (高位 Bit20-低位 Bit16)	定义
0 0 0 0 0	所有限值测试通过且合规性检查通过
1 0 0 0 1	NO.1 限值测试因超出上限失败
0 0 0 0 1	NO.1 限值测试因低于下限失败/合规性检查失败
1 0 0 1 0	NO.2 限值测试因超出上限失败

0 0 0 1 0	NO.2 限值测试因低于下限失败/合规性检查失败
1 0 0 1 1	NO.3 限值测试因超出上限失败
0 0 0 1 1	NO.3 限值测试因低于下限失败/合规性检查失败
1 0 1 0 0	NO.4 限值测试因超出上限失败
0 0 1 0 0	NO.4 限值测试因低于下限失败/合规性检查失败
1 0 1 0 1	NO.5 限值测试因超出上限失败
0 0 1 0 1	NO.5 限值测试因低于下限失败/合规性检查失败
1 0 1 1 0	NO.6 限值测试因超出上限失败
0 0 1 1 0	NO.6 限值测试因低于下限失败/合规性检查失败
1 0 1 1 1	NO.7 限值测试因超出上限失败
0 0 1 1 1	NO.7 限值测试因低于下限失败/合规性检查失败
1 1 0 0 0	NO.8 限值测试因超出上限失败
0 1 0 0 0	NO.8 限值测试因低于下限失败/合规性检查失败
1 1 0 0 1	NO.9 限值测试因超出上限失败
0 1 0 0 1	NO.9 限值测试因低于下限失败/合规性检查失败
1 1 0 1 0	NO.10 限值测试因超出上限失败
0 1 0 1 0	NO.10 限值测试因低于下限失败/合规性检查失败
1 1 0 1 1	NO.11 限值测试因超出上限失败
0 1 0 1 1	NO.11 限值测试因低于下限失败/合规性检查失败
1 1 1 0 0	NO.12 限值测试因超出上限失败
0 1 1 0 0	NO.12 限值测试因低于下限失败/合规性检查失败

### 3 公共命令

#### 3.1 \*IDN?

描述	获取设备信息字符串（返回内容包括：厂商，设备型号，设备序列号，软件版本号）
命令	*IDN?
参数	无
查询响应	<p>Siglent Technologies,model,serial,revision &lt;newline&gt;</p> <p><i>model</i>: 型号；</p> <p><i>serial</i>: 序列号；</p> <p><i>revision</i>: 固件版本号。</p> <p>响应数据格式为 AARD。</p>
示例	*IDN?

#### 3.2 \*CAL?

描述	此查询命令执行自校准，并返回执行结果。 在开始自校准之前，断开测量端子。
命令	*CAL?
参数	无
查询响应	<p><i>result</i>&lt;newline&gt;&lt;^END&gt;</p> <p><i>result</i> 为 0 或 1，表示自校准结果。</p> <p>0: Passed; 1: Failed。</p> <p>响应参数类型为 NR1。</p>
示例	*CAL?

#### 3.3 \*CLS

描述	将所有事件寄存器的值清零，同时清空错误列表
命令	*CLS
参数	无
查询响应	无
示例	*CLS

### 3.4 \*ESE

<b>描述</b>	设置标准事件状态寄存器的使能值
<b>命令</b>	*ESE enable_number *ESE?
<b>参数</b>	enable_number 十进制值，是所需位、十六进制、八进制或二进制值的二进制加权值之和。参数数据类型为 NR1 或 NDN。
<b>查询响应</b>	enable_number<newline> enable_number 是 enable 寄存器位的二进制加权值之和。可以通过：format:SREGister 命令选择返回格式。响应数据类型为 NR1 或 NDN。
<b>示例</b>	*ESE 16 *ESE?

### 3.5 \*ESR?

<b>描述</b>	询问及清除标准事件状态寄存器的事件值
<b>命令</b>	*ESR?
<b>参数</b>	无
<b>查询响应</b>	register<newline> register 是寄存器中设置的所有位的二进制加权和。例如，如果启用位 3 (十进制值=8) 和位 7 (十进制值=128)，查询命令将返回 136。 响应数据类型为 NR1。
<b>示例</b>	*ESR?

### 3.6 \*OPC

<b>描述</b>	所有操作结束后，标准事件状态寄存器中 OPC 位 (bit0) 置 1
<b>命令</b>	*OPC *OPC?
<b>参数</b>	无
<b>查询响应</b>	1<newline> 如果仪器已完成此命令之前发送的所有待处理操作，则查询返回 1。 响应数据类型为 NR1。
<b>示例</b>	*OPC *OPC?

### 3.7 \*RCL

<b>描述</b>	此命令将仪器恢复到之前使用 *SAV 命令存储在存储位置 0 至 9 之一的状态。
<b>命令</b>	*RCL memory
<b>参数</b>	<i>memory</i> 存储器位置 0 至 9。 参数类型为 NR1.
<b>查询响应</b>	无
<b>示例</b>	*RCL 0

### 3.8 \*RST

<b>描述</b>	复位仪器
<b>命令</b>	*RST
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	*RST

### 3.9 \*SAV

<b>描述</b>	将仪器的当前状态存储到非易失性存储器中的指定位置。
<b>命令</b>	*SAV <i>memory</i>
<b>参数</b>	<i>memory</i> 存储器位置 0 至 9。 参数类型为 NR1.
<b>查询响应</b>	无
<b>示例</b>	*SAV 1

### 3.10 \*SRE

<b>描述</b>	设置状态字节寄存器的使能值
<b>命令</b>	*SRE <i>value</i> *SRE?
<b>参数</b>	<i>value</i> 与寄存器中位的二进制加权和对应的十进制值。 数据类型为 NRf.

---

*value<newline>**value* 是寄存器中设置的所有位的二进制加权和。

**查询响应** 例如，如果启用位 3 (十进制值=8) 和位 7 (十进制值=128)，查询命令将返回 136。  
响应数据类型为 NR1。

---

**示例** \*SRE 24  
\*SRE?

---

### 3.11 \*STB?

<b>描述</b>	查询状态字节寄存器的事件值
<b>命令</b>	*STB?
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	*STB?

---

### 3.12 \*TRG

<b>描述</b>	当触发子系统选择总线作为其源时，此通用命令会生成一个触发器。 该命令与组执行触发器 (GET) 命令具有相同的效果。
<b>命令</b>	*TRG
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	*TRG

---

### 3.13 \*TST?

<b>描述</b>	查询仪器自检结果
<b>命令</b>	*TST?
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	*TST?

---

### 3.14 \*WAI

<b>描述</b>	等待所有未完成操作完成之后，再执行任何其他命令
<b>命令</b>	*WAI
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	*WAI

---

## 4 Subsystem Commands 子系统命令

### 4.1 CALCulate Subsystem 计算子系统

#### 4.1.1 :CALCulate:CLIMits:CLEar:AUTO

描述	启用或禁用复合限值测试的自动清除功能。
命令	:CALCulate[c]:CLIMits:CLEar:AUTO <i>mode</i> :CALCulate[c]:CLIMits:CLEar:AUTO?
参数	<i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean。 <i>mode</i> =1 or ON 启用自动清除功能，每次执行 :INITiate 命令将会自动清除复合极限测试结果和端口(GPIO)。 参见 126 页的 “:INITiate[:IMMediate]<:ACQuire:TRANsient[:ALL]>”。 <i>mode</i> =0 or OFF 禁用自动清除功能。在启动下一个复合极限测试之前，必须手动清除复合极限测试结果和 GPIO 端口。执行 :CALCulate:CLIMits:CLEar[:IMMediate] 指令立即清除。
查询响应	<i>mode</i> <newline> <i>mode</i> 为 0 或 1，分别表示自动清除功能已关闭或开启。 响应数据类型为 NR1。
示例	:CALC:CLIM:CLE:AUTO 1 :CALC2:CLIM:CLE:AUTO?

#### 4.1.2 :CALCulate:CLIMits:CLEar:AUTO:DELay

描述	设置自动清除复合极限测试的延迟时间。
命令	:CALCulate[c]:CLIMits:CLEar:AUTO:DELay <i>time</i> :CALCulate[c]:CLIMits:CLEar:AUTO:DELay? [ <i>time</i> ]
参数	<i>time</i> value (+1E-5 to 60 seconds) MINimum MAXimum DEFault (默认值为+1E-4)。参数类型为 NRf+。查询不支持 time=value 。 如果指定的值小于 MIN 或大于 MAX，time 被自动设置为 MIN 或 MAX。
查询响应	<i>time</i> <newline> <i>time</i> 返回自动清除的当前延迟时间设置。 如果指定了一个参数，time 将返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR3。
示例	:CALC:CLIM:CLE:AUTO:DEL 1E-3 :CALC2:CLIM:CLE:AUTO:DEL?

#### 4.1.3 :CALCulate:CLIMits:CLEar[:IMMEDIATE]

<b>描述</b>	立即清除复合极限测试结果和 GPIO 端口。
<b>命令</b>	:CALCulate[:c]:CLIMits:CLEar[:IMMEDIATE]
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	:CALC:CLIM:CLE:IMM :CALC2:CLIM:CLE

#### 4.1.4 :CALCulate:CLIMits:<FAIL|PASS>:DIGital[:DATA]

<b>描述</b>	定义用于指示复合极限测试结果 (fail/pass) 的位 pattern。 它必须以 :FORMat:DIGital 命令设置的格式输入。
<b>命令</b>	:CALCulate[:c]:CLIMits:<FAIL PASS>:DIGital[:DATA] <i>bit_pattern</i> :CALCulate[:c]:CLIMits:<FAIL PASS>:DIGital[:DATA]?
<b>参数</b>	<i>bit_pattern</i> 0 (默认设置为十进制) 到 63。参数类型为 NR1 或 NDN。 <i>bit_pattern</i> <newline>
<b>查询响应</b>	<i>bit_pattern</i> 以 :FORMat:DIGital 命令指定的格式返回 fail/pass 位模式。 响应数据类型为 NR1 或 NDN 。
<b>示例</b>	:CALC:CLIM:FAIL:DIG:DATA 63 :CALC2:CLIM:PASS:DIG?

#### 4.1.5 :CALCulate:CLIMits:MODE

<b>描述</b>	将复合极限测试的操作模式设置为分级 GRADING 或排序 SORTing 。
<b>命令</b>	:CALCulate[:c]:CLIMits:MODE <i>mode</i> :CALCulate[:c]:CLIMits:MODE?
<b>参数</b>	<i>mode</i> SORT (sorting) GRAD (grading, default)。参数类型为 CPD。  <i>mode</i> =GRAD 执行多达 12 个测试限制测试，直到检测到失败 Fail。有关分级模式下的流程图的示例，请参见用户手册。
<b>参数</b>	<i>mode</i> =SORT 执行最多 12 个测试限制测试，直到检测到通过 Pass。有关排序模式下的流程图的示例，请参见用户手册。 限值测试是指在复合限值测试期间执行的通过/失败的判断。在“单项限值测试/合规性检查”中被执行。

---

	<i>mode</i> <newline>
查询响应	<i>mode</i> 返回 GRAD 或 SORT。 响应数据类型为 CPD.
示例	:CALC:CLIM:MODE SORT :CALC2:CLIM:MODE?

---

#### 4.1.6 :CALCulate:CLIMits:STATe

---

描述	启用或禁用复合极限测试.
命令	:CALCulate[c]:CLIMits:STATe <i>mode</i> :CALCulate[c]:CLIMits:STATe?
参数	<i>mode</i> 1 ON (default) 0 OFF。参数类型为 boolean。 <i>mode</i> =1 or ON 启用复合极限测试。 <i>mode</i> =0 or OFF 禁用复合极限测试。
查询响应	<i>mode</i> <newline> <i>mode</i> 为 0 或 1， 并分别表示复合极限测试已关闭或开启。 响应数据类型为 NR1 或 NDN.
示例	:CALC:CLIM:STAT 1 :CALC2:CLIM:STAT?

---

#### 4.1.7 :CALCulate:CLIMits:STATe:ANY?

---

描述	检查当前的复合限值测试中是否包含一个在“单项限值测试/合规性检查”被执行的判断 pass/fail 的限值测试。
命令	:CALCulate[c]:CLIMits:STATe:ANY?
参数	无
查询响应	<i>status</i> <newline> <i>status</i> 返回 0 或 1。响应数据类型为 NR1。 0: 不存在极限测试。 1: 至少存在一个极限测试。
示例	:CALC2:CLIM:STAT:ANY?

---

#### 4.1.8 :CALCulate:CLIMits:UPDate

---

描述	仅适用于 GRAD 复合极限测试。启用或禁用即时结果输出或更新。 请参见用户手册中“分级模式下的复合限值测试流程图”所示的“Update = IMMEDIATE?”步骤。
----	--

---

启用后，当检测到第一次失败或全部通过时，将立即执行结果输出/更新。  
结果是由 :CALCulate:CLIMits:<FAILIPASS>:DIGItal[:DATA] 命令定义的通过/失败 bit pattern。如果检测到所有的通过，将输出 pass pattern。

<b>命令</b>	:CALCulate[c]:CLIMits:UPD <i>ate result</i> :CALCulate[c]:CLIMits:UPD <i>ate?</i>
<b>参数</b>	<i>result</i> END IMMEDIATE (default)。参数类型为 CPD。 <i>result</i> = IMM 启用立即结果输出。 <i>result</i> = END 禁用即时结果输出。
<b>查询响应</b>	<i>result</i> <newline> <i>result</i> 返回 IMM 或 END。响应数据类型为 CRD。
<b>示例</b>	:CALC:CLIM:UPD END :CALC2:CLIM:UPD?

#### 4.1.9 :CALCulate:DATA?

<b>描述</b>	返回限值测试数据。 返回数据的元素由 :FORMAT:ELEMents:CALCulate 命令指定。 极限试验数据可用以下公式表示： <i>limit test data = input data - null offset</i> <i>input data</i> : 由:CALCulate:FEED 指定的数据 <i>null offset</i> : 由指令 :CALCulate:OFFSet or :CALCulate:OFFSet:ACQuire 设置的数据 如果:CALCulate:OFFSet:STATe 命令禁用 null offset 功能，null offset = 0.
<b>命令</b>	:CALCulate[c]:DATA? [ <i>offset[, size]</i> ]

<b>参数</b>	<i>offset</i> 表示接收数据的起始位置。n CURRENT STARt (default). 参数类型为 NR1 或 CPD。 <i>offset</i> =n 表示从第 n+1 个数据开始接收。n 是一个整数，0 至最大值 (取决于缓冲区的状态)。 <i>offset</i> =CURRE 表示以当前的数据位置为起始位置。 <i>offset</i> =STAR 表示从数据缓冲区的顶部开始。等同于 offset = 0。  <i>size</i> 要接收的数据数。1 到最大值(取决于缓冲区的状态)。 参数类型为 NR1。 如果未指定此参数，则会返回从 <i>offset</i> 开始的所有数据。
<b>查询响应</b>	<i>data</i> <newline> 响应数据类型为 NR3。 参见“数据输出格式”。
<b>示例</b>	:CALC2:DATA? 0,10

#### 4.1.10 :CALCulate:DATA:LATest?

返回最新的极限测试数据。

返回数据的元素由 :FORMat:ELEMents:CALCulate 命令指定。

限值测试数据可用以下公式表示：

**描述**

*limit test data = input data - null offset*

*input data*: 由:CALCulate:FEED 指定的数据;

*null offset*: 由指令 :CALCulate:OFFSet or :CALCulate:OFFSet:ACQuire 设置的数据;

如果:CALCulate:OFFSet:STATe 命令禁用 *null offset* 功能, *null offset*= 0.

**命令**

:CALCulate[c]:DATA:LATest?

**参数**

无

*data <newline>*

**查询响应**

响应数据类型为 NR3。

参见“数据输出格式”。

**示例**

:CALC2:DATA:LAT?

#### 4.1.11 :CALCulate:DIGital:BIT

分配用于复合限值测试结果输出的 GPIO 引脚。

**描述**

结果是由 :CALCulate:CLIMits:<FAIL|PASS>:DIGital[:DATA] 命令定义的通过/失败位模式 bit pattern。

**命令**

:CALCulate[c]:DIGital:*pin*

:CALCulate[c]:DIGital:BIT?

*pin* EXTn|NONE (default)。参数类型为 CPD。

EXTn 表示后面板上的 Digital I/O D-sub 连接器的输出端口的一个 GPIO 引脚, n=1 到 6。

**参数**

*pin*=NONE 不分配 GPIO 引脚。

要分配 GPIO 引脚, *pin* 必须是逗号分隔的序号从小到大且连续的 EXT 字符串, 如 EXTn,EXTn+1,... 和 EXTn。例如, *pin*=EXT2,EXT3,EXT4 对此命令有效, EXT2、EXT3 和 EXT4 分别被分配给 BIT0 (LSB)、BIT1 和 BIT2。相反, 非连续的引脚分配, 如引脚=EXT2、EXT3、EXT6, 是无效的。

*pin <newline>*

**查询响应**

*pin* 返回 NONE 或以逗号分隔的 EXT 字符串。

响应数据类型为 CRD。

**示例**

:CALC:DIG:BIT EXT2,EXT3,EXT4

:CALC2:DIG:BIT?

#### 4.1.12 :CALCulate:DIGital:<BUSY|EOT|SOT>

描述	为复合极限测试的 BUSY (忙)、EOT (测试结束) 或 SOT (测试开始) 信号线分配 GPIO 引脚。
命令	:CALCulate[c]:DIGital:<BUSY EOT SOT> <i>pin</i> :CALCulate[c]:DIGital:<BUSY EOT SOT>? <BUSY EOT SOT>: BUSY 表示 busy 忙信号, EOT 表示测试结束信号, SOT 表示测试开始信号。
参数	<i>pin</i> EXTn NONE (default). 参数类型为 CPD. EXTn 表示后面板上的 Digital I/O D-sub 连接器的输出端口的一个 GPIO 引脚, n=1 至 6 可选为 BUSY 和 EOT, n=7 至 12 可选为 SOT。 要分配 GPIO 引脚, <i>pin</i> 脚必须是像 EXTn一样的 EXT 字符串。例如, <i>pin</i> = EXT2。 <i>pin</i> =NONE 不分配 GPIO 引脚。
查询响应	<i>pin</i> <newline> <i>pin</i> 返回 NONE 或 EXT 字符串。响应数据类型为 CRD。
示例	:CALC:DIG:EOT EXT2 :CALC2:DIG:SOT?

#### 4.1.13 :CALCulate:FEED

描述	指定用于计算限值测试数据的输入数据值。限值测试是在复合极限试验期间进行的通过/失败判断。限值测试在“单项限值测试/合规性检查”中执行。限值测试数据由 :CALCulate:DATA?或者:CALCulate:DATA:LATest?命令返回。
命令	:CALCulate[c]:FEED <i>type</i> :CALCulate[c]:FEED?
参数	<i>type</i> 数据类型。MATH RESistance CURRent VOLTage (default)。参数类型为 CPD。 <i>type</i> = VOLT 表示电压测量数据。 <i>type</i> = CURR 表示电流测量数据。 <i>type</i> = RES 表示电阻测量数据, 由以下公式给出: Resistance = Vmeas/Imeas 其中, Vmeas 是电压测量数据, Imeas 为电流测量数据。
	<i>type</i> = MATH 指定由数学表达式给出的数据。在执行 :CALC:FEED MATH 命令之前, 必须先指定数学表达式。 现有的数学表达式可以由:CALCulate:MATH[:EXPRESSION]:NAME 命令来指定。 一个新的数学表达式可以由 :CALCulate:MATH[:EXPRESSION]:NAME

---

和 :CALCulate:MATH[:EXPRession][:DEFine]命令定义。

---

查询响应	<i>type</i> <newline> <i>type</i> 返回当前设置的数据类型, MATH, RES, CURR 或 VOLT。 响应数据类型为 CRD。
示例	:CALC:FEED MATH :CALC2:FEED?

---

#### 4.1.14 :CALCulate:LIMit:COMPliance:DIGital[:DATA]

描述	定义用于指示第 <i>m</i> 次合规性测试 <i>failure</i> 状态的 bit pattern。 它必须以 :FORMAT:DIGital 命令设置的格式输入。
命令	:CALCulate[c]:LIMit[m]:COMPliance:DIGital[:DATA] <i>bit_pattern</i> :CALCulate[c]:LIMit[m]:COMPliance:DIGital[:DATA]?
参数	<i>bit_pattern</i> 0 (默认十进制) 到 63。参数类型为 NR1 或 NDN。
查询响应	<i>bit_pattern</i> <newline> <i>bit_pattern</i> 以 :FORMAT:DIGital 命令指定的格式返回失败位模式比他 pattern。 响应数据类型为 NR1 或 NDN。
示例	:CALC:LIM:COMP:DIG:DATA 63 :CALC2:LIM12:COMP:DIG?

---

#### 4.1.15 :CALCulate:LIMit:COMPliance:FAIL

描述	设置第 <i>m</i> 次合规性状态判断标准。
命令	:CALCulate[c]:LIMit[m]:COMPliance:FAIL <i>criteria</i> :CALCulate[c]:LIMit[m]:COMPliance:FAIL?
参数	<i>criteria</i> OUT IN (default)。参数类型为 CPD。 <i>criteria</i> = IN 如果通道进入合规性状态，则确定限值测试已失败。 <i>criteria</i> = OUT 如果通道脱离合规状态，则确定限值测试已失败。
查询响应	<i>criteria</i> <newline> <i>criteria</i> 返回 IN 或 OUT，表示当前设置的判断标准。 响应数据类型为 CRD。
示例	:CALC:LIM:COMP:FAIL OUT :CALC2:LIM12:COMP:FAIL?

---

#### 4.1.16 :CALCulate:LIMit:FAIL?

<b>描述</b>	返回由 <i>m</i> 指定的极限试验的结果。
<b>命令</b>	:CALCulate[c]:LIMit[m]:FAIL?
<b>参数</b>	无
	<i>result</i> <newline>
<b>查询响应</b>	<i>result</i> 返回 0 或 1。响应数据类型为 NR1。 0: Passed 1: Failed
<b>示例</b>	:CALC2:LIM12:FAIL?

#### 4.1.17 :CALCulate:LIMit:FUNCTION

<b>描述</b>	设置由 <i>m</i> 指定的限值测试的类型。
<b>命令</b>	:CALCulate[c]:LIMit[m]:FUNCTION type :CALCulate[c]:LIMit[m]:FUNCTION?
<b>参数</b>	<i>type</i> COMPliance LIMit (default)。参数类型为 CPD。 <i>type</i> =COMP 设置合规性状态检查，检查通道是否处于符合性状态。 <i>type</i> =LIM 设置极限测试，检查测量值是否在上限和下限之间。
<b>查询响应</b>	<i>type</i> <newline> <i>type</i> 返回该类型的当前设置，COMP 或 LIM。响应数据类型为 CRD。
<b>示例</b>	:CALC:LIM:FUNC COMP :CALC2:LIM12:FUNC?

#### 4.1.18 :CALCulate:LIMit:<LOWer|UPPer>

<b>描述</b>	设置由 <i>m</i> 指定的极限试验的上下限。
<b>命令</b>	:CALCulate[c]:LIMit[m]:<LOWer UPPer> / <i>limit</i> :CALCulate[c]:LIMit[m]:<LOWer UPPer>? [/ <i>limit</i> ] <LOWer UPPer>: LOWer 下限, UPPer 上限。
<b>参数</b>	<i>limit</i> value (-9.99999E+20 to +9.99999E+20) MINimum MAXimum  DEFault (下限的默认值为 -1, +1 为上限)。 参数类型为 NRf+。查询不支持 limit = value。
<b>查询响应</b>	/ <i>limit</i> <newline> <i>limit</i> 返回 <i>m</i> 指定的限值测试当前设置的下限/上限。 如果指定了一个参数， <i>limit</i> 返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR3。

---

示例	:CALC:LIM:LOW -2.5 :CALC2:LIM12:UPP?
----	---

---

#### 4.1.19 :CALCulate:LIMit:<LOWer|UPPer>:DIGital[:DATA]

描述	定义用于指示 m 所规定的极限试验的超过下限的失败或超过上限的失败的位 pattern。必须以 :FORMAT:DIGITAL 命令设置的格式输入。由此命令定义的位 pattern 用于 GRAD 复合限值测试。
命令	:CALCulate[c]:LIMit[m]:<LOWer UPPer>:DIGital[:DATA] <i>bit_pattern</i> :CALCulate[c]:LIMit[m]:<LOWer UPPer>:DIGital[:DATA]? <LOWer UPPer>: LOWER 超过下限的失败, UPPeR 超过上限的失败。
参数	<i>bit_pattern</i> 0 (默认十进制) 到 63。 参数类型为 NR1 或 NDN。
查询响应	<i>bit_pattern</i> <newline> <i>bit_pattern</i> 以:FORMAT:DIGITAL 命令指定的格式返回故障位 pattern。 响应数据类型为 NR1 或 NDN。
示例	:CALC:LIM:LOW:DIG:DATA 63 :CALC2:LIM12:UPP:DIG?

---

#### 4.1.20 :CALCulate:LIMit:PASS:DIGital[:DATA]

描述	定义用于指示通过由 m 指定的限值测试的位 pattern。 必须以 :FORMAT:DIGITAL 命令设置的格式输入。由此命令定义的位模式用于 SORT 复合限值测试。
命令	:CALCulate[c]:LIMit[m]:PASS:DIGital[:DATA] <i>bit_pattern</i> :CALCulate[c]:LIMit[m]:PASS:DIGital[:DATA]?
参数	<i>bit_pattern</i> 0 (默认十进制) 到 63。 参数类型为 NR1 或 NDN。
查询响应	<i>bit_pattern</i> <newline> <i>bit_pattern</i> 以:FORMAT:DIGITAL 命令指定的格式返回传递位模式。 响应数据类型为 NR1 或 NDN.
示例	:CALC:LIM:PASS:DIG:DATA 32 :CALC2:LIM12:PASS:DIG?

---

#### 4.1.21 CALCULATE:LIMit:STATe

---

描述	启用或禁用由 m 指定的限值测试.
----	-------------------

---

<b>命令</b>	:CALCulate[c]:LIMit[m]:STATe <i>mode</i> :CALCulate[c]:LIMit[m]:STATe?
<b>参数</b>	<i>mode</i> 1 ON 0 OFF (default)。 参数类型为 boolean。 <i>mode</i> =1 or ON 启用由 m 指定的限值测试。 <i>mode</i> =0 or OFF 禁用 m 规定的限值测试。
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 为 0 或 1，分别表示极限测试已关闭或开启。 响应数据类型为 NR1。
<b>示例</b>	:CALC:LIM:STAT 1 :CALC2:LIM12:STAT?

#### 4.1.22 :CALCulate:MATH:DATA?

<b>描述</b>	返回数学表达式计算结果数据。 返回数据的元素由 :FORMAT:ELEMents:CALCulate 命令指定。 计算的数学表达式由 :CALCulate:MATH[:EXPRession]:NAME 和 :CALCulate:MATH[:EXPRession][:DEFine]命令定义。
<b>命令</b>	:CALCulate[c]:MATH:DATA? [ <i>offset</i> , <i>size</i> ]
<b>参数</b>	<i>offset</i> 表示接收数据的起始位置。n CURR STARt (default). 参数类型为 NR1 或 CPD。 <i>offset</i> =n 表示从第 n+1 个数据开始接收。n 是一个整数，0 至最大值 (取决于缓冲区的状态)。 <i>offset</i> = CURR 表示以当前的数据位置为起始位置。 <i>offset</i> = STAR 表示从数据缓冲区的顶部开始。等同于 <i>offset</i> = 0.  <i>size</i> 要接收的数据数。1 到 maximum (取决于缓冲区的状态)。 参数类型为 NR1。 如果未指定此参数，则会返回从 offset 开始的所有数据。
<b>查询响应</b>	<i>data</i> <newline> 响应数据类型为 NR3。 参见“数据输出格式”。
<b>示例</b>	:CALC2:MATH:DATA? 0,10
<b>Remarks</b>	如果数学表达式包含多个测量结果，则需要多次测量以得到结果。 例如，必须执行两次测量，才能得到以下数学表达式的结果。 math expression = (CURR[1]-CURR[0])

#### 4.1.23 :CALCulate:MATH:DATA:LATest?

	返回最新的计算结果数据。
描述	返回数据的元素由 :FORMAT:ELEMENTS:CALCULATE 命令指定。 计算的数学表达式由 :CALCULATE:MATH[:EXPRESSION]:NAME 和 :CALCULATE:MATH[:EXPRESSION][:DEFINE]命令定义。
命令	:CALCULATE[c]:MATH:DATA:LATest?
参数	无
	<i>data</i> <newline>
查询响应	响应数据类型为 NR3。 参见“数据输出格式”。
示例	:CALC2:MATH:DATA:LAT?

#### 4.1.24 :CALCulate:MATH[:EXPRESSION]:CATalog?

	返回所有预定义和用户定义的数学表达式名称的列表。
描述	
命令	:CALCULATE[c]:MATH[:EXPRESSION]:CATalog?
参数	无
	<i>catalog</i> <newline>
	<i>catalog</i> 返回所有预定义的和用户定义的数学表达式名称。
查询响应	响应数据类型为 AARD。 例如, 如果该仪器存储了数学表达式 POWER, OFFCOMPOHM, VOLTCOEF, 和 VARALPHA, <i>catalog</i> 返回 “POWER”, “OFFCOMPOHM”, “VOLTCOEF”, “VARALPHA”。
示例	:CALC:MATH:EXPR:CAT?

#### 4.1.25 :CALCulate:MATH[:EXPRESSION][:DEFine]

	定义一个属于用户定义的数学表达式。 对于该表达式的有效资源, 请参见第 36 页中的“表达式中使用的资源”。 关于预定义表达式的定义, 也请参见第 37 页的“预定义数学表达式”。 在执行此命令之前, 必须通过 :CALCULATE:MATH[:EXPRESSION]:NAME 命令来选择或新建数学表达式名称。
命令	:CALCULATE[c]:MATH[:EXPRESSION][:DEFine] <i>definition</i> :CALCULATE[c]:MATH[:EXPRESSION][:DEFine]?
参数	<i>definition</i> 定义的数学表达式。最多有 256 个 ASCII 字符。 参数类型为 SPD。

该表达式必须用圆括号括起来。例如, `definition = (SOUR2/CURR2)`。  
最多可以定义 32 个数学表达式, 包括预定义的数学表达式。

查询响应	<code>definition &lt;newline&gt;</code> <code>definition</code> 返回当前选择的数学表达式的定义。 例如, <code>definition</code> 返回(SOUR2/CURR2)。 响应数据类型为 Expr。
------	--

示例	<code>:CALC:MATH:EXPR:NAME "Expression_for_ch1"</code> <code>:CALC:MATH:DEF "((CURR[1]-CURR[0])*(RES[1]-RES[0]))"</code> <code>:CALC:MATH?</code>
----	---

#### 4.1.25.1 表达式中使用的资源

以下资源可以用于用户定义的数学表达式中:

保留变量: 表 1 中列出的变量被保留用于读取通道输出或测量数据。

标量变量: 用于单点测量数据。

矢量 (数组) 变量: 用于 Sweep 扫描测量数据。

数学运算符: 可提供以下操作符:

算子: +, -, \*, /, ^, 见表 4-2。

基本函数: ln, log, sin, cos, tan, exp

数字型数值:

十进制 (0 ~ 4294967294, 4294967295 表示 -1);

二进制 (32 bit, 0 ~0b11111111111111111111111111111111);

十六进制 (0 ~0xFFFFFFFF)。

**注意:** log 和 ln 函数在计算绝对值后执行操作。因此, 如果指定了一个负值, 它们不会导致错误, 而是像指定了一个正值一样计算。例如, `log (-10)` 的结果同 `log (10) = 1`。

#### 保留变量

表 4-1

保留变量 <sup>1</sup>		描述
标量	向量	
SOUR[c]	SOUR[c][]	源输出设置数据

VOLT[c]	VOLT[c][0]	电压测量数据
CURR[c]	CURR[c][0]	电流测量数据
RES[c]	RES[c][0]	电阻测量数据
TIME[c]	TIME[c][0]	时间（时间戳）数据

### 算术运算符和一元运算符

表 4-2

优先级	运算符	描述
High	( )	圆括号
:	+ , -	一元加运算符和一元减运算符
:	^	取幂运算
:	* , /	乘法运算符和除法运算符
Low	+ , -	加法运算符和减法运算符

预定义数学表达式：以下数学表达式已经在仪器中定义了。预定义的数学表达式不会被断电/开机运算清除。

- 功率 (POWER)
- 电阻偏移补偿 (OFFCOMPOHM)
- 变阻器 (VARALPHA)
- 电压系数 (VOLTCOEF)

**POWER** 使用以下公式计算功率：

$$\text{POWER} = \text{VOLT}[\text{c}] * \text{CURR}[\text{c}]$$

**OFFCOMPOHM** 使用以下公式计算电阻偏移补偿（电阻）：

$$\text{OFFCOMPOHM} = (\text{VOLT}[\text{c}][1] - \text{VOLT}[\text{c}][0]) / (\text{CURR}[\text{c}][1] - \text{CURR}[\text{c}][0])$$

式中， $\text{VOLT}[\text{c}][0]$  和  $\text{CURR}[\text{c}][0]$  为用电流输出电平测量的数据， $\text{VOLT}[\text{c}][1]$  和  $\text{CURR}[\text{c}][1]$  为不同电  
流输出电平或零输出测量的数据。该功能可有效地降低低电阻测量中的测量误差。

**VARALPHA** 使用以下公式计算变阻器 Alpha：

$$\text{VARALPHA} = \log(\text{CURR}[\text{c}][1] / \text{CURR}[\text{c}][0]) / \log(\text{VOLT}[\text{c}][1] / \text{VOLT}[\text{c}][0])$$

其中，CURR[c][0]和 VOLT[c][0]是变阻器非线性 I-V 特性曲线上某一点的测量数据，CURR[c][1]和 VOLT[c][1]是另一个点的数据。

**VOLTCOE** 使用以下公式计算电压系数：

$$\text{VOLTCOEF} = (\text{RES}[c][1]-\text{RES}[c][0]) / (\text{RES}[c][1] * (\text{VOLT}[c][1]-\text{VOLT}[c][0])) * 100 \%$$

其中，RES[c][0] 和 RES[c][1] 分别是在第一和第二测量点上的电阻测量数据，VOLT[c][0] 和 VOLT[c][1] 分别为第一和第二测量点上的电压测量数据。电压系数称为电阻随电压变化的电阻的分数变化之比。

#### 4.1.26 :CALCulate:MATH[:EXPRESSION]:DElete:ALL

<b>描述</b>	删除所有用户定义的数学表达式。此命令不能删除预定义的数学表达式。
<b>命令</b>	:CALCulate[c]:MATH[:EXPRESSION]:DElete:ALL
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	:CALC2:MATH:EXPR:DEL:ALL

#### 4.1.27 :CALCulate:MATH[:EXPRESSION]:DElete[:SELected]

<b>描述</b>	删除一个用户定义的数学表达式。此命令不能删除预定义的数学表达式。
<b>命令</b>	:CALCulate[c]:MATH[:EXPRESSION]:DElete[:SELected] name
<b>参数</b>	无
<b>查询响应</b>	name 要删除的数学表达式的名称。最多有 32 个 ASCII 字符。 参数类型为 SPD。
<b>示例</b>	:CALC2:MATH:EXPR:DEL:SEL "TempExpression1"

#### 4.1.28 :CALCulate:MATH[:EXPRESSION]:NAME

<b>描述</b>	选择用于计算的数学表达式。预定义定义的数学表达式或用户定义的数学表达式可以由 name 名称参数指定。 有关预定义数学表达式的定义，请参见第 37 页上的“预定义数学表达式”。 通过使用新名称执行此命令，并使用新定义执行 :CALCulate:MATH[:EXPRESSION][:DEFine] 命令，可以添加新的用户定义数学表达式。 现有的用户定义的数学表达式可以通过执行带有名称的命令，并执行带有
-----------	---

---

	新定义的 :CALCulate:MATH[:EXPRESSION][:DEFine] 命令。
命令	:CALCulate[c]:MATH[:EXPRESSION]:NAME <i>name</i> :CALCulate[c]:MATH[:EXPRESSION]:NAME?
参数	<i>name</i> 一个数学表达式的名称。最多 32 个 ASCII 字符，没有任何控制字符、空格字符、单引号和双引号以及逗号。参数类型为 SPD。
查询响应	<i>name</i> <newline> <i>name</i> 返回当前选择的数学表达式的名称。 例如， <i>name</i> 返回“Expression_for_ch2”。响应数据类型为 SRD。
示例	:CALC2:MATH:EXPR:NAME “Expression_for_ch2” :CALC2:MATH:NAME?

---

#### 4.1.29 :CALCulate:MATH:STATE

---

描述	启用或禁用该数学表达式.
命令	:CALCulate[d]:MATH:STATe <i>mode</i> :CALCulate[d]:MATH:STATe?
参数	<i>mode</i> 1 ON 0 OFF (default)。参数类型为 boolean。 <i>mode</i> =1 or ON 启用数学表达式。 <i>mode</i> =0 or OFF 禁用数学表达式。
查询响应	<i>mode</i> <newline> <i>mode</i> 为 0 或 1，分别表示数学表达式已关闭或已打开。 响应数据类型为 NR1.
示例	:CALC:MATH:STAT 1 :CALC2:MATH:STAT?

---

#### 4.1.30 :CALCulate:MATH:UNITS

---

描述	定义数学表达式的单位名称.
命令	:CALCulate[c]:MATH:UNITS <i>unit</i> :CALCulate[c]:MATH:UNITS?
参数	<i>unit</i> 单位名称。最多有 32 个 ASCII 字符。参数类型为 SPD。
查询响应	<i>unit</i> <newline> <i>unit</i> 返回数学表达式的单位名称。 响应数据类型为 SRD.
示例	:CALC:MATH:UNIT “amps” :CALC2:MATH:UNIT?

---

#### 4.1.31 :CALCulate:OFFSet

<b>描述</b>	设置用于计算限值测试数据的 null offset 空偏移值。 空偏移功能由 :CALCulate:OFFSet:STATe 命令启用。
<b>命令</b>	:CALCulate[c]:OFFSet <i>offset</i> :CALCulate[c]:OFFSet? [ <i>offset</i> ]
<b>参数</b>	<i>offset</i> value (-9.999999E+20 到 +9.999999E+20) MINimum MAXimum DEFault (默认为 0.0)。 参数类型为 NRf+。查询不支持 offset=value。
<b>查询响应</b>	<i>offset</i> <newline> <i>offset</i> 返回当前设置的空偏移量值。 如果指定了一个参数，偏移量将返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR3。
<b>示例</b>	:CALC:OFFS 0.5 :CALC2:OFFS?

#### 4.1.32 :CALCulate:OFFSet:ACQuire

<b>描述</b>	自动设置用于计算限值测试数据的空偏移值。 空偏移量值将是由 :CALCulate:DATA:LATest?或 :SENSe:DATA:LATest? 命令读取的当前可用的值，如果当前可用的值不存在，则它将为 0.0。
<b>命令</b>	:CALCulate[c]:OFFSet:ACQuire
<b>参数</b>	无
<b>查询响应</b>	无
<b>示例</b>	:CALC:OFFS:ACQ :CALC2:OFFS:ACQ

#### 4.1.33 :CALCulate:OFFSet:STATe

<b>描述</b>	启用或禁用用于计算限值测试数据的空值偏移 null offset 功能。 空偏移值由 :CALCulate:OFFSet 或 :CALCulate:OFFSet:ACQuire 命令 设置。
<b>命令</b>	:CALCulate[c]:OFFSet:STATe <i>mode</i> :CALCulate[c]:OFFSet:STATe?
<b>参数</b>	<i>mode</i> 1 ON 0 OFF (default)。参数类型为 boolean。 <i>mode</i> =1 或 ON 启用空偏移。 <i>mode</i> =0 or OFF 禁用空偏移。
<b>查询响应</b>	<i>mode</i> <newline>

---

*mode* 为 0 或 1，分别表示空偏移量已关闭或开启。响应数据类型为 NR1。

---

**示例**

```
:CALC:OFFS:STAT 1
:CALC2:OFFS:STAT?
```

---

## 4.2 DISPlay Subsystem 显示子系统

### 4.2.1 :DISPlay:DIGits

<b>描述</b>	设置前面板上显示的测量读数的显示分辨率
<b>命令</b>	<pre>:DISPlay:DIGits <i>digits</i> :DISPlay:DIGits? [MINimum MAXimum DEFault]</pre>
<b>参数</b>	<p><i>digits</i> 分辨率。value (4 to 7) MINimum MAXimum DEFault (默认值为 7)。 参数类型为 NRf+。查询不支持 <i>digits</i>=value。</p> <p><i>digits</i>=4 选择 3½ 位分辨率。 <i>digits</i>=5 选择 4½ 位分辨率。 <i>digits</i>=6 选择 5½ 位分辨率。 <i>digits</i>=7 选择 6½ 位分辨率。</p>
<b>查询响应</b>	<p><i>digits</i>&lt;newline&gt;</p> <p><i>digits</i> 返回当前设置的分辨率。 如果指定了一个参数，<i>digits</i> 返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR1。</p>
<b>示例</b>	<pre>:DISP:DIG 4 :DISP:DIG? MAX</pre>

---

## 4.3 FETCh Subsystem

### 4.3.1 :FETCh:ARRay?

<b>描述</b>	返回测量数据，数据包含由 :FORMAT:ELEMents:SENSe 命令指定的数据类型，电压测量数据、电流测量数据、电阻测量数据、时间数据、状态数据或源输出设置数据。 测量数据直到执行 :INITiate、:MEASure 或 :READ 命令时才会被清除。
<b>命令</b>	<pre>:FETCh:ARRay? [<i>chanlist</i>]</pre>
<b>参数</b>	<p><i>chanlist</i> 指定返回数据的通道。 参数类型为 channel list. (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1)。</p> <p>(@1) 仅选择通道 1； (@2) 仅选择通道 2； (@1,2), (@1:2), (@2,1), 和 (@2:1) 同时选择通道 1 和通道 2。</p>

---

---

如果未指定此参数， 默认 chanlist = (@1)。

---

*response* <newline>

*response* 返回由 :FORMat:ELEMents:SENSe 命令指定的数组数据。

响应数据类型为 NR3。参见“数据输出格式”。

*response* 使用 ASCII 数据输出格式，每个数据都用逗号分隔，如果通道 1 和通道 2 都被 chanlist 选中，将按顺序返回通道 1 数据和通道 2 数据。示例如下：

*ch1curr1, ch1sour1, ch2curr1, ch2sour1,*

*ch1curr2, ch1sour2, ch2curr2, ch2sour2,.....*

*ch1curr5, ch1sour5, ch2curr5, ch2sour5,*

*ch1curr6, ch1sour6, +9.910000E+37, +9.910000E+37,.....*

*ch1curr10, ch1sour10, +9.910000E+37, +9.910000E+37*

该示例包含通道 1 的 10 步扫描测量的电流数据 (ch1currN) 和源数据

(ch1sourN)，以及通道 2 的 5 步扫描测量的电流数据 (ch2currN) 和源数据 (ch2sourN) 的数据。

如果未启用测量功能或不存在数据，*response* 将返回 +9.910000E+37 (ASCII) 或 NaN (IEEE-754)。

---

#### 查询响应

:FORM:ELEM:SENS CURR,SOUR

:FETC:ARR? (@1,2)

---

### 4.3.2 :FETCh:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

#### 描述

返回由 CURRent, RESistance, SOURce, STATus, TIME, 或 VOLTage 指定的电流测量数据、电阻测量数据、源输出设置值、状态数据、时间戳数据或电压测量数据。

测量数据直到执行 :INITiate、:MEASure 或 :READ 命令时才会被清除。

---

:FETCh:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

[*chanlist*]

CURRent 指定电流测量数据；

RESistance 指定电阻测量数据；

SOURce 源输出设置值；

STATus 状态数据；

TIME 时间戳数据；

VOLTage 电压测量数据。

---

#### 命令

*chanlist* 指定返回数据的通道。

参数类型为 channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1)。

(@1) 仅选择通道 1。

(@2) 仅选择通道 2。

(@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。

如果未指定此参数，*chanlist* = (@1) is set.

---

---

查询响应	<p><i>response</i>&lt;newline&gt;</p> <p><i>response</i> 返回由 CURRent, RESistance, SOURce, STATus, TIME, 或 VOLtage 指定的数组数据。响应数据类型为 NR3。参见“数据输出格式”。</p> <p><i>response</i> 使用 ASCII 数据输出格式，每个数据都用逗号分隔，如果通道 1 和通道 2 都被 chanlist 选中，将按顺序返回通道 1 数据和通道 2 数据。示例如下：</p> <pre>ch1curr1, ch2curr1, ch1curr2, ch2curr2, .... ch1curr5, ch2curr5, ch1curr6, +9.910000E+37, .... ch1curr10, +9.910000E+37</pre> <p>此示例包含通道 1 的 10 步扫描测量的电流数据 (ch1currN)，和通道 2 的 5 步扫描测量的电流数据 (ch2currN)。</p> <p>如果未启用测量功能或不存在数据，<i>response</i> 将返回 +9.910000E+37 (ASCII) 或 NaN (IEEE-754)。</p>
示例	:FETC:ARR:CURR? (@2,1)

---

#### 4.3.3 :FETCh[:SCALar]?

---

描述	获取由命令 :FORMat:ELEMents:SENSe 指定的最新电压测量数据、电流测量数据、电阻测量数据、时间数据、状态数据或源输出值；测量数据直到执行 :INITiate、:MEASure 或:READ 命令时才会被清除。
命令	:FETCh[:SCALar]? [ <i>chanlist</i> ]
参数	<p><i>chanlist</i> 指定返回数据的通道。参数类型为 channel list。</p> <p>(@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1).</p> <p>(@1) 仅选择通道 1。</p> <p>(@2) 仅选择通道 2。</p> <p>(@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。</p> <p>如果未指定此参数，<i>chanlist</i> = (@1)。</p>

---

查询响应	<p><i>response</i>&lt;newline&gt;</p> <p><i>response</i> 返回由 :FORMat:ELEMents:SENSe 命令指定的最新一次的测量数据。响应数据类型为 NR3。参见“数据输出格式”。</p> <p><i>response</i> 使用 ASCII 数据输出格式，每个数据都用逗号分隔，如果通道 1 和通道 2 都被 chanlist 选中，将按顺序返回通道 1 数据和通道 2 数据。示例如下：</p> <pre>ch1curr10, ch1sour10, ch2curr5, ch2sour5</pre> <p>此示例包含通道 1 的 10 步扫描测量的最新电流数据 (ch1curr10) 和源数据 (ch1sour10)，通道 2 的 5 步扫描测量的最新电流数据 (ch2curr5) 和源数据 (ch2sour5)。</p> <p>如果未启用测量功能或不存在数据，<i>response</i> 将返回 +9.910000E+37</p>
------	---

---

---

(ASCII)或 NaN (IEEE-754)。

---

**示例**

```
:FORM:ELEM:SENS CURR,SOUR
:FETC? (@1,2)
```

---

#### 4.3.4 :FETCh[:SCALar]:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

返回最新一次的电流测量数据、电阻测量数据、源输出数据、状态数据、时间数据或电压数据；

**描述** 数据类型由 CURRent, RESistance, SOURce, STATus, TIME, or VOLTage 参数指定。  
直到执行 :INITiate、:MEASure 或:READ 命令时才会清除数据。

---

:FETCh[:SCALar]:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?  
[chanlist]

CURRent 指定电流测量数据;  
RESistance 指定电阻测量数据;  
SOURce 源输出设置值;  
STATus 状态数据;  
TIME 时间数据;  
VOLTage 电压测量数据。

---

*chanlist* 指定返回数据的通道。参数类型为 channel list。

(@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1)。

**参数** (@1) 仅选择通道 1。  
(@2) 仅选择通道 2。  
(@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。  
如果未指定此参数, chanlist = (@1).

---

*response*<newline>

*response* 返回由 CURRent, RESistance, SOURce, STATus, TIME, or VOLTage 指定的最新一次的测量数据。响应数据类型为 NR3, 参见“数据输出格式”。

*response* 使用 ASCII 数据输出格式, 每个数据都用逗号分隔, 如果通道 1 和通道 2 都被 chanlist 选中, 将按顺序返回通道 1 数据和通道 2 数据。示例如下:

*ch1curr10,ch2curr5*

此示例包含通道 1 的 10 步扫描测量的最新电流数据 (ch1curr10) 和通道 2 的 5 步扫描测量的最新电流数据 (ch2curr5)。

如果未启用测量功能或不存在数据, *response* 将返回 +9.910000E+37 (ASCII)或 NaN (IEEE-754)。

---

**示例**

```
:FETC:CURR? (@2,1)
```

---

## 4.4 FORMat Subsystem

### 4.4.1 :FORMat:BORDer

<b>描述</b>	此命令将设置二进制输出数据的字节顺序，在使用 :FORMat[:DATA] 命令将数据输出格式设置为 IEEE-754 二进制格式时有效。
<b>命令</b>	:FORMat:BORDer <i>byte_order</i> :FORMat:BORDer?
<b>参数</b>	<i>byte_order</i> NORMal (default) SWAPped。参数数据类型为 CPD。 <i>byte_order</i> =NORMal 设置正常的字节顺序。对于 IEEE-754 单精度格式，发送顺序为从 byte1 到 byte4。对于 IEEE-754 双精度格式，发送顺序为从 byte1 到 byte8。 <i>byte_order</i> =SWAPped 设置反向字节的顺序。对于 IEEE-754 单精度格式，发送顺序为从 byte4 到 byte1。对于 IEEE-754 双精度格式，发送顺序为从 byte8 到 byte1。
<b>查询响应</b>	<i>byte_order</i> <newline> <i>byte_order</i> 返回 NORM 或 SWAP。 响应数据类型为 CRD。
<b>示例</b>	:FORM:BORD SWAP :FORM:BORD?

### 4.4.2 :FORMat[:DATA]

<b>描述</b>	设置数据输出格式。请参见“数据输出格式”。
<b>命令</b>	:FORMat [:DATA] <i>format</i> :FORMat [:DATA]?
<b>参数</b>	<i>format</i> 数据输出格式。参数数据类型为 CPD。 ASCII REAL,32 REAL,64。REAL,64 仅适用于由 :SYST:LANG “DEF”命令设置的 default 语言模式。 <i>format</i> =ASCII 指定 ASCII 格式（默认值）。 <i>format</i> =REAL,32 指定 IEEE-754 单精度格式。4 字节的数据。 <i>format</i> =REAL,64 指定 IEEE-754 双精度格式。8 字节的数据。
<b>查询响应</b>	<i>format</i> <newline> <i>format</i> 返回 ASC, REAL,32, 或 REAL,64。 响应数据类型为 CRD。
<b>示例</b>	:FORM REAL,32 :FORM?

#### 4.4.3 :FORMat:DIGital

	设置由以下命令定义的位模式的响应格式。 :CALCulate:CLIMits:<FAIL PASS>:DIGital[:DATA] :CALCulate:LIMit:COMPliance:DIGital[:DATA] :CALCulate:LIMit:<LOWer UPPer>:DIGital[:DATA] :CALCulate:LIMit:PASS:DIGital[:DATA]
描述	
命令	:FORMat:DIGital <i>format</i> :FORMat:DIGital?
参数	<i>format</i> 响应格式。ASCII (十进制, default) BINary OCTal HEXadecimal。 参数数据类型为 CPD。
查询响应	<i>format</i> <newline> <i>format</i> 返回 ASC、BIN、OCT 或 HEX。 响应数据类型为 CRD。
示例	:FORM:DIG BIN :FORM:DIG?

#### 4.4.4 :FORMat:ELEMents:CALCulate

	指定通过以下命令 :CALCulate:DATA?, :CALCulate:DATA:LATest?, :CALCulate:MATH:DATA?, :CALCulate:MATH:DATA:LATest? 或 :TRACe:DATA? 返回的计算结果数据中所包含的元素。 对于存储在 trace buffer 中的数据, 此命令对由 :TRACe:FEED MATH LIM 命令指定的计算结果数据或限值测试数据有效。 如果此命令指定所有元素, 结果数据包含如下所示的所有元素。元素的顺序是排他性的。例如, 如果未指定 time, 则数据按此顺序包含 time 计算和 status 状态数据, 如果未输入此命令, 则该数据只包含 calc 计算数据。 元素及其顺序: calc, time, status
描述	
命令	:FORMat:ELEMents:CALCulate <i>type{,type}</i> :FORMat:ELEMents:CALCulate?
参数	<i>type</i> 数据中包含的数据元素。 CALC (calculation data, default) TIME STATus。参数数据类型为 CPD。 CALC 选择计算数据 calc; TIME 选择时间 (时间戳) 数据 time; STAT 选择状态数据 status;
查询响应	<i>type{,type}</i> <newline> <i>type</i> 返回 CALC, TIME or STAT 响应数据类型为 CRD。多个响应用一个逗号隔开。

---

示例	:FORM:ELEM:CALC CALC,TIME,STAT :FORM:ELEM:CALC?
----	--

---

#### 4.4.5 :FORMat:ELEMents:SENSe

指定通过以下命令 :FETCh?, :READ?, :MEASure? 或 :TRACe:DATA? 返回的 sense 数据或测量结果数据中包含的元素。

对于存储在跟踪缓冲区中的数据，此命令对由 :TRACe:FEED SENS 命令指定的测量结果数据有效。

描述	如果未输入此命令，或者由此命令指定了所有元素，sense 或 measure 测量结果数据包含了下面所示的所有元素。元素的顺序是排他性的。 例如，如果没选中 VOLTage 和 RESistance，该数据按此顺序包含 current、time、status 和 source 数据。它将不包含电压和电阻的数据。 元素及其顺序：voltage, current, resistance, time, status, source
----	--

---

命令	:FORMat:ELEMents:SENSe type{,type} :FORMat:ELEMents:SENSe?
----	---

---

*type* 数据中包含的数据元素。  
VOLTage|CURRent|RESistance|TIME|STATus|SOURce。参数类型为 CPD。  
VOLT 选择电压测量数据 *voltage*.  
CURR 选择电流的测量数据 *current*.  
RES 选择电阻测量数据 *resistance*.  
TIME 选择时间数据 *time*(测量开始触发的时间戳).  
STAT 选择状态数据 *status*.  
SOUR 选择源输出设置数据 *source*.

---

查询响应	<i>type{,type}&lt;newline&gt;</i> <i>type</i> 返回 VOLT, CURR, RES, TIME, STAT, 或 SOUR。 响应数据类型为 CRD。多个响应用一个逗号隔开。
------	--

---

示例	:FORM:ELEM:SENS SOUR,CURR,VOLT,RES,TIME,STAT :FORM:ELEM:SENS?
----	--

---

#### 4.4.6 :FORMat:SREGister

描述	设置状态字节寄存器的响应格式。
----	-----------------

---

命令	:FORMat:SREGister format :FORMat:SREGister?
----	--

---

参数	<i>format</i> 响应格式。ASCII(十进制, default)  BINary OCTal HEXadecimal。 参数类型为 CPD。
----	--

---

---

	<i>format</i> <newline>
查询响应	<i>format</i> 返回 ASC、BIN、OCT 或 HEX。 响应数据类型为 CRD。
示例	:FORM:SREG BIN :FORM:SREG?

---

## 4.5 HCOPy Subsystem

### 4.5.1 :HCOPy:SDUMp:DATA?

---

描述	返回前面板屏幕图像的数据。 图像数据格式由命令 :HCOPy:SDUMp:FORMAT 指定。
命令	:HCOPy:SDUMp:DATA?
参数	无
查询响应	返回是一个确定长度的任意二进制数据块。
示例	:HCOP:SDUM:DATA?

---

### 4.5.2 :HCOPy:SDUMp:FORMAT

---

描述	设置图像数据格式。 前面板屏幕图像将按照此命令设置的格式创建。 图像数据将由通过 :HCOPy:SDUMp:DATA? 命令返回。
命令	:HCOPy:SDUMp:FORMAT format :HCOPy:SDUMp:FORMAT?
参数	<i>format</i> 图像数据格式。 JPG (default) BMP PNG WMF。 参数类型为 CPD.
查询响应	<i>format</i> <newline> <i>format</i> 返回 JPG, BMP, PNG, 或, WMF。 响应数据类型为 CRD。
示例	:HCOP:SDUM:FORM BMP :HCOP:SDUM:FORM?

---

## 4.6 MEASure Subsystem

### 4.6.1 :MEASure?

描述	对通过命令 :SENSe:FUNCTION[:ON] 打开的测量项目执行一次测量 (one-shot) , 并返回命令 :FORMAT:ELEMENTS:SENSe 指定的测量数据。在执行测量命令之前必须先通过 SCPI 指令或前面板设置测量参数。
命令	:MEASure? [chanlist]
参数	<p>chanlist 执行测量的通道。            参数类型为 channel list. (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1).            (@1) 仅选择通道 1。            (@2) 仅选择通道 2。            (@1,2), (@1:2), (@2,1), 和(@2:1) 同时选择通道 1 和通道 2。            如果未指定此参数, 默认 chanlist = (@1).</p>
查询响应	<p><i>response</i>&lt;newline&gt;  <i>response</i> 返回测量结果数据。响应数据类型为 NR3。参见“数据输出格式”。  <i>response</i> 使用 ASCII 数据输出格式, 每个数据都用逗号分隔, 如果通道 1 和通道 2 都被 chanlist 选中, 将按顺序返回通道 1 数据和通道 2 数据。示例如下:</p> <p><i>ch1curr, ch1sour, ch2curr, ch2sour</i></p> <p>该示例包含通道 1 的电流数据 (ch1curr) 和源数据 (ch1sour), 以及通道 2 的电流数据 (ch2curr) 和源数据 (ch2sour)。            如果未启用测量功能或不存在数据, <i>response</i> 将返回+9.910000E+37 (ASCII) 或 NaN (IEEE-754)。</p>
示例	<pre>:SENS:FUNC "CURR" :FORM:ELEM:SENS CURR,SOUR :MEAS? (@1,2)</pre>

### 4.6.2 :MEASure:<CURRent|RESistance|VOLTage>?

描述	执行一次 (one-shot) 测量, 并返回测量结果数据。在执行此命令之前, 必须通过 SCPI 命令或前面板操作来设置测量条件。测量项目可以设置为 CURRent, RESistance, 或 VOLTage.
命令	<pre>:MEASure:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;? [chanlist] For &lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;, select CURRent[:DC] for current measurement, RESistance for resistance measurement, or VOLTage[:DC] for voltage measurement.</pre>
参数	<i>chanlist</i> 进行测量的通道。

---

参数类型为 channel list. (@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1).

(@1) 仅选择通道 1。

(@2) 仅选择通道 2。

(@1,2), (@1:2), (@2,1), 和(@2:1) 同时选择通道 1 和通道 2。

如果未指定此参数, 默认 chanlist = (@1) 。

---

*response*<newline>

*response* 返回测量结果数据。响应数据类型为 NR3。参见“数据输出格式”。

*response* 使用 ASCII 数据输出格式, 每个数据都用逗号分隔, 如果通道 1 和通道 2 都被 chanlist 选中, 将按顺序返回通道 1 数据和通道 2 数据。示例如下:

*ch1curr, ch2curr*

示例包含通道 1 和通道 2 的电流测量数据 (*ch1curr*) 和 (*ch2curr*)

如果未启用测量功能或不存在数据, *response* 将返回+9.910000E+37 (ASCII) 或 NaN (IEEE-754)。

---

### 查询响应

### 示例

:SENS:FUNC CURR

:MEAS:CURR? (@2,1)

---

## 4.7 MMEMory Subsystem

### 4.7.1 :MMEMory:CATalog?

---

描述	返回内存的使用量和可用量。此外, 还将返回当前指定目录中的文件和文件夹的列表。
----	---

---

命令	:MMEMory:CATalog? [directory]
----	-------------------------------

---

*directory* 目录名, <path>|USB:\<path>。 / (斜杠) 或\ (反斜杠) 可以作为路径分隔符。最多有 255 个 ASCII 字符。参数类型为 SPD。

如果 *directory* 未设置, 此功能将应用于当前目录。

参数	如果 <i>directory</i> = <path>, 返回目录 <current directory>\<path> 的使用量和可用量。
----	---

---

如果 *directory* = USB:\<path>, 返回目录 USB:\<path>的使用量和可用量。其中, USB:\ 是连接到前面板的 USB 内存的根目录。

如果指定的目录不存在或被设置为隐藏或系统, 则会出错。

---

*used,free{,item}<newline>*

*used* 返回已使用的空间的大小, 以字节为单位。响应数据类型为 NR1。

*free* 返回可用空间的大小, 以字节为单位。响应数据类型为 NR1。

### 查询响应

*item* 返回该文件或目录信息。响应数据类型为 SRD。

对于文件, *item* 返回一个字符串 *name,type,size*, 以指示文件名、文件类型和文件大小。其中, *type* 返回“ASC”、“BIN”、“STAT”和“MACR”, 分别表示文件扩展名为“csv”、“dat”、“sta”、和“mac”的文件。

---

---

对于目录, *item* 返回字符串 *name,type,size*。其中, *name* 表示目录名称, *type,size* 总是返回“FOLD,0”。

---

**示例**

```
:MMEM:CAT? "USB:\\device1\\iv_test\\result"
:MMEM:CAT? "device1\\iv_test\\result"
```

---

#### 4.7.2 :MMEMory:CDIRectory

<b>描述</b>	切换当前目录到指定的目录.
<b>命令</b>	<pre>:MMEMory:CDIRectory directory :MMEMory:CDIRectory?</pre>
<b>参数</b>	<p><i>directory</i> 目录名称, &lt;path&gt; USB:\&lt;path&gt;。/ (斜杠) 或\ (反斜杠) 可以作为路径分隔符。最多有 255 个 ASCII 字符。参数类型为 SPD。</p> <p>如果 <i>directory</i>=&lt;path&gt;, 下一个当前目录将是&lt;current directory&gt;\&lt;path&gt;。</p> <p>如果 <i>directory</i>= USB:\&lt;path&gt;, 下一个当前目录将是 USB:\&lt;path&gt;。其中, USB:\ 是连接到前面板的 USB 内存的根目录。</p> <p>如果指定的目录不存在或被设置为隐藏或系统, 则会出错。</p>
<b>查询响应</b>	<i>directory</i> <newline> <i>directory</i> 返回当前目录的完整路径。响应数据类型为 SRD.
<b>示例</b>	<pre>:MMEM:CDIR "USB:\\device1\\iv_test\\result" :MMEM:CDIR?</pre>

---

#### 4.7.3 :MMEMory:COPY

<b>描述</b>	在当前目录中生成现有文件的副本.
<b>命令</b>	<pre>:MMEMory:COPY source,destination</pre>
<b>参数</b>	<p><i>source</i> 源文件名。原始文件的名称。</p> <p><i>destination</i> 复制文件名。或目录名称, &lt;path&gt; USB:\&lt;path&gt;. / (斜杠) 或\ (反斜杠) 可以作为路径分隔符。</p> <p>参数长度 最多有 255 个 ASCII 字符。参数类型为 SPD。</p> <p>如果 <i>destination</i> 是文件名, 该复制文件将在当前目录中创建。</p> <p>如果 <i>destination</i>=&lt;path&gt;, 源文件被复制在&lt;current directory&gt;\&lt;path&gt; 中。</p> <p>如果 <i>destination</i> = USB:\&lt;path&gt;, the source file is duplicated in USB:\&lt;path&gt;。其中, USB:\ 是连接到前面板的 USB 内存的根目录。</p> <p>如果源文件不存在或目标文件已存在, 则会出错。</p>
<b>查询响应</b>	无

---

---

示例	:MMEM:COPY "original.dat","original_copy.dat" :MMEM:COPY "original.dat","USB:\\device1\\iv_test\\result"
----	---

---

#### 4.7.4 :MMEMory:DELetE

---

描述	删除当前目录中的文件.
命令	:MMEMory:DELetE <i>file_name</i>
参数	<i>file_name</i> 要删除的文件的名称。 最多有 255 个 ASCII 字符。参数类型为 SPD。 如果指定的文件不存在，则会出错.
查询响应	无
示例	:MMEM:DEL "original_copy.dat"

---

#### 4.7.5 :MMEMory:LOAD:MACRo

---

描述	从当前目录中的指定文件中加载一个宏。
命令	:MMEMory:LOAD:MACRo <i>macro,file_name</i>
参数	<i>macro</i> 宏的名称。 <i>file_name</i> 包含该宏的文件的名称。文件扩展名必须是 <i>mac</i> 。 参数长度为最多有 255 个 ASCII 字符。参数类型为 SPD.
查询响应	无
示例	:MMEM:LOAD:MACR "abc","MacroData1.mac"

---

#### 4.7.6 :MMEMory:LOAD:STATe

---

描述	从当前目录中的指定文件加载仪器设置.
命令	:MMEMory:LOAD:STATe <i>file_name</i>
参数	<i>file_name</i> 包含仪器设置的文件的名称。文件扩展名必须是 <i>sta</i> 。 最多有 255 个 ASCII 字符。参数类型为 SPD.
查询响应	无
示例	:MMEM:LOAD:STAT "SetupData1.sta"

---

#### 4.7.7 :MMEMory:MDIRectomy

<b>描述</b>	创建新目录.
<b>命令</b>	:MMEMory:MDIRectomy directory
<b>参数</b>	<p>directory 目录名, &lt;path&gt; USB:\&lt;path&gt;。 / (斜杠) 或\ (反斜杠) 可以作为路径分隔符。最多有 255 个 ASCII 字符。参数类型为 SPD。</p> <p>如果 directory = &lt;path&gt;, 此命令将创建一个 &lt;current directory&gt;\&lt;path&gt; 目录。</p> <p>如果 directory = USB:\&lt;path&gt;, 此命令将创建一个 USB:\&lt;path&gt; 目录。</p> <p>其中, USB:\是连接到前面板的 USB 内存的根目录。</p>
<b>查询响应</b>	无
<b>示例</b>	:MMEM:MDIR "USB:\\device1\\iv_test\\setup"

#### 4.7.8 :MMEMory:MOVE

<b>描述</b>	移动或重命名当前目录中的现有文件.
<b>命令</b>	:MMEMory:MOVE source,destination
<b>参数</b>	<p>source 源文件名, 原始文件的名称。</p> <p>destination 新文件名, 或目录名称, &lt;path&gt; USB:\&lt;path&gt;。 / (斜杠) 或\ (反斜杠) 可以作为路径分隔符。</p> <p>参数长度最多有 255 个 ASCII 字符。参数类型为 SPD.</p> <p>如果 destination 是文件名, 源文件将被重命名为当前目录中的新文件名。</p> <p>如果 destination = &lt;path&gt;, 源文件将被移动到 &lt;current directory&gt;\&lt;path&gt;.</p> <p>如果 destination = USB:\&lt;path&gt;, 源文件将被移动到 USB:\&lt;path&gt;。</p> <p>其中, USB:\是连接到前面板的 USB 内存的根目录。</p> <p>如果源文件不存在或目标文件已存在, 则会出错.</p>
<b>查询响应</b>	无
<b>示例</b>	<pre>:MMEM:MOVE "original.dat","new.dat" :MMEM:MOVE "original.dat","USB:\\device1\\iv_test\\result"</pre>

#### 4.7.9 :MMEMory:RDIRectomy

<b>描述</b>	删除指定的空目录。
<b>命令</b>	:MMEMory:RDIRectomy directory
<b>参数</b>	directory 目录名称, <path> USB:\<path>。 / (斜杠) 或\ (反斜杠) 可

以作为路径分隔符。最多有 255 个 ASCII 字符。参数类型为 SPD。  
如果 *directory*=<path>, 此命令将删除 <current directory>\<path> 目录。  
如果 *directory*=USB:\<path>, 此命令将删除 USB:\<path>目录。其中,  
USB:\ 是连接到前面板的 USB 内存的根目录。  
如果指定的目录不为空，则会出错。

**查询响应**

无

**示例**

:MMEM:RDIR "USB:\\device1\\iv\_test\\setup"

**4.7.10 :MMEMory:STORe:DATA<:LIMit[:MATH[:SENSe[:ALL]]>**

**描述** 将指定通道的限值测试数据、数学表达式结果数据、sense 数据或所有这些数据保存到当前目录中的指定文件中。

:MMEMory:STORe:DATA&lt;:LIMit[:MATH[:SENSe[:ALL]]&gt;

*file\_name*[,chanlist]

对于&lt;:LIMit[:MATH[:SENSe[:ALL]]&gt;,

**命令**

:LIMit 表示限值测试数据；

:MATH 表示数学表达式计算结果数据；

:SENSe 表示 sense 测量数据；

[:ALL] 表示以上全部数据。

*file\_name* 用于保存指定数据的文件的名称。最多有 255 个 ASCII 字符。参数类型为 SPD。文件扩展名必须是 *dat*, 这对于 :MMEMory:CATalog? 结果是有意义的。**参数***chanlist* 用于收集数据的通道。参数类型为 channel list.。

(@1)|( @2)|( @1,2)|( @1:2)|( @2,1)|( @2:1).

( @1) 仅选择通道 1。

( @2) 仅选择通道 2。

( @1,2), ( @1:2), ( @2,1), 和( @2:1) 同时选择通道 1 和通道 2。

如果 chanlist 未被指定， 默认 chanlist = (@1)。

**查询响应**

无

**示例**

:MMEM:STOR:DATA "AllData1.dat"

**4.7.11 :MMEMory:STORe:MACRo****描述**

将宏保存到当前目录中的指定文件中.

**命令**

:MMEMory:STORe:MACRo macro,file\_name

*macro* 宏的名称。

<b>参数</b>	<i>file_name</i> 用于保存该宏的文件的名称。文件扩展名必须是 mac, 这对于 :MMEMory:CATalog? 结果是有意义的。 参数长度最多有 255 个 ASCII 字符。参数类型为 SPD。
<b>查询响应</b>	无
<b>示例</b>	:MMEM:STOR:MACR "abc","MacroData1.mac"

#### 4.7.12 :MMEMory:STORe:STATe

<b>描述</b>	将仪器设置保存到当前目录中的指定文件中。
<b>命令</b>	:MMEMory:STORe:STATe <i>file_name</i>
<b>参数</b>	<i>file_name</i> 用于保存仪器设置的文件的名称。最多有 255 个 ASCII 字符。 参数类型为 SPD。文件扩展名必须是 sta, 表示 :MMEMory:CATalog? 结果。
<b>查询响应</b>	无
<b>示例</b>	:MMEM:STOR:STAT "SetupData1.sta"

#### 4.7.13 :MMEMory:STORe:TRACe

<b>描述</b>	将指定通道的跟踪缓冲区中的所有数据保存到当前目录中的指定文件。
<b>命令</b>	:MMEMory:STORe:TRACe <i>file_name</i> [, <i>chanlist</i> ]
<b>参数</b>	<i>file_name</i> 用于保存指定数据的文件的名称。最多有 255 个 ASCII 字符。 参数类型为 SPD。文件扩展名必须是 tra, 这对 :MMEMory:CATalog? 结果有意义。  <i>chanlist</i> 指定获取数据的通道。参数类型为 channel list。 (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1). (@1) 仅选择通道 1。 (@2) 仅选择通道 2。 (@1,2), (@1:2), (@2,1), 和(@2:1) 同时选择通道 1 和通道 2。 如果 <i>chanlist</i> 未被指定, 默认 <i>chanlist</i> = (@1).
<b>查询响应</b>	无
<b>示例</b>	:MMEM:STOR:TRAC "AllTraceData1.dat"

## 4.8 OUTPut Subsystem

### 4.8.1 :OUTPut:FILT:AUTo

<b>描述</b>	启用或禁用自动过滤器功能。
<b>命令</b>	:OUTPut[c]:FILT:AUTo mode :OUTPut[c]:FILT:AUTo?
	<i>mode</i> 1 ON (default) 0 OFF。参数类型为 boolean 布尔型。 <i>mode</i> =0 or OFF 禁用自动过滤器功能。 <i>mode</i> =1 or ON 启用自动过滤器功能。
<b>参数</b>	如果启用此功能，该仪器将自动设置输出滤波器，从而提供最佳的滤波器特性和截止频率。此时将忽略以下命令设置。 :OUTPut:FILT[:LPASs]:FREQuency :OUTPut:FILT[:LPASs]:TCONstant
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 为 0 或 1，分别表示自动过滤功能关闭或开启。 响应数据类型为 NR1。
<b>示例</b>	:OUTP:FILT:AUTo 1 :OUTP2:FILT:AUTo?

### 4.8.2 :OUTPut:FILT[:LPASs]:FREQuency

<b>描述</b>	设置输出滤波器的截止频率。如果通过 :OUTPut:FILT:AUTo 命令启用了自动过滤器功能，则会忽略此命令设置。
<b>命令</b>	:OUTPut[c]:FILT[:LPASs]:FREQuency <i>frequency</i> :OUTPut[c]:FILT[:LPASs]:FREQuency? [ <i>frequency</i> ]
	<i>frequency</i> value (31.830 Hz to +31.831 kHz) MINimum MAXimum DEFault. Default 是 MAXimum 最大值。 参数类型为 NRF+。查询不支持 <i>frequency</i> = <i>value</i> 。
<b>参数</b>	如果您指定的值小于最小值或大于最大值， <i>frequency</i> 被自动设置为最小值或最大值。截止频率可以用以下公式表示， <i>time_constant</i> 时间常数由 :OUTPut:FILT[:LPASs]:TCONstant 指令设置。所以对于 <i>frequency</i> 和 <i>time_constant</i> ，改变其中一个，另一个同步变化。 $frequency = 1/(2 \times \pi \times time\_constant)$
<b>查询响应</b>	<i>frequency</i> <newline> <i>frequency</i> 返回当前设置的截止频率。 如果指定了一个参数， <i>frequency</i> 返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR3。

---

示例	:OUTP:FILT:FREQ 1E4 :OUTP2:FILT:LPAS:FREQ?
----	---

---

#### 4.8.3 :OUTPut:FILT[LPASs][STATe]

描述	启用或禁用输出过滤器。
命令	:OUTPut[c]:FILT[LPASs][STATe] mode :OUTPut[c]:FILT[LPASs][STATe]?
参数	mode 0 OFF 1 ON (default)。参数类型为 boolean。 mode=1 or ON 启用输出过滤器。 mode=0 or OFF 禁用输出过滤器。
查询响应	mode 为 0 或 1，并分别表示输出滤波器已关闭或打开。 响应数据类型为 NR1。
示例	:OUTP:FILT 0 :OUTP2:FILT:LPAS:STAT?

---

#### 4.8.4 :OUTPut:FILT[LPASs]:TCOnstant

描述	设置时间常数，而不是设置输出滤波器的截止频率。 如果通过:OUTPut:FILT:AUTO 命令启用了自动过滤器功能，则会忽略此命令设置。
命令	:OUTPut[c]:FILT[LPASs]:TCOnstant time_constant :OUTPut[c]:FILT[LPASs]:TCOnstant? [time_constant]
参数	time_constant value(5 μs to 5 ms) MINimum MAXimum DEFault。 参数类型 NRf+。查询不支持 time_constant = value。 如果您指定的值小于最小值或大于最大值，time 自动设置为最小值或最大值。时间常数可以用以下公式表示，截止频率 frequency 由指令 :OUTPut:FILT[LPASs]:FREQuency 设置。所以对于 frequency 和 time_constant，改变其中一个，另一个同步变化。 $time\_constant = 1/(2 \times \pi \times frequency)$
查询响应	time_constant<newline> time_constant 返回截止频率的当前设置。 如果指定了参数，time_constant 返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR3。
示例	:OUTP:FILT:TCON 5E-3 :OUTP2:FILT:LPAS:TCON?

---

#### 4.8.5 :OUTPut:HCPacitance[:STATe]

<b>描述</b>	启用或禁用高电容模式。该模式对高电容性待测设备很有效.
<b>命令</b>	:OUTPut[c]:HCPacitance[:STATe] mode :OUTPut[c]:HCPacitance[:STATe]?
<b>参数</b>	<i>mode</i> 1 ON 0 OFF (default)。参数类型为 boolean。 <i>mode</i> =1 or ON 启用高电容模式. <i>mode</i> =0 or OFF 禁用高电容模式。
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 为 0 或 1, 分别表示高电容模式为关闭或开启。 响应数据类型为 NR1.
<b>示例</b>	:OUTP:HCP 1 :OUTP2:HCP:STAT?

#### 4.8.6 :OUTPut:LOW

<b>描述</b>	选择低端子的状态。在执行此命令之前，必须通过 :OUTPut[:STATe] 命令禁用源输出。否则，就会发生一个错误。
<b>命令</b>	:OUTPut[c]:LOW low_state :OUTPut[c]:LOW?
<b>参数</b>	<i>low_state</i> FLOat GROund (default)。参数类型为 CPD。 <i>low_state</i> =FLOat 设置浮地状态。 <i>low_state</i> =GROund 设置 ground 模式， low 端子连接到接地。
<b>查询响应</b>	<i>low_state</i> <newline> <i>low_state</i> 为 FLO 或 GRO，并表示 low 端子状态。 响应数据类型为 CRD.
<b>示例</b>	:OUTP:LOW FLO :OUTP2:LOW?

#### 4.8.7 :OUTPut:OFF:AUTO

<b>描述</b>	启用或禁用自动关闭输出功能.
<b>命令</b>	:OUTPut[c]:OFF:AUTO mode :OUTPut[c]:OFF:AUTO?
<b>参数</b>	<i>mode</i> 1 ON 0 OFF (default)。参数类型为 boolean。 <i>mode</i> =0 or OFF 禁用自动关闭输出功能。 <i>mode</i> =1 or ON 启用自动关闭输出功能。如果启用此功能，则当分组通道的状态从繁忙变为空闲时，源输出将立即自动关闭。

---

	<i>mode</i> <newline>
查询响应	<i>mode</i> 为 0 或 1，分别表示自动关闭输出功能为关闭或开启。 响应数据类型为 NR1。
示例	:OUTP:OFF:AUTO 1 :OUTP2:OFF:AUTO?

---

#### 4.8.8 :OUTPut:OFF:MODE

---

描述	选择在输出关闭后的源状态.
命令	:OUTPut[c]:OFF:MODE <i>mode</i> :OUTPut[c]:OFF:MODE?
mode ZERO HIZ NORMAl (default). 参数类型为 CPD.	
mode = NORMAl 时:	
	输出关闭后断开输出继电器。 输出关闭时，为电压源模式，电压设置值 0V，电流 Limit 100uA。
mode = HIZ 时:	
	输出关闭后断开输出继电器。 输出关闭时，电压设置值≤40V。电流设置值≤100mA。.
mode = ZERO 时:	
	输出关闭后不断开输出继电器。 输出关闭时，为电压源模式，电压设置值 0V，电流 Limit 100uA。

**NOTE:** 此命令设置不适用于由过电压/电流保护、联锁打开和过温保护等紧急情况触发的输出关闭过程，输出电压被立即被设置为 0V，输出开关被设置为关闭。

---

	mode <newline>
查询响应	<i>mode</i> 返回 NORM, HIZ, 或 ZERO，并表示输出关闭后的源状态。 响应数据类型为 CRD。
示例	:OUTP:OFF:MODE HIZ :OUTP2:OFF:MODE?

---

#### 4.8.9 :OUTPut:ON:AUTO

---

描述	启用或禁用自动打开输出功能。
命令	:OUTPut[c]:ON:AUTO <i>mode</i> :OUTPut[c]:ON:AUTO?

---

---

<b>参数</b>	<i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean。 <i>mode</i> =0 or OFF 禁用自动打开输出功能。 <i>mode</i> =1 或 ON 启用自动打开输出功能。如果启用了此功能，则当发送 :INITiate 或:READ 命令时，源输出将自动打开。
<b>查询响应</b>	mode <newline> <i>mode</i> 为 0 或 1，分别表示自动开启输出功能关闭或开启。 响应数据类型为 NR1.
<b>示例</b>	:OUTP:ON:AUTO 0 :OUTP2:ON:AUTO?

---

#### 4.8.10 :OUTPut:PROTection[:STATe]

---

<b>描述</b>	启用或禁用过电压/电流保护。如果启用此功能且触发合规状态时，源/测量单元 (SMU) 将立即将源输出设置为 0V，并立即自动关闭输出。
<b>命令</b>	:OUTPut[c]:PROTection[:STATe] <i>mode</i> :OUTPut[c]:PROTection[:STATe]?
<b>参数</b>	<i>mode</i> 1 ON 0 OFF (default)。参数类型为 boolean。 <i>mode</i> =0 or OFF 禁用过电压/电流保护。 <i>mode</i> =1 or ON 启用过电压/电流保护。
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 为 0 或 1，分别表示过电压/电流保护功能关闭或打开。 响应数据类型为 NR1.
<b>示例</b>	:OUTP:PROT 1 :OUTP2:PROT:STAT?

---

#### 4.8.11 :OUTPut:RECall

---

<b>描述</b>	恢复至由:OUTPut:SAVE 命令保存的通道设置。
<b>命令</b>	:OUTPut[c]:RECall <i>index</i>
<b>参数</b>	<i>index</i> 0 1。参数类型为 NR1。 <i>index</i> =0 用于调用通道设置 0。 <i>index</i> =1 用于调用通道设置 1。
<b>查询响应</b>	无
<b>示例</b>	:OUTP:REC 1

---

#### 4.8.12 :OUTPut:SAVE

<b>描述</b>	保存通道设置。该设置可以通过 :OUTPut:RECall 命令恢复。
<b>命令</b>	:OUTPut[c]:SAVE index
<b>参数</b>	<p><i>index</i> 0 1。参数类型为 NR1。</p> <p><i>index</i>= 0 is 用于将当前的通道设置记忆为通道设置 0。</p> <p><i>index</i>= 1 is 用于记忆当前的通道设置作为通道设置 1。</p>
<b>查询响应</b>	无
<b>示例</b>	:OUTP:SAVE 1

#### 4.8.13 :OUTPut[:STATe]

<b>描述</b>	启用或禁用源输出.
<b>命令</b>	<p>:OUTPut[c][:STATe] mode</p> <p>:OUTPut[c][:STATe]?</p>
<b>参数</b>	<p><i>mode</i> 1 ON 0 OFF (default)。参数类型为 boolean。</p> <p><i>mode</i>= 1 or ON 启用源输出。</p> <p><i>mode</i>= 0 or OFF 禁用源输出。</p>
<b>查询响应</b>	<p><i>mode</i>&lt;newline&gt;</p> <p><i>mode</i> 为 0 或 1, 分别表示源输出已关闭或已打开。</p> <p>响应数据类型为 NR1。</p>
<b>示例</b>	<p>:OUTP 1</p> <p>:OUTP2:STAT?</p>

### 4.9 READ Subsystem

#### 4.9.1 :READ:ARRay?

<b>描述</b>	依次执行 :INITiate 命令和:FETCh:ARRay? 命令，并返回测量结果数据，其中包含由 :FORMat:ELEMents:SENSe 命令指定的电压测量数据、电阻测量数据、时间数据、状态数据或源输出设置数据。 在执行 :INITiate、:MEASure 或 :READ 命令之前，不会清除数据。
<b>命令</b>	:READ:ARRay? [chanlist]
<b>参数</b>	<p><i>chanlist</i> 用来返回数据的通道。参数类型为 channel list。</p> <p>(@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1).</p> <p>(@1) 仅选择通道 1。</p> <p>(@2) 仅选择通道 2。</p>

(@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2.

如果未指定此参数, chanlist = (@1) 被设置.

*response*<newline>

*response* 返回由 :FORMat, :ELEMents, :SENSe 命令指定的数组数据。

响应数据类型为 NR3。参见“数据输出格式”。

*response* 使用 ASCII 数据输出格式, 每个数据都用逗号分隔, 如果通道 1 和通道 2 都被 chanlist 选中, 将按顺序返回通道 1 数据和通道 2 数据。示例如下:

ch1curr1, ch1sour1, ch2curr1, ch2sour1,

ch1curr2, ch1sour2, ch2curr2, ch2sour2, .....

ch1curr5, ch1sour5, ch2curr5, ch2sour5,

ch1curr6, ch1sour6, +9.910000E+37, +9.910000E+37, .....

ch1curr10, ch1sour10, +9.910000E+37, +9.910000E+37

该示例显示了包含通过通道 1 的 10 步扫描测量的当前数据 (ch1currN)

和源数据 (ch1sourN) 的数据, 以及通道 2 的 5 步扫描测量的当前数据

(ch2currN) 和源数据 (ch2sourN) .

如果测量功能未启用或不存在数据, response 返回+9.910000E+37 (ASCII) 或 NaN (IEEE-754)。

#### 查询响应

:FORM:ELEM:SENS CURR,SOUR

:READ:ARR? (@1,2)

### 4.9.2 :READ:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

#### 描述

依次执行 :INITiate 命令和 :FETCh:ARRay< CURRent|RESistance|SOURce|STATus|TIME|VOLTage>? 命令, 并返回测量数据, 其中包含由 CURRent, RESistance, SOURce, STATus, TIME, 或 VOLTage 指定的电流测量数据、电阻测量数据、源输出数据、输出设置数据、状态数据、时间数据、或电压测量数据。

在执行 :INITiate、:MEASure 或:READ 命令之前, 数据不会被清除.

#### 命令

:READ:ARRay:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

[chanlist]

对于<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>, 指定电流测量数据 CURRent, 电阻测量数据 RESistance, 源输出设置数据 SOURce, 状态数据 STATus, 时间数据 TIME, 或电压测量数据 VOLTage。

#### 参数

*chanlist* 用来返回数据的通道。参数类型为 channel list。

(@1)|(@2)|(@1,2)|(@1:2)|(@2,1)|(@2:1)。

(@1) 仅选择通道 1。

(@2) 仅选择通道 2。

(@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。

如果未指定此参数, 默认 *chanlist*=(@1).

---

*response*<newline>  
*response* 返回由 CURRent、RESistance、SOURce、STATus, TIME, 或 VOLTage 指定的数据。响应数据类型为 NR3。参见“数据输出格式”。

*response* 使用 ASCII 数据输出格式，每个数据都用逗号分隔，如果通道 1 和通道 2 都被 chanlist 选中，将按顺序返回通道 1 数据和通道 2 数据。示例如下：

查询响应	<i>ch1curr1, ch2curr1, ch1curr2, ch2curr2, .....</i> <i>ch1curr5, ch2curr5, ch1curr6, +9.910000E+37, .....</i> <i>ch1curr10, +9.910000E+37</i>
------	--

该示例显示了包含由通道 1 的 10 步扫描测量的当前数据 (*ch1currN*) 和通道 2 的 5 步扫描测量的当前数据 (*ch2currN*) 的数据。

如果测量功能未启用或不存在数据，响应将返回+9.910000E+37 (ASCII) 或 NaN (IEEE-754)。

---

示例	:READ:ARR:CURR? (@2,1)
----	------------------------

---

#### 4.9.3 :READ[:SCALar]?

描述	依次执行 :INITiate 命令和 :FETCH[:SCALar]? 命令，并返回由 :FORMat:ELEMents:SENSe 命令指定的最新数据：电压测量数据、电流测量数据、电阻测量数据、时间数据、状态数据或源输出设置数据。 在执行 :INITiate、:MEASure 或 :READ 命令之前，不会清除数据。
----	---

命令	:READ[:SCALar]? [chanlist]
参数	<p><i>chanlist</i> 通过通道返回数据。参数类型为 channel list。            (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1)。            (@1) 仅选择通道 1。            (@2) 仅选择通道 2。            (@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。            如果未指定此参数，默认 chanlist = (@1).</p>

查询响应	<i>response</i> <newline> 响应返回由 :FORMat, :ELEMents, :SENSe 命令指定的最新数据。 响应数据类型为 NR3。请参见“数据输出格式”。 <i>response</i> 使用 ASCII 数据输出格式，每个数据都用逗号分隔，如果通道 1 和通道 2 都被 chanlist 选中，将按顺序返回通道 1 数据和通道 2 数据。示例如下：
------	--

*ch1curr10, ch1sour10, ch2curr5, ch2sour5*  
 该示例显示了包含通道 1 的 10 步扫描测量的最新当前数据 (*ch1curr10*) 和源数据 (*ch1sour10*)，以及通道 2 的 5 步扫描测量的最新电流数据 (*ch2curr5*) 和源数据 (*ch2sour5*) 的数据。  
 如果测量功能未启用或不存在数据，响应将返回+9.910000E+37 (ASCII) 或 NaN (IEEE-754)。

---

示例	:FORM:ELEM:SENS CURR,SOUR :READ? (@1,2)
----	--

---

#### 4.9.4 :READ[:SCALar]:<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>?

描述	依次执行 :INITiate 命令和:FETCH:ARRay< CURRent RESistance SOURce  STATus TIME VOLTage>? 命令，并返回最新的电流测量数据、电阻测量数据、源输出设置数据、状态数据、时间数据或电压测量数据。 在执行 :INITiate、:MEASure 或 :READ 命令之前，不会清除数据。
----	--

---

命令	:READ[:SCALar]:<CURRent RESistance SOURce STATus TIME VOLTage>? [chanlist]
----	---

对于<CURRent|RESistance|SOURce|STATus|TIME|VOLTage>，指定电流测量数据的 CURRent，电阻测量数据的 RESistance，源输出设置数据的 SOURce，状态数据的 STATus，时间数据的 TIME，或电压测量数据的 VOLTage.

---

参数	<p><i>chanlist</i> Channels to return the data。参数类型为 channel list。 (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1)。 (@1) 仅选择通道 1。 (@2) 仅选择通道 2。 (@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。 如果未指定此参数，默认 chanlist = (@1)。</p>
----	---

---

查询响应	<p><i>response</i>&lt;newline&gt; 响应返回由 CURRent、RESistance、SOURce、STATus、TIME，或 VOLTage 所指定的最新数据。响应数据类型为 NR3。请参见“数据输出格式”。 <i>response</i> 使用 ASCII 数据输出格式，每个数据都用逗号分隔，如果通道 1 和通道 2 都被 chanlist 选中，将按顺序返回通道 1 数据和通道 2 数据。示例如下：</p>
------	--

---

*ch1curr10, ch2curr5*

这个示例显示了包含由通道 1 的 10 步扫描测量的最新电流数据 (ch1curr10)，以及通过通道 2 的 5 步扫描测量的最新电流数据 (ch2curr5)。如果测量功能未启用或不存在数据，响应将返回+9.910000E+37 (ASCII) 或 NaN (IEEE-754)。

---

示例	:READ:CURR? (@2,1)
----	--------------------

---

### 4.10 SENSe Subsystem

#### 4.10.1 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture

描述	设置单点测量的积分时间。
----	--------------

---

命令	<p>:SENSe[:c]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:APERture <i>time</i>            :SENSe[:c]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:APERture? [<i>time</i>]            对于&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;, 指定 CURRent[:DC],            RESistance 或 VOLTage [:DC]测量项目并不重要, 因为时间值对所有项目            都通用。</p>
参数	<p><i>time</i> value (+8E-6 to +2 seconds) MINimum MAXimum DEFault            (默认 0.1 PLC, = 0.1/电源频率)。            参数类型为 NRf+。查询不支持 time = value。            如果指定的值小于 MIN 或大于 MAX, 则时间将自动设置为 MIN 或 MAX.            积分时间可以用以下公式来表示, 使用由            :SENSe:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:NPCLcycles 命令设            置的 NPCL 值。因此, 最后一个命令设置对时间和 npcl 都是有效的。  <i>time</i> = npcl /电源频率</p>
查询响应	<p><i>time</i>&lt;newline&gt;  <i>time</i> 返回集成时间的当前设置。            如果指定了一个参数, time 将返回分配给 DEF、MIN 或 MAX 的值。            响应数据类型为 NR3.</p>
示例	<pre>:SENS:CURR:APER 2E-3 :SENS2:CURR:DC:APER?</pre>

#### 4.10.2 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:APERture:AUTO

描述	启用或禁用自动孔径时间功能.
命令	<p>:SENSe[:d]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:APERture:AUTO  <i>mode</i>            :SENSe[:d]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:APERture:AUTO?            对于&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;, 指定 CURRent[:DC],            RESistance 或 VOLTage [:DC]测量项目并不重要, 因为时间值对所有项目            都通用.</p>
参数	<p><i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean.  <i>mode</i>=0 或 OFF 禁用自动孔径时间功能。  <i>mode</i>=1 或 ON 启用自动孔径时间功能。如果启用此功能, 仪器将自动            设置适合于测量量程的积分时间 (NPCL 值)。            自动孔径时间功能开/关与            :SENSe:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:NPCLcycles:AUTO            命令设置的自动 NPCL 开/关是等效的。</p>
查询响应	<p><i>mode</i>&lt;newline&gt;  <i>mode</i> 为 0 或 1, 分别表示自动孔径时间功能为关闭或开启。            响应数据类型为 NR1.</p>

---

示例	:SENS:CURR:APER:AUTO 0 :SENS2:CURR:DC:APER:AUTO?
----	---

---

#### 4.10.3 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles

描述	设置 NPLC 值，即设置单点测量的积分时间.
命令	<p>:SENSe[c]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:NPLCycles <i>nplc</i>            :SENSe[c]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:NPLCycles? [<i>nplc</i>]            对于&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;，指定 CURRent[:DC]，            RESistance 或 VOLTage [:DC]测量项目并不重要，因为时间值对所有项目            都通用.</p>
	<p><i>nplc</i> value (50Hz 电源下+4E-4 到 +100 秒, 60Hz 电源下+4.8E-4 到            +120 秒) MINimum MAXimum DEFault (默认为 0.1 PLC)。            参数类型为 NRf+。查询不支持 nplc= value。            如果指定的值小于 MIN 或大于 MAX, nplc 将自动设置为 MIN 或 MAX。</p>
参数	<p>NPLC 值与命令            :SENSe:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:APERture            设置的积分时间 time 可通过以下公式互相转换：  <math>nplc = time \times power\ line\ frequency</math></p>
查询响应	<p><i>nplc</i>&lt;newline&gt;  <i>nplc</i> 返回当前设置的 NPLC 值。            如果指定了一个参数, nplc 将返回分配给 DEF、MIN 或 MAX 的值。            响应数据类型为 NR3。</p>
示例	<p>:SENS:CURR:NPLC 0.2 :SENS2:CURR:DC:NPLC?</p>

---

#### 4.10.4 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:NPLCycles:AUTO

描述	启用或禁用自动 NPLC 功能.
命令	<p>:SENSe[d]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:NPLCycles:            AUTO mode            :SENSe[d]:&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;:NPLCycles:            AUTO?            对于&lt;CURRent[:DC] RESistance VOLTage[:DC]&gt;，指定 CURRent[:DC]，            RESistance 或 VOLTage [:DC]测量项目并不重要，因为时间值对所有项目            都通用.</p>
参数	<p><i>mode</i> 1 ON (default) 0 OFF. 参数类型为 boolean.  <i>mode</i>=0 或 OFF 禁用自动 NPLC 功能.</p>

---

*mode=1* 或 *ON* 启用自动 NPLC 功能。如果启用此功能，仪器将自动设置适合于测量量程的 NPLC 值（积分时间）。

自动 NPLC 开/关与 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:APERture:AUTO 命令设置的自动孔径时间开/关是等效的。

*mode<newline>*

**查询响应** *mode* 为 0 或 1，分别表示自动 NPLC 功能为关闭或开启。  
响应数据类型为 NR1。

**示例** :SENS:CURR:NPLC:AUTO 0  
:SENS2:CURR:DC:NPLC:AUTO?

#### 4.10.5 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel][:BOTH]

**描述** 设置指定通道的正负向合规值。

:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel][:BOTH]

*compliance*

:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel][:BOTH]?  
[*compliance*]

**命令** 对于<CURRent[:DC]|VOLTage[:DC]>，  
CURRent[:DC] 表示电流合规值；  
VOLTage[:DC] 表示电压合规值。

*compliance value* (取值范围参考数据手册) |MINimum| MAXimum|

**参数** DEFault(默认为 100μA 或 2 V)。参数类型为 NRf+。value 的有效值范围  
是从通道的最小测量值到最大测量值。查询不支持指定 *compliance = value*。

*compliance<newline>*

**查询响应** *compliance* 返回当前的合规值。  
如果指定了一个参数，则 *compliance* 将返回分配给 DEF、MIN 或 MAX 的  
值。响应数据类型为 NR3.

**示例** :SENS:CURR:PROT 1E-3  
:SENS2:CURR:DC:PROT:LEV?

#### 4.10.6 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:NEGative

**描述** 设置指定通道的负向合规值。

:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:NEGative

*compliance*

:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:NEGative?  
[*compliance*]

对于<CURRent[:DC]|VOLTage[:DC]>,  
 CURRent[:DC]表示电流合规值;  
 VOLTage[:DC]表示电压合规值。

**参数** *compliance* value (取值范围参考数据手册) |MINimum| MAXimum| DEFault (默认为-100μA 或 -2 V)。参数类型为 NRf+。value 的有效值范围是从通道的最小测量值到最大测量值。查询不支持指定 *compliance* = value.

**查询响应** *compliance*<newline> *compliance* 返回当前的合规值。如果指定了一个参数，则 *compliance* 将返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。

**示例** :SENS:CURR:PROT:NEG -1E-3  
 :SENS2:CURR:DC:PROT:NEG?

#### 4.10.7 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:POSitive

**描述** 设置指定通道的正侧符合性值.

:SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:POSitive  
*compliance*

**命令** :SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection[:LEVel]:POSitive?  
*[compliance]*  
 对于<CURRent[:DC]|VOLTage[:DC]>,  
 CURRent[:DC]表示电流合规值;  
 VOLTage[:DC]表示电压合规值。

**参数** *compliance* value (取值范围参考数据手册) |MINimum| MAXimum| DEFault (默认为+100μA 或 +2 V), 参数类型为 NRf+。value 的有效值范围是从通道的最小测量值到最大测量值。查询不支持指定 *compliance* = value.

**查询响应** *compliance*<newline> *compliance* 返回当前的合规值。  
 如果指定了一个参数，则 *compliance* 将返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。

**示例** :SENS:CURR:PROT:POS 1E-3  
 :SENS2:CURR:DC:PROT:POS?

#### 4.10.8 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:PROTection:TRIPped?

**描述** 返回指定通道的合规状态.

**命令** :SENSe[c]:<CURRent[:DC]|VOLTage[:DC]>:PROTection:TRIPped?

对于<CURRent[:DC]|VOLTage[:DC]>,  
 CURRent[:DC]表示电流合规值;  
 VOLTage[:DC]表示电压合规值。

参数	无
查询响应	<i>status</i> <newline> <i>status</i> 为 1 或 0, 表示该通道是否处于合规状态。响应数据类型为 NR1。
示例	:SENS:CURR:PROT:TRIP?

#### 4.10.9 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO

描述	启用或禁用指定的测量通道的自动量程功能。
命令	:SENSe[ <i>d</i> ]<CURRent[:DC] RESistance VOLTage[:DC]>:RANGe:AUTO <i>mode</i> :SENSe[ <i>d</i> ]<CURRent[:DC] RESistance VOLTage[:DC]>:RANGe:AUTO? 对于<CURRent[:DC] RESistance VOLTage[:DC]>, CURRent[:DC]表示电流测量; RESistance 表示电阻测量; VOLTage[:DC]表示电压测量
参数	<i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean。 <i>mode</i> =0 或 OFF 禁用自动量程. 如果禁用此功能, 则通过使用 :SENSe:<CURRent[:DC] RESistance VOLTage[:DC]>:RANGe[:UPPer] 命令设置的量程来执行测量。 <i>mode</i> =1 或 ON 实现自动量程。如果启用此功能, 该通道将自动设置为 执行测量提供最佳分辨率的量程。 如果手动选择了量程, 则禁用自动量程。
查询响应	<i>mode</i> <newline> <i>mode</i> 为 0 或 1, 分别表示自动量程为关闭或开启。响应数据类型为 NR1。
示例	:SENS:CURR:RANG:AUTO 0 :SENS2:CURR:DC:RANG:AUTO?

#### 4.10.10 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO:LLIMit

描述	指定自动量程切换的下限, 并设置最小测量量程, 以提供测量指定值的最大分辨率。 如果最小测量量程与最大测量量程相同, 则使用该量程进行测量。 如果最小量程大于最大量程, 则会发生错误。
命令	:SENSe[ <i>c</i> ]<CURRent[:DC] RESistance VOLTage[:DC]>:RANGe:AUTO:LLIMit <i>range</i>

---

:SENSe[:c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO:LLIMit? [*range*]

对于<CURRent[:DC]|RESistance|VOLTage[:DC]>,  
CURRent[:DC]表示电流测量;  
RESistance 表示电阻测量;  
VOLTage[:DC]表示电压测量;

---

*range* value|MINimum|MAXimum|DEFault。参数类型为 NRf+。查询不支持 range = value.

**参数**

*value* 用于电压测量时的取值范围。

*value* 用于电流测量时的取值范围。

*value* 用于电阻测量时的取值范围，电阻测量模式为 AUTO 时有效。

---

*range*<newline>

**查询响应**

*range* 返回当前设置的自动量程的最小量程。如果指定了一个参数, range 将返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。

**示例**


---

:SENS:CURR:RANG:AUTO:LLIM 1E-6

:SENS2:CURR:DC:RANG:AUTO:LLIM?

#### 4.10.11 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:MODE

---

选择自动量程测量的操作模式。

**描述** 如果 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:AUTO 命令禁用了自动测距，则此命令设置无效。

:SENSe[:d]:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:MODE *mode*

:SENSe[:d]:<CURRent[:DC]|VOLTage[:DC]>:RANGe:AUTO:MODE?

**命令**

对于<CURRent[:DC]|RESistance|VOLTage[:DC]>,  
CURRent[:DC]表示电流测量;  
RESistance 表示电阻测量;  
VOLTage[:DC]表示电压测量;

---

*mode* NORMal (default)|RESolution|SPEed。参数类型为 CPD。

*mode*=NORMal 支持基本操作和下换操作

*mode*=RESolution 支持基本操作和向上改变操作

*mode*=SPEed 支持基本操作和上下变化操作

**参数**

**基本操作**

自动选择能提供最佳分辨率的量程。

**向上变化操作**

如果测量值  $\text{data} \geq \text{value1}$ , 则测量后量程 range 向上增加。

$\text{value1} = \text{range} \times \text{rate} / 100$

## 向下变化操作

如果测量值  $\text{data} \leq \text{value2}$ ，则测量量程立即向下变化。

$$\text{value2} = \text{range} \times \text{rate} / 1000$$

## rate 值由命令

:SENSe:<CURRent[:DC]|VOLTage[:DC]>:RANGE:AUTO:THreshold 设置。

*mode*<newline>

**查询响应** *moder* 返回 NORM, RES, 或 SPE，分别表示自动测量量程的工作模式。  
响应数据类型为 CRD。

**示例** :SESS:CURR:RANG:AUTO:MODE SPE  
:SESS2:CURR:DC:RANG:AUTO:MODE?

#### 4.10.12 :SENSe:<CURRent[:DC]|VOLTage[:DC]>:RANGE:AUTO:THreshold

<b>描述</b>	设置自动测量量程切换的阈值率
<b>命令</b>	:SENSe[c]:<CURRent[:DC] VOLTage[:DC]>:RANGE:AUTO:THreshold <i>rate</i> :SENSe[c]:<CURRent[:DC] VOLTage[:DC]>:RANGE:AUTO:THreshold? [ <i>rate</i> ] 对于<CURRent[:DC] VOLTage[:DC]>, CURRent[:DC] 表示电流测量; VOLTage[:DC] 表示电压测量;
<b>参数</b>	<i>rate</i> value MINimum MAXimum DEFault (默认为 90 %)。 参数类型为 NRf+。有效值为 11 % 到 100 %。查询不支持 <i>rate</i> = value。
<b>查询响应</b>	<i>rate</i> <newline> <i>rate</i> 返回当前设置的自动测量量程切换的阈值率。如果指定了一个参数, <i>rate</i> 将返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
<b>示例</b>	:SESS:CURR:RANG:AUTO:THR 60 :SESS2:CURR:DC:RANG:AUTO:THR?

#### 4.10.13 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGE:AUTO:ULIMit

<b>描述</b>	指定自动测量量程的上限，并设置最大测量量程，以提供最佳分辨率对指定值的测量。 这对于通过 :SENSe:RESistance:MODE 命令设置为 AUTO 模式的电阻测量是有效的。 对于电流测量和电压测量，最大测量量程取决于合规值量程。
<b>命令</b>	:SENSe[d]:RESistance:RANGE:AUTO:ULIMit <i>range</i>

---

:SENSe[:d]:RESistance:RANGe:AUTO:ULIMit? [*range*]  
 :SENSe[:d]:<CURRent|VOLTage>[:DC]:RANGe:AUTO:ULIMit?  
 对于<CURRent[:DC] |VOLTage[:DC]>,  
 CURRent[:DC]表示电流测量; VOLTage[:DC]表示电压测量;

---

**参数** *range* value |MINimum|MAXimum| DEFault(默认为 200 M)。  
 参数类型为 NRf+。查询不支持 range = value。

---

**查询响应** *range*<newline>  
*range* 返回当前设置的自动范围的最大测量量程. 如果指定了一个参数,  
 则 range 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3.

---

**示例** :SENS:RES:RANG:AUTO:ULIM 1E6  
 :SENS2:CURR:DC:RANG:AUTO:ULIM?

---

#### 4.10.14 :SENSe:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe[:UPPer]

**描述** 指定预期的测量值, 自动选择能提供最佳分辨率的测量量程.

---

:SENSe[:c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:UPPer  
*range*  
 :SENSe[:c]:<CURRent[:DC]|RESistance|VOLTage[:DC]>:RANGe:UPPer?  
 [*range*]  
 对于<CURRent[:DC]|RESistance|VOLTage[:DC]>,  
 CURRent[:DC]表示电流测量;  
 RESistance 表示电阻测量;  
 VOLTage[:DC]表示电压测量;

---

*range* value|UP|DOWN|MINimum|MAXimum|DEFault.  
 参数类型为 NRf+。查询不支持 range = value, UP 和 DOWN。  
 value 用于电压测量时的取值范围。

**参数** value 用于电流测量时的取值范围。  
 value 用于电阻测量时的取值范围, 电阻测量模式为 AUTO 时有效。  
*range*= UP 设置下一个更高的测量量程。  
*range*= DOWN 设置下一个较低的测量量程.

---

**查询响应** *range*<newline>  
*range* 返回当前设置的测量量程。如果指定了一个参数, range 将返回分配  
 给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3.

---

**示例** :SENS:CURR:RANG:UPP 1  
 :SENS2:CURR:DC:RANG:UPP?

---

#### 4.10.15 :SENSe:DATA?

描述	返回命令 :FORMat:ELEMents:SENSe 指定的电流测量数据、电压测量数据、电阻测量数据、电阻测量数据、源输出设置数据、状态数据或时间数据的阵列数据。 在执行 :INITiate、:MEASure 或 :READ 命令之前，不会清除数据.
命令	:SENSe[c]:DATA? [ <i>offset[, size]</i> ]
参数	<p><i>offset</i> 指示所接收到的数据的开始时间。n CURREnt STARt (default)。 参数类型为 NR1 或 CPD。</p> <p><i>offset</i>= n 指定第 n+1 个数据。n 是一个整数，从 0 到最大值（取决于缓冲区状态）。</p> <p><i>offset</i>= CURR 指定当前的数据位置。</p> <p><i>offset</i>= STAR 指定数据缓冲区的顶部。与偏移量=0 相同.</p>
查询响应	<p><i>size</i> 要接收的数据数。1 到最大值(取决于缓冲区状态)。参数类型为 NR1。 如果未指定此参数，则会返回来自偏移量的所有数据.</p> <p><i>response&lt;newline&gt;</i>  <i>response</i> 返回由 :FORMat:ELEMents:SENSe 命令指定的数据。          响应数据类型为 NR3。请参见“数据输出格式”。          如下示例所示，响应可能包含多个数据和元素。此示例包含由通道 1 进行的 10 步扫描测量的当前数据 (ch1curr N) 和源数据 (ch1sour N)。使用 ASCII 数据输出格式，每个数据都用逗号分隔。  <i>ch1curr1,ch1sour1,ch1curr2,ch1sour2,..... ch1curr10,ch1sour10</i>          如果测量功能未启用或不存在数据，响应将返回+9.910000E+37 (ASCII)          或 NaN (IEEE-754)。</p>
示例	<pre>:FORM:ELEM:SENS CURR,SOUR :SENS:DATA?</pre>

#### 4.10.16 :SENSe:DATA:LATest?

描述	返回最新电流测量数据、电压测量数据、seme 测量数据、电阻测量数据、源输出设置数据、状态数据或时间数据。 在执行 :INITiate、:MEASure 或 :READ 命令之前，不会清除数据.
命令	:SENSe:DATA:LATest?
参数	无
查询响应	<p><i>response&lt;newline&gt;</i>  <i>response</i> 返回由 :FORMat:ELEMents:SENSe 命令指定的最新数据。          响应数据类型为 NR3。请参见“数据输出格式”。          如下示例所示，响应可能包含多个数据元素。此示例包含通过通道 1 进行</p>

的 10 步扫描测量的最新当前数据 (ch1curr10) 和源数据 (ch1sour10)。

使用 ASCII 数据输出格式，每个数据都用逗号分隔。

*ch1curr10, ch1sour10*

如果测量功能未启用或不存在数据，响应将返回+9.910000E+37 (ASCII) 或 NaN (IEEE-754)。

---

示例	:FORM:ELEM:SENS CURR,SOUR :SENS:DATA:LAT?
----	--

---

#### 4.10.17 :SENSe:FUNCtion:OFF

描述	禁用指定的测量功能。
命令	:SENSe[c]:FUNCtion:OFF <i>function[, function[, function]]</i> :SENSe[c]:FUNCtion:OFF?
参数	<i>function</i> “CURREnt[:DC]” “VOLTage[:DC]” “RESistance” (default)。不分大小写。参数类型为 SPD。 <i>function</i> = “CURREnt[:DC]” 选择电流测量功能。 <i>function</i> = “VOLTage[:DC]” 选择电压测量功能。 <i>function</i> = “RESistance” 选择电阻测量功能。
查询响应	<i>function[, function[, function]]&lt;newline&gt;</i> <i>function</i> 返回“CURR”、“VOLT”或“RES”，并指示当前已禁用的测量功能。 如果未选择一个函数，则查询将返回“”(空字符串)。响应数据类型为 SRD。
示例	:SENS:FUNC:OFF “RES”, “VOLT” :SENS2:FUNC:OFF?

---

#### 4.10.18 :SENSe:FUNCtion:OFF:ALL

描述	禁用所有测量功能。
命令	:SENSe[c]:FUNCtion:OFF:ALL
参数	无
查询响应	无
示例	:SENS:FUNC:OFF:ALL

---

#### 4.10.19 :SENSe:FUNCtion:OFF:COUNt?

描述	查询被禁用的测量功能的数量。
命令	:SENSe[c]:FUNCtion:OFF:COUNt?

---

参数	无
查询响应	无
示例	:SENS:FUNC:OFF:COUN?

#### 4.10.20 :SENSe:FUNCtion[:ON]

描述	启用指定的测量功能
命令	:SENSe[c]:FUNCtion[:ON] <i>function[, function[, function]]</i> :SENSe[c]:FUNCtion[:ON]?
参数	<p><i>function</i> “CURRent[:DC]” “VOLTage[:DC]” “RESistance”            默认为“VOLT”,“CURR”。不分大小写。参数类型为 SPD。</p> <p><i>function</i>=“CURRent[:DC]” 选择电流测量功能。  <i>function</i>=“VOLTage[:DC]” 选择电压测量功能。  <i>function</i>=“RESistance” 选择电阻测量功能；            电阻测量模式参见命令 “:SENSe:RESistance:MODE”。</p>
查询响应	<p><i>function[, function[, function]]&lt;newline&gt;</i></p> <p><i>function</i> 返回“CURR”、“VOLT”或“RES”，并指示当前已禁用的测量功能。            如果未选择一个函数，则查询将返回“”(空字符串)。响应数据类型为 SRD。</p>
示例	:SENS:FUNC “RES”,“VOLT” :SENS2:FUNC:ON?

#### 4.10.21 :SENSe:FUNCtion[:ON]:ALL

描述	启用所有测量功能。 电阻测量功能参见命令“:SENSe:RESistance:MODE”。
命令	:SENSe[c]:FUNCtion[:ON]:ALL
参数	无
查询响应	无
示例	:SENS:FUNC:ALL

#### 4.10.22 :SENSe:FUNCtion[:ON]:COUNt?

描述	返回已启用的测量功能的数量。
命令	:SENSe[c]:FUNCtion[:ON]:COUNT?
参数	无

查询响应	无
示例	:SENS:FUNC:COUN?

#### 4.10.23 :SENSe:FUNCTION:STATE?

描述	查询指定的测量功能是否已启用或禁用。
命令	:SENSe[c]:FUNCTION:STATE? <i>function</i>
	<i>function</i> “CURRent[:DC]” “VOLTage[:DC]” “RESistance”。 参数类型为 SPD。
参数	<i>function</i> =“CURRent[:DC]” 选择电流测量功能。 <i>function</i> =“VOLTage[:DC]” 选择电压测量功能。 <i>function</i> =“RESistance” 选择电阻测量功能。
	<i>response</i> <newline>
查询响应	<i>response</i> 返回 0 或 1，表示指定的测量功能现在已禁用或启用。 响应数据类型为 NR1。
示例	:SENS:FUNC:STAT? “CURR”

#### 4.10.24 :SENSe:REMote

描述	启用或禁用遥感功能。 必须启用遥感功能，才能使用 4 线连接（开尔文连接）。
命令	:SENSe[c]:REMote <i>mode</i> :SENSe[c]:REMote?
参数	<i>mode</i> 1 ON 0 OFF (default)。参数类型为 boolean。 <i>mode</i> =0 或 OFF 禁用遥感。 <i>mode</i> =1 或 ON 启用遥感。
	<i>mode</i> <newline>
查询响应	<i>mode</i> 为 0 或 1，分别表示遥感器已关闭或开启。 响应数据类型为 NR1。
示例	:SENS:REM 1 :SENS2:REM?

#### 4.10.25 :SENSe:RESistance:MODE

描述	选择电阻测量模式。
命令	:SENSe[c]:RESistance:MODE <i>mode</i>

---

:SENSe[c]:RESistance:MODE?

*mode* MANual (default)|AUTO。参数类型为 CPD。

如果选择了 mode = MANual，则必须手动设置源和测量参数。如果启用了电阻测量功能，则通过 voltage/current 计算电阻。无法设置电阻测量量程。如果选择 mode = AUTO，如果启用电阻测量功能，通道会自动设置电流源和电压测量。当前输出值和范围取决于 :SENS:RES:RANG:UPP 或 :SENS:RES:RANG:AUTO:LLIM 和 :SENS:RES:RANG:AUTO:ULIM 命令选择的电阻测量量程。并自动设置以下参数。

**参数**

源输出模式: CURRent

电流源自动量程: OFF

源输出 shape: DC

电流源 mode: FIXed

电压测量量程: 2 V range

电压合规值: 2.1 V

电压测量自动量程: OFF

高电容模式: OFF

---

*mode*<newline>

**查询响应**

*mode* 返回 MAN 或 AUTO，并指示电阻测量模式。

响应数据类型为 CRD.

---

**示例**

:SENS:RES:MODE MAN

:SENS2:RES:MODE?

---

#### 4.10.26 :SENSe:RESistance:OCOMpensated

**描述**

启用或禁用电阻测量偏移补偿

---

**命令**

:SENSe[c]:RESistance:OCOMpensated *mode*

:SENSe[c]:RESistance:OCOMpensated?

---

**参数**

*mode* 1|ON|0|OFF (default)。参数类型为 boolean。

*mode*=0 或 OFF 禁用电阻测量偏移补偿。

*mode*=1 或 ON 实现电阻测量偏移补偿。

---

*mode*<newline>

**查询响应**

*mode* 是 0 或 1，分别表示偏移补偿电阻测量值关闭或打开。

响应数据类型为 NR1。

---

**示例**

:SENS:RES:OCOM 1

:SENS2:RES:OCOM?

---

#### 4.10.27 :SENSe:TOUTput:SIGNaL

<b>描述</b>	为 trigger 层和 acquire 设备动作之间的状态变化选择触发输出。 可以设置多个触发输出端口。
<b>命令</b>	:SENSe[c]:TOUTput:SIGNaL <i>output[1, output]</i> :SENSe[c]:TOUTput:SIGNaL?
<b>参数</b>	<i>output</i> 触发输出端口。EXT1 (default) EXT2 EXT3 EXT4 EXT5 EXT6  LAN  INT1 INT2。参数类型为 CPD。 <i>output</i> = INT1 或 INT2 分别选择内部总线 1 或 2。 <i>output</i> = LAN 选择 LAN 局域网端口。 <i>output</i> = EXTn 选择 GPIO 引脚 n, 它是后面板上的 Digital I/O D-sub 连接器的输出端口。n=1 到 6。
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回当前设置的端口, INT1、INT2、LAN 或 EXT1 ~ EXT6。 响应数据类型为 CRD。多个响应用一个逗号分隔。
<b>示例</b>	:SENS:TOUT:SIGN EXT3 :SENS2:TOUT:SIGN?

#### 4.10.28 :SENSe:TOUTput[:STATe]

<b>描述</b>	启用或禁用 trigger 层和 acquire 设备操作之间的状态更改的触发输出.
<b>命令</b>	:SENSe[c]:TOUTput[:STATe] <i>mode</i> :SENSe[c]:TOUTput[:STATe]?
<b>参数</b>	<i>mode</i> 触发输出打开或关闭。1 ON 0 OFF (default)。参数类型为 boolean <i>mode</i> = 1 或 ON 启用触发输出. <i>mode</i> = 0 或 OFF 禁用触发输出.
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回 1 或 0, 并分别指示触发器输出已打开或关闭。 响应数据类型为 NR1。
<b>示例</b>	:SENS:TOUT 1 :SENS2:TOUT:STAT?

#### 4.10.29 :SENSe:WAIT:AUTO

<b>描述</b>	启用或禁用用于计算指定通道的测量等待时间的初始等待时间。 初始等待时间由仪器自动设置, 不能更改。请参阅“:SENSe:WAIT[:STATe]”。
<b>命令</b>	:SENSe[c]:WAIT:AUTO <i>mode</i> :SENSe[c]:WAIT:AUTO?

---

<b>参数</b>	<i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean。 <i>mode</i> =1 或 ON 启用初始等待时间。 <i>mode</i> =0 或 OFF 禁用初始等待时间。初始等待时间设置为 0.
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 为 0 或 1，分别表示初始等待时间被禁用或启用。 响应数据类型为 NR1。
<b>示例</b>	:SENS:WAIT:AUTO 0 :SENS2:WAIT:AUTO?

---

#### 4.10.30 :SENSe:WAIT:GAIN

---

<b>描述</b>	设置用于计算指定通道的测量等待时间的增益值。 请参阅:SENSe:WAIT[:STATe]。
<b>命令</b>	:SENSe[c]:WAIT:GAIN <i>gain</i> :SENSe[c]:WAIT:GAIN? [ <i>gain</i> ]
<b>参数</b>	<i>gain</i> value (0 to 100) MINimum MAXimum DEFault (默认为 1)。 参数类型为 NRf。查询不支持 gain = value。
<b>查询响应</b>	<i>gain</i> <newline> <i>gain</i> 返回增益值的当前设置。如果指定了参数, <i>gain</i> 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
<b>示例</b>	:SENS:WAIT:GAIN 0.5 :SENS2:WAIT:GAIN?

---

#### 4.10.31 :SENSe:WAIT:OFFSet

---

<b>描述</b>	设置用于计算指定通道的测量等待时间的偏移值。 参见:SENSe:WAIT[:STATe].
<b>命令</b>	:SENSe[c]:WAIT:OFFSet <i>offset</i> :SENSe[c]:WAIT:OFFSet? [ <i>offset</i> ]
<b>参数</b>	<i>offset</i> value (0 到 1 秒) MINimum MAXimum DEFault (默认为 0)。 参数类型为 NRf。查询不支持 offset = value。
<b>查询响应</b>	<i>offset</i> <newline> <i>offset</i> 返回当前设置的偏移量值。如果指定了一个参数, 偏移量将返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
<b>示例</b>	:SENS:WAIT:OFFS 0.5 :SENS2:WAIT:OFFS?

---

#### 4.10.32 :SENSe:WAIT[:STATe]

描述	启用或禁用指定通道的测量等待时间。等待时间为直流输出开始或脉冲后缘后测量通道不能开始测量的时间。
命令	<pre>:SENSe[c]:WAIT[:STATe] mode :SENSe[c]:WAIT[:STATe]?</pre> <p><i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean。  <i>mode</i>=0 或 OFF 禁用测量等待时间。等待时间设置为 0。  <i>mode</i>=1 或 ON 启用由以下公式给出的测量等待时间。</p> <ul style="list-style-type: none"> <li>• :SENSe:WAIT:AUTO ON 1 condition:  <math>\text{wait time} = \text{gain} \times \text{initial wait time} + \text{offset}</math></li> <li>• :SENSe:WAIT:AUTO OFF 0 condition:  <math>\text{wait time} = \text{offset}</math></li> </ul>
参数	<p>初始等待时间由仪器自动设置，不能更改。</p> <p>增益和偏移量分别由 :SENSe:WAIT:GAIN 和 :SENSe:WAIT:OFFSet 命令设置。</p>
查询响应	<p><i>mode</i>&lt;newline&gt;</p> <p><i>mode</i> 为 0 或 1，分别表示已禁用或启用测量等待时间。</p> <p>响应数据类型为 NR1.</p>
示例	<pre>:SENS:WAIT 0 :SENS2:WAIT:STAT?</pre>

### 4.11 SOURce Subsystem

#### 4.11.1 [:SOURce]:<CURRent|VOLTage>:<CENTer|SPAN>

描述	设置电流或电压扫描输出的中心或跨度值。
命令	<pre>[:SOURce[c]]:&lt;CURRent VOLTage&gt;:&lt;CENTer SPAN&gt; <i>data</i> [:SOURce[c]]:&lt;CURRent VOLTage&gt;:&lt;CENTer SPAN&gt;? [<i>data</i>]</pre> <p>&lt;CURRent VOLTage&gt;：选择 CURRent 电流输出或 VOLTage 电压输出；  &lt;CENTer SPAN&gt;：选择 CENTer 扫描中心值或 SPAN 扫描跨度值。</p>
参数	<p><i>data</i> 扫描中心或跨度值。value (参见数据手册) MINimum MAXimum DEFault (0.0)。参数类型为 NRf+。查询不支持 <i>data</i>=value。</p> <p>中心值 <i>center</i> 和跨度值 <i>span</i> 通过以下公式计算得出，其中 <i>start</i> 和 <i>stop</i> 由[:SOURce]:&lt;CURRENT VOLTage&gt;:&lt;START STOP&gt;命令设置：</p> $\text{center} = (\text{start} + \text{stop})/2$ $\text{span} = \text{stop} - \text{start}$
查询响应	<p><i>data</i>&lt;newline&gt;</p> <p><i>data</i> 返回当前设置值。</p>

---

如果指定了一个参数，则返回分配给 DEF、MIN 或 MAX 的值。

响应数据类型为 NR3。

---

示例	:CURR:CENT 1E-3 :SOUR2:VOLT:SPAN?
----	--------------------------------------

---

#### 4.11.2 [:SOURce]:<CURRent|VOLTage>[:LEVel][:IMMEDIATE][:AMPLitude]

描述	立即更改指定通道的源输出值。
	[:SOURce[c]]:<CURRent VOLTage>[:LEVel][:IMMEDIATE][:AMPLitude] /level
命令	[:SOURce[c]]:<CURRent VOLTage>[:LEVel][:IMMEDIATE][:AMPLitude]? [/level]
	<CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。
参数	/level 电流或电压输出值。value (参见数据手册)  MINimum MAXimum  DEFault (默认为 0)。参数类型为 NRf+。查询不支持 level = value。
	/level<newline>
查询响应	/level 返回当前设置值。 如果指定了一个参数，/level 将返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR3。
示例	:VOLT 3 :SOUR2:CURR:LEV:IMM:AMPL?

---

#### 4.11.3 [:SOURce]:<CURRent|VOLTage>[:LEVel]:TRIGgered[:AMPLitude]

描述	设置触发输出值。 接收到由 :TRIGger<:ACQuire[:TRANSient[:ALL]]>:SOURce[:SIGNal] 命令 设置的触发信号时，立即更改指定通道的源输出值为触发输出值。 触发结束后的输出值由命令
	[:SOURce]:FUNCTION:TRIGgered:CONTinuous 控制。
命令	[:SOURce[c]]:<CURRent VOLTage>[:LEVel]:TRIGgered[:AMPLitude] /level [:SOURce[c]]:<CURRent VOLTage>[:LEVel]:TRIGgered[:AMPLitude]? [/level]
	<CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。
参数	/level 电流或电压输出水平。value (参见数据手册)  MINimum MAXimum  DEFault (默认为 0)。参数类型为 NRf+。查询不支持 level = value。
	level <newline>
查询响应	level 返回当前设置值。如果指定了一个参数，level 将返回分配给 DEF、 MIN 或 MAX 的值。响应数据类型为 NR3。

---

---

**示例**

```
:VOLT:TRIG 3
:SOUR2:CURR:LEV:TRIG:AMPL?
```

---

#### 4.11.4 [:SOURce]:<CURRent|VOLTage>:MODE

<b>描述</b>	选择源模式：恒定模式、列表扫描或扫描模式。
<b>命令</b>	<code>[:SOURce[c]]:&lt;CURRent VOLTage&gt;:MODE <i>mode</i></code> <code>[:SOURce[c]]:&lt;CURRent VOLTage&gt;:MODE?</code>
<b>参数</b>	<i>mode</i> 源模式。SWEep LIST FIxed (default)。参数类型为 CPD。 <i>mode</i> =FIX 设置为恒定电流或电压源。 <i>mode</i> =LIST 设置为用户指定的电流或电压列表扫描源。 <i>mode</i> =SWEep 设置为电流或电压扫描源。
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 返回 FIX, LIST, 或 SWE。响应数据类型为 CRD。
<b>示例</b>	<code>:VOLT:MODE SWE</code> <code>:SOUR2:CURR:MODE?</code>

---

#### 4.11.5 [:SOURce]:<CURRent|VOLTage>:POINts

<b>描述</b>	设置电流或电压扫描输出的扫描点数。
<b>命令</b>	<code>[:SOURce[c]]:&lt;CURRent VOLTage&gt;:POINts <i>points</i></code> <code>[:SOURce[c]]:&lt;CURRent VOLTage&gt;:POINts? [<i>points</i>]</code> <CURRent VOLTage>: 选择 CURRent 电流输出或 VOLTage 电压输出。
<b>参数</b>	<i>points</i> 扫描步骤的数量。value (可选 1-100,000)  MINimum  MAXimum  DEFault (默认为 1)。参数类型为 NRf+。查询不支持 <i>points</i> = value。 <i>points</i> 点值可以用以下公式表示，使用 [:SOURce]:<CURRent VOLTage>:STEP 设置的步进值 step 和 [:SOURce]:<CURRent VOLTage>:<CENTer SPAN> 命令设置的跨度值 span。 <i>points</i> = span /step + 1 (step 不能为 0) 当 <i>points</i> = 1 时 step = 0。 若扫描点数 <i>points</i> 被改变，则跨度 span 不变，步长 step 被改变； 若步进 step 被改变，跨度 span 不变，扫描点数 <i>points</i> 被改变； 若跨度 span 被改变，扫描点数 <i>points</i> 不变，步进 step 被改变。 计算得到的扫描点数 <i>points</i> 值将向下取整。 扫描测量的 stop 值有以下公式计算： stop = start + step × ( <i>points</i> -1) 对于 log 对数扫描，将忽略步长 step 值，并且不用于计算扫描点。

---

---

	<i>points</i> <newline>
查询响应	<i>points</i> 返回当前设置值。如果指定了一个参数, <i>points</i> 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR1。
示例	:CURR:POIN 11 :SOUR2:VOLT:POIN?

---

#### 4.11.6 [:SOURce]:<CURRent|VOLTage>:RANGe

---

描述	设置电流或电压输出量程。当自动量程功能关闭时有效。
命令	[:SOURce[c]]:<CURRent VOLTage>:RANGe range [:SOURce[c]]:<CURRent VOLTage>:RANGe? <CURRent VOLTage>: 选择 CURRent 电流输出或 VOLTage 电压输出。
参数	range value (参见数据手册)  MINimum  MAXimum DEFault。参数类型为 NRf+。 value 用于电流量程设置时。 value 用于电压量程设置时。
查询响应	range<newline> range 返回当前设置。响应数据类型为 NR3。
示例	:CURR:RANG 1E-6 :SOUR2:VOLT:RANG?

---

#### 4.11.7 [:SOURce]:<CURRent|VOLTage>:RANGe:AUTO

---

描述	打开或关闭自动量程。
命令	[:SOURce[c]]:<CURRent VOLTage>:RANGe:AUTO mode [:SOURce[c]]:<CURRent VOLTage>:RANGe:AUTO? <CURRent VOLTage>: 选择 CURRent 电流输出或 VOLTage 电压输出。
参数	mode 0 OFF 1 ON (default)。参数类型为 boolean 布尔型。 mode=0 或 OFF 关闭自动量程; 可通过[:SOURce[c]]:<CURRent VOLTage>:RANGe 命令设置源输出量程。 mode=1 或 ON 打开自动量程。机器将自动为源输出选择最佳分辨率的量程。 如果手动选择了量程, 则自动量程功能被关闭。
查询响应	mode<newline> mode 为 0 或 1, 分别表示自动量程已关闭或开启。响应数据类型为 NR1。
示例	:CURR:RANG:AUTO 0 :SOUR2:VOLT:RANG:AUTO?

---

#### 4.11.8 [:SOURce]:<CURRent|VOLTage>:RANGe:AUTO:LLIMit

描述	设置自动量程可调节范围的下限量程，也是为源输出值应用最佳分辨率可选择的最小量程。
命令	[:SOURce[c]]:<CURRent VOLTage>:RANGe:AUTO:LLIMit <i>range</i> [:SOURce[c]]:<CURRent VOLTage>:RANGe:AUTO:LLIMit? [ <i>range</i> ] <CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。
参数	<i>range</i> value (参见数据手册)  MINimum  MAXimum DEFault。参数类型为 NRf+。查询不支持 <i>range</i> = <i>value</i> 。 <i>value</i> 用于电流量程设置时。 <i>value</i> 用于电压量程设置时。
查询响应	<i>range</i> <newline> <i>range</i> 返回当前设置值。如果指定了一个参数，则 <i>range</i> 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
示例	:CURRENTRANG:AUTO:LLIM 1E-6 :SOUR2:VOLT:RANG:AUTO:LLIM?

#### 4.11.9 [:SOURce]:<CURRent|VOLTage>:RANGe:RPRiority

描述	选择输出脉冲优先模式，低噪声优先或瞬态速度优先。
命令	[:SOURce[c]]:<CURRent VOLTage>:RANGe:RPRiority <i>mode</i> [:SOURce[c]]:<CURRent VOLTage>:RANGe:RPRiority? <CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。
参数	<i>mode</i> NOISe (default) TRANsient。参数类型为 CPD。 <i>mode</i> = NOISe 设置低噪声优先级模式，获得干净的输出脉冲。 <i>mode</i> = TRANsient 设置瞬态速度优先模式，获得较高斜率的输出脉冲。 当电流输出或测量量程为 1A、1.5 A 或 3 A 量程时，该模式将有效。
查询响应	<i>mode</i> <newline> <i>mode</i> 返回 NOIS 或 TRAN。响应数据类型为 CRD。
示例	:CURRENTRANG:RPR TRAN :SOUR2:VOLT:RANG:RPR?

#### 4.11.10 [:SOURce]:<CURRent|VOLTage>:<START|STOP>

描述	设置电流或电压扫描输出的开始或停止值。
命令	[:SOURce[c]]:<CURRent VOLTage>:<START STOP> <i>data</i> [:SOURce[c]]:<CURRent VOLTage>:<START STOP>? [ <i>data</i> ] <CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。

---

	<START STOP>: 选择 START 扫描开始值, 或 STOP 扫描停止值。
参数	<p><i>data</i> 扫描开始或停止值。value (取值范围参见数据手册)  MINimum MAXimum DEFault (默认值 0.0)。参数类型为 NRf+。查询不支持 <i>data</i> = <i>value</i>。</p> <p>开始值和停止值可以用以下公式表示:</p> $\text{start} = \text{center} - \text{span}/2$ $\text{stop} = \text{center} + \text{span}/2$ <p>其中 center 为扫描中心值, span 为扫描跨度值, 由 [:SOURce[c]]:&lt;CURRent VOLTage&gt;:&lt;CENTer SPAN&gt;命令设置。</p>
查询响应	<i>data</i> <newline> <i>data</i> 返回当前设置。如果指定了一个参数, 则数据返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
示例	:VOLT:STOP 10 :SOUR2:CURR:STAR?

---

#### 4.11.11 [:SOURce]:<CURRent|VOLTage>:STEP

---

描述	设置电流或电压扫描输出的扫描步长值。
命令	<p>[:SOURce[c]]:&lt;CURRent VOLTage&gt;:STEP <i>step</i></p> <p>[:SOURce[c]]:&lt;CURRent VOLTage&gt;:STEP? [<i>step</i>]</p> <p>&lt;CURRent VOLTage&gt;: 选择 CURRent 电流输出或 VOLTage 电压输出。</p>
	<p><i>step</i> 扫描步进值。value (取值范围参见数据手册)  MINimum MAXimum DEFault (默认为 0)。参数类型为 NRf+。查询不支持 <i>step</i> = <i>value</i>。</p> <p><i>step</i> 步长值可以用以下公式表示, 使用 [:SOURce]:&lt;CURRent VOLTage&gt;: POINTs 设置的扫描点数 <i>points</i> 和 [:SOURce]:&lt;CURRent VOLTage&gt;:&lt;CENTer SPAN&gt; 命令设置的跨度值 <i>span</i>。</p> $\text{step} = \text{span}/(\text{points} - 1)$ <p>(<i>points</i> 不能为 1) 当 <i>points</i> = 1 时 <i>step</i> = 0。</p>
参数	<p>若扫描点数 <i>points</i> 被改变, 则跨度 <i>span</i> 不变, 步长 <i>step</i> 被改变;</p> <p>若步进 <i>step</i> 被改变, 跨度 <i>span</i> 不变, 扫描点数 <i>points</i> 被改变;</p> <p>若跨度 <i>span</i> 被改变, 扫描点数 <i>points</i> 不变, 步进 <i>step</i> 被改变。</p> <p>计算得到的扫描点数 <i>points</i> 值将向下取整。</p> <p>扫描测量的 <i>stop</i> 值有以下公式计算:</p> $\text{stop} = \text{start} + \text{step} \times (\text{points}-1)$ <p>对于 log 对数扫描, 将忽略步长 <i>step</i> 值, 并且不用于计算扫描点。</p> <p>步长 <i>step</i> 和跨度 <i>span</i> 的极性必须相同, 不同的极性会导致错误。</p>

---

---

	<i>step</i> <newline>
查询响应	<i>step</i> 返回当前设置值。如果指定了一个参数，则步骤返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
示例	:VOLT:STEP 0.5 :SOUR2:CURR:STEP?

---

#### 4.11.12 [:SOURce]:<CURRent|VOLTage>:TRANsient:SPEEd

---

描述	选择指定源通道的瞬态速度模式，正常或快速。
命令	[:SOURce[c]]:<CURRent VOLTage>:TRANsient:SPEEd <i>mode</i> [:SOURce[c]]:<CURRent VOLTage>:TRANsient:SPEEd? <CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。
参数	<i>mode</i> NORMAl (default) FAST。参数类型为 CPD。 <i>mode</i> =NORMAl 设置正常模式，以获得干净的输出。 <i>mode</i> =FAST 设置快速模式，以获得较高的输出变化率。当启用高电容模式时，此模式无效。
查询响应	<i>mode</i> <newline> <i>mode</i> 返回 NORM 或 FAST. 响应数据类型为 CRD.
示例	:CURREN:TRAN:SPE FAST :SOUR2:VOLT:TRAN:SPE?

---

#### 4.11.13 [:SOURce]:DIGItal:DATA

---

描述	读写 GPIO(数字控制端口)引脚数据。
命令	[:SOURce]:DIGItal:DATA <i>data</i> [:SOURce]:DIGItal:DATA?
参数	<i>data</i> 输出数据。value (0 to 63) (默认值为 0)。参数类型为 NR1。
查询响应	<i>data</i> <newline> <i>data</i> 返回从 GPIO 引脚读取的数据。 响应数据类型是由:FORMAT:DIGItal 命令选择的 NR1 或 NDN。
示例	:DIG:DATA 31 :SOUR:DIG:DATA?

---

#### 4.11.14 [:SOURce]:DIGItal:EXTernal:FUNCTION

---

描述	设置 GPIO(数字控制端口)引脚的输入/输出功能。
命令	[:SOURce]:DIGItal:EXTernal[n][.:FUNCTION] <i>function</i>

---

---

	<code>[:SOURce]:DIGItal:EXTernal[n][:FUNCTION]?</code>
	<i>function</i> 引脚功能。DINPut DIO TINPut TOUT。参数类型为 CPD。
参数	<i>function</i> =DINP 数字输入。
	<i>function</i> =DIO 数字 I/O。
	<i>function</i> =TINP 触发输入。
	<i>function</i> =TOUT 触发输出。
	<b>注意：</b> EXT1 到 EXT6 为输出引脚，EXT7 到 EXT12 为输出引脚。
查询响应	<i>function</i> <newline> <i>function</i> 返回 DIO, DINP, TOUT, 或 TINP。响应数据类型为 CRD。
示例	<code>:DIG:EXT TOUT</code> <code>:SOUR:DIG:EXT2:FUNC?</code>

---

#### 4.11.15 [:SOURce]:DIGItal:EXTernal:POLarity

---

描述	设置 GPIO(数字控制端口)引脚的输入/输出极性。 输入/输出功能由指令[:SOURce]:DIGItal:EXTernal:FUNCTION 设置。
命令	<code>[:SOURce]:DIGItal:EXTernal[n]:POLarity <i>polarity</i></code> <code>[:SOURce]:DIGItal:EXTernal[n]:POLarity?</code>
参数	<i>polarity</i> 输入/输出极性。NEG POS。参数类型为 CPD。 <i>polarity</i> =POS 正极性。 <i>polarity</i> =NEG 负极性。
查询响应	<i>polarity</i> <newline> <i>polarity</i> 返回 POS 或 NEG。响应数据类型为 CRD。
示例	<code>:DIG:EXT:POL NEG</code> <code>:SOUR:DIG:EXT6:POL?</code>

---

#### 4.11.16 [:SOURce]:DIGItal:EXTernal:TOUTput[:EDGE]:POSIon

---

描述	为指定的 GPIO 引脚选择触发信号输出时机。
命令	<code>[:SOURce]:DIGItal:EXTernal[n]:TOUTput[:EDGE]:POSIon <i>position</i></code> <code>[:SOURce]:DIGItal:EXTernal[n]:TOUTput[:EDGE]:POSIon?</code>
参数	<i>position</i> 触发信号输出的时机。BEFore AFTer BOTH (default)。参数类型为 CPD。 <i>type</i> =BEFor 在 arm, trigger 和 device actions (transient 或 acquire) 开始时输出触发信号。 <i>type</i> =AFTer 在 arm, trigger 和 device actions (transient 或 acquire) 结束时输出触发信号。 <i>type</i> =BOTH 在 arm, trigger 和 device actions (transient 或 acquire)

---

---

开始和结束时输出触发信号.

---

*response*<newline>

**查询响应** *response* 返回当前触发信号输出时机, BEF, AFT 或 BOTH。  
响应数据类型为 CRD.

---

**示例** :DIG:EXT:TOUT:POS BEF  
:SOUR:DIG:EXT2:TOUT:POS?

---

#### 4.11.17 [:SOURce]:DIGital:EXTernal:TOUTput[:EDGE]:WIDTh

<b>描述</b>	为指定的 GPIO 引脚设置触发信号输出的脉冲宽度。
<b>命令</b>	[:SOURce]:DIGital:EXTernal[n]:TOUTput[:EDGE]:WIDTh <i>width</i> [:SOURce]:DIGital:EXTernal[n]:TOUTput[:EDGE]:WIDTh? [ <i>width</i> ]
<b>参数</b>	<i>width</i> 脉宽。value (1E-5 到 1E-2 秒) MINimum MAXimum DEFault (默认 0.1 ms)。参数类型为 NRf+。查询不支持 <i>width</i> = <i>value</i> .
	<i>width</i> <newline>
<b>查询响应</b>	<i>width</i> 返回当前设置。如果指定了一个参数, <i>width</i> 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
<b>示例</b>	:DIG:EXT:TOUT:WIDT 1E-5 :SOUR:DIG:EXT3:TOUT:WIDT?

---

#### 4.11.18 [:SOURce]:DIGital:EXTernal:TOUTput:TYPE

<b>描述</b>	为指定的 GPIO 引脚选择触发输出信号类型.
<b>命令</b>	[:SOURce]:DIGital:EXTernal[n]:TOUTput:TYPE <i>type</i> [:SOURce]:DIGital:EXTernal[n]:TOUTput:TYPE?
<b>参数</b>	<i>type</i> 触发信号类型。EDGE (default) LEVel。参数类型为 CPD。 <i>type</i> = EDGE 输出边沿信号。 <i>type</i> = LEVel 输出电平信号。
	<i>response</i> <newline>
<b>查询响应</b>	<i>response</i> 返回当前触发信号类型, EDGE 或 LEV。 响应数据类型为 CRD。
<b>示例</b>	:DIG:EXT:TOUT:TYPE EDGE :SOUR:DIG:EXT4:TOUT:TYPE?

---

#### 4.11.19 [:SOURce]:DIGital:INTernal:TOUTput[:EDGE]:POSIon

<b>描述</b>	为内部端口 1 或 2 选择的触发信号输出的时机.
-----------	---------------------------

---

---

命令	[:SOURce]:DIGItal:INTernal[i]:TOUTput[:EDGE]:POSition position [:SOURce]:DIGItal:INTernal[i]:TOUTput[:EDGE]:POSition?
参数	<p><i>position</i> 触发信号输出的时机。BEFore AFTer BOTH (default)。参数类型为 CPD。</p> <p><i>type</i>=BEFor 在 arm, trigger 和 device actions (transient 或 acquire) 开始时输出触发信号。</p> <p><i>type</i>=AFTer 在 arm, trigger 和 device actions (transient 或 acquire) 结束时输出触发信号。</p> <p><i>type</i>=BOTH 在 arm, trigger 和 device actions (transient 或 acquire) 开始和结束时输出触发信号。</p>
查询响应	<i>response</i> <newline> <i>response</i> 返回当前设置触发输出时机。BEF, AFT 或 BOTH。响应数据类型为 CRD。
示例	:DIG:INT2:TOUT:POS BEF :SOUR:DIG:INT2:TOUT:POS?

---

#### 4.11.20 [:SOURce]:FUNCTION:MODE

---

描述	选择源输出模式。
命令	[:SOURce[c]]:FUNCTION:MODE <i>mode</i> [:SOURce[c]]:FUNCTION:MODE?
参数	<p><i>mode</i> 源输出模式。CURRent VOLTage (default)。参数类型为 CPD。</p> <p><i>mode</i>=CURR 电流源模式；电流源的电压合规性由 :SENS:VOLT:PROT[:LEV] 命令设置。</p> <p><i>mode</i>=VOLT 电压源模式；电压源的电流合规性由 :SENS:CURR:PROT[:LEV] 命令设置。</p> <p>详细参见 “:SENSe:&lt;CURRent[:DC] VOLTage[:DC]&gt;:PROTection[:LEVel][:BOTH]”命令。</p>
查询响应	<i>mode</i> <newline> <i>mode</i> 返回 CURR 或 VOLT。响应数据类型为 CRD。
示例	:FUNC:MODE CURR :SOUR2:FUNC:MODE?

---

#### 4.11.21 [:SOURce]:FUNCTION[:SHAPe]

---

描述	设置源输出形式。
----	----------

---

<b>命令</b>	<code>[:SOURce[c]]:FUNCTION[:SHAPe] <i>shape</i></code> <code>[:SOURce[c]]:FUNCTION[:SHAPe]?</code>
<b>参数</b>	<i>shape</i> 源输出形式。PULSe DC (default)。参数类型为 CPD。 <i>shape</i> = DC 直流(常量)输出。 <i>shape</i> = PULS 脉冲输出。
<b>查询响应</b>	<code>shape &lt;newline&gt;</code> <i>shape</i> 返回 DC 或 PULS。 响应数据类型为 CRD。
<b>示例</b>	<code>:FUNC PULS</code> <code>:SOUR2:FUNC:SHAP?</code>

#### 4.11.22 [:SOURce]:FUNCTION:TRIGgered:CONTinuous

<b>描述</b>	启用或禁用指定通道的连续触发输出.
<b>命令</b>	<code>[:SOURce[c]]:FUNCTION:TRIGgered:CONTinuous <i>mode</i></code> <code>[:SOURce[c]]:FUNCTION:TRIGgered:CONTinuous?</code>
<b>参数</b>	<i>mode</i> 0 OFF (default) 1 ON。参数类型为 boolean 布尔型。 <i>mode</i> =1 或 ON 使能连续触发输出。当触发系统由忙碌变为空闲时，源的立即输出值和量程变为最后的触发输出的值和量程。 <i>mode</i> =0 或 OFF 禁用连续触发输出。当触发系统由忙碌变为空闲时，源输出值和量程恢复到由 [SOURce]:<CURRent VOLTage>[:LEVel][:IMMediate][:AMPLitude] 命令和量程设置命令设置的值。
<b>查询响应</b>	<code>mode &lt;newline&gt;</code> <i>mode</i> 返回 0 或 1，分别表示连续触发已关闭或开启。 响应数据类型为 NR1。
<b>示例</b>	<code>:FUNC:TRIG:CONT 0</code> <code>:SOUR2:FUNC:TRIG:CONT?</code>

#### 4.11.23 [:SOURce]:LIST:<CURRent|VOLTage>

<b>描述</b>	设置指定通道的列表扫描输出（电流或电压）数据.
<b>命令</b>	<code>[:SOURce[c]]:LIST:&lt;CURRent VOLTage&gt; /list</code> <code>[:SOURce[c]]:LIST:&lt;CURRent VOLTage&gt;?</code> <CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。
<b>参数</b>	<i>list</i> 输出数据的列表。默认为 0。参数类型为 NRf。 <i>list</i> 最多可以添加最大的扫描步数的数据点 (可选 1-100,000)。每个数据都必须用一个逗号分隔，示例：list = 0.1,0.2,0.3。

---

对于输出电流或电压数据的有效数值，参见数据手册。

---

*/list<newline>*

**查询响应** */list* 返回当前设置的数据列表；  
多个数据用逗号分隔，响应数据类型为 NR3。

---

**示例** :LIST:VOLT 0.1,0.2,0.3  
:SOUR2:LIST:CURR?

---

#### 4.11.24 [:SOURce]:LIST:<CURRent|VOLTage>:APPend

---

将列表扫描输出（电流或电压）数据添加到由

**描述** [:SOURce]:LIST:<CURRent|VOLTage>命令设置的列表的结尾。  
最多可以添加的数据数量等于支持的最大的扫描步数（可选 1-100,000）。

---

**命令** [:SOURce[c]]:LIST:<CURRent|VOLTage>:APPend *append\_list*

<CURRent|VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。

---

*append\_list* List of the output data。参数类型为 NRf+。

**参数** *append\_list* 支持多个数据。  
每个数据都必须用一个逗号分隔，例如：append\_list = 1.1,1.2,1.3。  
输出电流或电压数据的有效数值，参见数据手册。

---

**查询响应** 无

---

**示例** :LIST:VOLT:APP 1.1,1.2,1.3  
:SOUR2:LIST:CURR:APP 1E-6,2E-6,3E-6

---

#### 4.11.25 [:SOURce]:LIST:<CURRent|VOLTage>:POINts?

---

查询 List 扫描数据的点数；

**描述** list 扫描数据由[:SOURce]:LIST:<CURRent|VOLTage>命令或  
[:SOURce]:LIST:<CURRent|VOLTage>:APPend 命令添加。

---

**命令** [:SOURce[c]]:LIST:<CURRent|VOLTage>:POINts?

<CURRent|VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。

---

**参数** 无

---

**查询响应** *number\_of\_data<newline>*  
*number\_of\_data* 返回 list 列表的点数。响应数据类型为 NR1。

---

**示例** :LIST:VOLT:POIN?  
:SOUR2:LIST:CURR:POIN?

---

#### 4.11.26 [:SOURce]:LIST:<CURRent|VOLTage>:STARt

<b>描述</b>	设置 list 扫描的起点。
<b>命令</b>	[:SOURce[c]]:LIST:<CURRent VOLTage>:STARtstart [:SOURce[c]]:LIST:<CURRent VOLTage>:STARt?
	<CURRent VOLTage>：选择 CURRent 电流输出或 VOLTage 电压输出。
<b>参数</b>	start 列表索引。1 到最大扫描步数 (可选 1-100,000)。其他索引值会导致错误。默认值为 1。 start = 1 表示列表中的第一个数据 (列表的顶部)。参数类型为 NR1。
<b>查询响应</b>	start <newline> start 返回当前设置 list 扫描起始点索引。响应数据类型为 NR1.
<b>示例</b>	:LIST:VOLT:STAR 10 :SOUR2:LIST:CURR:STAR?

#### 4.11.27 [:SOURce]:PULSe:DELy

<b>描述</b>	设置指定通道的脉冲延迟时间。脉冲延迟时间指从启动脉冲基础输出到启动脉冲峰值输出的时间。
<b>命令</b>	[:SOURce[c]]:PULSe:DELy delay [:SOURce[c]]:PULSe:DELy? [delay]
<b>参数</b>	delay 时延. value (0.0 到 99999.9 秒) MINimum MAXimum DEFault (0). 参数类型为 NRf+。查询不支持 delay = value。
<b>查询响应</b>	delay<newline> delay 返回当前设置。如果指定了一个参数，delay 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
<b>示例</b>	:PULS:DEL 1E-3 :SOUR2:PULS:DEL?

#### 4.11.28 [:SOURce]:PULSe:WIDTh

<b>描述</b>	设置指定通道的脉冲宽度。脉冲宽度是从开始脉冲峰值输出 (或开始脉冲电平转换) 到脉冲峰值输出结束的时间。被严格定义为从 10 % 峰值上升到 90 % 峰值的时间。
<b>命令</b>	[:SOURce[c]]:PULSe:WIDTh width [:SOURce[c]]:PULSe:WIDTh? [width]
<b>参数</b>	width 脉宽. value (5E-5 到 100000 秒, 分辨率 1E-6 秒) MINimum MAXimum DEFault (默认为 5E-5)。参数类型为 NRf+。查询不支持 width = value。

---

	脉冲的最小脉宽为 50μs。最小脉冲周期为 100μs。
查询响应	<i>width</i> <newline> <i>width</i> 返回当前设置。如果指定了一个参数, <i>width</i> 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
示例	:PULS:WIDT 2E-2 :SOUR2:PULS:WIDT?

---

#### 4.11.29 [:SOURce]:SWEEp:DIRECTION

---

描述	设置指定通道的扫描方向, UP 或 DOWN.
命令	[:SOURce[c]]:SWEEp:DIRECTION <i>direction</i> [:SOURce[c]]:SWEEp:DIRECTION?
参数	<i>direction</i> 扫描方向。DOWN UP (default)。参数类型为 CPD。 <i>direction</i> = UP 扫描方向从 <i>start</i> 开始值到 <i>stop</i> 停止值。即使指定的 <i>stop</i> 停止值不满足, 也会从以下公式给出的起始值到停止值进行扫描测量。 $stop = start + step \times (points - 1)$ <i>direction</i> = DOWN 扫描方向从 <i>stop</i> 停止值到 <i>start</i> 开始值。即使指定的 <i>start</i> 开始值不满足, 也会从以下公式给出的停止值到开始值进行扫描测量。 $start = stop - step \times (points - 1)$
查询响应	<i>direction</i> <newline> <i>direction</i> 返回当前设置的扫描方向, UP 或 DOWN. 响应数据类型为 CRD.
示例	:SWE:DIR DOWN :SOUR2:SWE:DIR?

---

#### 4.11.30 [:SOURce]:SWEEp:POINTS

---

描述	设置扫描步骤的数量. 此命令设置对电流扫描和电压扫描都有效。
命令	[:SOURce[c]]:SWEEp:POINTS <i>points</i> [:SOURce[c]]:SWEEp:POINTS? MINimum MAXimum DEFault
参数	<i>points</i> 扫描步骤数。value (可选 1-100,000)  MINimum MAXimum DEFault (默认为 1)。参数类型为 NRf+。  <i>points</i> 点值可以用以下公式表示, 使用[:SOURce]:<CURRent VOLTage>:STEP 设置的步进值 <i>step</i> 和[:SOURce]:<CURRent VOLTage>:<CENTer SPAN>命令设置的跨度值 <i>span</i> 。 $points = span / step + 1$ ( <i>step</i> 不能为 0)

---

当 *points* = 1 时 *step* = 0。  
 若扫描点数 *points* 被改变，则跨度 *span* 不变，步长 *step* 被改变；  
 若步进 *step* 被改变，跨度 *span* 不变，扫描点数 *points* 被改变；  
 若跨度 *span* 被改变，扫描点数 *points* 不变，步进 *step* 被改变。  
 计算得到的扫描点数 *points* 值将向下取整。  
 扫描测量的 *stop* 值有以下公式计算：  

$$\text{stop} = \text{start} + \text{step} \times (\text{points} - 1)$$
  
 对于 log 对数扫描，将忽略步长 *step* 值，并且不用于计算扫描点。

---

查询响应	<i>points</i> <newline> <i>points</i> 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR1。
示例	:SWE:POIN 11 :SOUR2:SWE:POIN? MAX

---

#### 4.11.31 [:SOURce]:SWEep:RANGing

---

描述	设置扫描输出的输出量程模式 .
命令	[:SOURce[c]]:SWEep:RANGing mode [:SOURce[c]]:SWEep:RANGing?
参数	<i>mode</i> 量程模式。BEST (default) FIXed AUTO。参数类型为 CPD。 <i>mode</i> = BEST，通道自动设置覆盖线性扫描的整个扫描输出电平的量程 (SPACing mode = LINear)，为对数扫描的每一步应用源输出提供最佳分辨率的量程(SPACing mode = LOGarithmic). <i>mode</i> = AUTO，为每一步扫描输出值设置能提供最佳分辨率的量程。 <i>mode</i> = FIX，仅使用扫描开始时的有效量程，扫描过程中量程不变。
查询响应	<i>mode</i> <newline> <i>mode</i> 返回当前设置的量程模式，BEST，FIX 或 AUTO。 响应数据类型为 CRD。
示例	:SWE:RANG BEST :SOUR2:SWE:RANG?

---

#### 4.11.32 [:SOURce]:SWEep:SPACing

---

描述	选择扫描输出的比例.
命令	[:SOURce[d]]:SWEep:SPACing mode [:SOURce[d]]:SWEep:SPACing?
参数	<i>mode</i> 扫描比例。LOGarithmic LINear (default)。参数类型为 CPD。 <i>mode</i> = LIN 线性比例扫描输出。 <i>mode</i> = LOG 对数尺度扫描输出。此模式下，扫描步长值将被忽略。

---

---

查询响应	<i>mode</i> <newline> <i>mode</i> 返回当前设置扫描输出比例, LIN 或 LOG。响应数据类型为 CRD。
示例	:SWE:SPAC LOG :SOUR2:SWE:SPAC?

---

#### 4.11.33 [:SOURce]:SWEP:STAir

---

描述	设置扫描模式.
命令	[:SOURce[c]]:SWEP:STAir mode [:SOURce[c]]:SWEP:STAir?
参数	<i>mode</i> 扫描模式。SINGle (default) DOUBlE。参数类型为 CPD。 <i>mode</i> =SINGle 单扫描模式. <i>mode</i> =DOUBlE 双扫描模式。双扫描执行从开始到停止再到开始的扫描。
查询响应	<i>mode</i> <newline> <i>mode</i> 返回 SING 或 DOUB, 分别表示单扫描或双扫描模式。 响应数据类型为 CRD。
示例	:SWEP:STA DOUB :SOUR2:SWEP:STA?

---

#### 4.11.34 [:SOURce]:TOUTput:SIGNAl

---

描述	设置 trigger 层和 transient 设备操作 action 之间的状态变化的触发输出信号。可以设置多个触发输出端口。
命令	[:SOURce[c]]:TOUTput:SIGNAl output{,output} [:SOURce[c]]:TOUTput:SIGNAl?
参数	<i>output</i> 触发输出端口。EXT1 (default) EXT2 EXT3 EXT4 EXT5  EXT6   LAN INT1 INT2, 参数类型为 CPD。 <i>output</i> = INT1 或 INT2 分别选择内部总线 1 或 2。 <i>output</i> = LAN 选择 LAN 局域网端口。 <i>output</i> = EXTn 选择 GPIO 引脚 n, 它是后面板上的 Digital I/O D-sub 连接器的输出端口。n=1 到 6。
查询响应	<i>response</i> <newline> <i>response</i> 返回当前设置, INT1, INT2, LAN, 或 EXT1 ~EXT 6 。 响应数据类型为 CRD。多个响应用逗号隔开。
示例	:TOUT:SIGN EXT3 :SOUR2:TOUT:SIGN?

---

#### 4.11.35 [:SOURce]:TOUTput[:STATe]

<b>描述</b>	启用或禁用在 trigger 层和 transient 设备操作 action 之间的状态更改的触发输出。
<b>命令</b>	[:SOURce[c]]:TOUTput[:STATe] mode [:SOURce[c]]:TOUTput[:STATe]?
<b>参数</b>	<i>mode</i> 触发输出打开或关闭。1 ON 0 OFF(default)。参数类型为 boolean。 <i>mode</i> =1 或 ON 使能触发输出。 <i>mode</i> =0 或 OFF 禁用触发输出。
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回 1 或 0，表示触发输出打开或关闭； 响应数据类型为 NR1.
<b>示例</b>	:TOUT 1 :SOUR2:TOUT:STAT?

#### 4.11.36 [:SOURce]:WAIT:AUTO

<b>描述</b>	启用或禁用用于计算源等待时间的初始等待时间。初始等待时间由仪器自动设置，不能更改。参见“[:SOURce]:WAIT[:STATe]”。
<b>命令</b>	[:SOURce[c]]:WAIT:AUTO mode [:SOURce[c]]:WAIT:AUTO?
<b>参数</b>	<i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean。 <i>mode</i> =1 或 ON 启用初始等待时间。 <i>mode</i> =0 或 OFF 禁用初始等待时间。初始等待时间设置为 0.
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 为 0 或 1，分别表示禁用或启用了初始等待时间。 响应数据类型为 NR1.
<b>示例</b>	:WAIT:AUTO 0 :SOUR2:WAIT:AUTO?

#### 4.11.37 [:SOURce]:WAIT:GAIN

<b>描述</b>	设置用于计算源等待时间的增益值。参见“[:SOURce]:WAIT[:STATe]”。
<b>命令</b>	[:SOURce[c]]:WAIT:GAIN <i>gain</i> [:SOURce[c]]:WAIT:GAIN? [ <i>gain</i> ]
<b>参数</b>	<i>gain</i> value (0 到 100) MINimum MAXimum DEFault (默认为 1)。 参数类型为 NRf。查询不支持 <i>gain</i> = <i>value</i> 。
<b>查询响应</b>	<i>gain</i> <newline>

---

*gain* 返回当前设置增益值。如果指定了一个参数, *gain* 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。

---

**示例**

```
:WAIT:GAIN 0.5
:SOUR2:WAIT:GAIN?
```

---

#### 4.11.38 [:SOURce]:WAIT:OFFSet

<b>描述</b>	设置用于计算源等待时间的偏移量值。参见命令“[:SOURce]:WAIT[:STATe]”。
<b>命令</b>	<code>[:SOURce[c]]:WAIT:OFFSet offset</code> <code>[:SOURce[c]]:WAIT:OFFSet? [offset]</code>
<b>参数</b>	<i>offset</i> value (0 to 1 seconds) MINimum MAXimum DEFault (默认为 0)。 参数类型为 NRf。查询不支持 <i>offset</i> = value。
<b>查询响应</b>	<i>offset</i> <newline> <i>offset</i> 返回当前设置偏移量。如果指定了一个参数, <i>offset</i> 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。
<b>示例</b>	<pre>:WAIT:OFFS 0.5 :SOUR2:WAIT:OFFS?</pre>

---

#### 4.11.39 [:SOURce]:WAIT[:STATe]

<b>描述</b>	启用或禁用指定通道的源等待时间。这个等待时间被定义为源通道在直流输出开始或脉冲的后沿后不能改变输出的时间。
<b>命令</b>	<code>[:SOURce[c]]:WAIT[:STATe] mode</code> <code>[:SOURce[c]]:WAIT[:STATe]?</code>
<b>参数</b>	<i>mode</i> 0 OFF 1 ON (default)。参数类型为 boolean。 <i>mode</i> =0 或 OFF 禁用源等待时间。等待时间设置为 0。 <i>mode</i> =1 或 ON 启用由以下公式给出的源等待时间。 [:SOURce]:WAIT:AUTO ON 1 时: wait time = <i>gain</i> × initial wait time + <i>offset</i> [:SOURce]:WAIT:AUTO OFF 0 时: wait time = <i>offset</i> 初始等待时间由仪器自动设置, 不能更改。 <i>gain</i> 和 <i>offset</i> 分别由命令 [:SOURce]:WAIT:GAIN 和 [:SOURce]:WAIT:OFFSet 设置。
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 为 0 或 1, 分别表示源等待时间已被禁用或启用。 响应数据类型为 NR1。

---

---

示例	:WAIT 0 :WAIT?
----	-------------------

---

## 4.12 STATus Subsystem

### 4.12.1 :STATus:<MEASurement|OPERation|QUEStionable>:CONDition?

---

描述	返回测量、操作或查询状态条件寄存器的值。位定义见表 4-3 至表 4-5。 此命令不会更改寄存器设置。
命令	:STATus:<MEASurement OPERation QUEStionable>:CONDition? 对于<MEASurement OPERation QUEStionable>: MEASurement: 指定测量状态条件寄存器; OPERation: 指定操作状态条件寄存器; QUEStionable: 指定查询状态条件寄存器。
参数	无
查询响应	value <newline> 返回指定寄存器的值，响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制)，由:FORMat:SREGister 命令选择。
示例	:STAT:MEAS:COND? :STAT:OPER:COND? :STAT:QUES:COND?

---

表 4-3 查询状态条件寄存器各比特位的描述和定义

比特位	十进制值	描述	定义
0	1	电压概要	通道 1 或/和 2 过压
1	2	电流概要	通道 1 或/和 2 过流
2	4	通道 1 输出保护	指定通道的输出继电器通过合规性自动输出关断功能打开
3	8	通道 2 输出保护	
4	16	温度概要	通道 1 或/和 2 过温
5 to 7		未使用	返回 0
8	256	校准	通道 1 或/和 2 校准失败
9	512	自测试	通道 1 或/和 2 自测试失败
10	1024	联锁	联锁端子打开
11	2048	通道 1 (Trans) 触发事件丢失	通道 1 的瞬时 (Trans) 操作时 Arm 层或 Trigger 层丢失触发事件

12	4096	通道 1 (Acq) 触发事件丢失	通道 1 的采集 (Acq) 操作时 Arm 层或 Trigger 层丢失触发事件
13	8192	通道 2 (Trans) 触发事件丢失	通道 2 的瞬时 (Trans) 操作时 Arm 层或 Trigger 层丢失触发事件
14	16384	通道 2 (Acq) 触发事件丢失	通道 2 的采集 (Acq) 操作时 Arm 层或 Trigger 层丢失触发事件
15		未使用	返回 0

表 4-4 测量状态条件寄存器各比特位的描述和定义

比特位	十进制值	描述	定义
0	1	通道 1 限值测试概要	通道 1 存在限值测试不通过项
1	2	通道 1 测量回读可用	通道 1 测量回读正常
2	4	通道 1 测量回读数据超量程	通道 1 的读数超出当前测量范围
3	8	通道 1 迹线缓冲区可用	通道 1 迹线缓冲区存在数据
4	16	通道 1 迹线缓冲区已满	通道 1 迹线缓冲区数据已满
5	32	未使用	返回 0
6	64	通道 2 限值测试概要	通道 2 存在限值测试不通过项
7	128	通道 2 测量回读可用	通道 2 测量回读正常
8	256	通道 2 测量回读数据超量程	通道 2 的读数超出当前测量范围
9	512	通道 2 迹线缓冲区可用	通道 2 迹线缓冲区存在数据
10	1024	通道 2 迹线缓冲区已满	通道 2 迹线缓冲区数据已满
11 to 15		未使用	返回 0

表 4-5 操作状态条件寄存器各比特位的描述和定义

比特位	十进制值	描述	定义
0	1	校准/自测试进行中	自校准/自测试正在运行
1	2	通道 1 瞬时 (Trans) 操作空闲	通道 1 瞬时 (Trans) 操作处于空闲层
2	4	通道 1 瞬时 (Trans) 操作于 Trigger 层等待	通道 1 瞬时 (Trans) 操作 Trigger 层等待输入触发源信号
3	8	通道 1 瞬时 (Trans) 操作于 Arm 层等待	通道 1 瞬时 (Trans) 操作 Arm 层等待输入触发源信号
4	16	通道 1 采集 (Acq) 操作空闲	通道 1 采集 (Acq) 操作处于空闲层

5	32	通道 1 采集 (Acq) 操作于 Trigger 层等待	通道 1 采集 (Acq) 操作于 Trigger 层等待输入触发源信号
6	64	通道 1 采集 (Acq) 操作于 Arm 层等待	通道 1 采集 (Acq) 操作于 Arm 层等待输入触发源信号
7	128	通道 2 瞬时 (Trans) 操作空闲	通道 2 瞬时 (Trans) 操作处于空闲层
8	256	通道 2 瞬时 (Trans) 操作于 Trigger 层等待	通道 2 瞬时 (Trans) 操作于 Trigger 层等待输入触发源信号
9	512	通道 2 瞬时 (Trans) 操作于 Arm 层等待	通道 2 瞬时 (Trans) 操作于 Arm 层等待输入触发源信号
10	1024	通道 2 采集 (Acq) 操作空闲	通道 2 采集 (Acq) 操作处于空闲层
11	2048	通道 2 采集 (Acq) 操作于 Trigger 层等待	通道 2 采集 (Acq) 操作于 Trigger 层等待输入触发源信号
12	4096	通道 2 采集 (Acq) 操作于 Arm 层等待	通道 2 采集 (Acq) 操作于 Arm 层等待输入触发源信号
13	8192	仪器锁定	当仪器被远程控制 (如 GPIB、USB 或 LAN) 且锁定时设置为 1; 在未被锁定时设置为 0
14	16384	程序进行中	程序正在运行; 若程序内存执行停止, 则设置为 0
15	32768	未使用	返回 0

#### 4.12.2 :STATus:<MEASurement|OPERation|QUESTIONable>:ENABLE

描述	设置测量、操作或可疑状态使能寄存器。启用寄存器是一个掩码，允许在摘要位中报告事件寄存器中的真实情况。
命令	:STATus:<MEASurement OPERation QUESTIONable>:ENABLE mask :STATus:<MEASurement OPERation QUESTIONable>:ENABLE? 对于<MEASurement OPERation QUESTIONable>: MEASurement: 指定测量状态使能寄存器; OPERation: 指定操作状态使能寄存器; QUESTIONable: 指定查询状态使能寄存器。
参数	mask 掩码, 0 到 65535 (十进制). 默认为 0。参数类型为 NR1 或 NDN。 mask 掩码是集合比特的二进制加权值之和。
查询响应	mask<newline> mask 返回指定启用寄存器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制), 由: FORMat:SREGister 命令选择。
示例	:STAT:MEAS:ENAB 65535

---

:STAT:QUES:ENAB?

---

#### 4.12.3 :STATus:<MEASurement|OPERation|QUESTIONable>[:EVENT]?

<b>描述</b>	返回测量、操作或可疑状态事件寄存器的值。此命令会更改寄存器设置。
	:STATus:<MEASurement OPERation QUESTIONable>[:EVENT]?
<b>命令</b>	对于<MEASurement OPERation QUESTIONable>: MEASurement: 指定测量状态事件寄存器; OPERation: 指定操作状态事件寄存器; QUESTIONable: 指定查询状态事件寄存器。
<b>参数</b>	无
<b>查询响应</b>	<i>value</i> <newline> <i>value</i> 返回指定事件寄存器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制), 由 :FORMAT:SREGister 命令选择。
<b>示例</b>	:STAT:MEAS:EVEN? :STAT:OPER:EVEN? :STAT:QUES:EVEN?

---

#### 4.12.4 :STATus:<MEASurement|OPERation|QUESTIONable>:NTRansition

<b>描述</b>	在测量、操作或可疑状态寄存器中设置负转换过滤器。如果设置了筛选器的一个位，则其寄存器位的 1 到 0 转换将设置事件寄存器的相应位。
	:STATus:<MEASurement OPERation QUESTIONable>:NTRansition filter :STATus:<MEASurement OPERation QUESTIONable>:NTRansition?
<b>命令</b>	对于<MEASurement OPERation QUESTIONable>: MEASurement: 指定测量状态寄存器; OPERation: 指定操作状态寄存器; QUESTIONable: 指定查询状态寄存器。
<b>参数</b>	<i>filter</i> 负转换滤波器, 0 到 65535(十进制), 默认值为 0。参数类型为 NR1 或 NDN。 <i>filter</i> 负转换滤波器是集合比特的二进制加权值之和。
<b>查询响应</b>	<i>filter</i> <newline> <i>filter</i> 返回指定寄存器的负转换滤波器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制), 由 :FORMAT:SREGister 命令选择。
<b>示例</b>	:STAT:MEAS:NTR 0 :STAT:QUES:NTR?

---

#### 4.12.5 :STATus:<MEASurement|OPERation|QUEStionable>:PTRansition

描述	在测量、操作或可疑状态寄存器中设置正转换过滤器。如果设置了筛选器的一个位，则其寄存器位的 0 到 1 转换将设置事件寄存器的相应位。
命令	:STATus:<MEASurement OPERation QUEStionable>:PTRansition filter :STATus:<MEASurement OPERation QUEStionable>:PTRansition? 对于<MEASurement OPERation QUEStionable>: MEASurement: 指定测量状态寄存器; OPERation: 指定操作状态寄存器; QUEStionable: 指定查询状态寄存器
参数	filter 正转换滤波器, 0 到 65535 (十进制), 默认值为 0。参数类型为 NR1 或 NDN。 <i>Filter</i> 正转换滤波器是集合比特的二进制加权值之和。
查询响应	filter<newline> filter 返回指定寄存器的正转换滤波器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制), 由 :FORMAT:SREGister 命令选择。
示例	:STAT:MEAS:PTR 32767 :STAT:QUES:PTR?

#### 4.12.6 :STATus:PRESet

描述	置位状态系统 PTR 寄存器中的所有定义位, 清除 NTR 和 Enable 寄存器中的所有位。寄存器将恢复到默认状态。
命令	:STATus:PRESet
参数	无
查询响应	无
示例	:STAT:PRES

#### 4.12.7 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:CONDition?

描述	返回查询状态条件寄存器的值。位 0 和 1 分别定义为通道 1 和通道 2 的状态。此命令不会更改寄存器设置。
命令	:STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:CONDition? 对于<CALibration CURRent TEMPerature TEST VOLTage>; CALibration: 指定查询校准状态是否失败的条件寄存器;

CURRent: 指定查询电流状态是否过流的条件寄存器;  
 TEMPerature: 指定查询温度状态是否过温的条件寄存器;  
 TEST: 指定查询自测试状态是否失败的条件寄存器;  
 VOLTage: 指定查询电压状态是否过压的条件寄存器;

<b>参数</b>	无
<b>查询响应</b>	<i>value&lt;newline&gt;</i> 返回指定寄存器的值，响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制)，由：FORMAT:SREGister 命令选择。
<b>示例</b>	:STAT:QUES:CAL:COND? :STAT:QUES:CURR:COND? :STAT:QUES:TEMP:COND? :STAT:QUES:TEST:COND? :STAT:QUES:VOLT:COND?

#### 4.12.8 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:ENABLE

<b>描述</b>	设置校准、电流、温度、测试、电压疑问使能寄存器。启用寄存器是一个掩码，允许在摘要位中报告事件寄存器中的真实情况。
<b>命令</b>	:STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:ENABLE mask :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:ENABLE?
<b>参数</b>	对于< CALibration CURRent TEMPerature TEST VOLTage >: CALibration: 指定可疑校准状态使能寄存器； CURRent: 指定可疑电流状态使能寄存器； TEMPerature: 指定可疑温度使能寄存器； TEST: 指定测试可疑状态使能寄存器； VOLTage: 指定电压可疑状态使能寄存器。 mask 掩码，0 到 65535(十进制)。默认为 0。参数类型为 NR1 或 NDN。 mask 掩码是集合比特的二进制加权值之和
<b>查询响应</b>	<i>mask&lt;newline&gt;</i> mask 返回指定启用寄存器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制)，由：FORMAT:SREGister 命令选择。
<b>示例</b>	:STAT:QUES:CURR:ENAB 65535 :STAT:QUES:TEMP:ENAB?

#### 4.12.9 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>[:EVENT]?

描述	返回有问题的校准、电流、温度、测试或电压状态事件寄存器的值。此命令可更改寄存器设置。
命令	:STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>[:EVENT]? 对于< CALibration CURRent TEMPerature TEST VOLTage >: CALibration: 指定可疑校准状态寄存器; CURRent: 指定可疑电流状态能寄存器; TEMPerature: 指定可疑温度寄存器; TEST: 指定测试可疑状态能寄存器; VOLTage: 指定电压可疑状态寄存器。
参数	无
查询响应	<i>value</i> <newline> <i>value</i> 返回指定事件寄存器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制), 由 :FORMAT:SREGister 命令选择。
示例	:STAT:QUES:CURR:EVEN? :STAT:QUES:VOLT:EVEN? :STAT:QUES:TEMP:EVEN?

#### 4.12.10 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>:NTRansition

描述	在有问题的校准、电流、温度、测试或电压状态寄存器中设置负转换过滤器。如果设置了筛选器的一个位，则其寄存器位的 1 到 0 转换将设置事件寄存器的相应位。
命令	:STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:NTRansition filter :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage>:NTRansition? 对于< CALibration CURRent TEMPerature TEST VOLTage >: CALibration: 指定可疑校准状态寄存器; CURRent: 指定可疑电流状态能寄存器; TEMPerature: 指定可疑温度寄存器; TEST: 指定测试可疑状态能寄存器; VOLTage: 指定电压可疑状态寄存器。
参数	<i>filter</i> 负转换滤波器, 0 到 65535(十进制), 默认值为 0。参数类型为 NR1 或 NDN。

---

	<i>filter</i> 负转换滤波器是集合比特的二进制加权值之和。
查询响应	<i>filter</i> <newline> <i>filter</i> 返回指定寄存器的负转换滤波器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制), 由: FORMat:SREGister 命令选择。
示例	:STAT:QUES:CURR:NTR 0 :STAT:QUES:TEMP:NTR?

---

#### 4.12.11 :STATus:QUEStionable:<CALibration|CURRent|TEMPerature|TEST|VOLTage>: PTRansition

---

描述	在有问题的校准、电流、温度、测试或电压状态寄存器中设置正转换过滤器。 如果设置了筛选器的一个位，则其寄存器位的 0 到 1 转换将设置事件寄存器的相应位。
命令	:STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage> :PTRansition filter :STATus:QUEStionable:<CALibration CURRent TEMPerature TEST VOLTage> :PTRansition? 对于< CALibration CURRent TEMPerature TEST VOLTage >: CALibration: 指定可疑校准状态寄存器; CURRent: 指定可疑电流状态能寄存器; TEMPerature: 指定可疑温度寄存器; TEST: 指定测试可疑状态能寄存器; VOLTage: 指定电压可疑状态寄存器。
参数	<i>filter</i> 正转换滤波器, 0 到 65535(十进制), 默认值为 32768。参数类型为 NR1 或 NDN。
查询响应	<i>filter</i> <newline> <i>filter</i> 返回指定寄存器的正转换滤波器的当前设置值。响应数据类型为 NR1 (十进制) 或 NDN (二进制、八进制或十六进制), 由: FORMat:SREGister 命令选择。
示例	:STAT:QUES:CURR:PTR 32767 :STAT:QUES:TEMP:PTR?

---

## 4.13 SYSTem Subsystem

### 4.13.1 :SYSTem:BEEPer[:IMMediate]

<b>描述</b>	控制蜂鸣器发出指定持续时间的声音。
<b>命令</b>	:SYSTem:BEEPer[:IMMediate] time
<b>参数</b>	time 持续的秒数。0.05 到 12.75 秒。参数类型为 NRf+。
<b>查询响应</b>	无
<b>示例</b>	:SYST:BEEP 0.5

### 4.13.2 :SYSTem:BEEPer:STATe

<b>描述</b>	启用或禁用蜂鸣器。*RST 复位操作会改变此蜂鸣器状态为启用。
<b>命令</b>	:SYSTem:BEEPer:STATe mode :SYSTem:BEEPer:STATe?
<b>参数</b>	mode 蜂鸣器使能或关闭。0 OFF 1 ON，参数类型为 boolean。 mode=1 或 ON 使能蜂鸣器。 mode=0 或 OFF 禁用蜂鸣器。
<b>查询响应</b>	mode<newline> mode 返回 0 或 1，表示蜂鸣器禁用或启用。响应数据类型为 NR1。
<b>示例</b>	:SYST:BEEP:STAT 1 :SYST:BEEP:STAT?

### 4.13.3 :SYSTem:COMMUnicatE:GPIB[:SELF]:ADDReSS

<b>描述</b>	设置 GPIB 地址。此设置不会因关机或执行*RST 复位命令而改变。
<b>命令</b>	:SYSTem:COMMUnicatE:GPIB[:SELF]:ADDReSS address :SYSTem:COMMUnicatE:GPIB[:SELF]:ADDReSS?
<b>参数</b>	address GPIB 地址，0 ~30。参数类型为 NR1。
<b>查询响应</b>	address<newline> address 返回 GPIB 地址。响应数据类型为 NR1。
<b>示例</b>	:SYST:COMM:GPIB:ADDR 15 :SYST:COMM:GPIB:ADDR?

#### 4.13.4 :SYSTem:COMMunicate:LAN:ADDRess

<b>描述</b>	设置 LAN(IP)静态 IP 地址。此设置在 :SYSTem:COMMunicate:LAN:DHCP 命令设置禁用 DHCP 后生效。此设置不会因关机或执行*RST 复位命令而改变。
<b>命令</b>	:SYSTem:COMMunicate:LAN:ADDRess address :SYSTem:COMMunicate:LAN:ADDRess? [CURRent STATic]
<b>参数</b>	<i>address</i> IP 地址。它必须是 A.B.C.D 格式，最多有 15 个字符。A, B, C, 和 D 必须是从 0 到 225 之间的数字。参数类型为 SPD。
<b>查询响应</b>	<i>address</i> <newline> <i>address</i> 返回静态局域网 (IP) 地址。如果设置了 CURRent 参数， <i>address</i> 返回当前设置。如果设置了 STATic 参数， <i>address</i> 返回下一次启动的保留值。响应数据类型为 SRD。
<b>示例</b>	:SYST:COMM:LAN:ADDR "192.168.100.200" :SYST:COMM:LAN:ADDR?

#### 4.13.5 :SYSTem:COMMunicate:LAN:DHCP

<b>描述</b>	启用或禁用动态主机配置协议 (DHCP)。此设置不会因关机或执行*RST 复位命令而改变。 启用 DHCP 后，仪器将尝试从 DHCP 服务器获取 IP 地址。如果 DHCP 服务器找到仪器，它将为仪器分配动态 IP 地址、子网掩码和默认网关。当 DHCP 被禁用或不可用时，仪器将在开机期间使用静态 IP 地址、子网掩码和默认网关。 如果 DHCP 服务器没有分配 DHCP LAN 地址，则将在大约 2 分钟的超时后使用静态 IP 地址。
<b>命令</b>	:SYSTem:COMMunicate:LAN:DHCP mode :SYSTem:COMMunicate:LAN:DHCP?
<b>参数</b>	<i>mode</i> DHCP 关闭或打开。0 OFF 1 ON。参数类型为 boolean。
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 返回 0 或 1，表示 DHCP 已关闭或开启。响应数据类型为 NR1。
<b>示例</b>	:SYST:COMM:LAN:DHCP 0 :SYST:COMM:LAN:DHCP?

#### 4.13.6 :SYSTem:COMMunicate:LAN:DNS

<b>描述</b>	设置 DNS 服务器的 IP 地址。此设置不会因关机或执行*RST 复位命令而改变。
-----------	--

---

<b>命令</b>	:SYSTem:COMMUnicatE:LAN:DNS[j] address :SYSTem:COMMUnicatE:LAN:DNS[j]? [CURREnt STATic]
<b>参数</b>	Address DNS 服务器的 IP 地址。必须是 A.B.C.D 格式，最多有 15 个字符。A, B, C, 和 D 必须是从 0 到 225 之间的数字。参数类型为 SPD.
<b>查询响应</b>	address <newline> address 返回 DNS 服务器的 IP 地址。如果设置了 CURREnt 参数，address 返回当前设置。如果设置了 STATic 参数，address 返回下一次启动的保留值。响应数据类型为 SRD.
<b>示例</b>	:SYST:COMM:LAN:DNS "192.168.0.254" :SYST:COMM:LAN:DNS?

---

#### 4.13.7 :SYSTem:COMMUnicatE:LAN:<GATE|GATeway>

---

<b>描述</b>	设置默认网关的 IP 地址。此设置在 :SYSTem:COMMUnicatE:LAN:DHCP 命令设置禁用 DHCP 后生效。此设置不会因关机或执行*RST 复位命令而改变。其中<GATE GATeway>，指定 GATE 或 GATeway。
<b>命令</b>	:SYSTem:COMMUnicatE:LAN:<GATE GATeway> address :SYSTem:COMMUnicatE:LAN:<GATE GATeway>? [CURREnt STATic]
<b>参数</b>	address 默认网关的 IP 地址。必须是 A.B.C.D 格式，最多有 15 个字符。A, B, C, 和 D 必须是从 0 到 225 之间的数字。参数类型为 SPD.
<b>查询响应</b>	address <newline> address 返回默认网关的 IP 地址。 如果设置了 CURREnt 参数，address 返回当前设置。 如果设置了 STATic 参数，address 返回下一次启动的保留值。 响应数据类型为 SRD.
<b>示例</b>	:SYST:COMM:LAN:GATE "192.168.100.1" :SYST:COMM:LAN:GATE?

---

#### 4.13.8 :SYSTem:COMMUnicatE:LAN:<HNAMe|HOSTname>

---

<b>描述</b>	设置仪器的主机名。此设置不会因关机或执行*RST 复位命令而改变。
<b>命令</b>	:SYSTem:COMMUnicatE:LAN:<HNAMe HOSTname> hostname :SYSTem:COMMUnicatE:LAN:<HNAMe HOSTname>? [CURREnt STATic]
<b>参数</b>	hostname 主机名。最多达 15 个字符。参数类型为 SPD.
<b>查询响应</b>	hostname <newline> hostname 返回该仪器的主机名。 如果设置了 CURREnt 参数，hostname 返回当前设置。 如果设置了 STATic 参数，hostname 返回下一次启动的保留值。

---

---

响应数据类型为 SRD。

---

**示例** :SYST:COMM:LAN:HNAM "S-SMM3312X-00001"  
:SYST:COMM:LAN:HOST?

---

#### 4.13.9 :SYSTem:COMMUnicatE:LAN:MAC?

<b>描述</b>	返回仪器的 MAC 地址。
<b>命令</b>	:SYSTem:COMMUnicatE:LAN:MAC?
<b>参数</b>	无
<b>查询响应</b>	mac_address <newline> mac_address 返回该仪器的 MAC 地址。响应数据类型为 SRD。
<b>示例</b>	:SYST:COMM:LAN:MAC?

---

#### 4.13.10 :SYSTem:COMMUnicatE:LAN:SMASK

<b>描述</b>	设置静态子网掩码。此设置在 :SYSTem:COMMUnicatE:LAN:DHCP 命令设置禁用 DHCP 后生效。此设置不会因关机或执行*RST 复位命令而改变。
<b>命令</b>	:SYSTem:COMMUnicatE:LAN:SMASK subnet_mask :SYSTem:COMMUnicatE:LAN:SMASK? [CURRent STATic]
<b>参数</b>	subnet_mask 子网掩码。必须是 A.B.C.D 格式，最多有 15 个字符。 A, B, C, 和 D 必须是从 0 到 225 之间的数字。参数类型为 SPD。
<b>查询响应</b>	subnet_mask <newline> subnet_mask 返回子网掩码。 如果设置了 CURRent 参数，subnet_mask 返回当前设置。 如果设置了 STATIC 参数，subnet_mask 返回下一次启动的保留值。 响应数据类型为 SRD。
<b>示例</b>	:SYST:COMM:LAN:SMAS "255.255.255.0" :SYST:COMM:LAN:SMAS?

---

#### 4.13.11 :SYSTem:DATA:QUANtity?

<b>描述</b>	返回数据缓冲区中指定通道的数据数。
<b>命令</b>	:SYSTem:DATA:QUANtity? [chanlist]
<b>参数</b>	chanlist Channels。参数类型为 channel list。 (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1). (@1) 仅选择通道 1。

---

(@2) 仅选择通道 2。  
 (@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。  
 如果未指定此参数, chanlist = (@1).

查询响应	<i>response</i> <newline>  <i>response</i> 返回数据的数量。响应数据类型为 NR1。如果频道列表同时选择了频道 1 和 2, 则响应将按顺序返回频道 1 数据的数量和频道 2 数据的数量。它们之间用逗号隔开。
------	---

示例	:SYST:DATA:QUAN? (@2)
----	-----------------------

#### 4.13.12 :SYSTem:DATE

描述	设置内部时钟的日期。此设置不会因关机或执行*RST 复位命令而改变。
命令	:SYSTem:DATE year, month, day :SYSTem:DATE?
参数	<i>year</i> 年, 4 位整数。参数类型为 NR1。 <i>month</i> 月, 从 1 到 12 的整数。参数类型为 NR1。 <i>day</i> 日, 从 1 到 31 的整数。参数类型为 NR1。
查询响应	<i>response</i> <newline> <i>response</i> 返回年、月、日。每个值都用一个逗号分隔。 响应数据类型为 NR1.
示例	:SYST:DATE 2025,1,1

#### 4.13.13 :SYSTem:ERRor:ALL?

描述	读取并返回错误/事件队列中的所有项, 并清除该队列。
命令	:SYSTem:ERRor:ALL?
参数	无
查询响应	<i>response</i> <newline> <i>response</i> 返回格式为 code, message 的消息, 表示错误/事件代码和消息。多个响应按 FIFO (先进先出) 顺序列出, 用逗号分隔。代码的数据类型为 NR1, 消息为 SRD。 如果该队列为空, 返回值为 +0,"No error"。
示例	:SYST:ERR:ALL?

#### 4.13.14 :SYSTem:ERRor:CODE:ALL?

<b>描述</b>	读取错误/事件队列中的所有项，返回所有代码，并清除该队列。
<b>命令</b>	:SYSTem:ERRor:CODE:ALL?
<b>参数</b>	无
	<i>code&lt;newline&gt;</i>
<b>查询响应</b>	<i>code</i> 返回错误/事件代码。多个响应以 FIFO（先入先出）的顺序列出，用逗号分隔。响应数据类型为 NR1。 如果该队列为空，返回值为+0。
<b>示例</b>	:SYST:ERR:CODE:ALL?

#### 4.13.15 :SYSTem:ERRor:CODE[:NEXT]?

<b>描述</b>	读取并删除错误/事件队列中的顶部项，并返回顶部代码。
<b>命令</b>	:SYSTem:ERRor:CODE[:NEXT]?
<b>参数</b>	无
	<i>code&lt;newline&gt;</i>
<b>查询响应</b>	<i>code</i> 返回错误/事件代码。响应数据类型为 NR1。 如果该队列为空，返回值为+0。
<b>示例</b>	:SYST:ERR:CODE?

#### 4.13.16 :SYSTem:ERRor:COUNt?

<b>描述</b>	返回错误/事件队列中的项目数。
<b>命令</b>	:SYSTem:ERRor:COUNt?
<b>参数</b>	无
	<i>response&lt;newline&gt;</i>
<b>查询响应</b>	<i>response</i> 返回项目数。响应数据类型为 NR1。 如果队列为空，则响应为+0。
<b>示例</b>	:SYST:ERR:COUN?

#### 4.13.17 :SYSTem:ERRor[:NEXT]?

<b>描述</b>	读取并删除错误/事件队列中的顶部项，并返回顶部代码和消息。
<b>命令</b>	:SYSTem:ERRor[:NEXT]?

<b>参数</b>	无
<b>查询响应</b>	<p><i>response</i>&lt;newline&gt;</p> <p><i>response</i> 返回格式为 code, message 的消息，表示错误/事件代码和消息。多个响应按 FIFO (先进先出) 顺序列出，用逗号分隔。代码的数据类型为 NR1，消息为 SRD。</p> <p>如果该队列为空，响应为 +0, “No error”。</p>
<b>示例</b>	:SYST:ERR?

#### 4.13.18 :SYSTem:FAN:MODE

<b>描述</b>	设置风扇控制模式。此设置不会因关机或执行*RST 复位命令而改变。
<b>命令</b>	<p>:SYSTem:FAN:MODE mode</p> <p>:SYSTem:FAN:MODE?</p>
<b>参数</b>	<p><i>mode</i> 风扇控制模式, NORMal RACK。参数类型为 CPD。</p> <p><i>mode</i>=NORM 正常模式。</p> <p><i>mode</i>=RACK 机柜模式。</p>
<b>查询响应</b>	<p><i>mode</i>&lt;newline&gt;</p> <p><i>mode</i> 返回当前设置, NORM or RACK, 响应数据类型为 CRD。</p>
<b>示例</b>	<p>:SYST:FAN:MODE RACK</p> <p>:SYST:FAN:MODE?</p>

#### 4.13.19 :SYSTem:GROup[:DEFine]

<b>描述</b>	<p>定义通道组。此设置不会因关机或执行*RST 复位命令而改变。</p> <p>通道分组用于自动控制通道输出时机以便在其他通道执行测量的时候保持输出。分组的通道按通道号的顺序开始源输出，然后同时开始测量，并保持输出，直到测量完成。</p> <p>如果分组被释放了，则不管其他通道的状况如何，这些通道都可以独立工作。</p> <p>通道分组的条件:</p> <ul style="list-style-type: none"> <li>• 触发源必须相同。</li> <li>• 如果将触发源设置为 TIMER，则源输出触发（瞬态动作）的触发周期必须相同。这不适用于 trigger count 触发计数设置为 1 的情况。</li> </ul>
<b>命令</b>	<p>:SYSTem:GROup[:DEFine] grouplist</p> <p>:SYSTem:GROup[:DEFine]?</p>
<b>参数</b>	<p><i>grouplist</i> 通道组设置。参数类型为 channel list。</p> <p><i>grouplist</i>=(@1,2) 将通道 1 和通道 2 组成分组;</p> <p><i>grouplist</i>=(@1),(@2) 释放分组。</p>

---

查询响应	<i>group[&lt;newline&gt;]</i> <i>group</i> 返回通道组设置。响应数据类型为 channel list。
示例	:SYST:GRO (@1,2) :SYST:GRO:DEF?

---

#### 4.13.20 :SYSTem:GROup:RESet

---

描述	释放由:SYSTem:GROup[:DEFine]命令定义的通道组。
命令	:SYSTem:GROup:RESet
参数	无
查询响应	无
示例	:SYST:GRO:RES

---

#### 4.13.21 :SYSTem:INTerlock:TRIPped?

---

描述	查询联锁电路闭合或打开状态.
命令	:SYSTem:INTerlock:TRIPped?
参数	无
查询响应	mode <newline> mode 返回 0 (闭合) 或 1 (开路)， 分别表示联锁电路处于闭合或开路。 响应数据类型为 NR1.
示例	:SYST:INT:TRIP?

---

#### 4.13.22 :SYSTem:LANGuage

---

描述	选择 SMM3000X 控制命令集。此命令设置不会因电源关闭或*RST 命令而改变。
命令	:SYSTem:LANGuage <i>mode</i> :SYSTem:LANGuage?
参数	<i>mode</i> SMM3000X 控制命令集。“DEFault” “2400”。参数数据类型为 SPD。 <i>mode</i> =“DEF”选择支持所有 SMM3000X 功能的默认命令集。 <i>mode</i> =“2400”选择为现有程序设计的常规命令集，这些程序是您为控制现有仪器而创建的，例如 Keithley 2400 标准系列。设置此模式后，SMM3000X 不支持本手册中描述的 SCPI 命令，而仅支持“SMM3000X 支持的常规命令”中列出的 SCPI 指令。

---

---

查询响应	<i>mode&lt;newline&gt;</i> <i>mode</i> 返回当前设置, DEF 或 2400。响应数据类型为 SRD。
示例	:SYST:LANG "2400" :SYST:LANG?

---

#### 4.13.23 :SYSTem:LFRrequency

---

描述	选择电源频率。此设置不会因关机或执行*RST 复位命令而改变。
命令	:SYSTem:LFRrequency <i>frequency</i> :SYSTem:LFRrequency?
参数	<i>frequency</i> 电源线路频率。50 (50 Hz, default) 60 (60 Hz)。参数类型为 NR1.
查询响应	<i>frequency&lt;newline&gt;</i> <i>frequency</i> 返回当前设置值, 50 或 60。响应数据类型为 NR1。
示例	:SYST:LFR 60 :SYST:LFR?

---

#### 4.13.24 :SYSTem:PON

---

描述	指定上电启动状态。 通电状态可以从出厂默认重置条件 (RST) 和用户条件 RCL0、RCL1、RCL2、RCL3 和 RCL4 中选择, 这些条件可以分别由*SAV 0、*SAV 1、*SAV 2、*SAV 3 和*SAV 4 命令定义。
命令	:SYSTem:PON <i>memory</i>
参数	<i>memory</i> 启动状态, RST(default) RCL0 RCL1 RCL2 RCL3 RCL4; 参数类型为 CPD。
查询响应	无
示例	:SYST:PON RCL0

---

#### 4.13.25 :SYSTem:PRESet

---

描述	复位仪器设置。
命令	:SYSTem:PRESet
参数	无
查询响应	无
示例	:SYST:PRESet

---

#### 4.13.26 :SYSTem:TIME

<b>描述</b>	设置内部时钟的时间。此设置不会因关机或执行*RST 复位命令而改变。
<b>命令</b>	:SYSTem:TIME <i>hour, minute, second</i> :SYSTem:TIME?
<b>参数</b>	<i>hour</i> 时。从 0 到 23 的整数。参数类型为 NR1。 <i>minute</i> 分。从 0 到 59 的整数。参数类型为 NR1。 <i>second</i> 秒。从 0 到 59 的整数。参数类型为 NR1。
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回 hour, minute, second。每个值都用一个逗号分隔。 响应数据类型为 NR1。
<b>示例</b>	:SYST:TIME 23,59,59 :SYST:TIME?

#### 4.13.27 :SYSTem:TIME:TIMer:COUNt?

<b>描述</b>	返回计时器的当前计数值。
<b>命令</b>	:SYSTem:TIME:TIMer:COUNt?
<b>参数</b>	无
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回当前的计时器计数值。 响应数据类型为 NR3。
<b>示例</b>	:SYST:TIME:TIM:COUN?

#### 4.13.28 :SYSTem:TIME:TIMer:COUNt:RESet:AUTO

<b>描述</b>	启用或禁用定时器的自动复位功能。如果此功能启用，则当启动操作发生时定时器计数。
<b>命令</b>	:SYSTem:TIME:TIMer:COUNt:RESet:AUTO mode :SYSTem:TIME:TIMer:COUNt:RESet:AUTO?
<b>参数</b>	<i>mode</i> 自动复位功能的开启或关闭。0 OFF 1 ON (default)。参数类型为 boolean。 <i>mode</i> =1 或 ON 启用自动复位功能。 <i>mode</i> =0 或 OFF 禁用自动复位功能。
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 返回 0 或 1，分别表示自动复位功能为关闭或开启。响应数据类型为 NR1。
<b>示例</b>	:SYST:TIME:TIM:COUN:RES:AUTO 0

---

:SYST:TIME:TIM:COUN:RES:AUTO?

---

#### 4.13.29 :SYSTem:TIME:TIMer:COUNT:RESet[:IMMEDIATE]

描述	立即重置计时器计数。
命令	:SYSTem:TIME:TIMer:COUNT:RESet[:IMMEDIATE]
参数	无
查询响应	无
示例	:SYST:TIME:TIM:COUN:RES

---

#### 4.13.30 :SYSTem:VERSion?

描述	返回 SCPI 标准的版本。此设置不会因关机或执行*RST 复位命令而改变。
命令	:SYSTem:VERSion?
参数	<i>response</i> <newline> <i>response</i> 返回 SCPI 标准的版本。比如：1999.0。响应数据类型为 NR2。
查询响应	无
示例	:SYST:VERS?

---

### 4.14 TRACe Subsystem

#### 4.14.1 :TRACe:CLEar

描述	清除指定通道的跟踪缓冲区。当通过:TRACe:FEED:CONTrol 命令将跟踪缓冲区控制模式设置为 NEV 时，此命令有效。
命令	:TRACe[c]:CLEar
参数	无
查询响应	无
示例	:TRAC:CLE

---

#### 4.14.2 :TRACe:DATA?

描述	返回跟踪缓冲区中的数据。放置在缓冲区中的数据由:TRACe: FEED 命令指定。
----	---

---

<b>命令</b>	:TRACe[c]:DATA? [offset[, size]]
	<i>offset</i> 表示所接收到的数据的开始索引。n CURRENT STARt (default)。参数类型为 NR1 或 CPD。 <i>offset</i> = n 从第 n+1 个数据开始, n 是整数, 0 到最大值 (取决于缓冲区状态)。
<b>参数</b>	<i>offset</i> = CURR 指定当前的数据的位置。 <i>offset</i> = STAR 指定跟踪缓冲区的顶部。等同于 <i>offset</i> = 0。
	<i>size</i> 要接收的数据数量。1 到最大值 (取决于缓冲区状态)。参数类型为 NR1。如果未指定此参数, 此时将返回从 <i>offset</i> 开始的所有数据
<b>查询响应</b>	data <newline> 响应数据类型为 NR3。参见“数据输出格式”。
<b>示例</b>	:TRAC:DATA? 0,10

#### 4.14.3 :TRACe:FEED

<b>描述</b>	指定放置在跟踪缓冲区中的数据。当通过 :TRACe:FEED:CONTrol 命令将跟踪缓冲区控制模式设置为 NEV 时, 此命令有效。
<b>命令</b>	:TRACe[c]:FEED <i>type</i> :TRACe[c]:FEED?
	<i>type</i> 数据类型。MATH LIMit SENSe (default)。参数类型为 CPD。 <i>type</i> = SENS 指定测量结果数据, 其中包含由:FORMAT:ELEMENTS:SENSe 命令指定的所有电压测量数据、电流测量数据、电阻测量数据、时间数据、状态数据或源输出设置数据。 <i>type</i> = LIM 指定极限测试数据。该数据包含由:FORMAT:ELEMENTS:CALCULATE 命令指定的极限测试数据、时间数据或状态数据。对于极限测试数据, 参见:CALCULATE: DATA?。 <i>type</i> = MATH 指定计算结果数据。数据包含:FORMAT:ELEMENTS:CALCULATE 所指定的计算结果、时间数据或状态数据。参见“:CALCULATE:MATH:DATA?”以获取更多信息。
<b>参数</b>	<i>type</i> <newline>
<b>查询响应</b>	<i>type</i> 返回数据类型的当前设置, MATH, LIM, 或 SENS。 响应数据类型为 CRD。
<b>示例</b>	:TRAC:FEED MATH :TRAC:FEED?

#### 4.14.4 :TRACe:FEED:CONTrol

<b>描述</b>	选择跟踪缓冲区控制模式。
<b>命令</b>	:TRACe[c]:FEED:CONTrol mode :TRACe[c]:FEED:CONTrol?
<b>参数</b>	<p><i>mode</i> 跟踪缓冲区控制模式。NEXT NEVer (default)。参数类型为 CPD。</p> <p><i>mode</i>=NEV 禁用对跟踪缓冲区的写入操作。命令 :TRACe:CLEar, :TRACe:FEED, 和:TRACe:POINts 可用。</p> <p><i>mode</i>=NEXT 启用写入操作, 直到缓冲区已满。缓冲区满后自动更改模式为 NEV。不会报错。</p>
<b>查询响应</b>	<i>mode</i> <newline> <i>mode</i> 返回当前设置的控制模式, NEXT 或 NEV。响应数据类型为 CRD。
<b>示例</b>	:TRAC:FEED:CONT NEXT :TRAC2:FEED:CONT?

#### 4.14.5 :TRACe:FREE?

<b>描述</b>	返回跟踪缓冲区的可用大小 (available) 和总大小 (total)。
<b>命令</b>	:TRACe[c]:FREE?
<b>参数</b>	无
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回 available, total。每个值都用一个逗号分隔。 响应数据类型为 NR1。
<b>示例</b>	:TRAC2:FREE?

#### 4.14.6 :TRACe:POINts

<b>描述</b>	设置跟踪缓冲区的大小。当通过 :TRACe:FEED:CONTrol 命令将跟踪缓冲区控制模式设置为 NEV 时, 此命令将有效。
<b>命令</b>	:TRACe[c]:POINts points :TRACe[c]:POINts? [points]
<b>参数</b>	<i>points</i> 大小。value (1 to 100000) MINimum MAXimum DEFault (默认为 100000)。参数类型为 NR1。查询不支持 points = value。
<b>查询响应</b>	<i>points</i> <newline> <i>points</i> 返回缓冲区大小的当前设置。 如果指定了一个参数, 则点将返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR1。

---

示例	:TRAC:POIN 10000 :TRAC2:POIN?
----	----------------------------------

---

#### 4.14.7 :TRACe:POINts:ACTual?

---

描述	返回跟踪缓冲区中的数据数。
命令	:TRACe[c]:POINts:ACTual?
参数	无
查询响应	points <newline> points 返回跟踪缓冲区中的数据数。响应数据类型为 NR1。
示例	:TRAC2:POIN:ACT?

---

#### 4.14.8 :TRACe:STATistic:DATA?

---

返回对存储在跟踪缓冲区中的数据的统计操作的结果。在执行此命令之前，必须由 :TRACe:STATistic:FORMAT 命令来指定统计操作。	
描述	如果跟踪缓冲区正在存储多种数据类型的原始测量数据，则对所有测量数据执行统计操作。  对 TIME 和 STATus 数据没有统计操作。
命令	:TRACe[c]:STATistic:DATA?
参数	response <newline> response 返回统计操作的结果。响应数据类型为 NR3。 参见“数据输出格式”。
查询响应	无
示例	:TRAC:STAT:DATA?

---

#### 4.14.9 :TRACe:STATistic:FORMAT

---

描述	选择通过 :TRACe:STATistic:DATA? 命令执行的统计操作。
命令	:TRACe[c]:STATistic:FORMAT operation :TRACe[c]:STATistic:FORMAT?
参数	operation 统计操作。MINimum MAXimum SDEViation  PKPK MEAN (default)。参数类型为 CPD。  operation=MEAN 设置获取平均值的操作。 operation=SDEV 设置获取标准偏差的操作。 operation=PKPK 设置获取峰值到峰值的操作。

---

---

	<i>operation</i> =MIN 设置获取最小值的操作。 <i>operation</i> =MAX 设置获取最大值的操作。
查询响应	<i>operation</i> 返回当前设置的统计操作, MEAN, SDEV, PKPK, MIN 或 MAX。 响应数据类型为 CRD。
示例	:TRAC:STAT:FORM PKPK :TRAC2:STAT:FORM?

---

#### 4.14.10 :TRACe:TSTamp:FORMAT

---

描述	选择用于在跟踪缓冲区中读取时间戳数据的规则。.
命令	:TRACe[c]:TSTamp:FORMAT <i>rule</i> :TRACe[c]:TSTamp:FORMAT?
参数	<i>rule</i> 用于读取时间戳数据的规则。DELTaABSolute (default)。参数类型为 CPD。 <i>rule</i> = ABS 将返回的数据设置为第一个时间戳数据的增量值. <i>rule</i> = DELT 将返回的数据设置为先前时间戳数据的增量值.
查询响应	<i>rule</i> <newline> <i>rule</i> 返回当前设置的规则, DELT 或 ABS。响应数据类型为 CRD。
示例	:TRAC:TST:FORM DELT :TRAC2:TST:FORM?

---

### 4.15 TRIGger Subsystem

#### 4.15.1 :ABORT<:ACQuire[:TRANSient|[:ALL]]>

---

描述	中止针对指定通道的指定设备操作, 触发状态变为空闲。
命令	:ABORT<:ACQuire[:TRANSient [:ALL]]> [ <i>chanlist</i> ] :ACQuire 指定测量; :TRANSient 指定源输出; :ALL 指定测量和源输出;
参数	<i>chanlist</i> 通道。参数类型为 channel list。 (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1) (@1) 仅选择通道 1。 (@2) 仅选择通道 2。 (@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。 如果未指定此参数, <i>chanlist</i> = (@1) 。
查询响应	无

---

---

示例	:ABOR:ACQ (@2)
----	----------------

---

#### 4.15.2 :ARM<:ACQuire[:TRANsient][:ALL]>[:IMMEDIATE]

描述	向指定通道的指定设备动作发送一个 arm 立即触发信号，当启动了指定的设备动作时，arm 触发信号将导致从 arm 层到 trigger 层的变化。
命令	:ARM<:ACQuire[:TRANsient][:ALL]>[:IMMEDIATE] [chanlist] :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
参数	<i>chanlist</i> 通道。参数类型为 channel list。 (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1)。
查询响应	无
示例	:ARM:ACQ (@2)

---

#### 4.15.3 :ARM<:ACQuire[:TRANsient][:ALL]>[:LAYer]:BYPass

描述	启用或禁用 ARM 层中的事件检测器的旁路。
命令	:ARM[c]<:ACQuire[:TRANsient][:ALL]>[:LAYer]:BYPass <i>bypass</i> :ARM[c]<:ACQuire[:TRANsient]>[:LAYer]:BYPass?
参数	:ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;  <i>bypass</i> 旁路设置。ONCE OFF (default)。参数类型为 CPD。 <i>bypass=OFF</i> 禁用旁路。 <i>bypass=ONCE</i> 启用旁路，但仅为第一个循环。
查询响应	<i>response</i> <newline> <i>response</i> 返回旁路、OFF 或 ONCE 的当前设置。 响应数据类型为 CRD。
示例	:ARM:BYP ONCE :ARM2:TRAN:BYP?

---

#### 4.15.4 :ARM<:ACQuire[:TRANsient[:ALL]>[:LAYer]:COUNt

<b>描述</b>	设置所指定的设备操作的 ARM 层计数.
<b>命令</b>	:ARM[c]<:ACQuire[:TRANsient[:ALL]>[:LAYer]:COUNT <i>arm_count</i> :ARM[c]<:ACQuire[:TRANsient]>[:LAYer]:COUNT? [ <i>arm_count</i> ] :ARM[c][:ALL][:LAYer]:COUNT? <i>arm_count</i> :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
<b>参数</b>	<i>arm_count</i> Arm 数量。 value (1 到 100000 或 2147483647) INFinity MINimum MAXimum DEFault(默认为 1)。参数类型为 NRf+。value=2147483647 表示无穷。 查询不支持 <i>arm_count</i> = value 和 INFinity. <i>Arm count</i> 触发器计数必须小于 100001。
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回当前设置的 ARM 计数。如果指定了一个参数，响应将返回分配给 DEF、MIN、MAX 或 INF 的值。响应数据类型为 NR1。如果 ARM 计数设置为无穷大，则响应返回 2147483647.
<b>示例</b>	:ARM:COUN 10 :ARM2:TRAN:COUN?

#### 4.15.5 :ARM<:ACQuire[:TRANsient[:ALL]>[:LAYer]:DELay

<b>描述</b>	设置指定设备操作的 ARM 层延迟.
<b>命令</b>	:ARM[d]<:ACQuire[:TRANsient[:ALL]>[:LAYer]:DELay <i>delay</i> :ARM[d]<:ACQuire[:TRANsient]>[:LAYer]:DELay? [ <i>delay</i> ] :ARM[d][:ALL][:LAYer]:DELay? <i>delay</i> :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
<b>参数</b>	<i>delay</i> Arm 延迟, 以秒为单位。 value (0 到 1 00000) MINimum MAXimum DEFault (0)。 参数类型为 NRf+。查询不支持 <i>delay</i> = value。
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回当前设置的 ARM 延迟。如果指定了一个参数，则响应返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3.
<b>示例</b>	:ARM:DEL 0.1 :ARM2:TRAN:DEL?

#### 4.15.6 :ARM<:ACQuire[:TRANsient[:ALL]>[:LAYer]:SOURce:LAN

<b>描述</b>	指定的一个或多个 LXI 触发用作指定设备操作的 ARM 层触发源.
<b>命令</b>	:ARM[c]<:ACQuire[:TRANsient[:ALL]>[:LAYer]:SOURce:LAN lan_id{,lan_id} :ARM[c]<:ACQuire[:TRANsient]>[:LAYer]:SOURce:LAN? :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
<b>参数</b>	<i>lan_id</i> LXI 触发的 LAN ID。 LAN0 LAN1 LAN2 LAN3 LAN4  LAN5 LAN6 LAN7。全部选择为默认值。 参数类型为 CPD.
<b>查询响应</b>	<i>response</i> <newline> <i>response</i> 返回当前的设置, LAN0 到 LAN7。响应数据类型为 CRD。多个响应用一个逗号分隔。
<b>示例</b>	:ARM:SOUR:LAN LAN7 :ARM2:TRAN:SOUR:LAN?

#### 4.15.7 :ARM<:ACQuire[:TRANsient[:ALL]>[:LAYer]:SOURce[:SIGNAl]

<b>描述</b>	为指定的设备操作选择 ARM 层触发源.
<b>命令</b>	:ARM[c]<:ACQuire[:TRANsient[:ALL]>[:LAYer]:SOURce[:SIGNAl] source :ARM[c]<:ACQuire[:TRANsient]>[:LAYer]:SOURce[:SIGNAl]? :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
<b>参数</b>	<i>source</i> Arm 触发源。AIN1 (default) BUS TIM1 INT1 INT2 LAN EXT7  EXT8 EXT9 EXT10  EXT11 EXT12。参数类型为 CPD。 <i>source</i> =AIN1, (内部自动)通过使用内部算法, 自动选择最适合当前工作模式的 ARM 源。 <i>source</i> =BUS, 选择远程接口触发器命令, 如组执行触发 (GET) 和*TRG 命令。 <i>source</i> =TIM1, 选择 :ARM<:ACQuire[:TRANsient[:ALL]>[:LAYer]:TIM1 命令设置的每个间隔内部生成的信号。 <i>source</i> =INT1 或 INT2, 分别从内部总线 1 或 2 中选择一个信号. <i>source</i> =LAN, 选择由 :ARM<:ACQuire[:TRANsient[:ALL]>[:LAYer]:SOURce:LAN 命令指定的 LXI 触发器。 <i>source</i> =EXTn, 从 GPIO 引脚 n 中选择一个信号, 这是后面板上的 Digital I/O D-sub 连接器的输入端口。n = 7 到 12.

---

	<i>response</i> <newline>
查询响应	<i>response</i> 返回当前设置 Arm 触发, AINT、BUS、TIM、INT1、INT2、LAN 或 EXT7 到 EXT12 的。响应数据类型为 CRD。
示例	:ARM:SOUR AINT :ARM2:TRAN:SOUR?

---

#### 4.15.8 :ARM<:ACQuire[:TRANsient[:ALL]]>[:LAYer]:TlMer

---

描述	设置指定设备操作的 TlMer 定时 ARM 层的时间间隔.
命令	:ARM[c]<:ACQuire[:TRANsient[:ALL]]>[:LAYer]:TlMer <i>interval</i> :ARM[c]<:ACQuire[:TRANsient]>[:LAYer]:TlMer? [ <i>interval</i> ] :ARM[c][:ALL][:LAYer]:TlMer? <i>interval</i> :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
参数	<i>interval</i> 间隔, 以秒为单位。 value (1E-5 到 1E+5 秒) MINimum MAXimum DEFault(默认值为最小值)。 参数类型为 NRf+。查询不支持 interval = value.
查询响应	<i>response</i> <newline> <i>response</i> 返回当前设置的时间间隔。 如果指定了一个参数, <i>response</i> 返回分配给 DEF、MIN 或 MAX 的值。 响应数据类型为 NR3。
示例	:ARM:TlMER 2E-4 :ARM2:TRAN:TlMER?

---

#### 4.15.9 :ARM<:ACQuire[:TRANsient[:ALL]]>[:LAYer]:TOUtput:SIGNal

---

描述	选择从空闲状态进入 ARM 层时的触发输出。可以设置多个触发输出端口.
命令	:ARM[c]<:ACQuire[:TRANsient[:ALL]]>[:LAYer]:TOUtput:SIGNal <i>output1,output2</i> :ARM[c]<:ACQuire[:TRANsient]>[:LAYer]:TOUtput:SIGNal? :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
参数	<i>output</i> 触发输出端口。 EXT1 (default) EXT2 EXT3 EXT4  EXT5 EXT6 LAN INT1 INT2。 参数类型为 CPD. <i>output</i> =INT1 / INT2, 选择内部总线 1 或 2。

---

---

*output= LAN* 选择一个 LAN 端口。  
*output=EXTn* 选择后面板上的 Digital I/O D-sub 连接器上的 GPIO 引脚，  
 n=1 to 6.

---

	<i>response&lt;newline&gt;</i>
查询响应	<i>response</i> 返回当前设置的触发输出，INT1, INT2, LAN 或 EXT1 到 EXT6。 响应数据类型为 CRD。多个响应用一个逗号隔开。
示例	:ARM:TOUT:SIGN EXT1 :ARM2:TRAN:TOUT:SIGN?

---

#### 4.15.10 :ARM<:ACQuire[:TRANsient[:ALL]>[:LAYer]:TOUTput[:STATe]

描述	启用或禁用触发状态从空闲状态切换到 ARM 层时的触发输出.
命令	:ARM[c]<:ACQuire[:TRANsient[:ALL]>[:LAYer]:TOUTput[:STATe] <i>mode</i> :ARM[c]<:ACQuire[:TRANsient]>[:LAYer]:TOUTput[:STATe]?
参数	:ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;  <i>mode</i> 触发器输出：打开或关闭。 1 ON 0 OFF (default)。参数类型为 boolean。 <i>mode</i> =1 或 ON 启用触发器输出。 <i>mode</i> =0 或 OFF 禁用触发器输出。
查询响应	<i>response&lt;newline&gt;</i> <i>response</i> 返回 1 或 0，分别指示触发输出已打开或关闭。 响应数据类型为 NR1。
示例	:ARM:TOUT 1 :ARM2:TRAN:TOUT:STAT?

---

#### 4.15.11 :IDLE<:ACQuire[:TRANsient[:ALL]>?

描述	检查指定通道的指定设备操作的状态，并等待触发状态变为空闲.
命令	:IDLE[c]<:ACQuire[:TRANsient[:ALL]>? :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
参数	<i>response &lt;newline&gt;</i> <i>response</i> 如果指定的设备操作处于空闲状态，则返回 1。 响应数据类型为 NR1。
查询响应	无

---

---

示例	:IDLE2:ACQ?
----	-------------

---

#### 4.15.12 :INITiate[:IMMediate]<:ACQuire[:TRANsient][:ALL]>

描述	启动指定的通道的指定的设备操作。触发状态从空闲状态更改为启动状态。
命令	:INITiate[:IMMediate]<:ACQuire[:TRANsient][:ALL]> [ <i>chanlist</i> ] :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
参数	<i>chanlist</i> 通道。参数类型为 channel list。 (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1). (@1) 仅选择通道 1。 (@2) 仅选择通道 2。 (@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。 如果未指定此参数, chanlist = (@1).
查询响应	无
示例	:INIT:ACQ (@2)

---

#### 4.15.13 :TRIGger<:ACQuire[:TRANsient][:ALL]>:BYPass

描述	启用或禁用 trigger 层中的事件检测器的旁路。
命令	:TRIGger[c]<:ACQuire[:TRANsient][:ALL]>:BYPass <i>bypass</i> :TRIGger[c]<:ACQuire[:TRANsient]>:BYPass? :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
参数	<i>bypass</i> 旁路设置。ONCE OFF (default)。参数类型为 CPD。 <i>bypass</i> = OFF 禁用旁路。 <i>bypass</i> = ONCE 启用旁路, 但仅作用于第一循环。
查询响应	<i>response</i> <newline> <i>response</i> 返回当前设置的旁路使能状态 OFF 或 ONCE。 响应数据类型为 CRD.
示例	:TRIG:BYP ONCE :TRIG2:TRAN:BYP?

---

#### 4.15.14 :TRIGger<:ACQuire[:TRANSient[:ALL]]>:COUNT

<b>描述</b>	设置所指定的设备操作的 trigger 计数。
<b>命令</b>	:TRIGger[c]<:ACQuire[:TRANSient[:ALL]]>:COUNT <i>trigger_count</i> :TRIGger[c]<:ACQuire[:TRANSient]>:COUNT? [ <i>trigger_count</i> ] :TRIGger[c][:ALL]:COUNT? <i>trigger_count</i> :ACQuire 指定测量; :TRANSient 指定源输出; :ALL 指定测量和源输出;
<b>参数</b>	<i>trigger_count</i> 触发计数。 value (1 到 100000 或 2147483647)  INFinity MINimum MAXimum DEFault (1)。参数类型为 NRf+。 value=2147483647 表示无穷大。查询不支持 trigger_count = value 和 INFinity。 Arm count × Trigger count 必须小于 100001。
<b>查询响应</b>	response <newline> response 返回当前设置的 trigger 计数. 如果指定了一个参数, response 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR1.
<b>示例</b>	:TRIG:COUN 10 :TRIG2:TRAN:COUN?

#### 4.15.15 :TRIGger<:ACQuire[:TRANSient[:ALL]]>:DElay

<b>描述</b>	设置所指定的设备操作的 trigger 延迟。
<b>命令</b>	:TRIGger[c]<:ACQuire[:TRANSient[:ALL]]>:DElay <i>delay</i> :TRIGger[c]<:ACQuire[:TRANSient]>:DElay? [ <i>delay</i> ] :TRIGger[c][:ALL]:DElay? <i>delay</i> :ACQuire 指定测量; :TRANSient 指定源输出; :ALL 指定测量和源输出;
<b>参数</b>	<i>delay</i> trigger 延迟, 以秒为单位。 value (0 到 100000) MINimum MAXimum DEFault (0)。 参数类型为 NRf+。查询不支持 delay = value。
<b>查询响应</b>	response <newline> response 返回当前设置的 trigger 延迟。如果指定了一个参数, response 返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3.
<b>示例</b>	:TRIG:DEL 0.1 :TRIG2:TRAN:DEL?

#### 4.15.16 :TRIGger<:ACQuire[:TRANSient[:ALL]]>[:IMMEDIATE]

<b>描述</b>	向指定的通道的指定的设备操作发送一个即时 trigger 触发。当指定的设备操作处于启动状态时，该触发信号将触发指定的设备操作。
<b>命令</b>	:TRIGger<:ACQuire[:TRANSient[:ALL]]>[:IMMEDIATE] [ <i>chanlist</i> ] :ACQuire 指定测量; :TRANSient 指定源输出; :ALL 指定测量和源输出
<b>参数</b>	<i>chanlist</i> Channels。参数类型为 channel list。 (@1) (@2) (@1,2) (@1:2) (@2,1) (@2:1). (@1) 仅选择通道 1。 (@2) 仅选择通道 2。 (@1,2), (@1:2), (@2,1), and (@2:1) 同时选择通道 1 和通道 2。 如果未指定此参数, <i>chanlist</i> = (@1)。
<b>查询响应</b>	无
<b>示例</b>	:TRIG:ACQ (@2)

#### 4.15.17 :TRIGger<:ACQuire[:TRANSient[:ALL]]>:SOURce:LAN

<b>描述</b>	指定的一个或多个 LXI 触发用作指定设备操作的 trigger 触发源.
<b>命令</b>	:TRIGger[c]<:ACQuire[:TRANSient[:ALL]]>:SOURce:LAN <i>lan_id</i> , <i>lan_id</i> ? :TRIGger[c]<:ACQuire[:TRANSient]>:SOURce:LAN?
<b>参数</b>	<i>lan_id</i> LXI 触发器的 LAN ID。 LAN0 LAN1 LAN2 LAN3 LAN4  LAN5 LAN6 LAN7。默认全选。 参数类型为 CPD。
<b>查询响应</b>	response <newline> response 返回当前设置, LAN0 到 LAN7。响应数据类型为 CRD。多个响应 应用一个逗号隔开。
<b>示例</b>	:TRIG:SOUR:LAN LAN7 :TRIG2:TRAN:SOUR:LAN?

#### 4.15.18 :TRIGger<:ACQuire[:TRANSient[:ALL]]>:SOURce[:SIGNAl]

<b>描述</b>	为指定的设备操作选择 trigger 触发源。
<b>命令</b>	:TRIGger[c]<:ACQuire[:TRANSient[:ALL]]>:SOURce[:SIGNAl] <i>source</i>

---

	:TRIGger[c]<:ACQuirel:TRANsient>:SOURce[:SIGNal]?
	:ACQuire 指定测量;
	:TRANsient 指定源输出;
	:ALL 指定测量和源输出;
参数	<i>source</i> trigger 触发源。 AINT (default) BUS TIM INT1 INT2 LAN EXT7 EXT8 EXT9 EXT10  EXT11 EXT12。参数类型为 CPD。
	<i>source</i> =AINT, (内部自动)通过使用内部算法, 自动选择最适合当前工作模式的 ARM 源。
	<i>source</i> =BUS, 选择远程接口触发器命令, 如组执行触发 (GET) 和*TRG 命令。
	<i>source</i> =TIM, 选择
	:TRIGger <:ACQuirel:TRANsient[:ALL]>[:LAYer]:TIM 命令设置的每个间隔内部生成的信号。
	<i>source</i> =INT1 或 INT2, 分别从内部总线 1 或 2 中选择一个信号.
	<i>source</i> =LAN, 选择由 :TRIGger <:ACQuirel:TRANsient[:ALL]>[:LAYer]:SOURce:LAN 命令指定的 LXI 触发器.
查询响应	<i>source</i> =EXTn, 从 GPIO 引脚 n 中选择一个信号, 这是后面板上的 Digital I/O D-sub 连接器的输入端口。n = 7 到 12。
	<i>response</i> <newline>
示例	:TRIG:SOUR EXT8 :TRIG2:TRAN:SOUR:SIGN?

---

#### 4.15.19 :TRIGger<:ACQuirel:TRANsient[:ALL]>:TIMer

---

描述	设置指定设备操作的 TIMer 定时 trigger 触发源的时间间隔.
命令	:TRIGger[c]<:ACQuirel:TRANsient[:ALL]>:TIMer <i>interval</i> :TRIGger[c]<:ACQuirel:TRANsient>:TIMer? [ <i>interval</i> ] :TRIGger[c][:ALL]:TIMer? <i>interval</i> :ACQuire 指定测量; :TRANsient 指定源输出; :ALL 指定测量和源输出;
参数	<i>interval</i> 间隔, 以秒为单位。 value (1E-5 到 1E+5) MINimum MAXimum DEFault (默认值为最小值)。 参数类型为 NRf+。查询不支持 <i>interval</i> = value。
查询响应	<i>response</i> <newline>

---

*response* 返回当前设置的时间间隔。如果指定了一个参数，则响应返回分配给 DEF、MIN 或 MAX 的值。响应数据类型为 NR3。

**示例**

```
:TRIG:TIM 2E-4
:TRIG2:TRAN:TIM?
```

#### 4.15.20 :TRIGger<:ACQuire[:TRANSient[:ALL]]>:TOUtput:SIGNal

**描述** 为 ARM 层和 Trigger 层之间的状态更改选择触发输出。可以设置多个触发器输出端口。

**命令**

```
:TRIGger[c]<:ACQuire[:TRANSient[:ALL]]>:TOUtput:SIGNal
    output{,output}
:TRIGger[c]<:ACQuire[:TRANSient]>:TOUtput:SIGNal?
:ACQuire 指定测量;
:TRANSient 指定源输出;
:ALL 指定测量和源输出;
```

**参数**

*output* 触发输出端口。EXT1 (default)|EXT2|EXT3|EXT4|EXT5|EXT6|LAN|INT1|INT2。参数类型为 CPD。  
*output*=INT1 or INT2 分别选择内部总线 1 或 2。  
*output*=LAN 选择 LAN 端口。  
*output*=EXTn 选择后面板上的 Digital I/O D-sub 连接器上的 GPIO 引脚，  
n=1 to 6.

**查询响应**

*response* 返回当前设置的触发输出，INT1, INT2, LAN 或 EXT1 到 EXT6。  
响应数据类型为 CRD。多个响应用一个逗号隔开。

**示例**

```
:TRIG:TOUT:SIGN EXT3
:TRIG2:TRAN:TOUT:SIGN?
```

#### 4.15.21 :TRIGger<:ACQuire[:TRANSient[:ALL]]>:TOUtput[:STATe]

**描述** 启用或禁用 ARM 层和 Trigger 层之间状态更改的触发输出。

**命令**

```
:TRIGger[c]<:ACQuire[:TRANSient[:ALL]]>:TOUtput[:STATe] mode
:TRIGger[c]<:ACQuire[:TRANSient]>:TOUtput[:STATe]?
:ACQuire 指定测量;
:TRANSient 指定源输出;
:ALL 指定测量和源输出;
```

**参数**

*mode* 触发器输出：打开或关闭。  
1|ON|0|OFF (default)。参数类型为 boolean 布尔型。  
*mode*=1 或 ON 启用触发器输出。

---

*mode*=0 或 OFF 禁用触发器输出。

---

*response*<newline>

**查询响应** *response* 返回 1 或 0，分别指示触发输出已打开或关闭。  
响应数据类型为 NR1。

---

**示例** :TRIG:TOUT 1

:TRIG2:TRAN:TOUT:STAT?

---

## 5 使用现有程序

### 5.1 SMM3000X 支持的常规命令

:ABORt  
:ARM[:SEQUence[1]][:LAYer[1]]:COUNT <NRf|DEFault|MINimum|MAXimum>  
:ARM[:SEQUence[1]][:LAYer[1]]:COUNT?  
:ARM[:SEQUence[1]][:LAYer[1]]:SOURce?  
:ARM[:SEQUence[1]][:LAYer[1]][:TCONfigure]:DIRection SOURcelACCeptor  
:ARM[:SEQUence[1]][:LAYer[1]][:TCONfigure]:DIRection?  
:ARM[:SEQUence[1]][:LAYer[1]]:TIMer <NRf>  
:ARM[:SEQUence[1]][:LAYer[1]]:TIMer?  
:CALCulate[1]:DATA:LATest?  
:CALCulate[1]:DATA?  
:CALCulate[1]:MATH[:EXPRESSION]:CATalog?  
:CALCulate[1]:MATH[:EXPRESSION][:DEFine] <form>  
:CALCulate[1]:MATH[:EXPRESSION][:DEFine]?  
:CALCulate[1]:MATH[:EXPRESSION]:DElete:ALL  
:CALCulate[1]:MATH[:EXPRESSION]:DElete[:SELected] <SPD>  
:CALCulate[1]:MATH[:EXPRESSION]:NAME <SPD>  
:CALCulate[1]:MATH[:EXPRESSION]:NAME?  
:CALCulate[1]:MATH:UNITs <name>  
:CALCulate[1]:MATH:UNITs?  
:CALCulate[1]:STATe <Bool>  
:CALCulate[1]:STATe?  
:CALCulate2:CLIMits:BControl IMMEDIATE|END  
:CALCulate2:CLIMits:BControl?  
:CALCulate2:CLIMits:CLEar:AUTO <Bool>  
:CALCulate2:CLIMits:CLEar:AUTO?  
:CALCulate2:CLIMits:CLEar[:IMMediate]  
:CALCulate2:CLIMits:FAIL:SOURce2 <NRf|NDN>  
:CALCulate2:CLIMits:FAIL:SOURce2?

:CALCulate2:CLIMits:MODE GRADing|SORTing  
:CALCulate2:CLIMits:MODE?  
:CALCulate2:CLIMits:PASS:SOURce2 <NRf|NDN>  
:CALCulate2:CLIMits:PASS:SOURce2?  
:CALCulate2:DATA:LATest?  
:CALCulate2:DATA?  
:CALCulate2:FEED CALCulate[1]|VOLTage|CURRent|RESistance  
:CALCulate2:FEED?  
:CALCulate2:LIMit[1]:COMPliance:FAIL IN|OUT  
:CALCulate2:LIMit[1]:COMPliance:FAIL?  
:CALCulate2:LIMit[1]:COMPliance:SOURce2 <NRf|NDN>  
:CALCulate2:LIMit[1]:COMPliance:SOURce2?  
:CALCulate2:LIMit[1]:FAIL?  
:CALCulate2:LIMit[1]:STATe <Bool>  
:CALCulate2:LIMit[1]:STATe?  
:CALCulate2:LIMit[2|3|5-12]:FAIL?  
:CALCulate2:LIMit[2|3|5-12]:LOWer[:DATA] <NRf>  
:CALCulate2:LIMit[2|3|5-12]:LOWer[:DATA]? [DEFault|MINimum|MAXimum]  
:CALCulate2:LIMit[2|3|5-12]:LOWer:SOURce2 <NRf|NDN>  
:CALCulate2:LIMit[2|3|5-12]:LOWer:SOURce2?  
:CALCulate2:LIMit[2|3|5-12]:PASS:SOURce2 <NRf|NDN>  
:CALCulate2:LIMit[2|3|5-12]:PASS:SOURce2?  
:CALCulate2:LIMit[2|3|5-12]:STATe <Bool>  
:CALCulate2:LIMit[2|3|5-12]:STATe?  
:CALCulate2:LIMit[2|3|5-12]:UPPer[:DATA] <NRf|DEFault|MINimum|MAXimum>  
:CALCulate2:LIMit[2|3|5-12]:UPPer[:DATA]? [DEFault|MINimum|MAXimum]  
:CALCulate2:LIMit[2|3|5-12]:UPPer:SOURce2 <NRf|NDN>  
:CALCulate2:LIMit[2|3|5-12]:UPPer:SOURce2?  
:CALCulate2:LIMit4:SOURce2 <NRf>  
:CALCulate2:LIMit4:SOURce2?  
:CALCulate2:NULL:ACQuire

:CALCulate2:NULL:OFFSet <NRf|DEFault|MINimum|MAXimum>  
:CALCulate2:NULL:OFFSet? [DEFault|MINimum|MAXimum]  
:CALCulate2:NULL:STATe <Bool>  
:CALCulate2:NULL:STATe?  
:CALCulate3:DATA?  
:CALCulate3:FORMAT MEAN|SDEViation|MAXimum|MINimum|PKPK  
:CALCulate3:FORMAT?  
:CONFigure:CURREnt[:DC]  
:CONFigure:RESistance  
:CONFigure:VOLTage[:DC]  
:CONFigure?  
:DISPlay:CNDisplay  
:DISPlay:DIGits <4|5|6|7|DEFault|MINimum|MAXimum>  
:DISPlay:DIGits? [DEFault|MINimum|MAXimum]  
:DISPlay:ENABLE <Bool>  
:DISPlay:ENABLE?  
:DISPlay[:WINDOW[1]]:TEXT:DATA <SPD>  
:DISPlay[:WINDOW[1]]:TEXT:DATA?  
:DISPlay[:WINDOW[1]]:TEXT:STATe <Bool>  
:DISPlay[:WINDOW[1]]:TEXT:STATe?  
:DISPlay:WINDOW2:TEXT:DATA <SPD>  
:DISPlay:WINDOW2:TEXT:DATA?  
:DISPlay:WINDOW2:TEXT:STATe <Bool>  
:DISPlay:WINDOW2:TEXT:STATe?  
:FETCh?  
:FORMAT:BORDer <NORMAL|SWAPPed>  
:FORMAT:BORDer?  
:FORMAT:ELEMents:CALCulate <CALC|TIME|STATus>  
:FORMAT:ELEMents:CALCulate?  
:FORMAT:ELEMents[:SENSe[1]] <VOLTage|CURREnt|RESistance|TIME|STATus>  
:FORMAT:ELEMents[:SENSe[1]]?

:FORMat:SOURce2 <ASCIi|HEXadecimal|OCTal|BINary>  
 :FORMat:SOURce2?  
 :FORMat:SREGister <ASCIi|HEXadecimal|OCTal|BINary>  
 :FORMat:SREGister?  
 .FORMat[:DATA] <ASCIi|REAL|SREal>[,NRf]  
 :FORMat[:DATA]?  
 :INITiate[:IMMediate]  
 :MEASure:CURRent[:DC]?  
 :MEASure:RESistance?  
 :MEASure:VOLTage[:DC]?  
 :MEASure?  
 :OUTPut[1]:INTerlock:TRIPped?  
 :OUTPut[1]:SMODe <HIMPedance|NORMal|ZERO|GURAd>  
 :OUTPut[1]:SMODe?  
 :OUTPut[1][:STATe] <Bool>  
 :OUTPut[1][:STATe]?  
 :READ?  
 [:SENSe[1]]:CURRent[:DC]:NPLCycles <NRf|DEFault|MINimum|MAXimum>  
 [:SENSe[1]]:CURRent[:DC]:NPLCycles? [DEFault|MINimum|MAXimum]  
 [:SENSe[1]]:CURRent[:DC]:PROTection[:LEVel] <NRf|DEFault|MINimum|MAXimum>  
 [:SENSe[1]]:CURRent[:DC]:PROTection[:LEVel]? DEFault|MINimum|MAXimum  
 [:SENSe[1]]:CURRent[:DC]:PROTection:TRIPped?  
 [:SENSe[1]]:CURRent[:DC]:RANGE:AUTO <Bool>  
 [:SENSe[1]]:CURRent[:DC]:RANGE:AUTO?  
 [:SENSe[1]]:CURRent[:DC]:RANGE:AUTO:LLIMit <NRf>  
 [:SENSe[1]]:CURRent[:DC]:RANGE:AUTO:LLIMit?  
 [:SENSe[1]]:CURRent[:DC]:RANGE:AUTO:ULIMit?  
 [:SENSe[1]]:CURRent[:DC]:RANGE[:UPPer] <NRf|DEFault|MINimum|MAXimum|UP|DOWN>  
 [:SENSe[1]]:CURRent[:DC]:RANGE[:UPPer]? [DEFault|MINimum|MAXimum]  
 [:SENSe[1]]:DATA[:LATest]?  
 [:SENSe[1]]:FUNCTION:CONCurrent <Bool>

[:SENSe[1]]:FUNCTION:CONCurrent?  
[:SENSe[1]]:FUNCTION:OFF <CURRent[:DC]|VOLTage[:DC]|RESistance>,,  
[:SENSe[1]]:FUNCTION:OFF:ALL  
[:SENSe[1]]:FUNCTION:OFF:COUNT?  
[:SENSe[1]]:FUNCTION:OFF?  
[:SENSe[1]]:FUNCTION[:ON] <CURRent[:DC]|VOLTage[:DC]|RESistance>,,  
[:SENSe[1]]:FUNCTION[:ON]:ALL  
[:SENSe[1]]:FUNCTION[:ON]:COUNT?  
[:SENSe[1]]:FUNCTION[:ON]?  
[:SENSe[1]]:FUNCTION:STATE? <“CURRent[:DC]”|“VOLTage[:DC]”|“RESistance”>  
[:SENSe[1]]:RESistance:MODE <MANual|AUTO>  
[:SENSe[1]]:RESistance:MODE?  
[:SENSe[1]]:RESistance:NPLCycles <NRf|DEFault|MINimum|MAXimum>  
[:SENSe[1]]:RESistance:NPLCycles? [DEFault|MINimum|MAXimum]  
[:SENSe[1]]:RESistance:OCOMPensated <Bool>  
[:SENSe[1]]:RESistance:OCOMPensated?  
[:SENSe[1]]:RESistance:RANGE:AUTO <Bool>  
[:SENSe[1]]:RESistance:RANGE:AUTO?  
[:SENSe[1]]:RESistance:RANGE:AUTO:LLIMit <NRf>  
[:SENSe[1]]:RESistance:RANGE:AUTO:LLIMit?  
[:SENSe[1]]:RESistance:RANGE:AUTO:ULIMit <NRf>  
[:SENSe[1]]:RESistance:RANGE:AUTO:ULIMit?  
[:SENSe[1]]:RESistance:RANGE[:UPPer] <NRf|DEFault|MINimum|MAXimum|UP|DOWN>  
[:SENSe[1]]:RESistance:RANGE[:UPPer]? [DEFault|MINimum|MAXimum]  
[:SENSe[1]]:VOLTage[:DC]:NPLCycles <NRf|DEFault|MINimum|MAXimum>  
[:SENSe[1]]:VOLTage[:DC]:NPLCycles? [DEFault|MINimum|MAXimum]  
[:SENSe[1]]:VOLTage[:DC]:PROTection[:LEVel] <NRf|DEFault|MINimum|MAXimum>  
[:SENSe[1]]:VOLTage[:DC]:PROTection[:LEVel]? <DEFault|MINimum|MAXimum>  
[:SENSe[1]]:VOLTage[:DC]:PROTection:TRIPped?  
[:SENSe[1]]:VOLTage[:DC]:RANGE:AUTO <Bool>  
[:SENSe[1]]:VOLTage[:DC]:RANGE:AUTO?

[:SENSe[1]]:VOLTage[:DC]:RANGE:AUTO:LLIMit <NRf>  
 [:SENSe[1]]:VOLTage[:DC]:RANGE:AUTO:LLIMit?  
 [:SENSe[1]]:VOLTage[:DC]:RANGE:AUTO:ULIMit?  
 [:SENSe[1]]:VOLTage[:DC]:RANGE[:UPPer] <NRf|DEFault|MINimum|MAXimum|UP|DOWN>  
 [:SENSe[1]]:VOLTage[:DC]:RANGE[:UPPer]? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CLEar:AUTO <Bool>  
 :SOURce[1]:CLEar:AUTO?  
 :SOURce[1]:CLEar[:IMMediate]  
 :SOURce[1]:CURRent:CENTER <NRf|DEFault|MINimum|MAXimum>  
 :SOURce[1]:CURRent:CENTER? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CURRent[:LEVel][:IMMediate][:AMPLitude] <NRf|DEFault|MINimum|MAXimum>  
 :SOURce[1]:CURRent[:LEVel][:IMMediate][:AMPLitude]? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CURRent[:LEVel]:TRIGgered[:AMPLitude] <NRf|DEFault|MINimum|MAXimum>  
 :SOURce[1]:CURRent[:LEVel]:TRIGgered[:AMPLitude]? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CURRent:MODE <FIXed|LIST|SWEep>  
 :SOURce[1]:CURRent:MODE?  
 :SOURce[1]:CURRent:RANGE <NRf|DEFault|MINimum|MAXimum|UP|DOWN>  
 :SOURce[1]:CURRent:RANGE? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CURRent:RANGE:AUTO <Bool>  
 :SOURce[1]:CURRent:RANGE:AUTO?  
 :SOURce[1]:CURRent:SPAN <NRf|DEFault|MINimum|MAXimum>  
 :SOURce[1]:CURRent:SPAN? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CURRent:STARt <NRf|DEFault|MINimum|MAXimum>  
 :SOURce[1]:CURRent:STARt? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CURRent:STEP <NRf|DEFault|MINimum|MAXimum>  
 :SOURce[1]:CURRent:STEP? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:CURRent:STOP <NRf|DEFault|MINimum|MAXimum>  
 :SOURce[1]:CURRent:STOP? [DEFault|MINimum|MAXimum]  
 :SOURce[1]:DELay <NRf|MINimum|MAXimum|DEFault>  
 :SOURce[1]:DELay? [MINimum|MAXimum|DEFault]  
 :SOURce[1]:DELay:AUTO <Bool>

:SOURce[1]:DELay:AUTO?  
:SOURce[1]:FUNCtion[:MODE]?  
:SOURce[1]:LIST:CURRent <NRf>{,<NRf>}..  
:SOURce[1]:LIST:CURRent?  
:SOURce[1]:LIST:CURRent:APPend <NRf>{,<NRf>}..  
:SOURce[1]:LIST:CURRent:POINts?  
:SOURce[1]:LIST:CURRent:STARt <NRf>  
:SOURce[1]:LIST:CURRent:STARt?  
:SOURce[1]:LIST:VOLTage <NRf>{,<NRf>}..  
:SOURce[1]:LIST:VOLTage?  
:SOURce[1]:LIST:VOLTage:APPend <NRf>{,<NRf>}..  
:SOURce[1]:LIST:VOLTage:POINts?  
:SOURce[1]:LIST:VOLTage:STARt <NRf>  
:SOURce[1]:LIST:VOLTage:STARt?  
:SOURce[1]:SWEep:DIRection <UP|DOWN>  
:SOURce[1]:SWEep:DIRection?  
:SOURce[1]:SWEep:POINts <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:SWEep:POINts? [DEFault|MINimum|MAXimum]  
:SOURce[1]:SWEep:RANGing <BEST|AUTO|FIXed>  
:SOURce[1]:SWEep:RANGing?  
:SOURce[1]:SWEep:SPACing <LINEar|LOGarithmic>  
:SOURce[1]:SWEep:SPACing?  
:SOURce[1]:VOLTage:CENTER <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:VOLTage:CENTER? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage[:LEVel][,:IMMEDIATE][,:AMPLitude] <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:VOLTage[:LEVel][,:IMMEDIATE][,:AMPLitude]? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage[:LEVel]:TRIGgered[:AMPLitude] <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:VOLTage[:LEVel]:TRIGgered[:AMPLitude]? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage:MODE <FIXed|LIST|SWEep>  
:SOURce[1]:VOLTage:MODE?  
:SOURce[1]:VOLTage:PROTection[:LEVel] <NRf|NONE|DEFault|MINimum|MAXimum>

:SOURce[1]:VOLTage:PROTection[:LEVel]? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage:PROTection:TRIPped?  
:SOURce[1]:VOLTage:RANGE <NRf|DEFault|MINimum|MAXimum|UP|DOWN>  
:SOURce[1]:VOLTage:RANGE? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage:RANGE:AUTO <Bool>  
:SOURce[1]:VOLTage:RANGE:AUTO?  
:SOURce[1]:VOLTage:SPAN <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:VOLTage:SPAN? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage:STARt <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:VOLTage:STARt? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage:STEP <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:VOLTage:STEP? [DEFault|MINimum|MAXimum]  
:SOURce[1]:VOLTage:STOP <NRf|DEFault|MINimum|MAXimum>  
:SOURce[1]:VOLTage:STOP? [DEFault|MINimum|MAXimum]  
:SOURce2:BSIze <3|4>  
:SOURce2:BSIze?  
:SOURce2:CLEAR:AUTO <Bool>  
:SOURce2:CLEAR:AUTO?  
:SOURce2:CLEAR:AUTO:DELay <NRf|DEFault|MINimum|MAXimum>  
:SOURce2:CLEAR:AUTO:DELay? [DEFault|MINimum|MAXimum]  
:SOURce2:CLEAR[:IMMEdiate]  
:SOURce2:TTL[:LEVel]:ACTual?  
:SOURce2:TTL[:LEVel][[:DEFault]] <NRf|NDN>  
:SOURce2:TTL[:LEVel][[:DEFault]]?  
:SOURce2:TTL4:BSTate <Bool>  
:SOURce2:TTL4:BSTate?  
:SOURce2:TTL4:MODE <EOTest|BUSY>  
:SOURce2:TTL4:MODE?  
:STATus:MEASurement:CONDition?  
:STATus:MEASurement:ENABLE <NDN|NRf>  
:STATus:MEASurement:ENABLE?

:STATUs:MEASurement[:EVENT]?

:STATUs:OPERation:CONDition?

:STATUs:OPERation:ENABLE <NDN|NRf>

:STATUs:OPERation:ENABLE?

:STATUs:OPERation[:EVENT]?

:STATUs:PRESet

:STATUs:QUEstionable:CONDition?

:STATUs:QUEstionable:ENABLE <NDN|NRf>

:STATUs:QUEstionable:ENABLE?

:STATUs:QUEstionable[:EVENT]?

:STATUs:QUEue:CLEar

:STATUs:QUEue[:NEXT]?

:SYSTem:BEEPer:STATe <Bool>

:SYSTem:BEEPer:STATe?

:SYSTem:CLEar

:SYSTem:ERRor:ALL?

:SYSTem:ERRor:CODE:ALL?

:SYSTem:ERRor:CODE[:NEXT]?

:SYSTem:ERRor:COUNt?

:SYSTem:ERRor[:NEXT]?

:SYSTem:LFRequency <50|60>

:SYSTem:LFRequency?

:SYSTem:POSetup <RST|PRESet|SAV0|SAV1|SAV2|SAV3|SAV4>

:SYSTem:POSetup?

:SYSTem:PRESet

:SYSTem:RSENse <Bool>

:SYSTem:RSENse?

:SYSTem:TIME?

:SYSTem:TIME:RESet

:SYSTem:TIME:RESet:AUTO <Bool>

:SYSTem:TIME:RESet:AUTO?

---

```
:SYSTem:VERSion?
:TRACe:CLEar
:TRACe:DATA?
:TRACe:FEED <SENSe[1]|CALCulate[1]|CALCulate2>
:TRACe:FEED?
:TRACe:FEED:CONTrol <NEXT|NEVer>
:TRACe:FEED:CONTrol?
:TRACe:FREE?
:TRACe:POINts <NR1|MINimum|MAXimum|DEFault>
:TRACe:POINts? [MINimum|MAXimum|DEFault]
:TRACe:POINts:ACTual?
:TRACe:TStamp:FORMat <ABSolute|DELTa>
:TRACe:TStamp:FORMat?
:TRIGger[:SEQUence[1]]:COUNT <NRf|DEFault|MINimum|MAXimum>
:TRIGger[:SEQUence[1]]:COUNT? [DEFault|MINimum|MAXimum]
:TRIGger[:SEQUence[1]]:DELay <NRf|DEFault|MINimum|MAXimum>
:TRIGger[:SEQUence[1]]:DELay? [DEFault|MINimum|MAXimum]
:TRIGger[:SEQUence[1]]:SOURce?
:TRIGger[:SEQUence[1]][:TCONFIGure]:DIRection <SOURCE|ACCeptor>
:TRIGger[:SEQUence[1]][:TCONFIGure]:DIRection?
```

## 5.2 SMM3000X 部分支持的常规命令

可能需要更改参数值。

```
:SYSTem:BEEPer[:IMMediate] <NRf>
```

参数 TLINK 和 BSTest 不起作用。不报错误。

```
:ARM[:SEQUence[1]][:LAYer[1]]:SOURce<IMMediate|TLINK|TIMer|MANual|BUS|NSTest|
PSTest|BSTest>
:TRIGger[:SEQUence[1]]:SOURce<IMMediate|TLINK|TIMer|MANual|BUS|NSTest|PSTest|BSTest>
```

忽略。不会引起任何动作或响应

```
:ROUTe:TERMinals <FRONT|REAR>
[:SENSe[1]]:AVERage:COUNt <NRf|DEFault|MINimum|MAXimum>
[:SENSe[1]]:AVERage:TCONtrol <REPeat|MOVing>
:SOURce[1]:PULSe:DELay <NRf>
:SOURce[1]:PULSe:DELay?
:SOURce[1]:PULSe:WIDTH <NRf>
:SOURce[1]:PULSe:WIDTH?
:SYSTem:AZERo:CACHing:REFresh
:SYSTem:AZERo:CACHing:RESET
:SYSTem:AZERo:CACHing[:STATe] <Bool>
:SYSTem:AZERo:STATe <Bool>
:SYSTem:CCheck:RESistance <NRf>
:SYSTem:GUARD <OHMS|CABLE>
:SYSTem:KEY <NR1>
:SYSTem:LOCal
:SYSTem:MEMOry:INITialize
:SYSTem:MEP:HOLDoff
:SYSTem:RWLock <Bool>
:TRIGger:CLEAr
:TRIGger:SEQUence2:SOURce <name>
:TRIGger:SEQUence2:TOUT <NRf|DEFault|MINimum|MAXimum>
```

始终返回特定值。

```
:ARM[:SEQUence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:LINE? (returns 1)
:ARM[:SEQUence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:OLINE? (returns 2)
:ARM[:SEQUence[1]][:LAYer[1]][:TCONfigure][:ASYNchronous]:OUTPut? (returns NONE)
:CALCulate2:CLIMits:FAIL:SMLocation? (returns NEXT)
:CALCulate2:CLIMits:PASs:SMLocation? (returns NEXT)
:DISPLAY[:WINDow[1]]:ATTRibutes? (returns 20 zeros)
:DISPLAY:WINDow2:ATTRibutes? (returns 32 zeros)
```

:OUTPut[1]:INTerlock:STATe? (returns 1)  
:ROUTe:TERMinals? (returns FRON)  
[:SENSe[1]]:AVERage:COUNt? [DEFault|MINimum|MAXimum] (returns 10)  
[:SENSe[1]]:AVERage:TCONtrol? (retruns REP)  
[:SENSe[1]]:AVERage:STATe? (returns 1)  
[:SENSe[1]]:CURRent[:DC]:PROTection:RSYNchronize? (returns 1)  
[:SENSe[1]]:VOLTage[:DC]:PROTection:RSYNchronize? (returns 1)  
:SOURce[1]:CLEar:AUTO:MODE? (returns TCO)  
:SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACTOR? (returns 1.0)  
:SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACTOR:STATe? (returns 0)  
:SOURce[1]:FUNCTION:SHAPe? (returns DC)  
:SOURce[1]:MEMory:POINts? (returns 1)  
:SOURce[1]:MEMory:STARt? (returns 1)  
:SOURce[1]:SOAK? (returns 0)  
:SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACTOR? (returns 1.0)  
:SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACTOR:STATe? (returns 0)  
:STATus:QUEue:DISable? (returns (+0))  
:STATus:QUEue:ENABLE? (returns (-440:-100,+111:+954))  
:SYSTem:AZERo:CACHing:NPLCycles? (returns 0)  
:SYSTem:AZERo:CACHing[:STATe]? (returns 0)  
:SYSTem:AZERo:STATe? (returns 0)  
:SYSTem:CCheck? (returns 0)  
:SYSTem:CCheck:RESistance? (returns 50)  
:SYSTem:GUARD? (returns CABL)  
:SYSTem:KEY? (returns 0)  
:SYSTem:LFREquency:AUTO? (returns 0)  
:SYSTem:MEP[:STATe]? (returns 1)  
:SYSTem:MEP:HOLDoff? (returns 0)  
:SYSTem:RCMode? (returns SING)  
:SYSTem:RWLock? (returns 0)  
:TRIGger[:SEQUence[1]][:TCONfigure][:ASYNchronous]:ILINe? (returns 1)

:TRIGger[:SEQUence[1]][:TCOnfigure][:ASYNchronous]:INPut? (returns NONE)  
 :TRIGger[:SEQUence[1]][:TCOnfigure][:ASYNchronous]:OLINe? (returns 2)  
 :TRIGger[:SEQUence[1]][:TCOnfigure][:ASYNchronous]:OUTPut? (returns NONE)  
 :TRIGger:SEQUence2:SOURce? (returns IMM)  
 :TRIGger:SEQUence2:TOUT? [DEFault|MINimum|MAXimum] (returns 0.0)

## 5.3 SMM3000X 不支持的常规命令

:ARM[:SEQUence[1]][:LAYer[1]][:TCOnfigure][:ASYNchronous]:ILINe <1|2|3|4>  
 :ARM[:SEQUence[1]][:LAYer[1]][:TCOnfigure][:ASYNchronous]:OLINe <1|2|3|4>  
 :ARM[:SEQUence[1]][:LAYer[1]][:TCOnfigure][:ASYNchronous]:OUTPut <TERTer|TEXit|NONE>  
 :CALCulate2:CLIMits:FAIL:SMLocation <NR1>|NEXT  
 :CALCulate2:CLIMits:PASS:SMLocation <NR1>|NEXT  
 :DISPlay[:WINDOW[1]]:DATA?  
 :DISPlay:WINDOW2:DATA?  
 :OUTPut[1]:INTerlock:STATe <Bool>  
 [:SENSe[1]]:AVERage:STATe <Bool>  
 [:SENSe[1]]:CURRent[:DC]:PROTection:RSYNchronize <Bool>  
 [:SENSe[1]]:VOLTage[:DC]:PROTection:RSYNchronize <Bool>  
 :SOURce[1]:CLEar:AUTO:MODE <ALWays|TCOut>  
 :SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACtor <NRf>  
 :SOURce[1]:CURRent[:LEVel]:TRIGgered:SFACtor:STATe <Bool>  
 :SOURce[1]:FUNCTION[:MODE] <VOLTage|CURRent|MEMORY>  
 :SOURce[1]:FUNCTION:SHAPe <DC|PULSE>  
 :SOURce[1]:MEMORY:POINts <NR1>  
 :SOURce[1]:MEMORY:RECall <NR1>  
 :SOURce[1]:MEMORY:SAVE <NR1>  
 :SOURce[1]:MEMORY:STARt <NR1>  
 :SOURce[1]:SOAK <NRf>  
 :SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACtor <NRf>  
 :SOURce[1]:VOLTage[:LEVel]:TRIGgered:SFACtor:STATe <Bool>

:STATus:QUEue:DISable (NR1 list)  
:STATus:QUEue:ENABLE (NR1 list)  
:SYSTem:CCheck ON|OFF  
:SYSTem:LFRequency:AUTO <Bool>  
:SYSTem:MEP[:STATe]  
:SYSTem:RCMode <SiNGle|MULTiple>  
:TRIGger[:SEQUence[1]][:TCOnfigure][:ASYNchronous]:ILINe <1|2|3|4>  
:TRIGger[:SEQUence[1]][:TCOnfigure][:ASYNchronous]:INPut <SOURce|DELay|SENSe|NONE>  
:TRIGger[:SEQUence[1]][:TCOnfigure][:ASYNchronous]:OLINe <1|2|3|4>  
:TRIGger[:SEQUence[1]][:TCOnfigure][:ASYNchronous]:OUTPut<SOURce|DELay|SENSe|NONE>



## 关于鼎阳

鼎阳科技 (SIGLENT) 是通用电子测试测量仪器领域的行业领军企业，A 股上市公司。

2002 年，鼎阳科技创始人开始专注于示波器研发，2005 年成功研制出鼎阳第一款数字示波器。历经多年发展，鼎阳产品已扩展到数字示波器、手持示波表、函数/任意波形发生器、频谱分析仪、矢量网络分析仪、射频/微波信号源、台式万用表、直流电源、电子负载等基础测试测量仪器产品，是全球极少数能够同时研发、生产、销售数字示波器、信号发生器、频谱分析仪和矢量网络分析仪四大通用电子测试测量仪器主力产品的厂家之一，国家重点“小巨人”企业。同时也是国内主要竞争对手中极少数同时拥有这四大主力产品并且四大主力产品全线进入高端领域的厂家。公司总部位于深圳，在美国克利夫兰、德国奥格斯堡、日本东京成立了子公司，在成都成立了分公司，产品远销全球 80 多个国家和地区，SIGLENT 已经成为全球知名的测试测量仪器品牌。

## 联系我们

深圳市鼎阳科技股份有限公司  
全国免费服务热线：400-878-0807  
网址：[www.siglent.com](http://www.siglent.com)

## 声明

 SIGLENT 鼎阳 是深圳市鼎阳科技股份有限公司的注册商标，事先未经过允许，不得以任何形式或通过任何方式复制本手册中的任何内容。  
本资料中的信息代替原先的此前所有版本。技术数据如有变更，恕不另行通告。

## 技术许可

对于本文档中描述的硬件和软件，仅在得到许可的情况下才会提供，并且只能根据许可进行使用或复制。

