SSG6M80A 系列

多通道微波信号发生器

编程手册

CN01A



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

目录

1 编程概述 1

1.1 建立通信 1

1.1.1 NI-VISA 的安装 1

1.1.2 连接仪器 4

1.2 远程控制的实现 6

1.2.1 用户自定义程序 6

1.2.2 通过 NI-MAX 发送 SCPI 命令 6

1.2.3 通过 Telnet 发送 SCPI 命令 11

1.2.4 通过 Socket 发送 SCPI 命令 12

2 SCPI 语言简介 13

2.1 命令格式 13

2.2 命令符号 13

2.3 参数类型 14

2.4 命令缩写 15

3 命令系统 16

3.1 IEEE 通用命令子系统 16

3.1.1 设备信息查询（*IDN?） 16

3.1.2 复位（*RST） 16

3.1.3 清除状态（*CLS） 16

3.1.4 标准事件状态使能（*ESE） 17

3.1.5 标准事件状态寄存器询问（*ESR?） 17

3.1.6 操作完成（*OPC） 17

3.1.7 允许服务请求（*SRE） 18

3.1.8 状态字节询问（*STB?） 18

3.1.9 等待继续（*WAI） 18

3.1.10 自测试询问（*TST?） 18

3.2 SYSTEM 命令子系统 19

3.2.1 系统配置 19
3.2.2 系统复位 27
3.3 OUTPUT 命令子系统 29
3.3.1 RF 端口输出（[:SOURce<ch>]:OUTPut[:STATe]） 29
3.3.2 模拟调制开关状态（[:SOURce<ch>]:OUTPut:MODulation[:STATe]） 29
3.4 SOURCE 命令子系统 30
3.4.1 RF 端口输出（[:SOURce<ch>]:OUTPut） 30
3.4.2 软件触发（[:SOURce<ch>]:*TRG） 30
3.4.3 频率设置 30
3.4.4 幅度设置 34
3.4.5 扫描设置 41
3.4.6 功率计设置 53
3.4.7 模拟调制设置 56
3.4.8 脉冲调制设置 64
3.4.9 相位相干 76
3.4.10 系统复位 77
3.5 SENSE 命令子系统 78
3.5.1 功率计设置 78
4 编程示例 89
4.1 使用 VISA 的编程示例 89
4.1.1 VC++示例 89
4.1.2 VB 示例 96
4.1.3 MATLAB 示例 102
4.1.4 LabVIEW 示例 104
4.2 使用 SOCKET 的编程示例 107
4.2.1 Python 示例 107

1 编程概述

用户可以通过使用 SSG6M80A 多通道微波信号发生器的 USB、LAN 或 USB-GPIB 接口，并结合 NI-VISA 和程序语言，远程控制多通道微波信号发生器。基于 LAN 接口，SSG6M80A 支持 VXI-11、Sockets 和 Telnet 通信协议。本节介绍了如何建立多通道微波信号发生器和计算机之间的通信，同时介绍了如何远程控制多通道微波信号发生器。

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。

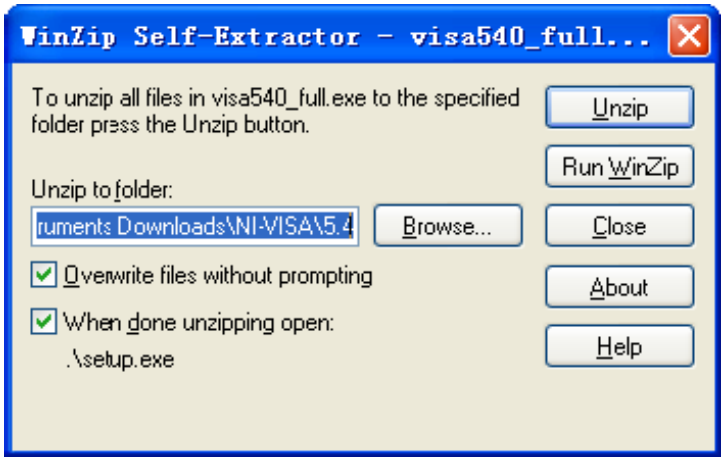
NI-VISA 是用于计算机与设备之间通信的通信库。NI 软件有两种有效 VISA 安装包：完整版和运行引擎版（Run-Time Engine）。运行引擎版本提供 NI 设备驱动程序，例如 USB-TMC，VXI 和 GPIB 等，它主要用于远程控制。完整版本包括运行引擎和 NI MAX 工具，其中 NI MAX 是用于控制设备的用户界面。

您可以在 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。

请参阅以下步骤安装 NI-VISA（示例使用 NI-VISA5.4 完整版）：

a. 下载合适版本的 NI-VISA。

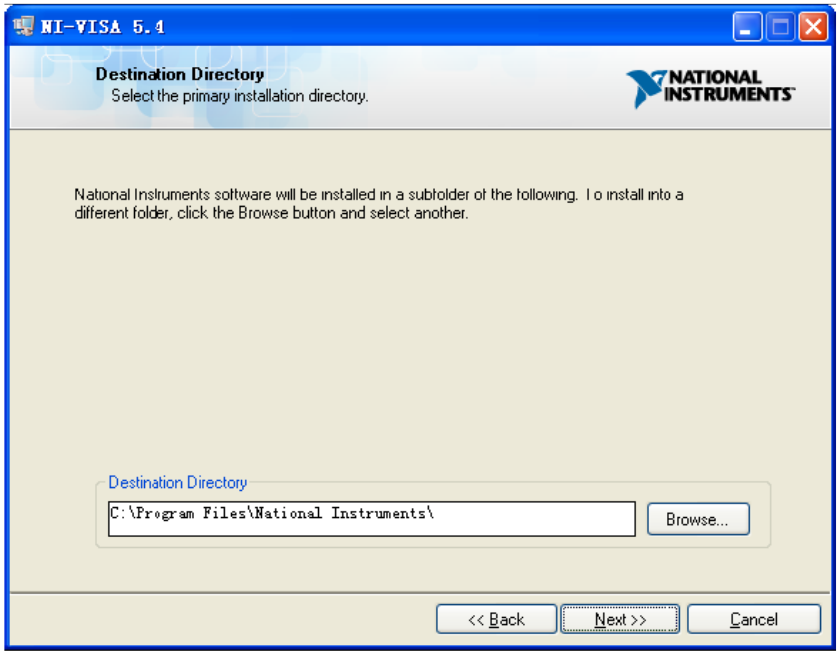
b. 双击 visa540_full.exe，弹出对话框如下：



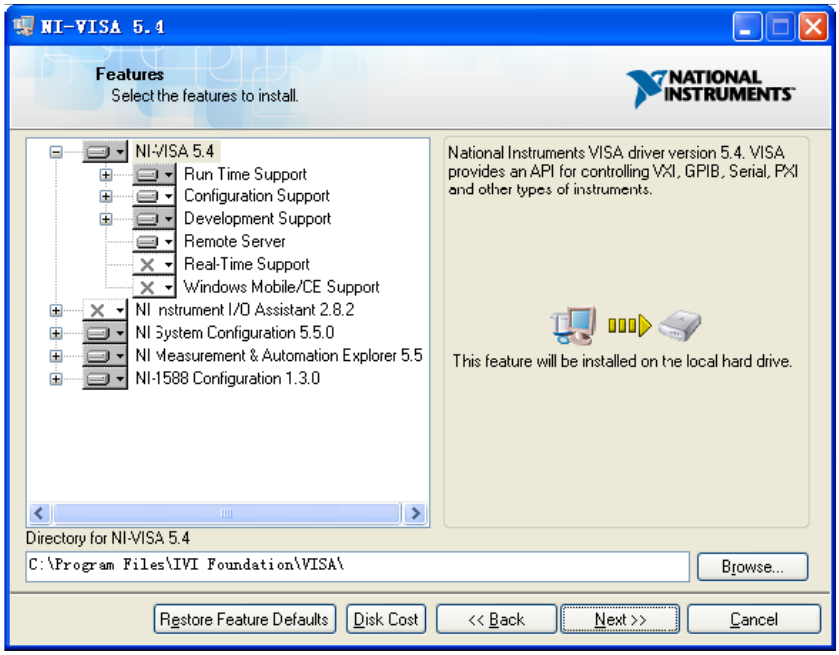
c. 点击 Unzip 解压文件，当解压完成后，安装程序将自动执行。若您的计算机需要安装.NET Framework4，则在安装过程会自动安装.NET Framework4。



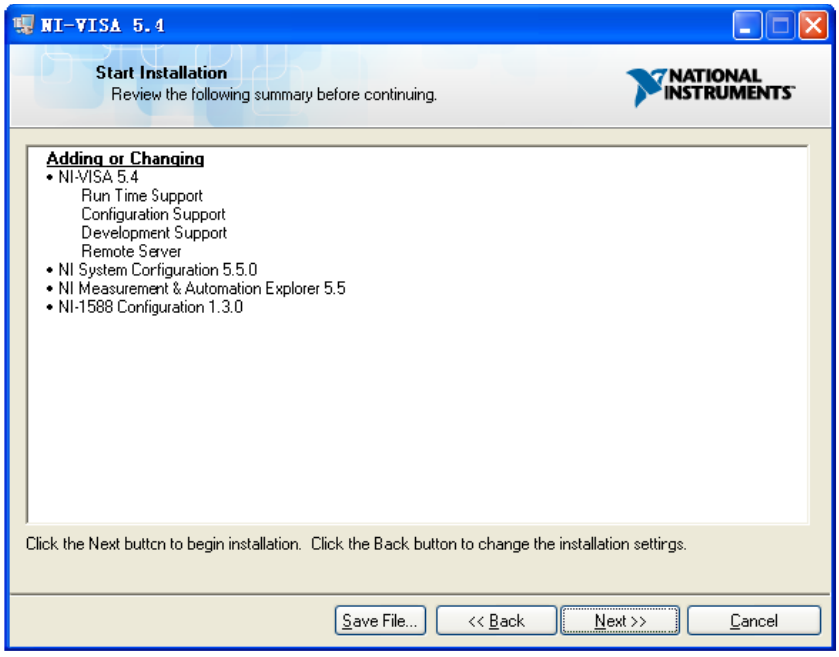
d. NI-VISA 安装对话框如上图所示，点击 Next 开始安装过程。



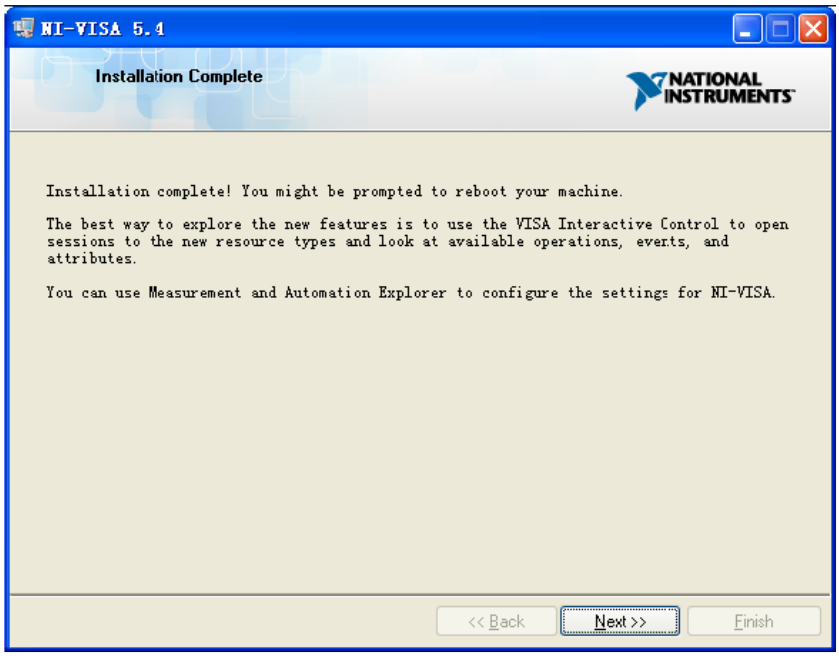
e. 设置安装路径，默认路径为“C:\Program Files\National Instruments\”。您也可以修改安装路径。点击 Next，对话框如下图所示。



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



g. 点击 Next 开始安装：



h. 安装完成后，重启电脑。

1.1.2 连接仪器

多通道微波信号发生器能够通过 USB 接口、LAN 接口或 USB-GPIB 接口连接计算机。

1.1.2.1 使用 USB 接口连接

请参阅以下步骤实现多通道微波信号发生器通过 USB 接口与计算机的连接：

1. 在您的计算机上安装 NI-VISA 以获得 USB-TMC 驱动。
2. 使用 USB A-B 电缆将多通道微波信号发生器的 USB Device 接口连接到计算机的 USB Host 接口。
3. 打开多通道微波信号发生器。

多通道微波信号发生器将被自动检测为新的 USB 设备。

1.1.2.2 使用 LAN 接口连接

请参阅以下步骤实现多通道微波信号发生器通过 LAN 接口与计算机的连接：

1. 在计算机上安装 NI-VISA 以获得 VXI 驱动。
2. 使用网络电缆将多通道微波信号发生器连接到计算机或局域网。
3. 打开多通道微波信号发生器。

4. 按下 HOME → → → 接口 进入 LAN 设置功能菜单。

5. 选择静态或动态的 IP 配置：

   - 动态：当前网络中的 DHCP 服务器将为分析仪自动分配网络参数（IP 地址，子网掩码，网关）。
   - 静态：您可以手动设置 IP 地址，子网掩码，网关方式。



多通道微波信号发生器将被自动或手动检测为新的 LAN 设备。

1.1.2.3 通过 USB-GPIB 接口连接

请参阅以下步骤实现多通道微波信号发生器通过 USB-GPIB 接口与计算机的连接：

1. 在计算机上安装 NI-VISA 的 GPIB 驱动。
2. 使用 SIGLENT USB-GPIB 适配器将多通道微波信号发生器的 USB Host 接口连接到计算机的 GPIB 接口：



3. 打开多通道微波信号发生器。
4. 按下 HOME → → → 接口，向下滚动页面，在 GPIB 地址中输入 GPIB 地址号码。

多通道微波信号发生器将被自动检测为新的 GPIB 设备。

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多通道微波信号发生器。相关内容，请参阅“编程示例”章节中的介绍。

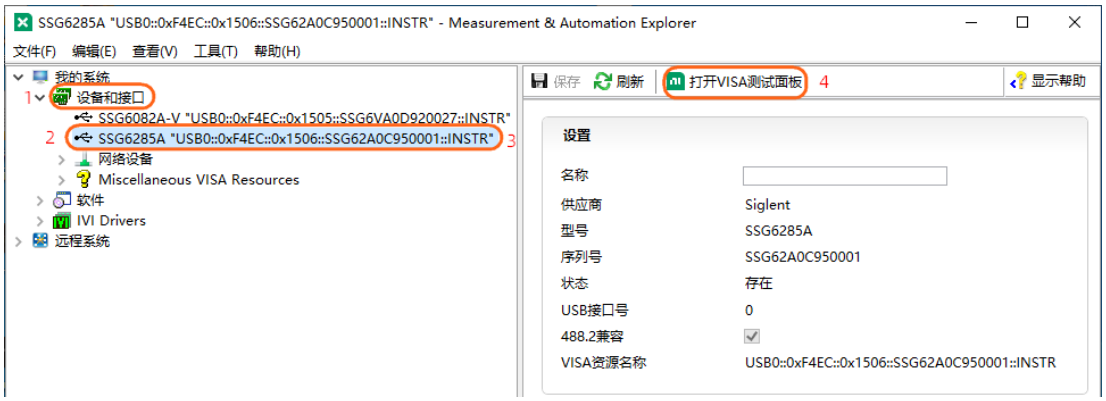
1.2.2 通过 NI-MAX 发送 SCPI 命令

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 命令远程控制多通道微波信号发生器。

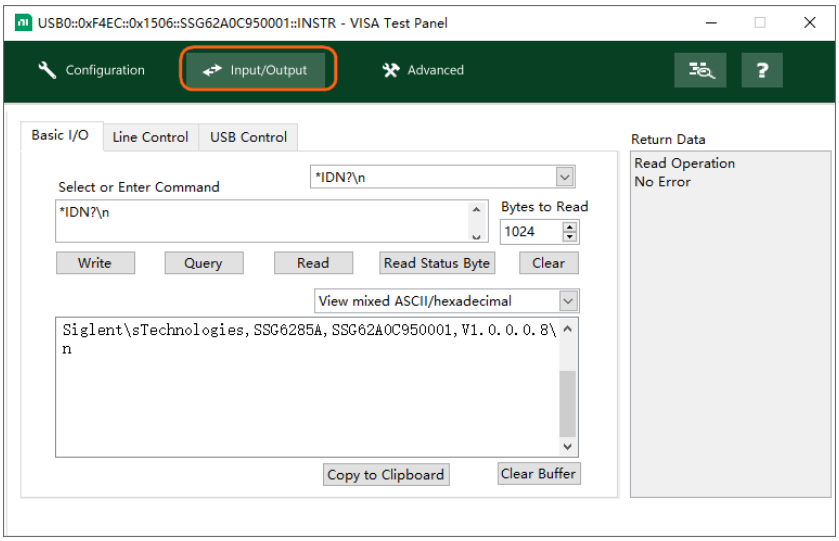
以下介绍在 NI-MAX 中通过 USB、LAN 或 GPIB 接口连接设备并发送向设备发送 SCPI 的步骤。

1.2.2.1 使用 USB 接口

1. 运行 NI-MAX。
2. 点击软件左上角的“设备和接口”。
3. 找到 USBTMC 设备符号 下的多通道微波信号发生器。
4. 选择多通道微波信号发生器设备，并点击“打开 VISA 测试面板”按钮。

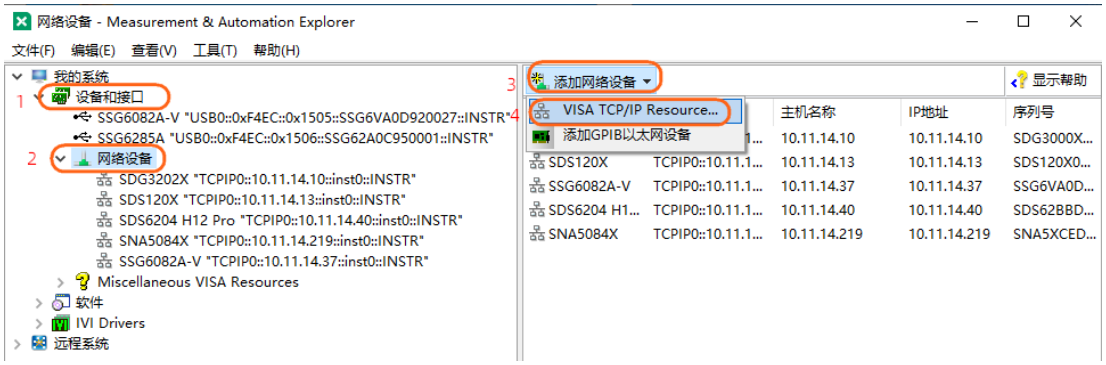


5. 在 VISA 测试面板中选择“Input/Output”页面，此时您可以在输入栏中输入 SCPI 进行写入或查询。如下图点击“Query”按钮查询设备的 IDN：

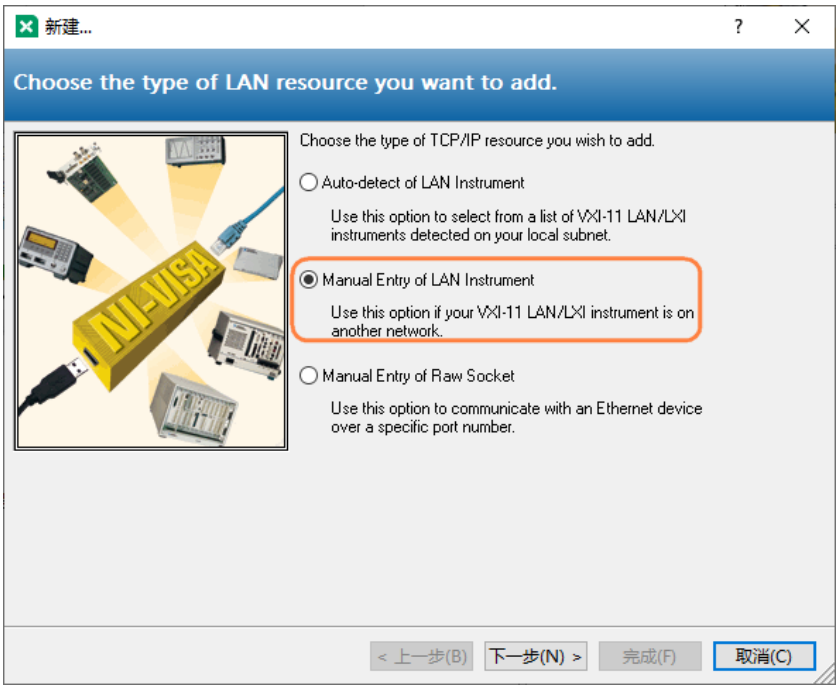


1.2.2.2 使用 LAN 接口

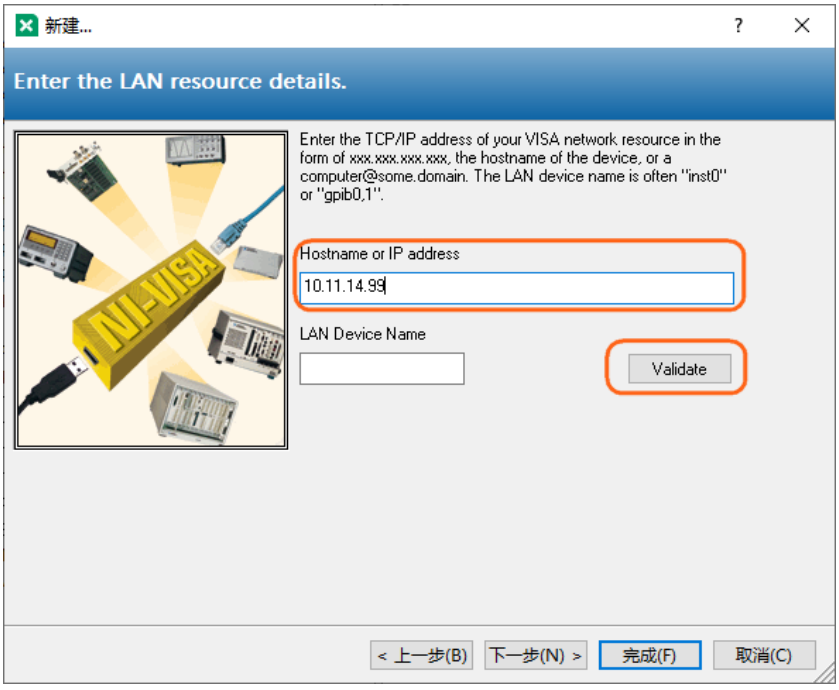
1. 运行 NI-MAX。
2. 点击软件左上角的“设备和接口”→“网络设备”。
3. 点击“添加网络设备”，然后点击“VISA TCP / IP Resource”：



4. 在弹出的“新建...”窗口，选择“Manual Entry of LAN Instrument”，并点击下一步。

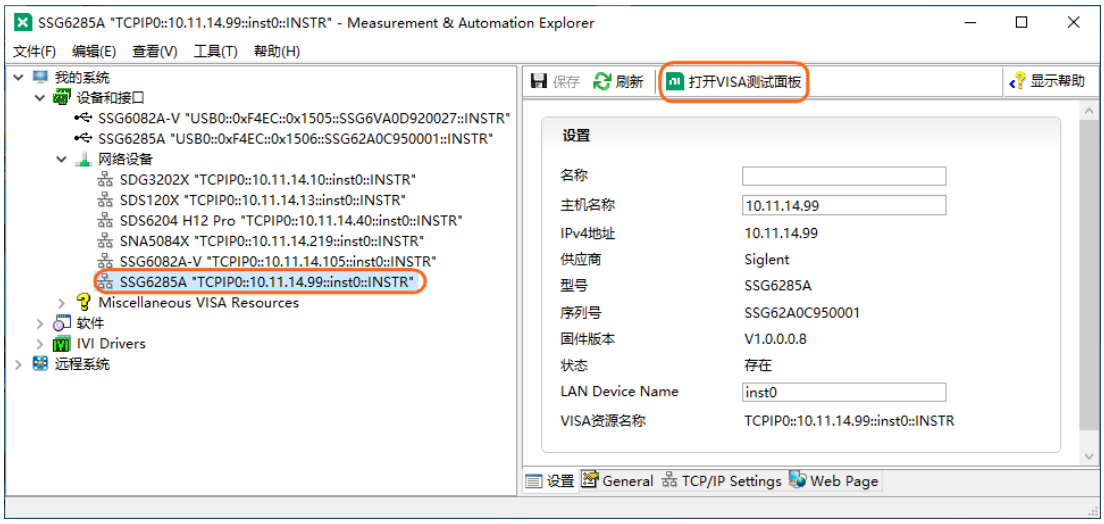


5. 在“Hostname or IP address”中输入多通道微波信号发生器的 IP 地址。您可以点击“Validate”来验证能否通过输入的 IP 来连接设备。

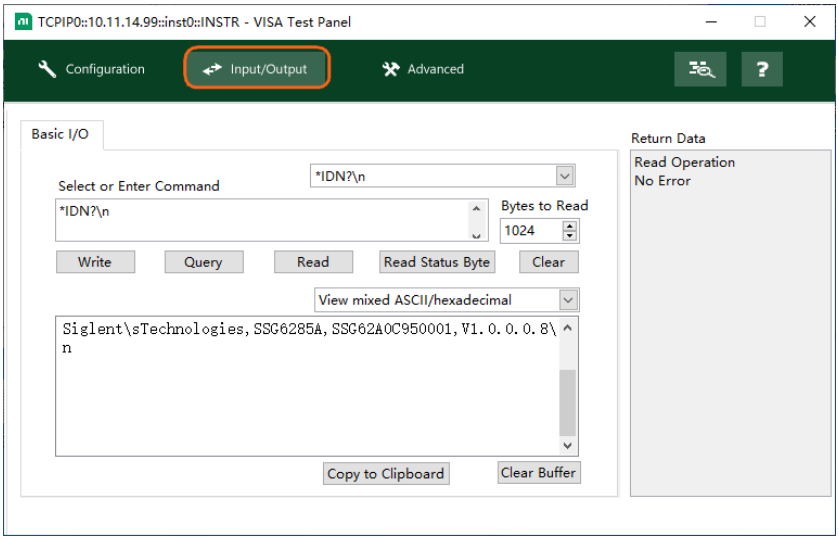


6. 点击“完成”建立连接。

7. 短暂扫描后，多通道微波信号发生器的资源名称应显示在“网络设备”下：



8. 选择多通道微波信号发生器设备，并点击“打开 VISA 测试面板”按钮。
9. 在 VISA 测试面板中选择“Input/Output”页面，此时您可以在输入栏中输入 SCPI 进行写入或查询。如下图点击“Query”按钮查询设备的 IDN：

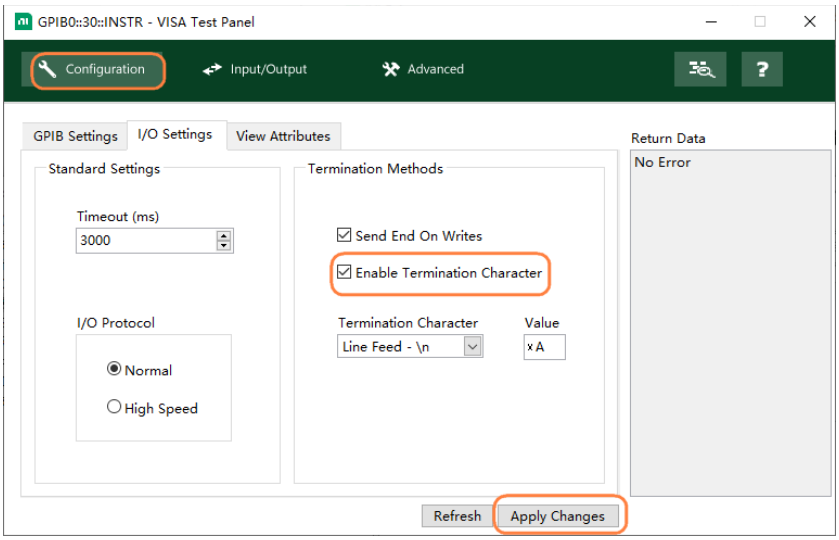


1.2.2.3 使用 USB-GPIB 接口

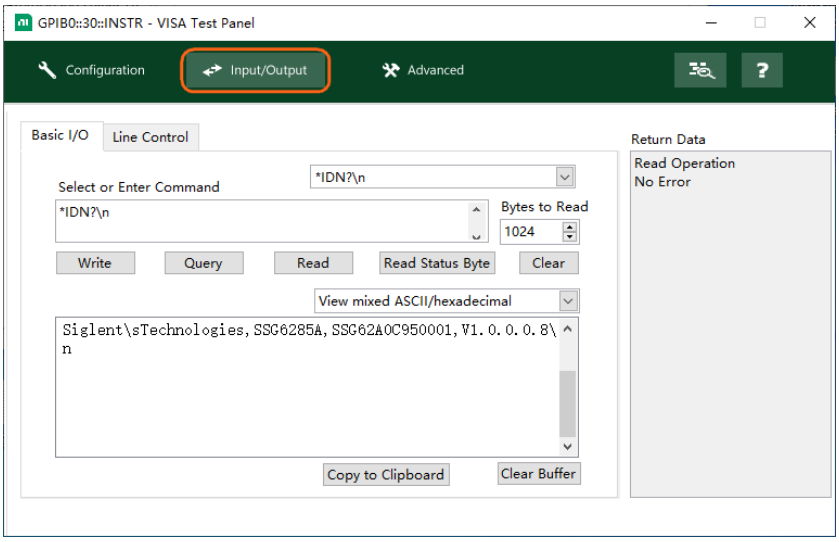
1. 运行 NI-MAX。
2. 点击软件左上角的“设备和接口”。
3. 找到“GPIB-USB-HS”设备符号下的多通道微波信号发生器。
4. 选择多通道微波信号发生器设备，并点击“打开 VISA 测试面板”按钮。



5. 在 VISA 测试面板中选择“Configuration”>“I/O Settings”。勾选启用终止字符选项，然后单击应用更改按钮。



6. 在 VISA 测试面板中选择“Input/Output”页面，此时您可以在输入栏中输入 SCPI 进行写入或查询。如下图点击“Query”按钮查询设备的 IDN：

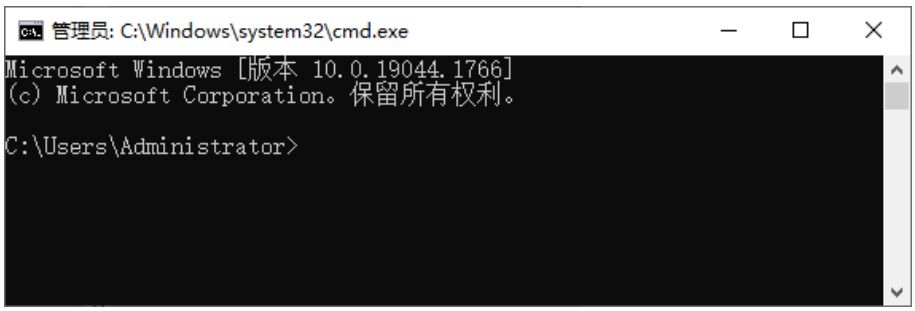


1.2.3 通过 Telnet 发送 SCPI 命令

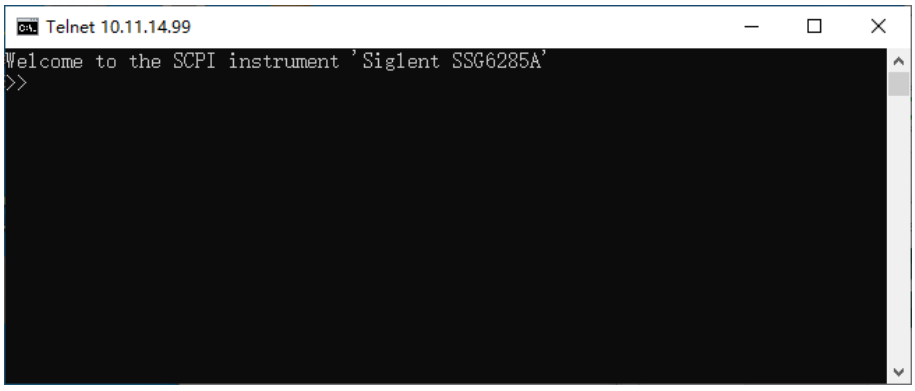
Telnet 提供一种通过 LAN 接口与多通道微波信号发生器通信的方式。Telnet 协议支持从计算机向多通道微波信号发生器发送 SCPI 命令，该方式类似于通过 USB 与多通道微波信号发生器通信。发送和接受信息是交互的，一次只能发送一个命令。Windows 操作系统使用命令提示符样式接口作为 Telnet 客户端。

步骤如下：

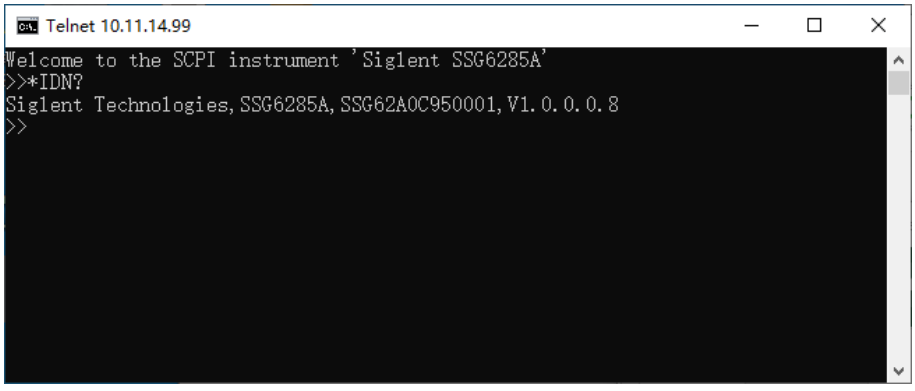
1. 在计算机桌面，点击开始，然后右键选择运行。在运行窗口中输入 cmd 并点击确定，此时打开了命令提示符窗口。



2. 在命令提示符窗口，输入 telnet <ip address> 5024 并回车，将弹出可与仪器会话的 Telnet 窗口：



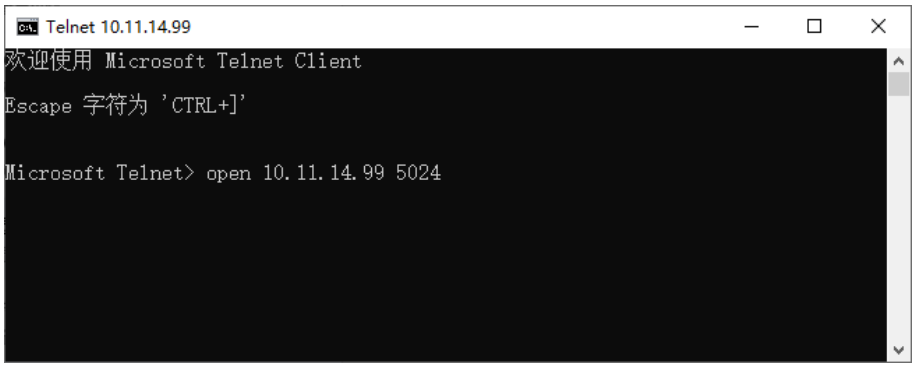
3. 在“>>”提示符后，您可以输入 SCPI 命令远程控制多通道微波信号源。例如输入*IDN?，该命令将返回公司名、机器型号、序列号和固件版本号。



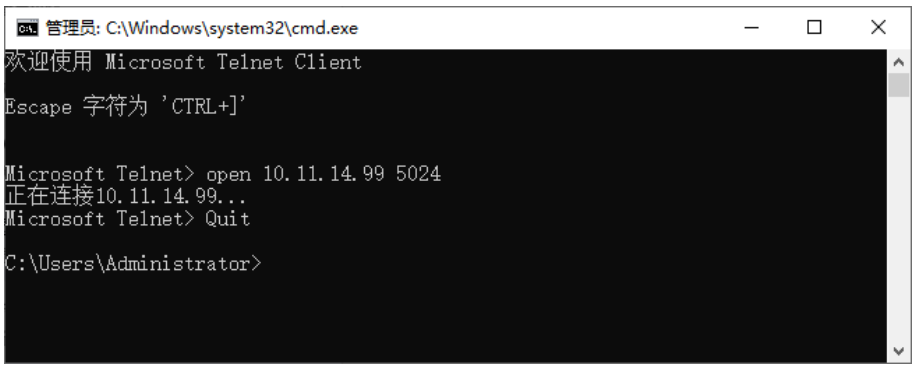
4. 同时按下 Ctrl+] 键将退出与仪器的 SCPI 会话：



5. 想要再次进入与仪器的 SCPI 会话，您可以输入 open <ip Address> 5024 并按回车：



6. 想要关闭 Telnet 窗口，输入 Quit 并按回车：



1.2.4 通过 Socket 发送 SCPI 命令

Socket 接口可以在不安装其他库的情况下通过 LAN 接口控制多通道微波信号发生器，它可以减少编程的复杂度。详细信息请查阅“编程示例”的“使用 Socket 的编程示例”章节。

SOCKET 地址	IP 地址+接口号
IP 地址	SSG IP 地址
接口号	5025

2 SCPI 语言简介

2.1 命令格式

SCPI 命令为树状层次结构，包括多个子系统。每个子系统由一个根关键字和一个或数个层次关键字构成。命令行通常以冒号“:”开始，关键字之间用冒号“:”分隔。关键字后面跟随可选的参数设置，命令和参数以“空格”分开，多个参数的，参数之间用逗号“,”分隔。命令行后面添加问号“?”，表示对此功能进行查询。

例如：

:SOURce:FREQuency <freq>

:SOURce:FREQuency?

SOURce 是命令的根关键字，FREQuency 是第二级关键字。命令行以冒号“:”开始，同时冒号将各级关键字分开。<freq>表示可设置的参数，命令:SOURce:FREQuency 和参数<freq>之间用“空格”分开。问号“?”表示查询，仪器收到查询命令后将返回一个响应字符串。

2.2 命令符号

下面是用在命令中的符号：

1. **三角括号 < >**

   三角括号< >中的内容是命令参数，必须用一个有效值来替换。例如：

   POWer:SPC:TARGet <power>命令，您可以发送为 POWer:SPC:TARGet 0。

2. **方括号 [ ]**

   方括号中的内容（命令关键字或默认参数）是可省略的。如果省略方括号中的关键字，该命令仍可以产生相同的效果。如果省略方括号中的默认参数，该默认参数仍生效。

3. **竖线 |**

   竖线用于分隔多个参数枚举值，发送命令时必须选择其中一个参数。例如：

   在[:SOURce<ch>]:AM:STATe OFF|ON|0|1 命令中，可选择的命令参数为“OFF”、“ON”、“0”或“1”。

4. **大括号 { }**

   大括号中的参数是可选项，可以不设置，也可以设置一次或多次。例如：

   :CALCulate:LLINe[1]|2:DATA <x-axis>,<ampl>{,<x-axis>,<ampl>}，大括号中的{,<x-axis>,<ampl>}可以省略，也可以设置一对或多对频率、幅度参数。

2.3 参数类型

本手册介绍的命令中所含的参数可以分为以下 6 种类型：布尔型、枚举、整型、浮点型、ASCII 字符串和离散型。

1. **布尔型**

   参数取值为“OFF”、“ON”、“0”或“1”。例如：

   [:SOURce<ch>]:FM:STATe OFF|ON|0|1

2. **枚举**

   参数取值为所列举的值。例如：

   [:SOURce<ch>]:SWEep:STATe OFF|FREQuency|LEVel|LEV_FREQ

   参数为“OFF”、“FREQuency”、“LEVel”或“LEV_FREQ”。

3. **整型**

   除非另有说明，参数在有效值范围内可以取任意整数值。注意：此时请不要设置参数为小数格式，否则将出现异常。例如：

   [:SOURce<ch>]:SWEep:STEP:POINts <value>

   参数<value>可取 2 到 65535 范围内的任一整数。

4. **浮点型**

   参数可根据精度要求在有效范围内取任意值（通常默认精度为小数点后九位）。例如：

   [:SOURce<ch>]:POWer:OFFSet <value>

   参数<value>可取-100 到 100 之间的实数。

5. **字符串**

   参数取值为 ASCII 字符的组合。例如：

   :SYSTem:COMMunicate:LAN:IPADdress <“xxx.xxx.xxx.xxx”>

   参数为设定的 IP 地址格式字符串。

6. **离散型**

   参数只能取指定的数值，并且这些数值不是连续的。例如：

   [:SENSe]:BWIDth:VIDeo:RATio <number>

   参数<number>只能取值为 0.001、0.003、0.01、0.03、0.1、0.3、1.0、3.0、10.0、30.0、100.0、300.0、1000.0。

2.4 命令缩写

所有 SCPI 命令都对大小写不敏感。您可以以全大写或全小写的形式输入完整的命令。您还可以使用缩写，在这种情况下，缩写命令必须包含命令格式中的所有大写字母。

例如：

:CORRection:FLATness:COUNt?

您可以通过以下任意一种书写方式发送：

:CORRection:FLATness:COUNt?

:CORRECTION:FLATNESS:COUNT?

:correction:flatness:count?

您也可以缩写为：

:CORR:FLAT:COUN?

3 命令系统

本章主要介绍鼎阳 SSG6M80A 的 SCPI 命令系统。

3.1 IEEE 通用命令子系统

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 设备信息查询（*IDN?）

命令格式	*IDN?
功能描述	用于获取设备信息字符串。字符串内容包括厂商，设备型号，设备序列号和软件版本号。
对应菜单	无
示例	**IDN?* 返回值： *Siglent\sTechnologies,SSG6285A,SSG62A0C950001,V1.0.0.0.8\n*

3.1.2 复位（*RST）

命令格式	*RST
功能描述	重新恢复设备状态为初始状态。*RST 相当于执行两个命令：“:SOURce:PRESet”和“*CLS”。
对应菜单	无
示例	**RST*

3.1.3 清除状态（*CLS）

命令格式	*CLS
功能描述	将所有事件寄存器的值清零，同时清空错误列表。
对应菜单	无
示例	**CLS*

3.1.4 标准事件状态使能（*ESE）

命令格式	*ESE <number> *ESE?
功能描述	设定标准事件状态寄存器的使能值。 查询标准事件状态寄存器的使能值。
参数类型	整型
参数范围	0 ~ 255
对应菜单	无
示例	**ESE 16* **ESE?* 返回值： *16\n*

3.1.5 标准事件状态寄存器询问（*ESR?）

命令格式	*ESR?
功能描述	询问及清除标准事件状态寄存器的事件值。
对应菜单	无
示例	**ESR?* 返回值： *0\n*

3.1.6 操作完成（*OPC）

命令格式	*OPC *OPC?
功能描述	所有操作结束后，在标准事件状态寄存器中设置比特 0 为 1。 查询当前操作是否完成。
对应菜单	无
示例	**OPC* **OPC?* 返回值： *1\n*

3.1.7 允许服务请求（*SRE）

命令格式	*SRE <integer> *SRE?
功能描述	设置状态字节寄存器的使能值。 查询状态字节寄存器的使能值。
参数类型	整型
参数范围	0 ~ 255
对应菜单	无
示例	*SRE 24 *SRE? 返回值： 24\n

3.1.8 状态字节询问（*STB?）

命令格式	*STB?
功能描述	查询状态字节寄存器的事件值。
对应菜单	无
示例	*STB? 返回值： 72\n

3.1.9 等待继续（*WAI）

命令格式	*WAI
功能描述	等待所有未完成操作完成之后，再执行任何其他命令。
对应菜单	无
示例	*WAI

3.1.10 自测试询问（*TST?）

命令格式	*TST?
功能描述	查询仪器自检结果。
对应菜单	无
示例	*TST? 返回值： 0\n

3.2 SYSTem 命令子系统

3.2.1 系统配置

3.2.1.1 系统时间（:SYSTem:TIME）

命令格式	:SYSTem:TIME <hhmmss> :SYSTem:TIME?
功能描述	设置系统时间。 查询系统时间。
参数类型	字符串
参数范围	小时（0~23），分（0~59），秒（0~59）
返回值	字符串
默认值	无
对应菜单	HOME >  >  > 设置 > 时间设置
示例	*:SYSTem:TIME 182559* *:SYSTem:TIME?* 返回值: *182613\n*

3.2.1.2 系统日期（:SYSTem:DATE）

命令格式	:SYSTem:DATE <yyyymmdd> :SYSTem:DATE?
功能描述	设置系统日期。 查询系统日期。
参数类型	字符串
参数范围	年（四位长度），月（1~12），日（1~31）
返回值	字符串
默认值	无
对应菜单	HOME >  >  > 设置 > 时间设置
示例	*:SYSTem:DATE 20050101* *:SYSTem:DATE?* 返回值: *20050101\n*

3.2.1.3 IP 地址（:SYSTem:COMMunicate:LAN:IPADdress）

命令格式	:SYSTem:COMMunicate:LAN:IPADdress <“xxx.xxx.xxx.xxx”> :SYSTem:COMMunicate:LAN:IPADdress?
功能描述	当 IP 分配类型为静态时，设置仪器的 IP 地址。 查询仪器的 IP 地址。
参数类型	字符串
参数范围	必须符合 IP 设置规范（0-255:0-255:0-255:0-255）
返回值	字符串
默认值	无
对应菜单	HOME > > > 接口 > LAN 设置 > IP 地址
示例	*:SYSTem:COMMunicate:LAN:IPADdress “192.168.1.12”* *:SYSTem:COMMunicate:LAN:IPADdress?* 返回值： *“192.168.1.12”\n*

3.2.1.4 网关（:SYSTem:COMMunicate:LAN:GATeway）

命令格式	:SYSTem:COMMunicate:LAN:GATeway <“xxx.xxx.xxx.xxx”> :SYSTem:COMMunicate:LAN:GATeway?
功能描述	设置仪器的网关。 查询仪器的网关。
参数类型	字符串
参数范围	必须符合 IP 地址的网卡规范（0-255:0-255:0-255:0-255）
返回值	字符串
默认值	无
对应菜单	HOME > > > 接口 > LAN 设置 > 网关
示例	*:SYSTem:COMMunicate:LAN:GATeway “192.168.1.1”* *:SYSTem:COMMunicate:LAN:GATeway?* 返回值： *“192.168.1.1”\n*

3.2.1.5 子网掩码（:SYSTem:COMMunicate:LAN:SMASk）

命令格式	:SYSTem:COMMunicate:LAN:SMASk <“xxx.xxx.xxx.xxx”> :SYSTem:COMMunicate:LAN:SMASk?

功能描述	设置仪器的子网掩码。 查询仪器的子网掩码。
参数类型	字符串
参数范围	必须符合 IP 地址的网卡规范（0-255:0-255:0-255:0-255）
返回值	字符串
默认值	无
对应菜单	HOME >  >  > 接口 > LAN 设置 > 子网掩码
示例	*:SYSTem:COMMunicate:LAN:SMASk "255.255.255.0"* *:SYSTem:COMMunicate:LAN:SMASk?* 返回值： *"255.255.255.0"\n*

3.2.1.6 IP 分配类型（:SYSTem:COMMunicate:LAN:TYPE）

命令格式	:SYSTem:COMMunicate:LAN:TYPE STATIC|DHCP :SYSTem:COMMunicate:LAN:TYPE?
功能描述	设置仪器的 IP 分配类型。 查询仪器的 IP 分配类型。
参数类型	枚举
参数范围	STATIC：静态分配， DHCP：动态分配。
返回值	枚举
默认值	无
对应菜单	HOME >  >  > 接口 > LAN 设置 > DHCP 状态
示例	*:SYSTem:COMMunicate:LAN:TYPE STATIC* *:SYSTem:COMMunicate:LAN:TYPE?* 返回值： *STATIC\n*

3.2.1.7 系统语言（:SYSTem:LANGuage）

命令格式	:SYSTem:LANGuage CHINese|ENGLish :SYSTem:LANGuage?
功能描述	设置系统语言。 查询系统语言。

参数类型	枚举
参数范围	CHINese：中文， ENGLish：英文。
返回值	枚举
默认值	无
对应菜单	HOME > > > 设置> Language
示例	*:SYSTem:LANGuage CHINese* *:SYSTem:LANGuage?* 返回值： *CHINese\n*

3.2.1.8 屏保（:SYSTem:SCReen:SAVer）

命令格式	:SYSTem:SCReen:SAVer OFF|10S|1MIN|5MIN|15MIN|30MIN|1HOUR|2HOUR :SYSTem:SCReen:SAVer?
功能描述	设置系统屏保类型。 查询系统屏保类型。
参数类型	枚举
参数范围	OFF：关闭屏保， 10S|1MIN|5MIN|15MIN|30MIN|1HOUR|2HOUR：设置不同屏保时间。
返回值	枚举
默认值	OFF
对应菜单	HOME > > > 设置 > 屏保
示例	*:SYSTem:SCReen:SAVer 30MIN* *:SYSTem:SCReen:SAVer?* 返回值： *30MIN\n*

3.2.1.9 蜂鸣器（:SYSTem:ALARm）

命令格式	:SYSTem:ALARm ON|OFF|1|0 :SYSTem:ALARm?
功能描述	设置系统蜂鸣器开关状态。 查询系统蜂鸣器开关状态。
参数类型	布尔型
参数范围	ON|OFF|1|0

返回值	1|0
默认值	无
对应菜单	HOME >  >  > 设置 > 蜂鸣器
示例	*:SYSTem:ALARm ON* *:SYSTem:ALARm?* 返回值: *1\n*

3.2.1.10 启动设置（:SYSTem:PON:TYPE）

命令格式	:SYSTem:PON:TYPE DFT|LAST :SYSTem:PON:TYPE?
功能描述	设置系统的启动类型。 查询系统的启动类型。
参数类型	枚举
参数范围	DFT：默认， LAST：上次。
返回值	枚举
默认值	无
对应菜单	HOME >  >  > 设置 > 启动设置
示例	*:SYSTem:PON:TYPE LAST* *:SYSTem:PON:TYPE?* 返回值: *LAST\n*

3.2.1.11 上电开机（:SYSTem:POWeron:TYPE）

命令格式	:SYSTem:POWeron:TYPE ON|OFF|1|0 :SYSTem:POWeron:TYPE?
功能描述	设置系统是否上电开机。 查询系统上电开机状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0

对应菜单	HOME >  >  > 设置 > 上电开机
示例	*:SYSTem:POWeron:TYPE ON* *:SYSTem:POWeron:TYPE?* 返回值: *1\n*

3.2.1.12 参考校正（:SYSTem:REF:DAC:STAT）

命令格式	:SYSTem:REF:DAC:STAT ON|OFF|1|0 :SYSTem:REF:DAC:STAT?
功能描述	设置系统的参考校正状态。 查询系统的参考校正状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	HOME >  >  > 设置 > 参考校正
示例	*:SYSTem:REF:DAC:STAT ON* *:SYSTem:REF:DAC:STAT?* 返回值: *1\n*

3.2.1.13 参考振荡器码字（:SYSTem:REF:DAC）

命令格式	:SYSTem:REF:DAC <value> :SYSTem:REF:DAC?
功能描述	设置系统的参考振荡器码字。 查询系统的参考振荡器码字。
参数类型	整型
参数范围	0 ~ 65535
返回值	整型
默认值	无
对应菜单	HOME >  >  > 设置 > 参考校正 > 参考振荡器码字
示例	*:SYSTem:REF:DAC 43000* *:SYSTem:REF:DAC?*

	返回值: *43000\n*

3.2.1.14 保存参考振荡器设置（:SYSTem:REF:DAC:SAVE）

命令格式	:SYSTem:REF:DAC:SAVE <"file_name">
功能描述	把系统的参考振荡器码字保存进文件。
参数类型	字符串
参数范围	无
对应菜单	HOME >  >  > 设置 > 参考校正 > 保存参考振荡器设置
示例	*:SYSTem:REF:DAC:SAVE "U-disk3/test.dac"*

3.2.1.15 加载参考振荡器设置（:SYSTem:REF:DAC:LOAD）

命令格式	:SYSTem:REF:DAC:LOAD <"file_name">
功能描述	从文件中加载参考振荡器码字。
参数类型	字符串
参数范围	无
对应菜单	HOME >  >  > 设置 > 参考校正 > 加载参考振荡器设置
示例	*:SYSTem:REF:DAC:LOAD "U-disk3/test.dac"*

3.2.1.16 重置参考振荡器码字（:SYSTem:REF:DAC:DEFault）

命令格式	:SYSTem:REF:DAC:DEFault
功能描述	恢复参考振荡器码字为默认值。
对应菜单	HOME >  >  > 设置 > 参考校正 > 重置为默认值
示例	*:SYSTem:REF:DAC:DEFault*

3.2.1.17 GPIB 地址（:SYSTem:GPIB）

命令格式	:SYSTem:GPIB <value> :SYSTem:GPIB?

功能描述	设置系统的 GPIB 地址。 查询系统的 GPIB 地址。
参数类型	整型
参数范围	1 ~ 30
返回值	整型
默认值	18
对应菜单	HOME >  >  > 接口 > GPIB 设置 > GPIB 地址
示例	*:SYSTem:GPIB 10* *:SYSTem:GPIB?* 返回值： *10\n*

3.2.1.18 退出远程模式（:SYSTem:REMote 0）

命令格式	:SYSTem:REMote 0
功能描述	退出系统的远程模式。
对应菜单	ESC
示例	*:SYSTem:REMote 0*

3.2.1.19 查询仪器时钟参考源（:SYStem:CLOCk?）

命令格式	:SYStem:CLOCk?
功能描述	查询仪器使用的时钟参考源是内部还是外部。
返回值	EXTERNAL：仪器使用外部时钟参考， INTERNAL：仪器使用内部时钟参考。
默认值	无
对应菜单	EXTERNAL：HOME > EXT REF 标识， INTERNAL：无标识。
示例	*:SYStem:CLOCk?* 返回值： *EXTERNAL\n*

3.2.2 系统复位

3.2.2.1 系统复位（:SYSTem:PRESet）

命令格式	:SYSTem:PRESet
功能描述	根据复位类型，复位仪器的参数配置。
对应菜单	HOME >  >  > 复位，或 PRESET
示例	复位仪器为默认配置： *:SYSTem:PRESet:TYPE DFT* *:SYSTem:PRESet* 复位仪器为当前配置： *:SYSTem:PRESet:TYPE USER* *:SYSTem:PRESet:SAVE* *:SYSTem:PRESet* 复位仪器为文件中的配置： *:SYSTem:PRESet:TYPE USER* *:SYSTem:PRESet:PATH "Local/test.xml"* *:SYSTem:PRESet*

3.2.2.2 保存系统配置（:SYSTem:PRESet:SAVE）

命令格式	:SYSTem:PRESet:SAVE
功能描述	保存当前的系统配置。
对应菜单	无
示例	复位仪器为当前配置： *:SYSTem:PRESet:TYPE USER* *:SYSTem:PRESet:SAVE* *:SYSTem:PRESet*

3.2.2.3 保存系统配置（:SYSTem:PRESet:PATH）

命令格式	:SYSTem:PRESet:PATH <"file_name">
功能描述	把当前的系统配置保存进文件。
参数类型	字符串
参数范围	无
对应菜单	HOME >  >  > 设置 > 复位类型（用户） > 保存

示例	*:SYSTem:PRESet:PATH "Local/test.xml"* *:SYSTem:PRESet:PATH "U-disk1/test.xml"*

3.2.2.4 复位类型（:SYSTem:PRESet:TYPE）

命令格式	:SYSTem:PRESet:TYPE DFT|USER :SYSTem:PRESet:TYPE?
功能描述	设置系统的复位类型。 查询系统的复位类型。
参数类型	枚举
参数范围	DFT：默认， USER：用户。
返回值	枚举
默认值	DFT
对应菜单	HOME >  >  > 设置 > 复位类型
示例	*:SYSTem:PRESet:TYPE DFT* *:SYSTem:PRESet:TYPE?* 返回值： *DFT\n*

3.2.2.5 恢复出厂设置（:SYSTem:FDEFault）

命令格式	:SYSTem:FDEFault
功能描述	将仪器状态恢复为出厂设置。
对应菜单	HOME >  >  > 设置 > 恢复出厂设置
示例	*:SYSTem:FDEFault*

3.2.2.6 复位清除（:SYSTem:RESet:CLEar）

命令格式	:SYSTem:RESet:CLEar
功能描述	将仪器状态恢复为出厂设置，同时清空 Local 文件夹下的用户文件。
对应菜单	HOME >  >  > 设置 > 复位清除
示例	*:SYSTem:RESet:CLEar*

3.3 OUTPut 命令子系统

3.3.1 RF 端口输出（[:SOURce<ch>]:OUTPut[:STATe]）

命令格式	[:SOURce<ch>]:OUTPut[:STATe] ON|OFF|1|0 [:SOURce<ch>]:OUTPut[:STATe]?
功能描述	设置通道 ch 的射频输出状态。 查询通道 ch 的射频输出状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
等效命令	[:SOURce<ch>]:OUTPut ON|OFF|1|0
对应菜单	RF ON/OFF
示例	*:SOURce1:OUTPut ON* *:SOURce1:OUTPut?* 返回值： *1\n*

3.3.2 模拟调制开关状态（[:SOURce<ch>]:OUTPut:MODulation[:STATe]）

命令格式	[:SOURce<ch>]:OUTPut:MODulation[:STATe] ON|OFF|1|0 [:SOURce<ch>]:OUTPut:MODulation[:STATe]?
功能描述	设置通道 ch 的模拟调制的开关状态。 查询通道 ch 的模拟调制的开关状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
等效命令	:MODulation ON|OFF|1|0 :MODulation?
对应菜单	Analog Mod > On
示例	*:SOURce1:OUTPut:MODulation ON* *:SOURce1:OUTPut:MODulation?* 返回值： *1\n*

3.4 SOURce 命令子系统

3.4.1 RF 端口输出（[:SOURce<ch>]:OUTPut）

命令格式	[:SOURce<ch>]:OUTPut ON|OFF|1|0
功能描述	设置通道 ch 的射频输出状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
等效命令	:OUTPut[:STATe] ON|OFF|1|0
对应菜单	RF ON/OFF
示例	*:SOURce1:OUTPut ON*

3.4.2 软件触发（[:SOURce<ch>]:*TRG）

命令格式	[:SOURce<ch>]:*TRG
功能描述	当触发源是总线（涉及的触发功能有扫描触发、扫描点触发、LF 扫描触发、脉冲调制触发等）时，执行通道 ch 的软件触发。
对应菜单	TRIG
示例	*:SOURce1:*TRG*

3.4.3 频率设置

3.4.3.1 显示频率（[:SOURce<ch>]:FREQuency:DISPlay）

命令格式	[:SOURce<ch>]:FREQuency:DISPlay <freq> [:SOURce<ch>]:FREQuency:DISPlay?
功能描述	设置屏幕上方参数栏的通道 ch 的频率显示值。 查询屏幕上方参数栏的通道 ch 的频率显示值。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
返回值	浮点型，单位：Hz
默认值	最大频率值
对应菜单	频率
示例	*:SOURce1:FREQuency:DISPlay 2 MHz* *:SOURce1:FREQuency:DISPlay?* 返回值： *2000000\n*

3.4.3.2 频率（[:SOURce<ch>]:FREQuency）

命令格式	[:SOURce<ch>]:FREQuency <freq> [:SOURce<ch>]:FREQuency?
功能描述	设置通道 ch 的 RF 输出信号的频率。 查询通道 ch 的 RF 输出信号的频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
返回值	浮点型，单位：Hz
默认值	最大频率值
对应菜单	RF > 频率 > 频率
示例	*:SOURce1:FREQuency 2 MHz* *:SOURce1:FREQuency?* 返回值： *2000000\n*

3.4.3.3 频率偏移（[:SOURce<ch>]:FREQuency:OFFSet）

命令格式	[:SOURce<ch>]:FREQuency:OFFSet <freq> [:SOURce<ch>]:FREQuency:OFFSet?
功能描述	设置通道 ch 的 RF 输出的频率偏移。 查询通道 ch 的 RF 输出的频率偏移。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	-200 GHz ~ 200 GHz
返回值	浮点型，单位：Hz
默认值	0 Hz
对应菜单	RF > 频率 > 频率偏移
示例	*:SOURce1:FREQuency:OFFSet -2 MHz* *:SOURce1:FREQuency:OFFSet?* 返回值： *-2000000\n*

3.4.3.4 相位偏移（[:SOURce<ch>]:PHASe）

命令格式	[:SOURce<ch>]:PHASe <phase> [:SOURce<ch>]:PHASe?

功能描述	设置通道 ch 的 RF 信号的相位。 查询通道 ch 的 RF 信号的相位。
参数类型	浮点型，单位：度（°）
参数范围	-360 ~ 360
返回值	浮点型，单位：度（°）
默认值	0
对应菜单	RF > 频率 > 相位偏移
示例	*:SOURce1:PHASe 20* *:SOURce1:PHASe?* 返回值： *20\n*

3.4.3.5 重置相位（[:SOURce<ch>]:PHASe:RESet）

命令格式	[:SOURce<ch>]:PHASe:RESet
功能描述	将通道 ch 的 RF 信号的相位偏移显示值重置为 0。
等效命令	[:SOURce<ch>]:PHASe:REF
对应菜单	RF > 频率 > 重置相位差显示值
示例	*SOURce1:PHASe:RESet*

3.4.3.6 重置相位（[:SOURce<ch>]:PHASe:REF）

命令格式	[:SOURce<ch>]:PHASe:REF
功能描述	将通道 ch 的 RF 信号的相位偏移显示值重置为 0。
等效命令	[:SOURce<ch>]:PHASe:RESet
对应菜单	RF > 频率 > 重置相位差显示值
示例	*SOURce1:PHASe:REF*

3.4.3.7 相位记忆（[:SOURce<ch>]:PHASe:MEMory）

命令格式	[:SOURce<ch>]:PHASe:MEMory ON|OFF|1|0 [:SOURce<ch>]:PHASe:MEMory?
功能描述	设置通道 ch 的 RF 相位记忆功能开关。 查询通道 ch 的 RF 相位记忆功能开关。
参数类型	布尔型，

参数范围	ON|OFF|1|0
返回值	ON|OFF|1|0
默认值	0
对应菜单	RF > 频率 > 相位记忆
示例	*:SOURce1:PHASe:MEMory 1* *:SOURce1:PHASe:MEMory?* 返回值: *1\n*

3.4.3.8 相位记忆触发（[:SOURce<ch>]:PHASe:MEMory:TRIG）

命令格式	[:SOURce<ch>]:PHASe:MEMory:TRIG
功能描述	触发通道 ch 的 RF 相位记忆。
参数类型	无
参数范围	ON|OFF|1|0
返回值	ON|OFF|1|0
默认值	无
对应菜单	RF > 频率 > 相位记忆触发
示例	*:SOURce1:PHASe:MEMory:TRIG*

3.4.3.9 相位记忆参数查询（[:SOURce<ch>]:PHASe:MEMory:PARAM?）

命令格式	[:SOURce<ch>]:PHASe:MEMory:PARAM?
功能描述	查询通道 ch 的触发相位记忆时存储的参数
返回值	字符串 格式为："有效情况","Freq: <频率值>\nPhase: <相位值>" Index：整型，扫描点索引， Freq：浮点型，扫描点频率，单位：Hz， Level：浮点型，扫描点幅度，单位：dBm， Time：浮点型，扫描点驻留时间，单位：s。
默认值	"Invalid","Freq:\s-1\nPhase:\s0"\n
对应菜单	RF > 频率 > 相位记忆触发
示例	*:SOURce1:PHASe:MEMory 1* *:SOURce1:PHASe:MEMory:TRIG* *:PHASe:MEMory:PARAM?*

	返回值： *"Valid","Freq: 2e+10\nPhase: 0"\n*

3.4.4 幅度设置

3.4.4.1 显示幅度（[:SOURce<ch>]:POWer:POWer）

命令格式	[:SOURce<ch>]:POWer:POWer <power> [:SOURce<ch>]:POWer:POWer?
功能描述	设置屏幕上方参数栏的通道 ch 的幅度显示值。 查询屏幕上方参数栏的通道 ch 的幅度显示值。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
返回值	浮点型，单位：dBm
默认值	对应最大输出频率的最小输出幅度
对应菜单	幅度
示例	*:SOURce1:POWer:POWer -2* *:SOURce1:POWer:POWer?* *:SOURce1: POWer:POWer -2* *SOURce1:POWer:POWer?* 返回值： *-2\n*

3.4.4.2 幅度（[:SOURce<ch>]:POWer）

命令格式	[:SOURce<ch>]:POWer <power> [:SOURce<ch>]:POWer?
功能描述	设置通道 ch 的 RF 输出信号的幅度。 查询通道 ch 的 RF 输出信号的幅度。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	请参考数据手册
返回值	浮点型，单位：dBm
默认值	对应最大输出频率的最小输出幅度
对应菜单	RF > 幅度 > 幅度
示例	*:SOURce1:POWer 2* *:SOURce1:POWer?*

	返回值： *2\n*

3.4.4.3 幅度（[:SOURce<ch>]:POWer[:LEVel][:IMMediate][:AMPLitude]）

命令格式	[:SOURce<ch>]:POWer[:LEVel][:IMMediate][:AMPLitude] <power> [:SOURce<ch>]:POWer[:LEVel][:IMMediate][:AMPLitude]?
功能描述	设置通道 ch 的 RF 输出信号的幅度。 查询通道 ch 的 RF 输出信号的幅度。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	请参考数据手册
返回值	浮点型，单位：dBm
默认值	对应最大输出频率的最小输出幅度
对应菜单	RF > 幅度 > 幅度
示例	*:SOURce1:SOURce1:POWer:LEVel -5* *:SOURce1:SOURce1:POWer:LEVel?* 返回值： *-5\n*

3.4.4.4 幅度偏移（[:SOURce<ch>]:POWer:OFFSet）

命令格式	[:SOURce<ch>]:POWer:OFFSet <power> [:SOURce<ch>]:POWer:OFFSet?
功能描述	设置通道 ch 的 RF 输出的幅度偏移。 查询通道 ch 的 RF 输出的幅度偏移。
参数类型	浮点型，单位：dB
参数范围	-100 ~ 100
返回值	浮点型，单位：dB
默认值	0
对应菜单	RF > 幅度 > 幅度偏移
示例	*:SOURce1:POWer:OFFSet 2* *:SOURce1:POWer:OFFSet?* 返回值： *2\n*

3.4.4.5 ALC 状态（[:SOURce<ch>]:POWer:ALC）

命令格式	[:SOURce<ch>]:POWer:ALC ON|OFF|AUTO [:SOURce<ch>]:POWer:ALC?
功能描述	设置通道 ch 的 ALC 的状态。 查询通道 ch 的 ALC 的状态。
参数类型	枚举
参数范围	ON|OFF|AUTO
返回值	ON|OFF|AUTO
默认值	AUTO
对应菜单	RF > 幅度 > ALC 状态
示例	*:SOURce1:POWer:ALC ON* *:SOURce1:POWer:ALC?* 返回值： *ON\n*

3.4.4.6 平坦度状态（[:SOURce<ch>]:CORRection[:FLATness]）

命令格式	[:SOURce<ch>]:CORRection[:FLATness] ON|OFF|1|0 [:SOURce<ch>]:CORRection[:FLATness]?
功能描述	设置通道 ch 的平坦度校正的开启状态。 查询通道 ch 的平坦度校正的开启状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	RF > 幅度 > 平坦度
示例	*:SOURce1:CORRection:FLATness ON* *:SOURce1:CORRection?* 返回值： *1\n*

3.4.4.7 平坦度列表添加行（[:SOURce<ch>]:CORRection:FLATness:PAIR）

命令格式	[:SOURce<ch>]:CORRection:FLATness:PAIR <freq>,<power>
功能描述	在通道 ch 的平坦度列表中添加一行。

参数类型	频率：浮点型，单位：Hz，kHz，MHz 或 GHz，默认为 Hz， 幅度：浮点型，单位：dB
参数范围	频率：全频率范围， 幅度：-100 ~ 100
对应菜单	RF > 幅度 > 平坦度 > 添加
示例	*:SOURce1:CORRection:FLATness:PAIR 3 MHz,-3*

3.4.4.8 平坦度列表删除行（[:SOURce<ch>]:CORRection:FLATness:DELete）

命令格式	[:SOURce<ch>]:CORRection:FLATness:DELete <row>
功能描述	从通道 ch 的平坦度列表中删除指定的行。
参数类型	整型
参数范围	1 ~ 平坦度列表的总行数
对应菜单	RF > 幅度 > 平坦度 > 删除
示例	*:SOURce1:CORRection:FLATness:DELete 1*

3.4.4.9 平坦度列表行数（[:SOURce<ch>]:CORRection:FLATness:COUNt?）

命令格式	[:SOURce<ch>]:CORRection:FLATness:COUNt?
功能描述	查询通道 ch 的平坦度列表的总行数。
返回值	整型
默认值	0
对应菜单	RF > 幅度 > 平坦度
示例	*:SOURce1:CORRection:FLATness:COUNt?* 返回值： *5\n*

3.4.4.10 保存平坦度列表（[:SOURce<ch>]:CORRection:STORe）

命令格式	[:SOURce<ch>]:CORRection:STORe <"file_name">
功能描述	将通道 ch 的平整度列表保存到文件。
参数类型	字符串
参数范围	无
对应菜单	RF > 幅度 > 平坦度 > 保存
示例	*:SOURce1:CORRection:STORe "U-disk3/test.uflt"*

3.4.4.11 加载平坦度列表（[:SOURce<ch>]:CORRection:LOAD）

命令格式	[:SOURce<ch>]:CORRection:LOAD <"file_name">
功能描述	从文件中加载通道 ch 的平整度列表。
参数类型	字符串
参数范围	无
对应菜单	RF > 幅度 > 平坦度 > 加载
示例	*:SOURce1:CORRection:LOAD "U-disk3/test.uflt"*

3.4.4.12 清空平坦度列表（[:SOURce<ch>]:CORRection:FLATness:PRESet）

命令格式	[:SOURce<ch>]:CORRection:FLATness:PRESet
功能描述	清空通道 ch 的平坦度列表。
对应菜单	RF > 幅度 > 平坦度 > 清空
示例	*:SOURce1:CORRection:FLATness:PRESet*

3.4.4.13 平坦度列表填充类型（[:SOURce<ch>]:CORRection:FLATness:FILL:TYPE）

命令格式	[:SOURce<ch>]:CORRection:FLATness:FILL:TYPE FLATness|MANUal|SWEEPlist [:SOURce<ch>]:CORRection:FLATness:FILL:TYPE?
功能描述	设置通道 ch 的平坦度列表的填充类型。 查询通道 ch 的平坦度列表的填充类型。
参数类型	枚举
参数范围	FLATness|MANUal|SWEEPlist
返回值	枚举
默认值	FLATness
对应菜单	RF > 幅度 > 平坦度 > 设置 > 填充方式
示例	*:SOURce1:CORRection:FLATness:FILL:TYPE FLATness* *:SOURce1:CORRection:FLATness:FILL:TYPE?* 返回值： *FLATness\n*

3.4.4.14 步进填充开始频率（[:SOURce<ch>]:CORRection:FLATness:STARtfreq）

命令格式	[:SOURce<ch>]:CORRection:FLATness:STARtfreq <freq> [:SOURce<ch>]:CORRection:FLATness:STARtfreq?

功能描述	当需要用功率计填充通道 ch 的平整度列表且填充类型为“步进填充”时，设置通道 ch 的步进填充的起始频率。 查询通道 ch 的步进填充的起始频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	全频率范围
返回值	浮点型，单位：Hz
默认值	最大频率值
对应菜单	RF > 幅度 > 平坦度 > 设置 > 填充方式 > 步进填充 > 开始频率
示例	*:SOURce1:CORRection:FLATness:STARtfreq 200 MHz* *:SOURce1:CORRection:FLATness:STARtfreq?* 返回值： *200000000\n*

3.4.4.15 步进填充结束频率（[:SOURce<ch>]:CORRection:FLATness:STOPfreq）

命令格式	[:SOURce<ch>]:CORRection:FLATness:STOPfreq <freq> [:SOURce<ch>]:CORRection:FLATness:STOPfreq?
功能描述	当需要用功率计填充通道 ch 的平整度列表且填充类型为“步进填充”时，设置通道 ch 的步进填充的结束频率。 查询通道 ch 的步进填充的结束频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	全频率范围
返回值	浮点型，单位：Hz
默认值	最大频率值
对应菜单	RF > 幅度 > 平坦度 > 设置 > 填充方式 > 步进填充 > 结束频率
示例	*:SOURce1:CORRection:FLATness:STOPfreq 500 MHz* *:SOURce1:CORRection:FLATness:STOPfreq?* 返回值： *500000000\n*

3.4.4.16 步进填充频率步进方式（[:SOURce<ch>]:CORRection:FLATness:SPACe）

命令格式	[:SOURce<ch>]:CORRection:FLATness:SPACe LINear|LOGarithmic [:SOURce<ch>]:CORRection:FLATness:SPACe?
功能描述	当需要用功率计填充通道 ch 的平整度列表且填充类型为“步进填充”时，设置

	通道 ch 的步进填充的频率步进方式。 查询通道 ch 的步进填充的频率步进方式。
参数类型	枚举
参数范围	LINear|LOGarithmic
返回值	枚举
默认值	LINear
对应菜单	RF > 幅度 > 平坦度 > 设置 > 填充方式 > 步进填充 > 频率步进
示例	*:SOURce1:CORRection:FLATness:SPACe LINear* *:SOURce1:CORRection:FLATness:SPACe?* 返回值： *LINear\n*

3.4.4.17 步进填充线性频率步进（[:SOURce<ch>]:CORRection:FLATness:LINStep）

命令格式	[:SOURce<ch>]:CORRection:FLATness:LINStep <freq> [:SOURce<ch>]:CORRection:FLATness:LINStep?
功能描述	设置通道 ch 的步进填充的线性频率步进。 查询通道 ch 的步进填充的线性频率步进。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
返回值	浮点型，单位：Hz
默认值	0
对应菜单	RF > 幅度 > 平坦度 > 设置 > 填充方式 > 步进填充 > 线性步进
示例	*:SOURce1:CORRection:FLATness:LINStep 200 MHz* *:SOURce1:CORRection:FLATness:LINStep?* 返回值： *200000000\n*

3.4.4.18 步进填充对数频率步进（[:SOURce<ch>]:CORRection:FLATness:LOGStep）

命令格式	[:SOURce<ch>]:CORRection:FLATness:LOGStep <value> [:SOURce<ch>]:CORRection:FLATness:LOGStep?
功能描述	设置通道 ch 的步进填充的对数频率步进。 查询通道 ch 的步进填充的对数频率步进。
参数类型	浮点型，单位：%
返回值	浮点型，单位：%
默认值	0

对应菜单	RF > 幅度 > 平坦度 > 设置 > 填充方式 > 步进填充 > 对数步进
示例	*:SOURce1:CORRection:FLATness:LOGStep 20* *:SOURce1:CORRection:FLATness:LOGStep?* 返回值: *20\n*

3.4.4.19 步进填充点数（[:SOURce<ch>]:CORRection:FLATness:POINt）

命令格式	[:SOURce<ch>]:CORRection:FLATness:POINt <points> [:SOURce<ch>]:CORRection:FLATness:POINt?
功能描述	设置通道 ch 的步进填充的扫描点数。 查询通道 ch 的步进填充的扫描点数。
参数类型	整型
参数范围	2 ~ 500
返回值	整型
默认值	2
对应菜单	RF > 幅度 > 平坦度 > 设置 > 填充方式 > 步进填充 > 点数
示例	*:SOURce1:CORRection:FLATness:POINt 5* *:SOURce1:CORRection:FLATness:POINt?* 返回值: *5\n*

3.4.4.20 用功率计填充平坦度（[:SOURce<ch>]:CORRection:CSET:DATA[:SENSor][:POWer]:SONCe）

命令格式	[:SOURce<ch>]:CORRection:CSET:DATA[:SENSor][:POWer]:SONCe
功能描述	用功率计填充通道 ch 的平坦度列表的幅度修正值。
对应菜单	RF > 幅度 > 平坦度 > 设置 > 幅度校准
示例	*:SOURce1:CORRection:CSET:DATA:SONCe*

3.4.5 扫描设置

3.4.5.1 扫描状态（[:SOURce<ch>]:SWEep:STATe）

命令格式	[:SOURce<ch>]:SWEep:STATe OFF|FREQuency|LEVel|LEV_FREQ [:SOURce<ch>]:SWEep:STATe?

功能描述	设置通道 ch 的扫描状态。 查询通道 ch 的扫描状态。
参数类型	枚举
参数范围	OFF|FREQuency|LEVel|LEV_FREQ
返回值	枚举
默认值	OFF
对应菜单	RF > 扫描 > 扫描状态
示例	*:SOURce1:SWEep:STATe LEV_FREQ* *:SWEep:STATe?* 返回值: *LEV_FREQ\n*

3.4.5.2 扫描类型（[:SOURce<ch>]:SWEep:TYPE）

命令格式	[:SOURce<ch>]:SWEep:TYPE LIST|STEP [:SOURce<ch>]:SWEep:TYPE?
功能描述	设置通道 ch 的扫描类型为步进扫描或列表扫描。 查询通道 ch 的扫描类型。
参数类型	枚举
参数范围	LIST|STEP
返回值	枚举
默认值	STEP
对应菜单	RF > 扫描 > 步进扫描 或 列表扫描
示例	*:SOURce1:SWEep:TYPE STEP* *:SOURce1:SWEep:TYPE?* 返回值: *STEP\n*

3.4.5.3 步进扫描开始频率（[:SOURce<ch>]:SWEep:STEP:STARt:FREQuency）

命令格式	[:SOURce<ch>]:SWEep:STEP:STARt:FREQuency <freq> [:SOURce<ch>]:SWEep:STEP:STARt:FREQuency?
功能描述	设置通道 ch 的步进扫描的起始频率。 查询通道 ch 的步进扫描的起始频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz

参数范围	全频率范围
返回值	浮点型，单位：Hz
默认值	最大频率值
对应菜单	RF > 扫描 > 步进扫描 > 开始频率
示例	*:SOURce1:SWEep:STEP:STARt:FREQuency 1 GHz* *:SOURce1:SWEep:STEP:STARt:FREQuency?* 返回值： *1000000000\n*

3.4.5.4 步进扫描结束频率（[:SOURce<ch>]:SWEep:STEP:STOP:FREQuency）

命令格式	[:SOURce<ch>]:SWEep:STEP:STOP:FREQuency <freq> [:SOURce<ch>]:SWEep:STEP:STOP:FREQuency?
功能描述	设置通道 ch 的步进扫描的结束频率。 查询通道 ch 的步进扫描的结束频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	全频率范围
返回值	浮点型，单位：Hz
默认值	最大频率值
对应菜单	RF > 扫描 > 步进扫描 > 结束频率
示例	*:SOURce1:SWEep:STEP:STOP:FREQuency 1 GHz* *:SOURce1:SWEep:STEP:STOP:FREQuency?* 返回值： *1000000000\n*

3.4.5.5 步进扫描开始幅度（[:SOURce<ch>]:SWEep:STEP:STARt:LEVel）

命令格式	[:SOURce<ch>]:SWEep:STEP:STARt:LEVel <level> [:SOURce<ch>]:SWEep:STEP:STARt:LEVel?
功能描述	设置通道 ch 的步进扫描的起始幅度。 查询通道 ch 的步进扫描的起始幅度。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	全幅度范围
返回值	浮点型，单位：dBm

默认值	-130 dBm
对应菜单	RF > 扫描 > 步进扫描 > 开始幅度
示例	*:SOURce1:SWEep:STEP:STARt:LEVel 0 dBm* *:SOURce1:SWEep:STEP:STARt:LEVel?* 返回值： *0\n*

3.4.5.6 步进扫描结束幅度（[:SOURce<ch>]:SWEep:STEP:STARt:LEVel）

命令格式	[:SOURce<ch>]:SWEep:STEP:STOP:LEVel <level> [:SOURce<ch>]:SWEep:STEP:STOP:LEVel?
功能描述	设置通道 ch 的步进扫描的结束幅度。 查询通道 ch 的步进扫描的结束幅度。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	全幅度范围
返回值	浮点型，单位：dBm
默认值	-130 dBm
对应菜单	RF > 扫描 > 步进扫描 > 结束幅度
示例	*:SOURce1:SWEep:STEP:STOP:LEVel 0 dBm* *:SOURce1:SWEep:STEP:STOP:LEVel?* 返回值： *0\n*

3.4.5.7 步进扫描驻留时间（[:SOURce<ch>]:SWEep:STEP:DWELl）

命令格式	[:SOURce<ch>]:SWEep:STEP:DWELl <time> [:SOURce<ch>]:SWEep:STEP:DWELl?
功能描述	设置通道 ch 的步进扫描的驻留时间。 查询通道 ch 的步进扫描的驻留时间。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	10 ms ~ 100 s
返回值	浮点型，单位：s
默认值	30 ms
对应菜单	RF > 扫描 > 步进扫描 > 驻留时间
示例	*:SOURce1:SWEep:STEP:DWELl 20 ms*

	:SOURce1:SWEep:STEP:DWELl? 返回值： *0.02\n*

3.4.5.8 步进扫描点数（[:SOURce<ch>]:SWEep:STEP:POINts）

命令格式	[:SOURce<ch>]:SWEep:STEP:POINts <points> [:SOURce<ch>]:SWEep:STEP:POINts?
功能描述	设置通道 ch 的步进扫描的扫描点数。 查询通道 ch 的步进扫描的扫描点数。
参数类型	整型
参数范围	2 ~ 65535
返回值	整型
默认值	11
对应菜单	RF > 扫描 > 步进扫描 > 扫描点数
示例	*:SOURce1:SWEep:STEP:POINts 2* *:SOURce1:SWEep:STEP:POINts?* 返回值： *2\n*

3.4.5.9 步进扫描形状（[:SOURce<ch>]:SWEep:STEP:SHAPe）

命令格式	[:SOURce<ch>]:SWEep:STEP:SHAPe TRIangle|SAWtooth [:SOURce<ch>]:SWEep:STEP:SHAPe?
功能描述	设置通道 ch 的步进扫描的扫描形状。 查询通道 ch 的步进扫描的扫描形状。
参数类型	枚举
参数范围	TRIangle|SAWtooth
返回值	枚举
默认值	SAWtooth
对应菜单	RF > 扫描 > 步进扫描 > 扫描形状
示例	*:SOURce1:SWEep:STEP:SHAPe TRIangle* *:SOURce1:SWEep:STEP:SHAPe?* 返回值： *TRIangle\n*

3.4.5.10 步进扫描频率步进方式（[:SOURce<ch>]:SWEep:STEP:SPACe）

命令格式	[:SOURce<ch>]:SWEep:STEP:SPACe LINear|LOGarithmic [:SOURce<ch>]:SWEep:STEP:SPACe?
功能描述	设置通道 ch 的步进扫描的频率步进方式。 查询通道 ch 的步进扫描的频率步进方式。
参数类型	枚举
参数范围	LINear|LOGarithmic
返回值	枚举
默认值	LINear
对应菜单	RF > 扫描 > 步进扫描 > 扫描步进
示例	*:SOURce1:SWEep:STEP:SPACe LOGarithmic* *:SOURce1:SWEep:STEP:SPACe?* 返回值： *LOGarithmic\n*

3.4.5.11 步进扫描线性频率步进（[:SOURce<ch>]:SWEep[:FREQuency]:STEP[:LINear]）

命令格式	[:SOURce<ch>]:SWEep[:FREQuency]:STEP[:LINear] <freq> [:SOURce<ch>]:SWEep[:FREQuency]:STEP[:LINear]?
功能描述	设置通道 ch 的步进扫描的线性频率步进。 查询通道 ch 的步进扫描的线性频率步进。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
返回值	浮点型，单位：Hz
默认值	0
对应菜单	RF > 扫描 > 步进扫描 > 频率线性步进
示例	*:SOURce1:SWEep:STEP 200 MHz* *:SOURce1:SWEep:STEP?* 返回值： *200000000\n*

3.4.5.12 步进扫描对数频率步进（[:SOURce<ch>]:SWEep[:FREQuency]:STEP:LOGarithmic）

命令格式	[:SOURce<ch>]:SWEep[:FREQuency]:STEP:LOGarithmic <value> [:SOURce<ch>]:SWEep[:FREQuency]:STEP:LOGarithmic?
功能描述	设置通道 ch 的步进扫描的对数频率步进。 查询通道 ch 的步进扫描的对数频率步进。

参数类型	浮点型，单位：%
返回值	浮点型，单位：%
默认值	0
对应菜单	RF > 扫描 > 步进扫描 > 频率对数步进
示例	*:SOURce1:SWEep:STEP:LOGarithmic 20* *:SOURce1:SWEep:STEP:LOGarithmic?* 返回值： *20\n*

3.4.5.13 扫描列表添加行（[:SOURce<ch>]:SWEep:LIST:ADDList）

命令格式	[:SOURce<ch>]:SWEep:LIST:ADDList <freq>,<level>,<time>
功能描述	在通道 ch 的扫描列表中添加一行。
参数类型	频率：浮点型，单位：Hz，kHz，MHz 或 GHz，默认为 Hz， 幅度：浮点型，单位：dB， 时间：浮点型，单位：ns，us，ms 或 s，默认为 s
参数范围	频率：全频率范围， 幅度：全幅度范围， 时间：10.0 ms ~ 100.0 s
对应菜单	RF > 扫描 > 列表扫描 > 添加
示例	*:SOURce1:SWEep:LIST:ADDList 1 GHz,0 dBm,1 s*

3.4.5.14 扫描列表删除行（[:SOURce<ch>]:SWEep:LIST:DELete）

命令格式	[:SOURce<ch>]:SWEep:LIST:DELete <row>
功能描述	从通道 ch 的扫描列表中删除指定的行。
参数类型	整型
参数范围	1 ~ 扫描列表的总行数
对应菜单	RF > 扫描 > 列表扫描 > 删除
示例	*:SOURce1:SWEep:LIST:DELete 1*

3.4.5.15 编辑扫描列表（[:SOURce<ch>]:SWEep:LIST:CHANGe）

命令格式	[:SOURce<ch>]:SWEep:LIST:CHANGe <row>,<freq>,<power>,<time>
功能描述	编辑通道 ch 的扫描列表中的指定行。
参数类型	行号：整型，

	频率：浮点型，单位：Hz，kHz，MHz 或 GHz，默认为 Hz， 幅度：浮点型，单位：dB， 时间：浮点型，单位：ns，us，ms 或 s，默认为 s
参数范围	行号：1 ~ 扫描列表的总行数， 频率：全频率范围， 幅度：全幅度范围， 时间：10.0 ms ~ 100.0 s
对应菜单	RF > 扫描 > 列表扫描
示例	*:SOURce1:SWEep:LIST:CHANGe 1,1 GHz,1 dBm, 1 s*

3.4.5.16 查询扫描列表行数（[:SOURce<ch>]:SWEep:LIST:CPOint?）

命令格式	[:SOURce<ch>]:SWEep:LIST:CPOint?
功能描述	查询通道 ch 的扫描列表的行数。
返回值	整型
默认值	1
对应菜单	RF > 扫描 > 列表扫描
示例	*:SOURce1:SWEep:LIST:CPOint?* 返回值： *5\n*

3.4.5.17 查询扫描列表数据（[:SOURce<ch>]:SWEep:LIST:LIST?）

命令格式	[:SOURce<ch>]:SWEep:LIST:LIST? <begin_row>,<end_row>
功能描述	查询通道 ch 的扫描列表中 begin_ row 到 end_row 的数据。
参数类型	整型，整型
参数范围	1 ~ 扫描列表的总行数， 开始行 ~ 扫描列表的总行数
返回值	字符串
对应菜单	RF > 扫描 > 列表扫描
示例	*:SOURce1:SWEep:LIST:LIST? 1,3* 返回值： *1000000000,0,0.03\s2000000000,-2.4,0.03\s3000000000,-4.8,0.03\n*

3.4.5.18 清空扫描列表（[:SOURce<ch>]:SWEep:LIST:INITialize:PRESet）

命令格式	[:SOURce<ch>]:SWEep:LIST:INITialize:PRESet
功能描述	清空通道 ch 的扫描列表。
对应菜单	RF > 扫描 > 列表扫描 > 清空
示例	*:SOURce1:SWEep:LIST:INITialize:PRESet*

3.4.5.19 从步进扫描导入扫描列表（[:SOURce<ch>]:SWEep:LIST:INITialize:FSTep）

命令格式	[:SOURce<ch>]:SWEep:LIST:INITialize:FSTep
功能描述	从通道 ch 的步进扫描导入扫描列表。
对应菜单	RF > 扫描 > 列表扫描 > 导入
示例	*:SOURce1:SWEep:LIST:INITialize:FSTep*

3.4.5.20 加载扫描列表（[:SOURce<ch>]:SWEep:LOAD）

命令格式	[:SOURce<ch>]:SWEep:LOAD <"file_name">
功能描述	从文件中加载通道 ch 的扫描列表。
参数类型	字符串
参数范围	无
对应菜单	RF > 扫描 > 列表扫描 > 加载
示例	*:SOURce1:SWEep:LOAD "U-disk3/test.lsw"*

3.4.5.21 保存扫描列表（[:SOURce<ch>]:SWEep:STORe）

命令格式	[:SOURce<ch>]:SWEep:STORe <"file_name">
功能描述	将通道 ch 的扫描列表保存到文件。
参数类型	字符串
参数范围	无
对应菜单	RF > 扫描 > 列表扫描 > 保存
示例	*:SOURce1:SWEep:STORe "U-disk3/test.lsw"*

3.4.5.22 扫描方向（[:SOURce<ch>]:SWEep:DIRect）

命令格式	[:SOURce<ch>]:SWEep:DIRect FWD|REV [:SOURce<ch>]:SWEep:DIRect?
功能描述	设置通道 ch 的扫描方向。 查询通道 ch 的扫描方向。
参数类型	枚举
参数范围	FWD|REV
返回值	枚举
默认值	FWD
对应菜单	RF > 扫描 > 扫描方向
示例	*:SOURce1:SWEep:DIRect REV* *:SOURce1:SWEep:DIRect?* 返回值： *REV\n*

3.4.5.23 扫描模式（[:SOURce<ch>]:SWEep:MODE）

命令格式	[:SOURce<ch>]:SWEep:MODE CONTinuel|SINGle [:SOURce<ch>]:SWEep:MODE?
功能描述	设置通道 ch 的扫描模式为连续或单次。 查询通道 ch 的扫描模式。
参数类型	枚举
参数范围	CONTinuel|SINGle
返回值	枚举
默认值	CONTinue
对应菜单	RF > 扫描 > 扫描模式
示例	*:SOURce1:SWEep:MODE SINGle* *:SOURce1:SWEep:MODE?* 返回值： *SINGle\n*

3.4.5.24 执行单次扫描（[:SOURce<ch>]:SWEep:EXECute）

命令格式	[:SOURce<ch>]:SWEep:EXECute
功能描述	当通道 ch 的扫描模式是单次时，在通道 ch 中执行单次扫描。
对应菜单	RF > 扫描 > 执行单次扫描
示例	*:SOURce1:SWEep:EXECute*

3.4.5.25 扫描触发方式（[:SOURce<ch>]:SWEep:SWEep:TRIGger:TYPE）

命令格式	[:SOURce<ch>]:SWEep:SWEep:TRIGger:TYPE AUTO|KEY|BUS|EXT [:SOURce<ch>]:SWEep:SWEep:TRIGger:TYPE?
功能描述	设置通道 ch 的扫描触发方式。 查询通道 ch 的扫描触发方式。
参数类型	枚举
参数范围	AUTO|KEY|BUS|EXT
返回值	枚举
默认值	AUTO
对应菜单	RF > 扫描 > 触发方式
示例	*:SOURce1:SWEep:SWEep:TRIGger:TYPE KEY* *:SOURce1:SWEep:SWEep:TRIGger:TYPE?* 返回值： *KEY\n*

3.4.5.26 扫描点触发方式（[:SOURce<ch>]:SWEep:POINt:TRIGger:TYPE）

命令格式	[:SOURce<ch>]:SWEep:POINt:TRIGger:TYPE AUTO|KEY|BUS|EXT [:SOURce<ch>]:SWEep:POINt:TRIGger:TYPE?
功能描述	设置通道 ch 的扫描的点触发方式。 查询通道 ch 的扫描的点触发方式。
参数类型	枚举
参数范围	AUTO|KEY|BUS|EXT
返回值	枚举
默认值	AUTO
对应菜单	RF > 扫描 > 点触发方式
示例	*:SOURce1:SWEep:POINt:TRIGger:TYPE KEY* *:SOURce1:SWEep:POINt:TRIGger:TYPE?* 返回值： *KEY\n*

3.4.5.27 扫描触发沿（[:SOURce<ch>]:INPut:TRIGger:SLOPe）

命令格式	[:SOURce<ch>]:INPut:TRIGger:SLOPe POSitive|NEGative [:SOURce<ch>]:INPut:TRIGger:SLOPe?
功能描述	设置通道 ch 的扫描功能的外部触发信号的触发沿。 查询通道 ch 的扫描功能的外部触发信号的触发沿。

参数类型	枚举
参数范围	POSitive|NEGative
返回值	枚举
默认值	POSitive
对应菜单	RF > 扫描 > 触发沿
示例	*:SOURce1:INPut:TRIGger:SLOPe NEGative* *:SOURce1:INPut:TRIGger:SLOPe?* 返回值： *NEGative\n*

3.4.5.28 查询当前的扫描点（[:SOURce<ch>]:SWEep:CURRent:DATA?）

命令格式	[:SOURce<ch>]:SWEep:CURRent:DATA?
功能描述	查询通道 ch 的当前的扫描点。
返回值	字符串 格式为：index,{freq,level,time} Index：整型，扫描点索引， Freq：浮点型，扫描点频率，单位：Hz， Level：浮点型，扫描点幅度，单位：dBm， Time：浮点型，扫描点驻留时间，单位：s。
默认值	无
对应菜单	屏幕上方的显示频率和显示幅度
示例	*:SOURce1:SWEep:CURRent:DATA?* 返回值： *1,{1000000000,0,0.03}\n*

3.4.5.29 查询当前扫描点的频率（[:SOURce<ch>]:SWEep:CURRent:FREQuency?）

命令格式	[:SOURce<ch>]:SWEep:CURRent:FREQuency?
功能描述	查询通道 ch 的当前扫描点的频率。
返回值	浮点型，单位：Hz
默认值	无
对应菜单	屏幕上方的显示频率
示例	*:SOURce1:SWEep:CURRent:FREQuency?* 返回值： *1000000000.000000\n*

3.4.5.30 查询当前扫描点的幅度（[:SOURce<ch>]:SWEep:CURRent:LEVel?）

命令格式	[:SOURce<ch>]:SWEep:CURRent:LEVel?
功能描述	查询通道 ch 的当前扫描点的幅度。
返回值	浮点型，单位：dBm
默认值	无
对应菜单	屏幕上方的显示幅度
示例	*:SOURce1:SWEep:CURRent:LEVel?* 返回值： *0.000000\n*

3.4.6 功率计设置

3.4.6.1 功率计功率控制状态（[:SOURce]:POWer:SPC:STATe）

命令格式	[:SOURce]:POWer:SPC:STATe ON|OFF|1|0 [:SOURce]:POWer:SPC:STATe?
功能描述	设置功率计测量的功率控制状态。 查询功率计测量的功率控制状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
等效命令	:SENSe[:POWer]:LEV:CTL:STATe ON|OFF|1|0 :SENSe[:POWer]:LEV:CTL:STATe?
对应菜单	HOME > > Power Sensor > 功率控制
示例	*:POWer:SPC:STATe ON* *:POWer:SPC:STATe?* 返回值： *1\n*

3.4.6.2 功率计通道功率控制状态（:SOURce<ch>:POWer:LEV:CTL:STATe）

命令格式	:SOURce<ch>:POWer:LEV:CTL:STATe ON|OFF|1|0 :SOURce<ch>:POWer:LEV:CTL:STATe?
功能描述	设置功率计测量的功率控制某个通道的启用状态。 查询功率计测量的功率控制某个通道的启用状态。

参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
等效命令	:SENSe[:POWer]:LEV:CTL:STATe ON|OFF|1|0 :SENSe[:POWer]:LEV:CTL:STATe?
对应菜单	HOME >  >  > 功率控制
示例	*:SOURce1:POWer: LEV:CTL:STATe ON* *:SOURce1:POWer: LEV:CTL:STATe?* 返回值: *1\n*

3.4.6.3 功率控制目标功率（[:SOURce]:POWer:SPC:TARGet）

命令格式	[:SOURce]:POWer:SPC:TARGet <power> [:SOURce]:POWer:SPC:TARGet?
功能描述	设置功率计功率控制的目标功率。 查询功率计功率控制的目标功率。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	-120 dBm ~ 20 dBm
返回值	浮点型，单位：dBm
默认值	0 dBm
等效命令	:SENSe[:POWer]:SPC:TARGet <power> :SENSe[:POWer]:SPC:TARGet?
对应菜单	HOME >  >  >功率控制 > 目标功率
示例	*:POWer:SPC:TARGet 0* *:POWer:SPC:TARGet?* 返回值: *0\n*

3.4.6.4 功率控制限制功率（[:SOURce]:POWer:LIMit）

命令格式	[:SOURce]:POWer:LIMit <power> [:SOURce]:POWer:LIMit?
功能描述	设置功率计功率控制的限制功率。

	查询功率计功率控制的限制功率。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	-120 dBm ~ 20 dBm
返回值	浮点型，单位：dBm
默认值	0 dBm
等效命令	:SENSe[:POWer]:LIMit <power> :SENSe[:POWer]:LIMit?
对应菜单	HOME >  >  >功率控制 > 限制功率
示例	*POWer:LIMit 1* *POWer:LIMit?* 返回值： *1\n*

3.4.6.5 功率控制捕获范围（[:SOURce<ch>]:POWer:SPC:CRANge）

命令格式	[SOURce]:POWer:SPC:CRANge <power> [SOURce]:POWer:SPC:CRANge?
功能描述	设置功率计功率控制的捕获范围。 查询功率计功率控制的捕获范围。
参数类型	浮点型，单位：dB
参数范围	0 dB ~ 50 dB
返回值	浮点型，单位：dB
默认值	0 dB
等效命令	:SENSe[:POWer]:SPC:CRANge <power> :SENSe[:POWer]:SPC:CRANge?
对应菜单	HOME >  >  >功率控制 > 捕获范围
示例	*:POWer:SPC:CRANge 5* *:POWer:SPC:CRANge?* 返回值： *5\n*

3.4.7 模拟调制设置

3.4.7.1 模拟调制开关状态（[:SOURce<ch>]:MODulation）

命令格式	[:SOURce<ch>]:MODulation ON|OFF|1|0 [:SOURce<ch>]:MODulation?
功能描述	设置通道 ch 的模拟调制的开关状态。 查询通道 ch 的模拟调制的开关状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
等效命令	:OUTPut:MODulation[:STATe] ON|OFF|1|0 :OUTPut:MODulation[:STATe]?
对应菜单	ANALOG MOD > On
示例	*:SOURce1:MODulation ON* *:SOURce1:MODulation?* 返回值： *1\n*

3.4.7.2 频率调制设置

3.4.7.2.1 频率调制开关状态（[:SOURce<ch>]:FM:STATe）

命令格式	[:SOURce<ch>]:FM:STATe ON|OFF|1|0 [:SOURce<ch>]:FM:STATe?
功能描述	设置通道 ch 的频率调制的调制开关状态。 查询通道 ch 的频率调制的调制开关状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	Analog Mod > 调频 > FM 状态
示例	*:FM:STATe ON* *:FM:STATe?* 返回值： *1\n*

3.4.7.2.2 频率调制源（[:SOURce<ch>]:FM:SOURce）

命令格式	[:SOURce<ch>]:FM:SOURce INT1|INT2|INT1,INT2|EXTernal|INT1,EXT|DUAL [:SOURce<ch>]:FM:SOURce?
功能描述	设置通道 ch 的频率调制的调制源。 查询通道 ch 的频率调制的调制源。
参数类型	枚举
参数范围	INT1|INT2|INT1,INT2|EXTernal|INT1,EXT|DUAL
返回值	枚举
默认值	INT1
对应菜单	Analog Mod > 调频 > FM 源
示例	*:FM:SOURce EXT* *:FM:SOURce?* 返回值： *EXTernal\n*

3.4.7.2.3 频率调制波形 1（[:SOURce<ch>]:FM1:WAVeform）

命令格式	[:SOURce<ch>]:FM1:WAVeform SINE|SQUare|SAWTooth|TRIangle [:SOURce<ch>]:FM1:WAVeform?
功能描述	设置通道 ch 的频率调制的内部源 1 的调制波形。 查询通道 ch 的频率调制的内部源 1 的调制波形。
参数类型	枚举
参数范围	SINE|SQUare|SAWTooth|TRIangle
返回值	枚举
默认值	SINE
对应菜单	Analog Mod > 调频 > FM 波形 1
示例	*:FM1:WAVeform SQUare* *:FM1:WAVeform?* 返回值： *SQUare\n*

3.4.7.2.4 频率调制最大频偏 1（[:SOURce<ch>]:FM1:DEViation）

命令格式	[:SOURce<ch>]:FM1:DEViation <value> [:SOURce<ch>]:FM1:DEViation?
功能描述	设置通道 ch 的频率调制的内部源 1 的最大频偏。 查询通道 ch 的频率调制的内部源 1 的最大频偏。

参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	1 Hz ~ N*1 MHz， N 为频率分段系数，请查阅数据手册
返回值	浮点型，单位：Hz
默认值	100 kHz
对应菜单	Analog Mod > 调频 > 最大频偏 1
示例	*:FM1:DEViation 500 kHz* *:FM1:DEViation?* 返回值： *500000\n*

3.4.7.2.5 调制频率 1（[:SOURce<ch>]:FM1:FREQuency）

命令格式	[:SOURce<ch>]:FM1:FREQuency <value> [:SOURce<ch>]:FM1:FREQuency?
功能描述	设置通道 ch 的频率调制的内部源 1 的调制频率。 查询通道 ch 的频率调制的内部源 1 的调制频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	调制波为正弦波时：0.01 Hz ~ 100 kHz 调制波为方波/锯齿波/三角波时：0.01 Hz ~ 20 kHz
返回值	浮点型，单位：Hz
默认值	10 kHz
对应菜单	Analog Mod > 调频 > FM 调制频率 1
示例	*:FM1:FREQuency 40 kHz* *:FM1:FREQuency?* 返回值： *40000\n*

3.4.7.2.6 频率调制初相 1（[:SOURce<ch>]:FM1:PHASe）

命令格式	[:SOURce<ch>]:FM1:PHASe <value> [:SOURce<ch>]:FM1:PHASe?
功能描述	当通道 ch 的频率调制源为内部源 1+内部源 2 或双波形时，设置/查询内部源 1 的初相。
参数类型	浮点型，单位：degree (°)

参数范围	-360 ~ +360
返回值	浮点型，单位：degree (°)
默认值	0
对应菜单	Analog Mod > 调频 > FM 初相 1
示例	*:FM1:PHASe -30* *:FM1:PHASe?* 返回值： *-30\n*

3.4.7.2.7 频率调制波形 2（[:SOURce<ch>]:FM2:WAVeform）

命令格式	[:SOURce<ch>]:FM2:WAVeform SINE|SQUare|SAWTooth|TRIangle [:SOURce<ch>]:FM2:WAVeform?
功能描述	设置通道 ch 的频率调制的内部源 2 的调制波形。 查询通道 ch 的频率调制的内部源 2 的调制波形。
参数类型	枚举
参数范围	SINE|SQUare|SAWTooth|TRIangle
返回值	枚举
默认值	SINE
对应菜单	Analog Mod > 调频 > FM 波形 2
示例	*:FM2:WAVeform TRIangle* *:FM2:WAVeform?* 返回值： *TRIangle\n*

3.4.7.2.8 频率调制最大频偏 2（[:SOURce<ch>]:FM2:DEViation）

命令格式	[:SOURce<ch>]:FM2:DEViation <value> [:SOURce<ch>]:FM2:DEViation?
功能描述	设置通道 ch 的频率调制的内部源 2 的最大频偏。 查询通道 ch 的频率调制的内部源 2 的最大频偏。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	1 Hz ~ N*1 MHz N 为频率分段系数，请查阅数据手册
返回值	浮点型，单位：Hz

默认值	100 kHz
对应菜单	Analog Mod > 调频 > 最大频偏 2
示例	*:FM2:DEViation 500 kHz* *:FM2:DEViation?* 返回值： *500000\n*

3.4.7.2.9 调制频率 2（[:SOURce<ch>]:FM2:FREQuency）

命令格式	[:SOURce<ch>]:FM2:FREQuency <value> [:SOURce<ch>]:FM2:FREQuency?
功能描述	设置通道 ch 的频率调制的内部源 2 的调制频率。 查询通道 ch 的频率调制的内部源 2 的调制频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	调制波为正弦波时：0.01 Hz ~ 100 kHz 调制波为方波/锯齿波/三角波时：0.01 Hz ~ 20 kHz
返回值	浮点型，单位：Hz
默认值	10 kHz
对应菜单	Analog Mod > 调频 > FM 调制频率 2
示例	*:FM2:FREQuency 40 kHz* *:FM2:FREQuency?* 返回值： *40000\n*

3.4.7.2.10 频率调制初相 2（[:SOURce<ch>]:FM2:PHASe）

命令格式	[:SOURce<ch>]:FM2:PHASe <value> [:SOURce<ch>]:FM2:PHASe?
功能描述	当通道 ch 的频率调制源为内部源 1+内部源 2 或双波形时，设置/查询内部源 2 的初相。
参数类型	浮点型，单位：deg 或 rad，默认为 deg
参数范围	-360° ~ +360°
返回值	浮点型，单位：deg
默认值	0
对应菜单	Analog Mod > 调频 > FM 初相 2
示例	*:FM2:PHASe -30 deg*

	:FM2:PHASe? 返回值： *-30\n*

3.4.7.2.11 内部源 1 幅度占比（[:SOURce<ch>]:FM1:PROPortion）

命令格式	[:SOURce<ch>]:FM1:PROPortion <value> [:SOURce<ch>]:FM1:PROPortion?
功能描述	当通道 ch 的频率调制源为双波形时，设置/查询内部源 1 的幅度占比。
参数类型	浮点型
参数范围	0 ~ 1
返回值	浮点型
默认值	0.5
对应菜单	Analog Mod > 调频 > 内部源 1 幅度占比
示例	*:FM1:PROPortion 0.6* *:FM1:PROPortion?* 返回值： *0.6\n*

3.4.7.2.12 外调制灵敏度（[:SOURce<ch>]:FM1:SENSitivity?）

命令格式	[:SOURce<ch>]:FM1:SENSitivity?
功能描述	当通道 ch 的频率调制源包含外部源时，查询外部调制源的灵敏度。
返回值	浮点型，单位：Hz/V
默认值	无
对应菜单	Analog Mod > 调频 > 外调制灵敏度
示例	*FM1:SENSitivity?* 返回值： *125000\n*

3.4.7.3 相位调制设置

3.4.7.3.1 相位调制开关状态（[:SOURce<ch>]:PM:STATe）

命令格式	[:SOURce<ch>]:PM:STATe ON|OFF|1|0 [:SOURce<ch>]:PM:STATe?
功能描述	设置通道 ch 的相位调制的调制开关状态。

	查询通道 ch 的相位调制的调制开关状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	Analog Mod > 调相 > PM 状态
示例	*:PM:STATe ON* *:PM:STATe?* 返回值： *1\n*

3.4.7.3.2 相位调制波形（[:SOURce<ch>]:PM:WAVeform）

命令格式	[:SOURce<ch>]:PM:WAVeform SINE|SQUare [:SOURce<ch>]:PM:WAVeform?
功能描述	设置通道 ch 的相位调制的调制波形。 查询通道 ch 的相位调制的调制波形。
参数类型	枚举
参数范围	SINE|SQUare
返回值	枚举
默认值	SINE
对应菜单	Analog Mod > 调相 > PM 波形
示例	*:PM:WAVeform SQUare* *:PM:WAVeform?* 返回值： *SQUare\n*

3.4.7.3.3 相位调制源（[:SOURce<ch>]:PM:SOURce）

命令格式	[:SOURce<ch>]:PM:SOURce INTernal|EXTernal|INT,EXT [:SOURce<ch>]:PM:SOURce?
功能描述	设置通道 ch 的相位调制的调制源。 查询通道 ch 的相位调制的调制源。
参数类型	枚举
参数范围	INTernal|EXTernal|INT,EXT
返回值	枚举
默认值	INTernal

对应菜单	Analog Mod > 调相 > PM 源
示例	*:PM:SOURce EXTernal* *:PM:SOURce?* 返回值： *EXTernal\n*

3.4.7.3.4 最大相位偏移（[:SOURce<ch>]:PM:DEViation）

命令格式	[:SOURce<ch>]:PM:DEViation <value> [:SOURce<ch>]:PM:DEViation?
功能描述	设置通道 ch 的相位调制的最大相位偏移。 查询通道 ch 的相位调制的最大相位偏移。
参数类型	浮点型，单位：rad
参数范围	0.01 rad ~ N*5 rad， N 为频率分段系数，请查阅数据手册
返回值	浮点型，单位：rad
默认值	1
对应菜单	Analog Mod > 调相 > 最大相位偏移
示例	*:PM:DEViation 2* *:PM:DEViation?* 返回值： *2\n*

3.4.7.3.5 相位调制频率（[:SOURce<ch>]:PM:FREQuency）

命令格式	[:SOURce<ch>]:PM:FREQuency <value> [:SOURce<ch>]:PM:FREQuency?
功能描述	设置通道 ch 的相位调制的调制频率。 查询通道 ch 的相位调制的调制频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	调制波为正弦波时：0.01 Hz ~ 100 kHz， 调制波为方波时：0.01 Hz ~ 20 kHz。
返回值	浮点型，单位：Hz
默认值	10 kHz
对应菜单	Analog Mod > 调相 > PM 调制频率

示例	*:PM:FREQuency 1 kHz* *:PM:FREQuency?* 返回值： *1000\n*

3.4.7.3.6 外调制灵敏度（[:SOURce<ch>]:PM:SENSitivity?）

命令格式	[:SOURce<ch>]:PM:SENSitivity?
功能描述	查询通道 ch 的相位调制的外调制灵敏度。
返回值	浮点型，单位：rad/V
默认值	无
对应菜单	Analog Mod > 调相 > 外调制灵敏度
示例	*PM:SENSitivity?* 返回值： *0.25\n*

3.4.8 脉冲调制设置

3.4.8.1 脉冲调制开关状态（[:SOURce<ch>]:PULM:STATe）

命令格式	[:SOURce<ch>]:PULM:STATe ON|OFF|1|0 [:SOURce<ch>]:PULM:STATe?
功能描述	设置通道 ch 的脉冲调制的调制开关状态。 查询通道 ch 的脉冲调制的调制开关状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	Analog Mod > 脉冲 > 脉冲状态
示例	*:SOURce1:PULM:STATe ON* *:SOURce1:PULM:STATe?* 返回值： *1\n*

3.4.8.2 脉冲调制源（[:SOURce<ch>]:PULM:SOURce）

命令格式	[:SOURce<ch>]:PULM:SOURce INTernal|EXTernal [:SOURce<ch>]:PULM:SOURce?
功能描述	设置通道 ch 的脉冲调制源为内部或外部。 查询通道 ch 的脉冲调制源。
参数类型	枚举
参数范围	INTernal|EXTernal
返回值	枚举
默认值	INTernal
等效命令	[:SOURce<ch>]:PULM:SOURce:INT INTernal|EXTernal [:SOURce<ch>]:PULM:SOURce:INT?
对应菜单	Analog Mod > 脉冲 > 脉冲源
示例	*:SOURce1:PULM:SOURce INTernal* *:SOURce1:PULM:SOURce?* 返回值： *INTernal\n*

3.4.8.3 脉冲调制源（[:SOURce<ch>]:PULM:SOURce:INT）

命令格式	[:SOURce<ch>]:PULM:SOURce:INT INTernal|EXTernal [:SOURce<ch>]:PULM:SOURce:INT?
功能描述	设置通道 ch 的脉冲调制源为内部或外部。 查询通道 ch 的脉冲调制源。
参数类型	枚举
参数范围	INTernal|EXTernal
返回值	枚举
默认值	INTernal
等效命令	[:SOURce<ch>]:PULM:SOURce INTernal|EXTernal [:SOURce<ch>]:PULM:SOURce?
对应菜单	Analog Mod > 脉冲 > 脉冲源
示例	*:SOURce1:PULM:SOURce:INT EXTernal* *:SOURce1:PULM:SOURce:INT?* 返回值： *EXTernal\n*

3.4.8.3.1 脉冲输出状态（[:SOURce<ch>]:PULM:STATe）

命令格式	[:SOURce<ch>]:PULM:OUT:STATe ON|OFF|1|0 [:SOURce<ch>]:PULM:OUT:STATe?
功能描述	设置通道 ch 的脉冲调制的脉冲输出状态。 查询通道 ch 的脉冲调制的脉冲输出状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	Analog Mod > 脉冲 > 脉冲输出
示例	*:SOURce1:PULM:OUT:STATe ON* *:SOURce1:PULM:OUT:STATe?* 返回值： *1\n*

3.4.8.3.2 脉冲输出极性（[:SOURce<ch>]:PULM:POLarity）

命令格式	[:SOURce<ch>]:PULM:POLarity NORMal|INVerted [:SOURce<ch>]:PULM:POLarity?
功能描述	设置通道 ch 的脉冲输出极性。 查询通道 ch 的脉冲输出极性。
参数类型	枚举
参数范围	NORMal|INVerted
返回值	枚举
默认值	NORMal
对应菜单	Analog Mod > 脉冲 > 脉冲输出极性
示例	*:SOURce1:PULM:POL INV* *:SOURce1:PULM:POLarity?* 返回值： *INVerted\n*

3.4.8.3.3 脉冲类型（[:SOURce<ch>]:PULM:MODE）

命令格式	[:SOURce<ch>]:PULM:MODE SINGle|DOUBle|PTRain [:SOURce<ch>]:PULM:MODE?

功能描述	设置通道 ch 的脉冲类型为单脉冲、双脉冲或脉冲序列。 查询通道 ch 的脉冲类型。
参数类型	枚举
参数范围	SINGle|DOUBle|PTRain
返回值	枚举
默认值	SINGle
对应菜单	Analog Mod > 脉冲 > 脉冲类型
示例	*:SOURce1:PULM:MODE DOUB* *:SOURce1:PULM:MODE?* 返回值： *DOUBle\n*

3.4.8.3.4 脉冲周期（[:SOURce<ch>]:PULM:PERiod）

命令格式	[:SOURce<ch>]:PULM:PERiod <value> [:SOURce<ch>]:PULM:PERiod?
功能描述	当通道 ch 的脉冲类型是单脉冲或双脉冲时，设置脉冲周期。 当通道 ch 的脉冲类型是单脉冲或双脉冲时，查询脉冲周期。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	40 ns ~ 300 s
返回值	浮点型，单位：s
默认值	10 ms
等效命令	[:SOURce<ch>]:PULM:INT[1]:PERiod <value> [:SOURce<ch>]:PULM:INT[1]:PERiod?
对应菜单	Analog Mod > 脉冲 > 脉冲周期
示例	*:SOURce1:PULM:PER 220 us* *:SOURce1:PULM:PER?* 返回值： *0.00022\n*

3.4.8.3.5 脉冲周期（[:SOURce<ch>]:PULM:INT[1]:PERiod）

命令格式	[:SOURce<ch>]:PULM:INT[1]:PERiod <value> [:SOURce<ch>]:PULM:INT[1]:PERiod?

功能描述	当通道 ch 的脉冲类型是单脉冲或双脉冲时，设置脉冲周期。 当通道 ch 的脉冲类型是单脉冲或双脉冲时，查询脉冲周期。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	40 ns ~ 300 s
返回值	浮点型，单位：s
默认值	10 ms
等效命令	[:SOURce<ch>]:PULM:PERiod <value> [:SOURce<ch>]:PULM:PERiod?
对应菜单	Analog Mod > 脉冲 > 脉冲周期
示例	*:SOURce1:PULM:INT1:PERiod 900 ns* *:SOURce1:PULM:INT1:PERiod?* 返回值： *9e-07\n*

3.4.8.3.6 脉冲宽度（[:SOURce<ch>]:PULM:WIDTh）

命令格式	[:SOURce<ch>]:PULM:WIDTh <value> [:SOURce<ch>]:PULM:WIDTh?
功能描述	当通道 ch 的脉冲类型是单脉冲时，设置通道 ch 的脉冲宽度；当通道 ch 的脉冲类型是双脉冲时，设置通道 ch 的第一个脉冲的脉冲宽度。 查询通道 ch 的脉冲宽度。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	20 ns ~ 300 s
返回值	浮点型，单位：s
默认值	2 ms
等效命令	[:SOURce<ch>]:PULM:INT[1]:PWIDth <value> [:SOURce<ch>]:PULM:INT[1]:PWIDth?
对应菜单	Analog Mod > 脉冲 > 脉冲宽度
示例	*:SOURce1:PULM:WIDT 33 us* *:SOURce1:PULM:WIDT?* 返回值： *3.3e-05\n*

3.4.8.3.7 脉冲宽度（[:SOURce<ch>]:PULM:INT[1]:PWIDth）

命令格式	[:SOURce<ch>]:PULM:INT[1]:PWIDth <value> [:SOURce<ch>]:PULM:INT[1]:PWIDth?
功能描述	当通道 ch 的脉冲类型是单脉冲时，设置通道 ch 的脉冲宽度；当通道 ch 的脉冲类型是双脉冲时，设置通道 ch 的第一个脉冲的脉冲宽度。 查询通道 ch 的脉冲宽度。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	20 ns ~ 300 s
返回值	浮点型，单位：s
默认值	2 ms
等效命令	[:SOURce<ch>]:PULM:WIDTh <value> [:SOURce<ch>]:PULM:WIDTh?
对应菜单	Analog Mod > 脉冲 > 脉冲宽度
示例	*:SOURce1:PULM:INT:PWIDth 400 ns* *:SOURce1:PULM:INT:PWIDth?* 返回值： *4e-07\n*

3.4.8.3.8 双脉冲延迟（[:SOURce<ch>]:PULM:DOUBle:DELay）

命令格式	[:SOURce<ch>]:PULM:DOUBle:DELay <value> [:SOURce<ch>]:PULM:DOUBle:DELay?
功能描述	当通道 ch 的脉冲类型是双脉冲时，设置通道 ch 的双脉冲延迟。 当通道 ch 的脉冲类型是双脉冲时，查询通道 ch 的双脉冲延迟。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	20 ns ~ 300 s
返回值	浮点型，单位：s
默认值	4 ms
对应菜单	Analog Mod > 脉冲 > 双脉冲延迟
示例	*:SOURce1:PULM:DOUBle:DELay 2 ms* *:SOURce1:PULM:DOUBle:DELay?* 返回值： *0.002\n*

3.4.8.3.9 #2 脉冲宽度（[:SOURce<ch>]:PULM:DOUBle:WIDTh）

命令格式	[:SOURce<ch>]:PULM:DOUBle:WIDTh <time> [:SOURce<ch>]:PULM:DOUBle:WIDTh?
功能描述	当通道 ch 的脉冲类型是双脉冲时，设置通道 ch 的第二个脉冲的脉冲宽度。 当通道 ch 的脉冲类型是双脉冲时，查询通道 ch 的第二个脉冲的脉冲宽度。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	20 ns ~ 300 s
返回值	浮点型，单位：s
默认值	2 ms
对应菜单	Analog Mod > 脉冲 > #2 脉冲宽度
示例	*:SOURce1:PULM:DOUBle:WIDTh 5 ms* *:SOURce1:PULM:DOUBle:WIDTh?* 返回值： *0.005\n*

3.4.8.3.10 脉冲序列添加行（[:SOURce<ch>]:PULM:TRAin:PAIR）

命令格式	[:SOURce<ch>]:PULM:TRAin:PAIR <row>
功能描述	复制指定行并粘贴到通道 ch 的脉冲序列的指定行的前面。
参数类型	整型
参数范围	1 ~ 当前脉冲序列的总行数
对应菜单	Analog Mod > 脉冲 > 脉冲序列 > 添加
示例	*PULM:TRAin:PAIR 1*

3.4.8.3.11 脉冲序列删除行（[:SOURce<ch>]:PULM:TRAin:DELete）

命令格式	[:SOURce<ch>]:PULM:TRAin:DELete <row>
功能描述	从通道 ch 的脉冲序列中删除指定的行。
参数类型	整型
参数范围	1 ~ 当前脉冲序列的总行数
对应菜单	Analog Mod > 脉冲 > 脉冲序列 > 删除
示例	*:SOURce1:PULM:TRAin:DELete 5*

3.4.8.3.12 编辑脉冲序列的正脉宽（[:SOURce<ch>]:PULM:TRAin:DATA:ONTIme）

命令格式	[:SOURce<ch>]:PULM:TRAin:DATA:ONTIme <raw>,<on_time>
功能描述	编辑通道 ch 的脉冲序列中指定行的正脉宽。
参数类型	行号：整型， 正脉宽：浮点型，单位：ns，us，ms 或 s，默认为 s
参数范围	行号：1 ~ 当前脉冲序列的总行数， 正脉宽：20 ns ~ 300 s
对应菜单	Analog Mod > 脉冲 > 脉冲序列
示例	*:SOURce1:PULM:TRAin:DATA:ONTIme 1,10 ms*

3.4.8.3.13 编辑脉冲序列的负脉宽（[:SOURce<ch>]:PULM:TRAin:DATA:OFFTime）

命令格式	[:SOURce<ch>]:PULM:TRAin:DATA:OFFTime <raw>,<off_time>
功能描述	编辑通道 ch 的脉冲序列中指定行的负脉宽。
参数类型	行号：整型， 负脉宽：浮点型，单位：ns，us，ms 或 s，默认为 s
参数范围	行号：1 ~ 当前脉冲序列的总行数， 负脉宽：20 ns ~ 300 s
对应菜单	Analog Mod > 脉冲 > 脉冲序列
示例	*:SOURce1:PULM:TRAin:DATA:OFFTime 1, 20 ms*

3.4.8.3.14 编辑脉冲序列的重复次数（[:SOURce<ch>]:PULM:TRAin:DATA:COUNt）

命令格式	[:SOURce<ch>]:PULM:TRAin:DATA:COUNt <raw>,<count>
功能描述	编辑通道 ch 的脉冲序列指定行的重复次数。
参数类型	行号：整型， 重复次数：整型
参数范围	行号：1 ~ 当前脉冲序列的总行数， 重复次数：1 ~ 65535
对应菜单	Analog Mod > 脉冲 > 脉冲序列
示例	*:SOURce1:PULM:TRAin:DATA:COUNt 1,3*

3.4.8.3.15 编辑脉冲序列（[:SOURce<ch>]:PULM:TRAin:CHANGe）

命令格式	[:SOURce<ch>]:PULM:TRAin:CHANGe <row>,<on_time>,<off_time>,<count>
功能描述	编辑通道 ch 的脉冲序列中的指定行。
参数类型	行号：整型， 正脉宽：浮点型，单位：ns，us，ms 或 s，默认为 s， 负脉宽：浮点型，单位：ns，us，ms 或 s，默认为 s， 重复次数：整型
参数范围	行号：1 ~ 扫描列表的总行数， 正脉宽：20 ns ~ 300 s， 负脉宽：20 ns ~ 300 s， 重复次数：1 ~ 65535
对应菜单	Analog Mod > 脉冲 > 脉冲序列
示例	*:SOURce1:PULM:TRAin:CHANGe 2,3 ms,500 ns,4*

3.4.8.3.16 查询脉冲序列数据（[:SOURce<ch>]:PULM:TRAin:LIST?）

命令格式	[:SOURce<ch>]:PULM:TRAin:LIST? <begin_row>,<end_row>
功能描述	查询通道 ch 的脉冲序列中 begin_ row 到 end_row 的数据。
参数类型	整型，整型
参数范围	1 ~ 脉冲序列的总行数， 开始行 ~ 脉冲序列的总行数
返回值	字符串
对应菜单	Analog Mod > 脉冲 > 脉冲序列
示例	*:SOURce1:PULM:TRAin:LIST? 1,3* 返回值： *0.001,0.005,1\s0.003,0.003,2\s0.004,0.002,1\n*

3.4.8.3.17 查询脉冲序列行数（[:SOURce<ch>]:PULM:TRAin:COUNt?）

命令格式	[:SOURce<ch>]:PULM:TRAin:COUNt?
功能描述	查询通道 ch 的脉冲序列的行数。
对应菜单	Analog Mod > 脉冲 > 脉冲序列
示例	*:SOURce1:PULM:TRAin:COUNt?* 返回值： *5\n*

3.4.8.3.18 清空脉冲序列（[:SOURce<ch>]:PULM:TRAin:CLEAr）

命令格式	[:SOURce<ch>]:PULM:TRAin:CLEAr
功能描述	清空通道 ch 的脉冲序列，恢复默认值。
对应菜单	Analog Mod > 脉冲 > 脉冲序列 > 清空
示例	*:SOURce1:PULM:TRAin:CLEAr*

3.4.8.3.19 加载脉冲序列（[:SOURce<ch>]:PULM:TRAin:LOAD）

命令格式	[:SOURce<ch>]:PULM:TRAin:LOAD <"file_name">
功能描述	从文件中加载通道 ch 的脉冲序列。
参数类型	字符串
参数范围	无
对应菜单	Analog Mod > 脉冲 > 脉冲序列 > 加载
示例	*:SOURce1:PULM:TRAin:LOAD "U-disk3/test.pulstrn"*

3.4.8.3.20 保存脉冲序列（[:SOURce<ch>]:PULM:TRAin:STORE）

命令格式	[:SOURce<ch>]:PULM:TRAin:STORE <"file_name">
功能描述	将通道 ch 的脉冲序列保存到文件。
参数类型	字符串
参数范围	无
对应菜单	Analog Mod > 脉冲 > 脉冲序列 > 保存
示例	*:SOURce1:PULM:TRAin:STORE "test.pulstrn"* *:SOURce1:PULM:TRAin:STORE "U-disk1/test.pulstrn"*

3.4.8.3.21 脉冲触发输出状态（[:SOURce<ch>]:PULM:TRIGger:STATe）

命令格式	[:SOURce<ch>]:PULM:TRIGger:STATe ON|OFF|1|0 [:SOURce<ch>]:PULM:TRIGger:STATe?
功能描述	设置通道 ch 的脉冲触发输出状态。 查询通道 ch 的脉冲触发输出状态。
参数类型	布尔型

参数范围	ON|OFF|1|0
返回值	1|0
默认值	1
对应菜单	Analog Mod > 脉冲 > 触发输出
示例	*:SOURce1:PULM:TRIGger:STATe OFF* *:SOURce1:PULM:TRIGger:STATe?* 返回值： *0\n*

3.4.8.3.22 脉冲触发方式（[:SOURce<ch>]:PULM:TRIGger:MODE）

命令格式	[:SOURce<ch>]:PULM:TRIGger:MODE AUTO|KEY|EXTernal|EGATe [:SOURce<ch>]:PULM:TRIGger:MODE?
功能描述	设置通道 ch 的脉冲触发方式为自动、按键、总线、外部触发或外部门控。 查询通道 ch 的脉冲触发方式。
参数类型	枚举
参数范围	AUTO|KEY|EXTernal|EGATe
返回值	枚举
默认值	AUTO
对应菜单	Analog Mod > 脉冲 > 触发方式
示例	*:SOURce1:PULM:TRIG:MODE EXTernal* *:SOURce1:PULM:TRIGger:MODE?* 返回值： *EXTernal\n*

3.4.8.3.23 脉冲外触发延迟（[:SOURce<ch>]:PULM:DELay）

命令格式	[:SOURce<ch>]:PULM:DELay <value> [:SOURce<ch>]:PULM:DELay?
功能描述	设置通道 ch 的脉冲外部触发的触发延迟时间。 查询通道 ch 的脉冲外部触发的触发延迟时间。
参数类型	浮点型， 单位：ns，us，ms 或 s，默认为 s
参数范围	140 ns ~ 300 s
返回值	浮点型，单位：s
默认值	140 ns

对应菜单	Analog Mod > 脉冲 > 触发延迟
示例	*:SOURce1:PULM:DEL 30 ms* *:SOURce1:PULM:DELay?* 返回值: *0.03\n*

3.4.8.3.24 脉冲外触发沿（[:SOURce<ch>]:PULM:TRIGger:EXTernal:SLOPe）

命令格式	[:SOURce<ch>]:PULM:TRIGger:EXTernal:SLOPe NEGative|POSitive [:SOURce<ch>]:PULM:TRIGger:EXTernal:SLOPe?
功能描述	设置通道 ch 的脉冲外部触发的触发沿。 查询通道 ch 的脉冲外部触发的触发沿。
参数类型	枚举
参数范围	NEGative|POSitive
返回值	枚举
默认值	POSitive
对应菜单	Analog Mod > 脉冲 > 触发沿
示例	*:SOURce1:PULM:TRIG:EXT:SLOP NEG* *:SOURce1:PULM:TRIG:EXT:SLOP?* 返回值: *NEGative\n*

3.4.8.3.25 脉冲外部门控触发极性（[:SOURce<ch>]:PULM:TRIGger:EXTernal:GATE:POLarity）

命令格式	[:SOURce<ch>]:PULM:TRIGger:EXTernal:GATE:POLarity NORMal|INVerted [:SOURce<ch>]:PULM:TRIGger:EXTernal:GATE:POLarity?
功能描述	设置通道 ch 的脉冲外部门控触发的触发极性。 查询通道 ch 的脉冲外部门控触发的触发极性。
参数类型	枚举
参数范围	NORMal|INVerted
返回值	枚举
默认值	NORMal
对应菜单	Analog Mod > 脉冲 > 触发极性
示例	*:SOURce1:PULM:TRIG:EXT:GATE:POL INVerted* *:SOURce1:PULM:TRIGger:EXTernal:GATE:POLarity?* 返回值: *INVerted\n*

3.4.9 相位相干

3.4.9.1 相位相干（[:SOURce]:PHASe:COHerent）

命令格式	[:SOURce]:PHASe:COHerent ON|OFF|1|0 [:SOURce]:PHASe:COHerent?
功能描述	设置 RF 相位相干功能开关。 查询 RF 相位相干功能开关。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	ON|OFF|1|0
默认值	0
对应菜单	HOME >  > 
示例	*:SOURce:PHASe:COHerent 1* *:SOURce:PHASe:COHerent?* 返回值： *1\n*

3.4.9.2 相位相干选择通道（[:SOURce<ch>]:PHASe:COHerent:CHANnel）

命令格式	[:SOURce<ch>]:PHASe:COHerent:CHANnel ON|OFF|1|0 [:SOURce<ch>]:PHASe:COHerent:CHANnel?
功能描述	设置 RF 通道在相位相干功能开启时是否与其他通道相位相干。 查询 RF 通道在相位相干功能开启时是否与其他通道相位相干。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	ON|OFF|1|0
默认值	0
对应菜单	HOME >  > 
示例	*:SOURce1:PHASe:COHerent:CHANnel 1* *:SOURce:PHASe:COHerent:CHANne?* 返回值： *1\n*

3.4.9.3 相位相干频率配置（[:SOURce]:PHASe:COHerent:FREQuency）

命令格式	[:SOURce]:PHASe:COHerent:FREQuency 22E9 [:SOURce]:PHASe:COHerent:FREQuency?
功能描述	相位相干时要求通道的频率相同 设置 RF 通道用于同步所有通道的相同频率值。 查询 RF 通道用于同步所有通道的相同频率值。
参数类型	浮点型
参数范围	RF 通道最小频率范围~ RF 通道最大频率范围
返回值	配置频率值
默认值	22000000000
对应菜单	HOME >  > 
示例	*:SOURce:PHASe:COHerent:FREQuency 22E9* *:SOURce:PHASe:COHerent:FREQuency?* 返回值： *22000000000\n*

3.4.9.4 相位相干同步通道频率（[:SOURce]:PHAse:COHerent:SYNC:FREQuency）

命令格式	:SOURce:PHAse:COHerent:SYNC:FREQuency
功能描述	同步选中的相位相干功能通道的频率，频率值为相位相干频率配置值
参数类型	无
参数范围	无
返回值	无
默认值	无
对应菜单	HOME >  > 
示例	:PHAse:COHerent:SYNC:FREQuency

3.4.10 系统复位

3.4.10.1 系统复位（:SOURce:PRESet）

命令格式	:SOURce:PRESet
功能描述	将仪器状态设置为出厂设置。
对应菜单	无
示例	*:SOURce:PRESet*

3.5 SENSe 命令子系统

3.5.1 功率计设置

3.5.1.1 查询功率计信息（:SENSe[:POWer]:TYPE?）

命令格式	:SENSe[:POWer]:TYPE?
功能描述	查询连接到多通道微波信号源的功率计的型号。
返回值	字符串
默认值	无
对应菜单	HOME >  >  > 功率计信息
示例	*SENSe:TYPE?* 返回值： *NRP6A\n*

3.5.1.2 功率计状态（:SENSe[:POWer]:STATus）

命令格式	:SENSe[:POWer]:STATus OFF|ON|0|1 :SENSe[:POWer]:STATus?
功能描述	设置功率计的测量状态为打开或关闭。 查询功率计的测量状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	HOME >  >  >功率计状态
示例	*SENSe:STATus ON* *SENSe:STATus?* 返回值： *1\n*

3.5.1.3 查询功率计测量值（:SENSe[:POWer]:VALue?）

命令格式	:SENSe[:POWer]:VALue?
功能描述	查询功率计的测量值。

返回值	浮点型，单位：dBm
默认值	无
对应菜单	HOME >  >  >测量功率
示例	*SENSe:VALue?* 返回值： *-0.02600282\n*

3.5.1.4 功率计测量统计状态（:SENSe[:POWer]:STATIStics:STATe）

命令格式	:SENSe[:POWer]:STATIStics:STATe ON|OFF|1|0 :SENSe[:POWer]:STATIStics:STATe?
功能描述	设置功率计的测量统计状态为打开或关闭。 查询功率计的测量统计状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	HOME >  >  >统计功能
示例	*SENSe:STATIStics:STATe ON* *SENSe:STATIStics:STATe?* 返回值： *1\n*

3.5.1.5 查询功率计测量统计值（:READ[:POWer]?）

命令格式	:READ[:POWer]?
功能描述	查询功率计测量统计值的平均值和最大值。
返回值	字符串 字符串形式：平均值，最大值
默认值	无
对应菜单	HOME >  >  >统计功能 > 平均值/最大值
示例	*READ?* 返回值： *-0.05,-0.04\n*

3.5.1.6 查询功率计测量统计的最大值（:SENSe[:POWer]:STATIStics:MAX?）

命令格式	:SENSe[:POWer]:STATIStics:MAX?
功能描述	查询功率计测量统计的最大值。
返回值	浮点型，单位：dBm
默认值	无
对应菜单	HOME >  >  >统计功能 > 最大值
示例	*SENSe:STATIStics:MAX?* 返回值： *-0.03117205\n*

3.5.1.7 查询功率计测量统计的最小值（:SENSe[:POWer]:STATIStics:MIN?）

命令格式	:SENSe[:POWer]:STATIStics:MIN?
功能描述	查询功率计测量统计的最小值。
返回值	浮点型，单位：dBm
默认值	无
对应菜单	HOME >  >  >统计功能 > 最小值
示例	*SENSe:STATIStics:MIN?* 返回值： *-0.06101395\n*

3.5.1.8 查询功率计测量统计的平均值（:SENSe[:POWer]:STATIStics:AVG?）

命令格式	:SENSe[:POWer]:STATIStics:AVG?
功能描述	查询功率计测量统计的平均值。
返回值	浮点型，单位：dBm
默认值	无
对应菜单	HOME >  >  >统计功能 > 平均值
示例	*SENSe:STATIStics:AVG?* 返回值： *-0.0322136383619456\n*

3.5.1.9 查询功率计测量统计的计数（:SENSe[:POWer]:STATIStics:COUNt?）

命令格式	:SENSe[:POWer]:STATIStics:COUNt?
功能描述	查询功率计测量统计的计数。
返回值	整型
默认值	无
对应菜单	HOME >  >  >统计功能 > 计数
示例	*SENSe:STATIStics:COUNt?* 返回值： *2035\n*

3.5.1.10 清空功率计的测量统计（:SENSe[:POWer]:STATIStics:CLEAr）

命令格式	:SENSe[:POWer]:STATIStics:CLEAr
功能描述	清空功率计的测量统计值。
对应菜单	HOME >  >  >统计功能 > 清空
示例	*SENSe:STATIStics:CLEAr*

3.5.1.11 功率计功率控制状态（:SENSe[:POWer]:LEV:CTL:STATe）

命令格式	:SENSe[:POWer]:LEV:CTL:STATe ON|OFF|1|0 :SENSe[:POWer]:LEV:CTL:STATe?
功能描述	设置功率计测量的功率控制状态。 查询功率计测量的功率控制状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
等效命令	[:SOURce<ch>]:POWer:SPC:STATe ON|OFF|1|0 [:SOURce<ch>]:POWer:SPC:STATe?
对应菜单	HOME >  >  >功率控制
示例	*:SENSe:LEV:CTL:STATe OFF* *:SENSe:LEV:CTL:STATe?*

	返回值： *0\n*

3.5.1.12 功率控制目标功率（:SENSe[:POWer]:SPC:TARGet）

命令格式	:SENSe[:POWer]:SPC:TARGet <power> :SENSe[:POWer]:SPC:TARGet?
功能描述	设置功率计功率控制的目标功率。 查询功率计功率控制的目标功率。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	-120 dBm ~ 20 dBm
返回值	浮点型，单位：dBm
默认值	0 dBm
等效命令	[:SOURce<ch>]:POWer:SPC:TARGet <power> [:SOURce<ch>]:POWer:SPC:TARGet?
对应菜单	HOME >  >  >功率控制 > 目标功率
示例	*SENSe:SPC:TARGet -6* *SENSe:SPC:TARGet?* 返回值： *-6\n*

3.5.1.13 功率计控制通道选择（:SENSe[:POWer]:LOGGing:STATe）

命令格式	:SENSe[:POWer]:LEV:CTL:CHANnel :SENSe[:POWer]:LEV:CTL:CHANnel? [:SOURce]:POWer:LEV:CTL:CHANnel [:SOURce]:POWer:LEV:CTL:CHANnel?
功能描述	设置功率计控制的通道。 查询功率计控制的通道。
参数类型	整型
参数范围	双通道：1|2 四通道：1|2|3|4
返回值	1|0
默认值	0
对应菜单	HOME >  >  >日志

示例	*:SENSe[:POWer]:LEV:CTL:CHANnel 1* *:SENSe[:POWer]:LEV:CTL:CHANnel?* 返回值： *1\n*

3.5.1.14 功率控制限制功率（:SENSe[:POWer]:LIMit）

命令格式	:SENSe[:POWer]:LIMit <power> :SENSe[:POWer]:LIMit?
功能描述	设置功率计功率控制的限制功率。 查询功率计功率控制的限制功率。
参数类型	浮点型， 单位：dBm，dBuV，uV，mV，V，nW，uW，mW 或 W，默认为 dBm
参数范围	-120 dBm ~ 20 dBm
返回值	浮点型，单位：dBm
默认值	0 dBm
等效命令	[:SOURce<ch>]:POWer:LIMit <power> [:SOURce<ch>]:POWer:LIMit?
对应菜单	HOME >  >  >功率控制 > 限制功率
示例	*SENSe:LIMit 2* *SENSe:LIMit?* 返回值： *2\n*

3.5.1.15 功率控制捕获范围（:SENSe[:POWer]:SPC:CRANge）

命令格式	:SENSe[:POWer]:SPC:CRANge <power> :SENSe[:POWer]:SPC:CRANge?
功能描述	设置功率计功率控制的捕获范围。 查询功率计功率控制的捕获范围。
参数类型	浮点型，单位：dB
参数范围	0 dB ~ 50 dB
返回值	浮点型，单位：dB
默认值	0 dB
等效命令	[SOURce]:POWer:SPC:CRANge <power> [SOURce]:POWer:SPC:CRANge?

对应菜单	HOME >  >  >功率控制 > 捕获范围
示例	*:SENSe:SPC:CRANge 10* *:SENSe:SPC:CRANge?* 返回值: *10\n*

3.5.1.16 功率计自动调零类型（:CALibration:ZERO:TYPE）

命令格式	:CALibration:ZERO:TYPE INTernal|EXTernal :CALibration:ZERO:TYPE?
功能描述	设置功率计的自动调零类型。 查询功率计的自动调零类型。
参数类型	枚举
参数范围	INTernal|EXTernal
返回值	枚举
默认值	INTernal
对应菜单	HOME >  >  >自动调零
示例	*:CALibration:ZERO:TYPE EXTernal* *:CALibration:ZERO:TYPE?* 返回值: *EXTernal\n*

3.5.1.17 执行清零（:SENSe[:POWer]:ZERO）

命令格式	:SENSe[:POWer]:ZERO
功能描述	对功率计执行清零操作。
对应菜单	HOME >  >  >执行清零
示例	*:SENSe:ZERO*

3.5.1.18 功率计测量频率类型（:SENSe[:POWer]:SOURce）

命令格式	:SENSe[:POWer]:SOURce RF|USER :SENSe[:POWer]:SOURce?
功能描述	设置功率计的测量频率类型。

	查询功率计的测量频率类型。
参数类型	枚举
参数范围	RF|USER
返回值	枚举
默认值	RF
对应菜单	HOME >  >  >测量频率
示例	*SENSe:SOURce USER* *SENSe:SOURce?* 返回值: *USER\n*

3.5.1.19 功率计手动测量频率（:SENSe[:POWer]:FREQuency）

命令格式	:SENSe[:POWer]:FREQuency <freq> :SENSe[:POWer]:FREQuency?
功能描述	设置功率计的手动测量频率。 查询功率计的手动测量频率。
参数类型	浮点型， 单位：Hz，kHz，MHz 或 GHz，默认为 Hz
参数范围	功率计测量量程
返回值	浮点型，单位：Hz
默认值	无
对应菜单	HOME >  >  >测量频率 > 手动
示例	*SENSe:FREQuency 1 MHz* *SENSe:FREQuency?* 返回值: *1000000\n*

3.5.1.20 功率计幅度偏移状态（:SENSe[:POWer]:OFFSet:STATe）

命令格式	:SENSe[:POWer]:OFFSet:STATe ON|OFF|1|0 :SENSe[:POWer]:OFFSet:STATe?
功能描述	设置功率计的幅度偏移状态。 查询功率计的幅度偏移状态。
参数类型	布尔型

参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	HOME >  >  >幅度偏移
示例	*SENSe:OFFSet:STATe ON* *SENSe:OFFSet:STATe?* 返回值： *1\n*

3.5.1.21 功率计幅度偏移（:SENSe[:POWer]:OFFSet）

命令格式	:SENSe[:POWer]:OFFSet <power> :SENSe[:POWer]:OFFSet?
功能描述	设置功率计的幅度偏移。 查询功率计的幅度偏移。
参数类型	浮点型，单位：dB
参数范围	-200 ~ 200
返回值	浮点型，单位：dB
默认值	0
对应菜单	HOME >  >  >幅度偏移
示例	SENSe:OFFSet 10 SENSe:OFFSet? 返回值： 10\n

3.5.1.22 功率计平均类型（:SENSe[:POWer]:FILTer:TYPE）

命令格式	:SENSe[:POWer]:FILTer:TYPE AUTO|USER|NSRatio :SENSe[:POWer]:FILTer:TYPE?
功能描述	设置功率计的测量平均类型。 查询功率计的测量平均类型。
参数类型	枚举
参数范围	AUTO|USER|NSRatio
返回值	枚举
默认值	AUTO

对应菜单	HOME >  >  >平均类型
示例	*SENSe:FILTer:TYPE USER* *SENSe:FILTer:TYPE?* 返回值： *USER\n*

3.5.1.23 功率计平均次数（:SENSe[:POWer]:FILTer:LENGth）

命令格式	:SENSe[:POWer]:FILTer:LENGth <length> :SENSe[:POWer]:FILTer:LENGth?
功能描述	设置功率计的测量平均次数。 查询功率计的测量平均次数。
参数类型	整型
参数范围	1 ~ 65536
返回值	整型
默认值	无
对应菜单	HOME >  >  >平均次数
示例	*SENSe:FILTer:LENGth 10* *SENSe:FILTer:LENGth?* 返回值： *10\n*

3.5.1.24 功率计固定噪声（:SENSe[:POWer]:FILTer:NSRatio）

命令格式	:SENSe[:POWer]:FILTer:NSRatio <noise> :SENSe[:POWer]:FILTer:NSRatio?
功能描述	功率计的测量平均类型为固定噪声时，设置功率计的固定噪声。 查询功率计的固定噪声。
参数类型	浮点型，单位：dB
参数范围	无
返回值	浮点型，单位：dB
默认值	无
对应菜单	HOME >  >  >固定噪声
示例	*SENSe:FILTer:NSRatio 1* *SENSe:FILTer:NSRatio?*

	返回值： *1\n*

3.5.1.25 功率计日志状态（:SENSe[:POWer]:LOGGing:STATe）

命令格式	:SENSe[:POWer]:LOGGing:STATe ON|OFF|1|0 :SENSe[:POWer]:LOGGing:STATe?
功能描述	设置功率计的日志记录状态。 查询功率计的日志记录状态。
参数类型	布尔型
参数范围	ON|OFF|1|0
返回值	1|0
默认值	0
对应菜单	HOME >  >  >日志
示例	*SENSe:LOGGing:STATe ON* *SENSe:LOGGing:STATe?* 返回值： *1\n*

4 编程示例

本章为程序员提供了一些示例。在这些示例中，您可以了解如何将 VISA 或 Sockets 以及上述 SCPI 命令来控制多通道微波信号发生器。通过遵循这些示例，您可以开发更多的应用程序。

4.1 使用 VISA 的编程示例

4.1.1 VC++示例

系统环境： Windows 10，64 位操作系统。

编程软件： Visual Studio

示例内容： 使用 NI-VISA 通过 USBTMC 或 TCP/IP 访问控制设备，并执行读写操作。

请按照以下步骤完成示例：

1、 打开 Visual Studio，新建一个 VC++ win32 控制台项目。

2、 设置项目环境使用 NI-VISA 库，使用 NI-VISA 库有两种方式，静态或自动：

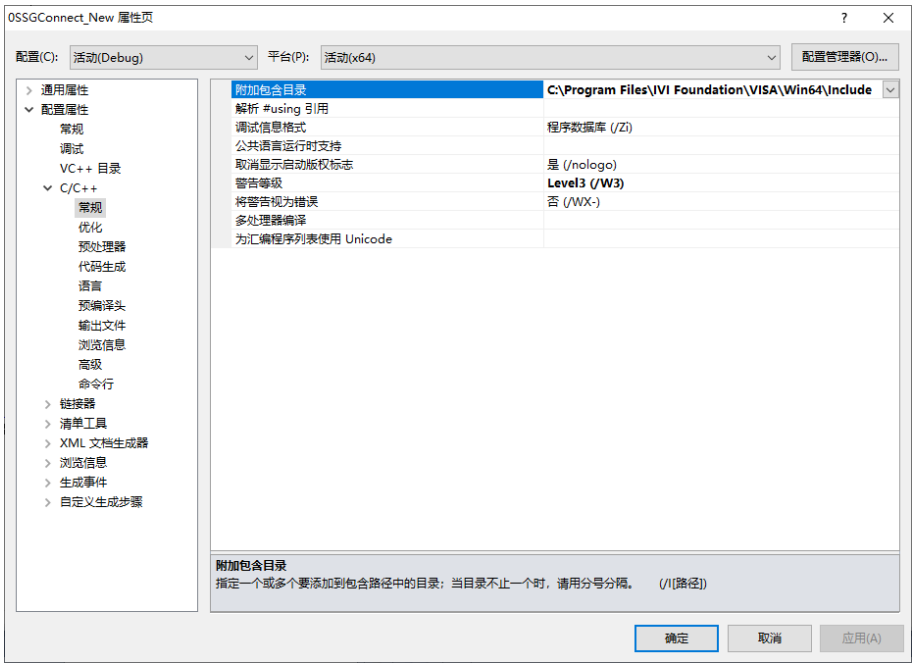
(1) 静态：

在 NI-VISA 的安装路径中查找文件：visa.h、visatype.h、visa32.lib，其中 NI-VISA 的默认安装路径为：C:\Program Files\IVI Foundation\VISA\Win64\。将上述文件复制到您的项目中，并将它们添加到项目中。然后在项目.cpp 文件中添加以下两行代码：

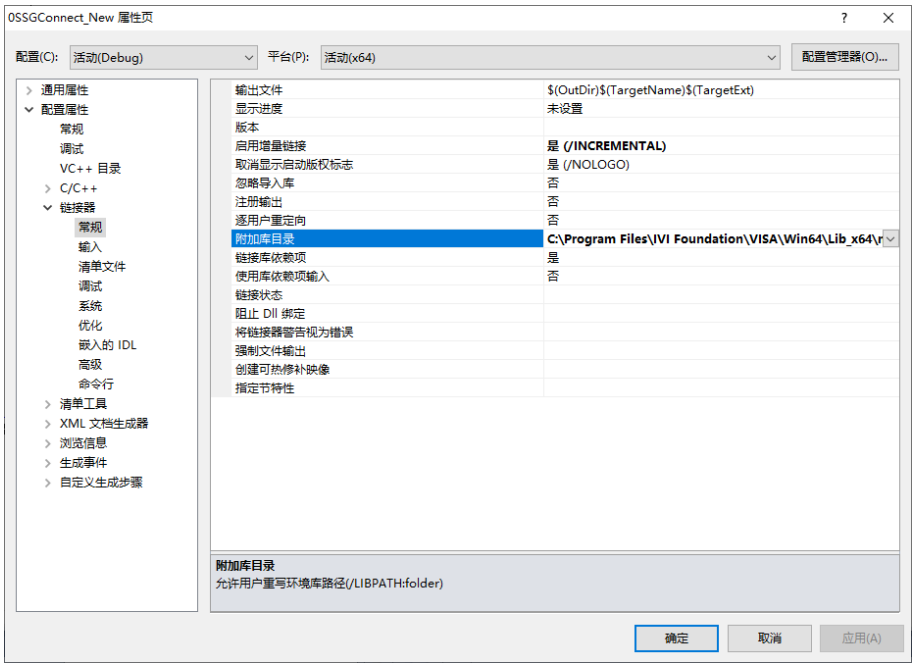
```
#include "visa.h"
#pragma comment(lib,"visa32.lib")
```

(2) 动态：

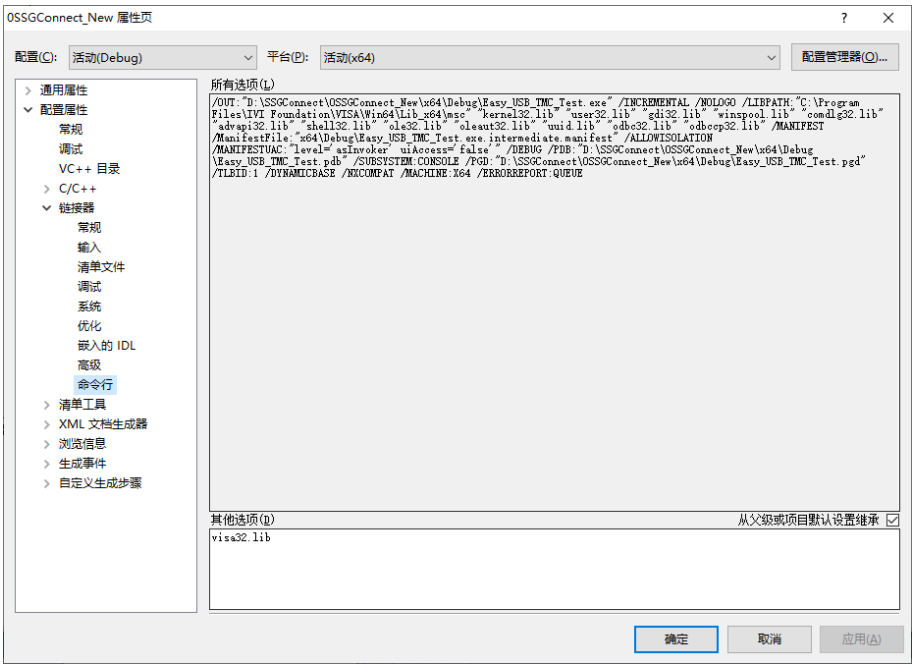
设置.h 头文件的包含目录。NI-VISA 的头文件默认安装路径为：C:\Program Files\IVI Foundation\VISA\Win64\Include。在项目 --- 属性 --- C/C++ --- 常规 --- 附加包含目录中填入头文件安装路径，如下图示：



设置 lib 库文件的包含目录。NI-VISA 的库文件默认安装路径为：C:\Program Files\IVI Foundation\VISA\Win64\Lib_x64\msc。在项目 --- 属性 --- 链接器 --- 常规 --- 附加库目录中填入库文件安装路径，如下图示：



设置库文件。在项目 --- 属性 --- 链接器 --- 命令行 --- 其它选项中填入库文件名称，如下图示：



最后在项目.cpp 文件中引用头文件：

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

3、 增加代码

(1) USBTMC 接口访问代码：

```
int Usbtmc_test()
{
    /*这段代码演示了使用 NI-VISA 发送同步读取和写入命令到          */
    /*一个 USB 测试&测量类(USBTMC)仪器。                          */
    /*这个例子写"* IDN ?\ n "字符串到所有连接到系统的 USBTMC      */
    /*设备并试图使用读写函数读回结果。                            */
    /*代码的一般流程是打开资源管理器                              */
    /*打开 VISA 会话到仪器                                        */
    /*使用 viPrintf 写仪器标志查询                                */
    /*尝试随 viScanf 读取一个响应                                 */
    /*关闭 VISA 会话                                              */
    /*************************************************************/
    ViSession    defaultRM;
    ViSession    instr;
```

```
ViUInt32    numInstrs;
ViFindList    findList;
ViStatus    status;
char      instrResourceString[VI_FIND_BUFLEN];
unsigned char    buffer[100];
int    i;
/*首先，我们必须调用 viOpenDefaultRM 得到管理器的句柄。       */
/*我们将在 defaultRM 存储此手柄。                              */
status=viOpenDefaultRM (&defaultRM);
if (status<VI_SUCCESS)
{
        printf ("Could not open a session to the VISA Resource Manager!\n");
        return    status;
}
/*寻找我们的系统中所有的 USB TMC VISA 资源                     */
/*和然后将资源的数目存储在系统中的 numInstrs 里。               */
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;
}
/**现在，我们将对所有 USB TMC 仪器打开 VISA 会话。我们必须
*  从 viOpenDefaultRM 使用句柄，也必须使用一个字符串指示要
*打开的仪器，这就是所谓的仪器描述符。该字符串的格式可以在功
*能面板中右键单击参数描述中找到。打开一个会话到设备后，我们
*将得到一个仪器使用的句柄，在之后使用 VISA 功能时用到。在这
*个函数的 AccessMode 和超时参数是为将来的功能预留。这两个参
*数被给予值 VI_NULL。*/
```

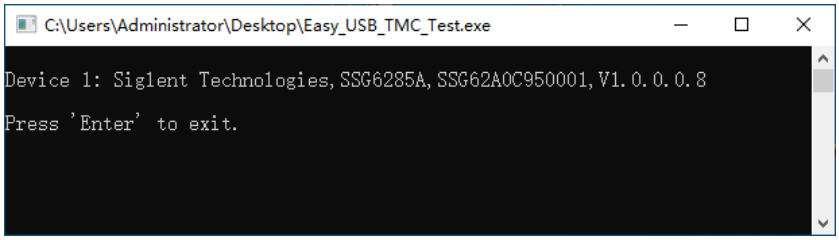
```
for (i=0; i<int(numInstrs); i++)
{
        if (i> 0)
        {
                viFindNext (findList, instrResourceString);
        }
        status = viOpen (defaultRM, instrResourceString, VI_NULL, VI_NULL, &instr);
        if (status<VI_SUCCESS)
        {
                printf ("Cannot open a session to the device %d.\n", i+1);
                continue;
        }
        /* *在这一点上，我们现在有一个会话打开到 USB TMC 仪器。现在，
        *我们将使用 viPrintf 函数发送字符串“*IDN?\n”到设备，要求设备识别。*/
        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;
        }
        /**  现在我们将尝试从设备读回一个设备信息查询的响应。我们将
        *使用 viScanf 函数来获取数据。在数据被读出后，响应显示出来   */
        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);

    /**现在，我们将关闭会话使用 viClose 仪器。此操作释放所有系统资源。*/
    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：



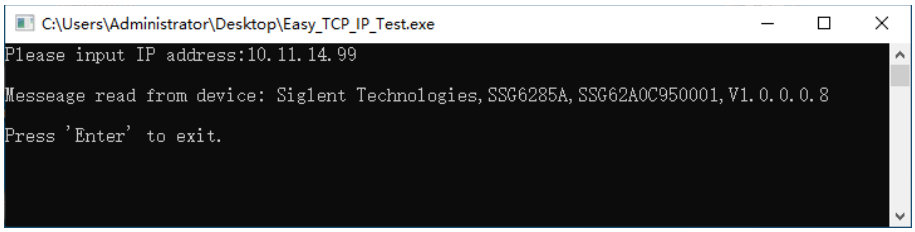
(2) TCP/IP 接口访问代码：

```
int TCP_IP_Test(char *pIP)
{
    char outputBuffer[VI_FIND_BUFLEN];
    ViSession defaultRM, instr;
    ViStatus status;
    /* 首先，我们需要打开默认的资源管理器。                    */
    status = viOpenDefaultRM (&defaultRM);
```

```
if (status<VI_SUCCESS)
{
        printf("Could not open a session to the VISA Resource Manager!\n");
}
/*现在，我们将通过 TCP / IP 设备打开一个会话          */
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：



4.1.2 VB 示例

系统环境：Windows 7

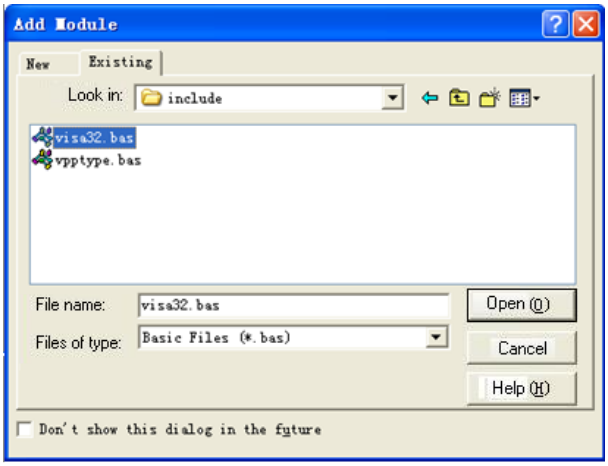
编程软件：Microsoft Visual Basic 6.0

示例内容：使用 NI-VISA 通过 USBTMC 或 TCP/IP 访问控制设备，并执行读写操作。

请按照以下步骤完成示例：

1、 打开 Visual Basic，建立一个标准的应用程序项目（标准 EXE）。

2、 设置项目环境中使用 NI-VISA 库。单击项目的现有标签>>添加模块，在 NI-VISA 安装路径下的 include 文件夹中搜索 visa32.bas 文件并添加该文件。这将使得 VISA 功能和 VISA 的数据类型在程序中使用。



3、 添加代码

(1) USBTMC 接口访问代码

```
Private Function Usbtmc_test() As Long
'这段代码演示了使用 NI-VISA 发送同步读取和写入命令到
'一个 USB 测试&测量类(USBTMC)仪器。
'这个例子写“* IDN ?\ n "字符串到所有连接到系统的 USBTMC
'设备并试图使用读写函数读回结果。
'代码的一般流程是打开资源管理器
'打开 VISA 会话到仪器
'使用 viPrintf 写仪器标志查询
'尝试随 viScanf 读取一个响应
'关闭 VISA 会话
    Const MAX_CNT = 200


    Dim defaultRM As Long
    Dim instrsesn As Long
    Dim numInstrs As Long
    Dim findList As Long
    Dim retCount As Long
    Dim status As Long
    Dim instrResourceString As String * VI_FIND_BUFLEN
    Dim Buffer As String * MAX_CNT
```

```
    Dim i As Integer

'首先，我们必须调用 viOpenDefaultRM 得到管理器的句柄。
'我们将在 defaultRM 存储此手柄。
    status = viOpenDefaultRM(defaultRM)
    If (status < VI_SUCCESS) Then
        resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
        Usbtmc_test = status
        Exit Function
    End If

'寻找我们的系统中所有的 USB TMC VISA 资源
'和然后将资源的数目存储在系统中的 numInstrs 里。
    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
        Exit Function
    End If

'现在，我们将对所有 USB TMC 仪器打开 VISA 会话。我们必须
'从 viOpenDefaultRM 使用句柄， 也必须使用一个字符串指示要
'打开的仪器，这就是所谓的仪器描述符。该字符串的格式可以在功
'能面板中右键单击参数描述中找到。打开一个会话到设备后，我们
'将得到一个仪器使用的句柄，在之后使用 VISA 功能时用到。在这
'个函数的 AccessMode 和超时参数是为将来的功能预留。这两个参
'数被给予值 VI_NULL。
    For i = 0 To numInstrs
        If (i > 0) Then
            status = viFindNext(findList, instrResourceString)
        End If
```

```
        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
        End If

    '在这一点上，我们现在有一个会话打开到 USB TMC 仪器。现在，
    '我们将使用 viPrintf 函数发送字符串“*IDN?\n”到设备，要求设备识别。
        status = viWrite(instrsesn, "*IDN?", 5, retCount)
        If (status < VI_SUCCESS) Then
            resultTxt.Text = "Error writing to the device."
            status = viClose(instrsesn)
            GoTo NextFind
        End If

    '现在我们将尝试从设备读回一个设备信息查询的响应。我们将
    '使用 viScanf 函数来获取数据。在数据被读出后，响应显示出来
        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
        End If
        status = viClose(instrsesn)
    Next i

    '现在，我们将关闭会话使用 viClose 仪器。此操作释放所有系统资源。
    status = viClose(defaultRM)
    Usbtmc_test = 0
End Function
```

(2) TCP/IP 接口访问代码

```
Private Function TCP_IP_Test(ByVal ip As String) As Long
    Dim outputBuffer As String * VI_FIND_BUFLEN
    Dim defaultRM As Long
    Dim instrsesn As Long
    Dim status As Long
    Dim count As Long

    '首先，我们需要打开默认的资源管理器。
    status = viOpenDefaultRM(defaultRM)
    If (status < VI_SUCCESS) Then
        resultTxt.Text = "Could not open a session to the VISA Resource Manager!"
        TCP_IP_Test = status
        Exit Function
    End If

    '现在，我们将通过 TCP / IP 设备打开一个会话
        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
        Exit Function
    End If

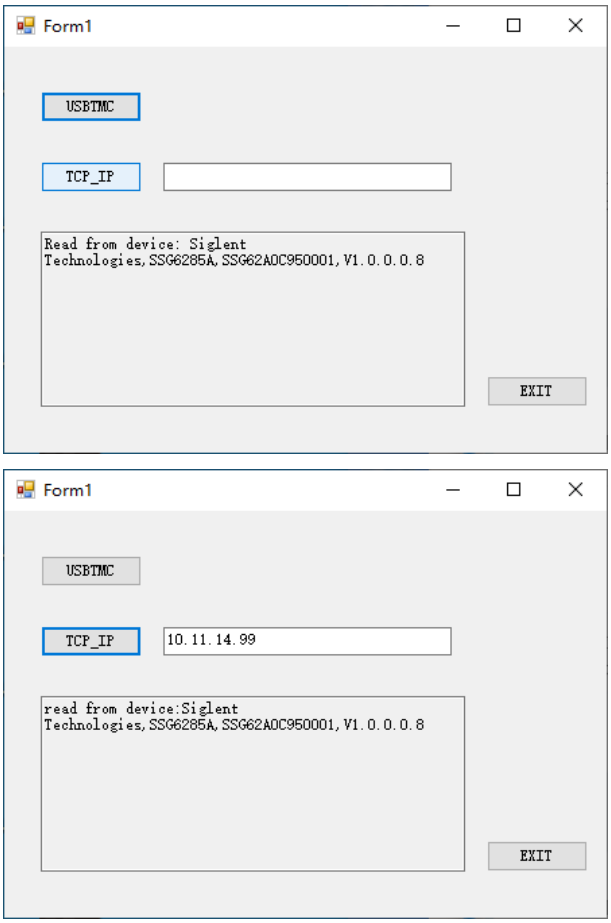
    status = viWrite(instrsesn, "*IDN?", 5, count)
    If (status < VI_SUCCESS) Then
        resultTxt.Text = "Error writing to the device."
    End If
    status = viRead(instrsesn, outputBuffer, VI_FIND_BUFLEN, count)
    If (status < VI_SUCCESS) Then
        resultTxt.Text = "Error reading a response from the device." + CStr(i + 1)
```

```
        Else
            resultTxt.Text = "read from device:" + outputBuffer
        End If
        status = viClose(instrsesn)
        status = viClose(defaultRM)
        TCP_IP_Test = 0
    End Function
```

(3) 按键控制代码

```
    Private Sub exitBtn_Click()
        End
    End Sub
    Private Sub tcpipBtn_Click()
        Dim stat As Long
        stat = TCP_IP_Test(ipTxt.Text)
        If (stat < VI_SUCCESS) Then
            resultTxt.Text = Hex(stat)
        End If
    End Sub
    Private Sub usbBtn_Click()
        Dim stat As Long
        stat = Usbtmc_test
        If (stat < VI_SUCCESS) Then
            resultTxt.Text = Hex(stat)
        End If
    End Sub
```

运行结果：



4.1.3 MATLAB 示例

系统环境：Windows 7

编程软件：MATLAB R2013a

示例内容：使用 NI-VISA 通过 USBTMC 或 TCP/IP 访问控制设备，并执行读写操作。

请按照以下步骤完成示例：

1、 打开 MATLAB，修改当前目录。在本例中将当前目录修改为 D：\ USBTMC_TCPIP_Demo。

2、 点击文件>>新建>>脚本（File>>New>>Script）在 MATLAB 界面来创建一个空的 M 文档。

3、 添加代码

(1) USBTMC 接口访问代码

```
function USBTMC_test()
%这段代码演示了使用 NI-VISA 发送同步读取和写入命令到
%一个 USB 测试&测量类(USBTMC)仪器。


%创建一个 VISA-USB 对象连接到 USB 仪器上
```

```
vu = visa('ni','USB0::0xF4EC::0x1506::SSG62A0C950001::INSTR');

%打开创建的 VISA 对象
fopen(vu);

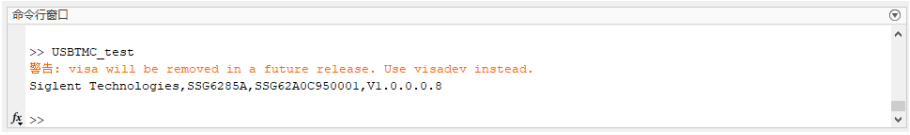
%发送字符串“* IDN? ”，查询设备信息。
fprintf(vu,'*IDN?');

%请求数据
outputbuffer = fscanf(vu);
disp(outputbuffer);

%关闭 VISA 对象
fclose(vu);
delete(vu);
clear vu;

end
```

运行结果：



(2) TCP/IP 接口访问代码

```
function TCP_IP_test()
%这段代码演示了使用 NI-VISA 发送同步读取和写入命令到
%一个 TCP/IP 仪器。

%创建一个 VISA-TCPIP 对象连接到配置了 IP 地址的仪器。
vt = visa('ni',['TCPIP0::',IPstr,'::INSTR']);

%打开创建的 VISA 对象
```

```
fopen(vt);

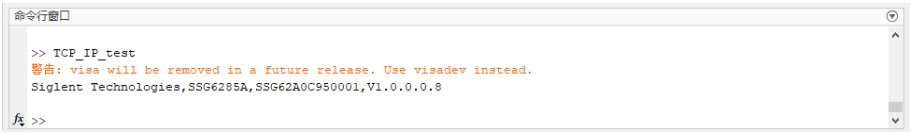
%发送字符串"*IDN?"，查询设备信息
fprintf(vt,'*IDN?');

%请求数据
outputbuffer = fscanf(vt);
disp(outputbuffer);

%关闭 VISA 对象
fclose(vt);
delete(vt);
clear vt;

end
```

运行结果：



4.1.4 LabVIEW 示例

系统环境：Windows 7

编程软件：LabVIEW 2011

示例内容：使用 NI-VISA 通过 USBTMC 或 TCP/IP 访问控制设备，并执行读写操作。

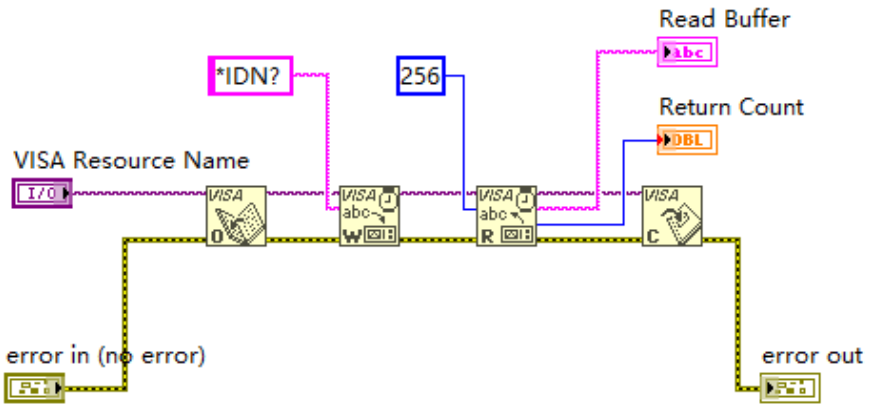
请按照以下步骤完成示例：

1、 打开 LabVIEW，创建 VI 文件。

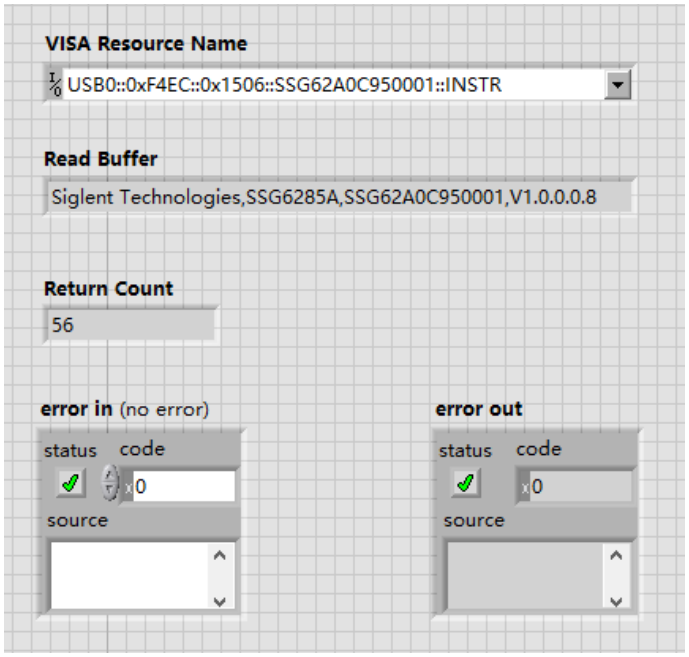
2、 添加控件。右键单击前面板接口，选择并加入 VISA 资源名称，错误输入，错误输出以及控制栏的一些指标。

3、 打开框图接口。在 VISA 资源名称单击鼠标右键，可以从 VISA 调色板的快捷菜单中选择添加以下功能：VISA 写，VISA 读，VISA 打开和 VISA 关闭。

4、 把它们连接起来，如下图所示：



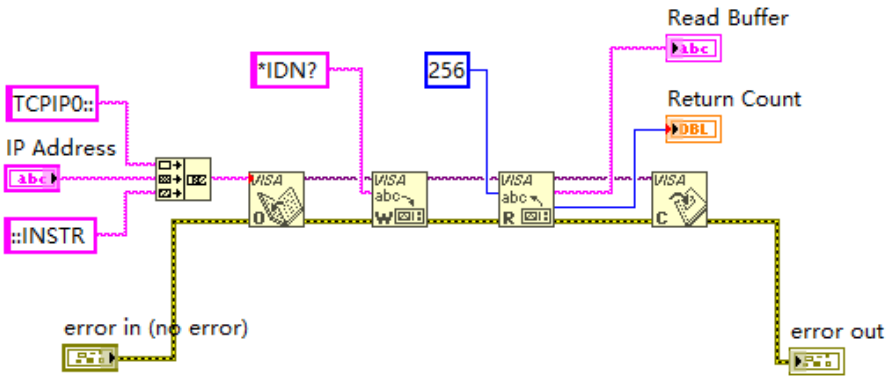
5、 从 VISA 资源名称列表框中选择设备资源然后运行程序。



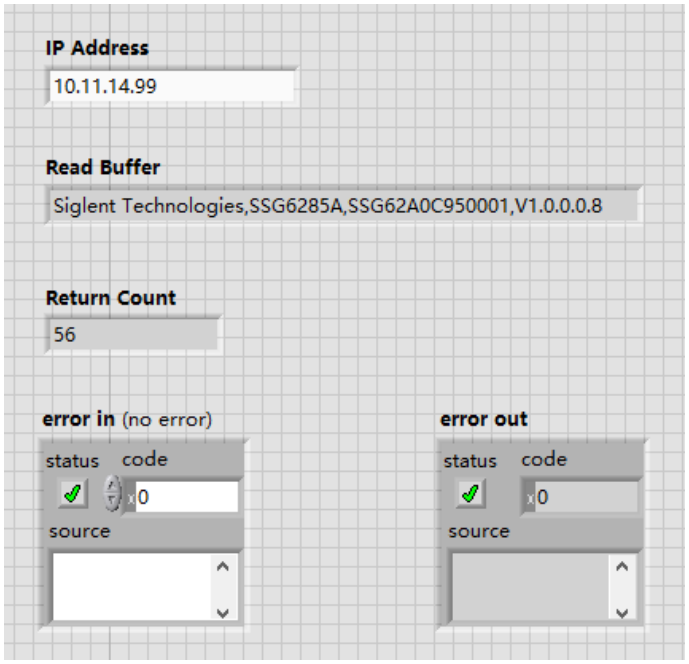
在这个例子中，VI 打开一个 VISA 会话到 USBTMC 设备，写入一个命令到设备，然后读回并响应。此例中，发送的特定的命令是设备 ID 查询。请与您的设备制造商核对设备命令集。在所有通讯完成后，VI 关闭 VISA 会话。

通过 TCP/IP 与以太网仪器通信是类似于 USBTMC 的。但是，你需要改变 VISA 写入和 VISA 读取功能来同步 I/O。LabVIEW 的默认值是异步的 I/O。右键单击该节点，然后从快捷菜单中选择同步 I/O 模式>>同步来写入或读取同步数据。

1、 把它们连接起来，如下图所示：



2、 输入 IP 地址然后运行程序。



4.2 使用 Socket 的编程示例

Windows 操作系统自身支持 Sockets 通信，这种通信方式也是比较简明的。需要注意的是 SCPI 命令字符串的结尾需要加上“\n”（换行符）。

4.2.1 Python 示例

Python 是一种解释性的编程语言，可让您快速工作并且非常可移植。Python 有一个底层网络模块，可以提供对 Socket 接口的访问，端口为 5025。可以为 Socket 接口编写 Python 脚本来执行各种测试和测量任务。

系统环境：Windows 10，64 位操作系统。

编程软件：Python v3.6.5

示例内容：打开 Socket，发送询问 SCPI，然后关闭 Socket，以上循环十次。

脚本内容：

```
#!/usr/bin/env python
#-*- coding:utf-8 -*-
#-----------------------------------------------------------------------------
# The short script is an 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 = "10.11.22.27"    # should match the instrument's IP address
port = 5025 # the port number of the instrument service


def SocketConnect():
    try:
        #create an AF_INET, STREAM socket (TCP)
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    except socket.error:
        input ('Failed to create socket. \nPress "Enter" to exit: ')
        sys.exit()
    try:
        #Connect to remote server
```

```
            s.connect((remote_ip , port))
        except socket.error:
            input('Failed to connect to ip %s!\nPress "Enter" to exit: ' % remote_ip)
            s.close()
            sys.exit()
        return s

    def SocketQuery(Sock, cmd):
        try :
            #Send cmd string
            Sock.sendall(cmd)
            time.sleep(1)
        except socket.error:
            #Send failed
            input('Send failed!\nPress "Enter" to exit: ')
            SocketClose(Sock)
            sys.exit()
        reply = Sock.recv(4096)
        reply = reply.decode()
        return reply

    def SocketClose(Sock):
        #close the socket
        Sock.close()
        time.sleep(.300)

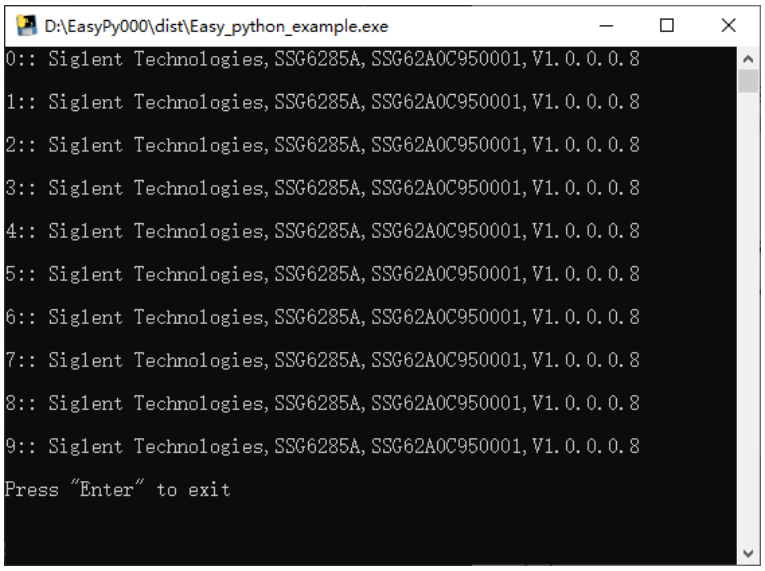
    def main():
        # Body: send the SCPI commands *IDN? 10 times and print the return message
        s = SocketConnect()

        count = 0
        for i in range(10):
            qStr = SocketQuery(s, b'*IDN?\n')
            print (str(count) + ":: " + qStr)
```

```
            count = count + 1
        SocketClose(s)
        input('Press "Enter" to exit')

if __name__ == '__main__':
        main()
```

运行结果：



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

SIGLENT 鼎阳是深圳市鼎阳科技股份有限公司的注册商标，事先未经过允许，不得以任何形式或通过任何方式复制本手册中的任何内容。

本资料中的信息代替原先的此前所有版本。技术数据如有变更，恕不另行通告。

技术许可

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