

SDG7000A 系列 函数/任意波形发生器

编程手册

CN03A

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

目录

1	编程概述	10
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
2	SCPI 语言简介	16
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	命令&查询表	17
3	命令与查询	19
3.1	IEEE488.2 通用命令介绍	19
3.1.1	*IDN	19
3.1.2	*OPC	20
3.1.3	*RST	20
3.2	输出命令	21
3.3	噪声叠加命令	22
3.4	幅度限制命令	22
3.5	基本波形命令	22
3.6	调制波形命令	25
3.7	扫描波形命令	30
3.7.1	<channel>:SWEep <switch>	30
3.7.2	<channel>:SWEep:TYPE <type>	30

3.7.3	<channel>:SWEep:SOURce <src>	31
3.7.4	<channel>:SWEep:FMODe <mode>	31
3.7.5	<channel>:SWEep:AMODe <mode>	32
3.7.6	<channel>:SWEep:FSNumber <value>	32
3.7.7	<channel>:SWEep:ASNumber <value>	33
3.7.8	<channel>:SWEep:TIME <value>	33
3.7.9	<channel>:SWEep:SHTime <value>	34
3.7.10	<channel>:SWEep:EHTime <value>	34
3.7.11	<channel>:SWEep:RTIME <value>	35
3.7.12	<channel>:SWEep:SFRequency <value>	35
3.7.13	<channel>:SWEep:EFRequency <value>	36
3.7.14	<channel>:SWEep:CFRequency <value>	36
3.7.15	<channel>:SWEep:FSPan <value>	37
3.7.16	<channel>:SWEep:SAMPLitude <value>	37
3.7.17	<channel>:SWEep:EAMPLitude <value>	38
3.7.18	<channel>:SWEep:CAMPLitude <value>	38
3.7.19	<channel>:SWEep:ASPan <value>	39
3.7.20	<channel>:SWEep:FDIRection <direction>	39
3.7.21	<channel>:SWEep:ADIRection <direction>	40
3.7.22	<channel>:SWEep:FSYMMetry <value>	40
3.7.23	<channel>:SWEep:ASYMMetry <value>	41
3.7.24	<channel>:SWEep:TOUT <switch>	41
3.7.25	<channel>:SWEep:EDGE <polarity>	42
3.7.26	<channel>:SWEep:MTRigger	42
3.7.27	<channel>:SWEep:FMARker <switch>	42
3.7.28	<channel>:SWEep:MFRequency <value>	43
3.7.29	<channel>:SWEep:MSNumber <value>	43
3.7.30	<channel>:SWEep:CARRier:WTYPE <wave>	44
3.7.31	<channel>:SWEep:CARRier:FREQuency <value>	44

3.7.32	<code><channel>:SWEp:CARRier:PHASe <value></code>	45
3.7.33	<code><channel>:SWEp:CARRier:PAMPlitude <value></code>	45
3.7.34	<code><channel>:SWEp:CARRier:RAMPlitude <value></code>	46
3.7.35	<code><channel>:SWEp:CARRier:OFFSet <value></code>	46
3.7.36	<code><channel>:SWEp:CARRier:SYMMetry <value></code>	47
3.7.37	<code><channel>:SWEp:CARRier:DUTY <value></code>	47
3.8	脉冲串命令	48
3.9	参数复制命令	50
3.10	任意波形命令	51
3.10.1	任意波形切换波形命令	51
3.10.2	任意波形标记设置命令	54
3.11	同步命令	55
3.12	同相位命令	55
3.13	语言命令	56
3.14	配置命令	56
3.15	日期和时间命令	57
3.16	开机上电模式命令	57
3.17	按键命令	58
3.18	蜂鸣器命令	58
3.19	通道触发源设置命令	58
3.20	屏幕保护命令	59
3.21	时钟源命令	59
3.22	计数器命令	60
3.22.1	<code>:SENSe:COUNTer:CLEAr</code>	62
3.22.2	<code>:SENSe:COUNTer:FREQuency:MEASure[:type]</code>	62
3.22.3	<code>:SENSe:COUNTer:FREQuency:RFREQuency</code>	62
3.22.4	<code>:SENSe:COUNTer:FREQuency?</code>	63
3.22.5	<code>:SENSe:COUNTer:FREQuency:SNUMBer?</code>	63
3.22.6	<code>:SENSe:COUNTer:FREQuency:FDEVIation?</code>	63

3.22.7	:SENSe:COUNTer:FREQuency:MEAN?	64
3.22.8	:SENSe:COUNTer:FREQuency:MEAN:FDEVIation?	64
3.22.9	:SENSe:COUNTer:FREQuency:MAX?	64
3.22.10	:SENSe:COUNTer:FREQuency:MAX:FDEVIation?	65
3.22.11	:SENSe:COUNTer:FREQuency:MIN?	65
3.22.12	:SENSe:COUNTer:FREQuency:MIN:FDEVIation?	65
3.22.13	:SENSe:COUNTer:FREQuency:SDEVIation?	66
3.22.14	:SENSe:COUNTe:FREQuency:SDEVIation:FDEVIation?	66
3.22.15	:SENSe:COUNTer:FREQuency:PERiod?	66
3.22.16	:SENSe:COUNTer:FREQuency:PERiod:MEAN?	67
3.22.17	:SENSe:COUNTer:FREQuency:PERiod:MAX?	67
3.22.18	:SENSe:COUNTer:FREQuency:PERiod:MIN?	67
3.22.19	:SENSe:COUNTer:FREQuency:PERiod:SDEVIation?	68
3.22.20	:SENSe:COUNTer:FREQuency:DUTY?	68
3.22.21	:SENSe:COUNTer:FREQuency:DUTY:MEAN?	68
3.22.22	:SENSe:COUNTer:FREQuency:DUTY:MAX?	69
3.22.23	:SENSe:COUNTer:FREQuency:DUTY:MIN?	69
3.22.24	:SENSe:COUNTer:FREQuency:DUTY:SDEVIation?	69
3.22.25	:SENSe:COUNTer:TOTalizer:GATE:STATE	70
3.22.26	:SENSe:COUNTer:TOTalizer:EDGE	70
3.22.27	:SENSe:COUNTer:TOTalizer:GATE:MODE	70
3.22.28	:SENSe:COUNTer:TOTalizer:GOLarity:GATE:POLarity	71
3.22.29	:SENSe:COUNTer:TOTalizer:GOLarity:GATE:EDGE	71
3.22.30	:SENSe:COUNTer:TOTalizer?	71
3.23	反相命令	72
3.24	数字滤波器命令	72
3.25	通道耦合命令	73
3.26	过压保护命令	74
3.27	过流保护命令	74

3.28	过压过流状态查询	75
3.29	OUTPUT 时滞命令	75
3.30	保存列表命令	75
3.31	任意波数据命令	79
3.32	序列波命令	81
3.32.1	<channel>:ARBMode <Mode>	81
3.32.2	<channel>:SEQuence <switch>	81
3.32.3	<channel>:SEQuence:STATe <state>	82
3.32.4	<channel>:SEQuence:BURSt <count>	82
3.32.5	<channel>:SEQuence:RMODe <mode>	83
3.32.6	<channel>:TRIGger[:SEQuence]:SOURce	83
3.32.7	<channel>:TRIGger[:SEQuence][:IMMediate]	84
3.32.8	<channel>:TRIGger:TIMer	84
3.32.9	<channel>:TRIGger[:SEQuence]:SLOPe	84
3.32.10	<channel>:SEQuence:COUNT	85
3.32.11	<channel>:SEQuence:DEFAult	85
3.32.12	<channel>:SEQuence:NEW	85
3.32.13	<channel>:SEQuence:SEGMent<x>:WAVeform	86
3.32.14	<channel>:SEQuence:SEGMent<x>:REPeat:COUNT	86
3.32.15	<channel>:SEQuence:SEGMent<x>:TSOURce	87
3.32.16	<channel>:SEQuence:SEGMent<x>:AMPLitude	87
3.32.17	<channel>:SEQuence:SEGMent<x>:OFFset	88
3.32.18	<channel>:SEQuence:SEGMent<x>:VOLTage:HIGH	88
3.32.19	<channel>:SEQuence:SEGMent<x>:VOLTage:LOW	89
3.32.20	<channel>:SEQuence:SEGMent<x>:LENGth	89
3.32.21	<channel>:SEQuence:SEGMent<x>:DELEte	90
3.32.22	<channel>:SEQuence:SEGMent<x>:INSErt	90
3.32.23	<channel>:SEQuence:SEGMent<x>:MARKer:SWITCh	90
3.32.24	<channel>:SEQuence:SEGMent<x>:MARKer:POS	91

3.32.25	<channel>:SEquence:OUTList.....	91
3.32.26	<channel>:SEquence:SCALE.....	92
3.32.27	<channel>:SEquence:INCRaising.....	92
3.32.28	<channel>:SEquence:DECRaising.....	92
3.32.29	<channel>:SEquence:RECall.....	93
3.32.30	<channel>:SEquence:SAVe.....	93
3.33	数字通道命令.....	94
3.33.1	DIG:SRATe.....	94
3.33.2	DIG:PERiod.....	94
3.33.3	DIG:CHANnel<x>:STATe.....	94
3.33.4	DIG:OUTPut.....	95
3.33.5	DIG:WAVEform.....	95
3.33.6	DIG:RMODE <mode>.....	97
3.33.7	DIG:TRIGger:SOURce <type>.....	98
3.33.8	DIG:TRIGger:SLOPe <type>.....	98
3.33.9	DIG:TRIGger:TIMer <value>.....	99
3.33.10	DIG:BURSt:COUNT <value>.....	99
3.33.11	DIG:HOLD:VALUe <value>.....	99
3.33.12	DIG:TRIGger.....	100
3.34	跳频命令.....	101
3.34.1	<channel>:FHOP:SWITCh.....	101
3.34.2	<channel>:FHOP:STATe.....	101
3.34.3	<channel>:FHOP:TYPE.....	102
3.34.4	<channel>:FHOP:TIME.....	102
3.34.5	<channel>:FHOP:SFREquency.....	102
3.34.6	<channel>:FHOP:EFREquency.....	103
3.34.7	<channel>:FHOP:FSTep.....	103
3.34.8	<channel>:FHOP:RPATtern.....	104
3.34.9	<channel>:FHOP:RLPAttern.....	104

3.34.10	<channel>:FHOP:ALStAt	105
3.34.11	<channel>:FHOP:AFList	105
3.34.12	<channel>:FHOP:DFList	105
3.34.13	<channel>:FHOP:CFList	106
3.34.14	<channel>:FHOP:MFList	106
3.34.15	<channel>:FHOP:AOList	106
3.34.16	<channel>:FHOP:DOList	106
3.34.17	<channel>:FHOP:COList	107
3.34.18	<channel>:FHOP:MOList	107
3.34.19	<channel>:FHOP:AAList	107
3.34.20	<channel>:FHOP:MAList	108
3.34.21	<channel>:FHOP:DAList	108
3.34.22	<channel>:FHOP:CAList	108
3.34.23	<channel>:FHOP:LFList	108
3.34.24	<channel>:FHOP:SFList	109
3.34.25	<channel>:FHOP:LOList	109
3.34.26	<channel>:FHOP:SOList	109
3.34.27	<channel>:FHOP:LAList	110
3.34.28	<channel>:FHOP:SAList	110
3.35	虚拟键命令	111
3.36	IP 命令	113
3.37	子网掩码命令	113
3.38	网关命令	114
3.39	采样率命令	114
3.40	非线性补偿命令	115
3.41	谐波命令	115
3.42	波形合并命令	117
3.43	模式选择命令	117
3.44	多设备同步命令	117

3.45	IQ 命令	119
3.45.1	IQ:WAVeinfo?	119
3.45.2	IQ:CENTerfreq	119
3.45.3	IQ:SAMPLerate	119
3.45.4	IQ:SYMBolrate	120
3.45.5	IQ:AMPLitude	120
3.45.6	IQ:IQADjustment:GAIN	120
3.45.7	IQ:IQADjustment:IOffset	121
3.45.8	IQ:IQADjustment:QOffset	121
3.45.9	IQ:IQADjustment:QSkew	121
3.45.10	IQ:TRIGger:SOURce	122
3.45.11	IQ:WAVEload:BUILtin	122
3.45.12	IQ:WAVEload:USERstored	123
3.45.13	IQ:FrequencySampling	123
3.46	文件管理命令	124
3.46.1	MMEMory:DELeTe	124
3.46.2	MMEMory:RDIRectory	124
3.46.3	MMEMory:MDIRectory	124
3.46.4	MMEMory:CATalog	124
3.46.5	MMEMory:COPIY	125
3.46.6	MMEMory:MOVE	125
3.46.7	MMEMory:SAVE:XML	126
3.46.8	MMEMory:LOAD:XML	126
3.46.9	MMEMory:TRANsfer	127
4	波形格式	128
4.1	BIN	128
4.2	CSV/DAT	129
4.3	MAT	131
4.4	AWG	134

4.5	HOP.....	136
4.6	WAV/ARB.....	137
5	编程示例.....	138
5.1	VISA 应用示例.....	138
5.1.1	VC++示例.....	138
5.1.2	VB 示例.....	147
5.1.3	MATLAB 示例.....	154
5.1.4	LabVIEW 示例.....	157
5.1.5	Python2 示例.....	159
5.1.6	Python3 示例.....	162
5.1.7	Python3(Digital)示例.....	164
5.2	SOCKETS 应用示例.....	167
5.2.1	Python 示例.....	167
6	索引.....	170

1 编程概述

用户可以通过使用信号源的 USB、LAN 或 GPIB 端口，并结合 NI-VISA 和程序语言，远程控制信号源。基于 LAN 端口，SDG 支持 VXI-11、Sockets 和 Telnet 通信协议。本节介绍了如何建立 SDG 系列函数/任意波形发生器和计算机之间的通信，同时介绍如何远程控制信号源。

1.1 建立通信

1.1.1 NI-VISA 的安装

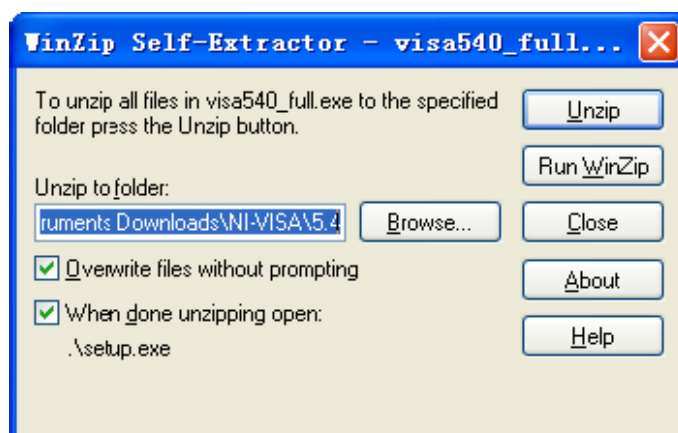
在编程之前，请确保正确安装 NI-VISA 软件的最新版本。

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）：

- a. 下载合适版本的 NI-VISA（推荐运行引擎版）
- b. 双击 visa540_full.exe，弹出对话框如下：

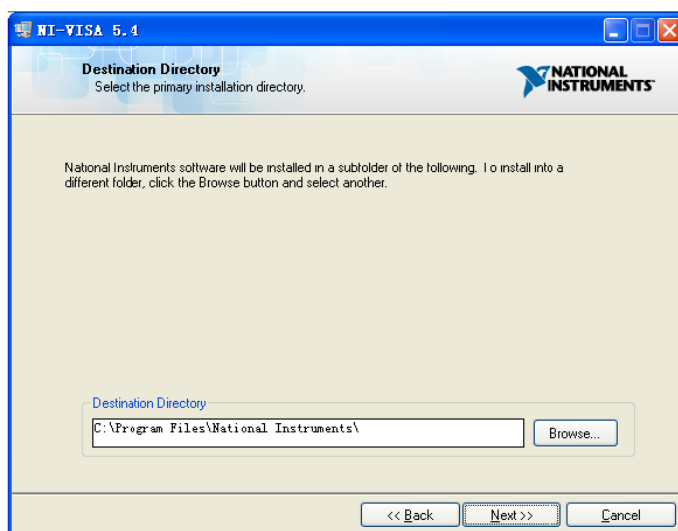


- c. 点击 Unzip 解压文件，当解压完成后，安装程序将自动执行。若你的计算机需要安装.NET

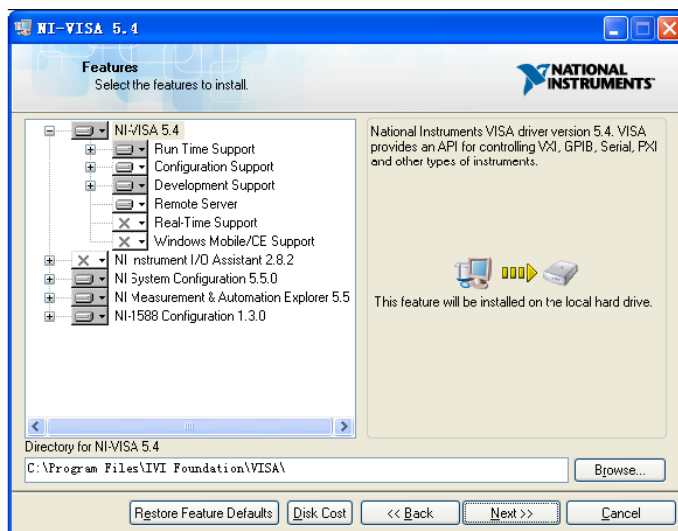
Framework4，则在安装过程会自动安装.NET Framework4。



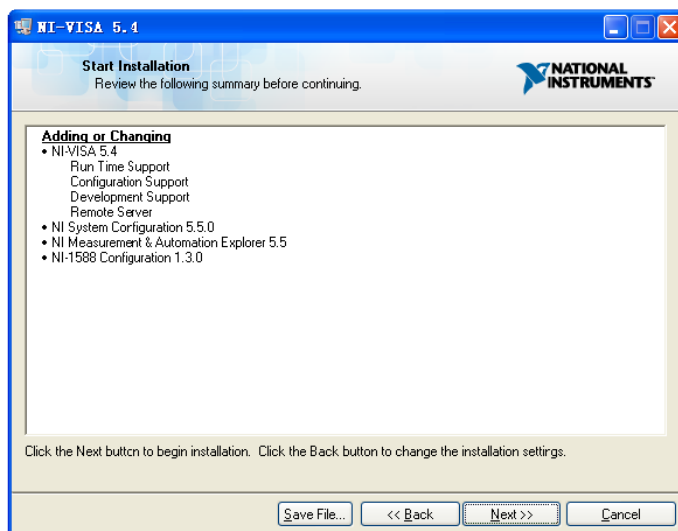
d. NI-VISA 安装对话框如上图所示，点击 Next 开始安装过程。



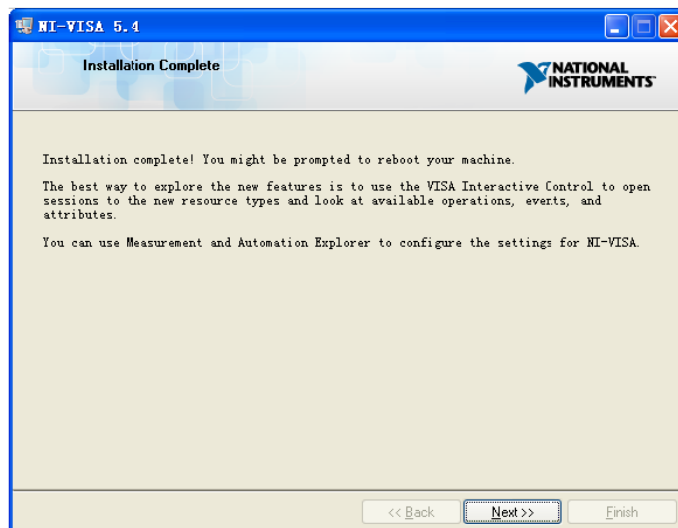
e. 设置安装路径，默认路径为“C:\Program Files\National Instruments\”。你也可以修改安装路径。点击 Next，对话框如下图所示。



- 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，对话框如下图所示：



- g. 点击 Next 开始安装：

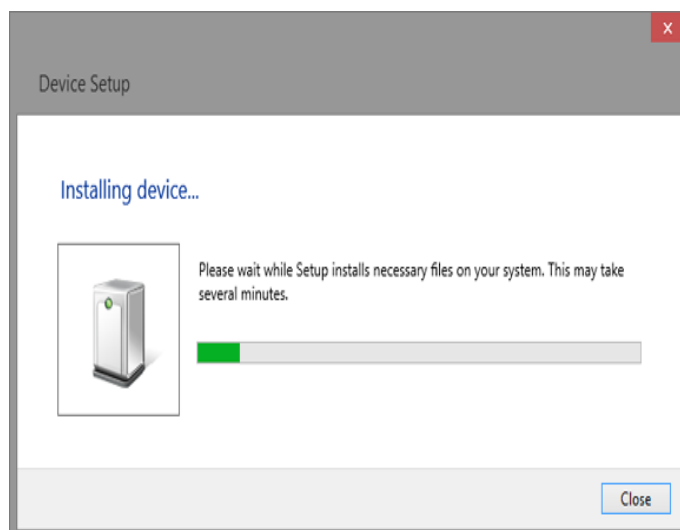


- h. 安装完成后，重启电脑。

1.1.2 连接仪器

根据具体信号源机型，函数/任意波形发生器能够通过 USB、LAN 端口或 GPIB（选配）接口连接计算机。

使用 USB 线将函数/任意波形发生器的 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，打开信号源后，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任意波形发生器。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

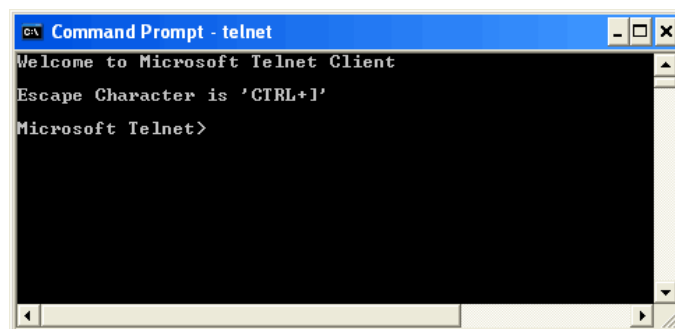
NI-MAX 是由 NI 公司创建和维护的程序。它为 VXI、LAN、USB、GPIB 和串行通信提供基础的远程控制接口。用户可以通过 NI-MAX 发送 SCPI 命令远程控制信号源。

1.2.3 通过 Telnet 发送 SCPI 命令

Telnet 提供一种通过 LAN 端口与信号源通信的方式。Telnet 协议支持从计算机向信号源发送 SCPI 命令，该方式类似于通过 USB 与信号源通信。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：

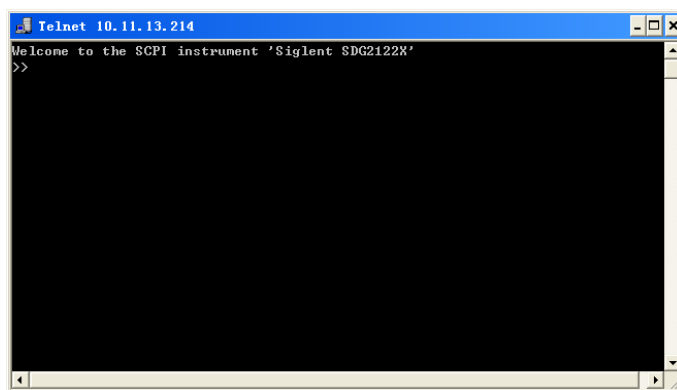
1. 在计算机桌面，点击开始>所有程序>附件>命令提示符
2. 在命令提示符窗口，输入 Telnet
3. 按下 Enter 键。将弹出 Telnet 显示窗



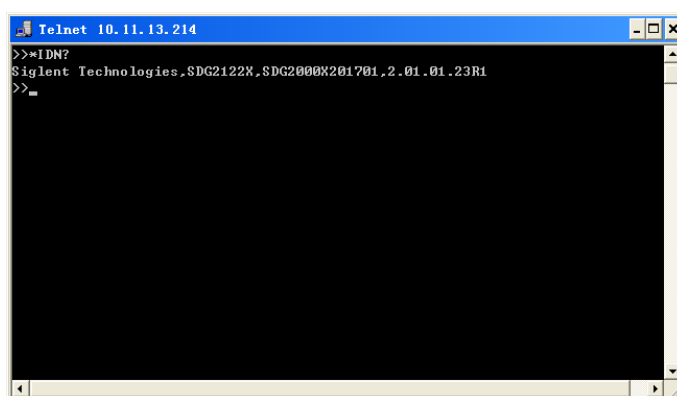
4. 在 Telnet 命令行，键入：

```
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. 通过在提示符后键入 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 SDG 系列产品。它可以减少编程的复杂度。

SOCKET 地址	IP 地址+端口号
IP 地址	SDG IP 地址
端口号	5025

相关内容，请查阅第 5.2 节“Sockets 应用示例”。

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 SDG7000A 系列的函数/任意波形发生器上。

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，至少要选择一個参数

[] 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 命令&查询表

短格式	长格式	子系统	命令/查询的作用
*IDN	*IDN	系统	获取设备 id
*OPC	*OPC	系统	设置或获取事件状态寄存器 (ESR) 的 OPC 位。
*RST	*RST	系统	设为默认设置。
CHDR	COMM_HEADER	信号	设置或获取命令格式
OUTP	OUTPUT	信号	设置或获取输出状态。
BSWV	BASIC_WAVE	信号	设置或获取基本波形参数。
MDWV	MODULATEWAVE	信号	设置或获取调制参数。
SWWV	SWEEPWAVE	信号	设置或获取扫频参数。
BTWV	BURSTWAVE	信号	设置或获取脉冲串功能参数
PACP	PARACOPY	信号	将参数从通道复制到另一通道
ARWV	ARBWAVE	数据	设置任意波类型
SYNC	SYNC	信号	设置或获取同步信号
NBFM	NUMBER_FORMAT	系统	设置或获取数据格式
LAGG	LANGUAGE	系统	设置或获取语言
SCFG	SYS_CFG	系统	设置或获取上电开机模式
BUZZ	BUZZER	系统	设置或获取蜂鸣器状态
SCSV	SCREEN_SAVE	系统	设置或获取屏幕保护状态
ROSC	ROSCILLATOR	信号	设置或获取时钟源的状态
FCNT	FREQCOUNTER	信号	设置或获取频率计参数
INVT	INVERT	信号	设置或获取当前通道的极性
COUP	COUPLING	信号	设置或获取通道耦合参数
VOLTPRT	VOLTPRT	系统	设置或获取过压保护的状态
STL	STORELIST	信号	列出所有存储波形
WVDT	WVDT	信号	设置或获取任意波波形数据
VKEY	VIRTUALKEY	系统	设置虚拟键
SYST:COMM:LAN:IPAD	SYSTEM:COMMUNICATE:LAN:IPADDRESS	系统	设置或获取设备 IP 地址

短格式	长格式	子系统	命令/查询的作用
SYST:COMM:LAN:SMAS	SYSTEM:COMMUNICATE:LAN:SMASK	系统	设置或获取设备子网掩码
SYST:COMM:LAN:GAT	SYSTEM:COMMUNICATE:LAN:GATEWAY	系统	设置或获取设备网关
SRATE	SAMPLERATE	信号	设置或获取采样频率，仅在逐点输出（TrueArb）模式下使用。
HARM	HARMonic	信号	设置或获取谐波信息
CMBN	CoMBiNe	信号	设置或获取波形合并信息
MODE	MODE	信号	设置或获取波形的相位模式
IQ:CENT	IQ:CENTerfreq	信号	设置或获取 I/Q 调制的中心频率
IQ:SAMP	IQ:SAMPlerate	信号	设置 I/Q 采样率
IQ:SYMB	IQ:SYMBolrate	信号	设置 I/Q 符号率
IQ:AMPL	IQ:AMPLitude	信号	设置 I/Q 幅度
IQ:IQAD:GAIN	IQ:IQADjustment:GAIN	信号	在保持复合状态下调节 I 与 Q 比例
IQ:IQAD:I OFF	IQ:IQADjustment:I OFFset	信号	调节 I 通道的偏置
IQ:IQAD:Q OFF	IQ:IQADjustment:Q OFFset	信号	调节 Q 通道的偏置
IQ:IQAD:QSK	IQ:IQADjustment:QSKew	信号	通过增大或减小 Q 的相位角来调节矢量 I 和矢量 Q 之间的相位角
IQ:TRIG:SOUR	IQ:TRIGger:SOURce	信号	设置 I/Q 的触发源
IQ:WAVE:BUIL	IQ:WAVEload:BUILtin	信号	从内建波形列表选择 I/Q 波形
IQ:WAVE:USER	IQ:WAVEload:USERstored	信号	从用户保存波形列表选择 I/Q 波形
:IQ: Frequency Sampling	:IQ:Frequency Sampling	信号	设置或获取 I/Q 的采样率

3 命令与查询

3.1 IEEE488.2 通用命令介绍

IEEE 标准定义的通用命令适用于查询设备的基本信息和执行基本操作。这些命令通常以“*”开头以及命令的关键字长度为 3 个字符。

3.1.1 *IDN

描述	*IDN?是用于查询设备的 id。响应包含厂商、机型、序列号、软件版本和硬件版本。
查询语法	*IDN?
响应格式	<p>格式 1: *IDN, <设备 id>, <机型>, <序列号>, <软件版本>, <硬件版本>.</p> <p>格式 2: <设备商>, <机型>, <序列号>, <软件版本>, <硬件版本>.</p> <p><设备 id>: = “SDG” 被用于识别仪器。</p> <p><设备商>: = “Siglent Technologies”.</p> <p><机型>: = 机器型号</p> <p><序列号>: = 每个产品有自己的编号, 此序列号标注产品的唯一性。</p> <p><软件版本>: = 与软件版本信息相关的编号</p> <p><硬件版本>: = 固件级字段应该包含所有可单独修改子系统的信息。这些信息包含在单个或多个修订版本代码中。</p>
示例	<p>读取版本信息。</p> <p><i>*IDN?</i></p> <p>返回值:</p> <p><i>Siglent Technologies,SDG7102A,SDG1XAAD4R0004,1.1.1.35T</i>(每个版本可能不同)</p>

备注:

<硬件版本>格式: value1-value2-value3-value4-value5.

value1: PCB 版本;

value2: 硬件版本;

value3: 硬件修订版;

value4: FPGA 版本;

value5: CPLD 版本。

3.1.2 *OPC

描述	<p>*OPC（操作完成）命令设置在标准事件状态寄存器 [ESR] 的 OPC 位（位 0）。此命令对设备操作无其他影响。因为仪器是在处理完上一条设置或查询语句之后，才开始解析设置或查询命令。</p> <p>*OPC? 查询命令总是返回 ASCII 码的 1。因为在前一个命令已经完全执行完后设备才响应该查询命令。</p>
命令语法	*OPC
查询语法	*OPC?
响应格式	1

3.1.3 *RST

描述	*RST 命令启动仪器重置并调用默认设置。
命令语法	*RST
示例	该例子将信号发生器重置成默认设置： <i>*RST</i>

3.2 输出命令

描述	启用和禁用通道对应前面板[OUTPUT]接口的输出。 查询结果返回“ON”、“OFF”、LOAD 和 PLRT 参数。
命令语法	<通道>: OUTPutON OFF, LOAD, <负载>, PLRT, <极性> <通道>: = {C1, C2}. <负载>: = {见备注}. 默认单位为 ohm。 <极性>: = {NOR, INVT}, 其中 NOR 代表正常, INVT 代表反相。
查询语法	<通道>: OUTPut?
响应格式	<通道>: OUTP ON OFF, LOAD, <负载>, PLRT, <极性>
示例	<p>打开 CH1:</p> <pre>C1:OUTP ON</pre> <p>读取 CH1 输出状态:</p> <pre>C1:OUTP?</pre> <p>返回值:</p> <pre>C1:OUTP ON,LOAD,HZ,PLRT,NOR</pre> <p>设置 CH1 的负载为 50ohm:</p> <pre>C1:OUTP LOAD,50</pre> <p>设置 CH1 的负载为高阻:</p> <pre>C1:OUTP LOAD,HZ</pre> <p>设置 CH1 的极性为正常:</p> <pre>C1:OUTP PLRT,NOR</pre>

备注: * "HZ"代表高阻态。

描述	同时设置两通道的输出状态
命令语法	OUT_BOTHCH <STATE> <STATE>: = {ON, OFF}.
查询语法	
响应格式	
示例	<p>两通道打开输出:</p> <pre>OUT_BOTHCH ON</pre>

3.3 噪声叠加命令

描述	开启噪声叠加，并以指定信噪比向通道添加噪声
命令语法	<p><通道>: OUTPutONIOFF, LOAD, <负载>, PLRT, <极性></p> <p><通道>: NOISE_ADD STATE, ONIOFF, RATIO, <值></p> <p>或</p> <p><通道>: NOISE_ADD STATE, ONIOFF, RATIO_DB, <值(dB)></p> <p><通道>: = {C1, C2}.</p> <p>信噪比的值详见数据手册</p>
查询语法	<通道>: NOISE_ADD?
响应格式	<p><通道>: NOISE_ADD STATE, ONIOFF, RATIO, <值></p> <p>或</p> <p><通道>: NOISE_ADD STATE, ONIOFF, RATIO_DB, <值(dB)></p>
示例	<p>打开通道一噪声叠加并设置信噪比为 120:</p> <p><i>C1:NOISE_ADD STATE,ON,RATIO,120</i></p>

3.4 幅度限制命令

描述	设置通道最大幅度限制
命令语法	<p><通道>: BaSic_WaVe<参数>, <值></p> <p><通道>: = {C1, C2}.</p> <p><参数>: = MAX_OUTPUT_AMP</p> <p><值>: = {可在数据手册查阅参数值的有效范围}.</p>
示例	<p>设置通道 1 最大输出幅度为 10V:</p> <p><i>C1:BSWV MAX_OUTPUT_AMP,10</i></p>

3.5 基本波形命令

描述	设置或者获取基本波参数。
命令语法	<p><通道>: BaSic_WaVe <参数>,<值></p> <p><通道>: = {C1, C2}.</p> <p><参数>: = MAX_OUTPUT_AMP</p>

<值>: = {相关参数的值}。

参数	值	描述
WVTP	<type>	:= {SINE, SQUARE, RAMP, PULSE, NOISE, ARB, DC, PRBS, IQ}。如果命令未设置基本波形类型, WVTP 默认设置为当前波形。
FRQ	<frequency>	:=频率。单位是赫兹“Hz”, 可在数据手册查阅参数值的有效范围。当 WVTP 为噪声和直流, 该值无效。
PERI	<period>	:=周期。单位是秒“s”。可在数据手册查阅参数值的有效范围。当 WVTP 为噪声和直流, 该值无效。
AMP	<amplitude>	:=幅值。单位是伏特, 峰峰值“Vpp”。可在数据手册查阅参数值的有效范围。当 WVTP 为噪声和直流, 该参数无效。
OFST	<offset>	:=偏置。单位是伏特“V”。可在数据手册查阅参数值的有效范围。当 WVTP 为噪声, 该值无效。
COM_OFST	<common offset>	:= {-1 至 1}。共模偏置电压。单位是伏特“V”。仅当通道差分输出开启才可设置。
SYM	<symmetry>	:= {0 至 100}。三角波的对称度。单位是“%”。仅当 WVTP 为三角波才能设置该参数。
DUTY	<duty>	:= {0 至 100}。占空比。单位是“%”。该参数的值取决于频率。仅当 WVTP 是方波和脉冲才能设置该参数。
PHSE	<phase>	:= {0 至 360}。单位是“°”, 当 WVTP 是噪声、脉冲波、直流时, 该参数无效。
STDEV	<stdev>	:=噪声的标准差。单位是伏特“V”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是噪声时, 才能设置。
MEAN	<mean>	:=噪声的均值。单位是伏特“V”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是噪声时, 才能设置该参数。
WIDTH	<width>	:=正脉宽。单位是秒“s”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是脉冲波时, 才能设置该参数。
RISE	<rise>	:=上升时间 (10%~90%)。单位是秒“s”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是脉冲波时, 才能设置该参数。

FALL	<fall>	:=下降时间 (10%~90%)。单位是秒“s”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是脉冲波时, 才能设置该参数。
DLY	<delay>	:=波形延迟。则可在数据手册查阅参数值的有效范围
HLEV	<high level>	:=高电平。单位是伏特“V”。当 WVTP 为噪声, 该值无效。
LLEV	<low level>	:=低电平。单位是伏特“V”。当 WVTP 为噪声, 该值无效。
BANDSTATE	<bandwidth switch>	:= {ON (打开), OFF (关闭)}。当 WVTP 为噪声, 该参数才能设置
BANDWIDTH	<bandwidth value>	:=噪声带宽。单位为赫兹“Hz”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是噪声时, 才能设置该参数。
LENGTH	<prbs length>	:= {3~32}。PRBS 实际长度=2 ^{长度} - 1。仅当 WVTP 是 PRBS 时, 才能设置该参数。
EDGE	<prbs rise/fall>	:= PRBS 的上升/下降时间。单位是秒“s”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是 PRBS 时, 才能设置该参数。
FORMAT	<output format>	:= {DIFF, SINGLE}。设置通道差分输出或者单端输出
DIFFSTATE	<prbs differential switch>	:= {ON (打开), OFF (关闭)}。PRBS 差分输出模式。仅当 WVTP 是 PRBS 时, 才能设置该参数。
BITRATE	<prbs bit rate>	:= PRBS 比特率。单位是位每秒“bps”。可在数据手册查阅参数值的有效范围。仅当 WVTP 是 PRBS 时, 才能设置该参数。
LOGICLEVEL	<prbslogiclevel rate>	:= {TTL_CMOS, LVTTTL_LVCMOS, ECL, LVPECL, LVDS, CUSTOM}。用于设置 PRBS 的逻辑电平, 仅当 WVTP 是 PRBS 时, 才能设置该参数。
AMPVRMS	<amplitude>	:=幅度。设置幅度单位为 V _{rm} 。
AMPDBM	<amplitude>	:=幅度。设置幅度单位为 dB _m 。
MAX_OUTPUT _AMP	<amplitude>	:=通道最大幅度限制, 可在数据手册查阅参数值的有效范围

查询语法

<通道>: BaSic_WaVe?

	<通道>: ={C1, C2}.
响应格式	<通道>: BSWV<参数> <参数>: = {当前波形的所有参数}.
示例	<p>改变 C1 波形为三角波: <i>C1:BSWV WVTP,RAMP</i></p> <p>改变 C1 频率为 2000Hz: <i>C1:BSWV FRQ,2000</i></p> <p>设置 C1 的幅度为 3Vpp: <i>C1:BSWV AMP,3</i></p> <p>从设备读取 C1 的参数: <i>C1:BSWV?</i></p> <p>返回值: <i>C1:BSWV WVTP,SINE,FRQ,100HZ,PERI,0.01S,AMP,2V, OFST,0V,HLEV,1V,LLEV,-1V,PHSE,0</i></p> <p>设置 C1 的噪声带宽为 100MHz: <i>C1:BSWV BANDWIDTH,100E6</i> 或 <i>C1:BSWV BANDWIDTH,100000000</i></p>

3.6 调制波形命令

描述	该命令可以设置或者获取调制波形参数
命令语法	<p><通道>: MoDulateWaVe <类型> <通道>: MoDulateWaVe<参数>, <值> <通道>: = {C1, C2}</p> <p><类型>: = {AM, DSBAM, FM, PM, PWM, ASK, FSK, PSK}. <参数>: = {下表的值}. <值>: = {相关参数的值}.</p>

参数	值	描述
STATE	<state>	:= {ON (打开), OFF (关闭)}。启用和禁用调制。 如果你想设置或者读取调制的其他参数,必须先将 STATE 设置为“ON”。
AM, SRC	<src>	:= {INT (内调制), EXT (外调制), CH1 (通道一调制), CH2 (通道二调制)}。AM 调制源。
AM, MDSP	<mod wave shape>	:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}。 AM 调制波形。仅当调制源设置为内调制时,该参数才能设置。
AM, FRQ	<AM frequency>	:= AM 频率。单位是赫兹“Hz”,可在数据手册查阅参数值的有效范围。调制源设置为内调制时,该参数才能设置。
AM, DEPTH	<depth>	:= {0 至 120}。AM 深度。单位是“%”。触发源设置为内触发时,该参数才能设置。
DSBSC, SRC	<src>	:= {INT (内调制), EXT (外调制)}。DSBSC 调制源。
DSBSC, MDSP	<mod wave shape>	:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}。 DSB-SC 调制波形。调制源设置为内调制时,该参数才能设置。
DSBSC, FRQ	<DSB-AM frequency>	:= DSB AM 频率。单位是赫兹“Hz”,可在数据手册查阅参数值的有效范围。设置为内调制时,该参数才能设置。
FM, SRC	<src>	:= {INT (内调制), EXT (外调制), CH1 (通道一调制), CH2 (通道二调制)}。FM 调制源
FM, MDSP	<mod wave shape>	:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}。 FM 调制波形。调制源设置为内调制时,该参数才能设置。
FM, FRQ	<FM frequency>	:= FM 频率。单位是赫兹“Hz”,可在数据手册查阅参数值的有效范围。调制源设置为内调制时,该参数才能设置。

FM, DEVI	<FM frequency deviation>	:= {0 至当前载波频率}。 FM 频率偏差。该值取决于载波频率和带宽频率的差值。调制源设置为内调制时，该参数才能设置。
PM, SRC	<src>	:= {INT (内调制), EXT (外调制), CH1 (通道一调制), CH2 (通道二调制)}。PM 调制源。
PM, MDSP	<mod wave shape>	:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}。 PM 调制波形。调制源设置为内调制时，该参数才能设置。
PM, FRQ	<PM frequency>	:= PM 频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。调制源设置为内调制时，该参数才能设置。
PM, DEVI	<PM phase offset>	:= {0 至 360}。PM 相位偏差。单位是“°”，调制源设置为内调制时，该参数才能设置。
PWM, SRC	<src>	:= {INT (内调制), EXT (外调制), CH1 (通道一调制), CH2 (通道二调制)}。PWM 调制源。
PWM, FRQ	<PWM frequency>	:= PWM 频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。调制源设置为内调制时，该参数才能设置。
PWM, DEVI	<PWM dev>	:= 占空比偏差。单位是“s”。值取决于载波的脉宽。
PWM, MDSP	<mod wave shape>	:= {SINE, SQUARE, TRIANGLE, UPRAMP, DNRAMP, NOISE, ARB}。 PWM 调制波形。调制源设置为内调制时，该参数才能设置。
ASK, SRC	<src>	:= {INT (内调制), EXT (外调制)}。ASK 调制源。
ASK, KFRQ	< key frequency>	:= ASK 键控频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。调制源设置为内调制时，该参数才能设置。
FSK, SRC	<src>	:= {INT (内调制), EXT (外调制)}。FSK 调制源。
FSK, KFRQ	< key frequency>	:= FSK 键控频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。调制源设置为内调制时，该参数才能设置。

FSK, HFRQ	<FSK_hop_freq>	:= FSK 跳频频率。与基础波形频率相同。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。
PSK, SRC	<src>	:= {INT, EXT}。PSK 调制源。
PSK, KFRQ	< key frequency>	:= PSK 键控频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。调制源设置为内调制时，该参数才能设置。
CARR, WVTP	<wave type>	:= {SINE, SQUARE, RAMP, ARB, PULSE}。载波频率类型。
CARR, FRQ	<frequency>	:= 载波频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。
CARR, PHSE	<phase>	:= {0 至 360}。载波相位。单位为“度”。
CARR, AMP	<amplitude>	:=载波幅值。单位是伏特，峰峰值“Vpp”。可在数据手册查阅参数值的有效范围。
CARR, OFST	<offset>	:=载波偏置。单位是伏特。可在数据手册查阅参数值的有效范围。
CARR, SYM	<symmetry>	:= {0 至 100}。当载波为 RAMP 时，载波对称度。单位是“%”。
CARR, DUTY	<duty>	:= {0 至 100}。当载波为方波或者脉冲波时，载波占空比。单位是“%”。
CARR, RISE	<rise>	:=当载波为方波或者脉冲波时，载波上升时间。单位是秒“s”。可在数据手册查阅参数值的有效范围。
CARR, FALL	<fall>	:=当载波为方波或者脉冲波时，载波下降时间。单位是秒“s”。可在数据手册查阅参数值的有效范围。
CARR, DLY	<delay>	:=当载波为方波或者脉冲波时，载波延时。单位是秒“s”。可在数据手册查阅参数值的有效范围。

查询语法 <通道>: MoDulateWaVe?
 <通道>: = {C1, C2}.

响应格式 <通道>: MDWV<参数>
 <参数>: = {当前调制的所有参数}.

示例 设置 CH1 调制状态为打开：
 C1:MDWV STATE,ON

设置 CH1 调制类型为 AM:

C1:MDWV AM

设置 AM 调制，且调制波形设为正弦波:

C1:MDWV AM,MDSP,SINE

读取状态值为 ON 的 C1 的调制参数:

C1:MDWV?

返回值:

*C1:MDWVAM,STATE,ON,MDSP,SINE,SRC,INT,FRQ,100HZ,
DEPTH,100,CARR,WVTP,RAMP,FRQ,1000HZ,AMP,4V,AMPVRMS,
1.15473Vrms,OFST,0V,PHSE,0,SYM,50*

读取状态为 OFF 的 CH1 的调制参数:

C1:MDWV?

返回值:

C1:MDWV STATE,OFF

设置 CH1 的 FM 频率为 1000Hz:

C1:MDWV FM,FRQ,1000

设置 CH1 的载波为正弦波:

C1:MDWV CARR,WVTP,SINE

设置 CH1 载波频率为 1000Hz:

C1:MDWV CARR,FRQ,1000

3.7 扫描波形命令

3.7.1 <channel>:SWEep <switch>

描述	该命令用于设置(查询)扫描波形的开启状态。
命令语法	<pre><channel>:SWEep <switch> <channel>:= {C1, C2}. <switch>:= {ON, OFF}.</pre>
查询语法	<pre><channel>: SWEep? <channel>:= {C1, C2}.</pre>
示例	<p>开启通道 1 的扫描输出</p> <pre><i>:C1:SWEep ON</i></pre> <p>读取通道 1 的扫描输出状态</p> <pre><i>:C1:SWEep?</i></pre> <p>返回值:</p> <pre><i>"ON"</i></pre>

3.7.2 <channel>:SWEep:TYPE <type>

描述	该命令用于设置(查询)扫描波形的类型。
命令语法	<pre><channel>:SWEep:TYPE <type> <channel>:= {C1, C2}. <type>:= {FREQ, AMP, BOTH}.</pre>
查询语法	<pre><channel>: SWEep:TYPE? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 的扫描类型为频率扫描</p> <pre><i>:C1:SWEep:TYPE FREQ</i></pre> <p>查询通道 1 的扫描类型</p> <pre><i>:C1:SWEep:TYPE?</i></pre> <p>返回值:</p> <pre><i>"FREQ"</i></pre>

3.7.3 <channel>:SWEep:SOURce <src>

描述	该命令用于设置(查询)扫描波形的触发源。
命令语法	<pre><channel>:SWEep:SOURce <src> <channel>:= {C1, C2}. <src>:= {INT, EXT, MAN}.</pre>
查询语法	<pre><channel>: SWEep:SOURce? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫描为内部触发</p> <pre><i>:C1:SWEep:SOURce INT</i></pre> <p>查询通道 1 的扫描触发源</p> <pre><i>:C1:SWEep:SOURce?</i></pre> <p>返回值:</p> <pre><i>"INT"</i></pre>

3.7.4 <channel>:SWEep:FMODE <mode>

描述	该命令用于设置(查询)频率扫描的扫描模式。
命令语法	<pre><channel>:SWEep:FMODE <mode> <channel>:= {C1, C2}. <mode>:= {LINE, LOG, STEP}.</pre>
查询语法	<pre><channel>: SWEep:FMODE? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 频率扫描的扫描模式为线性扫描</p> <pre><i>:C1:SWEep:FMODE LINE</i></pre> <p>查询通道 1 频率扫描的扫描模式</p> <pre><i>:C1:SWEep:FMODE?</i></pre> <p>返回值:</p> <pre><i>"LINE"</i></pre>

3.7.5 <channel>:SWEep:AMODe <mode>

描述	该命令用于设置(查询)幅度扫描的扫描模式。
命令语法	<pre><channel>:SWEep:AMODe <mode> <channel>:= {C1, C2}. <mode>:= {LINE, STEP}.</pre>
查询语法	<pre><channel>: SWEep:AMODe? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 幅度扫描的扫描模式为线性扫描</p> <pre>:C1:SWEep:AMODe LINE</pre> <p>查询通道 1 幅度扫描的扫描模式</p> <pre>:C1:SWEep:AMODe?</pre> <p>返回值:</p> <pre>"LINE"</pre>

3.7.6 <channel>:SWEep:FSNumber <value>

描述	该命令用于设置(查询)扫频模式为步进扫频时的步进数。
命令语法	<pre><channel>:SWEep:FSNumber <value> <channel>:= {C1, C2}. <value>:= {2 到 1024 之间的整数}</pre>
查询语法	<pre><channel>: SWEep: FSNumber? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫频模式为步进扫频时的步进数为 10</p> <pre>:C1:SWEep:FSNumber 10</pre> <p>查询通道 1 扫频模式为步进扫频时的步进数</p> <pre>:C1:SWEep:FSNumber?</pre> <p>返回值:</p> <pre>"10"</pre>

3.7.7 <channel>:SWEep:ASNumber <value>

描述	该命令用于设置(查询)扫幅模式为步进扫幅时的步进数。
命令语法	<pre><channel>:SWEep:ASNumber <value> <channel>:= {C1, C2}. <value>:= {2 到 1024 之间的整数}</pre>
查询语法	<pre><channel>: SWEep: ASNumber? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫幅模式为步进扫幅时的步进数为 10</p> <pre><i>:C1:SWEep:ASNumber 10</i></pre> <p>查询通道 1 扫幅模式为步进扫幅时的步进数</p> <pre><i>:C1:SWEep:ASNumber?</i></pre> <p>返回值:</p> <pre><i>"10"</i></pre>

3.7.8 <channel>:SWEep:TIME <value>

描述	该命令用于设置(查询)扫描波形的扫描时间。
命令语法	<pre><channel>:SWEep:TIME <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位秒</pre>
查询语法	<pre><channel>: SWEep: TIME? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫描波形的扫描时间为 100ms</p> <pre><i>:C1:SWEep:TIME 0.1</i></pre> <p>查询通道 1 扫描波形的扫描时间</p> <pre><i>:C1:SWEep:TIME?</i></pre> <p>返回值:</p> <pre><i>"0.1"</i></pre>

3.7.9 <channel>:SWEep:SHTime <value>

描述	该命令用于设置(查询)扫描波形的扫描起始保持时间。
命令语法	<pre><channel>:SWEep:SHTime <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位秒</pre>
查询语法	<pre><channel>: SWEep: SHTime? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫描波形的扫描起始保持时间为 100ms</p> <pre><i>:C1:SWEep:SHTime 0.1</i></pre> <p>查询通道 1 扫描波形的扫描起始保持时间</p> <pre><i>:C1:SWEep:SHTime?</i></pre> <p>返回值:</p> <pre><i>"0.1"</i></pre>

3.7.10 <channel>:SWEep:EHTime <value>

描述	该命令用于设置(查询)扫描波形的扫描结束保持时间。
命令语法	<pre><channel>:SWEep:EHTime <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位秒</pre>
查询语法	<pre><channel>: SWEep: EHTime? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫描波形的扫描结束保持时间为 100ms</p> <pre><i>:C1:SWEep:EHTime 0.1</i></pre> <p>查询通道 1 扫描波形的扫描结束保持时间</p> <pre><i>:C1:SWEep:EHTime?</i></pre> <p>返回值:</p> <pre><i>"0.1"</i></pre>

3.7.11 <channel>:SWEep:RTIME <value>

描述	该命令用于设置(查询)扫描波形扫描结束后的返回时间。
命令语法	<pre><channel>:SWEep:RTIME <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位秒</pre>
查询语法	<pre><channel>: SWEep:RTIME? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫描波形扫描结束后的返回时间为 100ms</p> <pre><i>:C1:SWEep:RTIME 0.1</i></pre> <p>查询通道 1 扫描波形扫描结束后的返回时间</p> <pre><i>:C1:SWEep:RTIME?</i></pre> <p>返回值:</p> <pre><i>"0.1"</i></pre>

3.7.12 <channel>:SWEep:SFRequency <value>

描述	该命令用于设置(查询)频率扫描的起始频率。
命令语法	<pre><channel>:SWEep:SFRequency <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位赫兹 (Hz)</pre>
查询语法	<pre><channel>: SWEep:SFRequency? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 频率扫描的起始频率为 1kHz</p> <pre><i>:C1:SWEep:SFRequency 1000</i></pre> <p>查询通道 1 频率扫描的起始频率</p> <pre><i>:C1:SWEep:SFRequency?</i></pre> <p>返回值:</p> <pre><i>"1000"</i></pre>

3.7.13 <channel>:SWEep:EFRequency <value>

描述	该命令用于设置(查询)频率扫描的终止频率。
命令语法	<channel>:SWEep:EFRequency <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位赫兹 (Hz)
查询语法	<channel>: SWEep:EFRequency? <channel>:= {C1, C2}.
示例	设置通道 1 频率扫描的终止频率为 9KHz <i>:C1:SWEep:EFRequency 9000</i> 查询通道 1 频率扫描的终止频率 <i>:C1:SWEep:EFRequency?</i> 返回值: <i>"9000"</i>

3.7.14 <channel>:SWEep:CFRequency <value>

描述	该命令用于设置(查询)频率扫描的中心频率。
命令语法	<channel>:SWEep:CFRequency <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位赫兹 (Hz)
查询语法	<channel>: SWEep:CFRequency? <channel>:= {C1, C2}.
示例	设置通道 1 频率扫描的中心频率为 5KHz <i>:C1:SWEep:CFRequency 5000</i> 查询通道 1 频率扫描的中心频率 <i>:C1:SWEep:CFRequency?</i> 返回值: <i>"5000"</i>

3.7.15 <channel>:SWEep:FSPan <value>

描述	该命令用于设置(查询)频率扫描的频率范围。
命令语法	<pre><channel>:SWEep:FSPan <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位赫兹 (Hz)</pre>
查询语法	<pre><channel>: SWEep: FSPan? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 频率扫描的频率范围为 8KHz</p> <pre><i>:C1:SWEep:FSPan 8000</i></pre> <p>查询通道 1 频率扫描的频率范围</p> <pre><i>:C1:SWEep: FSPan?</i></pre> <p>返回值:</p> <pre><i>"8000"</i></pre>

3.7.16 <channel>:SWEep:SAMPlitude <value>

描述	该命令用于设置(查询)幅度扫描的起始幅度。
命令语法	<pre><channel>:SWEep:SAMPlitude <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位伏特 (V)</pre>
查询语法	<pre><channel>: SWEep: SAMPlitude? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 幅度扫描的起始幅度为 100mV</p> <pre><i>:C1:SWEep:SAMPlitude 0.1</i></pre> <p>查询通道 1 幅度扫描的起始幅度</p> <pre><i>:C1:SWEep: SAMPlitude?</i></pre> <p>返回值:</p> <pre><i>"0.1"</i></pre>

3.7.17 <channel>:SWEep:EAMPlitude <value>

描述	该命令用于设置(查询) 幅度扫描的终止幅度。
命令语法	<pre><channel>:SWEep:EAMPlitude <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位伏特 (V)</pre>
查询语法	<pre><channel>: SWEep:EAMPlitude? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 幅度扫描的终止幅度为 900mV</p> <pre><i>:C1:SWEep:EAMPlitude 0.9</i></pre> <p>查询通道 1 幅度扫描的终止幅度</p> <pre><i>:C1:SWEep:EAMPlitude?</i></pre> <p>返回值:</p> <pre><i>"0.9"</i></pre>

3.7.18 <channel>:SWEep:CAMPlitude <value>

描述	该命令用于设置(查询) 幅度扫描的中心幅度。
命令语法	<pre><channel>:SWEep:CAMPlitude <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位伏特 (V)</pre>
查询语法	<pre><channel>: SWEep:CAMPlitude? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 幅度扫描的中心幅度为 500mV</p> <pre><i>:C1:SWEep:CAMPlitude 0.5</i></pre> <p>查询通道 1 幅度扫描的中心幅度</p> <pre><i>:C1:SWEep:CAMPlitude?</i></pre> <p>返回值:</p> <pre><i>"0.5"</i></pre>

3.7.19 <channel>:SWEep:ASPan <value>

描述	该命令用于设置(查询) 幅度扫描的幅度范围。
命令语法	<pre><channel>:SWEep:ASPan <value> <channel>:= {C1, C2}. <value>:= 浮点类型的值, 单位伏特 (V)</pre>
查询语法	<pre><channel>: SWEep: ASPan? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 幅度扫描的幅度范围为 800mV</p> <pre><i>:C1:SWEep:ASPan 0.8</i></pre> <p>查询通道 1 幅度扫描的幅度范围</p> <pre><i>:C1:SWEep:ASPan?</i></pre> <p>返回值:</p> <pre><i>"0.8"</i></pre>

3.7.20 <channel>:SWEep:FDIRectioN <direction>

描述	该命令用于设置(查询) 频率扫描的扫描方向。
命令语法	<pre><channel>:SWEep:FDIRectioN <direction> <channel>:= {C1, C2}. <direction>:= {UP, DOWN, UP_DOWN}.</pre>
查询语法	<pre><channel>: SWEep: FDIRectioN? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 频率扫描的扫描方向为向上扫描</p> <pre><i>:C1:SWEep:FDIRectioN UP</i></pre> <p>查询通道 1 频率扫描的扫描方向</p> <pre><i>:C1:SWEep:FDIRectioN?</i></pre> <p>返回值:</p> <pre><i>"UP"</i></pre>

3.7.21 <channel>:SWEep:ADIRecton <direction>

描述	该命令用于设置(查询) 幅度扫描的扫描方向。
命令语法	<pre><channel>:SWEep:ADIRecton <direction> <channel>:= {C1, C2}. <direction>:= {UP, DOWN, UP_DOWN}.</pre>
查询语法	<pre><channel>: SWEep:ADIRecton? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 幅度扫描的扫描方向为向上扫描</p> <pre><i>:C1:SWEep:ADIRecton UP</i></pre> <p>查询通道 1 幅度扫描的扫描方向</p> <pre><i>:C1:SWEep:ADIRecton?</i></pre> <p>返回值:</p> <pre><i>"UP"</i></pre>

3.7.22 <channel>:SWEep:FSYMmetry <value>

描述	该命令用于设置(查询) 扫频方向为上下扫频时的扫描对称性。
命令语法	<pre><channel>:SWEep:FSYMmetry <value> <channel>:= {C1, C2}. <value>:= {0 至 100 的浮点数}.</pre>
查询语法	<pre><channel>: SWEep:FSYMmetry? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫频方向为上下扫频时的扫描对称性为 50%</p> <pre><i>:C1:SWEep:FSYMmetry 50</i></pre> <p>查询通道 1 扫频方向为上下扫频时的扫描对称性</p> <pre><i>:C1:SWEep:FSYMmetry?</i></pre> <p>返回值:</p> <pre><i>"50"</i></pre>

3.7.23 <channel>:SWEep:ASYMmetry <value>

描述	该命令用于设置(查询) 扫幅方向为上下扫幅时的扫描对称性。
命令语法	<pre><channel>:SWEep:ASYMmetry <value> <channel>:= {C1, C2}. <value>:= {0 至 100 的浮点数}.</pre>
查询语法	<pre><channel>: SWEep:ASYMmetry? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫幅方向为上下扫幅时的扫描对称性为 50%</p> <pre><i>:C1:SWEep:ASYMmetry 50</i></pre> <p>查询通道 1 扫幅方向为上下扫幅时的扫描对称性</p> <pre><i>:C1:SWEep:ASYMmetry?</i></pre> <p>返回值:</p> <pre><i>"50"</i></pre>

3.7.24 <channel>:SWEep:TOUT <switch>

描述	该命令用于设置(查询) 扫描波形的触发输出开关状态。并且触发源为外部触发时无触发输出。
命令语法	<pre><channel>:SWEep:TOUT <switch> <channel>:= {C1, C2}. <switch>:= {ON, OFF}.</pre>
查询语法	<pre><channel>: SWEep:TOUT? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 扫描波形的触发输出状态为开启</p> <pre><i>:C1:SWEep:TOUT ON</i></pre> <p>查询通道 1 扫描波形的触发输出开关状态</p> <pre><i>:C1:SWEep:TOUT?</i></pre> <p>返回值:</p> <pre><i>"ON"</i></pre>

3.7.25 <channel>:SWEep:EDGe <polarity>

描述	该命令用于设置(查询) 触发源为外部触发时, 触发沿的极性。
命令语法	<channel>:SWEep:EDGe <polarity> <channel>:= {C1, C2}. <polarity>:= {RISE, FALL}.
查询语法	<channel>: SWEep:EDGe? <channel>:= {C1, C2}.
示例	设置通道 1 扫描波形的的外部触发源为上升沿触发 <i>:C1:SWEep:EDGe RISE</i> 查询通道 1 扫描波形的的外部触发源的极性 <i>:C1:SWEep:EDGe?</i> 返回值: <i>"RISE"</i>

3.7.26 <channel>:SWEep:MTRigger

描述	该命令用于触发源为手动触发时, 手动触发一次扫描。
命令语法	<channel>:SWEep:MTRigger <channel>:= {C1, C2}.
示例	手动触发一次通道 1 <i>:C1:SWEep:MTRigger</i>

3.7.27 <channel>:SWEep:FMARker <switch>

描述	该命令用于设置(查询) 频率扫描的标记频率开关状态。
命令语法	<channel>:SWEep:FMARker <switch> <channel>:= {C1, C2}. <switch>:= {ON, OFF}.
查询语法	<channel>: SWEep:FMARker? <channel>:= {C1, C2}.
示例	设置通道 1 频率扫描的标记频率为开启状态 <i>:C1:SWEep:FMARker ON</i>

查询通道 1 频率扫描的标记频率开关状态

:C1:SWEEP:FMARker?

返回值:

"ON"

3.7.28 <channel>:SWEep:MFRequency <value>

描述	该命令用于设置(查询) 频率扫描标记频率的频率值。
命令语法	<channel>:SWEep:MFRequency<value> <channel>:= {C1, C2}. <value>:=浮点类型的值, 单位赫兹 (Hz)
查询语法	<channel>: SWEep:MFRequency? <channel>:= {C1, C2}.
示例	设置通道 1 频率扫描标记频率的频率值为 5kHz <i>:C1:SWEEP:MFRequency 5000</i> 查询通道 1 频率扫描标记频率的频率值 <i>:C1:SWEEP:MFRequency?</i> 返回值: <i>"5000"</i>

3.7.29 <channel>:SWEep:MSNumber <value>

描述	该命令用于设置(查询) 频率扫描标记频率标记的哪个步进。并且仅对步进扫频有效
命令语法	<channel>:SWEep:MSNumber <value> <channel>:= {C1, C2}. <value>:= {2~1024}
查询语法	<channel>: SWEep:MSNumber ? <channel>:= {C1, C2}.
示例	设置通道 1 频率扫描标记频率标记第 5 个步进 <i>:C1:SWEEP:MSNumber 5</i> 查询通道 1 频率扫描标记频率标记的哪个步进 <i>:C1:SWEEP:MSNumber?</i>

返回值:

"5"

3.7.30 <channel>:SWEep:CARRier:WTYPE <wave>

描述	该命令用于设置(查询) 波形扫描载波的类型。
命令语法	<pre><channel>:SWEep:CARRier:WTYPE <wave> <channel>:= {C1, C2}. <wave>:={SINE,SQUARE,RAMP,AFG}</pre>
查询语法	<pre><channel>: SWEep:CARRier:WTYPE? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 波形扫描的载波为正弦</p> <pre><i>:C1:SWEep:CARRier:WTYPE SINE</i></pre> <p>查询通道 1 波形扫描载波的类型</p> <pre><i>:C1:SWEep:CARRier:WTYPE?</i></pre> <p>返回值:</p> <p><i>"SINE"</i></p>

3.7.31 <channel>:SWEep:CARRier:FREQuency <value>

描述	该命令用于设置(查询) 波形扫描载波的频率。
命令语法	<pre><channel>:SWEep:CARRier:FREQuency <value> <channel>:= {C1, C2}. <value>:=浮点类型的值, 单位赫兹 (Hz)</pre>
查询语法	<pre><channel>: SWEep:CARRier:FREQuency? <channel>:={C1, C2}.</pre>
示例	<p>设置通道 1 波形扫描载波的频率为 1MHz</p> <pre><i>:C1:SWEep:CARRier:FREQuency 1000000</i></pre> <p>查询通道 1 波形扫描载波的频率</p> <pre><i>:C1:SWEep:CARRier:FREQuency?</i></pre> <p>返回值:</p> <p><i>"1000000"</i></p>

3.7.32 <channel>:SWEep:CARRier:PHASe <value>

描述	该命令用于设置(查询) 波形扫描载波的相位。
命令语法	<channel>:SWEep:CARRier:PHASe <value> <channel>:= {C1, C2}. <value>:=浮点类型的值, 单位度
查询语法	<channel>: SWEep:CARRier:PHASe ? <channel>:={C1, C2}.
示例	设置通道 1 波形扫描载波的相位为 90 度 <pre><i>:C1:SWEep:CARRier:PHASe 90</i></pre> 查询通道 1 波形扫描载波的相位 <pre><i>:C1:SWEep:CARRier:PHASe ?</i></pre> 返回值: <pre><i>"90"</i></pre>

3.7.33 <channel>:SWEep:CARRier:PAMPlitude <value>

描述	该命令用于使用峰峰值设置(查询) 波形扫描载波的幅度。
命令语法	<channel>:SWEep:CARRier:PAMPlitude <value> <channel>:= {C1, C2}. <value>:=浮点类型的值, 单位伏特 (V)
查询语法	<channel>: SWEep:CARRier:PAMPlitude? <channel>:={C1, C2}.
示例	设置通道 1 波形扫描载波的幅度为 4Vpp <pre><i>:C1:SWEep:CARRier:PAMPlitude 4</i></pre> 查询通道 1 波形扫描载波的幅度 <pre><i>:C1:SWEep:CARRier:PAMPlitude?</i></pre> 返回值: <pre><i>"4"</i></pre>

3.7.34 <channel>:SWEep:CARRier:RAMPlitude <value>

描述	该命令用于使用有效值设置(查询) 波形扫描载波的幅度。
命令语法	<pre><channel>:SWEep:CARRier:RAMPlitude <value> <channel>:= {C1, C2}. <value>:=浮点类型的值, 单位 Vrms</pre>
查询语法	<pre><channel>: SWEep:CARRier:RAMPlitude? <channel>:={C1, C2}.</pre>
示例	<p>设置通道 1 波形扫描载波的幅度为 1.414Vrms</p> <pre><i>:C1:SWEep:CARRier:RAMPlitude 1.414</i></pre> <p>查询通道 1 波形扫描载波的幅度</p> <pre><i>:C1:SWEep:CARRier:RAMPlitude?</i></pre> <p>返回值:</p> <pre><i>"1.414"</i></pre>

3.7.35 <channel>:SWEep:CARRier:OFFSet <value>

描述	该命令用于设置(查询) 波形扫描载波的偏移。
命令语法	<pre><channel>:SWEep:CARRier:OFFSet <value> <channel>:= {C1, C2}. <value>:=浮点类型的值, 单位伏特 (V)</pre>
查询语法	<pre><channel>: SWEep:CARRier:OFFSet? <channel>:={C1, C2}.</pre>
示例	<p>设置通道 1 波形扫描载波的偏移为 2V</p> <pre><i>:C1:SWEep:CARRier:OFFSet 2</i></pre> <p>查询通道 1 波形扫描载波的偏移</p> <pre><i>:C1:SWEep:CARRier:OFFSet?</i></pre> <p>返回值:</p> <pre><i>"2"</i></pre>

3.7.36 <channel>:SWEep:CARRier:SYMMetry <value>

描述	该命令用于设置(查询) 波形扫描载波的对称性。并且仅当载波为三角波时有效
命令语法	<channel>:SWEep:CARRier:SYMMetry <value> <channel>:= {C1, C2}. <value>:={0 到 100 之间的浮点数}
查询语法	<channel>: SWEep:CARRier:SYMMetry? <channel>:= {C1, C2}.
示例	设置通道 1 波形扫描载波的对称性为 50% <i>:C1:SWEep:CARRier:SYMMetry 50</i> 查询通道 1 波形扫描载波的对称性 <i>:C1:SWEep:CARRier:SYMMetry?</i> 返回值: <i>"50"</i>

3.7.37 <channel>:SWEep:CARRier:DUTY <value>

描述	该命令用于设置(查询) 波形扫描载波的占空比。并且仅当载波为方波时有效
命令语法	<channel>:SWEep:CARRier:DUTY <value> <channel>:= {C1, C2}. <value>:={0 到 100 之间的浮点数}
查询语法	<channel>: SWEep:CARRier:DUTY? <channel>:= {C1, C2}.
示例	设置通道 1 波形扫描载波的占空比为 50% <i>:C1:SWEep:CARRier:DUTY 50</i> 查询通道 1 波形扫描载波的占空比 <i>:C1:SWEep:CARRier:DUTY?</i> 返回值: <i>"50"</i>

3.8 脉冲串命令

描述	此命令用于设置或者读取脉冲串波形参数。
命令语法	<p><通道>: BursTWaVe <参数>, <值></p> <p><通道>: = {C1, C2}。</p> <p><参数>: = {下表的参数}。</p> <p><值>: = {相关参数的值}。</p>

参数	值	描述
STATE	<state>	:= {ON (打开), OFF (关闭)}。启用和禁用脉冲串。 如果你想设置或者读取脉冲串的其他参数, 必须先将 STATE 设置为“ON”。
PRD	<period>	:=脉冲串周期。单位是秒“s”。可在数据手册查阅参数值的有效范围。在下列情况下, 该值无效: 在载波为噪声 GATE_NCYC 是门控 ("X"系列除外) TRSR 是外触发
STPS	<start_phase>	:= {0 至 360}。载波的起始相位。单位是“°”。当载波为噪声或者脉冲波时, 该值无效。
GATE_NCYC	<burst_mode>	:= {GATE (门控模式), NCYC (N 循环模式)}。脉冲串模式。当载波为噪声时, 该值无效。
TRSR	<trig_src>	:= {EXT, INT, MAN}触发源。EXT 代表外部触发。 INT 代表内部触发、MAN 代表手动触发。
MTRIG		:=发送一个手动触发信号。仅当 TRSR 是 MAN 时, 该参数有效。
DLAY	<delay>	:=触发延时。单位是秒“s”。可在数据手册查阅参数值的有效范围。当 GATE_NCYC 为 N 循环时, 该值有效。当在载波为噪声时该值无效。
PLRT	<polarity>	:= {NEG (负极性), POS (正极性)}。门控极性, 负极性或者正极性。
TRMD	<trig_mode>	:= {RISE (向上), FALL (向下), OFF (关闭)}。触发输出模式。当 GATE_NCYC 为 N 循环且触发源为内触发或者手动触发时, 该参数有效。当载波为噪声时该值无效。

EDGE	<edge>	:= {RISE, FALL}。有效触发边沿。当 TRST 为手动触发或者外触发时，该值有效。当载波为噪声时该值无效。
TIME	<circle_time>	:= {INF, 1, 2, ..., M}, M 值支持的最大 N 循环数取决于机型。INF 设置脉冲串为无限模式。当 GATE_NCYC 为 N 循环时有效。当载波为噪声时，该值无效。
COUNTER	<counter>	:= 脉冲串数，当触发源为外部或手动时有效，可在数据手册查阅参数的有效值范围。
CARR, WVTP	<wave type>	:= {SINE, SQUARE, RAMP, ARB, PULSE, NOISE}。 载波波形类型
CARR, FRQ	<frequency>	:= 载波频率。单位是赫兹“Hz”，可在数据手册查阅参数值的有效范围。
CARR, PHSE	<phase>	:= {0 至 360}。载波相位。单位是“度”。
CARR, AMP	<amplitude>	:= 载波幅值。单位是伏特，峰峰值“Vpp”。可在数据手册查阅参数值的有效范围。
CARR, OFST	<offset>	:= 载波偏置。单位是幅度“V”。可在数据手册查阅参数值的有效范围。
CARR, SYM	<symmetry>	:= {0 至 100}。载波对称度，当载波为三角波时，该值有效。单位是“%”。
CARR, DUTY	<duty>	:= {0 至 100}。载波占空比，当载波为方波和脉冲波，该值有效。单位是“%”。
CARR, RISE	<rise>	:= 上升时间。当载波为脉冲波时该值有效。单位是“s”。可在数据手册查阅参数值的有效范围。
CARR, FALL	<fall>	:= 下降时间。当载波为脉冲波时该值有效。单位是“s”。可在数据手册查阅参数值的有效范围。
CARR, DLY	<delay>	:= 脉冲波延时。当载波为脉冲波时该值有效。单位是“s”。可在数据手册查阅参数值的有效范围。
CARR, STDEV	<stdev>	:= 噪声的标准差。单位是伏特“V”。可在数据手册查阅参数值的有效范围。
CARR, MEAN	<mean>	:= 噪声的均值。单位是伏特“V”。可在数据手册查阅参数值的有效范围。

查询语法 <通道>: BTWV(BursTWaVe)?

 <通道>: = {C1, C2}

响应格式 <通道>: BTWV <参数>

 <参数>: = {当前脉冲串的所有参数}

示例

设置 CH1 脉冲串状态为 ON:

C1:BTWW STATE,ON

设置 CH1 脉冲串周期为 1s:

C1:BTWW PRD,1

设置脉冲串延时为 1s:

C1:BTWW DLAY,1

设置 CH1 脉冲串为无限:

C1:BTWW TIME,INF

读取状态为 ON 的 CH2 脉冲串参数:

C2:BTWW?

返回值:

*C2:BTWW STATE,ON,PRD,0.01S,STPS,0,TRSR,INT,
TRMD,OFF,TIME,1,DLAY,2.4e-07S,GATE_NCYC,NCYC,
CARR,WVTP,SINE,FRQ,1000HZ,AMP,4V,OFST,0V,PHSE,0*

读取状态为 OFF 的 CH2 脉冲串参数:

C2:BTWW?

返回值:

C2:BTWW STATE,OFF

3.9 参数复制命令

描述	该命令将一个通道的参数复制到另一通道上。
命令语法	<p>ParaCoPy <目标通道>, <通道源></p> <p><目标通道>: = {C1, C2}.</p> <p><通道源>: = {C1, C2}.</p> <p>备注: C1 和 C2 的参数必须一起设置到设备。</p>
示例	<p>通道 1 的参数复制到通道 2 上。</p> <p><i>PACP C2,C1</i></p>

3.10 任意波形命令

3.10.1 任意波形切换波形命令

描述	该命令用于设置或者读取任意波的类型
命令语法	<p>格式 1: <通道>:ArbWaVe INDEX, <索引号></p> <p>格式 2: <通道>:ArbWaVe NAME, <名称></p> <p>格式 3: <通道>:ArbWaVe NAME, <路径></p> <p><通道>: = {C1, C2}.</p> <p><索引号>: 下表中任意波的索引号</p> <p><名称>: 下表的任意波名称</p> <p><路径>: 波形路径</p>
查询语法	<p><通道>: ARbWaVe?</p> <p><通道>: = {C1, C2}.</p>
响应格式	<通道>: ARWV INDEX, <索引号>, NAME, <名称>
示例	<p>通过索引号 2 设置 CH1 的当前波形:</p> <pre>C1:ARWV INDEX,2</pre> <p>读取 CH1 的当前波形:</p> <pre>C1:ARWV?</pre> <p>返回值:</p> <pre>C1:ARWV INDEX,2,NAME,StairUp</pre> <p>通过波形名称设置 CH1 的波形为心波:</p> <pre>C1:ARWV NAME,"Cardiac"</pre> <p>通过波形路径设置 CH1 的波形:</p> <pre>C1:ARWV NAME, "Local/wave1.bin"</pre> <pre>C1:ARWV NAME, "Local/wave2.mat"</pre> <pre>C1:ARWV NAME, "Local/wave3.csv"</pre> <pre>C1:ARWV NAME, "net_storage/wave4.bin"</pre> <pre>C1:ARWV NAME, "U-disk0/wave1.bin"</pre>
注	具体路径参照文件管理器中路径

相关命令

[STL](#)

索引号	名称	索引号	名称	索引号	名称	索引号	名称
0	Sine	51	AttALT	102	LFPulse	153	Duty18
1	Noise	52	RoundHalf	103	Tens1	154	Duty20
2	StairUp	53	RoundsPM	104	Tens2	155	Duty22
3	StairDn	54	BlaseiWave	105	Tens3	156	Duty24
4	Stairud	55	DampedOsc	106	Airy	157	Duty26
5	Ppulse	56	SwingOsc	107	Besselj	158	Duty28
6	Npulse	57	Discharge	108	Bessely	159	Duty30
7	Trapezia	58	Pahcur	109	Dirichlet	160	Duty32
8	Upramp	59	Combin	110	Erf	161	Duty34
9	Dnramp	60	SCR	111	Erfc	162	Duty36
10	ExpFal	61	Butterworth	112	Erfclnv	163	Duty38
11	ExpRise	62	Chebyshev1	113	Erflnv	164	Duty40
12	Logfall	63	Chebyshev2	114	Laguerre	165	Duty42
13	Logrise	64	TV	115	Legend	166	Duty44
14	Sqrt	65	Voice	116	Versiera	167	Duty46
15	Root3	66	Surge	117	Weibull	168	Duty48
16	X^2	67	NA	118	LogNormal	169	Duty50
17	X^3	68	Ripple	119	Laplace	170	Duty52
18	Sinc	69	Gamma	120	Maxwell	171	Duty54
19	Gaussian	70	StepResp	121	Rayleigh	172	Duty56
20	Dlorentz	71	BandLimited	122	Cauchy	173	Duty58
21	Haversine	72	CPulse	123	CosH	174	Duty60
22	Lorentz	73	CWPulse	124	CosInt	175	Duty62
23	Gauspuls	74	GateVibr	125	CotH	176	Duty64
24	Gmonopuls	75	LFMPulse	126	CscH	177	Duty66
25	Tripuls	76	MCNoise	127	SecH	178	Duty68

索引号	名称	索引号	名称	索引号	名称	索引号	名称
26	Cardiac	77	AM	128	SinH	179	Duty70
27	Quake	78	FM	129	SinInt	180	Duty72
28	Chirp	79	PFM	130	TanH	181	Duty74
29	Twotone	80	PM	131	ACosH	182	Duty76
30	SNR	81	PWM	132	ASecH	183	Duty78
31	Hamming	82	EOG	133	ASinH	184	Duty80
32	Hanning	83	EEG	134	ATanH	185	Duty82
33	Kaiser	84	EMG	135	ACsch	186	Duty84
34	Blackman	85	Pulseilogram	136	ACoth	187	Duty86
35	Gausswin	86	ResSpeed	137	Bartlett	188	Duty88
36	Triang	87	ECG1	138	BohmanWin	189	Duty90
37	BlackmanH	88	ECG2	139	ChebWin	190	Duty92
38	Bartlett-Hann	89	ECG3	140	FlatTopWin	191	Duty94
39	Tan	90	ECG4	141	ParzenWin	192	Duty96
40	Cot	91	ECG5	142	TaylorWin	193	Duty98
41	Sec	92	ECG6	143	TukeyWin	194	Duty99
42	Csc	93	ECG7	144	Duty01	195	demo1_375
43	Asin	94	ECG8	145	Duty02	196	demo1_16k
44	Acos	95	ECG9	146	Duty04	197	demo2_3k
45	Atan	96	ECG10	147	Duty06	198	demo2_16k
46	Acot	97	ECG11	148	Duty08		
47	Square	98	ECG12	149	Duty10		
48	SineTra	99	ECG13	150	Duty12		
49	SineVer	100	ECG14	151	Duty14		
50	AmpALT	101	ECG15	152	Duty16		

3.10.2 任意波形标记设置命令

描述	此命令用于设置任意波形标记开关
命令语法	格式: <通道>:MSWitch <状态>
查询语法	格式: <通道>:MSWitch?
响应格式	<状态>
示例	<p>打开标记开关:</p> <p><i>C1:MSWitch ON</i></p> <p>读取同步功能状态:</p> <p><i>C1:MSWitch?</i></p> <p>返回值:</p> <p><i>ON</i></p>
描述	此命令用于设置任意波形标记位置
命令语法	格式: <通道>:MPOS <参数>
查询语法	格式: <通道>:MPOS?
响应格式	<参数>
示例	<p>设置标记位置为 100:</p> <p><i>C1:MPOS 100</i></p> <p>读取标记位置:</p> <p><i>C1:MPOS?</i></p> <p>返回值:</p> <p><i>100</i></p>

3.11 同步命令

描述	此命令用于设置同步信号。
命令语法	<通道>:SYNC <状态> <通道>:={C1,C2}. <状态>:={ON,OFF}.
查询语法	<通道>:SYNC? <通道>:={C1, C2}.
响应格式	<通道>:SYNC <状态>
示例	打开 CH1 的同步功能: <i>C1:SYNC ON</i> 读取 CH1 的同步功能状态 <i>C1:SYNC?</i> 返回值: <i>C1:SYNC ON,TYPE,CH1</i>

描述	此命令用于设置同步信号类型。
命令语法	<通道>:SYNC <参数>, <值> <通道>:={C1, C2} <参数>:={TYPE} <值>:={CH1, CH2, MOD_CH1, MOD_CH2}
示例	输出 CH1 同步信号: <i>C1:SYNC TYPE, CH1</i>

3.12 同相位命令

描述	此命令用于设置两个通道的相位同步
命令语法	EQPHASE
响应格式	EQPHASE <状态>
示例	设置同相位: <i>EQPHASE</i>

3.13 语言命令

描述	此命令设置或者获取系统语言。
命令语法	LanGuaGe <语言> <语言>: = {EN, CH}, 其中 EN 是英语, CH 是中文
查询语法	LAnGuaGe?
响应格式	LAGG <语言>
示例	设置语言为英语: <i>LAGG EN</i> 读取当前系统语言: <i>LAGG?</i> 返回值: <i>LAGG EN</i>

3.14 配置命令

描述	设置或者获取上电开机模式
命令语法	格式 1: Sys_CFG<模式> <模式>: = {DEFAULT (默认), LAST (上次), USER (用户) } 格式 2: Sys_CFG <配置>, <路径> <配置>: = {USER (选择开机恢复文件), PRESET (选择恢复文件) } <路径>: = {用户存储 (本地, 网络存储, U 盘) 的配置文件的的路径, 包括文件名和后缀}
查询语法	Sys_CFG?
响应格式	SCFG<模式> SCFG<配置>, <路径>
示例 1	设置上电开机系统为上次: <i>SCFG LAST</i>
示例 2	设置开机恢复文件: <i>SCFG USER, "net_storage/config/state.xml"</i> 或: <i>SCFG USER, "U-disk0/config/state.xml"</i> 或: <i>SCFG USER, "Local/state.xml"</i>

示例 3	设置恢复文件 <i>SCFG PRESET,RESETconfig/state.xmlig/state.</i> 或: <i>SCFG PRESET,RU-disk0/config/state.xmlli</i> 或: <i>SCFG PRESET,RESETconfig/state</i>
------	---

备注 1: 路径必须使用双引号, 英文包括且必须添加后缀名 .xml。具体可用路径请参考文件管理器。

3.15 日期和时间命令

描述	此命令用于设置设备的日期和时间
命令语法	SYST:DATE <日期> <日期>: = {要设置的日期, 格式: yyyy/mm/dd}. SYST: TIME <时间> <时间>: = {要设置的时间, 格式: hh/mm/ss}.
查询语法	SYST:DATE? SYST:TIME?
示例 1	设置日期为 2021/01/10 <i>SYST:DATE 20210110</i>
示例 2	设置时间为 10:06:32 <i>SYST:TIME 100632</i>

3.16 开机上电模式命令

描述	此命令用于设置上电直接开机或按键开机
命令语法	POWER:ON:MODE<值> <值>: = {1, 2}. 模式 1: 按键开机 模式 2: 上电直接开机
查询语法	POWER:ON:MODE?
响应格式	POWER:ON:MODE<值>
示例	设置上电直接开机: <i>POWER:ON:MODE 2</i>

3.17 按键命令

描述	此命令用于打开或者关闭前面板按键
命令语法	KEY<状态> <状态>: = {ON, OFF}.
查询语法	KEY?
响应格式	KEY<状态>
示例	打开前面板按键: <i>KEY ON</i>

3.18 蜂鸣器命令

描述	此命令用于打开或者关闭蜂鸣器
命令语法	BUZZer<状态> <状态>: = {ON, OFF}.
查询语法	BUZZer?
响应格式	BUZZ<状态>
示例	打开蜂鸣器: <i>BUZZ ON</i>

3.19 通道触发源设置命令

描述	此命令用于手动触发时，设置是否同时触发两通道。
命令语法	TRIGger:TYPE <参数> <参数>: = {ALONE, ALL}.
查询语法	TRIGger:TYPE?
示例	设置手动触发时，同时触发两通道: <i>TRIGger:TYPE ALL</i> 读取当前的通道触发设置状态: <i>TRIGger:TYPE?</i> 返回值: <i>ALL\n</i>

3.20 屏幕保护命令

描述	此命令用于关闭或者设置屏幕保护时间（单位为分钟）。
命令语法	SCreen_SaVe <参数> <参数>:= {OFF, 1, 5, 15, 30, 60, 120, 300}.
查询语法	SCreen_SaVe?
响应格式	SCSV<参数>
示例	设置屏幕保护时间为 5 分钟： <i>SCSV 5</i> 读取当前的屏幕保护时间： <i>SCreen_SaVe?</i> 返回值： <i>SCSV 5MIN</i>

3.21 时钟源命令

描述	此命令设置或者获取时钟源，或设置 10MHZ 时钟输出。
命令语法	格式 1: ROscillator <src> <src>:= {INT (内部时钟), EXT (外部时钟) } 格式 2: ROscillator 10MOUT, <state> <state>:= { ON, OFF } 时钟输出状态
查询语法	ROscillator?
响应格式	ROSC <src>,10MOUT,<state>
示例	设置内部时基作为时钟源： <i>ROSC INT</i> 打开 10MHz 时钟输出： <i>ROSC 10MOUT,ON</i>

3.22 计数器命令

命令格式	:SENSe:COUNTer
描述	用于设置计数器各项参数的设置与获取。 该命令设置计数器配置
命令语法	:SENSe:COUNTer:CONFig: <参数> <值> <参数>:= {下表参数} <值>:= {相关参数的值}.

参数	值	描述
STATE	<state>	= {OFF, ON} 或 {0, 1} 开启或关闭计数器
MODE	<state>	:= {FREQuency, TOTALizer} 频率计模式与计数器模式
COUPLing	<mode>	:= {AC, DC}, 耦合模式
HFREJect	<HFR>	:= {OFF, ON} 或 {0, 1}, 高频抑制状态
TLEVel	<triglev><unit>	触发电平。有效值的范围取决于机型。单位是伏特“V”。 <单位> := {V, mV, uV}
SEXIT	<mode>	= {OFF, ON} 或 {0, 1}, 开启或关闭
PAUSe	<state>	= {OFF, ON}, 暂停开关

查询语法	:SENSe:COUNTer:CONFig: <参数>?
响应格式	<模式>
示例	设置计数器耦合模式为交流模式： <i>:SENSe:COUNTer:CONFig:COUPLing AC</i>

描述	该命令设置频率计模式下测量参数
命令语法	:SENSe:COUNTer:FREQuency: <参数> <值> <参数>:= {下表参数} <值>:= {相关参数的值}.

参数	值	描述
MEASure	< type >	<类型>:={FREQ, PERIOD, DUTY_CYCLE}, 频率、周期、占空比
RFREQuency	<frequency><unit>	<单位>:={Hz, MHz, GHz}。默认单位是赫兹“HZ”。

查询语法 :SENSe:COUNTer:FREQuency:MEASure[type]?

响应格式 <模式>

示例 设置频率计模式下的测量类型为周期:

:SENSe:COUNTer:FREQuency:MEASure PERIOD

描述 该命令用于查询频率计模式下测量的结果

命令语法 :SENSe:COUNTer:FREQuency: <测量类型>:<参数><[类型]>?
 <测量类型>:={缺省, PERiod, DUTY} 频率、周期、占空比
 <参数>:={缺省, SNUMBer, MEAN, MAX, MIN, SDEViation}
 测量类型的实时值, 测量类型的采样次数, 测量类型的平均值, 测量类型的最大值, 测量类型的最小值, 测量类型的标准差
 <类型>:={FDEViation}.仅在测量类型是频率模式下可用

查询语法 :SENSe:COUNTer:FREQuency:SNUMBer?

响应格式 <模式>

描述 该命令用于设置计数器模式下测量参数

命令语法 :SENSe:COUNTer:TOTAlizer: <参数> <值>
 <参数>:={下表参数}
 <值>:={相关参数的值}

参数	值	描述
GATE:STATe	<type>	<状态>:={OFF, ON}或{0, 1}开启或关闭
EDGE	<edge>	<边沿>:={RISE, FALL}, 上升沿或下降沿
GATE:MODE	<mode>	<模式>:={LEVEL, AFTER_EDGE}, 电平或边沿后模式
GOLarity:GATE:POLarity	<polarity>	<极性>:={NEGative, POSitive}, 负极性或正极性
GOLarity:GATE:EDGE	<edge>	<边沿>:={RISE, FALL}, 上升沿或下降沿

3.22.1 :SENSe:COUNTER:CLEAr

描述	该命令用于清除计数器模式下测量的累计值结果
命令语法	:SENSe:COUNTER:CLEAr
示例	清除计数器模式下测量的结果： <i>:SENSe:COUNTER:CLEAr</i>

3.22.2 :SENSe:COUNTER:FREQuency:MEASure[:type]

描述	该命令设置频率计模式下的测量类型
命令语法	:SENSe:COUNTER:FREQuency:MEASure[:type] <类型> <类型>:= {FREQ, PERIOD, DUTY_CYCLE}, 频率、周期、占空比
查询语法	:SENSe:COUNTER:FREQuency:MEASure[:type]?
响应格式	<类型>
示例	设置频率计模式下的测量类型为周期： <i>:SENSe:COUNTER:FREQuency:MEASure PERIOD</i>
注	此命令仅在计数器模式为频率计时有效

3.22.3 :SENSe:COUNTER:FREQuency:RFREQuency

描述	该命令设置频率计模式下的参考频率
命令语法	:SENSe:COUNTER:FREQuency:RFREQuency <频率><单位> <频率>:= 频率计的参考频率，该参数的有效范围请查阅数据手册。 <单位>:= {Hz, MHz, GHz}。默认单位是赫兹“Hz”
查询语法	:SENSe:COUNTER:FREQuency:RFREQuency?
响应格式	<频率> (用 Hz 表示)
示例	设置频率计模式下的参考频率为 1MHz： <i>:SENSe:COUNTER:FREQuency:RFREQuency 1MHz</i>
注:	此命令仅在计数器模式下测量类型为频率时有效

3.22.4 :SENSe:COUNTer:FREQuency?

描述	该命令用于查询频率计模式下测量的频率结果
查询语法	:SENSe:COUNTer:FREQuency?
响应格式	<频率> (用 Hz 表示)
示例	查询频率计模式下测量的频率结果: :SENSe:COUNTer:FREQuency? 返回值: 9999996.75781744
注	此命令仅在频率计模式下有效

3.22.5 :SENSe:COUNTer:FREQuency:SNUMBer?

描述	该命令用于查询频率计模式下采样次数结果
查询语法	:SENSe:COUNTer:FREQuency:SNUMBer?
响应格式	<采样次数>
示例	查询频率计模式下采样次数结果: :SENSe:COUNTer:FREQuency:SNUMBer? 返回值: 2294
注	此命令仅在频率计模式下有效

3.22.6 :SENSe:COUNTer:FREQuency:FDEVIation?

描述	该命令用于查询频率计模式下测量的频率偏差结果
查询语法	:SENSe:COUNTer:FREQuency:FDEVIation?
响应格式	<频率偏差> (用 ppm 表示)
示例	查询频率计模式下测量的频率偏差结果: :SENSe:COUNTer:FREQuency:FDEVIation? 返回值: -0.324020794406533
注	此命令仅在频率计模式下有效

3.22.7 :SENSe:COUNTer:FREQuency:MEAN?

描述	该命令用于查询频率计模式下测量的频率平均值结果
查询语法	:SENSe:COUNTer:FREQuency:MEAN?
响应格式	<频率> (用 Hz 表示)
示例	查询频率计模式下测量的频率结果： :SENSe:COUNTer:FREQuency:MEAN? 返回值： 9999996.79101083
注	此命令仅在频率计模式下有效

3.22.8 :SENSe:COUNTer:FREQuency:MEAN:FDEVIation?

描述	该命令用于查询频率计模式下测量的频率平均值偏差结果
查询语法	:SENSe:COUNTer:FREQuency:MEAN:FDEVIation?
响应格式	<频率偏差> (用 ppm 表示)
示例	查询频率计模式下测量的频率平均值偏差结果： :SENSe:COUNTer:FREQuency:MEAN:FDEVIation? 返回值： -0.322511510334804
注	此命令仅在频率计模式下有效

3.22.9 :SENSe:COUNTer:FREQuency:MAX?

描述	该命令用于查询频率计模式下测量的频率最大值结果
查询语法	:SENSe:COUNTer:FREQuency:MAX?
响应格式	<频率> (用 Hz 表示)
示例	查询频率计模式下测量的频率最大值结果： :SENSe:COUNTer:FREQuency:MAX? 返回值： 9999996.8775536
注	此命令仅在频率计模式下有效

3.22.10 :SENSe:COUNTer:FREQuency:MAX:FDEVIation?

描述	该命令用于查询频率计模式下测量的频率最大值偏差结果
查询语法	:SENSe:COUNTer:FREQuency:MAX:FDEVIation?
响应格式	<频率偏差> (用 ppm 表示)
示例	查询频率计模式下测量的频率最大值偏差结果： :SENSe:COUNTer:FREQuency:MAX:FDEVIation? 返回值： -0.312244639918208
注	此命令仅在频率计模式下有效

3.22.11 :SENSe:COUNTer:FREQuency:MIN?

描述	该命令用于查询频率计模式下测量的频率最小值结果
查询语法	:SENSe:COUNTer:FREQuency:MIN?
响应格式	<频率> (用 Hz 表示)
示例	查询频率计模式下测量的频率最小值结果： :SENSe:COUNTer:FREQuency:MIN? 返回值： 9999996.67704201
注	此命令仅在频率计模式下有效

3.22.12 :SENSe:COUNTer:FREQuency:MIN:FDEVIation?

描述	该命令用于查询频率计模式下测量的频率最小值偏差结果
查询语法	:SENSe:COUNTer:FREQuency:MIN:FDEVIation?
响应格式	<频率偏差> (用 ppm 表示)
示例	查询频率计模式下测量的频率最小值偏差结果： :SENSe:COUNTer:FREQuency:MIN:FDEVIation? 返回值： -0.332295799069107
注	此命令仅在频率计模式下有效

3.22.13 :SENSe:COUNTer:FREQuency:SDEVIation?

描述	该命令用于查询频率计模式下测量的频率标准差结果
查询语法	:SENSe:COUNTer:FREQuency:SDEVIation?
响应格式	<频率> (用 Hz 表示)
示例	查询频率计模式下测量的频率标准差结果： :SENSe:COUNTer:FREQuency:SDEVIation? 返回值： 0.395284707521047
注	此命令仅在频率计模式下有效

3.22.14 :SENSe:COUNTe:FREQuency:SDEVIation:FDEVIation?

描述	该命令用于查询频率计模式下测量的频率标准差偏差结果
查询语法	:SENSe:COUNTe:FREQuency:SDEVIation:FDEVIation?
响应格式	<频率偏差> (用 ppm 表示)
示例	查询频率计模式下测量的频率标准差偏差结果： :SENSe:COUNTe:FREQuency:SDEVIation:FDEVIation? 返回值： -0.332295799069107
注	此命令仅在频率计模式下有效

3.22.15 :SENSe:COUNTer:FREQuency:PERiod?

描述	该命令用于查询频率计模式下测量的周期结果
查询语法	:SENSe:COUNTer:FREQuency:PERiod?
响应格式	<周期> (用 S 表示)
示例	查询频率计模式下测量的周期结果： :SENSe:COUNTer:FREQuency:PERiod? 返回值： 1.00000032802047e-07
注	此命令仅在频率计模式下有效

3.22.16 :SENSe:COUNTer:FREQuency:PERiod:MEAN?

描述	该命令用于查询频率计模式下测量的周期平均值结果
查询语法	:SENSe:COUNTer:FREQuency:PERiod:MEAN?
响应格式	<周期> (用 S 表示)
示例	查询频率计模式下测量的周期平均值结果： :SENSe:COUNTer:FREQuency:PERiod:MEAN? 返回值： 1.00000032802047e-07
注	此命令仅在频率计模式下有效

3.22.17 :SENSe:COUNTer:FREQuency:PERiod:MAX?

描述	该命令用于查询频率计模式下测量的周期最大值结果
查询语法	:SENSe:COUNTer:FREQuency:PERiod:MAX?
响应格式	<周期> (用 S 表示)
示例	查询频率计模式下测量的周期最大值结果： :SENSe:COUNTer:FREQuency:PERiod:MAX? 返回值： 1.00000033229591e-07
注	此命令仅在频率计模式下有效

3.22.18 :SENSe:COUNTer:FREQuency:PERiod:MIN?

描述	该命令用于查询频率计模式下测量的周期最小值结果
查询语法	:SENSe:COUNTer:FREQuency:PERiod:MIN?
响应格式	<周期> (用 S 表示)
示例	查询频率计模式下测量的周期最小值结果： :SENSe:COUNTer:FREQuency:PERiod:MIN? 返回值： 1.00000031224474e-07
注	此命令仅在频率计模式下有效

3.22.19 :SENSe:COUNTer:FREQuency:PERiod:SDEVIation?

描述	该命令用于查询频率计模式下测量的周期标准差结果
查询语法	:SENSe:COUNTer:FREQuency:PERiod:SDEVIation?
响应格式	<周期> (用 S 表示)
示例	查询频率计模式下测量的周期标准差结果： :SENSe:COUNTer:FREQuency:PERiod:SDEVIation? 返回值： 6.52675178645049e-15
注	此命令仅在频率计模式下有效

3.22.20 :SENSe:COUNTer:FREQuency:DUTY?

描述	该命令用于查询频率计模式下测量的占空比结果
查询语法	:SENSe:COUNTer:FREQuency:DUTY?
响应格式	<占空比> (用百分比表示)
示例	查询频率计模式下测量的占空比结果： :SENSe:COUNTer:FREQuency:DUTY? 返回值： 49.8178520697226
注	此命令仅在频率计模式下有效

3.22.21 :SENSe:COUNTer:FREQuency:DUTY:MEAN?

描述	该命令用于查询频率计模式下测量的占空比平均值结果
查询语法	:SENSe:COUNTer:FREQuency:DUTY:MEAN?
响应格式	<占空比> (用百分比表示)
示例	查询频率计模式下测量的占空比平均值结果： :SENSe:COUNTer:FREQuency:DUTY:MEAN? 返回值： 49.8204827053384
注	此命令仅在频率计模式下有效

3.22.22 :SENSe:COUNTer:FREQuency:DUTY:MAX?

描述	该命令用于查询频率计模式下测量的占空比最大值结果
查询语法	:SENSe:COUNTer:FREQuency:DUTY:MAX?
响应格式	<占空比> (用%百分比表示)
示例	查询频率计模式下测量的占空比最大值结果: :SENSe:COUNTer:FREQuency:DUTY:MAX? 返回值: 1.00000033229591e-07
注	此命令仅在频率计模式下有效

3.22.23 :SENSe:COUNTer:FREQuency:DUTY:MIN?

描述	该命令用于查询频率计模式下测量的占空比最小值结果
查询语法	:SENSe:COUNTer:FREQuency:DUTY:MIN?
响应格式	<占空比> (用%百分比表示)
示例	查询频率计模式下测量的占空比最小值结果: :SENSe:COUNTer:FREQuency:DUTY:MIN? 返回值: 49.8071405616451
注	此命令仅在频率计模式下有效

3.22.24 :SENSe:COUNTer:FREQuency:DUTY:SDEViation?

描述	该命令用于查询频率计模式下测量的占空比标准差结果
查询语法	:SENSe:COUNTer:FREQuency:DUTY:SDEViation?
响应格式	<占空比> (用%百分比表示)
示例	查询频率计模式下测量的周期标准差结果: :SENSe:COUNTer:FREQuency:DUTY:SDEViation? 返回值: 0.00505753455971934
注	此命令仅在频率计模式下有效

3.22.25 :SENSe:COUNTer:TOTalizer:GATE:STATe

描述	该命令用于设置计数器模式下的计数器门控状态
命令语法	:SENSe:COUNTer:TOTalizer:GATE:STATe <状态> <状态>:= {OFF, ON} 或 {0, 1}开启或关闭
查询语法	:SENSe:COUNTer:TOTalizer:GATE:STATe?
响应格式	<状态>
示例	设置计数器模式下的计数器触门控状态为 开启: <i>:SENSe:COUNTer:TOTalizer:GATE:STATe ON</i>
注	此命令仅在计数器模式为计数器时有效

3.22.26 :SENSe:COUNTer:TOTalizer:EDGE

描述	该命令用于设置计数器模式下的计数器触发边沿类型
命令语法	:SENSe:COUNTer:TOTalizer:EDGE <边沿> <边沿>:= {RISE, FALL}, 上升沿或下降沿
查询语法	:SENSe:COUNTer:TOTalizer:EDGE?
响应格式	<边沿>
示例	设置计数器模式下的计数器触发边沿类型为上升沿: <i>:SENSe:COUNTer:TOTalizer:EDGE RISE</i>
注	此命令仅在计数器模式为计数器时有效

3.22.27 :SENSe:COUNTer:TOTalizer:GATE:MODE

描述	该命令用于设置计数器模式下的计数器门控模式
命令语法	:SENSe:COUNTer:TOTalizer:GATE:MODE <模式> <模式>:= {LEVEL, AFTER_EDGE}, 电平或边沿后模式
查询语法	:SENSe:COUNTer:TOTalizer:GATE:MODE?
响应格式	<模式>
示例	设置计数器模式下的计数器门控模式为电平模式 <i>:SENSe:COUNTer:TOTalizer:GATE:MODE LEVEL</i>
注	此命令仅在计数器模式为计数器时有效

3.22.28 :SENSe:COUNTer:TOTalizer:GOLarity:GATE:POLarity

描述	该命令用于设置计数器模式下的计数器门控极性
命令语法	:SENSe:COUNTer:TOTalizer:GATE:POLarity <极性> <极性>:= {NEGative, POSitive}, 负性或正性
查询语法	:SENSe:COUNTer:TOTalizer:GATE:POLarity?
响应格式	<极性>
示例	设置计数器模式下的计数器门控极性为 正性: <i>:SENSe:COUNTer:TOTalizer:GATE:POLarity POSitive</i>
注	此命令仅在计数器模式为电平模式时有效

3.22.29 :SENSe:COUNTer:TOTalizer:GOLarity:GATE:EDGE

描述	该命令用于设置计数器模式下的计数器门控边沿
命令语法	:SENSe:COUNTer:TOTalizer:GATE:EDGE <边沿> <边沿>:= {RISE, FALL}, 上升沿或下降沿
查询语法	:SENSe:COUNTer:TOTalizer:GATE:EDGE?
响应格式	<边沿>
示例	设置计数器模式下的计数器门控边沿为 上升沿: <i>:SENSe:COUNTer:TOTalizer:GATE:EDGE RISE</i>
注	此命令仅在计数器模式为边沿后模式时有效

3.22.30 :SENSe:COUNTer:TOTalizer?

描述	该命令用于查询计数器模式下测量的累计值结果
查询语法	:SENSe:COUNTer:TOTalizer?
响应格式	<累计值> (用 Hits 表示)
示例	查询计数器模式下测量的累计值结果 <i>:SENSe:COUNTer:TOTalizer?</i> 返回值: <i>13364879033</i>
注	此命令仅在计数器模式下有效

3.23 反相命令

描述	此命令用于设置或者获取指定通道的极性。
命令语法	<通道>: INVerT <状态> <通道>: = {C1, C2}. <状态>: = {ON, OFF}.
查询语法	<通道>: INVerT? <通道>: = {C1, C2}.
响应格式	<通道>:INVT <状态>
示例	设置 CH1 的极性为反相: <i>C1:INVT ON</i> 读取 CH1 的极性: <i>C1:INVT?</i> 返回值: <i>C1:INVT ON</i>

3.24 数字滤波器命令

描述	此命令用于设置数字滤波器的开关和截止频率。	
命令语法	<通道>: FILTer<参数>, <值> <参数>: = {下表的参数}. <值>: = {相关参数的值}.	
	参数	描述
	STATE	<state> := {ON, OFF}, 数字滤波器状态
	COFF_FRQ	<cutoff_freq> := {100MHz 至 机器带宽}, 单位为赫兹, 用于设置数字滤波器的截止频率。
查询语法	<通道>: FILTer? <通道>: = {C1, C2}.	
响应格式	<通道>: FILT <参数>, <值>	
示例	设置 CH1 的数字滤波器开启: <i>C1:FILT STATE,ON</i> 设置 CH1 的数字滤波器截止频率 200MHZ: <i>C1:FILT COFF_FRQ,200000000</i>	

查询 CH1 的数字滤波器信息：

C1:FLT?

返回值：

STATE,OFF,COFF_FRQ,350000000HZ

3.25 通道耦合命令

描述	设置或者获取通道耦合参数。只有在追踪功能关闭时才能设置耦合值。	
命令语法	COUPling<参数>, <值> <参数>: = {下表的参数}. <值>: = {相关参数的值}.	
参数	值	描述
TRACE	<track_enable>	:= {ON, OFF}; 通道跟踪状态
STATE	<state>	:= {ON, OFF}; 通道耦合状态
FCOUP	<fcoup>	:= {ON, OFF}; 频率耦合状态
FDEV	<frq_dev>	:=两通道间的频率差值。单位是赫兹'Hz'
FRAT	<frat>	:=两通道间的频率比值
PCOUP	<pcoup>	:= {ON, OFF}; 相位耦合状态
PDEV	<pha_dev>	:=两通道间的相位差值。单位是'度'
PRAT	<prat>	:=两通道间的相位比值。
ACOUP	<acoup>	:= {ON, OFF}; 幅度耦合的状态
ARAT	<arat>	:=两通道间的幅度比值
ADEV	<adev>	:=两通道间的幅度偏差。单位为伏特，峰峰值'Vpp'。
查询语法	COUPling?	
响应格式	COUP <参数> <参数> := {所有耦合参数值}.	
示例	设置耦合状态为打开： <i>COUP STATE,ON</i> 设置频率偏差为 5Hz： <i>COUP FDEV,5</i>	

设置幅度比例为 2:

COUPARAT,2

查询耦合信息:

COUP?

返回值:

*COUPTRACE,OFF,FCOUP,ON,PCOUP,ON,ACOUP,ON,FDEV,
5HZ,PRAT,1,ARAT,2*

查询跟踪下的通道间相位差:

TRACK:DEV?

返回值:

PDEV,60\n

3.26 过压保护命令

描述	此命令用于设置或者获取过压保护状态。
命令语法	VOLTPRT<状态> <状态>: = {ON, OFF}
查询语法	VOLTPRT?
响应格式	VOLTPRT<状态>

3.27 过流保护命令

描述	此命令用于设置或者获取过流保护状态。
命令语法	CURRPRT<状态> <状态>: = {ON, OFF}
查询语法	CURRPRT?
响应格式	CURRPRT <状态>

3.28 过压过流状态查询

描述	此命令用于获取当前过载状态 (过压, 过流, 无)。
命令语法	VOLTSTAT?
查询语法	VOLTSTAT?

3.29 OUTPUT 时滞命令

描述	此命令用于设置或查询 output 时滞。
命令语法	<通道>:OUTPut:SKEW <值> <通道>={C1,C2} <值>={-0.2 至 0.2}, 单位“ns”
查询语法	<通道>:OUTPut:SKEW? <通道>={C1,C2}
响应格式	<值> <值>: ={当前通道设置的值}
示例	设置 C1 通道 output 时滞为 0.2ns: <i>C1:OUTPut:SKEW 0.2e-9</i> 读取 C1 通道 output 时滞: <i>C1:OUTPut:SKEW?</i> 返回值: <i>2e-10</i>

3.30 保存列表命令

描述	此命令用于读取存储波形数据的名称。若存储单位为空, 该命令将返回“EMPTY”字段。
查询语法	SToreList? SToreList? <参数> SToreList? <USER>, <路径> <参数>=: {下表所示参数} <路径>=: {指定网络存储或者 U 盘路径, 见下表}

响应格式 <波形名称>

参数		功能
BUILDIN	<参数>	查询内建波形名称及索引号
USER	<参数>	查询本地保存的波形名称

路径（仅为示例，可指定网络存储或 U 盘目录下任意路径）		功能
“net_storage/wave”	<路径>	查询网络存储路径下保存的波形名称
“U-disk0/wave”	<路径>	查询 U 盘 0 路径下保存的波形名称
“U-disk1/wave”	<路径>	查询 U 盘 1 路径下保存的波形名称
“U-disk2/wave”	<路径>	查询 U 盘 2 路径下保存的波形名称

示例

(1) 读取保存在设备中的所有任意波数据（不包括网络存储，U 盘中的数据）：

STL?

返回值：

STL M0, sine, M1, noise, M2, stairup, M3, staidn, M4, stairud, M5, ppulse, M6, npulse, M7, trapezia, M8, upramp, M9, dn ramp, M10, exp_fall, M11, exp_rise, M12, logfall, M13, logrise, M14, sqrt, M15, root3, M16, x^2, M17, x^3, M18, sinc, M19, gaussian, M20, dlorentz, M21, haversine, M22, lorentz, M23, gauspuls, M24, gmonopuls, M25, tripuls, M26, cardiac, M27, quake, M28, chirp, M29, twotone, M30, snr, M31, EMPTY, M32, EMPTY, M33, EMPTY, M34, hamming, M35, hanning, M36, kaiser, M37, blackman, M38, gaussiwin, M39, triangle, M40, blackmanharris, M41, bartlett, M42, tan, M43, cot, M44, sec, M45, csc, M46, asin, M47, acos, M48, atan, M49, acot, M50, EMPTY, M51, EMPTY, M52, EMPTY, M53, DDROPOUT, M54, FCLK1, M55, FSDA1, M56, EMPTY, M57, EMPTY, M58, EMPTY, M59, EMPTY

(2) 读取保存在设备中的内建波形数据：

STL? BUILDIN

返回值：

STL M10, ExpFal, M100, ECG14, M101, ECG15, M102, LFPulse,

M103, Tens1, M104, Tens2, M105, Tens3, M106, Airy, M107, Besselj, M108, Bessely, M109, Dirichlet, M11, ExpRise, M110, Erf, M111, Erfc, M112, Erfclnv, M113, Erflnv, M114, Laguerre, M115, Legend, M116, Versiera, M117, Weibull, M118, LogNormal, M119, Laplace, M12, LogFall, M120, Maxwell, M121, Rayleigh, M122, Cauchy, M123, CosH, M124, CosInt, M125, CotH, M126, Csch, M127, SecH, M128, SinH, M129, SinInt, M13, LogRise, M130, TanH, M131, ACosH, M132, ASecH, M133, ASinH, M134, ATanH, M135, ACsch, M136, ACoth, M137, Bartlett, M138, BohmanWin, M139, ChebWin, M14, Sqrt, M140, FlattopWin, M141, ParzenWin, M142, TaylorWin, M143, TukeyWin, M144, SquareDuty01, M145, SquareDuty02, M146, SquareDuty04, M147, SquareDuty06, M148, SquareDuty08, M149, SquareDuty10, M15, Root3, M150, SquareDuty12, M151, SquareDuty14, M152, SquareDuty16, M153, SquareDuty18, M154, SquareDuty20, M155, SquareDuty22, M156, SquareDuty24, M157, SquareDuty26, M158, SquareDuty28, M159, SquareDuty30, M16, X^2, M160, SquareDuty32, M161, SquareDuty34, M162, SquareDuty36, M163, SquareDuty38, M164, SquareDuty40, M165, SquareDuty42, M166, SquareDuty44, M167, SquareDuty46, M168, SquareDuty48, M169, SquareDuty50, M17, X^3, M170, SquareDuty52, M171, SquareDuty54, M172, SquareDuty56, M173, SquareDuty58, M174, SquareDuty60, M175, SquareDuty62, M176, SquareDuty64, M177, SquareDuty66, M178, SquareDuty68, M179, SquareDuty70, M18, Sinc, M180, SquareDuty72, M181, SquareDuty74, M182, SquareDuty76, M183, SquareDuty78, M184, SquareDuty80, M185, SquareDuty82, M186, SquareDuty84, M187, SquareDuty86, M188, SquareDuty88, M189, SquareDuty90, M19, Gaussian, M190, SquareDuty92, M191, SquareDuty94, M192, SquareDuty96, M193, SquareDuty98, M194, SquareDuty99, M195, demo1_375pts, M196, demo1_16kpts, M197, demo2_3kpts, M198, demo2_16kpts, M2, StairUp, M20, Dlorentz, M21, Haversine, M22, Lorentz, M23, Gauspuls, M24, Gmonopuls, M25, Tripuls, M26, Cardiac, M27, Quake, M28, Chirp, M29, Twotone, M3, StairDn, M30, SNR, M31, Hamming, M32, Hanning, M33, kaiser, M34, Blackman, M35, Gausswin, M36, Triangle, M37, Bartlett-Hann, M38, Bartlett, M39, Tan, M4, StairUD, M40, Cot, M41, Sec, M42, Csc, M43, Asin, M44, Acos, M45, Atan, M46, Acot, M47, Square, M48, SineTra, M49, SineVer, M5, Ppulse, M50, AmpALT, M51, AttALT, M52, RoundHalf, M53, RoundsPM, M54, BlaseiWave, M55, DampedOsc, M56, SwingOsc, M57, Discharge, M58, Pahcur, M59, Combin, M6, Npulse, M60, SCR, M61, Butterworth, M62, Chebyshev1, M63,

Chebyshev2, M64, TV, M65, Voice, M66, Surge, M67, Na, M68, Ripple, M69, Gamma, M7, Trapezia, M70, StepResp, M71, BandLimited, M72, CPulse, M73, CWPulse, M74, GateVibr, M75, LFMPulse, M76, MCNoise, M77, AM, M78, FM, M79, PFM, M8, Upramp, M80, PM, M81, PWM, M82, EOG, M83, EEG, M84, EMG, M85, Pulseilogram, M86, ResSpeed, M87, ECG1, M88, ECG2, M89, ECG3, M9, Dnramp, M90, ECG4, M91, ECG5, M92, ECG6, M93, ECG7, M94, ECG8, M95, ECG9, M96, ECG10, M97, ECG11, M98, ECG12, M99, ECG13

(3) 读取来自设备中的用户定义的波形名称:

STL? USER

返回值:

*STLWVNM,sinc_8M,sinc_3000000,sinc_1664000,
ramp_8M,sinc_2000000,sinc_50000,square_8M,sinc_5000,wave1,
square_1M*

(4) 读取来自网络存储的波形名称:

STL? USER, "net_storage/wave"

返回值:

*net_storage/wave,STLWVNM,AutoWave2,wave1,AutoWave,
ExpFal, test-sq,Besselj,libEasyLib*

3.31 任意波数据命令

描述	此命令用于设置或者获取任意波形数据。
命令语法	<p><通道>: WVDT, <参数>, <值></p> <p><通道>: = {C1, C2, DIG}.</p> <p><参数>: = {下表的参数}.</p> <p><值>: = {相关参数的值}.</p>

参数	值	描述
WVNM	<波形名>	:= 波形名称
LENGTH	<长度>	:= 波形的字节数, 有效范围取决于机型。详细信息查阅备注 1.
FREQ	<频率>	:= 频率。单位为赫兹“Hz”。
AMPL	<幅值>	:= 幅值。单位是伏特, 峰峰值“Vpp”。
OFST	<偏置>	:= 偏置。单位是伏特“V”。
PHASE	<相位>	:= 相位。单位是“度”。
WAVEDATA	<波形数据>	:= 波形数据 写入到波形中的数据。

查询语法	<p>格式 1: WVDT? Mn</p> <p>格式 2: WVDT? USER, <波形名称></p> <p>格式 3: WVDT? USER, <路径>, <波形名称></p> <p><路径>指定存储路径, 如 U 盘路径, 网络存储路径见下表示例</p> <p><波形名称>: = {用户定义的波形名称}.</p>
-------------	---

示例	<p>示例 1: <i>WVDT?lsUSER,"net_storage/wave",wave1</i></p> <p>返回值: <i>WVDT POS, net_storage/wave, WVNM, wave1, LENGTH, 300B, TYPE, 10, WAVEDATA,</i></p> <p>示例 2: 将数据 0x6000c0006000 写到‘wave1’波形中并发送到通道 1 <i>C1:WVDT WVNM,"wave1",WAVEDATA, b'0x6000c0006000'</i></p>
-----------	---

备注 1:

- 1) 使用路径查询功能必须为网络存储或者 U 盘目录下，使用命令时必须在路径两端添加英文引号。
- 2) 网络存储或 U 盘顶层目录可在文件管理器中获取。
- 3) 不指定路径默认为本地路径（格式 2）。
- 4) 尽量固定存放波形文件的目录，以减少因文件检索带来的等待时间。
- 5) 下表展示了不同存储方式的路径，仅为示例。

路径		功能
“net_storage/wave”	<路径>	读取网络存储路径下的波形文件信息
“U-disk0/wave”	<路径>	读取 U 盘 0 路径下的波形文件信息
“U-disk1/wave”	<路径>	读取 U 盘 1 路径下的波形文件信息
“U-disk2/wave”	<路径>	读取 U 盘 2 路径下的波形文件信息

3.32 序列波命令

3.32.1 <channel>:ARBMode <Mode>

描述	该命令用于设置(查询)ARB 波形的工作模式
命令语法	<pre><channel>:ARBMode <mode> <channel>:= {C1, C2}. <mode>:= {AFG, AWG}.</pre>
查询语法	<pre><channel>:ARBMode? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1ARB 为 AFG 模式</p> <pre><i>:C1:ARBMode AFG</i></pre> <p>读取通道 1 的模式</p> <pre><i>:C1:ARBMode?</i></pre> <p>返回值:</p> <pre><i>"AFG"</i></pre>

3.32.2 <channel>:SEquence <switch>

描述	该命令用于设置(查询)序列的输出状态
命令语法	<pre><channel>:SEquence <switch> <channel>:= {C1, C2}. <switch>:= {0, 1} 或者 {ON, OFF}.</pre>
查询语法	<pre><channel>:AWG:SEquence? <channel>:= {C1, C2}.</pre>
示例	<p>开启通道 1 的序列输出</p> <pre><i>:C1:SEquence ON</i></pre> <p>读取通道 1 的序列输出状态</p> <pre><i>:C1:SEquence?</i></pre> <p>返回值:</p> <pre><i>"ON"</i></pre>

3.32.3 <channel>:SEQuence:STATe <state>

描述	该命令用于设置(查询)序列的运行状态
命令语法	<pre><channel>:SEQuence:STATe<state> <channel>:= {C1, C2}. <state>:= {RUN, STOP}.</pre>
查询语法	<pre><channel>:SEQuence:STATe? <channel>:= {C1, C2}.</pre>
示例	<p>开设置通道 1 序列的运行状态为运行</p> <pre><i>:C1:SEQuence:STATe RUN</i></pre> <p>读取通道 1 序列的运行状态</p> <pre><i>:C1:SEQuence:STATe?</i></pre> <p>返回值:</p> <pre><i>RUN\n</i></pre>

3.32.4 <channel>:SEQuence:BURSt <count>

描述	该命令用于设置(查询)序列在单次输出模式时，每次输出波形的周期数
命令语法	<pre><channel>:SEQuence:BURSt <count> <channel>:= {C1, C2}. <count>:= {1 到 65535}.</pre>
查询语法	<pre><channel>:SEQuence:BURSt? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 序列在单次输出模式时，每次输出 2 个周期</p> <pre><i>:C1:SEQuence:BURSt 2</i></pre> <p>查询通道 1 序列在单次输出模式时，每次输出波形的周期数</p> <pre><i>:C1:SEQuence:BURSt?</i></pre> <p>返回值:</p> <pre><i>"2"</i></pre>

3.32.5 <channel>:SEQuence:RMODe <mode>

描述	该命令用于设置(查询)序列的运行模式
命令语法	<pre><channel>:SEQuence:RMODe <mode> <channel>:= {C1, C2}. <mode>:= {CONT, TCON, BURS, STEP, ADV}.</pre>
查询语法	<pre><channel>:SEQuence:RMODe? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 序列的运行模式为连续运行</p> <pre><i>:C1:SEQuence:RMODe CONT</i></pre> <p>查询通道 1 序列的运行模式</p> <pre><i>:C1:SEQuence:RMODe?</i></pre> <p>返回值:</p> <pre><i>"CONT"</i></pre>

3.32.6 <channel>:TRIGger[:SEQuence]:SOURce

描述	该命令用于设置(查询)序列运行时的触发方式
命令语法	<pre><channel>:TRIGger:SOURce <src> <channel>:= {C1, C2}. <src>:= {MAN, TIME, EXT}.</pre>
查询语法	<pre><channel>:TRIGger:SOURce? <channel>:= {C1, C2}.</pre>
示例	<p>设置通道 1 序列的触发方式为外部触发</p> <pre><i>:C1:TRIGger:SOURce EXT</i></pre> <p>查询通道 1 序列的触发方式</p> <pre><i>:C1:TRIGger:SOURce?</i></pre> <p>返回值:</p> <pre><i>"EXT"</i></pre>

3.32.7 <channel>:TRIGger[:SEQuence][:IMMediate]

描述	该命令立即触发一次序列输出
命令语法	<channel>:TRIGger <channel>:= {C1, C2}.
示例	设置立即触发通道 1 的序列输出 <i>C1:TRIGger</i>

3.32.8 <channel>:TRIGger:TIMer

描述	该命令用于设置(查询)序列定时器触发的时间间隔
命令语法	<channel>:TRIGger:TIMer <time> <channel>:= {C1, C2}.
查询语法	<channel>:TRIGger:TIMer? <channel>:= {C1, C2}.
示例	设置通道 1 序列的定时器触发的时间间隔为 1ms <i>:C1:TRIGger:TIMer 0.001</i> 查询通道 1 序列定时器触发的时间间隔 <i>:C1:TRIGger:TIMer?</i> 返回值: <i>"0.001"</i>

3.32.9 <channel>:TRIGger[:SEQuence]:SLOPe

描述	该命令用于设置(查询)序列外部触发的触发沿
命令语法	<channel>:TRIGger:SLOPe <slope> <channel>:= {C1, C2}. <slope>:= {RISe, FALL}.
查询语法	<channel>:TRIGger:SLOPe? <channel>:= {C1, C2}.
示例	设置通道 1 序列外部触发的触发极性为上升沿 <i>:C1:TRIGger:SLOPe RISe</i> 查询通道 1 序列外部触发的触发极性

```
:C1:TRIGger:SLOPe?
```

```
返回值:
```

```
"RISe"
```

3.32.10 <channel>:SEquence:COUNT

描述	该命令用于设置(查询)序列的总段数
命令语法	<channel>:SEquence:COUNT <count> <channel>:= {C1, C2}.
查询语法	<channel>:SEquence:COUNT? <channel>:= {C1, C2}.
示例	设置通道 1 为由 10 个段组成的序列波 <i>:C1:SEquence:COUNT 10</i> 查询通道 1 序列波形总共有多少个段 <i>:C1:SEquence:COUNT?</i> 返回值: <i>"10"</i>

3.32.11 <channel>:SEquence:DEFAult

描述	该命令用于设置(查询)序列波形参数为默认值
命令语法	<channel>:SEquence:DEFAult <channel>:= {C1, C2}.
示例	设置通道 1 的序列波形参数为默认设置 <i>:C1:SEquence:DEFAult</i>

3.32.12 <channel>:SEquence:NEW

描述	该命令用于新建一个序列波形
命令语法	<channel>:SEquence:NEW <段数> <channel>:= {C1, C2}. <段数>:= 需要新建的段数。

示例	通道 1 的序列新建 10 段波形 <i>C1:SEquence:NEW 10</i>
-----------	--

3.32.13 <channel>:SEquence:SEGMent<x>:WAVeform

描述	该命令用于通过波形名字设置(查询)序列某一段的波形
命令语法	格式 1: <channel>:SEquence:SEGMent<x>:WAVeform <name> 格式 2: <channel>:SEquence:SEGMent<x>:WAVeform <path> <channel>:= {C1, C2}. <x>:= 段编号 <name>:= 波形名字。可用波形名见 3.33.5 节 <path>:= 波形路径, 可以是 Local、U 盘或网络存储盘, 包括波形名称, 需要双引号
查询语法	<channel>:SEquence:SEGMent<x>:WAVeform? <channel>:= {C1, C2}. <x>:= 段编号
示例	设置通道 1 序列第 1 段的波形为 Local 下的 wave1.bin <i>C1:SEquence:SEGMent1:WAVeform "Local/wave1.bin"</i> 设置通道 1 序列第三段的波形为 stairup <i>C1:SEquence:SEGMent3:WAVeform stairup</i> 查询通道 1 序列的第三段波形 (返回波形名字) <i>C1:SEquence:SEGMent3:WAVeform?</i> 返回值: <i>"stairup"</i>

3.32.14 <channel>:SEquence:SEGMent<x>:REPeat:COUNT

描述	该命令用于设置(查询)序列某一段波形的重复次数
命令语法	<channel>:SEquence:SEGMent<x>:REPeat:COUNT <count> <channel>:= {C1, C2}. <x>:= 段编号 <count>:= 最大值与本段波形长度, 以及其他段波形的总长度相关。
查询语法	<channel>:SEquence:SEGMent<x>:REPeat:COUNT?

	<p><channel>:= {C1, C2}.</p> <p><x>:= 段编号</p>
示例	<p>设置通道 1 序列的第三段波形重复 2 次.</p> <p><i>:C1:SEquence:SEGMent3:REPeat:COUNT 2</i></p> <p>查询通道 1 序列的第三段波形重复次数</p> <p><i>:C1:SEquence:SEGMent3:REPeat:COUNT?</i></p> <p>返回值:</p> <p><i>"2"</i></p>

3.32.15 <channel>:SEquence:SEGMent<x>:TSOUrce

描述	该命令用于高级运行模式下设置(查询)序列某一段波形的触发方式
命令语法	<p><channel>:SEquence:SEGMent<x>:TSOUrce <type></p> <p><channel>:= {C1, C2}.</p> <p><x>:=段编号</p> <p>< type >:= {NONE,MAN,TIMe,EXT}</p>
查询语法	<p><channel>:SEquence:SEGMent<x>:TSOUrce?</p> <p><channel>:= {C1, C2}.</p> <p><x>:= 段编号</p>
示例	<p>设置通道 1 序列第三段波形的触发方式为手动</p> <p><i>:C1:SEquence:SEGMent3:TSOUrce MAN</i></p> <p>查询通道 1 序列第三段波形的触发方式</p> <p><i>:C1:SEquence:SEGMent3:TSOUrce?</i></p> <p>返回值:</p> <p><i>MAN\rn</i></p>

3.32.16 <channel>:SEquence:SEGMent<x>:AMPlitude

描述	该命令用于设置(查询)序列某一段波形的幅度
命令语法	<p><channel>:SEquence:SEGMent<x>:AMPlitude <amp></p> <p><channel>:= {C1, C2}.</p> <p><x>:=段编号</p> <p><amp>:= {0 到 24Vpp}</p>

查询语法	<code><channel>:SEquence:SEGMent<x>:AMPlitude?</code> <code><channel>:= {C1, C2}.</code> <code><x>:= 段编号</code>
示例	设置通道 1 序列第三段波形的幅度为 2Vpp <code>:C1:SEquence:SEGMent3:AMPlitude 2</code> 查询通道 1 序列第三段波形的幅度 <code>:C1:SEquence:SEGMent3:AMPlitude?</code> 返回值: <code>"2"</code>

3.32.17 <channel>:SEquence:SEGMent<x>:OFFset

描述	该命令用于设置(查询)序列某一段波形的偏置
命令语法	<code><channel>:SEquence:SEGMent<x>:OFFset <offset></code> <code><channel>:= {C1, C2}.</code> <code><x>:= 段编号</code> <code><offset>:= {-12V 到 12V}</code>
查询语法	<code><channel>:SEquence:SEGMent<x>:OFFset?</code> <code><channel>:={C1, C2}.</code> <code><x>:= 段编号</code>
示例	设置通道 1 序列第三段波形的偏置为 2Vdc <code>:C1:SEquence:SEGMent3:OFFset 2</code> 查询通道 1 序列第三段波形的偏置 <code>:C1:SEquence:SEGMent3:OFFset?</code> 返回值: <code>"2"</code>

3.32.18 <channel>:SEquence:SEGMent<x>:VOLTage:HIGH

描述	该命令用于设置(查询)序列某一段波形的高电平值
命令语法	<code><channel>:SEquence:SEGMent<x>:VOLTage:HIGH <highLevel></code> <code><channel>:= {C1, C2}.</code> <code><x>:= 段编号</code> <code><highLevel>:= {lowLevel 到 12V}</code>

查询语法	<pre><channel>:SEquence:SEGMent<x>:VOLTage:HIGH? <channel>:= {C1, C2}. <x>:= 段编号</pre>
示例	<pre>设置通道 1 序列第三段波形的高电平为 10V :C1:SEquence:SEGMent3:VOLTage:HIGH 10 查询通道 1 序列第三段波形的高电平值 :C1:SEquence:SEGMent3:VOLTage:HIGH? 返回值: "10"</pre>

3.32.19 <channel>:SEquence:SEGMent<x>:VOLTage: LOW

描述	该命令用于设置(查询)序列某一段波形的低电平值
命令语法	<pre><channel>:SEquence:SEGMent<x>:VOLTage:LOW <lowLevel> <channel>:= {C1, C2}. <x>:= 段编号 <lowLevel>:= {-12V 到 highLevel}</pre>
查询语法	<pre><channel>:SEquence:SEGMent<x>:VOLTage:LOW? <channel>:= {C1, C2}. <x>:= 段编号</pre>
示例	<pre>设置通道 1 序列第三段波形的低电平为 10V :C1:SEquence:SEGMent3:VOLTage:LOW 10 查询通道 1 序列第三段波形的低电平值 :C1:SEquence:SEGMent3:VOLTage:LOW? 返回值: "10"</pre>

3.32.20 <channel>:SEquence:SEGMent<x>:LENGth

描述	该命令用于设置(查询)序列某一段波形的长度
命令语法	<pre><channel>:SEquence:SEGMent<x>:LENGth <len> <channel>:= {C1, C2}. <x>:= 段编号 <len>:= 会自动截断设置到 16 的整数倍</pre>

查询语法	<pre><channel>:SEquence:SEGMent<x>:LENGth? <channel>:= {C1, C2}. <x>:= 段编号</pre>
示例	<p>设置通道 1 序列第三段波形的长度为 16384</p> <pre><i>:C1:SEquence:SEGMent3:LENGth 16384</i></pre> <p>查询通道 1 序列第三段波形的长度</p> <pre><i>:C1:SEquence:SEGMent3:LENGth?</i></pre> <p>返回值:</p> <pre><i>"16384"</i></pre>

3.32.21 <channel>:SEquence:SEGMent<x>:DELEte

描述	该命令用于删除序列波形中的某一段
命令语法	<pre><channel>:SEquence:SEGMent<x>:DELEte <channel>:= {C1, C2}. <x>:= 段编号</pre>
示例	<p>删除通道 1 序列中的第三段波形</p> <pre><i>:C1:SEquence:SEGMent3:DELEte</i></pre>

3.32.22 <channel>:SEquence:SEGMent<x>:INSErt

描述	该命令用于在序列波形中的某一段之后插入一个新的段
命令语法	<pre><channel>:SEquence:SEGMent<x>:INSErt <channel>:= {C1, C2}. <x>:= 段编号</pre>
示例	<p>在通道 1 序列的第三段之后插入一个新的段</p> <pre><i>:C1:SEquence:SEGMent3:INSErt</i></pre>

3.32.23 <channel>:SEquence:SEGMent<x>:MARKer:SWITCh

描述	该命令用于设置或查询序列波形中的某一段的标记开关
命令语法	<pre><channel>:SEquence:SEGMent<x>:MARKer:SWITCh <state></pre>

	<pre><channel>:= {C1, C2}. <x>:= 段编号 <state>:={ON,OFF}</pre>
示例	<p>设置通道 1 序列的第 3 段标记开关为关 <i>C1:SEquence:SEGMent3:MARKer:SWITCh OFF</i></p> <p>查询通道 1 序列的第 3 段的标记开关状态: <i>C1:SEquence:SEGMent3:MARKer:SWITCh?</i></p> <p>返回值: <i>OFF\n</i></p>

3.32.24 <channel>:SEquence:SEGMent<x>:MARKer:POS

描述	该命令用于设置或查询序列波形中的某一段的标记点位置
命令语法	<pre><channel>:SEquence:SEGMent<x>:MARKer:POS <pos> <channel>:= {C1, C2}. <x>:= 段编号 <pos>:={0~最大波形点数}</pre>
示例	<p>设置通道 1 序列的第 3 段标记点为 300 <i>C1:SEquence:SEGMent3:MARKer:POS 300</i></p> <p>查询通道 1 序列的第 3 段的标记点位置: <i>C1:SEquence:SEGMent3:MARKer:POS?</i></p> <p>返回值: <i>300\n</i></p>

3.32.25 <channel>:SEquence: OUTList

描述	该命令用于设置序列高级运行模式下波形的输出顺序
命令语法	<pre><channel>:SEquence: OUTList <list> <channel>:= {C1, C2}. <list>:= 输出段的序号, 以逗号分隔</pre>
示例	<p>设置通道 1 序列按 2,1,3,4 的顺序循环输出列表中的段 <i>:C1:SEquence:OUTList "2,1,3,4"</i></p>

3.32.26 <channel>:SEQuence:SCALe

描述	该命令用于设置(查询)序列的输出幅度
命令语法	<channel>:SEQuence:SCALe <scale> <channel>:= {C1, C2}. <scale>:= {0.01 到 1}
查询语法	<channel>:SEQuence:SCALe? <channel>:= {C1, C2}.
示例	设置通道 1 序列的输出幅度为 80% <i>:C1:SEQuence:SCALe 0.8</i> 查询通道 1 序列的输出幅度 <i>:C1:SEQuence:SCALe?</i> 返回值: <i>"0.8"</i>

3.32.27 <channel>:SEQuence:INCReasing

描述	该命令用于设置(查询)序列的插值方式
命令语法	<channel>:SEQuence:INCReasing <mode> <channel>:= {C1, C2}. <mode>:= {INT, ZERo, HLAS, DUPL}
查询语法	<channel>:SEQuence:INCReasing? <channel>:= {C1, C2}.
示例	设置通道 1 序列的插值方式为线性插值 <i>:C1:SEQuence:INCReasing INT</i> 查询通道 1 序列的插值方式 <i>:C1:SEQuence:INCReasing?</i> 返回值: <i>"INT"</i>

3.32.28 <channel>:SEQuence:DECReasing

描述	该命令用于设置(查询)序列的抽值方式
命令语法	<channel>:SEQuence:DECReasing <mode> <channel>:= {C1, C2}.

	<mode>:= {DECI, CTAI, CHEA}
查询语法	<channel>:SEquence:DECRaising? <channel>:= {C1, C2}.
示例	设置通道 1 序列的抽值方式为线性抽值 : <i>C1:SEquence:DECRaising DECI</i> 查询通道 1 序列的抽值方式 : <i>C1:SEquence:DECRaising?</i> 返回值: <i>"DECI"</i>

3.32.29 <channel>:SEquence:RECall

描述	该命令用于从文件加载序列波形
命令语法	<channel>:SEquence:RECall <path> <channel>:= {C1, C2}. <path>:=波形的完整路径
示例	从文件 sequence. awg 加载序列波形 : <i>C1:SEquence:RECall "Local/sequence.awg"</i> : <i>C1:SEquence:RECall "U-disk0/sequence.awg"</i> : <i>C1:SEquence:RECall "net_storage/sequence.awg"</i>
注	具体路径参照文件管理器中路径

3.32.30 <channel>:SEquence:SAVe

描述	该命令用于保存序列波形到文件
命令语法	<channel>:SEquence:SAVe<path> <channel>:= {C1, C2}. <path>:= 保存波形的完整路径
示例	保存序列波形到 sequence. awg : <i>C1:SEquence:SAVe "Local/sequence.awg"</i> : <i>C1:SEquence:SAVe "U-disk0/sequence.awg"</i> : <i>C1:SEquence:SAVe "net_storage/sequence.awg"</i>
注	具体路径参照文件管理器中路径

3.33 数字通道命令

3.33.1 DIG:SRATe

描述	该命令用于设置(查询)数字通道的比特率
命令语法	DIG:SRATe <value> <value>:= {0 到 1000000000}
查询语法	DIG:SRATe?
示例	设置数字通道的比特率为 100M : <i>DIG:SRATe 100000000</i> 查询数字通道的比特率 : <i>DIG:SRATe?</i> 返回值: : <i>100000000</i>

3.33.2 DIG:PERiod

描述	该命令用于设置(查询)数字通道的周期
命令语法	DIG:PERiod <value> <value>:= {1ns 到 1Ks}
查询语法	DIG:PERiod?
示例	设置数字通道的周期为 10ns : <i>DIG:PERiod 0.00000001</i> 查询数字通道的周期 : <i>DIG:PERiod?</i> 返回值: : <i>0.00000001</i>

3.33.3 DIG:CHANnel<x>:STATe

描述	该命令用于设置(查询)数字通道某一通道的状态
命令语法	DIG:CHANnel<x>:STATe <value> <X>:= {1 到 16} <value>:= {0 或者 1} 或= {OFF, ON}

查询语法	DIG:CHANnel<index>:STATe?
示例	<p>设置通道 2 为开启状态</p> <p><i>:DIG:CHANnel2:STATe 1</i></p> <p>查询数字通道 2 的状态</p> <p><i>:DIG:CHANnel2:STATe?</i></p> <p>返回值:</p> <p><i>"1"</i></p>

3.33.4 DIG:OUTPut

描述	该命令用于设置(查询)数字通道的输出状态
命令语法	<p>DIG:OUTPut <value></p> <p><value>:= {0 或者 1}或={OFF, ON}</p>
查询语法	DIG:OUTPut?
示例	<p>开启数字通道输出</p> <p><i>:DIG:OUTPut 1</i></p> <p>查询数字通道的输出状态</p> <p><i>:DIG:OUTPut?</i></p> <p>返回值:</p> <p><i>"1"</i></p>

3.33.5 DIG:WAVeform

描述	该命令用于通过波形名字设置(查询)数字通道的数据源
命令语法	<p>DIG:WAVeform <name></p> <p><name>:= 内建波形的名字, 或波形的完整路径</p>
查询语法	DIG:WAVeform?
示例	<p>设置数字通道的数据源为 sine</p> <p><i>:DIG:WAVeform sine</i></p> <p>查询数字通道数据源</p> <p><i>:DIG:WAVeform?</i></p> <p>返回值:</p> <p><i>"sine"</i></p>

备注：自定义编辑波形见 3.31

sine	Noise	erfc	erfcinv
stairup	stairdn	erfinv	laguerre
stairud	ppulse	legend	versiera
npulse	trapezia	weibull	lognormal
upramp	dnramp	laplace	maxwell
exp_fall	exp_rise	rayleigh	cauchy
logfall	logrise	cosh	cosint
sqrt	root3	coth	csch
x^2	x^3	sech	sinh
sinc	gussian	sinint	tanh
dlorentz	haversine	acosh	asech
lorentz	gauspuls	asinh	atanh
gmonopuls	Tripuls	acsch	acoth
cardiac	Quake	bartlett	bohmanwin
chirp	twotone	chebwin	flattopwin
snr	digit_clock	parzenwin	taylorwin
digit_counter	digit_zero	tukeywin	square_duty01
hamming	hanning	square_duty02	square_duty04
kaiser	blackman	square_duty06	square_duty08
gausswin	Triang	square_duty10	square_duty12
blackmanharris	barthannwin	quare_duty14	square_duty16
tan	Cot	square_duty18	square_duty20
sec	Csc	square_duty22	square_duty24
asin	Acos	square_duty26	square_duty28
atan	Acot	square_duty30	square_duty32
square	sinetra	square_duty34	square_duty36
sinever	ampalt	square_duty38	square_duty40
attalt	roundhalf	square_duty42	square_duty44
roundspm	blaseiwave	square_duty46	square_duty48
dampedosc	swingosc	square_duty50	square_duty52
discharge	pahcur	square_duty54	square_duty56
combin	Scr	square_duty58	square_duty60
butterworth	chebyshev1	square_duty62	square_duty64

chebyshev2	Tv	square_duty66	square_duty68
voice	Surge	square_duty70	square_duty72
NA	Ripple	square_duty74	square_duty76
gamma	stepresp	square_duty78	square_duty80
bandlimited	Cpulse	square_duty82	square_duty84
cwpulse	gatevibr	square_duty86	square_duty88
lfpulse	mcnoise	square_duty90	square_duty92
am	Fm	square_duty94	square_duty96
pfm	Pm	square_duty98	square_duty99
pwm	Eog	demo1_375pts	demo1_16kpts
eeg	Emg	demo2_3kpts	demo2_16kpts
pulseilogram	resspeed	sine_harmonic2	sine_harmonic3
ecg1	ecg2	sine_harmonic4	sine_harmonic5
ecg3	ecg4	sine_harmonic6	sine_harmonic7
ecg5	ecg6	sine_harmonic8	sine_harmonic9
ecg7	ecg8	sine_harmonic10	sine_harmonic11
ecg9	ecg10	sine_harmonic12	sine_harmonic13
ecg11	ecg12	sine_harmonic14	sine_harmonic15
ecg13	ecg14	sine_harmonic16	digit_one
ecg15	Lfpulse		
tens1	tens2		
tens3	Airy		
besselj	bessely		
dirichlet	Erf		

3.33.6 DIG:RMODE <mode>

描述	该命令用于设置(查询)数字通道的运行模式
命令语法	DIG:RMODE <mode> <mode>:= {CONT,TCON,BURS}
查询语法	DIG:RMODE?
示例	设置数字通道的运行模式为 TCON :DIG:RMODE TCON

查询数字通道的运行模式

:DIG:RMODe?

返回值:

TCONIn

3.33.7 DIG:TRIGger:SOURce <type>

描述	该命令用于设置(查询)数字通道的触发源
命令语法	DIG:TRIGger:SOURce <type> <type>:= {MAN,TIME,EXT}
查询语法	DIG:TRIGger:SOURce?
示例	设置数字通道的触发源为 TIME <i>:DIG:TRIGger:SOURce TIME</i> 查询数字通道的触发源 <i>:DIG:TRIGger:SOURce?</i> 返回值: <i>TIMEIn</i>

3.33.8 DIG:TRIGger:SLOPe <type>

描述	该命令用于设置(查询)数字通道外部触发源下的极性
命令语法	DIG:TRIGger:SLOPe <type> <type>:= {RISe,FALL}
查询语法	DIG:TRIGger:SLOPe?
示例	设置数字通道外部触发源的极性为 FALL <i>:DIG:TRIGger:SLOPe FALL</i> 查询数字通道外部触发源的极性 <i>:DIG:TRIGger:SLOPe?</i> 返回值: <i>FALLIn</i>

3.33.9 DIG:TRIGger:TIMer <value>

描述	该命令用于设置(查询)数字通道定时器触发源下的定时时间
命令语法	DIG:TRIGger:TIMer <value> <value>:= {定时时间}
查询语法	DIG:TRIGger:TIMer?
示例	设置数字通道定时器触发源的定时时间为 20ms <pre><i>:DIG:TRIGger:TIMer 0.02</i></pre> 查询数字通道定时器触发源的定时时间 <pre><i>:DIG:TRIGger:TIMer?</i></pre> 返回值: <pre><i>0.02\n</i></pre>

3.33.10 DIG:BURSt:COUNT <value>

描述	该命令用于设置(查询)数字通道 Burst 运行模式下的脉冲串数
命令语法	DIG:BURSt:COUNT <value> <value>:= {脉冲串数}
查询语法	DIG:BURSt:COUNT?
示例	设置数字通道 Burst 运行模式下的脉冲串数为 20 <pre><i>:DIG:BURSt:COUNT 20</i></pre> 查询数字通道 Burst 运行模式下的脉冲串数 <pre><i>:DIG:BURSt:COUNT?</i></pre> 返回值: <pre><i>20\n</i></pre>

3.33.11 DIG:HOLD:VALUe <value>

描述	该命令用于设置(查询)数字通道的空闲输出值
命令语法	DIG:HOLD:VALUe <value> <value>:= { 0~65535}, 空闲输出值
查询语法	DIG:HOLD:VALUe?
示例	设置数字通道的空闲输出值为 300

```
:DIG:HOLD:VALUe 300
```

查询数字通道的空闲输出值

```
:DIG:HOLD:VALUe?
```

返回值:

```
300\n
```

3.33.12 DIG:TRIGger

描述	该命令用于设置数字通道按键触发源时的手动触发一次
命令语法	DIG:TRIGger
示例	设置数字通道手动触发一次 <i>:DIG:TRIGger</i>

3.34 跳频命令

3.34.1 <channel>:FHOP:SWITCh

描述	该命令用于设置跳频功能开关状态
命令语法	<channel>:FHOP:SWITCh <switch> <channel>:={C1, C2} <switch>:={ON, OFF}
查询语法	<channel>:FHOP:SWITCh?
示例	通道 1 开启跳频 <i>C1:FHOP:SWITCh ON</i> 查询通道 1 跳频开关状态 <i>C1:FHOP:SWITCh?</i> 返回值: <i>"ON"</i>

3.34.2 <channel>:FHOP: STATE

描述	该命令用于设置跳频输出状态
命令语法	<channel>:FHOP:STATE <state> <channel>:={C1, C2} <state>:={ON, OFF}
查询语法	<channel>:FHOP:STATE?
示例	通道 1 开启跳频输出 <i>C1:FHOP:STATE ON</i> 查询通道 1 跳频输出状态 <i>C1:FHOP:STATE?</i> 返回值: <i>"ON"</i>

3.34.3 <channel>:FHOP:TYPE

描述	该命令用于设置跳频模式
命令语法	<channel>:FHOP:TYPE <type> <channel>:={C1, C2} <type>:={"MANUAL", "RHOP", "RLIST"}
查询语法	<channel>:FHOP:TYPE?
示例	设置通道 1 跳频为列表模式 <i>C1:FHOP:TYPE MANUAL</i> 查询通道 1 跳频模式 <i>C1:FHOP:TYPE?</i> 返回值: <i>"MANUAL"</i>

3.34.4 <channel>:FHOP:TIME

描述	该命令用于设置跳频在每个频点上的持续时间
命令语法	<channel>:FHOP:TIME <time> <channel>:={C1, C2} <time>:={0 到 1000 的浮点数}
查询语法	<channel>:FHOP:TIME?
示例	设置通道 1 跳频在每个频点上的持续时间为 1ms <i>C1:FHOP:TIME 0.001</i> 查询通道 1 跳频在每个频点上的持续时间 <i>C1:FHOP:TIME?</i> 返回值: <i>"0.001"</i>

3.34.5 <channel>:FHOP:SFREquency

描述	该命令用于设置随机跳频的起始频率（最小频率）
命令语法	<channel>:FHOP:SFREquency <freq> <channel>:={C1, C2} <freq>:={不超过设备带宽的浮点数}

查询语法	<channel>:FHOP:SFREquency?
示例	<p>设置通道 1 随机跳频的起始频率（最小频率）为 1MHZ</p> <p><i>C1:FHOP:SFREquency 1000000</i></p> <p>查询通道 1 随机跳频的起始频率（最小频率）</p> <p><i>C1:FHOP:SFREquency?</i></p> <p>返回值:</p> <p><i>"1000000"</i></p>

3.34.6 <channel>:FHOP:EFREquency

描述	该命令用于设置随机跳频的结束频率（最大频率）
命令语法	<p><channel>:FHOP:EFREquency <freq></p> <p><channel>:={C1, C2}</p> <p><freq>:={不超过设备带宽的浮点数}</p>
查询语法	<channel>:FHOP:EFREquency?
示例	<p>设置通道 1 随机跳频的结束频率（最大频率）为 100MHZ</p> <p><i>C1:FHOP:EFREquency 100000000</i></p> <p>查询通道 1 随机跳频的结束频率（最大频率）</p> <p><i>C1:FHOP:EFREquency?</i></p> <p>返回值:</p> <p><i>"100000000"</i></p>

3.34.7 <channel>:FHOP:FSTep

描述	该命令用于设置随机跳频的频率步进
命令语法	<p><channel>:FHOP:FSTep <freq></p> <p><channel>:={C1, C2}</p> <p><freq>:={不超过设备带宽的浮点数}</p>
查询语法	<channel>:FHOP:FSTep?
示例	<p>设置通道 1 随机跳频的频率步进为 10MHZ</p> <p><i>C1:FHOP:FSTep 10000000</i></p> <p>查询通道 1 随机跳频的频率步进</p> <p><i>C1:FHOP:FSTep?</i></p>

返回值：
“10000000”

3.34.8 <channel>:FHOP:RPATtern

描述	该命令用于设置随机跳频的伪随机码型
命令语法	<channel>:FHOP:RPATtern <pattern> <channel>:={C1, C2} <pattern>:= {3 到 32 的整数}
查询语法	<channel>:FHOP:RPATtern?
示例	设置通道 1 随机跳频的伪随机码型为 PRBS-7 <i>C1:FHOP:RPATtern 7</i> 查询通道 1 随机跳频的伪随机码型 <i>C1:FHOP:RPATtern?</i> 返回值： “7”

3.34.9 <channel>:FHOP:RLPAttern

描述	该命令用于设置跳频随机列表模式的伪随机码型
命令语法	<channel>:FHOP:RLPAttern <pattern> <channel>:={C1, C2} <pattern>:= {3 到 32 的整数}
查询语法	<channel>:FHOP:RLPAttern?
示例	设置通道 1 跳频列表模式的伪随机码型为 PRBS-7 <i>C1:FHOP:RLPAttern 7</i> 查询通道 1 跳频列表模式的伪随机码型 <i>C1:FHOP:RLPAttern?</i> 返回值： “7”

3.34.10 <channel>:FHOP:ALState

描述	该命令用于设置随机跳频频率过滤表的使能状态
命令语法	<channel>:FHOP:ALState <state> <channel>:={C1, C2} <state>:={ON, OFF}
查询语法	<channel>:FHOP:ALState?
示例	使能通道 1 随机跳频的频率过滤表 <i>C1:FHOP:ALState ON</i> 查询通道 1 随机跳频频率过滤表的使能状态 <i>C1:FHOP:ALState?</i> 返回值: <i>"ON"</i>

3.34.11 <channel>:FHOP:AFLst

描述	该命令用于在频率表指定位置插入一项
命令语法	<channel>:FHOP:AFLst <index>,<freq> <channel>:={C1, C2} <index>:={1 到 4096 的整数} <freq>:={不超过设备带宽的浮点数}
示例	通道 1 在频率表第三项的位置插入一项, 频率为 1KHz <i>C1:FHOP:AFLst 3,1000</i>

3.34.12 <channel>:FHOP:DFLst

描述	该命令用于删除频率表指定位置的项
命令语法	<channel>:FHOP:DFLst <index> <channel>:={C1, C2} <index>:={1 到 4096 的整数}
示例	删除通道 1 频率表的第三项 <i>C1:FHOP:DFLst 3</i>

3.34.13 <channel>:FHOP:CFLIst

描述	该命令用于清空频率表（恢复为默认值）
命令语法	<channel>:FHOP:CFLIst <channel>:={C1, C2}
示例	清空通道 1 的频率表（恢复为默认值） <i>C1:FHOP:CFLIst</i>

3.34.14 <channel>:FHOP:MFLIst

描述	该命令用于将频率表指定位置的项修改为设置值
命令语法	<channel>:FHOP:MFLIst <index>,<freq> <channel>:={C1, C2} <index>:= {1 到 4096 的整数} <freq>:= {不超过设备带宽的浮点数}
示例	将通道 1 频率表第三项的频率修改为 2KHz <i>C1:FHOP:MFLIst 3,2000</i>

3.34.15 <channel>:FHOP:AOLIst

描述	该命令用于在频率跳转表指定位置插入一项
命令语法	<channel>:FHOP:AOLIst <index>,<freq_num> <channel>:={C1, C2} <index>:= {1 到 4096 的整数} <freq_num>:= {1 到 4096 的整数, 不超过频率表的长度}
示例	通道 1 在频率跳转表第三项的位置插入一项, 频率编号为 4 <i>C1:FHOP:AOLIst 3,4</i>

3.34.16 <channel>:FHOP:DOLIst

描述	该命令用于删除频率跳转表指定位置的项
命令语法	<channel>:FHOP:DOLIst <index> <channel>:={C1, C2}

	<index>:= {1 到 4096 的整数}
示例	删除通道 1 频率跳转表的第三项 <i>C1:FHOP:DOLst 3</i>

3.34.17 <channel>:FHOP:COList

描述	该命令用于清空频率跳转表（恢复为默认值）
命令语法	<channel>:FHOP:COList <channel>:={C1, C2}
示例	清空通道 1 的频率跳转表（恢复为默认值） <i>C1:FHOP:COList</i>

3.34.18 <channel>:FHOP:MOList

描述	该命令用于将频率跳转表指定位置的项修改为设置值
命令语法	<channel>:FHOP:MOList <index>,<freq_num> <channel>:={C1, C2} <index>:= {1 到 4096 的整数} <freq_num>:= {1 到 4096 的整数, 不超过频率表的长度}
示例	将通道 1 频率跳转表第三项的值修改为 8 <i>C1:FHOP:MOList 3,8</i>

3.34.19 <channel>:FHOP:AAList

描述	该命令用于在频率过滤表末尾插入一项
命令语法	<channel>:FHOP:AAList <start_freq>,<end_freq> <channel>:={C1, C2} <start_freq>:= {不超过设备带宽的浮点数,小于 end_freq} <end_freq>:= {不超过设备带宽的浮点数,大于 start_freq}
示例	通道 1 在频率过滤表末尾插入一项, 起始频率 1KHz, 结束频率 5KHz (过滤掉 1KHz~5KHz 之间的频率) <i>C1:FHOP:AAList 1000,5000</i>

3.34.20 <channel>:FHOP:MAList

描述	该命令用于修改频率过滤表某一项的参数
命令语法	<pre><channel>:FHOP:MAList <index>,<start_freq>,<end_freq> <channel>:={C1, C2} <index>:=项的序号 <start_freq>:=起始频率 <end_freq>:=终止频率</pre>
示例	修改通道 1 频率过滤表第二项的频率为 200Hz~2kHz <i>C1:FHOP:MAList 2,200,2000</i>

3.34.21 <channel>:FHOP:DAList

描述	该命令用于删除频率过滤表最后一项
命令语法	<pre><channel>:FHOP:DAList <channel>:={C1, C2}</pre>
示例	删除通道 1 频率过滤表最后一项 <i>C1:FHOP:DAList</i>

3.34.22 <channel>:FHOP:CAList

描述	该命令用于清空频率过滤表（恢复为默认值）
命令语法	<pre><channel>:FHOP:CAList <channel>:={C1, C2}</pre>
示例	清空通道 1 的频率过滤表（恢复为默认值） <i>C1:FHOP:CAList</i>

3.34.23 <channel>:FHOP:LFList

描述	该命令用于从文件加载频率表
命令语法	<pre><channel>:FHOP:LFList <file> <channel>:={C1, C2}</pre>

	<file>:=文件名（包含路径）
示例	从文件 freq.hop 加载通道 1 的频率表 <i>C1:FHOP:LFList "freq.hop"</i>

3.34.24 <channel>:FHOP:SFList

描述	该命令用于将当前频率表保存到文件
命令语法	<channel>:FHOP:SFList <file> <channel>:={C1, C2} <file>:=文件名（包含路径）
示例	通道 1 的频率表保存到文件 freq.hop <i>C1:FHOP:SFList "freq.hop"</i>

3.34.25 <channel>:FHOP:LOList

描述	该命令用于从文件加载频率跳转表
命令语法	<channel>:FHOP:LOList <file> <channel>:={C1, C2} <file>:=文件名（包含路径）
示例	从文件 order.hop 加载通道 1 的频率跳转表 <i>C1:FHOP:LOList "order.hop"</i>

3.34.26 <channel>:FHOP:SOList

描述	该命令用于将当前频率跳转表保存到文件
命令语法	<channel>:FHOP:SOList <file> <channel>:={C1, C2} <file>:=文件名（包含路径）
示例	通道 1 的频率跳转表保存到文件 order.hop <i>C1:FHOP:SOList "order.hop"</i>

3.34.27 <channel>:FHOP:LALst

描述	该命令用于从文件加载频率过滤表
命令语法	<channel>:FHOP:LALst <file> <channel>:={C1, C2} <file>:=文件名（包含路径）
示例	从文件 avoid.hop 加载通道 1 的频率过滤表 <i>C1:FHOP:LALst "avoid.hop"</i>

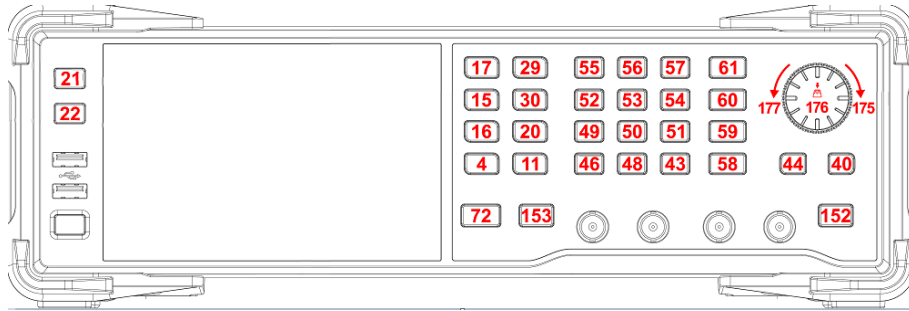
3.34.28 <channel>:FHOP:SALst

描述	该命令用于将当前频率过滤表保存到文件
命令语法	<channel>:FHOP:SALst <file> <channel>:={C1, C2} <file>:=文件名（包含路径）
示例	通道 1 的频率过滤表保存到文件 avoid.hop <i>C1:FHOP:SALst "avoid.hop"</i>

3.35 虚拟键命令

描述	此命令用于模拟前面板的按键的状态
命令语法	VirtualKEY VALUE,<值>,STATE,<状态> <值>:= {下表显示虚拟按键的索引号或者名称}. <状态>:= {0, 1}, “1”是有效的虚拟值，即按下前面板的对应按键；而“0”是无效值。

名称	索引	名称	索引
KB_FUNC1	28	KB_NUMBER_4	52
KB_FUNC2	23	KB_NUMBER_5	53
KB_FUNC3	18	KB_NUMBER_6	54
KB_FUNC4	13	KB_NUMBER_7	55
KB_FUNC5	8	KB_NUMBER_8	56
KB_MOD	15	KB_NUMBER_9	57
KB_SWEEP	16	KB_POINT	46
KB_BURST	17	KB_NEGATIVE	43
KB_WAVES	4	KB_LEFT	44
KB_UTILITY	11	KB_RIGHT	40
KB_NUMBER_0	48	KB_OUTPUT1	153
KB_NUMBER_1	49	KB_OUTPUT2	152
KB_NUMBER_2	50	KB_KNOB_RIGHT	175
KB_NUMBER_3	51	KB_KNOB_LEFT	177
KB_ENTER	58	KB_KNOB_DOWN	176
KB_AWG	29	KB_CHANNEL	72
KB_IQ	30	KB_K	59
KB_DIGITAL	20	KB_M	60
KB_HOME	21	KB_G	61
KB_TOUCH	22		



SDG7000A 上的按键和指示

示例

VKEY VALUE,15,STATE,1

VKEY VALUE,KB_SWEEP,STATE,1

3.36 IP 命令

描述	该参数用于设置或者读取设备的 IP 地址。
命令语法	<p>SYSTem:COMMunicate:LAN:IPADdress <参数 1>.<参数 2>.<参数 3>.<参数 4></p> <p><参数 1>: = {在 1 至 223 之间的整数值}. <参数 2>: = {在 0 至 255 之间的整数值}. <参数 3>: = {在 0 至 255 之间的整数值}. <参数 4>: = {在 0 至 255 之间的整数值}.</p>
查询语法	SYSTem:COMMunicate:LAN:IPADdress?
示例	<p>设置 IP 地址为 10.11.13.203:</p> <p><i>SYST:COMM:LAN:IPAD "10.11.13.203"</i></p> <p>读取 IP 地址:</p> <p><i>SYST:COMM:LAN:IPAD?</i></p> <p>返回值:</p> <p><i>"10.11.13.203"</i></p>

3.37 子网掩码命令

描述	该命令用于设置或者获取设备的子网掩码。
命令语法	<p>SYSTem:COMMunicate:LAN:SMASk <参数 1>.<参数 2>.<参数 3>.<参数 4></p> <p><参数 1>:= {在 0 至 255 之间的整数值}. <参数 2>:= {在 0 至 255 之间的整数值}. <参数 3>:= {在 0 至 255 之间的整数值}. <参数 4>:= {在 0 至 255 之间的整数值}.</p>
查询语法	SYSTem:COMMunicate:LAN:SMASk?
示例	<p>设置子网掩码为 255.0.0.0:</p> <p><i>SYST:COMM:LAN:SMAS "255.0.0.0"</i></p> <p>获取子网掩码:</p> <p><i>SYST:COMM:LAN:SMAS?</i></p> <p>返回值:</p> <p><i>"255.0.0.0"</i></p>

3.38 网关命令

描述	该命令设置并获取设备的网关。
命令语法	<p>SYSTem:COMMunicate:LAN:GATeway <参数 1>.<参数 2>.<参数 3>.<参数 4></p> <p><参数 1>:= {在 0 至 223 之间的整数值}. <参数 2>:= {在 0 至 255 之间的整数值}. <参数 3>:= {在 0 至 255 之间的整数值}. <参数 4>:= {在 0 至 255 之间的整数值}.</p>
查询语法	SYSTem:COMMunicate:LAN:GATeway?
示例	<p>设置网关为 10.11.13.5:</p> <p><i>SYSTem:COMMunicate:LAN:GATeway "10.11.13.1"</i></p> <p>获取网关:</p> <p><i>SYSTem:COMMunicate:LAN:GATeway?</i></p> <p>返回值:</p> <p><i>"10.11.13.1"</i></p>

3.39 采样率命令

描述	设置或者获取任意波模式、采样率和插值方法。采样率和插值方法仅能在逐点输出模式下设置。
命令语法	<p><通道>: SampleRATE MODE, <模式>, VALUE, <采样率>, INTER, <插值></p> <p><通道> := <C1, C2>. <模式>:= {AFG, AWG} <采样率> := 采样率。单位是 Sa/s。 <插值>:= {LINE, HOLD, SINC, SINC27, SINC13}, 其中 LINE 是线性插值, 而 HOLD 是零阶保持。</p>
查询语法	<通道>:SRATE?
示例	<p>获取 CH1 的采样率:</p> <p><i>C1:SRATE?</i></p> <p>返回值:</p> <p><i>C1:SRATE MODE,DDS</i></p>

设置 CH1 为 AFG 模式:

```
C1:SRATE MODE,AFG
```

设置 CH1 的采样率为 1000000Sa/s:

```
C1:SRATE VALUE,1000000
```

3.40 非线性补偿命令

描述	该命令用于设置或查询非线性补偿开关状态。仅当基本波形为正弦波时，该命令有效。
命令语法	<pre><通道>:NonLineComp STATE,<state> <通道> := <C1, C2>. <state > := {ON, OFF}</pre>
查询语法	<pre><通道>:NonLineComp?</pre>
示例	<p>设置通道 1 的非线性补偿开关为开:</p> <pre>C1:NonLineComp STATE,ON</pre> <p>查询通道 1 的非线性补偿开关状态:</p> <pre>C1:NonLineComp?</pre> <p>返回值:</p> <pre>C1:NonLineComp STATE,ON\rn</pre>

3.41 谐波命令

描述	该命令用于设置或者获取谐波参数。仅当基本波形为正弦波时，该命令有效。
命令语法	<pre><通道>:HARMonicHARMSTATE, <状态>, HARMTYPE, <类型>, HARMORDER, <级数>, <单位>, <值>, HARMPHASE, <相位></pre> <p><状态>:= <ON, OFF>.</p> <p><类型>:= <EVEN (偶级数), ODD (奇级数), ALL (所有级数) >.</p> <p><级数>:= {1, 2, ..., M}, 其中 M 是支持的最大级数。</p> <p><单位> := < HARMAMP, HARMDBC>.</p> <p><值>:= 指定谐波的幅值。值的范围取决于机型。当<单位>=</p>

	HARMAMP, 单位是伏特, 峰峰值“Vpp”, 而当<单位>= HARMDBC, 单位是"dBc". <相位>:= {0~360}, 单位是"度".
查询语法	<通道>:HARMonic? <通道>:={C1, C2}.
示例	启用 CH1 的谐波功能: <i>C1:HARMHARMSTATE,ON</i> 设置 CH1 的谐波级数为 2, 幅度为-6dBc: <i>C1:HARMHARMORDER,2,HARMDBC,-6</i> 获取 CH1 的谐波参数: <i>C1:HARM?</i> 返回值: <i>C1:HARM,HARMSTATE,ON,HARMTYPE,EVEN,HARMORDER,2,HARMAMP,2.004748935V,HARMDBC,-6dBc,HARMPHASE,0</i>

3.42 波形合并命令

描述	此命令设置或者获取波形合并。
命令语法	<通道>:CoMBiNe <状态> <通道>:= {C1, C2}. <状态>:= {ON, OFF}.
查询语法	<通道>:CoMBiNe? <通道>:= {C1, C2}.
响应格式	<通道>:CMBN<状态>
示例	打开 CH1 的波形合并: <i>C1:CoMBiNe ON</i> 查询 CH2 的波形合并状态: <i>C2:CMBN?</i> 返回值: <i>C2:CMBN OFF</i>

3.43 模式选择命令

描述	该参数设置或者获取相位模式
命令语法	MODE<参数> <参数>:= {PHASELOCKED (相位锁定), INDEPENDENT (相位独立)}.
查询语法	MODE?
响应格式	MODE<参数>
示例	设置相位模式为相位独立模式: <i>MODE INDEPENDENT</i>

3.44 多设备同步命令

描述	该命令设置两个或多个仪器之间的同步，并实现同相输出
命令语法	CASCADE STATE, ONIOFF, MODE, <MODE>, DELAY, <DELAY> <MODE>:= {MASTER, SLAVE} <DELAY>:= {0-0.000025}, 单位=s, 此参数只能在从机模式时设置

查询语法	CASCADE?
响应格式	从机模式返回值： CASCADE STATE, ON, MODE, SLAVE, DELAY, <DELAY> 主机模式返回值： CASCADE STATE, ON, MODE, MASTER
示例	设置设备为从机模式且延迟为 0.0000001s <i>CASCADE STATE,ON,MODE,SLAVE,DELAY,0.0000001</i> 设置主机设备输出同步设备信号 <i>CASCADE SYNC_PHASE</i>

3.45 IQ 命令

3.45.1 IQ:WAVeinfo?

描述	该命令查询 I/Q 的波形信息。
查询语法	IQ:WAVeinfo?
示例	<p>查询当前 I/Q 波形信息：</p> <p><i>IQ:WAVeinfo?</i></p> <p>返回值：</p> <p><i>WAVE_INFO,SYMBOL_LENGTH,1024,OVER_SAMPLING,4,MODULATION,2ASK,FILTER_TYPE,RootCosine,FILTER_ALPHA,0.35</i></p>

3.45.2 IQ:CENTerfreq

描述	该命令设置 I/Q 调制的中心频率。
命令语法	<p>IQ:CENTerfreq<中心频率><单位></p> <p><中心频率>:= 中心频率。该参数的有效范围请查阅数据手册。</p> <p><单位> := {Hz, kHz, MHz, GHz}. 默认单位是赫兹“Hz”。</p>
查询语法	IQ:CENTerfreq?
响应格式	<中心频率> (用 Hz 表示)
示例	<p>设置中心频率为 1kHz：</p> <p><i>IQ:CENTerfreq 1000Hz</i></p>

3.45.3 IQ:SAMPlerate

描述	该命令设置 I/Q 的采样率
命令语法	<p>IQ:SAMPlerate<采样率><单位></p> <p><采样率> := 采样率。该参数的有效范围请查阅数据手册。</p> <p><单位> := {Hz, kHz, MHz, GHz}. 默认单位是赫兹“Hz”</p>
查询语法	IQ:SAMPlerate?
响应格式	<采样率> (用 Hz 表示)

示例	设置采样率为 100kHz: <i>:IQ:SAMPlerate 100000</i> 或: <i>:IQ:SAMP 100kHz</i>
----	--

3.45.4 IQ:SYMBOLrate

描述	该命令用于设置 I/Q 的符号率
命令语法	<i>IQ:SYMBOLrate <符号率><单位></i> <符号率> := 符号率。该参数的有效范围请查阅数据手册。 <单位> := {S/s, kS/s, MS/s}。默认单位是符号每秒“S/s”
查询语法	<i>IQ:SYMBOLrate?</i>
响应格式	<符号率> (使用 S/s 表示)
示例	设置符号率为 1MS/s: <i>IQ:SYMB 1MS/s</i>

3.45.5 IQ:AMPLitude

描述	该命令设置 I/Q 幅度。
命令语法	<i>IQ:AMPLitude <幅值><单位></i> <幅值> := 幅度。该参数的有效范围请查阅数据手册。 <单位> := {Vrms, mVrms, dBm}。默认单位是伏特，均方根“Vrms”。
查询语法	<i>IQ:AMPLitude?</i>
响应格式	<幅值> (表示为 Vrms.)
示例	设置 IQ 的幅值($\sqrt{I^2+Q^2}$)为 0.2Vrms: <i>IQ:AMPL 0.2</i>

3.45.6 IQ:IQADjustment:GAIN

描述	此命令在保持 IQ 复合状态下调节 I 与 Q 的比例。
命令语法	<i>IQ:IQADjustment:GAIN <增益比><单位></i> <增益比> := I 与 Q 的增益比。

	<单位> := {dB}.
查询语法	IQ:IQADjustment:GAIN?
响应格式	<增益比> (用单位 dB 表示)
示例	设置 I/Q 的增益比例为 0.1dB: <i>IQ:IQADjustment:GAIN 0.1</i>

3.45.7 IQ:IQADjustment:IOFFset

描述	此命令调节 I 通道的偏移量。
命令语法	IQ:IQADjustment:IOFFset <偏置><单位> <偏置> := I 的偏移量 <单位> := {V, mV, uV}。默认单位是伏特“V”。
查询语法	IQ:IQADjustment:IOFFset?
响应格式	<偏置>(表示为 V.)
示例	设置 I 的偏置为 1mV: <i>IQ:IQADjustment:IOFFset 1mV</i>

3.45.8 IQ:IQADjustment:QOFFset

描述	此命令调节 Q 通道的偏移量。
命令语法	IQ:IQADjustment:QOFFset <偏置><单位> <偏置> := Q 的偏移量。 <单位> := {V, mV, uV}。默认单位是伏特，“V”。
查询语法	IQ:IQADjustment:QOFFset?
响应格式	<偏置>(用 V 表示)
示例	设置 Q 的偏置为-1mV: <i>IQ:IQAD:QOFF -0.001</i>

3.45.9 IQ:IQADjustment:QSKew

描述	该命令通过增加或者减少 Q 的相位角来调节 I 和 Q 向量的相位角。
命令语法	IQ:IQADjustment:QSKew <角度>

	<角度> := 角度。单位是度。
查询语法	IQ:IQADjustment:QSKew?
响应格式	<角度>(使用度表示)
示例	设置 Q 角度为 1°: <i>IQ:IQADjustment:QSKew 1.0</i>

3.45.10 IQ:TRIGger:SOURce

描述	该命令设置 I/Q 的触发源。
命令语法	IQ:TRIGger:SOURce <触发源> <触发源>:={INTernal, EXTernal, MANual}, 其中 INTernal 是内部触发, EXTernal 为外部触发, MANual 为手动触发。
查询语法	IQ:TRIGger:SOURce?
响应格式	<触发源>
示例	设置触发源为内触发: <i>IQ:TRIGger:SOURce INTernal</i>

3.45.11 IQ:WAVEload:BUILtin

描述	此命令用于从内建波形列表中选择 I/Q 波形。
命令语法	IQ:WAVEload:BUILtin <波形名称> <波形名称>: = {下表的波形名}.
查询语法	IQ:WAVEload?
响应格式	BUILtin USERstored <波形名称>
示例	设置 I/Q 波形为内建波形的 2ASK: <i>IQ:WAVE:BUIL "2ASK"</i>

2ASK	4ASK	8ASK	BPSK	4PSK
8PSK	DBPSK	4DPSK	8DPSK	8QAM
16QAM	32QAM	64QAM	128QAM	256QAM

3.45.12 IQ:WAVEload:USERstored

描述	此命令用于在用户存储波形中选择 I/Q 波形。
命令语法	IQ:WAVEload:USERstored<路径> <路径> := {来自用户存储（本地，网络存储，U 盘）中的波形路径，包括文件名和后缀}。
查询语法	IQ:WAVEload?
响应格式	BUILtin USERstored <波形名称>
示例 1	设置 I/Q 波形为用户本地存储波形 wave1.arb: <i>IQ:WAVEload:USERstored "wave1.arb"</i> 或: <i>IQ:WAVEload:USERstored "Local/wave1.arb"</i>
示例 2	设置 I/Q 波形为网络存储波形 wave1.arb: <i>IQ:WAVEload:USERstored "net_storage/wave/wave1.arb"</i>
示例 3	设置 I/Q 波形为 U 盘存储波形 wave1.arb: <i>IQ:WAVEload:USERstored "U-disk0/wave/wave1.arb"</i>

备注 1:

- 1) 路径必须使用双引号把英文包括，例如：“net_storage/wave/wave1.arb”；
“U-disk0/wave/wave1.arb”。具体可用路径请参考文件管理器。
- 2) 不指定路径默认为本地路径，不添加后缀名默认为 .wav 后缀。

3.45.13 IQ:FrequencySampling

描述	此命令设置 I/Q 采样率。
命令语法	:IQ: FrequencySampling <sampling> < sampling >:= {1000-1250000000}. 单位为“Hz”。
查询语法	IQ:FrequencySampling? 查询当前采样率 IQ:FrequencySamplingLimit? 查询可设置采样率范围
响应格式	<sampling> MAX, <max_sampling>, MIN, < min_sampling >
示例	设置 I/Q 采样率为：2000000 Hz <i>IQ:FrequencySampling 2000000</i>

3.46 文件管理命令

3.46.1 MMEMory:DELeTe

描述	该命令用于删除指定文件
命令语法	MMEMory:DELeTe <参数> <参数>:=文件路径（包含操作文件名称）
示例	删除路径为“Local/1000pts.bin”的文件： <i>MMEMory:DELeTe “Local/1000pts.bin”</i>

3.46.2 MMEMory:RDIReCtory

描述	该命令用于删除指定文件夹
命令语法	MMEMory:RDIReCtory <参数> <参数>:=文件路径（包含操作文件夹名称）
示例	删除文件夹路径为“Local/”下名为 test 的文件夹： <i>MMEMory:RDIReCtory “Local/test”</i>

3.46.3 MMEMory:MDIReCtory

描述	该命令用于新建指定路径的文件夹
命令语法	MMEMory:MDIReCtory <参数> <参数>:=文件路径（包含操作文件夹名称）
示例	新建一个“Local”路径下的名为 test 的文件夹： <i>MMEMory:MDIReCtory “Local/test”</i>

3.46.4 MMEMory:CATalog

描述	该命令用于查询指定路径下的全部文件和文件夹或查看指定格式文件
命令语法	
查询语法	查看路径下的所有文件和文件夹

	<p>MMEMory:CATalog? <参数> <参数>:=文件夹路径</p> <p>MMEMory:CATalog: <参数 1>?<参数 2> <参数 1>:={(DATA:ARbitrary),(STATe:XMLanguage)} <参数 2>:=文件夹路径</p>
响应格式	<p>剩余内存大小，已用内存大小 “文件名，文件类型，文件大小”</p>
示例	<p>查看“Local/”路径下的所有文件和文件夹： <i>MMEMory:CATalog? "Local"</i></p> <p>查看“Local/”路径下的“.arb”或“.ARB”后缀的文件： <i>MMEMory:CATalog:DATA:ARbitrary? "Local"</i></p> <p>查看“Local/”路径下的“.xml”或“.XML”后缀文件： <i>MMEMory:CATalog:STATe:XMLanguage? "Local"</i></p>

3.46.5 MMEMory:COPY

描述	此命令复制一个文件或文件夹。
命令格式	<p>MMEMory:COPY <参数> <参数>:=“源文件的路径”，“目标路径”。</p> <p>复制文件：源文件路径和目标路径均包含操作文件名称 复制文件夹：源文件路径包含操作文件夹和目标路径不包含操作文件名称</p>
示例	<p>复制路径为“Local/test/1000pts.bin”的文件并粘贴至 “Local/1000pts.bin” <i>MMEMory:COPY "Local/test/1000pts.bin", "Local/1000pts.bin"</i></p> <p>复制路径为“Local/src”的文件/文件夹并粘贴至“Local/copy/”文件夹中 <i>MMEMory:COPY "Local/src", "Local/copy"</i></p>

3.46.6 MMEMory:MOVE

描述	此命令移动一个文件或文件夹到新的路径下
命令语法	MMEMory:MOVE <参数>

	<p><参数>:= “源文件/文件夹的路径”, “目标路径”</p> <p>移动文件: 源文件路径和目标路径均包含操作文件名称</p> <p>移动文件夹: 源文件路径包含操作文件夹和目标路径不包含操作文件名称</p>
示例	<p>移动文件路径为“Local/test/1000pts.bin” 的文件到路径 “Local/1000pts.bin”</p> <p><i>MMEMory:MOVE “Local/test/1000pts.bin”, “Local/1000pts.bin”</i></p> <p>移动文件夹路径为“Local/src” 到文件夹路径为 “Local/copy/”下</p> <p><i>MMEMory:MOVE “Local/src”, “Local/copy/”</i></p>

3.46.7 MMEMory:SAVE:XML

描述	此命令保存一个 xml 配置文件到默认路径或指定路径
命令语法	<p>MMEMory:SAVE:XML <参数></p> <p><参数>:= {保存路径, 包括文件名和后缀}。</p>
示例	<p>保存 test.xml 文件到本地</p> <p><i>MMEMory:SAVE:XML “Local/test.xml” 或</i> <i>MMEMory:SAVE:XML “test.xml”</i></p> <p>保存 test.xml 文件到网路存储盘</p> <p><i>MMEMory:SAVE:XML “net_storage/test.xml”</i></p> <p>保存 test.xml 文件到 U 盘</p> <p><i>MMEMory:SAVE:XML “U-disk0/test.xml”</i></p>

3.46.8 MMEMory:LOAD:XML

描述	此命令从默认路径或指定路径加载一个 xml 配置文件
命令语法	<p>MMEMory:LOAD:XML <参数></p> <p><参数>:= {路径, 包括文件名和后缀}。</p>

示例	<p>从本地加载 test.xml 文件</p> <p><i>MMEMory:LOAD:XML "Local/test.xml" 或</i></p> <p><i>MMEMory:LOAD:XML "test.xml"</i></p>
	<p>从网路存储盘加载 test.xml 文件</p> <p><i>MMEMory:LOAD:XML "net_storage/test.xml"</i></p>
	<p>从 U 盘加载 test.xml 文件</p> <p><i>MMEMory:LOAD:XML "U-disk0/test.xml"</i></p>

3.46.9 MMEMory:TRANSfer

描述	此命令可发送自定义的数据到指定路径下指定的 bin 文件
命令语法	<p>MMEMory:TRANSfer <参数>#{data}</p> <p><参数>:= {保存路径, 包括路径、文件名及后缀}</p> <p>{data}:=数据的长度的长度+数据长度+二进制数据</p>
示例	<p>发送数据长度的长度为 1, 数据长度为 4 的数据到本地下的 wave1.bin</p> <p><i>MMEMory:TRANSfer "Local/wave1.bin",#14ABCD</i></p> <p>发送数据长度的长度为 1, 数据长度为 4 的数据到 U 盘下的 wave1.bin</p> <p><i>MMEMory:TRANSfer "U-disk0/wave1.bin",#14ABCD</i></p> <p>发送数据长度的长度为 1, 数据长度为 4 的数据到网络存储盘下的 wave1.bin</p> <p><i>MMEMory:TRANSfer "net_storage /wave1.bin",#14ABCD</i></p>

4 波形格式

本章节给出信号源使用到的波形文件格式的说明。通过这些说明，你可以了解如何自定义编辑波形文件，并结合上个章节列出的命令实现对信号源控制。

4.1 bin

bin 文件内容为二进制，文件内容就是各个点的码字值（码字范围-32768~32767），不支持手动编辑，sdg 导入文件时，保持当前的幅度，频率、偏移信息，直接将各码字值转换为电压输出。

备注：下表显示各机型 bin 文件长度

参数/机型	SDG 800	SDG 1000	SDG 2000X	SDG 5000	SDG 1000X	SDG 6000X-E	SDG 6000X	SDG 7000A
<length>	32KB	32KB	4B~16MB	32KB~1024KB	4B~16MB	4B~16MB	4B~40MB	4B~512MB
波形码字范围	-32768~32767	-32768~32767	-32768~32767	-32768~32767	-32768~32767	-32768~32767	-32768~32767	-32768~32767

4.2 csv/dat

csv 与 dat 文件内容为 text。朱雀支持的 CSV 文件分为头部信息段与波形数据段两部分。

1. csv 头部信息段内容包含以下四项，可以有多的信息项，但必须包含以下四项，多的项会被忽略。每项一行，以换行符结束。

信息项	描述
amp,value	波形的幅度
offset,value	波形的偏移
frequency,value	波形的频率
data length,value	波形的长度

2. 数据段数据每一组占一行。可用以下四种字符串之一作为起始标识（标识符占一行），放在正式数据之前。

Second,Volt

xpos,value

Time,Ampl

Second,Value

csv 格式数据示例图如下：

1	data length	16384	
2	frequency	1000	
3	amp	2	
4	offset	0	
5	phase	0	
6			
7			
8			
9			
10			
11			
12			
13	xpos	value	
14		1	0
15		2	0.000383

csv 文件去掉头部信息段，只保留数据段，就是.dat 格式

dat 格式数据示例图如下:

```
1 Source, CH1
2 Second, Value
3 -1.0000000000E-04, -3.764706E-02
4 -9.9999800000E-05, -3.764706E-02
5 -9.9999600000E-05, -4.705882E-02
6 -9.9999400000E-05, -6.117647E-02
7 -9.9999200000E-05, -4.705882E-02
8 -9.9999000000E-05, -6.117647E-02
9 -9.9998800000E-05, -2.823529E-02
10 -9.9998600000E-05, -4.235294E-02
11 -9.9998400000E-05, -3.764706E-02
```

4.3 mat

1. mat 文件格式说明

MAT 文件由一个格式固定的 128 字节的文件头部信息 (mat_head)，和两个数据单元组成

a) 文件头部用如下结构体表示，此处只需关注 endian_indicator，朱雀只支持 IM 模式：

```
typedef struct
{
    char descriptive[116];
    char data_offset[8];
    uint16 version;                //导出 mat 文件的软件版本
    char endian_indicator[2];      //值为 IM 或者 MI，表示大小端模式
}cfg_mat_header_t;
```

b) 每个数据单元起始处有一个数据头部信息 (data_head，为 88 字节)，用于说明数据单元的占用的字节数、数据类型，数据块名字等信息。

数据块头部用如下结构体表示：

```
typedef struct
{
    u32 data_type;                //值必须为 CFG_MI_MATRIX
    u32 array_len;
    u32 array_flag_data_type;
    u32 array_flag_data_len;
    u32 array_flag_data_1;
    u32 array_flag_data_2;
    u32 dimensions_array_data_type;
    u32 dimensions_array_data_len;
    u32 dimensions_array_data_1;
    u32 dimensions_array_data_2;
```

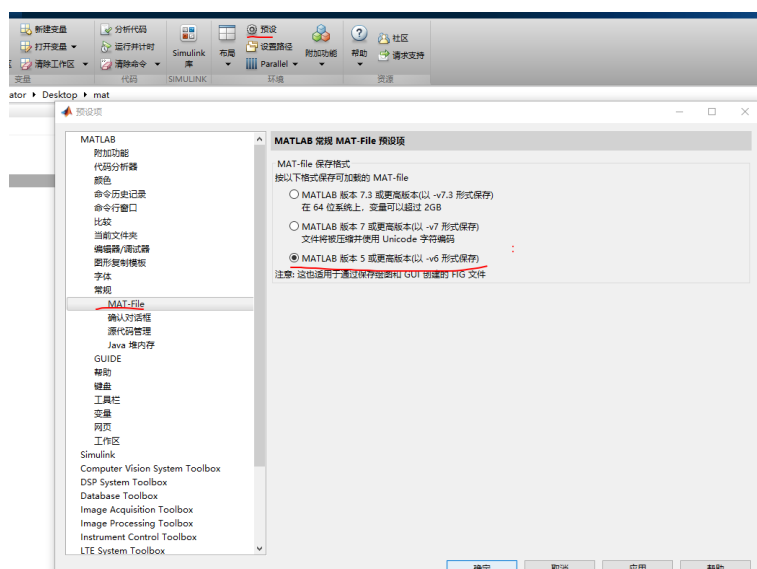
```
    u32 array_name_data_type;
    u32 array_name_data_len;
    char array_name_data[32];    //示波器导出文件此处为 XX_time 或者 XX_data
    u32 data_tag_data_type;      //后边数据区数据的类型
    u32 data_tag_data_len;      //后边数据区数据的大小
}matlab_data_head_t;
```

其中 XX_data_type 表示数据类型，对应到如下枚举值：

```
typedef enum
{
    CFG_MI_INT8=1,          //8 bit, signed
    CFG_MI_UINT8,          //8 bit, unsigned
    CFG_MI_INT16,          //16-bit, signed
    CFG_MI_UINT16,         //16-bit, unsigned
    CFG_MI_INT32,          //32-bit, signed
    CFG_MI_UINT32,         //32-bit, unsigned
    CFG_MI_SINGLE,         //IEEE 754 single format
    CFG_MI_RESERVED1,
    CFG_MI_DOUBLE,         //9, IEEE 754 double format
    CFG_MI_RESERVED2,
    CFG_MI_RESERVED3,
    CFG_MI_INT64,          //12, 64-bit, signed
    CFG_MI_UINT64,         //64-bit, unsigned
    CFG_MI_MATRIX,         //MATLAB array
    CFG_MI_COMPRESSED,     //Compressed Data
    CFG_MI_UTF8,           //Unicode UTF-8 Encoded Character Data
    CFG_MI_UTF16,          //Unicode UTF-16 Encoded Character Data
    CFG_MI_UTF32,          //Unicode UTF-32 Encoded Character Data
```

```
}cfg_mat_data_type_t;
```

2. matlab 导出的.mat 文件默认为压缩格式，需按以下方式设置为非压缩方式保存。并且只支持两个数据段的文件。时间数据需用 double，波形数据需用 single



4.4 awg

Awg 文件内容为 text。Awg 文件分为以下两个部分：

1. sequence 设置信息。可包括（非必须）以下字段，描述 sequence 设置。以下字段都不是必须，缺少部分字段不影响文件调用。缺少的参数将保持当前参数不变。也可有多余的字段，多余的字段会被忽略。

amplitude_scale,value	sequence 幅度。Value={0~1 的浮点数}。
run_mode,mode	运行模式。 mode = { continuous , tcontinuous , burst , step , advance }
burst_count,value	单次触发时的脉冲串数。Value=整数。
timer_interval,value	定时器触发时的定时时间。Value=浮点数。
trigger_mode,mode	触发方式。mode={ button , timer , external }
burst_hold_type,type	运行模式为 burst 时空闲电平类型。 type = { Zero , Start Value , End Value }
trigger_edge,edge	外部触发时的触发沿极性。edge={ raise , fall }
interp_mode,mode	插值算法类型。 Mode = { zero_hold , linear , sinc , sinc_27 , sinc_13 }
increasing_strategy,value	插值策略。 value={ interpolation , return_zero , hold_last , multiplication }
decreasing_strategy,value	抽值策略。value={ decimation , cut_tail , cut_head }
mark_skew,value	maker_skew 的值。Value=浮点数
file_type,SDS	示波器导出 awg 文件标志。（仅示波器导出的.awg 文件有此字段）。如果有此字段，段信息描述的波形文件需和.awg 文件在相同目录下。
start_play_segment,value	输出起始段。value=整数。若没有此字段，sequence 将从第一个段开始输出

2. segment 段信息。段信息每个 segment 占一行。各参数之间以逗号分割。每个段信息以 segment_N 开头。其中 N 为段编号，从 0 开始依次加 1。

段信息有两种格式，根据是否包含字段 file_name 来判断。

格式 1：包含 file_name 字段，按一定顺序存储每个参数的参数名与值。格式如下：

```
segment_0,file_name,2_stairup_ram.bin,offset,0,amplitude,4,cfg_len,32768,repetition,1,goto,0
,goto_mode,next,marker_switch,1,marker_pos,0,segment_numb,0,segment_store_addr,0,wait_ev
```

ent,none

goto_mode 的值可为: next, item。

wait_event 的值可为: none, button, timer, external。

格式 2: 无 file_name 字段, 按一定顺序存储每个参数的值 (无参数名)。格式如下:

```
segment_0,C1_seg00001.csv,-0.047059,4.141177,50000,1,-1,next,0,0
```

各参数含义:

段号 (segment_0), 文件名 (C1_seg00001.csv), 偏移 (-0.047059), 幅度 (4.141177), 点数 (50000), 重复次数 (1), 跳转段号 (-1, 因为后一个参数跳转模式为 next), 跳转模式 (next), marker_switch (0), marker_pos (0)

4.5 hop

hop 文件是用于存储跳频序列的文件。文件内分为三个数据块存储三个表格，分别存储：跳频频率表，频率跳转表，频率过滤表。文件内容为 text，可以手动编辑。

Hop 文件有三个数据块，分行存储，每一行存储一个参数或数据。

第一行为文件版本号：**Ver:1.0**

第二行为跳频频率表版本号：**FreqListVer:1.0**

第三行为频率表的数据起始标记：**freq_list_start**

下来是频率表的数据

最后频率表以结束标记结束：**freq_list_end**

接下来是频率跳转表、频率过滤表，结构和频率表一样，只是关键字不同，如下：

OrderListVer:1.0 -- 频率跳转表版本号

order_list_start -- 频率跳转表数据起始标记

order_list_end -- 频率跳转表数据结束标记

AvoidListVer:1.0 -- 频率过滤表版本号

avoid_list_start -- 频率过滤表数据起始标记

avoid_list_end -- 频率过滤表数据结束标记

数据块中每行两个数据以逗号分割。

数据示例：

```
Ver:1.0

FreqListVer:1.0
freq_list_start
1,1e+06
2,5e+06
3,5.25253e+06
freq_list_end

OrderListVer:1.0
order_list_start
1,1
2,2
3,3
order_list_end

AvoidListVer:1.0
avoid_list_start
1e+06,1.1e+06
2e+06,2e+06
5e+06,5e+06
avoid list end
```

4.6 wav/arb

wav/arb 文件为 IQ 波形文件。wav/arb 文件由头部信息 (header) 与波形数据 (data) 两个部分组成。header 是 text 数据。波形数据是二进制数据。

header 必须以字符串“IQData,”结尾。从文件起始到字符串“IQData,”，是 text 格式的文件信息。字符串“IQData,”之后就是二进制的波形数据。

朱雀在加载 wav/arb 文件时，会从 header 中读取以下关键字 (key_str) 进行解析。如果下面某些关键字在 header 中不存在，对应信息会被设为默认值或置空。如果有多余的信息，会被忽略。“key_str,value”的形式构成一组描述信息，每组描述信息之间以逗号分割。以下是一个 wav/arb 文件的 header 实例，及朱雀会解析的各关键字含义

```
FileType,IQ,Version,2.0,FileName,test.ARB,DataSourceType,PN9,SymbolLength,512,SymbolRate,1000000,ModulationType,16QAM,FilterType,RootCosine,FilterBandwidth,0,FilterAlpha,0.5,FilterLength,32,OverSampling,2,ActualSampleLength,512,SampleRate,2000000.000000,RMS,0.684953707203608,DataLength,1024,IQData,
```

FileType	文件类型 (IQ)
Version	版本号
DataSourceType	
SymbolLength	符号长度
SymbolRate	符号率
ModulationType	调制类型
FilterType	滤波器类型
FilterBandwidth	
FilterAlpha	滚降系数
FilterLength	滤波器长度
OverSampling	过采样点
ActualSampleLength	
SampleRate	采样率
DataLength	
RMS	
IQData	

5 编程示例

本章节给出了一些编程示例。通过这些例子，你可以了解如何使用 VISA 或者 sockets，并结合上节列出的命令实现对信号源控制。通过下面的例子，你可以开发更多应用。

5.1 VISA 应用示例

5.1.1 VC++ 示例

环境： Win7 32 位系统, Visual Studio

描述： 分别通过 USBTMC 和 TCP/IP 访问信号源，并在 NI-VISA 上发送“*IDN?”命令来查询设备信息。

步骤：

1. 打开 Visual Studio 软件，新建一个 VC++ win32 console project。

2. 设置调用 NI-VISA 库的项目环境。此处给出两种设置方法，分别为静态和动态：

a) 静态：

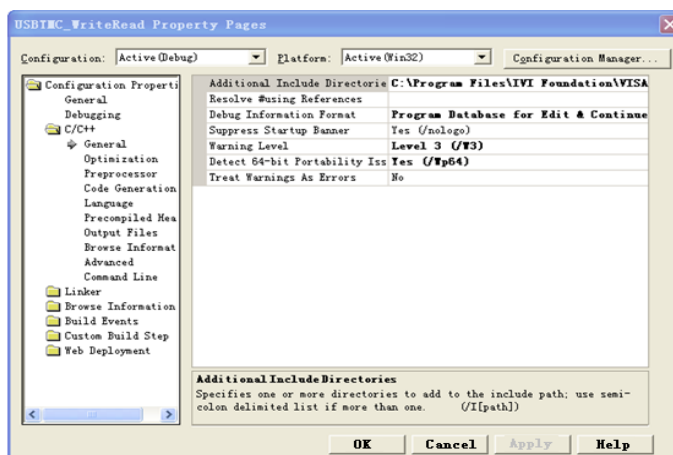
在 NI-VISA 安装路径找: visa.h、visatype.h、 visa32.lib 文件，将它们复制到 VC++项目的根路径下并添加到项目中。在 projectname.cpp 文件上添加下列两行代码：

```
#include "visa.h"
```

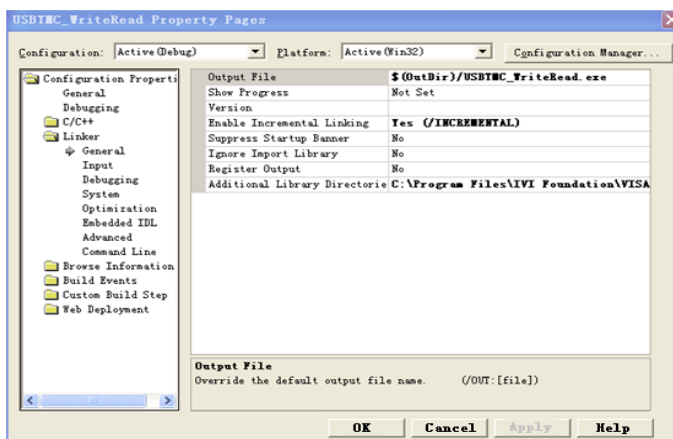
```
#pragma comment(lib,"visa32.lib")
```

b) 动态：

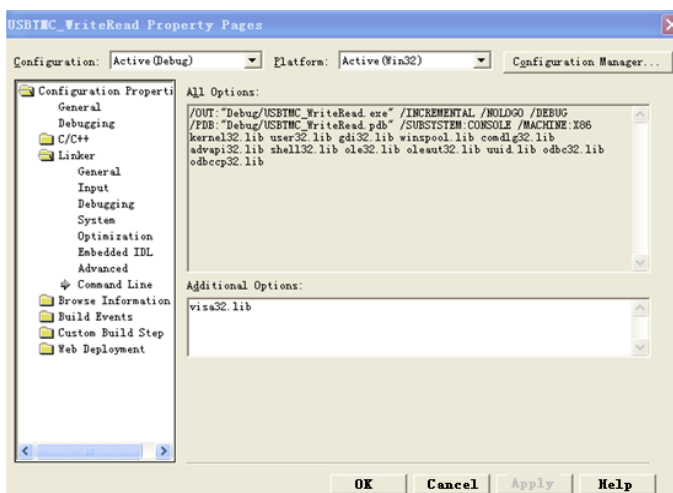
点击"project>>properties"，在属性对话框左侧选择“c/c++---General”中，将“Additional Include Directories”项的值设置为 NI-VISA 的安装路径（例如：C:\Program Files\NI Foundation\VISA\WinNT\include），如下图所示：



在属性对话框左侧选择" Linker---General " ,并将“Additional Library Directories”项的值设置为 NI-VISA 的安装路径。(例如: C:\Program Files\IVI Foundation\VISA\WinNT\include), 如下图所示:



在属性对话框左侧选择"Linker---Command Line" , 将“Additional”项的值设置为 visa32.lib, 如下图所示:



在 projectname.cpp 文件上添加 visa.h 文件:

```
#include<visa.h>
```

3. 编码:

a) USBTMC:

```
int Usbtmc_test()
{
    /* This code demonstrates sending synchronous read & write commands */
    /* to an USB Test & Measurement Class (USBTMC) instrument using      */
    /* NI-VISA                                                              */
    /* The example writes the "*IDN?\n" string to all the USBTMC          */
    /* devices connected to the system and attempts to read back          */
    /* results using the write and read functions.                          */
    /* The general flow of the code is  */
    /*   Open Resource Manager          */
    /*   Open VISA Session to an Instrument */
    /*   Write the Identification Query Using viPrintf */
    /*   Try to Read a Response With viScanf */
    /*   Close the VISA Session          */
    /*******/

    ViSession defaultRM;
    ViSession instr;
    ViUInt32 numInstrs;
    ViFindList findList;
    ViStatus status;

    char instrResourceString[VI_FIND_BUFLLEN];

    unsigned char buffer[100];
```

```
int i;

/** First we must call viOpenDefaultRM to get the manager
 * handle. We will store this handle in defaultRM.*/

status=viOpenDefaultRM (&defaultRM);

if (status<VI_SUCCESS)

{

    printf ("Could not open a session to the VISA Resource Manager!\n");

    return status;

}

/** Find all the USB TMC VISA resources in our system and store the number of
resources in the system in numInstrs. */

status = viFindRsrc (defaultRM, "USB?*INSTR", &findList, &numInstrs,
instrResourceString);

if (status<VI_SUCCESS)

{

    printf ("An error occurred while finding resources.\nPress 'Enter' to continue.");

    fflush(stdin);

    getchar();

    viClose (defaultRM);

    return status;

}

/** Now we will open VISA sessions to all USB TMC instruments.
 * We must use the handle from viOpenDefaultRM and we must
 * also use a string that indicates which instrument to open. This
 * is called the instrument descriptor. The format for this string
 * can be found in the function panel by right clicking on the
 * descriptor parameter. After opening a session to the
 * device, we will get a handle to the instrument which we
```

```
* will use in later VISA functions. The AccessMode and Timeout
* parameters in this function are reserved for future
* functionality. These two parameters are given the value VI_NULL.*/
for (i=0; i<int(numInstrs); i++)
{
    if (i> 0)
    {
        viFindNext (findList, instrResourceString);
    }
    status = viOpen (defaultRM, instrResourceString, VI_NULL, VI_NULL, &instr);
    if (status<VI_SUCCESS)
    {
        printf ("Cannot open a session to the device %d.\n", i+1);
        continue;
    }
    /* * At this point we now have a session open to the USB TMC instrument.
    * We will now use the viPrintf function to send the device the string "*IDN?\n",
    * asking for the device's identification. */
    char * cmmand ="*IDN?\n";
    status = viPrintf (instr, cmmand);
    if (status<VI_SUCCESS)
    {
        printf ("Error writing to the device %d.\n", i+1);
        status = viClose (instr);
        continue;
    }
    /** Now we will attempt to read back a response from the device to
    * the identification query that was sent. We will use the viScanf
```

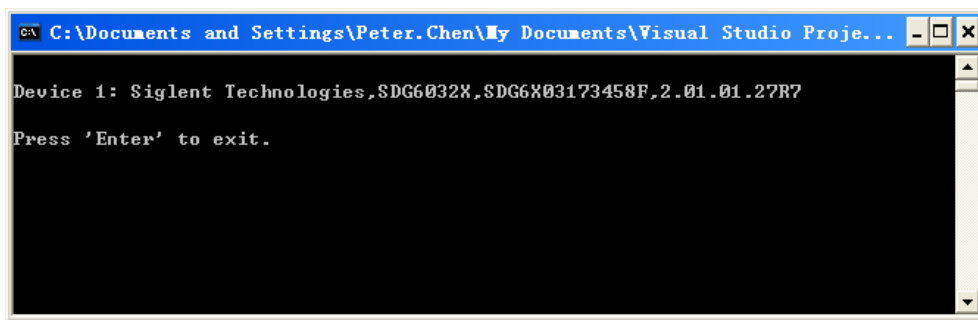
```
* function to acquire the data.
* After the data has been read the response is displayed.*/
status = viScanf(instr, "%t", buffer);
if (status<VI_SUCCESS)
{
    printf ("Error reading a response from the device %d.\n", i+1);
}
else
{
    printf ("\nDevice %d: %s\n", i+1 , buffer);
}
status = viClose (instr);

}

/** Now we will close the session to the instrument using
* viClose. This operation frees all system resources.          */
status = viClose (defaultRM);
printf("Press 'Enter' to exit.");
fflush(stdin);
getchar();
return 0;
}

int _tmain(int argc, _TCHAR* argv[])
{
    Usbtmc_test();
    return 0;
}
```

运行结果：



```
C:\Documents and Settings\Peter.Chen\My Documents\Visual Studio Proje...
Device 1: Siglent Technologies,SDG6032X,SDG6X03173458F,2.01.01.27R7
Press 'Enter' to exit.
```

b) TCP/IP:

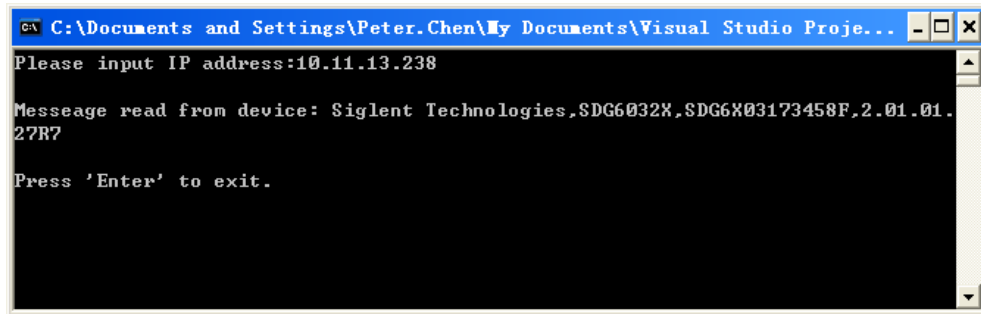
```
int TCP_IP_Test(char *pIP)
{
    char outputBuffer[VI_FIND_BUFLLEN];
    ViSession defaultRM, instr;
    ViStatus status;
    /* First we will need to open the default resource manager. */
    status = viOpenDefaultRM (&defaultRM);
    if (status<VI_SUCCESS)
    {
        printf("Could not open a session to the VISA Resource Manager!\n");
    }
    /* Now we will open a session via TCP/IP device */
    char head[256] ="TCPIP0::";
    char tail[] = "::INSTR";
    strcat(head,pIP);
    strcat(head,tail);
    status = viOpen (defaultRM, head, VI_LOAD_CONFIG, VI_NULL, &instr);
    if (status<VI_SUCCESS)
```

```
{
    printf ("An error occurred opening the session\n");
    viClose(defaultRM);
}
status = viPrintf(instr, "*idn?\n");
status = viScanf(instr, "%t", outputBuffer);
if (status<VI_SUCCESS)
{
    printf("viRead failed with error code: %x \n",status);
    viClose(defaultRM);
}
else
{
    printf ("\nMesseage read from device: %*s\n", 0,outputBuffer);
}
status = viClose (instr);
status = viClose (defaultRM);
printf("Press 'Enter' to exit.");
fflush(stdin);
getchar();
return 0;
}

int _tmain(int argc, _TCHAR* argv[])
{
    printf("Please input IP address:");
    char ip[256];
    fflush(stdin);
```

```
gets(ip);  
TCP_IP_Test(ip);  
return 0;  
}
```

运行结果：



```
GV C:\Documents and Settings\Peter.Chen\My Documents\Visual Studio Proje...  
Please input IP address:10.11.13.238  
Message read from device: Siglent Technologies,SDG6032X,SDG6X03173458F.2.01.01.  
27R7  
Press 'Enter' to exit.
```

5.1.2 VB 示例

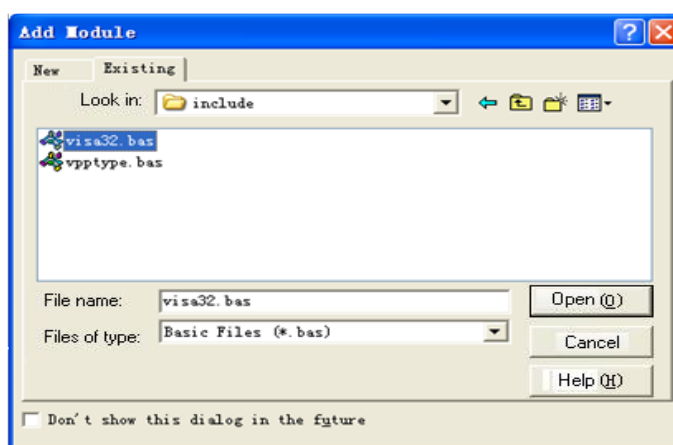
环境： Windows7 32 位系统, Microsoft Visual Basic 6.0

描述： 分别通过 USBTMC 和 TCP/IP 访问信号源，并在 NI-VISA 上发送“*IDN?”命令来查询设备信息。

步骤：

1. 打开 Visual Basic 软件，并新建一个标准的应用程序项目。

设置调用 NI-VISA 库项目环境：点击 Existing tab of Project>>Add Existing Item，在 NI-VISA 安装路径下的“include”文件夹中查找 visa32.bas 文件并添加该文件。如下图所示：



2. 编码：

a) USBTMC:

```
PrivateFunction Usbtmc_test() AsLong
```

```
' This code demonstrates sending synchronous read & write commands
```

```
' to an USB Test & Measurement Class (USBTMC) instrument using
```

```
' NI-VISA
```

```
' The example writes the "*IDN?\n" string to all the USBTMC
```

```
' devices connected to the system and attempts to read back
```

```
' results using the write and read functions.
```

```
' The general flow of the code is
```

```
'   Open Resource Manager
```

- ' Open VISA Session to an Instrument
- ' Write the Identification Query Using viWrite
- ' Try to Read a Response With viRead
- ' Close the VISA Session

```
Const MAX_CNT = 200
```

```
Dim defaultRM AsLong
```

```
Dim instrsesn AsLong
```

```
Dim numInstrs AsLong
```

```
Dim findList AsLong
```

```
Dim retCount AsLong
```

```
Dim status AsLong
```

```
Dim instrResourceString AsString * VI_FIND_BUFLLEN
```

```
Dim Buffer AsString * MAX_CNT
```

```
Dim i AsInteger
```

```
' First we must call viOpenDefaultRM to get the manager
```

```
' handle. We will store this handle in defaultRM.
```

```
status = viOpenDefaultRM(defaultRM)
```

```
If (status < VI_SUCCESS) Then
```

```
resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
```

```
Usbtmc_test = status
```

```
ExitFunction
```

```
EndIf
```

```
' Find all the USB TMC VISA resources in our system and store the
```

```
' number of resources in the system in numInstrs.
```

```
status = viFindRsrc(defaultRM, "USB?*INSTR", findList, numInstrs, instrResourceString)
```

```
If (status < VI_SUCCESS) Then
    resultTxt.Text = "An error occurred while finding resources."
    viClose(defaultRM)
    Usbtmc_test = status
```

```
ExitFunction
```

```
EndIf
```

```
' Now we will open VISA sessions to all USB TMC instruments.
' We must use the handle from viOpenDefaultRM and we must
' also use a string that indicates which instrument to open. This
' is called the instrument descriptor. The format for this string
' can be found in the function panel by right clicking on the
' descriptor parameter. After opening a session to the
' device, we will get a handle to the instrument which we
' will use in later VISA functions. The AccessMode and Timeout
' parameters in this function are reserved for future
' functionality. These two parameters are given the value VI_NULL.
```

```
For i = 0 To numInstrs
```

```
If (i > 0) Then
```

```
    status = viFindNext(findList, instrResourceString)
```

```
EndIf
```

```
    status = viOpen(defaultRM, instrResourceString, VI_NULL, VI_NULL, instrsesn)
```

```
If (status < VI_SUCCESS) Then
```

```
    resultTxt.Text = "Cannot open a session to the device " + CStr(i + 1)
```

```
GoTo NextFind
```

```
EndIf
```

```
' At this point we now have a session open to the USB TMC instrument.
```

' We will now use the viWrite function to send the device the string "*IDN?",
' asking for the device's identification.

```
status = viWrite(instrsesn, "*IDN?", 5, retCount)
```

```
If (status < VI_SUCCESS) Then
```

```
resultTxt.Text = "Error writing to the device."
```

```
status = viClose(instrsesn)
```

```
GoTo NextFind
```

```
EndIf
```

' Now we will attempt to read back a response from the device to
' the identification query that was sent. We will use the viRead
' function to acquire the data.

' After the data has been read the response is displayed.

```
status = viRead(instrsesn, Buffer, MAX_CNT, retCount)
```

```
If (status < VI_SUCCESS) Then
```

```
resultTxt.Text = "Error reading a response from the device." + CStr(i + 1)
```

```
Else
```

```
resultTxt.Text = "Read from device: " + CStr(i + 1) + " " + Buffer
```

```
EndIf
```

```
status = viClose(instrsesn)
```

```
Next i
```

' Now we will close the session to the instrument using
' viClose. This operation frees all system resources.

```
status = viClose(defaultRM)
```

```
Usbtmc_test = 0
```

```
EndFunction
```

b) TCP/IP:

```
PrivateFunction TCP_IP_Test(ByVal ip AsString) AsLong
```

```
Dim outputBuffer AsString * VI_FIND_BUFLen
```

```
Dim defaultRM AsLong
```

```
Dim instrsesn AsLong
```

```
Dim status AsLong
```

```
Dim count AsLong
```

```
' First we will need to open the default resource manager.
```

```
status = viOpenDefaultRM(defaultRM)
```

```
If (status < VI_SUCCESS) Then
```

```
resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
```

```
TCP_IP_Test = status
```

```
ExitFunction
```

```
EndIf
```

```
' Now we will open a session via TCP/IP device
```

```
status = viOpen(defaultRM, "TCPIP0::" + ip + "::INSTR", VI_LOAD_CONFIG, VI_NULL,  
instrsesn)
```

```
If (status < VI_SUCCESS) Then
```

```
resultTxt.Text = "An error occurred opening the session"
```

```
viClose(defaultRM)
```

```
TCP_IP_Test = status
```

```
ExitFunction
```

```
EndIf
```

```
status = viWrite(instrsesn, "*IDN?", 5, count)
```

```
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error writing to the device."
EndIf

status = viRead(instrsesn, outputBuffer, VI_FIND_BUFLLEN, count)
If (status < VI_SUCCESS) Then
    resultTxt.Text = "Error reading a response from the device." + CStr(i + 1)
Else
    resultTxt.Text = "read from device:" + outputBuffer
EndIf

status = viClose(instrsesn)
status = viClose(defaultRM)
TCP_IP_Test = 0
EndFunction
```

c) Button control code:

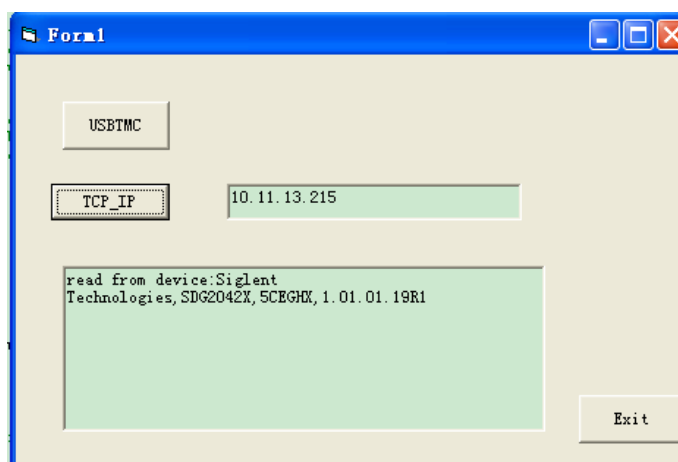
```
PrivateSub exitBtn_Click()
End
EndSub

PrivateSub tcpipBtn_Click()
Dim stat AsLong
    stat = TCP_IP_Test(ipTxt.Text)
If (stat < VI_SUCCESS) Then
    resultTxt.Text = Hex(stat)
EndIf
EndSub

PrivateSub usbBtn_Click()
Dim stat AsLong
```

```
stat = Usbtmc_test  
If (stat < VI_SUCCESS) Then  
    resultTxt.Text = Hex(stat)  
EndIf  
EndSub
```

运行结果：



5.1.3 MATLAB 示例

环境：Windows 7 32 位系统 MATLAB R2013a

描述：分别通过 USBTMC 和 TCP/IP 访问信号源，并在 NI-VISA 上发送“*IDN?”命令来查询设备信息。

步骤：

1. 打开 MATLAB 软件，并修改当前目录。在本示例中，当前目录修改为：“D:\USBTMC_TCPIP_Demo”。

2. 点击在 Matlab 界面的 File>>New>>Script 创建一个空的 M 文件。

3. 编码：

a) USBTMC:

```
function USBTMC_test()

% This code demonstrates sending synchronous read & write commands
% to an USB Test & Measurement Class (USBTMC) instrument using
% NI-VISA

%Create a VISA-USB object connected to a USB instrument
vu = visa('ni','USB0::0xF4ED::0xEE3A::sdg2000x::INSTR');

%Open the VISA object created
fopen(vu);

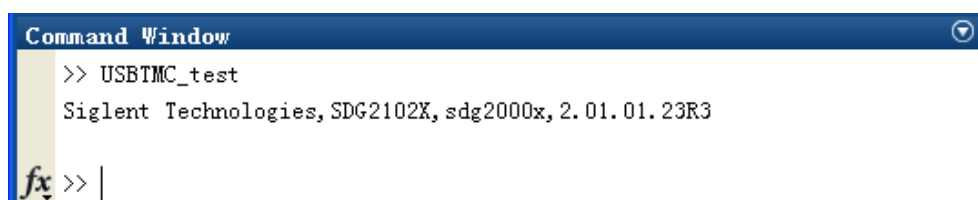
%Send the string "*IDN?",asking for the device's identification.
fprintf(vu,'*IDN?');

%Request the data
outputbuffer = fscanf(vu);
disp(outputbuffer);

%Close the VISA object
```

```
fclose(vu);  
delete(vu);  
clear vu;  
  
end
```

运行结果：



```
Command Window  
>> USBTMC_test  
Siglent Technologies, SDG2102X, sdg2000x, 2.01.01.23R3  
fx >> |
```

b) TCP/IP:

写一个名为 TCP_IP_Test 的函数:

```
function TCP_IP_test()  
  
% This code demonstrates sending synchronous read & write commands  
% to an TCP/IP instrument using NI-VISA  
  
%Create a VISA-TCP/IP object connected to an instrument  
%configured with IP address.  
vt = visa('ni',[TCPIP0::'10.11.13.32',::INSTR]);  
  
%Open the VISA object created  
fopen(vt);  
  
%Send the string "*IDN?", asking for the device's identification.  
fprintf(vt, '*IDN?');
```

%Request the data

```
outputbuffer = fscanf(vt);
```

```
disp(outputbuffer);
```

%Close the VISA object

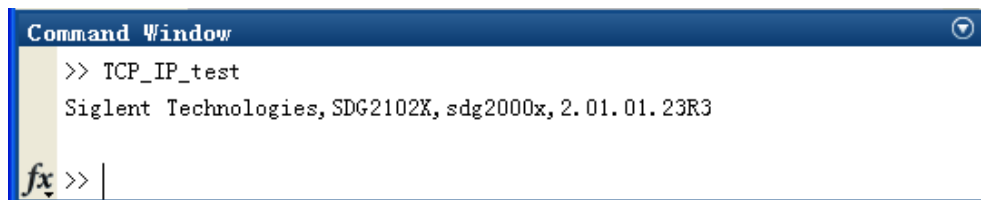
```
fclose(vt);
```

```
delete(vt);
```

```
clear vt;
```

```
end
```

运行结果：



```
Command Window
>> TCP_IP_test
Siglent Technologies, SDG2102X, sdg2000x, 2.01.01.23R3
fx >> |
```

5.1.4 LabVIEW 示例

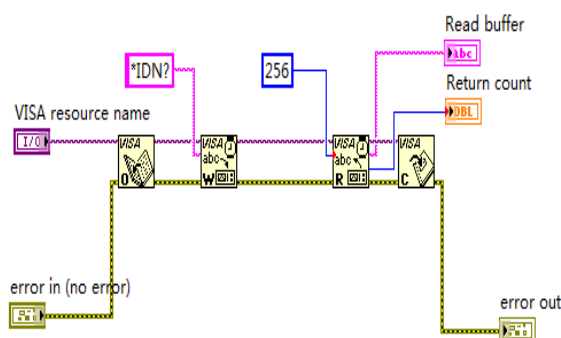
环境：Windows 7 32 位系统, LabVIEW 2011

描述：分别通过 USBTMC 和 TCP/IP 访问信号源，并在 NI-VISA 上发送“*IDN?”命令来查询设备信息。

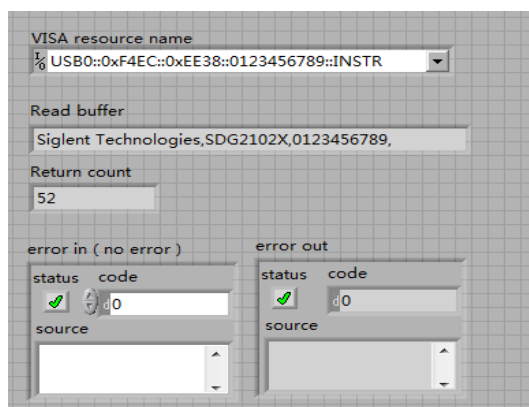
步骤：

1. 打开 LabVIEW 软件，并创建一个 VI 文件。
2. 添加控件。右击前面板界面，从控制列中选择并添加 VISA 资源名、错误输入、错误输出以及部分的指示符。
3. 打开框图界面。右击 VISA 资源名称，并在弹出菜单的 VISA Palette 中选择和添加下列功能：**VISA Write**、**VISA Read**、**VISA Open** 和 **VISA Close**。

4. 如下图连接它们：



5. 从 VISA 资源名列表中选择设备资源并运行程序。

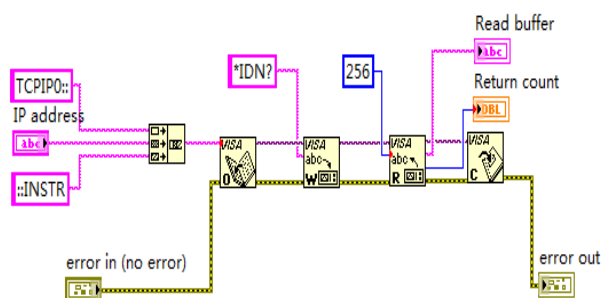


在此例中，VI 打开了一个 USBTMC 设备的 VISA 会话，并向设备写 *IDN? 命令，以及回读的

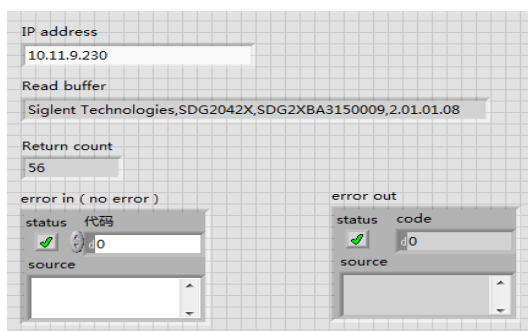
响应值。当所有通信完成时，VI 将关闭 VISA 会话。

6. 通过 TCP/IP 与设备通信类似于 USBTMC。但是你需要将 VISA 写函数和 VISA 读函数设置为同步 I/O。LabVIEW 默认设置为异步 I/O。右键单击节点，然后从快捷菜单中选择“Synchronous I/O Mod >>Synchronous”以实现同步写入或读取数据。

7. 按照下图连接它们：



8. 输入 IP 地址并运行程序。



5.1.5 Python2 示例

Environment: Python2.7, PyVISA 1.4

(请在安装 Python2.7 后安装 PyVISA。请在此网址上获取 PyVISA 安装指导：

<https://pyvisa.readthedocs.io/en/stable/getting.html>)

描述:使用 Python 脚本新建一个 8 个点的任意波形 (0x1000, 0x2000, 0x3000, 0x4000, 0x5000, 0x6000, 0x7000, 0x7fff) 并将波形数据保存为“wave1.bin”，然后将其发送到设备，最后从设备读取该波形并保存为“wave2.bin”。

下面是脚本的代码：

```
#!/usr/bin/env python2.7
# -*- coding: utf-8 -*-

import visa
import time
import binascii

#USB resource of Device
device_resource = "USB0::0xF4EC::0x1101::#15::INSTR"

#Little endian, 16-bit 2's complement
wave_points = [0x0010, 0x0020, 0x0030, 0x0040, 0x0050, 0x0060, 0x0070, 0xff7f]

def create_wave_file():
    """create a file"""
    f = open("wave1.bin", "wb")
    for a in wave_points:
```

```
b = hex(a)
b = b[2:]
len_b = len(b)
if (0 == len_b):
    b = '0000'
elif (1 == len_b):
    b = '000' + b
elif (2 == len_b):
    b = '00' + b
elif (3 == len_b):
    b = '0' + b
c = binascii.a2b_hex(b)    #Hexadecimal integer to ASCII encoded string
f.write(c)
f.close()

def send_wawe_data(dev):
    """send wave1.bin to the device"""
    f = open("wave1.bin", "rb")    #wave1.bin is the waveform to be sent
    data = f.read()
    print 'write bytes:',len(data)
    dev.write("C1:WVDT
WVNM,wave1,FREQ,2000.0,AMPL,4.0,OFST,0.0,PHASE,0.0,WAVEDATA,%s" % (data))    #"X" series
(SDG1000X/SDG2000X/SDG6000X/X-E)
    dev.write("C1:ARWV NAME,wave1")
    f.close()

def get_wave_data(dev):
    """get wave from the devide"""
```

```
f = open("wave2.bin", "wb")    #save the waveform as wave2.bin
dev.write("WVDT? user,wave1")    #"X" series (SDG1000X/SDG2000X/SDG6000X/X-E)
time.sleep(1)
data = dev.read()
data_pos = data.find("WAVEDATA,") + len("WAVEDATA,")
print data[0:data_pos]
wave_data = data[data_pos:]
print 'read bytes:',len(wave_data)
f.write(wave_data)
f.close()
```

```
if __name__ == '__main__':
    #####

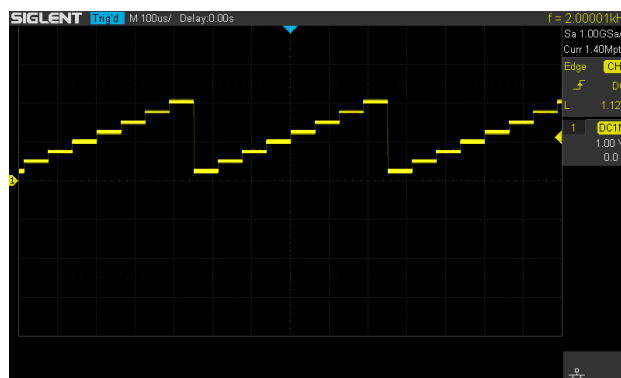
    device = visa.instrument(device_resource, timeout=5000, chunk_size = 40*1024)

    create_wave_file()

    send_wave_data(device)

    get_wave_data(device)
```

输出波形：



5.1.6 Python3 示例

Environment: Python3.6.5, PyVISA 1.9

```
#!/usr/bin/env python3.6.5
# -*- coding: utf-8 -*-

import visa
import time
import binascii

#USB resource of Device
device_resource = 'USB0::0xF4EC::0x1102::SDG7ABABQ5R0010::INSTR'

#Little endian, 16-bit2's complement
wave_points = [0x0080, 0x0070, 0x0060, 0x0040, 0x0050, 0x0060, 0x0070, 0xff7f,0x0050]

def create_wave_file():
    """create a file"""
    f = open("wave1.bin", "wb")
    for a in wave_points:
        b = hex(a)
        b = b[2:]
        len_b = len(b)
        if (0 == len_b):
            b = '0000'
        elif (1 == len_b):
            b = '000' + b
        elif (2 == len_b):
```

```

        b = '00' + b
    elif (3 == len_b):
        b = '0' + b
    c = binascii.a2b_hex(b)    #Hexadecimal integer to ASCII encoded string
    f.write(c)
f.close()

def send_wave_data(dev):
    """send wave1.bin to the device"""
    f = open("wave1.bin", "rb")    #wave1.bin is the waveform to be sent
    data = f.read()
    print ('write bytes:%s'%len(data))
    dev.write("C1:WVDT
WVNM,wave1,FREQ,2000.0,AMPL,4.0,OFST,0.0,PHASE,0.0,WAVEDATA,%s" % (data))    #"X" series
(SDG1000X/SDG2000X/SDG6000X/X-E)
    dev.write("C1:ARWV NAME,wave1")
    f.close()

def get_wave_data(dev):
    """get wave from the device"""
    f = open("wave2.bin", "wb")    #save the waveform as wave2.bin
    dev.write("WVDT? user,wave1")    #"X" series (SDG1000X/SDG2000X/SDG6000X/X-E)
    time.sleep(1)
    data = dev.read()
    data_pos = data.find("WAVEDATA,") + len("WAVEDATA,")
    print (data[0:data_pos])
    wave_data = data[data_pos:]
    print ('read bytes:%s'%len(wave_data))

```

```
f.write(wave_data)
```

```
f.close()
```

```
if __name__ == '__main__':
```

```
    """
```

```
    rm=visa.ResourceManager()
```

```
    device =rm.open_resource(device_resource, timeout=50000, chunk_size = 24*1024*1024)
```

```
    create_wave_file()
```

```
    send_wave_data(device)
```

```
    #get_wave_data(device)
```

5.1.7 Python3(Digital)示例

Environment: Python3.6.5, PyVISA 1.9

```
#!/usr/bin/env python3
```

```
# -*- coding: utf-8 -*-
```

```
import pyvisa as visa
```

```
import time
```

```
import binascii
```

```
# resource of Device
```

```
device_resource = 'USB0::0xF4EC::0x1102::SDG7ACBC5M0005::INSTR'
```

```
d7 = '000011110000' # Data stream of ch7 in digital
```

```
d6 = '101010101010' # Data stream of ch6 in digital
```

```
d5 = '010101010101' # Data stream of ch5 in digital
```

```
d4 = '110011001100' # Data stream of ch4 in digital
d3 = '000000111111' # Data stream of ch3 in digital
d2 = '111000111000' # Data stream of ch2 in digital
d1 = '001100110011' # Data stream of ch1 in digital
d0 = '110011001100' # Data stream of ch0 in digital
other = '00000000' # The last 8ch data is 0
wave_points = []
for i7, i6, i5, i4, i3, i2, i1, i0 in zip(d7, d6, d5, d4, d3, d2, d1, d0):
    a = i7 + i6 + i5 + i4 + i3 + i2 + i1 + i0 + other
    wave_points.append(int(a, 2))

def create_wave_file():
    """create a file"""
    f = open("wave1.bin", "wb")
    for a in wave_points:
        b = hex(a)
        b = b[2:]
        len_b = len(b)
        if (0 == len_b):
            b = '0000'
        elif (1 == len_b):
            b = '000' + b
        elif (2 == len_b):
            b = '00' + b
        elif (3 == len_b):
            b = '0' + b
        c = binascii.unhexlify(b) # Hexadecimal integer to ASCII encoded string
        f.write(c)
```

```
f.close()
```

```
def send_wave_data(dev):
```

```
    """send wave1.bin to the device"""
```

```
    f = open("wave1.bin", "rb") # wave1.bin is the waveform to be sent
```

```
    data = f.read().decode("latin1")
```

```
    print('write class:', type(data))
```

```
    print('write bytes:', len(data))
```

```
    dev.write_termination = "
```

```
    dev.write("DIG:WVDT WVNM,digital, WAVEDATA,%s" % (data), encoding='latin1')
```

```
    f.close()
```

```
    return data
```

```
if __name__ == '__main__':
```

```
    """
```

```
    rm = visa.ResourceManager()
```

```
    device = rm.open_resource(device_resource, timeout=50000, chunk_size=24 * 1024 *
```

```
1024)
```

```
    create_wave_file()
```

```
    send = send_wave_data(device)
```

```
    print('Done')
```

5.2 Sockets 应用示例

5.2.1 Python 示例

Python 有一个用于访问 socket 接口的低级的网络模块。基于 sockets 编写的 Python 脚本可用于用于执行各种测试和测量任务。

环境: Windows 7 32 位系统, Python v2.7.5

Description: 打开一个 socket,发送一个查询操作并循环执行 10 次后关闭 socket。注意: 程序中的 SCPI 命令的字符串必须以“\n”字符 (换行) 作为结尾。

下面是脚本的代码:

```
#!/usr/bin/env python
#-*- coding:utf-8 -*-
#-----
# The short script is a example that open a socket, sends a query,
# print the return message and closes the socket.
#-----

import socket # for sockets
import sys # for exit
import time # for sleep
#-----

remote_ip = "10.11.13.40" # should match the instrument's IP address
port = 5025 # the port number of the instrument service
count = 0

def SocketConnect():
    try:
```

```
#create an AF_INET, STREAM socket (TCP)
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

except socket.error:

    print ('Failed to create socket.')

    sys.exit();

try:

    #Connect to remote server

    s.connect((remote_ip , port))

except socket.error:

    print ('failed to connect to ip ' + remote_ip)

return s

def SocketQuery(Sock, cmd):

    try :

        #Send cmd string

        Sock.sendall(cmd)

        time.sleep(1)

    except socket.error:

        #Send failed

        print ('Send failed')

        sys.exit()

    reply = Sock.recv(4096)

    return reply

def SocketClose(Sock):

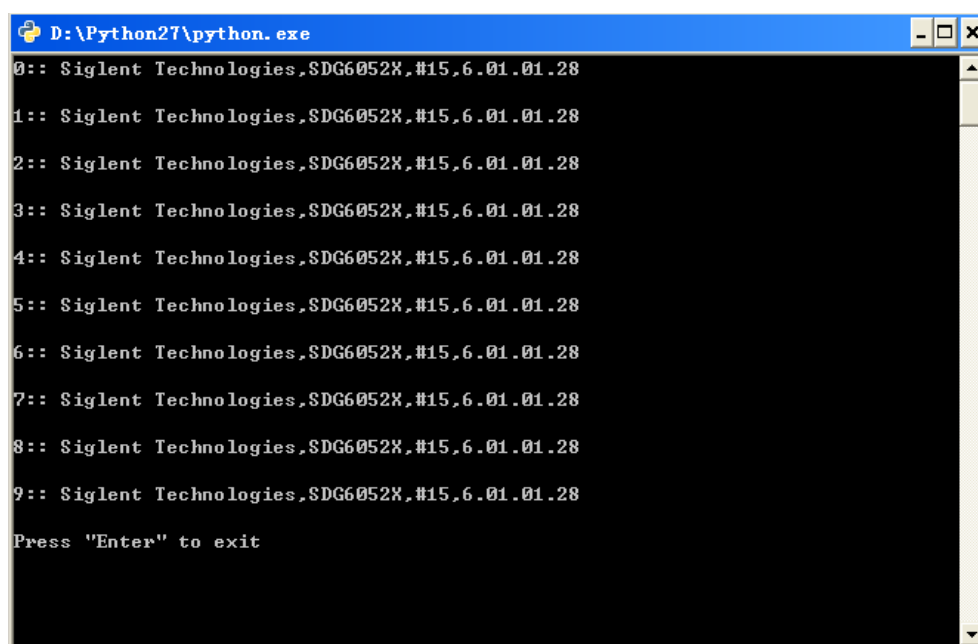
    #close the socket

    Sock.close()

    time.sleep(.300)
```

```
def main():  
    global remote_ip  
    global port  
    global count  
  
    # Body: send the SCPI commands *IDN? 10 times and print the return message  
    s = SocketConnect()  
    for i in range(10):  
qStr = SocketQuery(s, b'*IDN?\n')  
        print (str(count) + "::" + str(qStr))  
        count = count + 1  
    SocketClose(s)  
    input('Press "Enter" to exit')  
  
if __name__ == '__main__':  
    proc = main()
```

运行结果：



```
D:\Python27\python.exe  
0:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
1:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
2:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
3:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
4:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
5:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
6:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
7:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
8:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
9:: Siglent Technologies,SDG6052X,#15,6.01.01.28  
Press "Enter" to exit
```

6 索引

[*IDN](#)

[*OPC](#)

[*RST](#)

A

[ARWV ArbWaVe](#)

B

[BSWV BaSic_WaVe](#)

[BTWV BursTWaVe](#)

[BUZZ BUZZer](#)

C

[CHDR Comm_HeaDeR](#)

[COUP COUPling](#)

[CMBN CoMBiNe](#)

F

[FCNT FreqCouNter](#)

H

[HARM HARMonic](#)

I

[IQ:CENtIQ:CENTerfreq](#)

[IQ:SAMPIQ:SAMPlerate](#)

[IQ:SYMBIQ:SYMBOLrate](#)

[IQ:AMPLIQ:AMPLitude](#)

[IQ:IQAD:GAINIQ:IQADjustment:GAIN](#)

[IQ:IQAD:IOffsetIQ:IQADjustment:IOffset](#)

[IQ:IQAD:QOffsetIQ:IQADjustment:QOffset](#)

[IQ:IQAD:QSKIQ:IQADjustment:QSKew](#)

[IQ:TRIG:SOURIQ:TRIGger:SOURce](#)

[IQ:WAVE:BUILIQ:WAVEload:BUILtin](#)

[IQ:WAVE:USERIQ:WAVEload:USERstored](#)

[IVNT INVERT](#)

L

[LAGG LAnGuaGe](#)

M

[MDWV MoDulateWaVe](#)

[MODE MODE](#)

N

[NBFM NumBer_ForMat](#)

O

[OUTP OUTPut](#)

P

[PACP ParaCoPy](#)

R

[ROSC ROSCillator](#)

S

[SCFG Sys_CFG](#)

[SCSV SScreen_SaVe](#)

[SWWW SweepWaVe](#)

[SYNC SYNC](#)

[STL StoreList](#)

[SYST:COMM:LAN:IPADSYSTem:COMMunicate:LAN:IPADdress](#)

[SYST:COMM:LAN:SMASSYSTem:COMMunicate:LAN:SMASK](#)

[SYST:COMM:LAN:GAT SYSTem:COMMunicate:LAN:GATeway](#)

[SRATE SampleRATE](#)

W

[WVDT WVDT](#)

V

[VOLTPRTVOLTPT](#)

[VKEY VirtualKEY](#)

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、频谱分析仪、函数/任意波形发生器、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、精密源表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, 在马来西亚槟城州设有生产基地, 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

声明

 是深圳市鼎阳科技股份有限公司的注册商标, 事先未经允许, 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。

