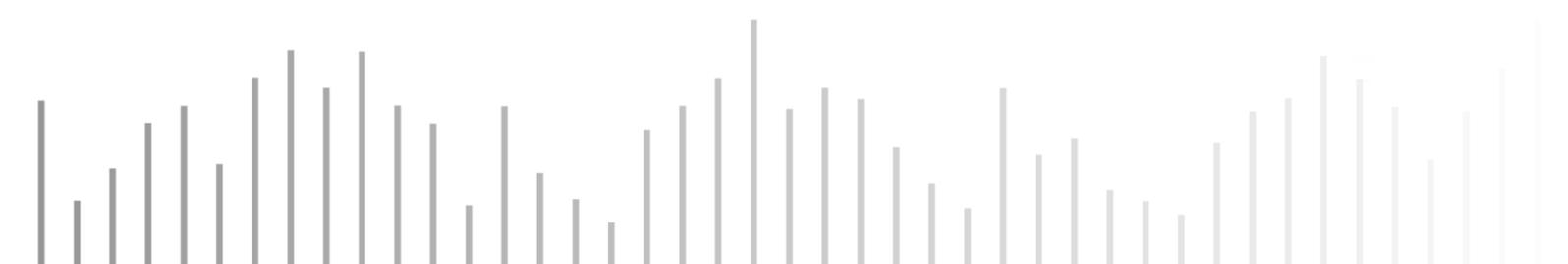


# 频谱分析仪编程手册

REV 01

2022. 12

**UNI-T**



# 保证和声明

## 版权

2022 优利德中国科技有限公司

## 商标信息

UNI-T是优利德中国科技有限公司的注册商标。

## 文档编号

1. 0. 1

## 软件版本

00. 00. 01

软件升级可能更改或增加产品功能，请关注 **UNI-T**网站获取最新版本手册或联系 **UNI-T**升级软件。

## 声明

- 本产品受中国及其它国家和地区的专利（包括已取得的和正在申请的专利）保护。
- 本公司保留改变规格及价格的权利。
- 本手册提供的信息取代以往出版的所有资料。
- 本手册提供的信息如有变更，恕不另行通知。
- 对于本手册可能包含的错误，或因手册所提供的信息及演绎的功能以及因使用本手册而导致的任何偶然或继发的损失，**UNI-T**概不负责。
- 未经 **UNI-T** 事先书面许可，不得影印、复制或改编本手册的任何部分。
- 

## 产品认证

**UNI-T**认证本产品符合中国国家产品标准和行业产品标准及 ISO9001：2008 标准和 ISO14001：2004 标准，并进一步认证本产品符合其它国际标准组织成员的相关标准。

## 联系我们

如您在使用此产品或本手册的过程中有任何问题或需求，可与 **UNI-T**联系：

电子邮箱：infosh@uni-trend.com.cn

网址：<http://www.uni-trend.com>

# SCPI 指令简介

SCPI (Standard Commands for Programmable Instruments, 即可编程仪器标准命令集) 是一种建立在现有标准 IEEE 488.1 和 IEEE 488.2 基础上, 并遵循了 IEEE754 标准中浮点运算规则、ISO646 信息交换 7 位编码符号 (相当于 ASCII 编程) 等多种标准的标准化仪器编程语言。本节简介 SCPI 命令的格式、符号、参数和缩写规则。

## 指令格式

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。命令关键字和第一个参数之间以空格分开。命令字符串必须以一个 <换行> (<NL>) 字符结尾。命令行后面添加问号 “?” 通常表示对此功能进行查询。

## 符号说明

下面四种符号不是 SCPI 命令中的内容, 不随命令发送, 但是通常用于辅助说明命令中的参数。

- **大括号 { }**

大括号中通常包含多个可选参数, 发送命令时必须选择其中一个参数。如: DISPLAY:GRID:MODE { FULL | GRID | CROSS | NONE} 命令。

- **竖线 |**

竖线用于分隔多个参数选项, 发送命令时必须选择其中一个参数。  
如: DISPLAY:GRID:MODE { FULL | GRID | CROSS | NONE} 命令。

- **方括号 [ ]**

方括号中的内容 (命令关键字) 是可省略的。如果省略参数, 仪器将该参数设置为默认值。例如: 对于: MEASURE:NDUTY? [<source>] 命令, [<source>] 表示当前通道。

- **三角括号 < >**

三角括号中的参数必须用一个有效值来替换。例如: 以 DISPLAY:GRID:BRIGHTNESS 30 的形式发送 DISPLAY:GRID:BRIGHTNESS <count> 命令。

## 参数说明

本手册介绍的命令中所含的参数可以分为以下 5 种类型：布尔型、整型、实型、离散型、ASCII 字符串。

- **布尔型**

参数取值为“ON”（1）或“OFF”（0）。例如：`:SYSTem:LOCK {{1 | ON} | {0 | OFF}}`。

- **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：`:DISPlay:GRID:BRIGhtness <count>`命令中的参数< count >可取 0 到 100 范围内的任一整数。

- **实型**

除非另有说明，参数在有效值范围内可以取任意值。

例如：对于 CH1，`CHANnel1:OFFSet <offset>`命令中的参数<offset>的取值为实型。

- **离散型**

参数只能取指定的几个数值或字符。例如：`:DISPlay:GRID:MODE { FULL | GRID | CROSS | NONE}`命令的参数只能为 FULL、GRID、CROSS、NONE。

- **ASCII 字符串**

字符串参数实际上可包含所有 ASCII 字符集。字符串必须以配对的引号开始和结尾；可以用单引号或双引号。引号分隔符也可以作为字符串的一部分，只需键入两次并且不在中间添加任何字符，例如设置 IP：`SYST:COMM:LAN:IPAD "192.168.1.10"`。

## 简写规则

所有命令对大小写都能识别，可以全部采用大写或小写。如果要缩写，必须输完命令格式中的所有大写字母。

## 数据返回

数据返回分为单个数据和批量数据返回，单个数据返回相对应的参数类型，其中实型返回用科学计数法表示，e 前部分小数点后面保留三位数据，e 部分保留三位数据；批量数据返回必须符合 IEEE 488.2 #格式的字符串数据，其格式：'#' + 长度所占的字符位数[固定为一个字符] + 有效数据长度的 ASCII 值 + 有效数据 + 结束符 [ '\n' ]，例如#3123xxxxxxxxxxxxxxxxxxxx\n 表示的具有 123 个字节有效批量数据返回格式，其中 '3' 表示“123”占 3 个字符位。

# SCPI 指令详解

## IEEE488.2 通用命令

### \*IDN?

- **命令格式:**  
\*IDN?
- **功能描述:**  
用于查询制造商名称、产品型号、产品序列号和软件版本号。
- **返回格式:**  
制造商名称, 产品型号, 产品序列号, 由点号分隔的软件版本号。  
注意: 返回的型号要与铭牌信息一致。
- **举例:**  
UNI-T Technologies, UTS3036B, 000000001, 00.00.01

### \*RST

- **命令格式:**  
\*RST
- **功能描述:**  
用于恢复出厂设置并清空所有的错误信息及发送接收队列缓冲。

### \*OPC

- **命令格式:**  
\*OPC  
\*OPC?
- **功能描述:**  
用于在当前操作完成后将标准事件状态寄存器 OPC 位置 1。
- **返回格式:**  
查询返回当前操作是否完成, 0 表示未完成, 1 表示完成。
- **举例:**  
\*OPC                                    将标准事件状态寄存器置 1  
\*OPC?                                   查询返回 1, 表示当前操作已完成, 否则返回 0

## 整机命令

全功能模块通用。

### INSTRument 命令

#### : INSTRument[:SElect]

- **命令格式：**  
: INSTRument[:SElect] {SA|EMI|AD}  
: INSTRument[:SElect]?
- **功能描述：**  
测量功能模块切换。  
SA：频谱扫描  
EMI：电磁干扰  
AD：模拟解调
- **返回格式：**  
查询返回当前测量功能模块，SA、EMI 或 AD。
- **举例：**  
: INSTRument:SElect SA                      选择频谱扫描  
: INSTRument:SElect?                      查询返回 SA

#### : INSTRument:DEFault

- **命令格式：**  
: INSTRument:DEFault
- **功能描述：**  
当前测量功能模块模式重置。
- **返回格式：**  
无返回。
- **举例：**  
: INSTRument:DEFault                      模式重置

### SYSTem 命令

用于对频谱仪进行最基本的操作，主要包括全键盘锁定、系统设置数据的操作。

#### : SYSTem:LOCK

- **命令格式：**

:SYSTem:LOCK {{1|ON} | {0|OFF}}

:SYSTem:LOCK?

➤ **功能描述:**

用于锁定或者解锁键盘按键和触屏输入。

1|ON: 锁定

0|OFF: 解锁

➤ **返回格式:**

查询返回键盘按键和触屏输入锁定状态, 0 表示解锁, 1 表示锁定。

➤ **举例:**

:SYSTem:LOCK ON                      键盘按键和触屏输入锁定

:SYSTem:LOCK OFF                     键盘按键和触屏输入解锁

:SYSTem:LOCK?                        查询返回 0, 表示解锁

### :SYSTem:CONFIgure

➤ **命令格式:**

:SYSTem:CONFIgure <filename>

:SYSTem:CONFIgure?

➤ **功能描述:**

用于读写配置文件, 先发送该指令, 然后发送配置文件数据到频谱仪。

<filename>: 配置文件文件名

➤ **返回格式:**

查询返回频谱仪当前配置文件数据, 二进制流。

➤ **举例:**

:SYSTem:CONFIgure test              写入配置文件数据到频谱仪中并使其加载

:SYSTem:CONFIgure?                 查询返回频谱仪当前配置文件数据二进制流

### :SYSTem:DEFault

➤ **命令格式:**

:SYSTem:DEFault

➤ **功能描述:**

用于恢复默认出厂设置。

➤ **返回格式:**

无返回值。

➤ **举例:**

:SYSTem:DEFault                      恢复默认出厂设置

### :SYSTem:DISPlay:LANGUage

➤ **命令格式:**

:SYSTem:DISPlay:LANGUage {CHINese|ENGLish}

:SYSTem:DISPlay:LANGUage?

- **功能描述:**  
调制系统显示语言。  
CHINese: 中文  
ENGLish: 英文
- **返回格式:**  
查询返回系统显示语言, CHINese 或 ENGLish。
- **举例:**  
:SYSTem:DISPlay:LANGUage CHINese      设置中文为系统显示语言  
:SYSTem:DISPlay:LANGUage?              查询返回 CHINese

#### :SYSTem:DISPlay:CFORmat

- **命令格式:**  
:SYSTem:DISPlay:CFORmat {HR12|HR24}  
:SYSTem:DISPlay:CFORmat?
- **功能描述:**  
调制系统时间显示格式。  
HR12: 12 小时制  
HR24: 24 小时制
- **返回格式:**  
查询返回系统时间显示格式, HR12 或 HR24。
- **举例:**  
:SYSTem:DISPlay:CFORmat HR24      设置 24 小时显示格式  
:SYSTem:DISPlay:CFORmat?              查询返回 HR24

#### :SYSTem:DISPlay:BACKlight:INTensity

- **命令格式:**  
:SYSTem:DISPlay:BACKlight:INTensity <integer>  
:SYSTem:DISPlay:BACKlight:INTensity?
- **功能描述:**  
控制系统背光亮度等级。  
<integer>: 连续整数, 默认单位%, 取值范围 0-100
- **返回格式:**  
查询返回系统背光亮度等级, 单位为%。
- **举例:**  
:SYSTem:DISPlay:BACKlight:INTensity 80      设置系统背光亮度 80%  
:SYSTem:DISPlay:BACKlight:INTensity?              查询返回 80

#### :SYSTem:VOLume

- **命令格式:**  
:SYSTem:VOLume <integer>

:SYSTem:VOLume?

➤ **功能描述:**

控制系统音量大小等级。

<integer>: 连续整数, 默认单位%, 取值范围 0-100

➤ **返回格式:**

查询返回系统音量大小等级, 单位为%。

➤ **举例:**

:SYSTem:VOLume 30

设置系统音量等级 30%

:SYSTem:VOLume?

查询返回 30

## :SYSTem:OUTPut:HDMI

➤ **命令格式:**

:SYSTem:OUTPut:HDMI {{1|ON} | {0|OFF}}

:SYSTem:OUTPut:HDMI?

➤ **功能描述:**

HDMI 输出开关。

1|ON: 打开

0|OFF: 关闭

➤ **返回格式:**

查询返回 HDMI 输出开关状态, 0 或 1。

➤ **举例:**

:SYSTem:OUTPut:HDMI ON

打开 HDMI 输出

:SYSTem:OUTPut:HDMI?

查询返回 1

## :SYSTem:PON:TYPE

➤ **命令格式:**

:SYSTem:PON:TYPE {MODE|LAST|USER}

:SYSTem:PON:TYPE?

➤ **功能描述:**

调整设备上电参数。

MODE: 默认参数

LAST: 最新参数

USER: 用户预设参数

➤ **返回格式:**

查询返回上电参数, MODE、LAST 或 USER。

➤ **举例:**

:SYSTem:PON:TYPE MODE

默认上电参数

:SYSTem:PON:TYPE?

查询 MODE

## :SYSTem:PICTure:FORMat

### ➤ 命令格式:

:SYSTem:PICTure:FORMat {BMP|PNG}

:SYSTem:PICTure:FORMat?

### ➤ 功能描述:

选择系统图片格式。

### ➤ 返回格式:

查询系统图片格式，BMP 或 PNG。

### ➤ 举例:

:SYSTem:PICTure:FORMat BMP

系统图片格式 BMP

:SYSTem:PICTure:FORMat?

查询返回 BMP

## :SYSTem:COMMUnicate:LAN:IPV4:ADAPter

### ➤ 命令格式:

:SYSTem:COMMUnicate:LAN:IPV4:ADAPter {{1|ON} | {0|OFF}}

:SYSTem:COMMUnicate:LAN:IPV4:ADAPter?

### ➤ 功能描述:

网络设置的适配器开关。

### ➤ 返回格式:

查询返回适配器开关状态，0 或 1。

### ➤ 举例:

:SYSTem:COMMUnicate:LAN:IPV4:ADAPter ON

打开网络适配器

:SYSTem:COMMUnicate:LAN:IPV4:ADAPter?

查询返回 1

## :SYSTem:COMMUnicate:LAN:IPV4:DHCP

### ➤ 命令格式:

:SYSTem:COMMUnicate:LAN:IPV4:DHCP {{1|ON} | {0|OFF}}

:SYSTem:COMMUnicate:LAN:IPV4:DHCP?

### ➤ 功能描述:

网络设置的 DHCP 开关。

1|ON: 打开

0|OFF: 关闭

### ➤ 返回格式:

查询返回 DHCP 开关状态，0 或 1。

### ➤ 举例:

:SYSTem:COMMUnicate:LAN:IPV4:DHCP ON

打开 DHCP

:SYSTem:COMMUnicate:LAN:IPV4:DHCP?

查询返回 1

## :SYSTem:COMMUnicate:LAN:IPV4:CONF IG

### ➤ 命令格式:

:SYSTem:COMMUnicate:LAN:IPV4:CONF IG <ip>

:SYSTem:COMMUnicate:LAN:IPV4:CONF IG?

### ➤ 功能描述:

配置网络设置的 IP 地址。

<ip>: 四段点分十进制数据字符, xxx.xxx.xxx.xxx

### ➤ 返回格式:

查询返回当前 IP 地址, 格式 xxx.xxx.xxx.xxx。

### ➤ 举例:

```
:SYSTem:COMMUnicate:LAN:IPV4:CONF IG "192.168.20.111"    设置 IP
:SYSTem:COMMUnicate:LAN:IPV4:CONF IG?                    返回 192.168.20.111
```

## :SYSTem:COMMUnicate:LAN:IPV4:MASK

### ➤ 命令格式:

:SYSTem:COMMUnicate:LAN:IPV4:MASK <mask>

:SYSTem:COMMUnicate:LAN:IPV4:MASK?

### ➤ 功能描述:

配置网络设置的掩码。

<mask>: 四段点分十进制数据, xxx.xxx.xxx.xxx

### ➤ 返回格式:

查询返回掩码数据, 格式 xxx.xxx.xxx.xxx。

### ➤ 举例:

```
:SYSTem:COMMUnicate:LAN:IPV4:MASK "255.255.255.0"    设置掩码为 255.255.255.0
:SYSTem:COMMUnicate:LAN:IPV4:MASK?                    返回 255.255.255.0
```

## :SYSTem:COMMUnicate:LAN:IPV4:GATEWay

### ➤ 命令格式:

:SYSTem:COMMUnicate:LAN:IPV4:GATEWay <gateway>

:SYSTem:COMMUnicate:LAN:IPV4:GATEWay?

### ➤ 功能描述:

配置网络设置的网关。

<gateway>: 四段点分十进制数据, xxx.xxx.xxx.xxx

### ➤ 返回格式:

查询返回网关数据, 格式 xxx.xxx.xxx.xxx。

### ➤ 举例:

```
:SYSTem:COMMUnicate:LAN:IPV4:GATEWay "192.168.20.1"    设置网关为 192.168.20.1
:SYSTem:COMMUnicate:LAN:IPV4:GATEWay?                    返回 192.168.20.1
```

## KEY 命令

:KEY:<key>

➤ **命令格式:**

:KEY:<key>  
:KEY:<key>:LOCK { {1 | ON} | {0 | OFF} }  
:KEY:<key>:LOCK?  
:KEY:<key>:LED?

➤ **功能描述:**

用于设置按键功能及该按键的锁定/解锁。按键<key>的定义和描述，详见附录 1: <key>列表。

➤ **返回格式:**

查询返回按键锁定状态或者具有 LED 按键灯状态。  
LED 灯状态: 0 表示不亮, 1 表示亮 (绿灯)

➤ **举例:**

:KEY:FREQ                                    进入频率设置菜单  
:KEY:LED?                                    查询返回 LED 灯状态, 0 表示不亮

:KEY:LOCK?

➤ **命令格式:**

:KEY:LOCK?

➤ **功能描述:**

用于查询所有按键的锁定状态。

➤ **返回格式:**

查询返回所有按键的锁定状态, 返回字符序列, 每个字符代表一个按键的锁定状态, 锁定为 ASCII '1', 未锁定为 ASCII '0', 按照附录 1: <key>列表顺序返回锁定状态。

➤ **举例:**

一共 38 个按键, 只有第 4、5 两个按键锁定, 返回 ASCII 字符串  
0001100

:KEY:LED?

➤ **命令格式:**

:KEY:LED?

➤ **功能描述:**

用于查询所有带灯按键灯状态。

➤ **返回格式:**

查询返回所有按键的灯状态, 返回字符序列, 每个字符代表一个按键灯状态, 亮为 ASCII '1', 不亮为 ASCII '0'。一共 3 个按键带灯, 按顺序是 TG 按键, Single 按键, Touch/Lock 按键。返回 3 位由 '1' 或 '0' 组成的字符串。

➤ **举例:**

TG 按键亮, Single 按键和 Touch/Lock 按键不亮, 返回 ASCII 字符串 100。

# 各功能模块指令

频谱分析、EMI 和模拟解调。

## 频谱分析

### CALCulate 命令

#### :CALCulate:MARKer:SElect

➤ **命令格式:**

```
:CALCulate:MARKer:SElect <integer>  
:CALCulate:MARKer:SElect?
```

➤ **功能描述:**

从标记序列中选择一个作为当前标记。

<integer>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

查询返回当前标记序列号, 范围 1 到 10。

➤ **举例:**

```
:CALCulate:MARKer:SElect 1      选择标记 1 为当前标记  
:CALCulate:MARKer:SElect?      查询返回 1
```

#### :CALCulate:MARKer<n>:MODE

➤ **命令格式:**

```
:CALCulate:MARKer<n>:MODE {OFF|POSition|DELTA|FIXed}  
:CALCulate:MARKer<n>:MODE?
```

➤ **功能描述:**

选择指定标记的模式。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

OFF: 关闭标记

POSition: 常规模式

DELTA: 差值模式

FIXed: 固定模式

➤ **返回格式:**

查询返回指定标记的模式, OFF、POSition、DELTA 或 FIXed。

➤ **举例:**

```
:CALCulate:MARKer1:MODE POSition  设置标记 1 为常规模式  
:CALCulate:MARKer1:MODE?          查询返回 POSition
```

## :CALCulate:MARKer<n>:TRACe

### ➤ 命令格式:

:CALCulate:MARKer<n>:TRACe <integer>

:CALCulate:MARKer<n>:TRACe?

### ➤ 功能描述:

选择指定标记对应的迹线。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

<integer>: 迹线序列号, 连续整数, 取值范围 1 到最大迹线数。

UTS1015B 的最大迹线数为 4, UTS3036B 的最大迹线数为 6。

### ➤ 返回格式:

查询返回指定标记对应迹线的序列号。

### ➤ 举例:

:CALCulate:MARKer1:TRACe 1            设置标记 1 对应迹线 1

:CALCulate:MARKer1:TRACe?            查询返回 1

## :CALCulate:MARKer<n>:X

### ➤ 命令格式:

:CALCulate:MARKer<n>:X <freq>|<time>

:CALCulate:MARKer<n>:X?

### ➤ 功能描述:

调整指定标记的 X 轴坐标值, 根据 X 轴刻度类型设置对应类型数据。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

<freq>: X 轴刻度类型为频率或者反转时间时, 设置频率, 默认单位为 Hz。

<time>: X 轴刻度类型为周期或时间时, 设置时间, 默认单位为 s。

### ➤ 返回格式:

查询返回指定标记的 X 轴坐标值, 以科学计数法返回数据。X 轴刻度类型为频率或者反转时间时, 单位为 Hz; X 轴刻度类型为周期或时间时, 单位为 s。

### ➤ 举例:

:CALCulate:MARKer1:X 1GHz            设置标记 1 的 X 轴坐标值为 1GHz

:CALCulate:MARKer1:X?            查询返回 1.000000e+09

## :CALCulate:MARKer<n>:Y

### ➤ 命令格式:

:CALCulate:MARKer<n>:Y <ampt>

:CALCulate:MARKer<n>:Y?

### ➤ 功能描述:

调整指定标记的幅度值。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

<ampt>: 标记幅度值, 默认单位为 dBm。

当标记功能关闭时, :CALCulate:MARKer<n>:Y? 返回指定标记的幅度值; 标记功能打

开, :CALCulate:MARKer<n>:Y?查询返回对应的标记功能测量值。

➤ **返回格式:**

查询返回指定标记的幅度值和标记功能的测量结果, 以科学计数法返回数据。幅度值的单位为 dBm, 标记噪声的单位为 dBm/Hz, 带内功率的单位为 dBm, 带内密度的单位为 dBm/Hz。

➤ **举例:**

:CALCulate:MARKer1:Y -50            设置标记 1 的幅度值为-50dBm  
:CALCulate:MARKer1:Y?            查询返回-5.000000e+01

**:CALCulate:MARKer<n>:REFerence**

➤ **命令格式:**

:CALCulate:MARKer<n>:REFerence <integer>  
:CALCulate:MARKer<n>:REFerence?

➤ **功能描述:**

选择指定标记的参考标记, 参考标记不能为自身。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

<integer>: 除自身外的其他标记序列号, 取值范围 1 到 10。

➤ **返回格式:**

查询返回指定标记的参考标记。

➤ **举例:**

:CALCulate:MARKer1:REFerence 2    设置标记 2 为标记 1 的参考标记  
:CALCulate:MARKer1:REFerence?    查询返回 2

**:CALCulate:MARKer<n>:X:READout**

➤ **命令格式:**

:CALCulate:MARKer<n>:X:READout {FREQuency|PERiod|TIME|ITIME}  
:CALCulate:MARKer<n>:X:READout?

➤ **功能描述:**

选择指定标记的 X 轴刻度类型。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

FREQuency: 频率

PERiod: 周期

TIME: 时间

ITIME: 反转时间。

➤ **返回格式:**

查询返回指定标记的 X 轴刻度类型, FREQuency、PERiod、TIME 或 ITIME。

➤ **举例:**

:CALCulate:MARKer1:X:READout FREQuency    设置标记 1 的 X 轴刻度类型为频率  
:CALCulate:MARKer1:X:READout?            查询返回 FREQuency

## **:CALCulate:MARKer<n>:X:READout:AUTO**

### ➤ **命令格式:**

```
:CALCulate:MARKer<n>:X:READout:AUTO {{1|ON} | {0|OFF}}  
:CALCulate:MARKer<n>:X:READout:AUTO?
```

### ➤ **功能描述:**

切换指定标记的 X 轴刻度模式为自动或手动。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

1|ON: 自动

0|OFF: 手动

### ➤ **返回格式:**

查询返回指定标记的参考标记, 0 或 1。

### ➤ **举例:**

```
:CALCulate:MARKer1:X:READout:AUTO ON      设置标记 1X 轴刻度模式为自动  
:CALCulate:MARKer1:X:READout:AUTO?      查询返回 1
```

## **:CALCulate:MARKer<n>:LINES[:STATe]**

### ➤ **命令格式:**

```
:CALCulate:MARKer<n>:LINES[:STATe] {{1|ON} | {0|OFF}}  
:CALCulate:MARKer<n>:LINES[:STATe]?
```

### ➤ **功能描述:**

指定标记的标记线开关。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

### ➤ **返回格式:**

查询返回指定标记的标记线开关状态, 0 或 1。

### ➤ **举例:**

```
:CALCulate:MARKer1:LINES ON      打开标记 1 的标记线  
:CALCulate:MARKer1:LINES?      查询返回 1
```

## **:CALCulate:MARKer<n>:FUNCTion**

### ➤ **命令格式:**

```
:CALCulate:MARKer<n>:FUNCTion {OFF|NOISe|BPOWer|BDENsity}  
:CALCulate:MARKer<n>:FUNCTion?
```

### ➤ **功能描述:**

选择指定标记的标记功能。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

NOISe: 标记噪声

BPOWer: 带内功率

BDENsity: 带内密度

### ➤ **返回格式:**

查询返回指定标记的标记功能, OFF、NOISe、BPOWer 或 BDENsity。

➤ **举例：**

:CALCulate:MARKer1:FUNCTion NOISe 选择标记 1 的标记功能为标记噪声  
:CALCulate:MARKer1:FUNCTion? 查询返回 NOISe

**:CALCulate:MARKer<n>:FUNCTion:BAND:SPAN**

➤ **命令格式：**

:CALCulate:MARKer<n>:FUNCTion:BAND:SPAN <freq>|<time>  
:CALCulate:MARKer<n>:FUNCTion:BAND:SPAN?

➤ **功能描述：**

调整指定标记的标记带宽。

<n>：标记序列号，连续整数，取值范围 1 到 10。

<freq>：X 轴刻度类型为频率或者反转时间时，设置频率，默认单位为 Hz。

<time>：X 轴刻度类型为周期或时间时，设置时间，默认单位为 s。

➤ **返回格式：**

查询返回指定标记的标记带宽，以科学计数法返回数据。X 轴刻度类型为频率或者反转时间时，单位为 Hz；  
X 轴刻度类型为周期或时间时，单位为 s。

➤ **举例：**

:CALCulate:MARKer1:FUNCTion:BAND:SPAN 10MHz 设置标记 1 的标记带宽为 10MHz  
:CALCulate:MARKer1:FUNCTion:BAND:SPAN? 查询返回 1.000000e+07

**:CALCulate:MARKer<n>:FUNCTion:BAND:LEFT**

➤ **命令格式：**

:CALCulate:MARKer<n>:FUNCTion:BAND:LEFT <freq>|<time>  
:CALCulate:MARKer<n>:FUNCTion:BAND:LEFT?

➤ **功能描述：**

调整指定标记的左沿带宽。

<n>：标记序列号，连续整数，取值范围 1 到 10。

<freq>：X 轴刻度类型为频率或者反转时间时，设置频率，默认单位为 Hz。

<time>：X 轴刻度类型为周期或时间时，设置时间，默认单位为 s。

➤ **返回格式：**

查询返回指定标记的左沿带宽，以科学计数法返回数据。X 轴刻度类型为频率或者反转时间时，单位为 Hz；  
X 轴刻度类型为周期或时间时，单位为 s。

➤ **举例：**

:CALCulate:MARKer1:FUNCTion:BAND:LEFT 5MHz 设置标记 1 的左沿带宽为 5MHz  
:CALCulate:MARKer1:FUNCTion:BAND:LEFT? 查询返回 5.000000e+06

**:CALCulate:MARKer<n>:FUNCTion:BAND:RIGHT**

➤ **命令格式：**

:CALCulate:MARKer<n>:FUNCTion:BAND:RIGHT <freq>|<time>  
:CALCulate:MARKer<n>:FUNCTion:BAND:RIGHT?

➤ **功能描述:**

调整指定标记的右沿带宽。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

<freq>, X 轴刻度类型为频率或者反转时间时, 设置频率, 默认单位为 Hz。

<time>, X 轴刻度类型为周期或时间时, 设置时间, 默认单位为 s。

➤ **返回格式:**

查询返回指定标记的右沿带宽, 以科学计数法返回数据。X 轴刻度类型为频率或者反转时间时, 单位为 Hz;  
X 轴刻度类型为周期或时间时, 单位为 s。

➤ **举例:**

:CALCulate:MARKer1:FUNCTion:BAND:RIGHT 5MHz 设置标记 1 的带宽右沿为 5MHz

:CALCulate:MARKer1:FUNCTion:BAND:RIGHT? 查询返回 5.000000e+06

**:CALCulate:MARKer<n>:BANDwidth[:STATe]**

➤ **命令格式:**

:CALCulate:MARKer<n>:BANDwidth[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:MARKer<n>:BANDwidth[:STATe]?

➤ **功能描述:**

指定标记的 NDB 带宽开关。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

查询返回指定标记的 NDB 带宽开关状态, 0 或 1。

➤ **举例:**

:CALCulate:MARKer1:BANDwidth ON 打开标记 1 的 NDB 带宽

:CALCulate:MARKer1:BANDwidth? 查询返回 1

**:CALCulate:MARKer<n>:BANDwidth|BWIDth:NDB**

➤ **命令格式:**

:CALCulate:MARKer<n>:BANDwidth|BWIDth:NDB <real>

:CALCulate:MARKer<n>:BANDwidth|BWIDth:NDB?

➤ **功能描述:**

调整指定标记的 NDB 点。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

<real>: 连续实数, 默认单位为 dB, 取值范围-0.01 到-140。

➤ **返回格式:**

查询返回指定标记的 NDB 值, 以科学计数法返回数据, 单位为 dB。

➤ **举例:**

:CALCulate:MARKer1:BANDwidth:NDB -3 打开标记 1 的 NDB 带宽为-3dB

:CALCulate:MARKer1:BANDwidth:NDB? 查询返回-3.000000e+00

## **:CALCulate:MARKer<n>:BANDwidth|BWIDth:RESult?**

### **命令格式:**

:CALCulate:MARKer<n>:BANDwidth|BWIDth:RESult?

### ➤ **功能描述:**

查询指定标记 NDB 带宽测量结果。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

### ➤ **返回格式:**

查询返回指定标记的 NDB 测量结果, NDB 带宽, NDB 带宽单位 Hz。

### ➤ **举例:**

:CALCulate:MARKer1:Bandwidth:RESult? 查询返回标记 1 的 NDB 测量结果

## **:CALCulate:MARKer:FCOunt[:STATe]**

### ➤ **命令格式:**

:CALCulate:MARKer:FCOunt[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:MARKer:FCOunt[:STATe]?

### ➤ **功能描述:**

频率计开关。

### ➤ **返回格式:**

查询返回频率计开关状态, 0 或 1。

### ➤ **举例:**

:CALCulate:MARKer:FCOunt ON 打开频率计

:CALCulate:MARKer:FCOunt? 查询返回 1

## **:CALCulate:MARKer:FCOunt:GATetime**

### ➤ **命令格式:**

:CALCulate:MARKer:FCOunt:GATetime <time>

:CALCulate:MARKer:FCOunt:GATetime?

### ➤ **功能描述:**

调整频率计门限时间。

<time>: 连续实数, 默认单位为 s, 取值范围 1us 到 500ms。

### ➤ **返回格式:**

查询返回频率计门限时间, 以科学计数法返回数据, 单位为 s。

### ➤ **举例:**

:CALCulate:MARKer:FCOunt:GATetime 0.1 设置频率计门限时间为 100ms

:CALCulate:MARKer:FCOunt:GATetime? 查询返回 1.000000e-01

## **:CALCulate:MARKer:AOFF**

### ➤ **命令格式:**

:CALCulate:MARKer:AOFF

- **功能描述:**  
关闭所有标记。
- **返回格式:**  
无返回值。
- **举例:**  
:CALCulate:MARKer:AOFF                      关闭所有标记

**:CALCulate:MARKer<n>[:SET]:CENTER**

- **命令格式:**  
:CALCulate:MARKer<n>[:SET]:CENTER
- **功能描述:**  
调整中心频率为指定标记频率。  
<n>: 标记序列号, 连续整数, 取值范围 1 到 10。
- **返回格式:**  
无返回值。
- **举例:**  
:CALCulate:MARKer1:CENTer                      设置中心频率为标记 1 频率

**:CALCulate:MARKer<n>[:SET]:STEP**

- **命令格式:**  
:CALCulate:MARKer<n>[:SET]:STEP
- **功能描述:**  
调整中心频率步进为指定标记频率。  
<n>: 标记序列号, 连续整数, 取值范围 1 到 10。
- **返回格式:**  
无返回值。
- **举例:**  
:CALCulate:MARKer1:STEP                      设置中心频率步进为标记 1 频率

**:CALCulate:MARKer<n>[:SET]:START**

- **命令格式:**  
:CALCulate:MARKer<n>[:SET]:START
- **功能描述:**  
调整起始频率为指定标记频率。  
<n>: 标记序列号, 连续整数, 取值范围 1 到 10。
- **返回格式:**  
无返回值。
- **举例:**  
:CALCulate:MARKer1:START                      设置起始频率为标记 1 频率

## **:CALCulate:MARKer<n>[:SET]:STOP**

### ➤ **命令格式:**

:CALCulate:MARKer<n>[:SET]:STOP

### ➤ **功能描述:**

调整截止频率为指定标记频率。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

### ➤ **返回格式:**

无返回值。

### ➤ **举例:**

:CALCulate:MARKer1:STOP                      设置截止频率为标记 1 频率

## **:CALCulate:MARKer<n>[:SET]:RLEVel**

### ➤ **命令格式:**

:CALCulate:MARKer<n>[:SET]:RLEVel

### ➤ **功能描述:**

调整参考电平为指定标记幅度。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

### ➤ **返回格式:**

无返回值。

### ➤ **举例:**

:CALCulate:MARKer1:RLEVel                      设置参考电平为标记 1 幅度

## **:CALCulate:MARKer:TABLE[:STATe]**

### ➤ **命令格式:**

:CALCulate:MARKer:TABLE[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:MARKer:TABLE[:STATe]?

### ➤ **功能描述:**

标记列表显示开关。

### ➤ **返回格式:**

查询返回标记列表显示状态, 0 或 1。

### ➤ **举例:**

:CALCulate:MARKer:TABLE ON                      显示标记列表

:CALCulate:MARKer:TABLE?                      查询返回 1

## **:CALCulate:MARKer<n>:MAXimum[:MAX]**

### ➤ **命令格式:**

:CALCulate:MARKer<n>:MAXimum[:MAX]

### ➤ **功能描述:**

对指定标记执行峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:MARKer1:MAXimum                    标记 1 执行峰值搜索

**:CALCulate:MARKer<n>:MAXimum:NEXT**

➤ **命令格式:**

:CALCulate:MARKer<n>:MAXimum:NEXT

➤ **功能描述:**

对指定标记执行下一峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:MARKer1:MAXimum:NEXT    标记 1 执行下一峰值搜索

**:CALCulate:MARKer<n>:MAXimum:LEFT**

➤ **命令格式:**

:CALCulate:MARKer<n>:MAXimum:LEFT

➤ **功能描述:**

对指定标记执行左侧下一峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:MARKer1:MAXimum:LEFT    标记 1 执行左侧下一峰值搜索

**:CALCulate:MARKer<n>:MAXimum:RIGHT**

➤ **命令格式:**

:CALCulate:MARKer<n>:MAXimum:RIGHT

➤ **功能描述:**

对指定标记执行右侧下一峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:MARKer1:MAXimum:RIGHT    标记 1 执行右侧下一峰值搜索

### **:CALCulate:MARKer<n>:MINimum**

➤ **命令格式:**

:CALCulate:MARKer<n>:MINimum

➤ **功能描述:**

对指定标记执行最小峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:MARKer1:MINimum                    标记 1 执行最小峰值搜索

### **:CALCulate:MARKer<n>:PTPeak**

➤ **命令格式:**

:CALCulate:MARKer<n>:PTPeak

➤ **功能描述:**

对指定标记执行峰峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:MARKer1:PTPeak                    标记 1 执行峰峰值搜索

### **:CALCulate:MARKer<n>:CPSearch[:STATe]**

➤ **命令格式:**

:CALCulate:MARKer<n>:CPSearch[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:MARKer<n>:CPSearch[:STATe]?

➤ **功能描述:**

对指定标记执行连续峰值搜索开关。

<n>: 标记序列号, 连续整数, 取值范围 1 到 10。

1|ON: 开

0|OFF: 关

➤ **返回格式:**

查询返回指定标记连续峰值搜索开关状态, 0 或 1。

➤ **举例:**

:CALCulate:MARKer1:CPSearch ON            打开标记 1 连续峰值搜索

:CALCulate:MARKer1:CPSearch?            查询返回 1

### **:CALCulate:MARKer:PEAK:TABLE[:STATe]**

➤ **命令格式:**

:CALCulate:MARKer:PEAK:TABLE[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:MARKer:PEAK:TABLE[:STATe]?

➤ **功能描述:**

峰值表显示开关。

1|ON: 开

0|OFF: 关

➤ **返回格式:**

查询返回峰值表显示状态, 0 或 1。

➤ **举例:**

:CALCulate:MARKer:PEAK:TABLE ON 显示峰值表

:CALCulate:MARKer:PEAK:TABLE? 查询返回 1

**:CALCulate:MARKer:PEAK:THReshold:LINE[:STATe]**

➤ **命令格式:**

:CALCulate:MARKer:PEAK:THReshold:LINE[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:MARKer:PEAK:THReshold:LINE[:STATe]?

➤ **功能描述:**

峰值阈值线显示开关。

1|ON: 开

0|OFF: 关

➤ **返回格式:**

查询返回阈值线显示状态, 0 或 1。

➤ **举例:**

:CALCulate:MARKer:PEAK:THReshold:LINE ON 显示峰值阈值线

:CALCulate:MARKer:PEAK:THReshold:LINE? 查询返回 1

**:CALCulate:MARKer:PEAK:THReshold**

➤ **命令格式:**

:CALCulate:MARKer:PEAK:THReshold <ampl>

:CALCulate:MARKer:PEAK:THReshold?

➤ **功能描述:**

调整峰值门限。

<ampl>: 连续实数, 默认单位为 dBm, 范围 30dBm 到-200dBm。

➤ **返回格式:**

查询返回峰值门限, 以科学计数法返回数据, 单位为 dBm。

➤ **举例:**

:CALCulate:MARKer:PEAK:THReshold -20 设置峰值门限-20dBm

:CALCulate:MARKer:PEAK:THReshold? 查询返回-2.000000e+01

## :CALCulate:MARKer:PEAK:THReshold:STATe

### ➤ 命令格式:

```
:CALCulate:MARKer:PEAK:THReshold:STATe {{1|ON} | {0|OFF}}  
:CALCulate:MARKer:PEAK:THReshold:STATe?
```

### ➤ 功能描述:

手动/自动峰值门限切换。

1|ON: 自动峰值门限。

0|OFF: 手动峰值门限。

### ➤ 返回格式:

查询返回自动峰值门限状态, 0 或 1。

### ➤ 举例:

```
:CALCulate:MARKer:PEAK:THReshold:STATe ON 选择自动峰值门限  
:CALCulate:MARKer:PEAK:THReshold:STATe? 查询返回 1
```

## :CALCulate:MARKer:PEAK:EXCursion

### ➤ 命令格式:

```
:CALCulate:MARKer:PEAK:EXCursion <ampl>  
:CALCulate:MARKer:PEAK:EXCursion?
```

### ➤ 功能描述:

调整峰值偏移。

<ampl>: 连续实数, 默认单位为 dB, 范围 0dB 到 100dB。

### ➤ 返回格式:

查询返回峰值偏移, 以科学计数法返回数据, 单位为 dB。

### ➤ 举例:

```
:CALCulate:MARKer:PEAK:EXCursion 20 设置峰值偏移 20dB  
:CALCulate:MARKer:PEAK:EXCursion? 查询返回 2.000000e+01
```

## :CALCulate:MARKer:PEAK:EXCursion:STATe

### ➤ 命令格式:

```
:CALCulate:MARKer:PEAK:EXCursion:STATe {{1|ON} | {0|OFF}}  
:CALCulate:MARKer:PEAK:EXCursion:STATe?
```

### ➤ 功能描述:

手动/自动峰值偏移切换。

1|ON: 自动峰值偏移。

0|OFF: 手动峰值偏移。

### ➤ 返回格式:

查询返回自动峰值偏移状态, 0 或 1。

### ➤ 举例:

```
:CALCulate:MARKer:PEAK:EXCursion:STATe ON 选择自动峰值偏移  
:CALCulate:MARKer:PEAK:EXCursion:STATe? 查询返回 1
```

## :CALCulate:MATH

### ➤ 命令格式:

:CALCulate:MATH {<trace>, <type>, <op1>, <op2>, <real>}

:CALCulate:MATH?

### ➤ 功能描述:

设置迹线数学运算。

<trace>: TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, 指定数学运算迹线。

<type>: OFF|PDIFference|PSUM|LDIFference|LOFFset, OFF 关闭, PDIFference 功率差, PSUM 功率和, LDIFference 对数差值, LOFFset 对数偏移。

<op1>: TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, 指定操作数 A 的迹线。

<op2>: TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, 指定操作数 B 的迹线。

<real>: 数学运算偏移, 单位 dB, 取值范围-100dB 到 100dB。在运算类型为对数差和对数偏移时才有效。

### ➤ 返回格式:

无返回。

### ➤ 举例:

:CALCulate:MATH TRACE1, PDIFference, TRACE2, TRACE3, 0 设置迹线 1 功率差运算, 操作数 1 为迹线 2, 操作数 2 为迹线 3, 偏移 0dB

## :CALCulate:LLINe:SElect

### ➤ 命令格式:

:CALCulate:LLINe:SElect <integer>

:CALCulate:LLINe:SElect?

### ➤ 功能描述:

从限值序列中选择一个作为当前限值。

<integer>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

### ➤ 返回格式:

查询返回当前限值序列号。

### ➤ 举例:

:CALCulate:LLINe:SElect 1 选择限值 1 为当前限值

:CALCulate:LLINe:SElect? 查询返回 1

## :CALCulate:LLINe<n>:DISPlay

### ➤ 命令格式:

:CALCulate:LLINe<n>:DISPlay {{1|ON} | {0|OFF}}

:CALCulate:LLINe<n>:DISPlay?

### ➤ 功能描述:

指定限值显示开关。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

1|ON: 显示打开  
0|OFF: 显示关闭

➤ **返回格式:**

查询返回当前限值序列号, 0 或 1。

➤ **举例:**

:CALCulate:LLINe1:DISPlay ON            显示限值 1

:CALCulate:LLINe1:DISPlay?            查询返回 1

**:CALCulate:LLINe<n>:TRACe**

➤ **命令格式:**

:CALCulate:LLINe<n>:TRACe <integer>

:CALCulate:LLINe<n>:TRACe?

➤ **功能描述:**

选择指定限值测试迹线。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

<integer>: 迹线序列号, 连续整数, 取值范围 1 到最大迹线数。

UTS1015B 的最大迹线数为 4, UTS3036B 的最大迹线数为 6。

➤ **返回格式:**

查询返回当前限值序列号。

➤ **举例:**

:CALCulate:LLINe1:TRACe 1            选择迹线 1 为限值 1 测试迹线

:CALCulate:LLINe1:TRACe?            查询返回 1

**:CALCulate:LLINe<n>:MARGin**

➤ **命令格式:**

:CALCulate:LLINe<n>:MARGin <real>

:CALCulate:LLINe<n>:MARGin?

➤ **功能描述:**

调整指定限值裕量。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

<real>: 连续实数, 默认单位为 dB, 范围-40dB 到 40dB。

➤ **返回格式:**

查询返回指定限值裕量, 以科学计数法返回数据, 单位为 dB。

➤ **举例:**

:CALCulate:LLINe1:MARGin 20            设置限值 1 裕量 20dB

:CALCulate:LLINe1:MARGin?            查询返回 2.000000e+01

## :CALCulate:LLINe<n>:MARGin:STATe

### ➤ 命令格式:

```
:CALCulate:LLINe<n>:MARGin:STATe {{1|ON} | {0|OFF}}
```

```
:CALCulate:LLINe<n>:MARGin:STATe?
```

### ➤ 功能描述:

指定限值裕量开关。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

1|ON: 开

0|OFF: 关

### ➤ 返回格式:

查询返回指定限值裕量开关状态, 0 或 1。

### ➤ 举例:

```
:CALCulate:LLINe1:MARGin:STATe ON 限值 1 打开限值裕量
```

```
:CALCulate:LLINe1:MARGin:STATe?          查询返回 1
```

## :CALCulate:LLINe<n>:TYPE

### ➤ 命令格式:

```
:CALCulate:LLINe<n>:TYPE {UPPer|LOWer}
```

```
:CALCulate:LLINe<n>:TYPE?
```

### ➤ 功能描述:

选择指定限值类型。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

UPPer: 高于

LOWer: 低于

### ➤ 返回格式:

查询返回指定限值类型, UPPer 或 LOWer。

### ➤ 举例:

```
:CALCulate:LLINe1:TYPE UPPer          设置限值 1 类型为高于
```

```
:CALCulate:LLINe1:TYPE?              查询返回 UPPer
```

## :CALCulate:LLINe<n>:DATA

### ➤ 命令格式:

```
:CALCulate:LLINe<n>:DATA {<freq>, <ampl>, <connect>, <freq>, <ampl>, <connect>, ... }
```

```
:CALCulate:LLINe<n>:DATA?
```

### ➤ 功能描述:

编辑指定限值数据, 以 {频率, 幅值, 连接属性} 为基本单元编辑。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

<freq>: 连续实数, 单位 Hz

<amp;lt;: 连续实数, 单位默认 dBm

<connect>: 取值为 0 或 1。当取值为 1 时, 表示当前点与前一个点相连接以确定限制线; 取值为 0 时, 表示当前点与前一个点不连接(断开)。第一个点的<connect>值是被忽略的, 0 和 1 都可以, 返回值默认为 1。

➤ **返回格式:**

查询返回指定限值数据, {频率, 幅值, 连接属性}为基本单元换行, 频率单位为 Hz, 幅值单位为 dBm, 连接属性为 0 或 1。

➤ **举例:**

```
:CALCulate:LLINe1:DATA 10000000, -50, 0, 100000000, -60, 1      设置限值 1 数据
:CALCulate:LLINe1:DATA?                                          查询返回限值 1 数据
```

**:CALCulate:LLINe<n>:COPY**

➤ **命令格式:**

```
:CALCulate:LLINe<n>:COPY {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6}
```

➤ **功能描述:**

拷贝指定限值数据到指定限值, 不能自己拷贝自己。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

LLINE1 - LLINE6: 对应限值 1 到限值 6。

UTS1015B 为 LLINE1 - LLINE4, UTS3036B 为 LLINE1 - LLINE6。

➤ **返回格式:**

无返回。

➤ **举例:**

```
:CALCulate:LLINe2:COPY LLINE1 拷贝限值 1 的数据到限值 2
```

**:CALCulate:LLINe<n>:BUIld**

➤ **命令格式:**

```
:CALCulate:LLINe<n>:BUIld {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6}
```

➤ **功能描述:**

指定限值数据从迹线构筑。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

TRACE1 - TRACE6: 对应迹线 1 到迹线 6。

UTS1015B 为 TRACE1 - TRACE4, UTS3036B 为 TRACE1- TRACE6。

➤ **返回格式:**

无返回。

➤ **举例:**

```
:CALCulate:LLINe1:BUIld TRACE1 限值 1 数据从迹线 1 构筑
```

### **:CALCulate:LLINe<n>:OFFSet:X**

➤ **命令格式:**

:CALCulate:LLINe<n>:OFFSet:X <freq>

:CALCulate:LLINe<n>:OFFSet:X?

➤ **功能描述:**

设置指定限值 X 轴偏移。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

<freq>: 连续实数, 默认单位为 Hz。

➤ **返回格式:**

查询返回指定限值 X 轴偏移, 以科学计数法返回数据, 单位 Hz。

➤ **举例:**

:CALCulate:LLINe1:OFFSet:X 10000000      设置限值 1 X 轴偏移 10MHz

:CALCulate:LLINe1:OFFSet:X?                查询返回 1.000000e+07

### **:CALCulate:LLINe<n>:OFFSet:Y**

➤ **命令格式:**

:CALCulate:LLINe<n>:OFFSet:Y <real>

:CALCulate:LLINe<n>:OFFSet:Y?

➤ **功能描述:**

设置指定限值 Y 轴偏移。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

<real>: 连续实数, 默认单位为 dB。

➤ **返回格式:**

查询返回指定限值 Y 轴偏移, 以科学计数法返回数据, 单位为 dB。

➤ **举例:**

:CALCulate:LLINe1:OFFSet:Y 5                设置限值 1 Y 轴偏移 5dB

:CALCulate:LLINe1:OFFSet:Y?                查询返回 5.000000e+00

### **:CALCulate:LLINe<n>:OFFSet:UPDate**

➤ **命令格式:**

:CALCulate:LLINe<n>:OFFSet:UPDate

➤ **功能描述:**

指定限值应用偏移。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

➤ **返回格式:**

无返回。

➤ **举例:**

:CALCulate:LLINe1:OFFSet:UPDate 限值 1 应用偏移

#### :CALCulate:LLINe<n>:DELeTe

➤ **命令格式:**

:CALCulate:LLINe<n>:DELeTe

➤ **功能描述:**

删除指定限值数据。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

➤ **返回格式:**

无返回。

➤ **举例:**

:CALCulate:LLINe1:DELeTe 删除限值 1 数据

#### :CALCulate:LLINe:ALL:DELeTe

➤ **命令格式:**

:CALCulate:LLINe:ALL:DELeTe

➤ **功能描述:**

删除所有限值数据。

➤ **返回格式:**

无返回。

➤ **举例:**

:CALCulate:LLINe:ALL:DELeTe 删除所有限值数据

## CONFiGure 命令

#### :CONFiGure:SANalyzer

➤ **命令格式:**

:CONFiGure:SANalyzer

➤ **功能描述:**

进入扫频。

➤ **返回格式:**

无返回。

➤ **举例:**

:CONFiGure:SANalyzer 进入扫频

#### :CONFiGure:CHPower

➤ **命令格式:**

:CONFigure:CHPower

- **功能描述:**  
进入通道功率测试和测量预置。通道功率的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:CHPower                    进入通道功率

#### :CONFigure:TPOwer

- **命令格式:**  
:CONFigure:TPOwer
- **功能描述:**  
进入时域功率测试和测量预置。时域功率的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:TPOwer                    进入时域功率

#### :CONFigure:OBWidth

- **命令格式:**  
:CONFigure:OBWidth
- **功能描述:**  
进入占用带宽测试和测量预置。占用带宽的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:OBWidth                  进入占用带宽

#### :CONFigure:T0I

- **命令格式:**  
:CONFigure:T0I
- **功能描述:**  
进入三阶交调测试和测量预置。三阶交调的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:T0I                        进入三阶交调

### :CONFiGure:ACPower

- **命令格式:**  
:CONFiGure:ACPower
- **功能描述:**  
进入相邻信道功率测试和测量预置。相邻信道功率的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:ACPower                                进入相邻信道功率

### :CONFiGure:SPECTrogram

- **命令格式:**  
:CONFiGure:SPECTrogram
- **功能描述:**  
进入频谱监测测试和测量预置。频谱监测的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:SPECTrogram                                进入频谱监测

### :CONFiGure:CNRatio

- **命令格式:**  
:CONFiGure:CNRatio
- **功能描述:**  
进入载噪比测试和测量预置。载噪比的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:CNRatio    进入载噪比

### :CONFiGure:HARMonics

- **命令格式:**  
:CONFiGure:HARMonics
- **功能描述:**  
进入谐波测试和测量预置。谐波测试的测量参数会恢复默认值。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:HARMonics    进入谐波

#### :CONFiGure:CHPower:NDEFault

- **命令格式:**  
:CONFiGure:CHPower:NDEFault
- **功能描述:**  
进入通道功率测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:CHPower:NDEFault                    进入通道功率

#### :CONFiGure:TPOWer:NDEFault

- **命令格式:**  
:CONFiGure:TPOWer:NDEFault
- **功能描述:**  
进入时域功率测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:TPOWer:NDEFault                    进入时域功率

#### :CONFiGure:OBWidth:NDEFault

- **命令格式:**  
:CONFiGure:OBWidth:NDEFault
- **功能描述:**  
进入占用带宽测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:OBWidth:NDEFault                    进入占用带宽

#### :CONFiGure:T0I:NDEFault

- **命令格式:**  
:CONFiGure:T0I:DEFault
- **功能描述:**  
进入三阶交调测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFiGure:T0I:DEFault                    进入三阶交调

### **:CONFigure:ACPower:NDEFault**

- **命令格式:**  
:CONFigure:ACPower:NDEFault
- **功能描述:**  
进入相邻信道功率测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:ACPower:NDEFault                    进入相邻信道功率

### **:CONFigure:SPECTrogram:NDEFault**

- **命令格式:**  
:CONFigure:SPECTrogram:NDEFault
- **功能描述:**  
进入频谱监测测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:SPECTrogram:NDEFault                    进入频谱监测

### **:CONFigure:CNRatio:NDEFault**

- **命令格式:**  
:CONFigure:CNRatio:NDEFault
- **功能描述:**  
进入载噪比测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:CNRatio:NDEFault                    进入载噪比

### **:CONFigure:HARMonics:NDEFault**

- **命令格式:**  
:CONFigure:HARMonics:NDEFault
- **功能描述:**  
进入谐波测量测试。
- **返回格式:**  
无返回。
- **举例:**  
:CONFigure:HARMonics:NDEFault                    进入谐波

## DISPlay 命令

### :DISPlay:ZBmp?

- **命令格式:**  
:DISPlay:ZBmp?
- **功能描述:**  
获取屏幕图像。
- **返回格式:**  
查询返回屏幕图像数据, 图片数据经过了压缩。
- **举例:**  
:DISPlay:ZBmp?                      获取屏幕图像

### :DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel

- **命令格式:**  
:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel <real>  
:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel?
- **功能描述:**  
调整参考电平。  
<real>: 连续实数, 默认单位为 dBm。
- **返回格式:**  
查询返回参考电平值, 以科学记数法返回数据, 单位为 dBm。
- **举例:**  
:DISPlay:WINDow:TRACe:Y:RLEVel -10              设置参考电平为-10dBm  
:DISPlay:WINDow:TRACe:Y:RLEVel?              查询返回-1.000000e+01

### :DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision

- **命令格式:**  
:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision <real>  
:DISPlay:WINDow:TRACe:Y[:SCALe]:PDIVision?
- **功能描述:**  
调整 Y 轴刻度。  
<real>: 离散实数, 默认单位 dB, 取值范围 0.1dB 到 20dB。
- **返回格式:**  
查询返回 Y 轴刻度值, 以科学记数法返回数据, 单位为 dB。
- **举例:**  
:DISPlay:WINDow:TRACe:Y:PDIVision 1              设置 Y 轴刻度为 1dB  
:DISPlay:WINDow:TRACe:Y:PDIVision?              查询返回 1.000000e+00

## :DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet

### ➤ 命令格式:

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet <real>

:DISPlay:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet?

### ➤ 功能描述:

调整参考电平偏移。

<real>: 连续实数, 默认单位为 dB。

### ➤ 返回格式:

查询返回参考电平偏移, 以科学记数法返回数据, 单位为 dB。

### ➤ 举例:

:DISPlay:WINDow:TRACe:Y:RLEVel:OFFSet 5      设置参考电平偏移 5dB

:DISPlay:WINDow:TRACe:Y:RLEVel:OFFSet?      查询返回 5.000000e+00

## :DISPlay:WINDow:TRACe:Y:DLINe

### ➤ 命令格式:

:DISPlay:WINDow:TRACe:Y:DLINe <amp;l>

:DISPlay:WINDow:TRACe:Y:DLINe?

### ➤ 功能描述:

调整显示线幅值。

<amp;l>: 连续实数, 默认单位为 dBm。

### ➤ 返回格式:

查询返回显示线幅值, 以科学记数法返回数据, 单位为 dBm。

### ➤ 举例:

:DISPlay:WINDow:TRACe:Y:DLINe -40      设置显示线幅值为-40dBm

:DISPlay:WINDow:TRACe:Y:DLINe?      查询返回-4.000000e+01

## :DISPlay:WINDow:TRACe:Y:DLINe:STATe

### ➤ 命令格式:

:DISPlay:WINDow:TRACe:Y:DLINe:STATe {{1|ON} | {0|OFF}}

:DISPlay:WINDow:TRACe:Y:DLINe:STATe?

### ➤ 功能描述:

显示线显示开关。

1|ON: 显示

0|OFF: 不显示

### ➤ 返回格式:

查询返回显示线状态, 0 或 1。

### ➤ 举例:

:DISPlay:WINDow:TRACe:Y:DLINe:STATe ON      显示显示线

:DISPlay:WINDow:TRACe:Y:DLINe:STATe?      查询返回 1

## :DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing

### ➤ 命令格式:

:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing {LOGarithmic|LINear}

:DISPlay:WINDow:TRACe:Y[:SCALe]:SPACing?

### ➤ 功能描述:

选择 Y 轴显示刻度类型。

LOGarithmic: 对数。

LINear: 线性。

### ➤ 返回格式:

查询返回 Y 轴显示刻度类型, LOGarithmic 或 LINear。

### ➤ 举例:

:DISPlay:WINDow:TRACe:Y:SPACing LOGarithmic      Y 轴显示刻度类型为对数

:DISPlay:WINDow:TRACe:Y:SPACing?                      查询返回 LOGarithmic

## INITiate 命令

### :INITiate:CONTinuous

### ➤ 命令格式:

:INITiate:CONTinuous {{1|ON} | {0|OFF}}

:INITiate:CONTinuous?

### ➤ 功能描述:

单次连续扫描切换。

1|ON: 连续扫描

0|OFF: 单次扫描

### ➤ 返回格式:

查询返回是否为连续扫描, 0 或 1。

### ➤ 举例:

:INITiate:CONTinuous ON                      连续扫描

:INITiate:CONTinuous?                      查询返回 1

## FETCh 命令

### :FETCh:CHPower?

➤ **命令格式:**

:FETCh:CHPower?

➤ **功能描述:**

查询通道功率测试全部结果，包括通道功率和通道功率密度。通道功率单位为 dBm，通道功率密度单位为 dBm/Hz。

➤ **返回格式:**

查询返回通道功率测试的全部结果，以科学记数法返回数据。

**举例:**

:FETCh:CHPower?

查询返回通道功率测试全部结果

### :FETCh:CHPower:CHPower?

➤ **命令格式:**

:FETCh:CHPower:CHPower?

➤ **功能描述:**

查询通道功率测试的通道功率，单位为 dBm。

➤ **返回格式:**

查询返回通道功率，以科学记数法返回数据。

**举例:**

:FETCh:CHPower:CHPower?

查询返回通道功率

### :FETCh:CHPower:DENSity?

➤ **命令格式:**

:FETCh:CHPower:DENSity?

➤ **功能描述:**

查询通道功率测试的通道功率密度，单位为 dBm/Hz。

➤ **返回格式:**

查询返回通道功率密度，以科学记数法返回数据。

**举例:**

:FETCh:CHPower:DENSity?

查询返回通道功率密度

### :FETCh:TPower?

➤ **命令格式:**

:FETCh:TPower?

➤ **功能描述:**

查询时域功率测试的总功率，单位为 dBm。





## **:FETCh:TOI:LOW:TONE**

➤ **命令格式:**

:FETCh:TOI:LOW:TONE?

➤ **功能描述:**

查询三阶交调测试的低基频的频率和幅值，频率单位 Hz，幅值单位 dBm。

➤ **返回格式:**

查询返回低基频的频率和幅值，以科学记数法返回数据。

**举例:**

:FETCh:TOI:LOW:TONE?                      查询返回低基频的频率和幅值

## **:FETCh:TOI:UP:TONE**

➤ **命令格式:**

:FETCh:TOI:UP:TONE?

➤ **功能描述:**

查询三阶交调测试的高基频的频率和幅值，频率单位 Hz，幅值单位 dBm。

➤ **返回格式:**

查询返回高基频的频率和幅值，以科学记数法返回数据。

**举例:**

:FETCh:TOI:UP:TONE?                      查询返回高基频的频率和幅值

## **:FETCh:TOI:IPT?**

➤ **命令格式:**

:FETCh:TOI:IPT?

➤ **功能描述:**

查询三阶交调测试的三阶交调结果。包括三阶交调系数，单位 dBc，三阶交调截点，单位 dBm。

➤ **返回格式:**

查询返回三阶交调结果，以科学记数法返回数据。

**举例:**

:FETCh:TOI:IPT?                      查询返回三阶交调结果

## **:FETCh:ACPower?**

➤ **命令格式:**

:FETCh:ACPower?

➤ **功能描述:**

查询相邻信道功率全部测试结果，依次是主信道功率，前一信道功率，前一信道相对功率，后一信道功率，后一信道相对功率。信道功率单位为 dBm，信道相对功率单位为 dBc。

➤ **返回格式:**

查询返回相邻信道功率全部测试结果，以科学记数法返回数据。

**举例:**

:FETCh:ACPower?

查询返回相邻信道功率全部测试结果

:FETCh:ACPower:MAIN?

➤ **命令格式:**

:FETCh:ACPower:MAIN?

➤ **功能描述:**

查询相邻信道功率全部测试的主信道功率。

➤ **返回格式:**

查询返回主信道功率，以科学记数法返回数据。

**举例:**

:FETCh:ACPower:MAIN?

查询返回主信道功率

:FETCh:ACPower:LOWer?

➤ **命令格式:**

:FETCh:ACPower:LOWer?

➤ **功能描述:**

查询相邻信道功率测试的前一信道功率及前一信道相对功率。信道功率单位为 dBm，信道功率单位为 dBc。

➤ **返回格式:**

查询返回前一信道功率，以科学记数法返回数据。

**举例:**

:FETCh:ACPower:LOWer?

查询返回前一信道功率

:FETCh:ACPower:UPPer?

➤ **命令格式:**

:FETCh:ACPower:UPPer?

➤ **功能描述:**

查询相邻信道功率的后一信道功率和后一信道相对功率。信道功率单位为 dBm，信道功率单位为 dBc。

➤ **返回格式:**

查询返回后一信道功率，以科学记数法返回数据。

**举例:**

:FETCh:ACPower:UPPer?

查询返回后一信道功率

:FETCh:CNRatio?

➤ **命令格式:**

:FETCh:CNRatio?

➤ **功能描述:**

查询载噪比测试的全部结果，依次是负载功率，噪声功率，载噪比。负载功率和噪声功率的单位是 dBm，载噪比的单位为 dB。

➤ **返回格式:**



值单位 dBc。

➤ **返回格式:**

查询返回全部谐波幅值，以科学记数法返回数据。

**举例:**

:FETCh:HARMonics:AMPLitude:ALL?          查询返回全部谐波幅值

**:FETCh:HARMonics:AMPLitude<n>?**

➤ **命令格式:**

:FETCh:HARMonics:AMPLitude<n>?

➤ **功能描述:**

查询谐波测试的指定次数谐波幅值，基波幅值单位 dBm，其它 n 次谐波幅值单位 dBc。

<n>: 谐波次数，整数，取值范围 1-10，1 次谐波就是谐波基波

➤ **返回格式:**

查询返回指定次数谐波幅值，以科学记数法返回数据。

**举例:**

:FETCh:HARMonics:AMPLitude2?          查询返回 2 次数谐波幅值

**:FETCh:HARMonics:FREQUENCY:ALL?**

➤ **命令格式:**

:FETCh:HARMonics:FREQUENCY:ALL?

➤ **功能描述:**

查询谐波测试的前 10 次谐波的频率值，以基波到 10 次谐波依次排列，单位为 Hz。

➤ **返回格式:**

查询返回指定次数谐波幅值，以科学记数法返回数据。

**举例:**

:FETCh:HARMonics:FREQUENCY:ALL?          查询前 10 次谐波所有频率

**:FETCh:HARMonics:FREQUENCY<n>?**

➤ **命令格式:**

:FETCh:HARMonics:FREQUENCY<n>?

➤ **功能描述:**

查询谐波测试的指定次数谐波频率，单位为 Hz。

<n>: 谐波次数，整数，取值范围 1-10，1 次谐波就是谐波基波

➤ **返回格式:**

查询返回指定次数谐波频率，以科学记数法返回数据。

**举例:**

:FETCh:HARMonics:FREQUENCY3?          查询返回 3 次数谐波频率

## :FETCh:HARMonics:DISToRtion?

➤ **命令格式:**

:FETCh:HARMonics:DISToRtion?

➤ **功能描述:**

查询谐波测试的谐波失真结果，依次包含谐波失真百分比和谐波失真相对功率。谐波失真百分比单位为%，谐波失真相对功率单位为 dBc。

➤ **返回格式:**

查询返回谐波失真结果，以科学记数法返回数据。

**举例:**

:FETCh:HARMonics:DISToRtion?                      查询返回谐波失真结果

## MMEMoRY 命令

### :MMEMoRY:LOAD:CORRection

➤ **命令格式:**

:MMEMoRY:LOAD:CORRection {<integer>,<filename>}

➤ **功能描述:**

指定修正加载数据。

<integer>: 修正序列号，连续整数，取值范围 1-10

<filename>: 文件名，文件后缀.corr

➤ **返回格式:**

无返回。

**举例:**

:MMEMoRY:LOAD:CORRection 1, "test.corr"                      修正 1 加载数据文件 test.corr

### :MMEMoRY:LOAD:LIMit

➤ **命令格式:**

:MMEMoRY:LOAD:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6,<filename>}

➤ **功能描述:**

指定限值加载数据。

LLINE1-LLINE6: 依次对应限值 1 到限值 6

<filename>: 文件名，文件后缀.limit

➤ **返回格式:**

无返回。

**举例:**

:MMEMoRY:LOAD:LIMit LLINE1, "test.limit "                      限值 1 加载数据文件 test.limit

## :MMEMory:LOAD:STATe

➤ **命令格式:**

:MMEMory:LOAD:STATe <filename>

➤ **功能描述:**

加载寄存器状态数据。

<filename>: 文件名, 文件后缀.state

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:LOAD:STATe "test.state"                    加载寄存器状态文件 test.state

## :MMEMory:LOAD:TRACe

➤ **命令格式:**

:MMEMory:LOAD:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, <filename>}

➤ **功能描述:**

指定迹线加载数据。

TRACE1-TRACE6: 依次对应迹线 1 到迹线 6

<filename>: 文件名, 文件后缀.trace

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:LOAD:TRACe TRACE1, "test.trace"            迹线 1 加载数据文件 test.trace

## :MMEMory:STORe:CORRection

➤ **命令格式:**

:MMEMory:STORe:CORRection {<integer>, <filename>}

➤ **功能描述:**

保存指定修正数据到默认目录。

<integer>: 修正序列号, 连续整数, 取值范围 1-10

<filename>: 文件名, 文件后缀.corr

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:CORRection 1, "test.corr"            保存修正 1 数据到文件 test.corr

## :MMEMory:STORe:LIMit

➤ **命令格式:**

:MMEMory:STORe:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6, <filename>}

➤ **功能描述:**

保存指定限值数据到默认目录。

LLINE1-LLINE6: 依次对应限值 1 到限值 6

<filename>: 文件名, 文件后缀.limit

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:LIMit LLINE1, "test.limit"                      保存限值 1 数据到文件 test.limit

**:MMEMory:STORe:STATe**

➤ **命令格式:**

:MMEMory:STORe:STATe <filename>

➤ **功能描述:**

保存寄存器状态到文件。

<filename>: 文件名, 文件后缀.state

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:STATe "test.state"                      保存寄存器状态到文件 test.state

**:MMEMory:STORe:TRACe**

➤ **命令格式:**

:MMEMory:STORe:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, <filename>}

➤ **功能描述:**

保存指定迹线数据到文件。

TRACE1-TRACE6: 依次对应迹线 1 到迹线 6

<filename>: 文件名, 文件后缀.trace

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:TRACe TRACE1, "test.trace"                      保存迹线 1 的数据到文件 test.trace

## OUTPut 命令

**:OUTPut[:EXTeRnAl][:STATe]**

➤ **命令格式:**

:OUTPut[:EXTeRnAl][:STATe] {{1|ON} | {0|OFF}}

:OUTPut[:EXTeRnAl][:STATe]?

➤ **功能描述:**

跟踪源开关。



**举例：**

:READ:CHPower:DENSity?                          查询返回通道功率密度

**:READ:TPower?****➤ 命令格式：**

:READ:TPower?

**➤ 功能描述：**

执行一次时域功率测量并返回时域功率测试的总功率，单位为 dBm。

**➤ 返回格式：**

查询返回时域功率测试的总功率，以科学记数法返回数据。

**举例：**

:READ:TPower?                          查询返回时域功率测试结果

**:READ:OBWidth?****➤ 命令格式：**

:READ:OBWidth?

**➤ 功能描述：**

执行一次占用带宽测量并返回占用带宽所有测试结果，依次包括占用带宽，传输频率误差，xdB 带宽和总功率。占用带宽，传输频率误差和 xdB 带宽的单位为 Hz，总功率单位为 dBm。

**➤ 返回格式：**

查询返回占用带宽所有测试结果，以科学记数法返回数据。

**举例：**

:READ:OBWidth?                          查询返回占用带宽所有测试结果

**:READ:OBWidth:OBWidth?****➤ 命令格式：**

:READ:OBWidth:OBWidth?

**➤ 功能描述：**

执行一次占用带宽测量并返回占用带宽测试的占用带宽，单位默认 Hz。

**➤ 返回格式：**

查询返回占用带宽，以科学记数法返回数据。

**举例：**

:READ:OBWidth:OBWidth?                          查询返回占用带宽

**:READ:OBWidth:FERRor?****➤ 命令格式：**

:READ:OBWidth:FERRor?

**➤ 功能描述：**

执行一次占用带宽测量并返回占用带宽测试的传输频率误差，单位为 Hz。



➤ **功能描述：**

执行一次三阶交调测量并返回三阶交调测试的高三阶交调结果。包括频率，单位 Hz，幅值，单位 dBm，高三阶交调系数，单位 dBc，高三阶交调截点，单位 dBm。

➤ **返回格式：**

查询返回高三阶交调结果，以科学记数法返回数据。

**举例：**

:READ:TOI:UP:TRD?                      查询返回高三阶交调

**:READ:TOI:LOW:TONE**

➤ **命令格式：**

:READ:TOI:LOW:TONE?

➤ **功能描述：**

执行一次三阶交调测量并返回三阶交调测试的低基频的频率和幅值，频率单位 Hz，幅值单位 dBm。

➤ **返回格式：**

查询返回低基频的频率和幅值，以科学记数法返回数据。

**举例：**

:READ:TOI:LOW:TONE?                      查询返回低基频的频率和幅值

**:READ:TOI:UP:TONE**

➤ **命令格式：**

:READ:TOI:UP:TONE?

➤ **功能描述：**

执行一次三阶交调测量并返回三阶交调测试的高基频的频率和幅值，频率单位 Hz，幅值单位 dBm。

➤ **返回格式：**

查询返回高基频的频率和幅值，以科学记数法返回数据。

**举例：**

:READ:TOI:UP:TONE?                      查询返回高基频的频率和幅值

**:READ:TOI:IPT?**

➤ **命令格式：**

:READ:TOI:IPT?

➤ **功能描述：**

执行一次三阶交调测量并返回三阶交调测试的三阶交调结果。包括三阶交调系数，单位 dBc，三阶交调截点，单位 dBm。

➤ **返回格式：**

查询返回三阶交调结果，以科学记数法返回数据。

**举例：**

:READ:TOI:IPT?                              查询返回三阶交调结果

#### :READ:ACPower?

➤ **命令格式:**

:READ:ACPower?

➤ **功能描述:**

执行一次相邻信道功率测量并返回相邻信道功率全部测试结果，依次是主信道功率，前一信道功率，前一信道相对功率，后一信道功率，后一信道相对功率。信道功率单位为 dBm，信道相对功率单位为 dBc。

➤ **返回格式:**

查询返回相邻信道功率全部测试结果，以科学记数法返回数据。

**举例:**

:READ:ACPower?

查询返回相邻信道功率全部测试结果

#### :READ:ACPower:MAIN?

➤ **命令格式:**

:READ:ACPower:MAIN?

➤ **功能描述:**

执行一次相邻信道功率测量并返回相邻信道功率全部测试的主信道功率。

➤ **返回格式:**

查询返回主信道功率，以科学记数法返回数据。

**举例:**

:READ:ACPower:MAIN?

查询返回主信道功率

#### :READ:ACPower:LOWer?

➤ **命令格式:**

:READ:ACPower:LOWer?

➤ **功能描述:**

执行一次相邻信道功率测量并返回相邻信道功率测试的前一信道功率及前一信道相对功率。信道功率单位为 dBm，信道功率单位为 dBc。

➤ **返回格式:**

查询返回前一信道功率，以科学记数法返回数据。

**举例:**

:READ:ACPower:LOWer?

查询返回前一信道功率

#### :READ:ACPower:UPPer?

➤ **命令格式:**

:READ:ACPower:UPPer?

➤ **功能描述:**

执行一次相邻信道功率测量并返回相邻信道功率的后一信道功率和后一信道相对功率。信道功率单位为 dBm，信道功率单位为 dBc。

➤ **返回格式:**

查询返回后一信道功率，以科学记数法返回数据。

**举例：**

:READ:ACPower:UPPer?                      查询返回后一信道功率

**:READ:CNRatio?**

➤ **命令格式：**

:READ:CNRatio?

➤ **功能描述：**

执行一次载噪比测量并返回载噪比测试的全部结果，依次是负载功率，噪声功率，载噪比。负载功率和噪声功率的单位是 dBm，载噪比的单位为 dB。

➤ **返回格式：**

查询返回载噪比测试的全部结果，以科学记数法返回数据。

**举例：**

:READ:CNRatio?                              查询返回载噪比测试的全部结果

**:READ:CNRatio:CARRIER?**

➤ **命令格式：**

:READ:CNRatio:CARRIER?

➤ **功能描述：**

执行一次载噪比测量并返回载噪比测试的负载功率，单位为 dBm。

➤ **返回格式：**

查询返回载噪比测试的负载功率，以科学记数法返回数据。

**举例：**

:READ:CNRatio:CARRIER?                查询返回载噪比测试的负载功率

**:READ:CNRatio:NOISE?**

➤ **命令格式：**

:READ:CNRatio:NOISE?

➤ **功能描述：**

执行一次载噪比测量并返回载噪比测试的噪声功率，单位为 dBm。

➤ **返回格式：**

查询返回载噪比测试的噪声功率，以科学记数法返回数据。

**举例：**

:READ:CNRatio:NOISE?                    查询返回载噪比测试的噪声功率

**:READ:CNRatio:CNRatio?**

➤ **命令格式：**

:READ:CNRatio:CNRatio?

➤ **功能描述：**



**:READ:HARMonics:FREQuency<n>?**

➤ **命令格式:**

:READ:HARMonics:FREQuency<n>?

➤ **功能描述:**

执行一次谐波测量并返回谐波测试的指定次数谐波频率，单位为 Hz。

<n>: 谐波次数，整数，取值范围 1-10，1 次谐波就是谐波基波

➤ **返回格式:**

查询返回指定次数谐波频率，以科学记数法返回数据。

**举例:**

:READ:HARMonics:FREQuency3?                      查询返回 3 次数谐波频率

**:READ:HARMonics:DISToRtion?**

➤ **命令格式:**

:READ:HARMonics:DISToRtion?

➤ **功能描述:**

执行一次谐波测量并返回谐波测试的谐波失真结果，依次包含谐波失真百分比和谐波失真相对功率。谐波失真百分比单位为%，谐波失真相对功率单位为 dBc。

➤ **返回格式:**

查询返回谐波失真结果，以科学记数法返回数据。

**举例:**

:READ:HARMonics:DISToRtion?                      查询返回谐波失真结果

## SENSe 命令

**[[:SENSe]:FREQuency:STARt**

➤ **命令格式:**

[[:SENSe]:FREQuency:STARt <freq>

[[:SENSe]:FREQuency:STARt?

➤ **功能描述:**

设置扫频起始频率。

<freq>: 连续实数，默认单位为 Hz。

起始频率范围是 0Hz 到最大频率-100Hz。

UTS1015B 的最大频率是 1.5GHz，UTS3036B 的最大频率是 3.6GHz，UTS3084B 的最大频率是 8.4GHz。

➤ **返回格式:**

查询返回起始频率值，以科学计数法返回数据，单位为 Hz。

➤ **举例:**

:FREQuency:STARt 10MHz      设置扫频起始频率为 10MHz

:FREQuency:STARt?              查询返回 1.000000e+07

## **[ :SENSe ] :FREQuency :STOP**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :STOP <freq>

[ :SENSe ] :FREQuency :STOP?

### ➤ **功能描述:**

设置扫频截止频率。

<freq>: 连续实数, 默认单位为 Hz。

截止频率范围是 100Hz 到最大频率。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

### ➤ **返回格式:**

查询返回截止频率值, 以科学计数法返回数据, 单位为 Hz。

### ➤ **举例:**

:FREQuency :STOP 1GHz 设置扫频截止频率为 1GHz

:FREQuency :STOP? 查询返回 1.000000e+09

## **[ :SENSe ] :FREQuency :CENTer**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :CENTer <freq>

[ :SENSe ] :FREQuency :CENTer?

### ➤ **功能描述:**

设置扫频中心频率。

<freq>: 连续实数, 默认单位为 Hz。

频率范围是 50Hz 到最大频率-50Hz。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

### ➤ **返回格式:**

查询返回中心频率值, 以科学计数法返回数据, 单位为 Hz。

### ➤ **举例:**

:FREQuency :CENTer 1GHz 设置扫频中心频率为 1GHz

:FREQuency :CENTer? 查询返回 1.000000e+09

## **[ :SENSe ] :FREQuency :SPAN**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :SPAN <freq>

[ :SENSe ] :FREQuency :SPAN?

### ➤ **功能描述:**

设置扫宽。

<freq>: 连续实数, 默认单位为 Hz。

扫宽范围是 100Hz 到最大频率。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

### ➤ **返回格式:**

查询返回扫宽值，以科学计数法返回数据，单位 Hz。

➤ **举例：**

:FREQUENCY:SPAN 1GHz 设置扫宽为 1GHz

:FREQUENCY:SPAN? 查询返回 1.000000e+09

**[ :SENSe ] :FREQUENCY :SPAN :FULL**

➤ **命令格式：**

[ :SENSe ] :FREQUENCY :SPAN :FULL

➤ **功能描述：**

设置扫宽为全扫宽(最大扫宽)。

UTS1015B 的最大扫宽是 1.5GHz，UTS3036B 的最大扫宽是 3.6GHz，UTS3084B 的最大扫宽是 8.4GHz。

➤ **返回格式：**

无返回。

➤ **举例：**

:FREQUENCY:SPAN:FULL 设置扫宽为全扫宽

**[ :SENSe ] :FREQUENCY :SPAN :ZERO**

➤ **命令格式：**

[ :SENSe ] :FREQUENCY :SPAN :ZERO

➤ **功能描述：**

设置扫宽为零扫宽。

➤ **返回格式：**

无返回。

➤ **举例：**

:FREQUENCY:SPAN:ZERO 设置扫宽为零扫宽

**[ :SENSe ] :FREQUENCY :ZOOM :IN**

➤ **命令格式：**

[ :SENSe ] :FREQUENCY :ZOOM :IN

➤ **功能描述：**

缩小，就是增大扫宽，使信号不易识别。

➤ **返回格式：**

无返回。

➤ **举例：**

:FREQUENCY:ZOOM:IN 信号缩小

**[ :SENSe ] :FREQUENCY :ZOOM :OUT**

➤ **命令格式：**

[ :SENSe ] :FREQUENCY :ZOOM :OUT

- **功能描述：**  
放大，就是减小扫宽，使信号更突出。
- **返回格式：**  
无返回。
- **举例：**  
:FREQuency:ZOOM:OUT 信号放大

#### **[ :SENSe ] :FREQuency:SPAN:PREVious**

- **命令格式：**  
[ :SENSe ] :FREQuency:SPAN:PREVious
- **功能描述：**  
设置扫宽为上次扫宽。
- **返回格式：**  
无返回。
- **举例：**  
:FREQuency:SPAN:PREVious 设置扫宽为上次扫宽

#### **[ :SENSe ] :FREQuency:TUNE:IMMEDIATE**

- **命令格式：**  
[ :SENSe ] :FREQuency:TUNE:IMMEDIATE
- **功能描述：**  
设置自动调谐。
- **返回格式：**  
无返回。
- **举例：**  
:FREQuency:TUNE:IMMEDIATE 设置自动调谐

#### **[ :SENSe ] :FREQuency:OFFSet**

- **命令格式：**  
[ :SENSe ] :FREQuency:OFFSet <freq>  
[ :SENSe ] :FREQuency:OFFSet?
- **功能描述：**  
调整频率偏移。  
<freq>: 连续实数，默认单位为 Hz。
- **返回格式：**  
查询返回频率偏移值，以科学计数法返回数据，单位为 Hz。
- **举例：**  
:FREQuency:OFFSet 10MHz 设置频率偏移为 10MHz  
:FREQuency:OFFSet? 查询返回 1.000000e+07

## **[ :SENSe ] :FREQuency :CENTer :STEP [ :INCRement ]**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :CENTer :STEP [ :INCRement ] <freq>

[ :SENSe ] :FREQuency :CENTer :STEP [ :INCRement ] ?

### ➤ **功能描述:**

调整中心频率步进。

<freq>: 连续实数, 默认单位为 Hz。

### ➤ **返回格式:**

查询返回中心频率步进值, 以科学计数法返回数据, 单位为 Hz。

### ➤ **举例:**

:FREQuency :CENTer :STEP 10MHz 设置中心频率步进为 10MHz

:FREQuency :CENTer :STEP? 查询返回 1.000000e+07

## **[ :SENSe ] :FREQuency :CENTer :STEP :AUTO**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :CENTer :STEP :AUTO {{1|ON} | {0|OFF}}

[ :SENSe ] :FREQuency :CENTer :STEP :AUTO ?

### ➤ **功能描述:**

手动/自动中心频率步进切换。

1|ON: 自动

0|OFF: 手动

### ➤ **返回格式:**

查询返回自动中心频率步进状态, 0 或 1。

### ➤ **举例:**

:FREQuency :CENTer :STEP :AUTO ON 设置自动中心频率步进

:FREQuency :CENTer :STEP :AUTO? 查询返回 1

## **[ :SENSe ] :POWer [ :RF ] :ATTenuation**

### ➤ **命令格式:**

[ :SENSe ] :POWer [ :RF ] :ATTenuation <amp;l>

[ :SENSe ] :POWer [ :RF ] :ATTenuation ?

### ➤ **功能描述:**

调整输入衰减。

<amp;l>: 连续整数, 默认单位为 dB, 取值范围 0dB 到 51dB

### ➤ **返回格式:**

查询返回输入衰减值, 单位为 dB。

### ➤ **举例:**

:POWer :ATTenuation 10 设置输入衰减 10dB

:POWer :ATTenuation? 查询返回 10

## **[ :SENSe ] :POWer [ :RF ] :ATTenuat ion :AUTO**

### ➤ **命令格式:**

[ :SENSe ] :POWer [ :RF ] :ATTenuat ion :AUTO { {1|ON} | {0|OFF} }

[ :SENSe ] :POWer [ :RF ] :ATTenuat ion :AUTO?

### ➤ **功能描述:**

手动/自动输入衰减切换。

1|ON: 自动

0|OFF: 手动

### ➤ **返回格式:**

查询返回自动输入衰减状态, 0 或 1。

### ➤ **举例:**

:POWer :ATTenuat ion :AUTO ON      设置自动输入衰减

:POWer :ATTenuat ion :AUTO?      查询返回 1

## **[ :SENSe ] :POWer [ :RF ] :GAIN :STATe**

### ➤ **命令格式:**

[ :SENSe ] :POWer [ :RF ] :GAIN :STATe { {1|ON} | {0|OFF} }

[ :SENSe ] :POWer [ :RF ] :GAIN :STATe?

### ➤ **功能描述:**

前置放大器开关。

### ➤ **返回格式:**

查询返回前置放大器开关状态, 0 或 1。

### ➤ **举例:**

:POWer :GAIN :STATe ON      打开前置放大器

:POWer :GAIN :STATe?      查询返回 1

## **[ :SENSe ] :CORRection :IMPedance [ :INPut ] [ :MAGNitude ]**

### **:INPut :IMPedance**

### ➤ **命令格式:**

[ :SENSe ] :CORRection :IMPedance [ :INPut ] [ :MAGNitude ] {50|75}

[ :SENSe ] :CORRection :IMPedance [ :INPut ] [ :MAGNitude ]?

:INPut :IMPedance {50|75}

:INPut :IMPedance?

### ➤ **功能描述:**

选择输入阻抗, 50 Ω 或 75 Ω。

### ➤ **返回格式:**

查询返回输入阻抗值, 50 Ω 或 75 Ω。

### ➤ **举例:**

:CORRection :IMPedance 50      设置输入阻抗为 50 Ω

:CORREction:IMPedance?            查询返回 50  
:INPut:IMPedance 50                设置输入阻抗为 50 Ω  
:INPut:IMPedance?                查询返回 50

#### [[:SENSe]:BANDwidth|BWIDth[:RESolution]

➤ **命令格式:**

[[:SENSe]:BANDwidth|BWIDth[:RESolution] <freq>  
[:SENSe]:BANDwidth|BWIDth[:RESolution]?

➤ **功能描述:**

调整分辨率带宽。

<freq>: 离散实数, 默认单位为 Hz, 取值范围 1Hz 到最大分辨率带宽, 以 1-3-10 为步进。  
UTS1015B 的最大分辨率带宽为 1MHz, UTS3036B 的最大分辨率带宽为 3MHz。

➤ **返回格式:**

查询返回分辨率带宽值, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:BANDwidth 1MHz            设置分辨率带宽为 1MHz  
:BANDwidth?                查询返回 1.000000e+06

#### [[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO

➤ **命令格式:**

[[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO {{1|ON} | {0|OFF}}  
[:SENSe]:BANDwidth|BWIDth[:RESolution]:AUTO?

➤ **功能描述:**

自动/手动分辨率带宽切换。

1|ON: 自动

0|OFF: 手动

➤ **返回格式:**

查询返回自动分辨率带宽状态, 0 或 1。

➤ **举例:**

:BANDwidth:AUTO ON            设置自动分辨率带宽  
:BANDwidth:AUTO?                查询返回 1

#### [[:SENSe]:BANDwidth|BWIDth:VIDeo

➤ **命令格式:**

[[:SENSe]:BANDwidth|BWIDth:VIDeo <freq>  
[:SENSe]:BANDwidth|BWIDth:VIDeo?

➤ **功能描述:**

调整视频带宽。

<freq>: 离散实数, 默认单位为 Hz, 取值范围 1Hz 到最大视频带宽, 以 1-3-10 为步进。  
UTS1015B 的最大视频带宽为 1MHz, UTS3036B 的最大视频带宽为 3MHz。

- **返回格式:**  
查询返回分辨率带宽状态, 以科学计数法返回数据, 单位为 Hz。
- **举例:**  
:BANDwidth:VIdEo 1MHz      设置视频带宽为 1MHz  
:BANDwidth:VIdEo?          查询返回 1.000000e+06

#### **[[:SENSe]:BANDwidth|BWIDth:VIdEo:AUTO**

- **命令格式:**  
[:SENSe]:BANDwidth|BWIDth:VIdEo:AUTO {{1|ON} | {0|OFF}}  
[:SENSe]:BANDwidth|BWIDth:VIdEo:AUTO?
- **功能描述:**  
自动/手动视频带宽切换。  
1|ON: 自动  
0|OFF: 手动
- **返回格式:**  
查询返回自动视频带宽状态, 0 或 1。
- **举例:**  
:BANDwidth:VIdEo:AUTO ON 设置自动视频带宽  
:BANDwidth:VIdEo:AUTO?          查询返回 1

#### **[[:SENSe]:BANDwidth|BWIDth:VIdEo:RATio**

- **命令格式:**  
[:SENSe]:BANDwidth|BWIDth:VIdEo:RATio <ratio>  
[:SENSe]:BANDwidth|BWIDth:VIdEo:RATio?
- **功能描述:**  
设置视分比, 视频带宽与分辨率带宽比值。  
<ratio>: 离散实数, UTS1015B 的取值范围为 0.000001 (1Hz/1MHz) 到 1000000 (1MHz/1Hz), UTS3036B 的最大视频带宽为 0.000003 (1Hz/3MHz) 到 3000000 (3MHz/1Hz)。
- **返回格式:**  
查询返回视分比值, 以科学计数法返回数据。
- **举例:**  
:BANDwidth:VIdEo:RATio 0.1      设置视分比为 0.1  
:BANDwidth:VIdEo:RATio?          查询返回 1.000000e-01

#### **[[:SENSe]:BANDwidth|BWIDth:SHAPE**

- **命令格式:**  
[:SENSe]:BANDwidth|BWIDth:SHAPE {GAUSSian|FLATtop}  
[:SENSe]:BANDwidth|BWIDth:SHAPE?
- **功能描述:**  
选择滤波器。

GAUSSian: 高斯滤波器。  
FLATtop: 平顶窗滤波器。

- **返回格式:**  
查询返回滤波器类型, GAUSSian 或 FLATtop。
- **举例:**  
:BANDwidth:SHAPE GAUSSian 选择高斯滤波器  
:BANDwidth:SHAPE? 查询返回 GAUSSian

#### [ :SENSe ] :SWEep:TIME

- **命令格式:**  
[ :SENSe ] :SWEep:TIME <time>  
[ :SENSe ] :SWEep:TIME?
- **功能描述:**  
设置扫描时间。  
<time>: 时间值, 单位为 s, 零扫宽取值范围 1 μs 到 4ks, 非零扫宽取值范围 1ms 到 4ks。
- **返回格式:**  
查询返回扫描时间, 以科学计数法返回数据, 单位为 s。
- **举例:**  
:SWEep:TIME 0.1 设置扫描时间为 100ms  
:SWEep:TIME? 查询返回 1.000000e-01

#### [ :SENSe ] :SWEep:TIME:AUTO

- **命令格式:**  
[ :SENSe ] :SWEep:TIME:AUTO {{1|ON} | {0|OFF}}  
[ :SENSe ] :SWEep:TIME:AUTO?
- **功能描述:**  
自动/手动扫描时间切换。  
1|ON: 自动  
0|OFF: 手动
- **返回格式:**  
查询返回扫描时间状态, 0 或 1。
- **举例:**  
:SWEep:TIME:AUTO ON 自动扫描时间打开  
:SWEep:TIME:AUTO? 查询返回 1

#### [ :SENSe ] :SWEep:TIME:AUTO:RULEs

- **命令格式:**  
[ :SENSe ] :SWEep:TIME:AUTO:RULEs {NORMAl | ACCuracy}  
[ :SENSe ] :SWEep:TIME:AUTO:RULEs?
- **功能描述:**

选择扫描时间规则。

NORMal：普通。

ACCuracy：精确。

➤ **返回格式：**

查询返回扫描时间规则，NORMal 或 ACCuracy。

➤ **举例：**

:SWEep:TIME:AUTO:RULes ACCuracy 选择精确扫描时间

:SWEep:TIME:AUTO:RULes? 查询返回 ACCuracy

## [ :SENSe ] :SWEep:TYPE

➤ **命令格式：**

[ :SENSe ] :SWEep:TYPE {SWEep|FFT}

[ :SENSe ] :SWEep:TYPE?

➤ **功能描述：**

选择扫描模式。

SWEep：扫频。

FFT：FFT。

➤ **返回格式：**

查询返回扫描模式，SWEep 或 FFT。

➤ **举例：**

:SWEep:TYPE SWEep 选择扫频模式

:SWEep:TYPE? 查询返回 SWEep

## [ :SENSe ] :SWEep:TYPE:AUTO

➤ **命令格式：**

[ :SENSe ] :SWEep:TYPE:AUTO {{1|ON} | {0|OFF}}

[ :SENSe ] :SWEep:TYPE:AUTO?

➤ **功能描述：**

自动/手动扫描模式切换。

1|ON：自动。

0|OFF：手动。

➤ **返回格式：**

查询返回扫描模式状态，0 或 1。

➤ **举例：**

:SWEep:TYPE:AUTO ON 设置自动扫描模式

:SWEep:TYPE:AUTO? 查询返回 1

## [ :SENSe ] :SWEep:POINts

➤ **命令格式：**

[ :SENSe ] :SWEep:POINts <integer>

[ :SENSe ] :SWEep :POINts?

➤ **功能描述:**

调整扫描点数。

<integer>: 点数为整数, 取值范围 11 到最大点数。

UTS1015B 的最大点数为 10001, UTS3036B 的最大点数为 40001。

➤ **返回格式:**

查询返回扫描点数。

➤ **举例:**

:SWEep:POINts 1001     设置扫描点数为 1001

:SWEep:POINts?         查询返回 1001

[ :SENSe ] :DETEctor [ :FUNCTion ]

➤ **命令格式:**

[ :SENSe ] :DETEctor [ :FUNCTion ] {SAMPlE|POSitive|NEGative|NORMal|AVERage}

[ :SENSe ] :DETEctor [ :FUNCTion ]?

➤ **功能描述:**

选择当前迹线检波器类型, 用于部分高级测量, 频谱扫描、三阶交调谐波不适用。

SAMPlE: 采样检波

POSitive: 峰值检波

NEGative: 负峰值检波

NORMal: 常态检波

AVERage: 平均检波

➤ **返回格式:**

查询返回当前迹线检波器类型, SAMPlE、POSitive、NEGative、NORMal 或 AVERage。

➤ **举例:**

:DETEctor POSitive         选择当前迹线检波器类型为峰值检波

:DETEctor?                 查询返回 POSitive

[ :SENSe ] :DETEctor :TRACe<n>

➤ **命令格式:**

[ :SENSe ] :DETEctor :TRACe<n> {SAMPlE|POSitive|NEGative|NORMal|AVERage}

[ :SENSe ] :DETEctor :TRACe<n>?

➤ **功能描述:**

选择指定迹线检波器类型。

<n>: 迹线序列号, 取值范围 1 到最大迹线数。

UTS1015B 的最大迹线数为 4, UTS3036B 的最大迹线数为 6。

SAMPlE: 采样检波

POSitive: 峰值检波

NEGative: 负峰值检波

NORMal: 常态检波

AVERage: 平均检波

➤ **返回格式:**

查询返回当前迹线检波器类型，SAMPLE、POSITIVE、NEGATIVE、NORMAL 或 AVERAGE。

➤ **举例：**

:DETECTOR:TRACE1 POSITIVE      选择迹线 1 检波器类型为峰值检波  
:DETECTOR:TRACE1?              查询返回 POSITIVE

### **[ :SENSE ] :CORRECTION:SELECT**

➤ **命令格式：**

[ :SENSE ] :CORRECTION:SELECT <integer>  
[ :SENSE ] :CORRECTION:SELECT?

➤ **功能描述：**

从修正序列中选择一个作为当前修正。  
<integer>：修正序列号，取值范围 1 到 10。

➤ **返回格式：**

查询返回当前修正序列号，范围 1 到 10。

➤ **举例：**

:CORRECTION:SELECT 2            选择修正 2  
:CORRECTION:SELECT?            查询返回 2

### **[ :SENSE ] :CORRECTION:CSET<n>[ :STATE ]**

➤ **命令格式：**

[ :SENSE ] :CORRECTION:CSET<n>[ :STATE ] {{1|ON} | {0|OFF}}  
[ :SENSE ] :CORRECTION:CSET<n>[ :STATE ]?

➤ **功能描述：**

指定修正开关。  
<n>：修正序列号，取值范围 1 到 10。

➤ **返回格式：**

查询返回指定修正开关状态，0 或 1。

➤ **举例：**

:CORRECTION:CSET1 ON            打开修正 1  
:CORRECTION:CSET1?            查询返回 1

### **[ :SENSE ] :CORRECTION:CSET:ALL [ :STATE ]**

➤ **命令格式：**

[ :SENSE ] :CORRECTION:CSET:ALL [ :STATE ]

➤ **功能描述：**

关闭所有修正。

➤ **返回格式：**

无返回。

➤ **举例：**

:CORRECTION:CSET:ALL            关闭所有修正

## **[ :SENSe ] :CORRection:CSET<n>:DATA**

### ➤ **命令格式:**

[ :SENSe ] :CORRection:CSET<n>:DATA {<freq>, <ampI>, <freq>, <ampI>, ... }

[ :SENSe ] :CORRection:CSET<n>:DATA?

### ➤ **功能描述:**

编辑指定修正数据。

<n>: 修正序列号, 取值范围 1 到 10。

<freq>: 修正点频率, 单位 Hz。

<ampI>: 修正点幅度, 单位 dB。

### ➤ **返回格式:**

查询返回指定修正数据, 以科学记数法返回数据, 以 {频率, 幅值, 频率, 幅值, ...} 结构返回, 频率单位为 Hz, 幅值单位为 dB。

### ➤ **举例:**

:CORRection:CSET1:DATA 10000000, 5

编辑修正 1 (10000000, 5)

:CORRection:CSET1:DATA?

查询返回 1.000000e+07, 5.000000e+00

## **[ :SENSe ] :CORRection:CSET<n>:DELeTe**

### ➤ **命令格式:**

[ :SENSe ] :CORRection:CSET<n>:DELeTe

### ➤ **功能描述:**

删除指定修正数据。

<n>: 修正序列号, 取值 1-10

### ➤ **返回格式:**

无返回。

### ➤ **举例:**

:CORRection:CSET1:DELeTe      删除修正 1 数据

## **[ :SENSe ] :CORRection:CSET:ALL:DELeTe**

### ➤ **命令格式:**

[ :SENSe ] :CORRection:CSET:ALL:DELeTe

### ➤ **功能描述:**

删除所有修正数据。

### ➤ **返回格式:**

无返回。

### ➤ **举例:**

:CORRection:CSET:ALL:DELeTe      删除所有修正数据

## **[ :SENSe ] :DEMod**

### ➤ **命令格式:**

[ :SENSe ] :DEMod {OFF|FM|AM}

[ :SENSe ] :DEMod?

➤ **功能描述:**

选择解调类型。

OFF: 关闭解调

FM: 频率解调

AM: 幅度解调

➤ **返回格式:**

查询返回解调类型, OFF、FM 或 AM。

➤ **举例:**

:DEMod FM            选择频率解调

:DEMod?            查询返回 FM

:DEMod?            查询返回 FM

### [ :SENSe ] :AVERage:COUNT

➤ **命令格式:**

[ :SENSe ] :AVERage:COUNT <integer>

[ :SENSe ] :AVERage:COUNT?

➤ **功能描述:**

设置平均次数。

<integer>: 整数, 取值范围 1 到 999。

➤ **返回格式:**

查询返回平均次数, 取值 1 到 999。

➤ **举例:**

:AVERage:COUNT 10    设置平均次数为 10

:AVERage:COUNT?    查询返回 10

### [ :SENSe ] :AVERage:TYPE

➤ **命令格式:**

[ :SENSe ] :AVERage:TYPE {VOLTage|POWER|LOG}

[ :SENSe ] :AVERage:TYPE?

➤ **功能描述:**

设置平均类型。

VOLTage: 电压平均

POWER: 功率平均

LOG: 对数平均

➤ **返回格式:**

查询返回平均类型, VOLTage、POWER 或 LOG。

➤ **举例:**

:AVERage:TYPE VOLTage    设置电压平均

:AVERage:TYPE?            查询返回 VOLTage

## **[[:SENSe]:CHPower:AVERage[:STATe]**

### ➤ **命令格式:**

[[:SENSe]:CHPower:AVERage[:STATe] {{1|ON} | {0|OFF}}

[[:SENSe]:CHPower:AVERage[:STATe]?

### ➤ **功能描述:**

通道功率平均开关。

### ➤ **返回格式:**

查询返回通道功率平均开关状态，0 或 1。

### ➤ **举例:**

:CHPower:AVERage ON            打开通道功率平均

:CHPower:AVERage?            查询返回 1

## **[[:SENSe]:CHPower:AVERage:COUNT**

### ➤ **命令格式:**

[[:SENSe]:CHPower:AVERage:COUNT <integer>

[[:SENSe]:CHPower:AVERage:COUNT?

### ➤ **功能描述:**

设置通道功率平均次数。

<integer>: 整数，取值范围 1 到 999。

### ➤ **返回格式:**

查询返回通道功率平均次数，范围 1 到 999。

### ➤ **举例:**

:CHPower:AVERage:COUNT 10 设置通道功率平均次数为 10

:CHPower:AVERage:COUNT?        查询返回 10

## **[[:SENSe]:CHPower:AVERage:TCONtrol**

### ➤ **命令格式:**

[[:SENSe]:CHPower:AVERage:TCONtrol {EXponential|REPeat}

[[:SENSe]:CHPower:AVERage:TCONtrol?

### ➤ **功能描述:**

设置通道功率平均模式。

EXponential: 指数

REPeat: 重复

### ➤ **返回格式:**

查询返回通道功率平均模式，EXponential 或 REPeat。

### ➤ **举例:**

:CHPower:AVERage:TCONtrol EXponential        设置通道功率指数平均

:CHPower:AVERage:TCONtrol?        查询返回 EXponential

## **[ :SENSe ] :CHPower :AVERage :TYPE**

### ➤ **命令格式:**

[ :SENSe ] :CHPower :AVERage :TYPE {VOLTage | POWER | LOG}

[ :SENSe ] :CHPower :AVERage :TYPE?

### ➤ **功能描述:**

设置通道功率平均类型。

VOLTage: 电压平均

POWER: 功率平均

LOG: 对数平均

### ➤ **返回格式:**

查询返回通道功率平均模式, VOLTage、POWER 或 LOG。

### ➤ **举例:**

:CHPower :AVERage :TYPE POWER                    设置通道功率功率平均

:CHPower :AVERage :TYPE?                            查询返回 POWER

## **[ :SENSe ] :CHPower :BANDwidth :INTEgration**

### ➤ **命令格式:**

[ :SENSe ] :CHPower :BANDwidth :INTEgration <freq>

[ :SENSe ] :CHPower :BANDwidth :INTEgration?

### ➤ **功能描述:**

设置通道功率积分带宽。

<freq>: 频率值, 单位 Hz。

### ➤ **返回格式:**

查询返回通道功率平均模式, 以科学计数法返回数据, 单位为 Hz。

### ➤ **举例:**

:CHPower :BANDwidth :INTEgration 2MHz            设置通道功率积分带宽 2MHz

:CHPower :BANDwidth :INTEgration?                查询返回 2.000000e+06

## **[ :SENSe ] :TPOWer :AVERage [ :STATe ]**

### ➤ **命令格式:**

[ :SENSe ] :TPOWer :AVERage [ :STATe ] {{1|ON} | {0|OFF}}

[ :SENSe ] :TPOWer :AVERage [ :STATe ]?

### ➤ **功能描述:**

时域功率平均开关。

### ➤ **返回格式:**

查询返回时域功率平均开关状态, 0 或 1。

### ➤ **举例:**

:TPOWer :AVERage ON                            打开时域功率平均

:TPOWer :AVERage?                                查询返回 1

## **[ :SENSe ] : TPOWer : AVERAge : COUNT**

### ➤ **命令格式:**

[ :SENSe ] : TPOWer : AVERAge : COUNT < integer >

[ :SENSe ] : TPOWer : AVERAge : COUNT?

### ➤ **功能描述:**

设置时域功率平均次数。

< integer >: 整数, 取值 1 到 999。

### ➤ **返回格式:**

查询返回时域功率平均次数, 范围 1 到 999。

### ➤ **举例:**

: TPOWer : AVERAge : COUNT 10          设置时域功率平均次数为 10

: TPOWer : AVERAge : COUNT?          查询返回 10

## **[ :SENSe ] : TPOWer : AVERAge : TCONtrol**

### ➤ **命令格式:**

[ :SENSe ] : TPOWer : AVERAge : TCONtrol { EXPonential | REPeat }

[ :SENSe ] : TPOWer : AVERAge : TCONtrol?

### ➤ **功能描述:**

设置时域功率平均模式。

EXPonential: 指数

REPeat: 重复

### ➤ **返回格式:**

查询返回时域功率平均模式, EXPonential 或 REPeat。

### ➤ **举例:**

: TPOWer : AVERAge : TCONtrol EXPonential          设置时域功率指数平均

: TPOWer : AVERAge : TCONtrol?          查询返回 EXPonential

## **[ :SENSe ] : TPOWer : AVERAge : TYPE**

### ➤ **命令格式:**

[ :SENSe ] : TPOWer : AVERAge : TYPE { VOLTage | POWEr | LOG }

[ :SENSe ] : TPOWer : AVERAge : TYPE?

### ➤ **功能描述:**

设置时域功率平均类型。

VOLTage: 电压平均

POWEr: 功率平均

LOG: 对数平均

### ➤ **返回格式:**

查询返回时域功率平均模式, VOLTage、POWEr 或 LOG。

### ➤ **举例:**

: TPOWer : AVERAge : TYPE POWEr          设置时域功率功率平均

:TPOWer:AVERage:TYPE?

查询返回 POWEr

### [[:SENSe]:TPOWer:LLIMit

➤ **命令格式:**

[[:SENSe]:TPOWer:LLIMit <time>

[[:SENSe]:TPOWer:LLIMit?

➤ **功能描述:**

设置时域功率开始时间。

<time>: 时间值, 单位 s, 取值范围 0 到 10ms。

➤ **返回格式:**

查询返回时域功率开始时间, 以科学计数法返回数据, 单位为 s。

➤ **举例:**

:TPOWer:LLIMit 5ms            设置时域功率开始时间 5ms

:TPOWer:LLIMit?            查询返回 5.000000e-03

### [[:SENSe]:TPOWer:RLIMit

➤ **命令格式:**

[[:SENSe]:TPOWer:RLIMit <time>

[[:SENSe]:TPOWer:RLIMit?

➤ **功能描述:**

调整时域功率截止时间。

<time>: 连续实数, 单位为 s, 取值范围 0 到 10ms。

➤ **返回格式:**

查询返回时域功率截止时间, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:TPOWer:RLIMit 5ms            设置时域功率截止时间 5ms

:TPOWer:RLIMit?            查询返回 5.000000e-03

### [[:SENSe]:OBWidth:AVERage[:STATe]

➤ **命令格式:**

[[:SENSe]:OBWidth:AVERage[:STATe] {{1|ON} | {0|OFF}}

[[:SENSe]:OBWidth:AVERage[:STATe]?

➤ **功能描述:**

占用带宽平均开关。

1|ON: 打开

0|OFF: 关闭

➤ **返回格式:**

查询返回占用带宽平均开关状态, 0 或 1。

➤ **举例:**

:OBWidth:AVERage ON            打开占用带宽平均

:OBWidth:AVERage?            查询返回 1

### **[[:SENSe]:OBWidth:AVERage:COUNT**

➤ **命令格式:**

[[:SENSe]:OBWidth:AVERage:COUNT <integer>

[[:SENSe]:OBWidth:AVERage:COUNT?

➤ **功能描述:**

调整占用带宽平均次数。

<integer>: 连续整数, 取值范围 1 到 999。

➤ **返回格式:**

查询返回占用带宽平均次数, 范围 1 到 999。

➤ **举例:**

:OBWidth:AVERage:COUNT 10 设置占用带宽平均次数为 10

:OBWidth:AVERage:COUNT?            查询返回 10

### **[[:SENSe]:OBWidth:AVERage:TCONTROL**

➤ **命令格式:**

[[:SENSe]:OBWidth:AVERage:TCONTROL {EXPONENTIAL|REPEAT}

[[:SENSe]:OBWidth:AVERage:TCONTROL?

➤ **功能描述:**

调整占用带宽平均模式。

EXPONENTIAL: 指数

REPEAT: 重复

➤ **返回格式:**

查询返回占用带宽平均模式, EXPONENTIAL 或 REPEAT。

➤ **举例:**

:OBWidth:AVERage:TCONTROL EXPONENTIAL            设置占用带宽指数平均

:OBWidth:AVERage:TCONTROL?            查询返回 EXPONENTIAL

### **[[:SENSe]:OBWidth:AVERage:TYPE**

➤ **命令格式:**

[[:SENSe]:OBWidth:AVERage:TYPE {VOLTAGE|POWER|LOG}

[[:SENSe]:OBWidth:AVERage:TYPE?

➤ **功能描述:**

设置占用带宽平均类型。

VOLTAGE: 电压平均

POWER: 功率平均

LOG: 对数平均

➤ **返回格式:**

查询返回占用带宽平均模式, VOLTAGE、POWER 或 LOG。

- **举例：**
  - :OBWidth:AVERage:TYPE POWER                    设置占用带宽功率平均
  - :OBWidth:AVERage:TYPE?                            查询返回 POWER

#### [ :SENSe ] :OBWidth:PERCent

- **命令格式：**
  - [ :SENSe ] :OBWidth:PERCent <ratio>
  - [ :SENSe ] :OBWidth:PERCent?
- **功能描述：**

调整占用带宽功率比率。

<ratio>: 连续实数，浮点型，取值范围 0 到 1。
- **返回格式：**

查询返回占用带宽功率比率，以科学记数法返回数据。
- **举例：**
  - :OBWidth:PERCent 0.5            设置占用带宽功率比率为 0.5
  - :OBWidth:PERCent?                查询返回 5.000000e-01

#### [ :SENSe ] :OBWidth:XDB

- **命令格式：**
  - [ :SENSe ] :OBWidth:XDB <real>
  - [ :SENSe ] :OBWidth:XDB?
- **功能描述：**

调整占用带宽 xdb 值。

<real>: 连续实数，单位 dB，取值范围-100dB 到-0.1dB
- **返回格式：**

查询返回占用带宽 xdb 值，以科学记数法返回数据，单位 dB。
- **举例：**
  - :OBWidth:XDB -10                设置占用带宽 xdb 值为-10dB
  - :OBWidth:XDB?                    查询返回-1.000000e+01

#### [ :SENSe ] :TOI:AVERage[:STATe]

- **命令格式：**
  - [ :SENSe ] :TOI:AVERage[:STATe] {{1|ON} | {0|OFF}}
  - [ :SENSe ] :TOI:AVERage[:STATe]?
- **功能描述：**

三阶交调平均开关。

1|ON: 打开

0|OFF: 关闭
- **返回格式：**

查询返回三阶交调平均开关状态，0 或 1。

- **举例：**  
:TO1:AVERage ON      打开三阶交调平均  
:TO1:AVERage?      查询返回 1

#### **[ :SENSe ] : TO1 : AVERage : COUNT**

- **命令格式：**  
[ :SENSe ] : TO1 : AVERage : COUNT <integer>  
[ :SENSe ] : TO1 : AVERage : COUNT?
- **功能描述：**  
调整三阶交调平均次数。  
<integer>: 连续整数, 取值范围 1 到 999。
- **返回格式：**  
查询返回三阶交调平均次数, 范围 1 到 999。
- **举例：**  
:TO1:AVERage:COUNT 10      设置三阶交调平均次数为 10  
:TO1:AVERage:COUNT?      查询返回 10

#### **[ :SENSe ] : TO1 : AVERage : TCONtrol**

- **命令格式：**  
[ :SENSe ] : TO1 : AVERage : TCONtrol {EXPonential | REPeat}  
[ :SENSe ] : TO1 : AVERage : TCONtrol?
- **功能描述：**  
选择三阶交调平均模式。  
EXPonential: 指数  
REPeat: 重复
- **返回格式：**  
查询返回三阶交调平均模式, EXPonential 或 REPeat。
- **举例：**  
:TO1:AVERage:TCONtrol EXPonential      设置三阶交调指数平均  
:TO1:AVERage:TCONtrol?      查询返回 EXPonential

#### **[ :SENSe ] : TO1 : AVERage : TYPE**

- **命令格式：**  
[ :SENSe ] : TO1 : AVERage : TYPE {VOLTage | POWEr | LOG}  
[ :SENSe ] : TO1 : AVERage : TYPE?
- **功能描述：**  
设置三阶交调平均类型。  
VOLTage: 电压平均  
POWEr: 功率平均  
LOG: 对数平均

- **返回格式:**  
查询返回三阶交调平均模式, VOLTage、POWER 或 LOG。
- **举例:**  
:TO1:AVERAge:TYPE POWER                    设置三阶交调功率平均  
:TO1:AVERAge:TYPE?                            查询返回 POWER

**[ :SENSe ] :ACPower :AVERAge [ :STATe ]**

- **命令格式:**  
[ :SENSe ] :ACPower :AVERAge [ :STATe ] { {1|ON} | {0|OFF} }  
[ :SENSe ] :ACPower :AVERAge [ :STATe ] ?
- **功能描述:**  
相邻信道功率平均开关。
- **返回格式:**  
查询返回相邻信道功率平均开关状态, 0 或 1。
- **举例:**  
:ACPower :AVERAge ON            打开相邻信道功率平均  
:ACPower :AVERAge?            查询返回 1

**[ :SENSe ] :ACPower :AVERAge :COUNT**

- **命令格式:**  
[ :SENSe ] :ACPower :AVERAge :COUNT <integer>  
[ :SENSe ] :ACPower :AVERAge :COUNT ?
- **功能描述:**  
调整相邻信道功率平均次数。  
<integer>: 整数, 取值范围 1 到 999。
- **返回格式:**  
查询返回相邻信道功率平均次数, 范围 1 到 999。
- **举例:**  
:ACPower :AVERAge :COUNT 10 设置相邻信道功率平均次数为 10  
:ACPower :AVERAge :COUNT?            查询返回 10

**[ :SENSe ] :ACPower :AVERAge :TCONtrol**

- **命令格式:**  
[ :SENSe ] :ACPower :AVERAge :TCONtrol {EXPonential|REPeat}  
[ :SENSe ] :ACPower :AVERAge :TCONtrol ?
- **功能描述:**  
选择相邻信道功率平均模式。  
EXPonential: 指数  
REPeat: 重复
- **返回格式:**

查询返回相邻信道功率平均模式，EXPOnential 或 REPeat。

➤ **举例：**

:ACPower:AVERage:TCONtrol EXPOnential      设置相邻信道功率指数平均  
:ACPower:AVERage:TCONtrol?                      查询返回 EXPOnential

**[ :SENSe ] :ACPower :AVERage :TYPE**

➤ **命令格式：**

[ :SENSe ] :ACPower :AVERage :TYPE {VOLTage | POWER | LOG}  
[ :SENSe ] :ACPower :AVERage :TYPE?

➤ **功能描述：**

设置相邻信道功率平均类型。

VOLTage：电压平均

POWER：功率平均

LOG：对数平均

➤ **返回格式：**

查询返回相邻信道功率平均模式，VOLTage、POWER 或 LOG。

➤ **举例：**

:ACPower:AVERage:TYPE POWER                      设置相邻信道功率功率平均  
:ACPower:AVERage:TYPE?                              查询返回 POWER

**[ :SENSe ] :ACPower :BANDwidth | BWIDth : INTegration**

➤ **命令格式：**

[ :SENSe ] :ACPower :BANDwidth | BWIDth : INTegration <freq>  
[ :SENSe ] :ACPower :BANDwidth | BWIDth : INTegration?

➤ **功能描述：**

调整相邻信道功率噪声带宽。

<freq>：频率值，单位 Hz。

➤ **返回格式：**

查询返回相邻信道功率噪声带宽，以科学记数法返回数据，单位 Hz。

➤ **举例：**

:ACPower:BANDwidth:INTegration 10MHz              设置相邻信道功率噪声带宽 10MHz  
:ACPower:BANDwidth:INTegration?                      查询返回 1.000000e+07

**[ :SENSe ] :ACPower :CARRier :LIST :BANDwidth | BWIDth [ : INTegration ]**

➤ **命令格式：**

[ :SENSe ] :ACPower :CARRier :LIST :BANDwidth | BWIDth [ : INTegration ] <freq>  
[ :SENSe ] :ACPower :CARRier :LIST :BANDwidth | BWIDth [ : INTegration ]?

➤ **功能描述：**

调整相邻信道功率积分带宽。

<freq>：连续实数，单位 Hz。

➤ **返回格式:**

查询返回相邻信道功率积分带宽，以科学记数法返回数据，单位 Hz。

➤ **举例:**

:ACPower:CARRIER:LIST:BANDwidth 10MHz      设置相邻信道功率积分带宽 10MHz  
:ACPower:CARRIER:LIST:BANDwidth?      查询返回 1.000000e+07

**[[:SENSe]:ACPower:OFFSet:LIST[:FREQUENCY]]**

➤ **命令格式:**

[[:SENSe]:ACPower:OFFSet:LIST[:FREQUENCY] <freq>  
[:SENSe]:ACPower:OFFSet:LIST[:FREQUENCY]?

➤ **功能描述:**

调整相邻信道功率频率偏移。  
<freq>: 连续实数，单位 Hz。

➤ **返回格式:**

查询返回相邻信道功率频率偏移，以科学记数法返回数据，单位 Hz。

➤ **举例:**

:ACPower:OFFSet:LIST 1MHz 设置相邻信道功率频率偏移 1MHz  
:ACPower:OFFSet:LIST?      查询返回 1.000000e+06

**[[:SENSe]:ACPower:OFFSet[:OUTer]:LIST:SIDE]**

➤ **命令格式:**

[[:SENSe]:ACPower:OFFSet[:OUTer]:LIST:SIDE {BOTH|NEGative|POSitive}  
[:SENSe]:ACPower:OFFSet[:OUTer]:LIST:SIDE?

➤ **功能描述:**

调整相邻信道功率测试偏移侧。  
BOTH: 两边  
NEGative: 负  
POSitive: 正

➤ **返回格式:**

查询返回相邻信道功率测试偏移侧，BOTH、NEGative 或 POSitive。

➤ **举例:**

:ACPower:OFFSet:LIST:SIDE BOTH      设置相邻信道功率测试偏移侧为两边  
:ACPower:OFFSet:LIST:SIDE?      查询返回 BOTH

**[[:SENSe]:SPECTrogram:AVERage[:STATe]]**

➤ **命令格式:**

[[:SENSe]:SPECTrogram:AVERage[:STATe] {{1|ON} | {0|OFF}}  
[:SENSe]:SPECTrogram:AVERage[:STATe]?

➤ **功能描述:**

频率监测平均开关。

- **返回格式:**  
查询返回频率监测平均开关状态, 0 或 1。
- **举例:**  
:SPECtrogram:AVERage ON            打开频率监测平均  
:SPECtrogram:AVERage?            查询返回 1

**[ :SENSe ] :SPECtrogram:AVERage:COUNT**

- **命令格式:**  
[ :SENSe ] :SPECtrogram:AVERage:COUNT <integer>  
[ :SENSe ] :SPECtrogram:AVERage:COUNT?
- **功能描述:**  
调整频率监测平均次数。  
<integer>: 连续整数, 取值范围 1 到 999。
- **返回格式:**  
查询返回频率监测平均次数, 范围 1 到 999。
- **举例:**  
:SPECtrogram:AVERage:COUNT 10 设置频率监测平均次数为 10  
:SPECtrogram:AVERage:COUNT?            查询返回 10

**[ :SENSe ] :SPECtrogram:AVERage:TCONtrol**

- **命令格式:**  
[ :SENSe ] :SPECtrogram:AVERage:TCONtrol {EXPonential|REPeat}  
[ :SENSe ] :SPECtrogram:AVERage:TCONtrol?
- **功能描述:**  
选择频率监测平均模式。  
EXPonential: 指数  
REPeat: 重复
- **返回格式:**  
查询返回频率监测平均模式, EXPonential 或 REPeat。
- **举例:**  
:SPECtrogram:AVERage:TCONtrol EXPonential            设置频率监测指数平均  
:SPECtrogram:AVERage:TCONtrol?            查询返回 EXPonential

**[ :SENSe ] :SPECtrogram:AVERage:TYPE**

- **命令格式:**  
[ :SENSe ] :SPECtrogram:AVERage:TYPE {VOLTage|POWER|LOG}  
[ :SENSe ] :SPECtrogram:AVERage:TYPE?
- **功能描述:**  
设置频率监测平均类型。  
VOLTage: 电压平均

POWER: 功率平均

LOG: 对数平均

➤ **返回格式:**

查询返回频率监测平均模式, VOLTage、POWER 或 LOG。

➤ **举例:**

:SPEctrogram:AVERage:TYPE POWER                    设置频率监测功率平均

:SPEctrogram:AVERage:TYPE?                            查询返回 POWER

**[[:SENSe]:CNRatio:AVERage[:STATe]**

➤ **命令格式:**

[[:SENSe]:CNRatio:AVERage[:STATe] {{1|ON} | {0|OFF}}

[[:SENSe]:CNRatio:AVERage[:STATe]?

➤ **功能描述:**

载噪比平均开关。

1|ON: 开

0|OFF: 关

➤ **返回格式:**

查询返回载噪比平均开关状态, 0 或 1。

➤ **举例:**

:CNRatio:AVERage ON            打开载噪比平均

:CNRatio:AVERage?            查询返回 1

**[[:SENSe]:CNRatio:AVERage:COUNT**

➤ **命令格式:**

[[:SENSe]:CNRatio:AVERage:COUNT <integer>

[[:SENSe]:CNRatio:AVERage:COUNT?

➤ **功能描述:**

调整载噪比平均次数。

<integer>: 连续整数, 取值范围 1 到 999。

➤ **返回格式:**

查询返回载噪比平均次数, 范围 1 到 999。

➤ **举例:**

:CNRatio:AVERage:COUNT 10 设置载噪比平均次数为 10

:CNRatio:AVERage:COUNT?            查询返回 10

**[[:SENSe]:CNRatio:AVERage:TCONTROL**

➤ **命令格式:**

[[:SENSe]:CNRatio:AVERage:TCONTROL {EXPONENTIAL|REPEAT}

[[:SENSe]:CNRatio:AVERage:TCONTROL?

➤ **功能描述:**

选择载噪比平均模式。

EXponential: 指数

REPeat: 重复

➤ **返回格式:**

查询返回载噪比平均模式, EXponential 到 REPeat。

➤ **举例:**

:CNRatio:AVERage:TCONtrol EXponential      设置载噪比指数平均

:CNRatio:AVERage:TCONtrol?      查询返回 EXponential

**[[:SENSe]:CNRatio:AVERage:TYPE**

➤ **命令格式:**

[[:SENSe]:CNRatio:AVERage:TYPE {VOLTage|POWER|LOG}

[[:SENSe]:CNRatio:AVERage:TYPE?

➤ **功能描述:**

设置载噪比平均类型。

VOLTage: 电压平均

POWER: 功率平均

LOG: 对数平均

➤ **返回格式:**

查询返回载噪比平均模式, VOLTage、POWER 或 LOG。

➤ **举例:**

:CNRatio:AVERage:TYPE POWER      设置载噪比功率平均

:CNRatio:AVERage:TYPE?      查询返回 POWER

**[[:SENSe]:CNRatio:BANDwidth:INTEgration**

➤ **命令格式:**

[[:SENSe]:CNRatio:BANDwidth:INTEgration <freq>

[[:SENSe]:CNRatio:BANDwidth:INTEgration?

➤ **功能描述:**

调整载噪比负载带宽。

<freq>: 连续实数, 单位 Hz。

➤ **返回格式:**

查询返回载噪比负载带宽, 以科学记数法返回数据, 单位 Hz。

➤ **举例:**

:CNRatio:BANDwidth:INTEgration 10MHz      设置载噪比负载带宽 10MHz

:CNRatio:BANDwidth:INTEgration?      查询返回 1.000000e+07

**[[:SENSe]:CNRatio:BANDwidth:NOISe**

➤ **命令格式:**

[[:SENSe]:CNRatio:BANDwidth:NOISe <freq>

`[ :SENSe ] : CNRatio : BANDwidth : NOISe ?`

➤ **功能描述:**

调整载噪比噪声带宽。

<freq>: 连续实数, 单位 Hz

➤ **返回格式:**

查询返回载噪比噪声带宽, 以科学记数法返回数据, 单位 Hz。

➤ **举例:**

<code>:CNRatio:BANDwidth:NOISe 10MHz</code>	设置载噪比噪声带宽 10MHz
<code>:CNRatio:BANDwidth:NOISe?</code>	查询返回 1.000000e+07

**[ :SENSe ] : CNRatio : OFFSet**

➤ **命令格式:**

`[ :SENSe ] : CNRatio : OFFSet <freq>`

`[ :SENSe ] : CNRatio : OFFSet ?`

➤ **功能描述:**

调整载噪比频率偏移。

<freq>: 连续实数, 单位 Hz。

➤ **返回格式:**

查询返回载噪比频率偏移, 以科学记数法返回数据, 单位 Hz。

➤ **举例:**

<code>:CNRatio:OFFSet 10MHz</code>	设置载噪比频率偏移 10MHz
<code>:CNRatio:OFFSet?</code>	查询返回 1.000000e+07

**[ :SENSe ] : HARMonics : AVERage [ :STATe ]**

➤ **命令格式:**

`[ :SENSe ] : HARMonics : AVERage [ :STATe ] { {1|ON} | {0|OFF} }`

`[ :SENSe ] : HARMonics : AVERage [ :STATe ] ?`

➤ **功能描述:**

谐波平均开关。

1|ON: 打开

0|OFF: 关闭

➤ **返回格式:**

查询返回谐波平均开关状态, 0 或 1。

➤ **举例:**

<code>:HARMonics:AVERage ON</code>	打开谐波平均
<code>:HARMonics:AVERage?</code>	查询返回 1

**[ :SENSe ] : HARMonics : AVERage : COUNT**

➤ **命令格式:**

`[ :SENSe ] : HARMonics : AVERage : COUNT <integer>`

[ :SENSe ] :HARMonics:AVERAge:COUNT?

➤ **功能描述:**

调整谐波平均次数。

<integer>: 连续整数, 取值范围 1 到 999。

➤ **返回格式:**

查询返回谐波平均次数, 范围 1 到 999。

➤ **举例:**

:HARMonics:AVERAge:COUNT 10            设置谐波平均次数为 10

:HARMonics:AVERAge:COUNT?            查询返回 10

**[ :SENSe ] :HARMonics:AVERAge:TCONtrol**

➤ **命令格式:**

[ :SENSe ] :HARMonics:AVERAge:TCONtrol {EXPonential|REPeat}

[ :SENSe ] :HARMonics:AVERAge:TCONtrol?

➤ **功能描述:**

选择谐波平均模式。

EXPonential: 指数

REPeat: 重复

➤ **返回格式:**

查询返回谐波平均模式, EXPonential 或 REPeat。

➤ **举例:**

:HARMonics:AVERAge:TCONtrol EXPonential            设置谐波指数平均

:HARMonics:AVERAge:TCONtrol?            查询返回 EXPonential

**[ :SENSe ] :HARMonics:AVERAge:TYPE**

➤ **命令格式:**

[ :SENSe ] :HARMonics:AVERAge:TYPE {VOLTage|POWER|LOG}

[ :SENSe ] :HARMonics:AVERAge:TYPE?

➤ **功能描述:**

设置谐波平均类型。

VOLTage: 电压平均

POWER: 功率平均

LOG: 对数平均

➤ **返回格式:**

查询返回谐波平均模式, VOLTage、POWER 或 LOG。

➤ **举例:**

:HARMonics:AVERAge:TYPE POWER            设置谐波功率平均

:HARMonics:AVERAge:TYPE?            查询返回 POWER

## **[ :SENSe ] :HARMonics:FREQuency:FUNDamental**

### ➤ **命令格式:**

[ :SENSe ] :HARMonics:FREQuency:FUNDamental <freq>

[ :SENSe ] :HARMonics:FREQuency:FUNDamental?

### ➤ **功能描述:**

调整谐波基波频率。

<freq>: 连续实数, 单位 Hz。

### ➤ **返回格式:**

查询返回谐波基波频率, 以科学记数法返回数据, 单位 Hz。

### ➤ **举例:**

:HARMonics:FREQuency:FUNDamental 100MHz                    设置谐波基波频率 100MHz

:HARMonics:FREQuency:FUNDamental?                    查询返回 1.000000e+08

## **[ :SENSe ] :HARMonics:NUMBer**

### ➤ **命令格式:**

[ :SENSe ] :HARMonics:NUMBer <integer>

[ :SENSe ] :HARMonics:NUMBer?

### ➤ **功能描述:**

调整谐波次数。

<integer>: 谐波次数值, 连续整数, 取值范围 2 到 10。

### ➤ **返回格式:**

查询返回谐波次数, 范围 2 到 10。

### ➤ **举例:**

:HARMonics:NUMBer 4                    设置谐波次数为 4

:HARMonics:NUMBer?                    查询返回 4

## **[ :SENSe ] :HARMonics:SWEeptime**

### ➤ **命令格式:**

[ :SENSe ] :HARMonics:SWEeptime <time>

[ :SENSe ] :HARMonics:SWEeptime?

### ➤ **功能描述:**

设置谐波驻留时间。

<time>: 驻留时间值, 连续整数, 取值范围 1ms 到 100s, 默认单位 s。

### ➤ **返回格式:**

查询返回谐波驻留时间, 以科学记数法返回数据, 单位 s。

### ➤ **举例:**

:HARMonics:SWEeptime 0.1                    设置谐波驻留时间为 100ms

:HARMonics:SWEeptime?                    查询返回 1.000000e-01

## TRACe 命令

### :TRACe:SElect

- **命令格式:**  
:TRACe:SElect <integer>  
:TRACe:SElect?
- **功能描述:**  
从所有迹线序列中选择一条作为当前迹线。  
<integer>: 迹线序列号, 整数, 取值范围 1 到最大迹线数。  
UTS1015B 的最大迹线数为 4, UTS3036B 的最大迹线数为 6。
- **返回格式:**  
查询返回当前迹线序列号。
- **举例:**  
:TRACe:SElect 1                    设置迹线 1 为当前迹线  
:TRACe:SElect?                    查询返回 1

### :TRACe<n>:TYPE

- **命令格式:**  
:TRACe<n>:TYPE <WRITe|AVERAge|MAXHold|MINHold>  
:TRACe<n>:TYPE?
- **功能描述:**  
选择指定迹线的迹线类型。  
<n>: 迹线序列号, 整数, 取值范围 1 到最大迹线数。  
UTS1015B 的最大迹线数为 4, UTS3036B 的最大迹线数为 6。  
WRITe: 刷新  
AVERAge: 迹线平均  
MAXHold: 最大保持  
MINHold: 最小保持
- **返回格式:**  
查询返回指定迹线的迹线类型, WRITe、AVERAge、MAXHold 或 MINHold。
- **举例:**  
:TRACe1:TYPE AVERAge            设置迹线 1 为迹线平均  
:TRACe1:TYPE?                    查询返回 AVERAge

### :TRACe:TYPE

- **命令格式:**  
:TRACe:TYPE <WRITe|AVERAge|MAXHold|MINHold>  
:TRACe:TYPE?
- **功能描述:**  
当前迹线设置迹线类型, 用于高级测量。

WRITe: 刷新

AVERAge: 迹线平均

MAXHold: 最大保持

MINHold: 最小保持

➤ **返回格式:**

查询返回当前迹线的迹线类型, WRITe、AVERAge、MAXHold 或 MINHold。

➤ **举例:**

:TRACe:TYPE AVERAge            设置当前迹线为迹线平均

:TRACe:TYPE?                    查询返回 AVERAge

**:TRACe<n>:UPDate:STATe**

➤ **命令格式:**

:TRACe<n>:UPDate:STATe {{1|ON} | {0|OFF}}

:TRACe<n>:UPDate:STATe?

➤ **功能描述:**

指定迹线刷新开关, ON 之后迹线持续刷新。

<n>: 迹线序列号, 整数, 取值范围 1 到最大迹线数。

UTS1015B 的最大迹线数为 4, UTS3036B 的最大迹线数为 6。

➤ **返回格式:**

查询返回指定迹线更新状态, 0 或 1。

➤ **举例:**

:TRACe1:UPDate:STATe ON        迹线 1 打开迹线更新

:TRACe1:UPDate:STATe?        查询返回 1

**:TRACe<n>:DISPlay[:STATe]**

➤ **命令格式:**

:TRACe<n>:DISPlay[:STATe] {{1|ON} | {0|OFF}}

:TRACe<n>:DISPlay[:STATe]?

➤ **功能描述:**

指定迹线显示开关。

<n>: 迹线序列号, 整数, 取值范围 1 到最大迹线数。

UTS1015B 的最大迹线数为 4, UTS3036B 的最大迹线数为 6。

➤ **返回格式:**

查询返回指定迹线显示状态, 0 或 1。

➤ **举例:**

:TRACe1:DISPlay ON            显示迹线 1

:TRACe1:DISPlay?            查询返回 1

**:TRACe:MATH:PEAK:TABLE:STATe**

➤ **命令格式:**

:TRACe:MATH:PEAK:TABLE:STATe {{1|ON} | {0|OFF}}

:TRACe:MATH:PEAK:TABLE:STATe?

➤ **功能描述:**

峰值表显示开关。

1|ON: 显示峰值表

0|OFF: 不显示峰值表

➤ **返回格式:**

查询返回峰值表显示状态, 0 或 1。

➤ **举例:**

:TRACe:MATH:PEAK:TABLE:STATe ON 显示峰值表

:TRACe:MATH:PEAK:TABLE:STATe? 查询返回 1

## TRIGger 命令

:TRIGger[:SEQuence]:SOURce

➤ **命令格式:**

:TRIGger[:SEQuence]:SOURce {IMMEDIATE|EXTernal|VIDeo}

:TRIGger[:SEQuence]:SOURce?

➤ **功能描述:**

选择触发类型。

IMMEDIATE: 自由触发

EXTernal: 外触发

VIDeo: 视频触发

➤ **返回格式:**

查询返回触发类型, IMMEDIATE、EXTernal 或 VIDeo。

➤ **举例:**

:TRIGger:SOURce IMMEDIATE 选择自由触发

:TRIGger:SOURce? 查询返回 IMMEDIATE

:TRIGger[:SEQuence]:EXTernal:SLOPe

➤ **命令格式:**

:TRIGger[:SEQuence]:EXTernal:SLOPe {POSitive|NEGative}

:TRIGger[:SEQuence]:EXTernal:SLOPe?

➤ **功能描述:**

选择外部触发的触发边沿。

POSitive: 上升沿

NEGative: 下降沿

➤ **返回格式:**

查询返回外部触发的触发边沿, POSitive 或 NEGative。

➤ **举例:**

:TRIGger:EXTeRnal:SLOPe POSitive 选择外部触发的触发边沿为上升沿  
:TRIGger:EXTeRnal:SLOPe? 查询返回 POSitive

#### :TRIGger[:SEqueNce]:EXTeRnal:DELAy

➤ **命令格式:**

:TRIGger[:SEqueNce]:EXTeRnal:DELAy <time>  
:TRIGger[:SEqueNce]:EXTeRnal:DELAy?

➤ **功能描述:**

设置外部触发的触发延时。

<time>: 连续正数, 默认单位 s, 范围 1000ps 到 500ms。

➤ **返回格式:**

查询返回外部触发的触发延时, 以科学记数法返回数据, 单位为 s。

➤ **举例:**

:TRIGger:EXTeRnal:DELAy 0.01 设置外部触发的触发延时为 10ms  
:TRIGger:EXTeRnal:DELAy? 查询返回 1.000000e-02

#### :TRIGger[:SEqueNce]:VIDeo:LEVel

➤ **命令格式:**

:TRIGger[:SEqueNce]:VIDeo:LEVel <ampI>  
:TRIGger[:SEqueNce]:VIDeo:LEVel?

➤ **功能描述:**

设置视频触发的触发电平。

<ampI>: 连续实数, 默认单位 dBm

➤ **返回格式:**

查询返回外部触发的触发边沿, 以科学记数法返回数据, 单位为 dBm。

➤ **举例:**

:TRIGger:VIDeo:LEVel -50 设置视频触发的触发电平为-50dBm  
:TRIGger:VIDeo:LEVel? 查询返回-5.000000e+01

## SOURce 命令

#### :SOURce[:EXTeRnal]:POWeR[:LEVel][:IMMediate][:AMPLitude]

➤ **命令格式:**

:SOURce[:EXTeRnal]:POWeR[:LEVel][:IMMediate][:AMPLitude] <ampI>  
:SOURce[:EXTeRnal]:POWeR[:LEVel][:IMMediate][:AMPLitude]?

➤ **功能描述:**

调整跟踪源幅值。

<ampI>: 连续实数值, 默认单位 dBm。

➤ **返回格式:**

查询返回跟踪源幅值，以科学记数法返回数据，单位为 dBm。

➤ **举例：**

:SOURce:POWer -40                      设置跟踪源幅值为-40dBm  
:SOURce:POWer?                         查询返回-4.000000e+01

**:SOURce:CORRection:OFFSet**

➤ **命令格式：**

:SOURce:CORRection:OFFSet <real>  
:SOURce:CORRection:OFFSet?

➤ **功能描述：**

调整跟踪源幅值偏移。

<real>: 连续实数值，默认单位 dB，取值范围-200dB 到 200dB。

➤ **返回格式：**

查询返回跟踪源幅值偏移，以科学记数法返回数据，单位为 dB。

➤ **举例：**

:SOURce:CORRection:OFFSet 10        设置跟踪源幅值偏移为 10dB  
:SOURce:CORRection:OFFSet?         查询返回 1.000000e+01

## UNIT 命令

**:UNIT:POWer**

➤ **命令格式：**

:UNIT:POWer {DBM|DBMV|DBUV|V|W}  
:UNIT:POWer?

➤ **功能描述：**

选择 Y 轴刻度单位。

➤ **返回格式：**

查询返回 Y 轴刻度单位，DBM、DBMV、DBUV、V 或 W。

➤ **举例：**

:UNIT:POWer DBM                        选择 DBM 为 Y 轴刻度单位  
:UNIT:POWer?                            查询返回 DBM

## CALCulate 命令

### :CALCulate:FSCan:MARKer:SElect

➤ **命令格式:**

```
:CALCulate:FSCan:MARKer:SElect <integer>
:CALCulate:FSCan:MARKer:SElect?
```

➤ **功能描述:**

从标记序列中选择一个作为当前标记。

<integer>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

查询返回当前标记序列号, 范围 1 到 6 的整数。

➤ **举例:**

```
:CALCulate:FSCan:MARKer:SElect 1           选择标记 1 为当前标记
:CALCulate:FSCan:MARKer:SElect?           查询返回 1
```

### :CALCulate:FSCan:MARKer<n>:MODE

➤ **命令格式:**

```
:CALCulate:FSCan:MARKer<n>:MODE {OFF|POSition|DELTA|FIXed}
:CALCulate:FSCan:MARKer<n>:MODE?
```

➤ **功能描述:**

选择指定标记的模式。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

OFF: 关闭标记

POSition: 常规模式

DELTA: 差值模式

FIXed: 固定模式

➤ **返回格式:**

查询返回指定标记的模式, OFF、POSition、DELTA 或 FIXed。

➤ **举例:**

```
:CALCulate:FSCan:MARKer1:MODE POSition     设置标记 1 为常规模式
:CALCulate:FSCan:MARKer1:MODE?             查询返回 POSition
```

### :CALCulate:FSCan:MARKer<n>:TRACe

➤ **命令格式:**

```
:CALCulate:FSCan:MARKer<n>:TRACe <integer>
:CALCulate:FSCan:MARKer<n>:TRACe?
```

➤ **功能描述:**

选择指定标记对应的迹线。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

<integer>: 迹线序列号, 连续整数, 取值范围 1 到 3。

➤ **返回格式:**

查询返回指定标记对应迹线的序列号。

➤ **举例:**

:CALCulate:FSCan:MARKer1:TRACe 1	设置标记 1 对应迹线 1
:CALCulate:FSCan:MARKer1:TRACe?	查询返回 1

**:CALCulate:MARKer<n>:X**

➤ **命令格式:**

:CALCulate:MARKer<n>:X <freq>

:CALCulate:MARKer<n>:X?

➤ **功能描述:**

调整指定标记的 X 轴坐标值。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

<freq>: 频率值, 默认单位为 Hz。

➤ **返回格式:**

查询返回指定标记的 X 轴坐标值, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:CALCulate:MARKer1:X 1GHz	设置标记 1 的 X 轴坐标值为 1GHz
:CALCulate:MARKer1:X?	查询返回 1.000000e+09

**:CALCulate:MARKer<n>:Y**

➤ **命令格式:**

:CALCulate:MARKer<n>:Y <ampt>

:CALCulate:MARKer<n>:Y?

➤ **功能描述:**

调整指定标记的幅度值。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

<ampt>: 标记幅度值, 默认单位为 dBm。

➤ **返回格式:**

查询返回指定标记的幅度值, 以科学计数法返回数据, 单位为 dBm。

➤ **举例:**

:CALCulate:MARKer1:Y -50	设置标记 1 的幅度值为-50dBm
:CALCulate:MARKer1:Y?	查询返回-5.000000e+01

**:CALCulate:FSCan:MARKer<n>:REFerence**

➤ **命令格式:**

:CALCulate:FSCan:MARKer<n>:REFerence <integer>

:CALCulate:FSCan:MARKer<n>:REFerence?

➤ **功能描述:**

选择指定标记的参考标记，参考标记不能为自身。

<n>: 标记序列号，连续整数，取值范围 1 到 6。

<integer>: 除自身外的其他标记序列号，取值范围 1 到 6。

➤ **返回格式:**

查询返回指定标记的参考标记。

➤ **举例:**

:CALCulate:FSCan:MARKer1:REFerence 2	设置标记 2 为标记 1 的参考标记
:CALCulate:FSCan:MARKer1:REFerence?	查询返回 2

:CALCulate:FSCan:MARKer:AOFF

➤ **命令格式:**

:CALCulate:FSCan:MARKer:AOFF

➤ **功能描述:**

关闭所有标记。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:MARKer:AOFF                      关闭所有标记

:CALCulate:FSCan:MARKer<n>[:SET]:SLIST

➤ **命令格式:**

:CALCulate:FSCan:MARKer<n>[:SET]:SLIST

➤ **功能描述:**

添加一个信号到信号表，频率为指定标记频率。

<n>: 标记序列号，连续整数，取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:MARKer1:SET:SLIST                      添加信号到信号表，频率为指定标记频率

:CALCulate:FSCan:MAMarker[:SET]:SLIST

➤ **命令格式:**

:CALCulate:FSCan:MAMarker[:SET]:SLIST

➤ **功能描述:**

添加标记测量结果到信号表。如果未测量标记，则不会加入到信号表。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:MArker:SET:SLIST

添加标记测量结果到信号表

:CALCulate:FSCan:MArker<n>[:SET]:METer

➤ **命令格式:**

:CALCulate:FSCan:MArker<n>[:SET]:METer

➤ **功能描述:**

指定标记频率设置为计量表频率。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:MArker1:SET:METer

标记 1 频率设置为计量表频率

:CALCulate:FSCan:MArker<n>:TO:METer

➤ **命令格式:**

:CALCulate:FSCan:MArker<n>:TO:METer

➤ **功能描述:**

计量表频率设置为指定标记频率。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:MArker1:TO:METer

计量表频率设置为标记 1 频率

:CALCulate:FSCan:SLIST:REPLace:METer

➤ **命令格式:**

:CALCulate:FSCan:SLIST:REPLace:METer <integer>

➤ **功能描述:**

指定信号的频率替换为计量表频率。

<integer>: 信号序列号, 连续整数, 取值范围 1 到 1000。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLIST:REPLace:METer 1

信号 1 的频率替换为计量表频率

:CALCulate:FSCan:SLIST:APPend:METer

➤ **命令格式:**

:CALCulate:FSCan:SLIST:APPend:METer

➤ **功能描述:**

添加一个信号到信号表，频率为计量表频率。

➤ **返回格式：**

无返回值。

➤ **举例：**

:CALCulate:FSCan:SLIST:APPend:METer 添加一个信号到信号表，频率为计量表频率

**:CALCulate:FSCan:MARKer<n>:MAXimum[:MAX]**

➤ **命令格式：**

:CALCulate:FSCan:MARKer<n>:MAXimum[:MAX]

➤ **功能描述：**

对指定标记执行峰值搜索。

<n>：标记序列号，连续整数，取值范围 1 到 6

➤ **返回格式：**

无返回值。

➤ **举例：**

:CALCulate:FSCan:MARKer1:MAXimum 标记 1 执行峰值搜索

**:CALCulate:FSCan:MARKer<n>:MAXimum:NEXT**

➤ **命令格式：**

:CALCulate:FSCan:MARKer<n>:MAXimum:NEXT

➤ **功能描述：**

对指定标记执行下一峰值搜索。

<n>：标记序列号，连续整数，取值范围 1 到 6。

➤ **返回格式：**

无返回值。

➤ **举例：**

:CALCulate:FSCan:MARKer1:MAXimum:NEXT 标记 1 执行下一峰值搜索

**:CALCulate:FSCan:MARKer<n>:MAXimum:LEFT**

➤ **命令格式：**

:CALCulate:FSCan:MARKer<n>:MAXimum:LEFT

➤ **功能描述：**

对指定标记执行左侧下一峰值搜索。

<n>：标记序列号，连续整数，取值范围 1 到 6。

➤ **返回格式：**

无返回值。

➤ **举例：**

:CALCulate:FSCan:MARKer1:MAXimum:LEFT 标记 1 执行左侧下一峰值搜索

### **:CALCulate:FSCan:MARKer<n>:MAXimum:RIGHT**

➤ **命令格式:**

:CALCulate:FSCan:MARKer<n>:MAXimum:RIGHT

➤ **功能描述:**

对指定标记执行右侧下一峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:MARKer1:MAXimum:RIGHT      标记 1 执行右侧下一峰值搜索

### **:CALCulate:FSCan:MARKer<n>:MINimum**

➤ **命令格式:**

:CALCulate:FSCan:MARKer<n>:MINimum

➤ **功能描述:**

对指定标记执行最小峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:MARKer1:MINimum      标记 1 执行最小峰值搜索

### **:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE[:STATe]**

➤ **命令格式:**

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE[:STATe]?

➤ **功能描述:**

峰值阈值线显示开关。

1|ON: 显示

0|OFF: 不显示

➤ **返回格式:**

查询返回阈值线显示状态。

➤ **举例:**

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE ON      显示峰值阈值线

:CALCulate:FSCan:MARKer:PEAK:THReshold:LINE?      查询返回 1

### **:CALCulate:FSCan:MARKer:PEAK:THReshold**

➤ **命令格式:**

:CALCulate:FSCan:MARKer:PEAK:THReshold <amp;I>

:CALCulate:FSCan:MARKer:PEAK:THReshold?

➤ **功能描述:**

调整峰值门限。

<ampl>: 连续实数, 默认单位为 dBm, 范围 30dBm 到-200dBm。

➤ **返回格式:**

查询返回峰值门限, 以科学计数法返回数据, 单位为 dBm。

➤ **举例:**

:CALCulate:FSCan:MARKer:PEAK:THReshold -20	设置峰值门限-20dBm
:CALCulate:FSCan:MARKer:PEAK:THReshold?	查询返回-2.000000e+01

**:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe**

➤ **命令格式:**

:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe {{1|ON} | {0|OFF}}  
:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe?

➤ **功能描述:**

手动/自动峰值门限切换。

1|ON: 自动峰值门限。

0|OFF: 手动峰值门限。

➤ **返回格式:**

查询返回自动峰值门限状态, 0 或 1。

➤ **举例:**

:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe ON	选择自动峰值门限
:CALCulate:FSCan:MARKer:PEAK:THReshold:STATe?	查询返回 1

**:CALCulate:FSCan:MARKer:PEAK:EXCursion**

➤ **命令格式:**

:CALCulate:FSCan:MARKer:PEAK:EXCursion <ampl>  
:CALCulate:FSCan:MARKer:PEAK:EXCursion?

➤ **功能描述:**

调整峰值偏移。

<ampl>: 连续实数, 默认单位为 dB, 范围 0dB 到 100dB。

➤ **返回格式:**

查询返回峰值偏移, 以科学计数法返回数据, 单位为 dB。

➤ **举例:**

:CALCulate:FSCan:MARKer:PEAK:EXCursion 20	设置峰值偏移 20dB
:CALCulate:FSCan:MARKer:PEAK:EXCursion?	查询返回 2.000000e+01

**:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe**

➤ **命令格式:**

:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe {{1|ON} | {0|OFF}}

:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe?

➤ **功能描述:**

手动/自动峰值偏移切换。

1|ON: 自动峰值偏移。

0|OFF: 手动峰值偏移。

➤ **返回格式:**

查询返回自动峰值偏移状态, 0 或 1。

➤ **举例:**

:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe ON

选择自动峰值偏移

:CALCulate:FSCan:MARKer:PEAK:EXCursion:STATe?

查询返回 1

**:CALCulate:FSCan:SLISt:DELeTe:SIGNaI**

➤ **命令格式:**

:CALCulate:FSCan:SLISt:DELeTe:SIGNaI <integer>

➤ **功能描述:**

从信号表中删除指定信号。

<integer>: 信号序列号, 连续整数, 取值范围 1 到 1001。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLISt:DELeTe:SIGNaI 1

从信号表中删除第一个信号

**:CALCulate:FSCan:SLISt:MARK:ALL**

➤ **命令格式:**

:CALCulate:FSCan:SLISt:MARK:ALL

➤ **功能描述:**

标记信号表中所有信号。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLISt:MARK:ALL

标记信号表中所有信号

**:CALCulate:FSCan:SLISt:MARK:CLEAr:ALL**

➤ **命令格式:**

:CALCulate:FSCan:SLISt:MARK:CLEAr:ALL

➤ **功能描述:**

清除信号表中标记。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLIST:MARK:CLEar:ALL 清除信号表中标记

#### :CALCulate:FSCan:SLIST:MARK:CLEar:SIGNal

➤ **命令格式:**

:CALCulate:FSCan:SLIST:MARK:CLEar:SIGNal <integer>

➤ **功能描述:**

清除信号表中选中信号标记。

<integer>: 信号序列号, 连续整数, 取值 1-1001。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLIST:MARK:CLEar:SIGNal 1 清除信号表中信号 1 标记

#### :CALCulate:FSCan:SLIST:MARK:SIGNal

➤ **命令格式:**

:CALCulate:FSCan:SLIST:MARK:SIGNal <integer>

➤ **功能描述:**

标记信号表中选中信号。

<integer>: 信号序列号, 连续整数, 取值 1-1001。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLIST:MARK:SIGNal 1 标记信号表中信号 1

#### :CALCulate:FSCan:SLIST:SET:METer

➤ **命令格式:**

:CALCulate:FSCan:SLIST:SET:METer <integer>

➤ **功能描述:**

设置计量表频率为指定信号的频率。

<integer>: 信号序列号, 连续整数, 取值 1-1001。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLIST:SET:METer 1 设置计量表频率为信号 1 的频率

#### :CALCulate:FSCan:SLIST:SORT:TYPE

➤ **命令格式:**

:CALCulate:FSCan:SLIST:SORT:TYPE

{FREQuency|DAMPliTude1|DAMPliTude2|DAMPliTude3|DLDeLta1|DLDeLta2|DLDeLta3}

:CALCulate:FSCan:SLIST:SORT:TYPE?

➤ **功能描述:**

设置信号表排序类型。

FREQUENCY: 频率值。

DAMPLitude1: 检波器 1 测量值。

DAMPLitude2: 检波器 2 测量值。

DAMPLitude3: 检波器 3 测量值。

DLDelta1: 检波器 1 测量值与对应限值差值。

DLDelta2: 检波器 3 测量值与对应限值差值。

DLDelta3: 检波器 3 测量值与对应限值差值。

➤ **返回格式:**

查询返回信号表排序类型。

➤ **举例:**

:CALCulate:FSCan:SLIST:SORT:TYPE FREQUENCY

信号表以频率值大小排序

:CALCulate:FSCan:SLIST:SORT:TYPE?

查询换回 FREQUENCY

**:CALCulate:FSCan:SLIST:SORT:ORDER**

➤ **命令格式:**

:CALCulate:FSCan:SLIST:SORT:ORDER {ASCending|DESCending}

:CALCulate:FSCan:SLIST:SORT:ORDER?

➤ **功能描述:**

设置信号表排序顺序，与排序类型共同信号排序。

ASCending: 升序。

DESCending: 降序。

➤ **返回格式:**

查询返回信号表排序顺序。

➤ **举例:**

:CALCulate:FSCan:SLIST:SORT:ORDER ASCending

信号表以升序排序

:CALCulate:FSCan:SLIST:SORT:ORDER?

查询换回 ASCending

**:CALCulate:FSCan:SLIST:ZOOM**

➤ **命令格式:**

:CALCulate:FSCan:SLIST:ZOOM <integer>

➤ **功能描述:**

信号表中指定信号放大，以信号为中心，减小带宽。

<integer>: 信号序列号，连续整数，取值 1-1001。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:FSCan:SLIST:ZOOM

放大当前信号显示

## :CALCulate:FSCan:SLISt:ZOOM:OUT

### ➤ 命令格式:

:CALCulate:FSCan:SLISt:ZOOM:OUT <integer>

### ➤ 功能描述:

信号表中指定信号缩小，以信号为中心，增加带宽。相当于信号放大的回退操作。

<integer>: 信号序列号，连续整数，取值 1-1001。

### ➤ 返回格式:

无返回值。

### ➤ 举例:

:CALCulate:FSCan:SLISt:ZOOM:OUT                      缩小当前信号显示

## :CALCulate:METer<n>:LIMit[:DATA]

### ➤ 命令格式:

:CALCulate:METer<n>:LIMit[:DATA] <ampl>

:CALCulate:METer<n>:LIMit[:DATA]?

### ➤ 功能描述:

设置指定计量表限值。

<ampl>: 限值幅度，连续实数，默认单位 dBm。

### ➤ 返回格式:

查询返回指定计量表限值幅值，以科学计数法返回数据，单位 dBm。

### ➤ 举例:

:CALCulate:METer1:LIMit:DATA -50dBm                      设置计量表 1 的限值幅值为-50dBm

:CALCulate:METer1:LIMit:DATA?                              查询返回-5.000000e+01

## :CALCulate:METer<n>:LIMit:STATe

### ➤ 命令格式:

:CALCulate:METer<n>:LIMit:STATe {{1|ON} | {0|OFF}}

:CALCulate:METer<n>:LIMit:STATe?

### ➤ 功能描述:

指定计量表限值开关。

### ➤ 返回格式:

查询返回指定计量表限值开关状态，0 或 1。

### ➤ 举例:

:CALCulate:METer1:LIMit:STATe ON                          打开计量表 1 限值

:CALCulate:METer1:LIMit:STATe?                              查询返回 1

## :CALCulate:FSCan:MARKer:COUPlE:METer

### ➤ 命令格式:

:CALCulate:FSCan:MARKer:COUPlE:METer {{1|ON} | {0|OFF}}

:CALCulate:FSCan:MARKer:COUPlE:METer?

➤ **功能描述:**

计量表耦合到标记开关。

➤ **返回格式:**

查询返回计量表耦合到标记开关状态, 0 或 1。

➤ **举例:**

:CALCulate:FSCan:MARKer:COUPlE:METer ON	打开计量表耦合到标记
:CALCulate:FSCan:MARKer:COUPlE:METer?	查询返回 1

### :CALCulate:LLINe:SElect

➤ **命令格式:**

:CALCulate:LLINe:SElect <integer>

:CALCulate:LLINe:SElect?

➤ **功能描述:**

从限值序列中选择一个作为当前限值。

<integer>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

➤ **返回格式:**

查询返回当前限值序列号。

➤ **举例:**

:CALCulate:LLINe:SElect 1	选择限值 1 为当前限值
:CALCulate:LLINe:SElect?	查询返回 1

### :CALCulate:LLINe<n>:DISPlay

➤ **命令格式:**

:CALCulate:LLINe<n>:DISPlay {{1|ON} | {0|OFF}}

:CALCulate:LLINe<n>:DISPlay?

➤ **功能描述:**

指定限值显示开关。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

➤ **返回格式:**

查询返回当前限值序列号, 0 或 1。

➤ **举例:**

:CALCulate:LLINe1:DISPlay ON	显示限值 1
:CALCulate:LLINe1:DISPlay?	查询返回 1

### :CALCulate:LLINe<n>:TRACe

➤ **命令格式:**

:CALCulate:LLINe<n>:TRACe <integer>

:CALCulate:LLINe<n>:TRACe?

➤ **功能描述:**

选择指定限值测试迹线。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

<integer>: 迹线序列号, 连续整数, 取值范围 1 到最大迹线数。

UTS1015B 的最大迹线数为 4, UTS3036B 和 UTS3084B 的最大迹线数为 6。

➤ **返回格式:**

查询返回当前限值序列号。

➤ **举例:**

:CALCulate:LLINe1:TRACe 1            选择迹线 1 为限值 1 测试迹线

:CALCulate:LLINe1:TRACe?            查询返回 1

**:CALCulate:LLINe<n>:MARGin**

➤ **命令格式:**

:CALCulate:LLINe<n>:MARGin <real>

:CALCulate:LLINe<n>:MARGin?

➤ **功能描述:**

调整指定限值裕量。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

<real>: 连续实数, 默认单位为 dB, 范围-40dB 到 40dB。

➤ **返回格式:**

查询返回指定限值裕量, 以科学计数法返回数据, 单位为 dB。

➤ **举例:**

:CALCulate:LLINe1:MARGin 20            设置限值 1 裕量 20dB

:CALCulate:LLINe1:MARGin?            查询返回 2.000000e+01

**:CALCulate:LLINe<n>:MARGin:STATe**

➤ **命令格式:**

:CALCulate:LLINe<n>:MARGin:STATe {[1|ON] | [0|OFF]}

:CALCulate:LLINe<n>:MARGin:STATe?

➤ **功能描述:**

指定限值裕量开关。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

➤ **返回格式:**

查询返回指定限值裕量开关状态, 0 或 1。

➤ **举例:**

:CALCulate:LLINe1:MARGin:STATe ON 限值 1 打开限值裕量

:CALCulate:LLINe1:MARGin:STATe?            查询返回 1

## :CALCulate:LLINe<n>:DATA

### ➤ 命令格式:

:CALCulate:LLINe<n>:DATA {<freq>, <ampl>, <connect>, <freq>, <ampl>, <connect>, ... }

:CALCulate:LLINe<n>:DATA?

### ➤ 功能描述:

编辑指定限值数据，以 {频率，幅值，连接属性} 为基本单元编辑。

<n>: 限值序列号，连续整数，取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4，UTS3036B 和 UTS3084B 的最大限值数为 6。

<freq>: 连续实数，单位 Hz

<ampl>: 连续实数，单位默认 dBm

<connect>: 取值为 0 或 1。当取值为 1 时，表示当前点与前一个点相连接以确定限制线；取值为 0 时，表示当前点与前一个点不连接(断开)。第一个点的<connect>取值为 0。

### ➤ 返回格式:

查询返回指定限值数据，{频率，幅值，连接属性} 为基本单元换行，频率单位为 Hz，幅值单位为 dBm，连接属性为 0 或 1。

### ➤ 举例:

:CALCulate:LLINe1:DATA 10000000, -50, 0, 100000000, -60, 1      设置限值 1 数据

:CALCulate:LLINe1:DATA?      查询返回限值 1 数据

## :CALCulate:LLINe<n>:COPY

### ➤ 命令格式:

:CALCulate:LLINe<n>:COPY {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6}

### ➤ 功能描述:

拷贝指定限值数据到指定限值，不能自己拷贝自己。

<n>: 限值序列号，连续整数，取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4，UTS3036B 和 UTS3084B 的最大限值数为 6。

LLINE1 - LLINE6: 对应限值 1 到限值 6。

UTS1015B 为 LLINE1 - LLINE4，UTS3036B 和 UTS3084B 为 LLINE1 - LLINE6。

### ➤ 返回格式:

无返回。

### ➤ 举例:

:CALCulate:LLINe2:COPY LLINE1 拷贝限值 1 的数据到限值 2

## :CALCulate:LLINe<n>:BUILd

### ➤ 命令格式:

:CALCulate:LLINe<n>:BUILd {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6}

### ➤ 功能描述:

指定限值数据从迹线构筑。

<n>: 限值序列号，连续整数，取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4，UTS3036B 和 UTS3084B 的最大限值数为 6。

TRACE1 - TRACE6: 对应迹线 1 到迹线 6。

UTS1015B 为 TRACE1 - TRACE4, UTS3036B 和 UTS3084B 为 TRACE1- TRACE6。

➤ **返回格式:**

无返回。

➤ **举例:**

:CALCulate:LLINe1:BUILd TRACE1      限值 1 数据从迹线 1 构筑

**:CALCulate:LLINe<n>:OFFSet:X**

➤ **命令格式:**

:CALCulate:LLINe<n>:OFFSet:X <freq>

:CALCulate:LLINe<n>:OFFSet:X?

➤ **功能描述:**

设置指定限值 X 轴偏移。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

<freq>: 连续实数, 默认单位为 Hz。

➤ **返回格式:**

查询返回指定限值 X 轴偏移, 以科学计数法返回数据, 单位 Hz。

➤ **举例:**

:CALCulate:LLINe1:OFFSet:X 10000000      设置限值 1 X 轴偏移 10MHz

:CALCulate:LLINe1:OFFSet:X?              查询返回 1.000000e+07

**:CALCulate:LLINe<n>:OFFSet:Y**

➤ **命令格式:**

:CALCulate:LLINe<n>:OFFSet:Y <real>

:CALCulate:LLINe<n>:OFFSet:Y?

➤ **功能描述:**

设置指定限值 Y 轴偏移。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

<real>: 连续实数, 默认单位为 dB。

➤ **返回格式:**

查询返回指定限值 Y 轴偏移, 以科学计数法返回数据, 单位为 dB。

➤ **举例:**

:CALCulate:LLINe1:OFFSet:Y 5              设置限值 1 Y 轴偏移 5dB

:CALCulate:LLINe1:OFFSet:Y?              查询返回 5.000000e+00

**:CALCulate:LLINe<n>:OFFSet:UPDate**

➤ **命令格式:**

:CALCulate:LLINe<n>:OFFSet:UPDate

➤ **功能描述:**

指定限值应用偏移。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

➤ **返回格式:**

无返回。

➤ **举例:**

:CALCulate:LLINe1:OFFSet:UPDate 限值 1 应用偏移

**:CALCulate:LLINe<n>:DELeTe**

➤ **命令格式:**

:CALCulate:LLINe<n>:DELeTe

➤ **功能描述:**

删除指定限值数据。

<n>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。

UTS1015B 的最大限值数为 4, UTS3036B 和 UTS3084B 的最大限值数为 6。

➤ **返回格式:**

无返回。

➤ **举例:**

:CALCulate:LLINe1:DELeTe 删除限值 1 数据

**:CALCulate:LLINe:ALL:DELeTe**

➤ **命令格式:**

:CALCulate:LLINe:ALL:DELeTe

➤ **功能描述:**

删除所有限值数据。

➤ **返回格式:**

无返回。

➤ **举例:**

:CALCulate:LLINe:ALL:DELeTe 删除所有限值数据

## CONFigure 命令

**:CONFigure:COUPle**

➤ **命令格式:**

:CONFigure:COUPle

➤ **功能描述:**

自动耦合。

➤ **返回格式:**

无返回值。

- **举例：**  
:CONFIgure:COUPle                      自动耦合

#### :CONFIgure:MEASure:DEFAult

- **命令格式：**  
:CONFIgure:MEASure:DEFAult
- **功能描述：**  
测量重置。
- **返回格式：**  
无返回值。
- **举例：**  
:CONFIgure:MEASure:DEFAult              测量重置

## DISPlay 命令

#### :DISPlay:ZBmp?

- **命令格式：**  
:DISPlay:ZBmp?
- **功能描述：**  
获取屏幕图像。
- **返回格式：**  
查询返回屏幕图像数据，图片数据经过了压缩。
- **举例：**  
:DISPlay:ZBmp?                      获取屏幕图像

#### :DISPlay:VIEW:WINDow:TRACe:X:SPACing

- **命令格式：**  
:DISPlay:VIEW:WINDow:TRACe:X:SPACing {LINear|LOGarithmic}  
:DISPlay:VIEW:WINDow:TRACe:X:SPACing?
- **功能描述：**  
选择 X 轴刻度类型。  
LINear：线性。  
LOGarithmic：对数。
- **返回格式：**  
查询返回 X 轴刻度类型，LINear 或 LOGarithmic。
- **举例：**  
:DISPlay:VIEW:WINDow:TRACe:X:SPACing LOGarithmic              选择 X 轴刻度类型为对数

:DISPlay:VIEW:WINDow:TRACe:X:SPACing?

查询返回 LOGarithmic

#### :DISPlay:FSCan:WINDow:MAMarker[:STATe]

➤ **命令格式:**

:DISPlay:FSCan:WINDow:MAMarker[:STATe] {{1|ON}|{0|OFF}}

:DISPlay:FSCan:WINDow:MAMarker[:STATe]?

➤ **功能描述:**

标记测量窗口显示开关。

1|ON: 显示。

0|OFF: 关闭。

➤ **返回格式:**

查询返回标记测量窗口显示开关状态, 0 或 1。

➤ **举例:**

:DISPlay:FSCan:WINDow:MAMarker:STATe ON

显示标记测量窗口

:DISPlay:FSCan:WINDow:MAMarker:STATe?

查询返回 1

#### :DISPlay:METER<n>[:STATe]

➤ **命令格式:**

:DISPlay:METER<n>[:STATe] {{1|ON}|{0|OFF}}

:DISPlay:METER<n>[:STATe]?

➤ **功能描述:**

指定计量表显示开关。

1|ON: 显示。

0|OFF: 关闭。

➤ **返回格式:**

查询返回指定计量表显示开关状态, 0 或 1。

➤ **举例:**

:DISPlay:METER1:STATe ON

显示标记测量窗口

:DISPlay:METER1:STATe?

查询返回 1

#### :DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel

➤ **命令格式:**

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel <real>

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel?

➤ **功能描述:**

调整参考电平。

<real>: 连续实数, 默认单位为 dBm。

➤ **返回格式:**

查询返回参考电平值, 以科学记数法返回数据, 单位为 dBm。

➤ **举例:**

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:SCALe:RLEVel -10 设置参考电平为-10dBm  
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:SCALe:RLEVel? 查询返回-1.000000e+01

#### :DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:PDIVision

➤ **命令格式:**

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:PDIVision <real>  
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:PDIVision?

➤ **功能描述:**

调整 Y 轴刻度。

<real>: 离散实数, 默认单位 dB, 取值范围 0.1dB 到 20dB。

➤ **返回格式:**

查询返回 Y 轴刻度值, 以科学记数法返回数据, 单位为 dB。

➤ **举例:**

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:PDIVision 1 设置 Y 轴刻度为 1dB  
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:PDIVision? 查询返回 1.000000e+00

#### :DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet

➤ **命令格式:**

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet <real>  
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y[:SCALe]:RLEVel:OFFSet?

➤ **功能描述:**

调整参考电平偏移。

<real>: 连续实数, 默认单位为 dB。

➤ **返回格式:**

查询返回参考电平偏移, 以科学记数法返回数据, 单位为 dB。

➤ **举例:**

:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:RLEVel:OFFSet 5 设置参考电平偏移 5dB  
:DISPlay:FSCan:VIEW:WINDow:TRACe:Y:RLEVel:OFFSet? 查询返回 5.000000e+00

## INITiate 命令

#### :INITiate2:CONTinuous

➤ **命令格式:**

:INITiate2:CONTinuous {{1|ON} | {0|OFF}}

➤ **功能描述:**

扫频单次/连续切换。

1|ON: 连续扫描

0|OFF: 单次扫描

- **返回格式:**  
无返回值。
- **举例:**  
:INITiate2:CONTinuous ON                    连续扫描

#### :INITiate:IMMediate

- **命令格式:**  
:INITiate:IMMediate
- **功能描述:**  
开始扫频。
- **返回格式:**  
无返回值。
- **举例:**  
:INITiate:IMMediate                    开始扫频

#### :INITiate:REStart

- **命令格式:**  
:INITiate: REStart
- **功能描述:**  
重新开始扫频。
- **返回格式:**  
无返回值。
- **举例:**  
:INITiate:REStart                    重新开始扫频

#### :INITiate:STOP

- **命令格式:**  
:INITiate: STOP
- **功能描述:**  
停止扫频。
- **返回格式:**  
无返回值。
- **举例:**  
:INITiate: STOP                    停止扫频

#### :INITiate:PAUSe

- **命令格式:**  
:INITiate:PAUSe
- **功能描述:**



## MMEMory 命令

### :MMEMory:LOAD:CORRection

➤ **命令格式:**

:MMEMory:LOAD:CORRection {<integer>,<filename>}

➤ **功能描述:**

在默认目录下加载文件到指定修正。

<integer>: 修正序列号, 连续整数, 取值范围 1-10

<filename>: 文件名, 文件后缀 .corr。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:LOAD:CORRection 1, "emi.corr"                      修正 1 加载数据文件 emi.corr

### :MMEMory:LOAD:LIMit

➤ **命令格式:**

:MMEMory:LOAD:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6,<filename>}

➤ **功能描述:**

在默认目录下加载文件到指定限值。

LLINE1-LLINE6: 依次对应限值 1 到限值 6

<filename>: 文件名, 文件后缀 .limit。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:LOAD:LIMit LLINE1, "emi.limit"                      限值 1 加载数据文件 emi.limit

### :MMEMory:LOAD:STATe

➤ **命令格式:**

:MMEMory:LOAD:STATe <filename>

➤ **功能描述:**

在默认目录下加载寄存器状态文件。

<filename>: 文件名, 文件后缀 .state。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:LOAD:STATe "emi.state"                      加载寄存器状态文件 emi.state

## :MMEMory:LOAD:TRACe

### ➤ 命令格式:

:MMEMory:LOAD:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, <filename>}

### ➤ 功能描述:

在默认目录下加载文件到指定迹线。

TRACE1-TRACE6: 依次对应迹线 1 到迹线 6

<filename>: 文件名, 文件后缀 .trace。文件名整体作为一个字符串, 需用引号包裹。

### ➤ 返回格式:

无返回。

### 举例:

:MMEMory:LOAD:TRACe TRACE1, "emi.trace"                    迹线 1 加载数据文件 emi.trace

## :MMEMory:LOAD:SCAN

### ➤ 命令格式:

:MMEMory:LOAD:SCAN <filename>

### ➤ 功能描述:

在默认目录下加载扫描表文件。

<filename>: 文件名, 文件后缀 .csv。文件名整体作为一个字符串, 需用引号包裹。

### ➤ 返回格式:

无返回。

### 举例:

:MMEMory:LOAD:SCAN "scan.csv"                                加载扫描表文件 scan.csv

## :MMEMory:LOAD:SLISt

### ➤ 命令格式:

:MMEMory:LOAD:SLISt <filename>

### ➤ 功能描述:

在默认目录下加载信号表文件。

<filename>: 文件名, 文件后缀 .csv。文件名整体作为一个字符串, 需用引号包裹。

### ➤ 返回格式:

无返回。

### 举例:

:MMEMory:LOAD:SLISt "slist.csv"                                加载信号表文件 slist.csv

## :MMEMory:STORe:CORRection

### ➤ 命令格式:

:MMEMory:STORe:CORRection {<integer>, <filename>}

### ➤ 功能描述:

以文件方式保存指定修正数据到默认目录。

<integer>: 修正序列号, 连续整数, 取值范围 1-10

<filename>: 修正文件名, 文件后缀 .corr。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:CORRection 1, "emi.corr"                    保存修正 1 数据到文件 emi.corr

**:MMEMory:STORe:LIMit**

➤ **命令格式:**

:MMEMory:STORe:LIMit {LLINE1|LLINE2|LLINE3|LLINE4|LLINE5|LLINE6, <filename>}

➤ **功能描述:**

以文件方式保存指定限值数据到默认目录。

LLINE1-LLINE6: 依次对应限值 1 到限值 6

<filename>: 限值文件名, 文件后缀 .limit。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:LIMit LLINE1, "emi.limit"                    保存限值 1 数据到文件 emi.limit

**:MMEMory:STORe:STATe**

➤ **命令格式:**

:MMEMory:STORe:STATe <filename>

➤ **功能描述:**

以文件方式保存寄存器状态到默认目录。

<filename>: 状态文件名, 文件后缀 .state。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:STATe "emi.state"                            保存寄存器状态到文件 emi.state

**:MMEMory:STORe:TRACe**

➤ **命令格式:**

:MMEMory:STORe:TRACe {TRACE1|TRACE2|TRACE3|TRACE4|TRACE5|TRACE6, <filename>}

➤ **功能描述:**

以文件方式保存指定迹线数据到默认目录。

TRACE1-TRACE6: 依次对应迹线 1 到迹线 6

<filename>: 迹线文件名, 文件后缀 .trace。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:TRACe TRACE1, "emi.trace" 保存迹线 1 的数据到文件 emi.trace

#### :MMEMory:STORe:SCAN

➤ **命令格式:**

:MMEMory:STORe:SCAN <filename>

➤ **功能描述:**

以文件方式保存扫描表数据到默认目录。

<filename>: 扫描表文件名, 文件后缀.csv。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:SCAN "scan.csv" 保存迹线 1 的数据到文件 scan.csv

#### :MMEMory:STORe:SLISt

➤ **命令格式:**

:MMEMory:STORe:SLISt <filename>

➤ **功能描述:**

以文件方式保存信号表数据到默认目录。

<filename>: 信号表文件名, 文件后缀.csv。文件名整体作为一个字符串, 需用引号包裹。

➤ **返回格式:**

无返回。

**举例:**

:MMEMory:STORe:SLISt "slist.csv" 保存迹线 1 的数据到文件 slist.csv

## SENSe 命令

### [ :SENSe ] :FREQuency:STARt

➤ **命令格式:**

[ :SENSe ] :FREQuency:STARt <freq>

[ :SENSe ] :FREQuency :STARt ?

➤ **功能描述:**

设置扫频起始频率。

<freq>: 连续实数, 默认单位为 Hz。

起始频率范围是 0Hz 到最大频率-100Hz。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

➤ **返回格式:**

查询返回起始频率值, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:FREQuency :STARt 10MHz      设置扫频起始频率为 10MHz

:FREQuency :STARt ?              查询返回 1.000000e+07

[ :SENSe ] :FREQuency :STARt :AUTO

➤ **命令格式:**

[ :SENSe ] :FREQuency :STARt :AUTO { {1|ON} | {0|OFF} }

[ :SENSe ] :FREQuency :STARt :AUTO ?

➤ **功能描述:**

手动/自动起始频率切换。

1|ON: 自动

0|OFF: 手动

➤ **返回格式:**

查询返回自动起始频率状态, 0 或 1。

➤ **举例:**

:FREQuency :STARt :AUTO ON              设置自动起始频率

:FREQuency :STARt :AUTO ?              查询返回 1

[ :SENSe ] :FREQuency :STOP

➤ **命令格式:**

[ :SENSe ] :FREQuency :STOP <freq>

[ :SENSe ] :FREQuency :STOP ?

➤ **功能描述:**

设置扫频截止频率。

<freq>: 连续实数, 默认单位为 Hz。

截止频率范围是 100Hz 到最大频率。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

➤ **返回格式:**

查询返回截止频率值, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:FREQuency :STOP 1GHz      设置扫频截止频率为 1GHz

:FREQuency :STOP ?              查询返回 1.000000e+09

## **[ :SENSe ] :FREQuency :STOP :AUTO**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :STOP :AUTO { {1|ON} | {0|OFF} }

[ :SENSe ] :FREQuency :STOP :AUTO?

### ➤ **功能描述:**

手动/自动截止频率切换。

1|ON: 自动

0|OFF: 手动

### ➤ **返回格式:**

查询返回自动截止频率状态, 0 或 1。

### ➤ **举例:**

:FREQuency :STOP :AUTO ON                    设置自动截止频率

:FREQuency :STOP :AUTO?                    查询返回 1

## **[ :SENSe ] :FREQuency :CENTer**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :CENTer <freq>

[ :SENSe ] :FREQuency :CENTer?

### ➤ **功能描述:**

设置计量表频率。

<freq>: 连续实数, 默认单位为 Hz。

频率范围是 0 到最大频率。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

### ➤ **返回格式:**

查询返回计量表频率值, 以科学计数法返回数据, 单位为 Hz。

### ➤ **举例:**

:FREQuency :CENTer 1GHz                    设置计量表频率为 1GHz

:FREQuency :CENTer?                    查询返回 1.000000e+09

## **[ :SENSe ] :FREQuency :MIDSpan**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :MIDSpan <freq>

[ :SENSe ] :FREQuency :MIDSpan?

### ➤ **功能描述:**

设置扫频中心频率。

<freq>: 连续实数, 默认单位为 Hz。

频率范围是 50Hz 到最大频率-50Hz。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

### ➤ **返回格式:**

查询返回中心频率值, 以科学计数法返回数据, 单位为 Hz。

- **举例：**  
:FREQUENCY:MIDSPAN 1GHz          设置扫频中心频率为 1GHz  
:FREQUENCY:MIDSPAN?              查询返回 1.000000e+09

#### **[ :SENSe ] :FREQUENCY :SPAN**

- **命令格式：**  
[ :SENSe ] :FREQUENCY :SPAN <freq>  
[ :SENSe ] :FREQUENCY :SPAN?
- **功能描述：**  
设置扫宽。  
<freq>：连续实数，默认单位为 Hz。  
扫宽范围是 100Hz 到最大频率。  
UTS1015B 的最大频率是 1.5GHz，UTS3036B 的最大频率是 3.6GHz。
- **返回格式：**  
查询返回扫宽值，以科学计数法返回数据，单位 Hz。
- **举例：**  
:FREQUENCY:SPAN 1GHz 设置扫宽为 1GHz  
:FREQUENCY:SPAN?          查询返回 1.000000e+09

#### **[ :SENSe ] :POWER [ :RF ] :ATTenuation**

- **命令格式：**  
[ :SENSe ] :POWER [ :RF ] :ATTenuation <ampl>  
[ :SENSe ] :POWER [ :RF ] :ATTenuation?
- **功能描述：**  
调整计量表输入衰减。  
<ampl>：连续整数，默认单位为 dB，取值范围 0dB 到 51dB
- **返回格式：**  
查询返回计量表输入衰减值，单位为 dB。
- **举例：**  
:POWER:ATTenuation 10          设置计量表输入衰减 10dB  
:POWER:ATTenuation?          查询返回 10

#### **[ :SENSe ] :POWER [ :RF ] :GAIN :STATe**

- **命令格式：**  
[ :SENSe ] :POWER [ :RF ] :GAIN :STATe {{1|ON} | {0|OFF}}  
[ :SENSe ] :POWER [ :RF ] :GAIN :STATe?
- **功能描述：**  
计量表前置放大器开关。
- **返回格式：**  
查询返回计量表前置放大器开关状态，0 或 1。

- **举例：**  
:POWER:GAIN:STATE ON 打开计量表前置放大器  
:POWER:GAIN:STATE? 查询返回 1

## [ :SENSe ] :CORRection:IMPedance [ :INPut ] [ :MAGNitude ]

### :INPut:IMPedance

- **命令格式：**  
[ :SENSe ] :CORRection:IMPedance [ :INPut ] [ :MAGNitude ] { 50 | 75 }  
[ :SENSe ] :CORRection:IMPedance [ :INPut ] [ :MAGNitude ] ?  
:INPut:IMPedance { 50 | 75 }  
:INPut:IMPedance ?
- **功能描述：**  
选择输入阻抗，50 Ω 或 75 Ω。
- **返回格式：**  
查询返回输入阻抗值，50 或 75，单位 Ω。
- **举例：**  
:CORRection:IMPedance 50            设置输入阻抗为 50 Ω  
:CORRection:IMPedance?            查询返回 50  
:INPut:IMPedance 50                设置输入阻抗为 50 Ω  
:INPut:IMPedance?                 查询返回 50

## [ :SENSe ] :BANDwidth|BWIDth [ :RESolution ]

- **命令格式：**  
[ :SENSe ] :BANDwidth|BWIDth [ :RESolution ] <freq>  
[ :SENSe ] :BANDwidth|BWIDth [ :RESolution ] ?
- **功能描述：**  
调整计量表分辨率带宽。  
<freq>：离散实数，默认单位为 Hz。EMI 标准选择 CISPR 时，取值限定 200Hz/9kHz/120kHz/1MHz；EMI 标准选择 None 时，取值范围 1kHz 到最大分辨率带宽，以 1-3-10 为步进。UTS1015B 的最大分辨率带宽为 1MHz，UTS3036B 和 UTS3084B 的最大分辨率带宽为 3MHz。
- **返回格式：**  
查询返回计量表分辨率带宽值，以科学计数法返回数据，单位为 Hz。
- **举例：**  
:BANDwidth 1MHz            设置计量表分辨率带宽为 1MHz  
:BANDwidth?                查询返回 1.000000e+06

## [ :SENSe ] :BANDwidth|BWIDth [ :RESolution ] :AUTO

- **命令格式：**  
[ :SENSe ] :BANDwidth|BWIDth [ :RESolution ] :AUTO { { 1 | ON } | { 0 | OFF } }

[ :SENSe ] :BANDwidth | BWIDth [ :RESolution ] :AUTO?

➤ **功能描述:**

计量表自动/手动分辨率带宽切换。

1|ON: 自动

0|OFF: 手动

➤ **返回格式:**

查询返回计量表自动分辨率带宽状态, 0 或 1。

➤ **举例:**

:BANDwidth:AUTO ON            设置计量表自动分辨率带宽

:BANDwidth:AUTO?            查询返回 1

[ :SENSe ] :FSCan:DETECTOR:TRACe<n>

➤ **命令格式:**

[ :SENSe ] :FSCan:DETECTOR:TRACe<n> { POSitive | QPEak | CAVERage | AVERage | NEGative }

[ :SENSe ] :FSCan:DETECTOR:TRACe<n>?

➤ **功能描述:**

选择指定迹线检波器类型。

<n>: 迹线序列号, 取值范围 1 到 3, EMI 只有 3 条迹线。

POSitive: 峰值检波

QPEak: 准峰值检波

CAVERage: EMI 平均值检波

AVERage: 平均检波

NEGative: 负峰值检波

➤ **返回格式:**

查询返回当前迹线检波器类型, POSitive、QPEak、CAVERage、AVERage 或 NEGative。

➤ **举例:**

:FSCan:DETECTOR:TRACe1 POSitive            选择迹线 1 检波器类型为峰值检波

:FSCan:DETECTOR:TRACe1?            查询返回 POSitive

[ :SENSe ] :DETECTOR:TRACe<n>:AUTO

➤ **命令格式:**

[ :SENSe ] :DETECTOR:TRACe<n>:AUTO { {1|ON} | {0|OFF} }

[ :SENSe ] :DETECTOR:TRACe<n>:AUTO?

➤ **功能描述:**

指定迹线自动检波器开关。

<n>: 迹线序列号, 取值范围 1 到 3, EMI 只有 3 条迹线。

➤ **返回格式:**

查询返回指定迹线自动检波器开关状态, 0 或 1。

➤ **举例:**

:DETECTOR:TRACe1:AUTO ON            打开迹线 1 自动检波器开关

:DETECTOR:TRACe1:AUTO?            查询返回 1

## **[ :SENSe] :CORRection:SElect**

### ➤ **命令格式:**

[ :SENSe] :CORRection:SElect <integer>

[ :SENSe] :CORRection:SElect?

### ➤ **功能描述:**

从修正序列中选择一个作为当前修正。

<integer>: 修正序列号, 取值范围 1 到 10。

### ➤ **返回格式:**

查询返回当前修正序列号, 范围 1 到 10。

### ➤ **举例:**

:CORRection:SElect 2                    选择修正 2

:CORRection:SElect?                    查询返回 2

## **[ :SENSe] :CORRection:CSET<n>[:STATe]**

### ➤ **命令格式:**

[ :SENSe] :CORRection:CSET<n>[:STATe] {{1|ON} | {0|OFF}}

[ :SENSe] :CORRection:CSET<n>[:STATe]?

### ➤ **功能描述:**

指定修正开关。

<n>: 修正序列号, 取值范围 1 到 10。

### ➤ **返回格式:**

查询返回指定修正开关状态, 0 或 1。

### ➤ **举例:**

:CORRection:CSET1 ON                    打开修正 1

:CORRection:CSET1?                    查询返回 1

## **[ :SENSe] :CORRection:CSET:ALL[:STATe]**

### ➤ **命令格式:**

[ :SENSe] :CORRection:CSET:ALL[:STATe]

### ➤ **功能描述:**

关闭所有修正。

### ➤ **返回格式:**

无返回。

### ➤ **举例:**

:CORRection:CSET:ALL                    关闭所有修正

## **[ :SENSe] :CORRection:CSET<n>:DATA**

### ➤ **命令格式:**

[ :SENSe] :CORRection:CSET<n>:DATA {<freq>, <ampl>, <freq>, <ampl>, ... }

[ :SENSe] :CORRection:CSET<n>:DATA?

➤ **功能描述:**

编辑指定修正数据。

<n>: 修正序列号, 取值范围 1 到 10。

<freq>: 修正点频率, 单位 Hz。

<ampl>: 修正点幅度, 单位 dB。

➤ **返回格式:**

查询返回指定修正数据, 以科学记数法返回数据, 以 {频率, 幅值, 频率, 幅值, ...} 结构返回, 频率单位为 Hz, 幅值单位为 dB。

➤ **举例:**

:CORRection:CSET1:DATA 10000000, 5

编辑修正 1 (10000000, 5)

:CORRection:CSET1:DATA?

查询返回 1.000000e+07, 5.000000e+00

**[ :SENSe] :CORRection:CSET<n>:DELeTe**

➤ **命令格式:**

[ :SENSe] :CORRection:CSET<n>:DELeTe

➤ **功能描述:**

删除指定修正数据。

<n>: 修正序列号, 取值 1-10

➤ **返回格式:**

无返回。

➤ **举例:**

:CORRection:CSET1:DELeTe      删除修正 1 数据

**[ :SENSe] :CORRection:CSET:ALL:DELeTe**

➤ **命令格式:**

[ :SENSe] :CORRection:CSET:ALL:DELeTe

➤ **功能描述:**

删除所有修正数据。

➤ **返回格式:**

无返回。

➤ **举例:**

:CORRection:CSET:ALL:DELeTe      删除所有修正数据

**[ :SENSe] :AVERAge:COUNT**

➤ **命令格式:**

[ :SENSe] :AVERAge:COUNT <integer>

[ :SENSe] :AVERAge:COUNT?

➤ **功能描述:**

设置平均次数。

<integer>: 整数, 取值范围 1 到 999。

➤ **返回格式:**

查询返回平均次数, 取值 1 到 999。

➤ **举例:**

:AVERage:COUNT 10      设置平均次数为 10

:AVERage:COUNT?      查询返回 10

**[ :SENSe ] :AVERage :TYPE**

➤ **命令格式:**

[ :SENSe ] :AVERage :TYPE {VOLTage | POWER | LOG}

[ :SENSe ] :AVERage :TYPE?

➤ **功能描述:**

设置平均类型。

VOLTage: 电压平均

POWER: 功率平均

LOG: 对数平均

➤ **返回格式:**

查询返回平均类型, VOLTage、POWER 或 LOG。

➤ **举例:**

:AVERage:TYPE VOLTage      设置电压平均

:AVERage:TYPE?      查询返回 VOLTage

**[ :SENSe ] :EMC :STANdard [ :SElect ]**

➤ **命令格式:**

[ :SENSe ] :EMC :STANdard [ :SElect ] {NONE | CISPr}

[ :SENSe ] :EMC :STANdard [ :SElect ]?

➤ **功能描述:**

选择 EMI 标准。

NONE: 无

CISPr: 国际无线电干扰特别委员会, 负责 >9KHz 所有类型电器的 EMI 无线电信号保护测试标准规范的编写。

➤ **返回格式:**

查询返回平均类型, NONE 或 CISPr。

➤ **举例:**

:EMC:STANdard CISPr      设置标准为 CISPR

:EMC:STANdard?      查询返回 CISPr

**[ :SENSe ] :FSCan :SEquence**

➤ **命令格式:**

[ :SENSe ] :FSCan :SEquence {SCAN | SEARch | SSAMeasure | SASearch | SAMeasure | REMeasure}

[ :SENSe ] :FSCan :SEquence?

- **功能描述:**  
选择 EMI 频谱扫描类型。  
SCAN: 仅扫描  
SEARCh: 仅搜索  
SSAMeasure: 扫描搜索测量  
SASearch: 扫描搜索  
SAMeasure: 搜索测量  
REMeasure: 测量
- **返回格式:**  
查询返回 EMI 频谱扫描类型, SCAN、SEARCh、SSAMeasure、SASearch、SAMeasure 或 REMeasure。
- **举例:**  
:FSCan:SEquence SCAN      设置 EMI 频谱扫描类型为 SCAN  
:FSCan:SEquence?          查询返回 SCAN

#### **[ :SENSe ] :FSCan:SEquence:REMeasure**

- **命令格式:**  
[ :SENSe ] :FSCan:SEquence:REMeasure {CURRent | ALL | MARKed}  
[ :SENSe ] :FSCan:SEquence:REMeasure?
- **功能描述:**  
选择测量类型。  
CURRent: 当前信号  
MARKed: 标记信号  
ALL: 所有信号
- **返回格式:**  
查询返回测量类型类型, CURRent、MARKed 或 ALL。
- **举例:**  
:FSCan:SEquence:REMeasure MARKed      设置测量类型为标记信号  
:FSCan:SEquence:REMeasure?          查询返回 MARKed

#### **[ :SENSe ] :METer:DETEctor:DWELI**

- **命令格式:**  
[ :SENSe ] :METer:DETEctor:DWELI <time>  
[ :SENSe ] :METer:DETEctor:DWELI?
- **功能描述:**  
设置计量表驻留时间。  
<time>: 驻留时间, 默认单位 s, 取值范围 1ms 到 100s。
- **返回格式:**  
查询返回计量表驻留时间, 以科学记数法返回数据, 单位 s。
- **举例:**  
:METer:DETEctor:DWELI 0.5              设置计量表驻留时间为 500ms  
:METer:DETEctor:DWELI?              查询返回 5.000000e-01

## [ :SENSe ] :METer<n> :DETEctor

### ➤ 命令格式:

[ :SENSe ] :METer<n> :DETEctor { POSitive | QPEak | CAVerage | AVERage | NEGative }

[ :SENSe ] :METer<n> :DETEctor ?

### ➤ 功能描述:

选择指定计量表检波器类型。

<n>: 指定计量表序列号, 连续整数, 取值 1-3。

POSitive: 峰值检波器。

QPEak: 准峰值检波器。

CAVerage: EMI 平均检波器。

AVERage: 平均检波器。

NEGative: 负峰值检波器。

### ➤ 返回格式:

查询返回指定计量表检波器类型, POSitive、QPEak、CAVerage、AVERage 或 NEGative。

### ➤ 举例:

:METer1:DETEctor POSitive                      选择计量表 1 的检波器为峰值检波器

:METer1:DETEctor ?                              查询返回 POSitive

## [ :SENSe ] :METer :PHOLd :TYPE

### ➤ 命令格式:

[ :SENSe ] :METer :PHOLd :TYPE { ADJustable | INFInite }

[ :SENSe ] :METer :PHOLd :TYPE ?

### ➤ 功能描述:

选择计量表最大保持类型。

INFInite: 无限, 不周期性更新最大值。

ADJustable: 可调, 以最大保持时间为间隔, 周期性更新最大值。

### ➤ 返回格式:

查询返回计量表最大保持类型, ADJustable 或 INFInite。

### ➤ 举例:

:METer:PHOLd:TYPE ADJustable                      选择计量表最大保持类型为无限

:METer:PHOLd:TYPE ?                              查询返回 ADJustable

## [ :SENSe ] :METer :PHOLd :ADJustable

### ➤ 命令格式:

[ :SENSe ] :METer :PHOLd :ADJustable <time>

[ :SENSe ] :METer :PHOLd :ADJustable ?

### ➤ 功能描述:

设置计量表最大保持时间。

<time>: 最大保持时间, 默认单位为 s, 取值范围 500ms 到 100s。

### ➤ 返回格式:

查询返回计量表最大保持时间，以科学记数法返回数据，单位 s。

➤ **举例：**

:METer:PHOLd:ADJustable 1                    设置计量表最大保持时间为 1s  
:METer:PHOLd:ADJustable?                    查询返回 1.000000e+00

**[ :SENSe ] :FSCan:SLISt:COUPl e:METer**

➤ **命令格式：**

[ :SENSe ] :FSCan:SLISt:COUPl e:METer {{1|ON} | {0|OFF}}  
[ :SENSe ] :FSCan:SLISt:COUPl e:METer?

➤ **功能描述：**

计量表耦合到信号表开关。

➤ **返回格式：**

查询返回计量表耦合到信号表开关状态，0 或 1。

➤ **举例：**

:FSCan:SLISt:COUPl e:METer ON                计量表耦合到信号表开  
:FSCan:SLISt:COUPl e:METer?                查询返回 1

**[ :SENSe ] :FSCan:SEARch:MODE**

➤ **命令格式：**

[ :SENSe ] :FSCan:SEARch:MODE {PONLy|PLIMits|SLIMits}  
[ :SENSe ] :FSCan:SEARch:MODE?

➤ **功能描述：**

选择信号搜索条件。

PONLy: 峰值。

PLIMits: 峰值与限值。

SLIMits: 分段与限值。

➤ **返回格式：**

查询返回信号搜索条件，PONLy、PLIMits 或 SLIMits。

➤ **举例：**

:FSCan:SEARch:MODE PONLy                    搜索条件选择峰值与限值  
:FSCan:SEARch:MODE?                    查询返回 PONLy

**[ :SENSe ] :FSCan:SEARch:PEAK:COUNT**

➤ **命令格式：**

[ :SENSe ] :FSCan:SEARch:PEAK:COUNT <integer>  
[ :SENSe ] :FSCan:SEARch:PEAK:COUNT?

➤ **功能描述：**

设置信号搜索中的峰值数。

<integer>: 峰值数，连续整数，取值范围 1-50。

➤ **返回格式：**

查询返回信号搜索中的峰值数。

➤ **举例：**

:FSCan:SEARch:PEAK:COUNT 10                    设置信号搜索中的峰值数为 10  
:FSCan:SEARch:MODE?                                查询返回 10

**[[:SENSe]:FSCan:SEARch:SUBRange:COUNT**

➤ **命令格式：**

[[:SENSe]:FSCan:SEARch:SUBRange:COUNT <integer>  
[:SENSe]:FSCan:SEARch:SUBRange:COUNT?

➤ **功能描述：**

设置信号搜索中的频段数。

<integer>: 频段数, 连续整数, 取值范围 1-50。

➤ **返回格式：**

查询返回信号搜索中的频段数。

➤ **举例：**

:FSCan:SEARch:SUBRange:COUNT 10                设置信号搜索中的频段数为 10  
:FSCan:SEARch:SUBRange:COUNT?                查询返回 10

**[[:SENSe]:FSCan:SCAN<n>:STATe**

➤ **命令格式：**

[[:SENSe]:FSCan:SCAN<n>:STATe {{1|ON} | {0|OFF}}  
[:SENSe]:FSCan:SCAN<n>:STATe?

➤ **功能描述：**

使能扫描表中指定频段。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

➤ **返回格式：**

查询返回扫描表中指定频段使能状态, 0 或 1。

➤ **举例：**

:FSCan:SCAN5:STATe ON                            使能扫描表中频段 5  
:FSCan:SCAN5:STATe?                            查询返回 1

**[[:SENSe]:FSCan:SCAN<n>:START**

➤ **命令格式：**

[[:SENSe]:FSCan:SCAN<n>:START <freq>  
[:SENSe]:FSCan:SCAN<n>:START?

➤ **功能描述：**

设置扫描表中指定频段起始频率。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

<freq>: 扫描表频段起始频率, 连续实数, 默认单位为 Hz。起始频率范围是 0Hz 到最大频率-100Hz。UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

➤ **返回格式:**

查询返回扫描表中指定频段起始频率, 用科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:FSCan:SCAN5:START 100MHz            扫描表中频段 5 起始频率设置为 100MHz  
:FSCan:SCAN5:START?                    查询返回 1.000000e+08

**[[:SENSe]:FSCan:SCAN<n>:STOP**

➤ **命令格式:**

[[:SENSe]:FSCan:SCAN<n>:STOP <freq>  
[:SENSe]:FSCan:SCAN<n>:STOP?

➤ **功能描述:**

设置扫描表中指定频段截止频率。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

<freq>: 扫描表频段截止频率, 连续实数, 默认单位为 Hz。截止频率范围是 100Hz 到最大频率。UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz, UTS3084B 的最大频率是 8.4GHz。

➤ **返回格式:**

查询返回扫描表中指定频段截止频率, 用科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:FSCan:SCAN5:STOP 1GHz                扫描表中频段 5 截止频率设置为 1GHz  
:FSCan:SCAN5:STOP?                    查询返回 1.000000e+09

**[[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]**

➤ **命令格式:**

[[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution] <freq>  
[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]?

➤ **功能描述:**

设置扫描表中指定频段分辨率带宽。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

<freq>: 离散实数, 默认单位为 Hz。当 EMI 测量标准为 CISPR 时, 分辨率带宽为 200Hz/9kHz/120kHz/1MHz; 当 EMI 测量标准为 None 时, 取值范围 1kHz 到最大分辨率带宽, 以 1-3-10 为步进。

UTS1015B 的最大分辨率带宽为 1MHz, UTS3036B 和 UTS3084B 的最大分辨率带宽为 3MHz。

➤ **返回格式:**

查询返回扫描表中指定频段分辨率带宽, 用科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:FSCan:SCAN5:BANDwidth:RESolution 1MHz    扫描表中频段 5 分辨率带宽设置为 1MHz  
:FSCan:SCAN5:BANDwidth:RESolution?        查询返回 1.000000e+06

**[[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO**

➤ **命令格式:**

[[:SENSe]:FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO {{1|ON} | {0|OFF}}

[ :SENSe] :FSCan:SCAN<n>:BANDwidth[:RESolution]:AUTO?

➤ **功能描述:**

扫描表中指定频段自动/手动分辨率带宽切换。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

➤ **返回格式:**

查询返回扫描表中指定频段分辨率带宽自动/手动状态, 0 或 1。

➤ **举例:**

:FSCan:SCAN5:BANDwidth:AUTO ON            扫描表中频段 5 自动分辨率带宽  
:FSCan:SCAN5:BANDwidth:AUTO?            查询返回 1

[ :SENSe] :FSCan:SCAN<n>:TIME

➤ **命令格式:**

[ :SENSe] :FSCan:SCAN<n>:TIME <time>

[ :SENSe] :FSCan:SCAN<n>:TIME?

➤ **功能描述:**

设置扫描表中指定频段扫描时间。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

< time >: 扫描表频段扫描时间, 连续实数, 取值范围 1ms 到 4ks。

➤ **返回格式:**

查询返回扫描表中指定频段扫描时间, 用科学计数法返回数据, 单位为 s。

➤ **举例:**

:FSCan:SCAN5:TIME 10ms    扫描表中频段 5 扫描时间设置为 10ms  
:FSCan:SCAN5:TIME?        查询返回 1.000000e-02

[ :SENSe] :FSCan:SCAN<n>:TIME:AUTO

➤ **命令格式:**

[ :SENSe] :FSCan:SCAN<n>:TIME:AUTO {{1|ON} | {0|OFF}}

[ :SENSe] :FSCan:SCAN<n>:TIME:AUTO?

➤ **功能描述:**

扫描表中指定频段自动/手动扫描时间切换。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

➤ **返回格式:**

查询返回扫描表中指定频段扫描时间自动/手动状态。

➤ **举例:**

:FSCan:SCAN5:TIME:AUTO ON    扫描表中频段 5 自动扫描时间  
:FSCan:SCAN5:TIME:AUTO?        查询返回 1

[ :SENSe] :FSCan:SCAN<n>:INPut:ATTenuat ion

➤ **命令格式:**

[ :SENSe] :FSCan:SCAN<n>:INPut:ATTenuat ion <integer>

[ :SENSe ] :FSCan :SCAN<n> :INPut :ATTenuation?

➤ **功能描述:**

设置扫描表中指定频段输入衰减值。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

<integer>: 频段输入衰减值, 为偶数, 默认单位为 dB, 取值范围 0dB 到 50dB。

➤ **返回格式:**

查询返回扫描表中指定频段输入衰减值, 单位为 dB。

➤ **举例:**

:FSCan:SCAN5:INPut:ATTenuation 6dB	设置扫描表中频段 5 输入衰减值为 6dB
:FSCan:SCAN5:INPut:ATTenuation?	查询返回 6

[ :SENSe ] :FSCan :SCAN<n> :POINt

➤ **命令格式:**

[ :SENSe ] :FSCan :SCAN<n> :POINt <integer>

[ :SENSe ] :FSCan :SCAN<n> :POINt?

➤ **功能描述:**

设置扫描表中指定频段扫描点数。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

<integer>: 频段扫描点数, 连续整数, 取值范围 11 到最大扫描点数。

UTS1015B 的最大点数为 10001, UTS3036B 和 UTS3084B 的最大点数为 40001。

➤ **返回格式:**

查询返回扫描表中指定频段扫描点数。

➤ **举例:**

:FSCan:SCAN5:POINt 2000	设置扫描表中频段 5 扫描点数为 2000
:FSCan:SCAN5:POINt?	查询返回 2000

[ :SENSe ] :FSCan :SCAN<n> :POWer :GAIN [ :STATe ]

➤ **命令格式:**

[ :SENSe ] :FSCan :SCAN<n> :POWer :GAIN [ :STATe ] { {1|ON} | {0|OFF} }

[ :SENSe ] :FSCan :SCAN<n> :POWer :GAIN [ :STATe ] ?

➤ **功能描述:**

扫描表中指定频段前置放大器开关。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

➤ **返回格式:**

查询返回扫描表中指定频段前置放大器开关状态, 0 或 1。

➤ **举例:**

:FSCan:SCAN5:POWer:GAIN:STATe ON	设置扫描表中频段 5 打开前置放大
:FSCan:SCAN5:POWer:GAIN:STATe?	查询返回 1

## **[[:SENSe]:FSCan:RANGe<n>:PREset**

### ➤ **命令格式:**

**[[:SENSe]:FSCan:RANGe<n>:PREset {CISA|CISB|CISC|CISD|CISCD|CISE}**

### ➤ **功能描述:**

扫描表中指定频段预设频段参数。

<n>: 扫描表频段序列号, 连续整数, 取值范围 1-10。

CISA: 预设频段 A, 9kHz-150kHz。

CISB: 预设频段 B, 150kHz-30MHz。

CISC: 预设频段 C, 30MHz-300MHz。

CISD: 预设频段 D, 300MHz-1GHz。

CISCD: 预设频段 CD, 30MHz-1GHz。

CISE: 预设频段 E, UTS1015B 为 1GHz-1.5GHz, UTS3036B 为 1GHz-3.6GHz。

### ➤ **返回格式:**

无返回值。

### ➤ **举例:**

**:FSCan:RANGe5:PREset CISD**      扫描表中频段 5 预设频段 D, 300MHz-1GHz

## **[[:SENSe]:FSCan:FINal:DETEctor<n>**

### ➤ **命令格式:**

**[[:SENSe]:FSCan:FINal:DETEctor<n> {POSitive|QPEak|CAVerage|AVERage|NEGative}**

**[[:SENSe]:FSCan:FINal:DETEctor<n>?**

### ➤ **功能描述:**

指定信号测量检波器设置检波器类型。

<n>: 信号测量检波器序列号, 连续整数, 取值范围 1-3。

POSitive: 峰值检波器。

QPEak: 准峰值检波器。

CAVerage: EMI 平均检波器。

AVERage: 平均检波器。

NEGative: 负峰值检波器。

### ➤ **返回格式:**

查询返回指定信号测量检波器类型, POSitive、QPEak、CAVerage、AVERage 或 NEGative。

### ➤ **举例:**

**:FSCan:FINal:DETEctor1 AVERage**      信号测量检波器 1 为平均检波器

**:FSCan:FINal:DETEctor1?**      查询返回 AVERage

## **[[:SENSe]:FSCan:FINal:DETEctor<n>:DWELI**

### ➤ **命令格式:**

**[[:SENSe]:FSCan:FINal:DETEctor<n>:DWELI <time>**

**[[:SENSe]:FSCan:FINal:DETEctor<n>:DWELI?**

### ➤ **功能描述:**

指定信号测量检波器设置驻留时间。

<n>: 信号测量检波器序列号, 连续整数, 取值范围 1-3。

<time>: 测量驻留时间, 连续实数, 默认单位为 s, 取值范围 1ms 到 60s。

➤ **返回格式:**

查询返回指定信号测量检波器驻留时间, 以科学计数法返回数据, 单位为 s。

➤ **举例:**

:FSCan:FINal:DETEctor1:DWELl 100ms	信号测量检波器 1 驻留时间设置为 100ms
:FSCan:FINal:DETEctor1:DWELl?	查询返回 1.000000e-01

## **[[:SENSe]:FSCan:FINal:DETEctor<n>:LDELta**

➤ **命令格式:**

[[:SENSe]:FSCan:FINal:DETEctor<n>:LDELta <integer>

[[:SENSe]:FSCan:FINal:DETEctor<n>:LDELta?

➤ **功能描述:**

指定信号测量检波器选择限值。

<n>: 信号测量检波器序列号, 连续整数, 取值范围 1-3。

<integer>: 限值序列号, 连续整数, 取值范围 1 到最大限值数。UTS1015B 的最大限值数为 4, UTS3036B 的最大限值数为 6。

➤ **返回格式:**

查询返回指定信号测量检波器限值。

➤ **举例:**

:FSCan:FINal:DETEctor1:LDELta 2	信号测量检波器 1 选择限值 2
:FSCan:FINal:DETEctor1:LDELta?	查询返回 2

## **TRACe 命令**

### **:TRACe:SElect**

➤ **命令格式:**

:TRACe:SElect <integer>

:TRACe:SElect?

➤ **功能描述:**

从所有迹线序列中选择一条作为当前迹线。

<integer>: 迹线序列号, 整数, 取值范围 1-3。

➤ **返回格式:**

查询返回当前迹线序列号。

➤ **举例:**

:TRACe:SElect 1	设置迹线 1 为当前迹线
:TRACe:SElect?	查询返回 1

## :TRACe<n>:TYPE

### ➤ 命令格式:

:TRACe<n>:TYPE <WRITe|AVERAge|MAXHold|MINHold>

:TRACe<n>:TYPE?

### ➤ 功能描述:

选择指定迹线的迹线类型。

<n>: 迹线序列号, 整数, 取值范围 1-3。

WRITe: 刷新

AVERAge: 迹线平均

MAXHold: 最大保持

MINHold: 最小保持

### ➤ 返回格式:

查询返回指定迹线的迹线类型, WRITe、AVERAge、MAXHold 或 MINHold。

### ➤ 举例:

:TRACe1:TYPE AVERAge                    设置迹线 1 为迹线平均

:TRACe1:TYPE?                            查询返回 AVERAge

## :TRACe<n>:UPDate:STATe

### ➤ 命令格式:

:TRACe<n>:UPDate:STATe {{1|ON} | {0|OFF}}

:TRACe<n>:UPDate:STATe?

### ➤ 功能描述:

指定迹线刷新开关, ON 之后迹线持续刷新。

<n>: 迹线序列号, 整数, 取值范围 1-3。

### ➤ 返回格式:

查询返回指定迹线更新状态, 0 或 1。

### ➤ 举例:

:TRACe1:UPDate:STATe ON                迹线 1 打开迹线更新

:TRACe1:UPDate:STATe?                 查询返回 1

## :TRACe<n>:DISPlay[:STATe]

### ➤ 命令格式:

:TRACe<n>:DISPlay[:STATe] {{1|ON} | {0|OFF}}

:TRACe<n>:DISPlay[:STATe]?

### ➤ 功能描述:

指定迹线显示开关。

<n>: 迹线序列号, 整数, 取值范围 1-3。

### ➤ 返回格式:

查询返回指定迹线显示状态, 0 或 1。

### ➤ 举例:

:TRACe1:DISPlay ON            显示迹线 1  
:TRACe1:DISPlay?            查询返回 1

## TRIGger 命令

### :TRIGger[:SEquence]:SOURce

➤ **命令格式:**

:TRIGger[:SEquence]:SOURce {IMMEDIATE|EXTERNAL|VIDEO}  
:TRIGger[:SEquence]:SOURce?

➤ **功能描述:**

选择触发类型。

IMMEDIATE: 自由触发

EXTERNAL: 外触发

VIDEO: 视频触发

➤ **返回格式:**

查询返回触发类型, IMMEDIATE、EXTERNAL 或 VIDEO。

➤ **举例:**

:TRIGger:SOURce IMMEDIATE            选择自由触发  
:TRIGger:SOURce?                    查询返回 IMMEDIATE

### :TRIGger[:SEquence]:EXTERNAL:SLOPe

➤ **命令格式:**

:TRIGger[:SEquence]:EXTERNAL:SLOPe {POSITIVE|NEGATIVE}  
:TRIGger[:SEquence]:EXTERNAL:SLOPe?

➤ **功能描述:**

选择外部触发的触发边沿。

POSITIVE: 上升沿

NEGATIVE: 下降沿

➤ **返回格式:**

查询返回外部触发的触发边沿, POSITIVE 或 NEGATIVE。

➤ **举例:**

:TRIGger:EXTERNAL:SLOPe POSITIVE    选择外部触发的触发边沿为上升沿  
:TRIGger:EXTERNAL:SLOPe?            查询返回 POSITIVE

### :TRIGger[:SEquence]:EXTERNAL:DELAy

➤ **命令格式:**

:TRIGger[:SEquence]:EXTERNAL:DELAy <time>  
:TRIGger[:SEquence]:EXTERNAL:DELAy?

➤ **功能描述:**

设置外部触发的触发延时。

<time>: 连续正数, 默认单位 s, 范围 1000ps 到 500ms。

➤ **返回格式:**

查询返回外部触发的触发延时, 以科学记数法返回数据, 单位为 s。

➤ **举例:**

:TRIGger:EXTernal:DElay 0.01            设置外部触发的触发延时为 10ms

:TRIGger:EXTernal:DElay?                查询返回 1.000000e-02

**:TRIGger[:SEquence]:VIDeo:LEVel**

➤ **命令格式:**

:TRIGger[:SEquence]:VIDeo:LEVel <ampl>

:TRIGger[:SEquence]:VIDeo:LEVel?

➤ **功能描述:**

设置视频触发的触发电平。

<ampl>: 连续实数, 默认单位 dBm

➤ **返回格式:**

查询返回外部触发的触发边沿, 以科学记数法返回数据, 单位为 dBm。

➤ **举例:**

:TRIGger:VIDeo:LEVel -50                设置视频触发的触发电平为-50dBm

:TRIGger:VIDeo:LEVel?                查询返回-5.000000e+01

## UNIT 命令

**:UNIT:POWer**

➤ **命令格式:**

:UNIT:POWer {DBM|DBMV|DBUV|V|W}

:UNIT:POWer?

➤ **功能描述:**

选择 Y 轴刻度单位。

➤ **返回格式:**

查询返回 Y 轴刻度单位, DBM、DBMV、DBUV、V 或 W。

➤ **举例:**

:UNIT:POWer DBM                        选择 DBM 为 Y 轴刻度单位

:UNIT:POWer?                            查询返回 DBM

## 模拟解调

### CALCulate 命令

**:CALCulate:AM|FM:MARKer:SElect**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer:SElect <integer>

:CALCulate:AM|FM:MARKer:SElect?

➤ **功能描述:**

模拟解调不同解调模式下选择标记作为当前标记。

<integer>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

查询返回当前标记序列号, 范围 1 到 6。

➤ **举例:**

:CALCulate:AM:MARKer:SElect 1

调幅模式选择标记 1 为当前标记

:CALCulate:AM:MARKer:SElect?

查询返回 1

**:CALCulate:AM|FM:MARKer<n>:MODE**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:MODE {OFF|POSition|DELTA|FIXed}

:CALCulate:AM|FM:MARKer<n>:MODE?

➤ **功能描述:**

选择模拟解调不同解调模式下指定标记的模式。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

OFF: 关闭标记

POSition: 常规模式

DELTA: 差值模式

FIXed: 固定模式

➤ **返回格式:**

查询返回指定标记的模式, OFF、POSition、DELTA 或 FIXed。

➤ **举例:**

:CALCulate:AM:MARKer1:MODE POSition

选择调幅模式标记 1 为常规模式

:CALCulate:AM:MARKer1:MODE?

查询返回 POSition

**:CALCulate:AM|FM:MARKer<n>:TRACe**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:TRACe {RFSpectrum|DEMod|DAverage|DMAximum|DMINimum|AFSpectrum}

:CALCulate:AM|FM:MARKer<n>:TRACe?

➤ **功能描述:**

选择模拟解调不同解调模式下指定标记对应的迹线。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

RFSPectrum: 射频频谱迹线。

DEMod: 解调迹线。

DAverage: 解调平均迹线。

DMAximum: 解调最大保持迹线。

DMINimum: 解调最小保持迹线。

AFSPectrum: 调制信号频谱迹线。

➤ **返回格式:**

查询返回指定标记对应迹线的类型, RFSPectrum、DEMod、DAverage、DMAximum、DMINimum 或 AFSPectrum。

➤ **举例:**

:CALCulate:AM:MARKer1:TRACe DEMod      选择调幅模式标记 1 对应迹线为解调迹线

:CALCulate:AM:MARKer1:TRACe?      查询返回 DEMod

**:CALCulate:AM|FM:MARKer<n>:X**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:X <freq>|<time>

:CALCulate:AM|FM:MARKer<n>:X?

➤ **功能描述:**

调整模拟解调不同解调模式下指定标记的 X 轴读数。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

<freq>|<time>: 根据指定标记对应的不同迹线, X 轴读数为频率或者时间。指定标记对应迹线为射频频谱或调制信号频谱时, 为频率值, 默认单位 Hz; 标记迹线为解调、解调平均、解调最大保持或解调最小保持时, 为时间值, 默认单位为 s。

➤ **返回格式:**

查询返回指定标记的 X 轴坐标值, 以科学计数法返回数据。指定标记对应迹线为射频频谱或音频频谱时, 单位 Hz; 标记迹线为解调、解调平均、解调最大保持或解调最小保持时, 单位为 s。

➤ **举例:**

:CALCulate:AM:MARKer1:X 60MHz      设置调幅模式标记 1 的 X 轴坐标值为 60MHz

:CALCulate:AM:MARKer1:X?      查询返回 6.000000e+07

**:CALCulate:AM|FM:MARKer<n>:Y**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:Y <ampt>|<freq>|<depth>

:CALCulate:AM|FM:MARKer<n>:Y?

➤ **功能描述:**

调整模拟解调不同解调模式下指定标记的 Y 轴读数。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

<ampt>|<freq>|<depth>: 根据在不同解调模式, 标记对应的不同迹线, Y 轴读数为幅度、频率或调制深度百分比。当标记对应的迹线为射频频谱时, 在调幅和调频模式下, Y 轴读数均为幅度值, 默认单位 dBm。当标记对应的迹线为射频频谱之外其它迹线, 在调幅模式下, Y 轴读数为调制深度, 百分比值, 无单位, 1 对应 100%; 在调频模式下, Y 轴读数为频率值, 默认单位 Hz。

➤ **返回格式:**

查询返回指定标记的 Y 轴读数，以科学计数法返回数据。当标记对应的迹线为射频频谱时，Y 轴读数为幅度值，单位 dBm。当标记对应的迹线为射频频谱之外其它迹线时，在调幅模式下，Y 轴读数为调制深度，无单位，1 对应 100%；在调频模式下，Y 轴读数频率值，单位为 Hz。

➤ **举例:**

:CALCulate:AM:MARKer1:Y -50dBm                    设置调幅模式标记 1 的幅度值为-50dBm

:CALCulate:AM:MARKer1:Y?                            查询返回-5.000000e+01

**:CALCulate:AM|FM:MARKer<n>:REFerence**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:REFerence <integer>

:CALCulate:AM|FM:MARKer<n>:REFerence?

➤ **功能描述:**

选择模拟解调不同解调模式下指定标记的参考标记，参考标记不能为自身。

<n>: 标记序列号，连续整数，取值范围 1 到 6。

<integer>: 除自身外的其他标记序列号，取值范围 1 到 6。

➤ **返回格式:**

查询返回指定标记的参考标记。

➤ **举例:**

:CALCulate:AM:MARKer1:REFerence 2                设置调幅模式标记 2 为标记 1 的参考标记

:CALCulate:AM:MARKer1:REFerence?                查询返回 2

**:CALCulate:AM|FM:MARKer:AOFF**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer:AOFF

➤ **功能描述:**

模拟解调不同解调模式下关闭所有标记。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:AM:MARKer:AOFF                        调幅模式关闭所有标记

**:CALCulate:AM|FM:MARKer:TABLE[:STATe]**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer:TABLE[:STATe] {{1|ON} | {0|OFF}}

:CALCulate:AM|FM:MARKer:TABLE[:STATe]?

➤ **功能描述:**

模拟解调不同解调模式下标记列表显示开关。

1|ON: 显示

0|OFF: 不显示

- **返回格式:**  
查询返回标记列表显示状态, 0 或 1。
- **举例:**  
:CALCulate:AM:MARKer:TABLE:STATe ON 调幅模式显示标记列表  
:CALCulate:AM:MARKer:TABLE:STATe? 查询返回 1

#### :CALCulate:AM|FM:MARKer<n>:MAXimum

- **命令格式:**  
:CALCulate:AM|FM:MARKer<n>:MAXimum
- **功能描述:**  
模拟解调不同解调模式下对指定标记执行峰值搜索。  
<n>: 标记序列号, 连续整数, 取值范围 1-6
- **返回格式:**  
无返回值。
- **举例:**  
:CALCulate:AM:MARKer1:MAXimum 调幅模式标记 1 执行峰值搜索

#### :CALCulate:AM|FM:MARKer<n>:MAXimum:NEXT

- **命令格式:**  
:CALCulate:AM|FM:MARKer<n>:MAXimum:NEXT
- **功能描述:**  
模拟解调不同解调模式下对指定标记执行下一峰值搜索。  
<n>: 标记序列号, 连续整数, 取值范围 1 到 6。
- **返回格式:**  
无返回值。
- **举例:**  
:CALCulate:AM:MARKer1:MAXimum:NEXT 调幅模式标记 1 执行下一峰值搜索

#### :CALCulate:AM|FM:MARKer<n>:MAXimum:LEFT

- **命令格式:**  
:CALCulate:AM|FM:MARKer<n>:MAXimum:LEFT
- **功能描述:**  
模拟解调不同解调模式下对指定标记执行左侧下一峰值搜索。  
<n>: 标记序列号, 连续整数, 取值范围 1 到 6。
- **返回格式:**  
无返回值。
- **举例:**  
:CALCulate:AM:MARKer1:MAXimum:LEFT 调幅模式标记 1 执行左侧下一峰值搜索

### **:CALCulate:AM|FM:MARKer<n>:MAXimum:RIGHt**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:MAXimum:RIGHt

➤ **功能描述:**

模拟解调不同解调模式下对指定标记执行右侧下一峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:AM:MARKer1:MAXimum:RIGHt      调幅模式标记 1 执行右侧下一峰值搜索

### **:CALCulate:AM|FM:MARKer<n>:MINimum**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:MINimum

➤ **功能描述:**

模拟解调不同解调模式下对指定标记执行最小峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:AM:MARKer1:MINimum      调幅模式标记 1 执行最小峰值搜索

### **:CALCulate:AM|FM:MARKer<n>:PTPeak**

➤ **命令格式:**

:CALCulate:AM|FM:MARKer<n>:PTPeak

➤ **功能描述:**

模拟解调不同解调模式下对指定标记执行峰峰值搜索。

<n>: 标记序列号, 连续整数, 取值范围 1 到 6。

➤ **返回格式:**

无返回值。

➤ **举例:**

:CALCulate:AM:MARKer1:PTPeak      调幅模式标记 1 执行峰峰值搜索



解调模式切换到调频，测量设置参数恢复默认值。

➤ **返回格式：**

无返回值。

➤ **举例：**

:CONFigure:FM 解调模式切换到调频

**:CONFigure:FM:NDEFault**

➤ **命令格式：**

:CONFigure:FM

➤ **功能描述：**

解调模式切换到调频。

➤ **返回格式：**

无返回值。

➤ **举例：**

:CONFigure:FM 解调模式切换到调频

## DISPlay 命令

**:DISPlay:ZBmp?**

➤ **命令格式：**

:DISPlay:ZBmp?

➤ **功能描述：**

获取屏幕图像。

➤ **返回格式：**

查询返回屏幕图像数据，图片数据经过了压缩。

➤ **举例：**

:DISPlay:ZBmp? 获取屏幕图像

**:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:RLEVel**

➤ **命令格式：**

:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:RLEVel <real>

:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:RLEVel?

➤ **功能描述：**

调整模拟解调不同解调模式下各测量窗口的参考电平。

[1]：射频频谱窗口，<real>为幅度值，单位 dBm。

2：解调波形窗口，调幅模式下<real>为调制深度，百分比值，无单位，1 对应 100%；调频模式下<real>为频率值，单位 Hz。

3：调制频谱窗口，调幅模式下<real>为调制深度，百分比值，无单位，1 对应 100%；调频模式下<real>为频率值，单位 Hz。

➤ **返回格式:**

查询返回参考电平值，以科学记数法返回数据，。

➤ **举例:**

:DISPlay:AM:WINDow:TRACe:Y:RLEVel -10dBm 设置射频频谱窗口参考电平为-10dBm

:DISPlay:AM:WINDow:TRACe:Y:RLEVel? 查询返回-1.000000e+01

**:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:PDIVision**

➤ **命令格式:**

:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:PDIVision <real>

:DISPlay:AM|FM:WINDow[1]|2|3:TRACe:Y[:SCALe]:PDIVision?

➤ **功能描述:**

调整模拟解调不同解调模式下各测量窗口的 Y 轴刻度。

[1]: 射频扫描窗口, <real>为相对值, 默认单位 dB。

2: 解调波形窗口, 调幅模式下<real>为调制深度, 百分比值, 无单位, 1 对应 100%; 调频模式下<real>为频率值, 单位 Hz。

3: 调制频谱窗口, <real>为相对值, 默认单位 dB。

➤ **返回格式:**

查询返回 Y 轴刻度值, 以科学记数法返回数据。

➤ **举例:**

:DISPlay:AM|FM:WINDow:TRACe:Y:PDIVision 10dB 设置射频频谱窗口 Y 轴刻度为 10dB

:DISPlay:WINDow:TRACe:Y:PDIVision? 查询返回 1.000000e+01

**:DISPlay:AM|FM:WINDow3:TRACe:Y[:SCALe]:SPACing**

➤ **命令格式:**

:DISPlay:AM|FM:WINDow3:TRACe:Y[:SCALe]:SPACing {LOGarithmic|LINear}

:DISPlay:AM|FM:WINDow3:TRACe:Y[:SCALe]:SPACing?

➤ **功能描述:**

调整模拟解调不同解调模式下调制频谱窗口选择 Y 轴显示刻度类型。

LOGarithmic: 对数。

LINear: 线性。

➤ **返回格式:**

查询返回 Y 轴显示刻度类型, LOGarithmic 或 LINear。

➤ **举例:**

:DISPlay:AM:WINDow3:TRACe:Y:SPACing LOGarithmic 调制频谱 Y 轴刻度类型为对数

:DISPlay:AM:WINDow3:TRACe:Y:SPACing? 查询返回 LOGarithmic

## INITiate 命令

### :INITiate:CONTinuous

➤ **命令格式:**

:INITiate:CONTinuous {{1|ON} | {0|OFF}}  
:INITiate:CONTinuous?

➤ **功能描述:**

单次连续扫描切换。

1|ON: 连续扫描

0|OFF: 单次扫描

➤ **返回格式:**

查询返回是否为连续扫描, 0 或 1。

➤ **举例:**

:INITiate:CONTinuous ON                      连续扫描  
:INITiate:CONTinuous?                      查询返回 1

## MMEMemory 命令

### :MMEMemory:LOAD:STATe

➤ **命令格式:**

:MMEMemory:LOAD:STATe <filename>

➤ **功能描述:**

加载寄存器状态数据。

<filename>: 文件名, 文件后缀 .state

➤ **返回格式:**

无返回。

**举例:**

:MMEMemory:LOAD:STATe "test.state"                      加载寄存器状态文件 test.state

### :MMEMemory:STORe:STATe

➤ **命令格式:**

:MMEMemory:STORe:STATe <filename>

➤ **功能描述:**

保存寄存器状态到文件。

<filename>: 文件名, 文件后缀 .state

➤ **返回格式:**

无返回。

**举例:**

:MMEMemory:STORe:STATe "test.state"                      保存寄存器状态到文件 test.state

## SENSe 命令

### [ :SENSe ] : AM | FM : AFSPectrum : FREQuency : START

➤ **命令格式:**

[ :SENSe ] : AM | FM : AFSPectrum : FREQuency : START <freq>

[ :SENSe ] : AM | FM : AFSPectrum : FREQuency : START?

➤ **功能描述:**

模拟解调不同解调模式下调制频谱扫描设置扫频起始频率。

<freq>: 连续实数, 默认单位为 Hz。起始频率范围是 0Hz 到 100MHz-100Hz。

➤ **返回格式:**

查询返回起始频率值, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:AM:AFSPectrum:FREQuency:START 10MHz 调幅模式设置调制频谱起始频率为 10MHz

:AM:AFSPectrum:FREQuency:START? 查询返回 1.000000e+07

### [ :SENSe ] : AM | FM : AFSPectrum : FREQuency : STOP

➤ **命令格式:**

[ :SENSe ] : AM | FM : AFSPectrum : FREQuency : STOP <freq>

[ :SENSe ] : AM | FM : AFSPectrum : FREQuency : STOP?

➤ **功能描述:**

模拟解调不同解调模式下调制频谱扫描设置扫频截止频率。

<freq>: 连续实数, 默认单位为 Hz。截止频率范围是 100Hz 到 100MHz。

➤ **返回格式:**

查询返回截止频率值, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:AM:AFSPectrum:FREQuency:STOP 20MHz 调幅模式设置调制频谱截止频率为 20MHz

:AM:AFSPectrum:FREQuency:STOP? 查询返回 2.000000e+07

### [ :SENSe ] : AM | FM : FREQuency : SPAN

➤ **命令格式:**

[ :SENSe ] : AM | FM : FREQuency : SPAN <freq>

[ :SENSe ] : AM | FM : FREQuency : SPAN?

➤ **功能描述:**

模拟解调不同解调模式下射频频谱扫描设置扫宽。

<freq>: 连续实数, 默认单位为 Hz。扫宽范围是 100Hz 到 12.8MHz。

➤ **返回格式:**

查询返回扫宽值, 以科学计数法返回数据, 单位 Hz。

➤ **举例:**

:AM:FREQuency:SPAN 1MHz 调幅模式设置射频频谱扫宽为 1MHz

:AM:FREQuency:SPAN? 查询返回 1.000000e+06

## **[ :SENSe ] :FREQuency :CENTer**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :CENTer <freq>

[ :SENSe ] :FREQuency :CENTer?

### ➤ **功能描述:**

模拟解调设置扫频中心频率。

<freq>: 连续实数, 默认单位为 Hz。频率范围是 50Hz 到最大频率-50Hz。

UTS1015B 的最大频率是 1.5GHz, UTS3036B 的最大频率是 3.6GHz。

### ➤ **返回格式:**

查询返回中心频率值, 以科学计数法返回数据, 单位为 Hz。

### ➤ **举例:**

:FREQuency :CENTer 1GHz      设置扫频中心频率为 1GHz

:FREQuency :CENTer?          查询返回 1.000000e+09

## **[ :SENSe ] :FREQuency :CENTer :STEP [ :INCRement ]**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :CENTer :STEP [ :INCRement ] <freq>

[ :SENSe ] :FREQuency :CENTer :STEP [ :INCRement ]?

### ➤ **功能描述:**

模拟解调调整中心频率步进。

<freq>: 连续实数, 默认单位为 Hz。

### ➤ **返回格式:**

查询返回中心频率步进值, 以科学计数法返回数据, 单位为 Hz。

### ➤ **举例:**

:FREQuency :CENTer :STEP 10MHz    设置中心频率步进为 10MHz

:FREQuency :CENTer :STEP?          查询返回 1.000000e+07

## **[ :SENSe ] :FREQuency :CENTer :STEP :AUTO**

### ➤ **命令格式:**

[ :SENSe ] :FREQuency :CENTer :STEP :AUTO {{1|ON} | {0|OFF}}

[ :SENSe ] :FREQuency :CENTer :STEP :AUTO?

### ➤ **功能描述:**

模拟解调手动/自动中心频率步进切换。

1|ON: 自动

0|OFF: 手动

### ➤ **返回格式:**

查询返回自动中心频率步进状态, 0 或 1。

### ➤ **举例:**

:FREQuency :CENTer :STEP :AUTO ON      设置自动中心频率步进

:FREQuency :CENTer :STEP :AUTO?      查询返回 1



**[[:SENSe]:CORRection:IMPedance[:INPut]][:MAGNitude]**

**:INPut:IMPedance**

➤ **命令格式:**

```
[[:SENSe]:CORRection:IMPedance[:INPut]][:MAGNitude] {50|75}
[:SENSe]:CORRection:IMPedance[:INPut]][:MAGNitude]?
:INPut:IMPedance {50|75}
:INPut:IMPedance?
```

➤ **功能描述:**

选择输入阻抗，50 Ω 或 75 Ω。

➤ **返回格式:**

查询返回输入阻抗值，50 或 75。

➤ **举例:**

:CORRection:IMPedance 50	设置输入阻抗为 50 Ω
:CORRection:IMPedance?	查询返回 50
:INPut:IMPedance 50	设置输入阻抗为 50 Ω
:INPut:IMPedance?	查询返回 50

**[[:SENSe]:AM|FM:BANDwidth[:RESolution]**

➤ **命令格式:**

```
[[:SENSe]:AM|FM:BANDwidth[:RESolution] <freq>
[:SENSe]:AM|FM:BANDwidth[:RESolution]?
```

➤ **功能描述:**

模拟解调不同解调模式下调整射频频谱扫描的分辨率带宽。

<freq>: 离散实数，默认单位为 Hz，取值范围 100Hz 到 1MHz，以 1-1.5-2-3-5-7.5-10 为步进。

➤ **返回格式:**

查询返回分辨率带宽值，以科学计数法返回数据，单位为 Hz。

➤ **举例:**

:AM:BANDwidth 1MHz	调幅模式设置分辨率带宽为 1MHz
:AM:BANDwidth?	查询返回 1.000000e+06

**[[:SENSe]:AM|FM:BANDwidth[:RESolution]:AUTO**

➤ **命令格式:**

```
[[:SENSe]:AM|FM:BANDwidth[:RESolution]:AUTO {{1|ON} | {0|OFF}}
[:SENSe]:AM|FM:BANDwidth[:RESolution]:AUTO?
```

➤ **功能描述:**

模拟解调不同解调模式下射频频谱扫描自动/手动分辨率带宽切换。

1|ON: 自动

0|OFF: 手动

➤ **返回格式:**



➤ **返回格式:**

查询返回解调带宽值, 以科学计数法返回数据, 单位为 Hz。

➤ **举例:**

:AM:BANDwidth:CHANnel 1MHz

调幅模式设置解调带宽为 1MHz

:AM:BANDwidth:CHANnel?

查询返回 1.000000e+06

**[ :SENSe ] : AM | FM : DWSweep : TIME**

➤ **命令格式:**

[ :SENSe ] : AM | FM : DWSweep : TIME <time>

[ :SENSe ] : AM | FM : DWSweep : TIME ?

➤ **功能描述:**

模拟解调不同解调模式下设置解调窗口扫描时间。

<time>: 时间值, 单位为 s, 取值范围 1 μs 到 500ms。

➤ **返回格式:**

查询返回解调窗口扫描时间, 以科学计数法返回数据, 单位为 s。

➤ **举例:**

:AM:DWSweep:TIME 0.1

调幅模式设置解调窗口扫描时间为 100ms

:AM:DWSweep:TIME?

查询返回 1.000000e-01

**[ :SENSe ] : AM | FM : DEMod : TIME : AUTO**

➤ **命令格式:**

[ :SENSe ] : AM | FM : DEMod : TIME : AUTO { {1|ON} | {0|OFF} }

[ :SENSe ] : AM | FM : DEMod : TIME : AUTO ?

➤ **功能描述:**

模拟解调不同解调模式下自动/手动解调时间切换。

1|ON: 自动

0|OFF: 手动

➤ **返回格式:**

查询返回扫描时间状态, 0 或 1。

➤ **举例:**

:AM:DEMod:TIME:AUTO ON

调幅模式自动解调时间

:AM:DEMod:TIME:AUTO?

查询返回 1

**[ :SENSe ] : AM | FM : DEMod : TIME**

➤ **命令格式:**

[ :SENSe ] : AM | FM : DEMod : TIME <time>

[ :SENSe ] : AM | FM : DEMod : TIME ?

➤ **功能描述:**

模拟解调不同解调模式下设置解调时间。

<time>: 时间值, 单位为 s, 取值范围 1 μs 到 500ms。

➤ **返回格式:**

查询返回解调时间，以科学计数法返回数据，单位为 s。

➤ **举例:**

:AM:DEMod:TIME 0.1

调幅模式设置解调时间为 100ms

:AM:DEMod:TIME?

查询返回 1.000000e-01

### [ :SENSe ] : AM | FM : AVERAge [ :STATe ]

➤ **命令格式:**

[ :SENSe ] : AM | FM : AVERAge [ :STATe ] { { 1 | ON } | { 0 | OFF } }

[ :SENSe ] : AM | FM : AVERAge [ :STATe ] ?

➤ **功能描述:**

模拟解调不同解调模式下平均开关。

1 | ON: 开

0 | OFF: 关

➤ **返回格式:**

查询返回扫描时间状态，0 或 1。

➤ **举例:**

:AM:AVERAge ON

调幅模式平均打开

:AM:AVERAge?

查询返回 1

### [ :SENSe ] : AM | FM : AVERAge : COUNT

➤ **命令格式:**

[ :SENSe ] : AM | FM | : AVERAge : COUNT < integer >

[ :SENSe ] : AM | FM | PM | FMSTereo : AVERAge : COUNT?

➤ **功能描述:**

模拟解调不同解调模式下设置平均次数。

< integer >: 平均次数，连续整数，取值范围 1-999。

➤ **返回格式:**

查询返回平均次数。

➤ **举例:**

:AM:AVERAge:COUNT 100

调幅模式设置平均次数为 100

:AM:AVERAge:COUNT?

查询返回 100

### [ :SENSe ] : SPEaker [ :STATe ]

➤ **命令格式:**

[ :SENSe ] : SPEaker [ :STATe ] { { 1 | ON } | { 0 | OFF } }

[ :SENSe ] : SPEaker [ :STATe ] ?

➤ **功能描述:**

音频播放开关。

1 | ON: 打开

0|OFF: 关闭

➤ **返回格式:**

查询返回音频播放开关状态, 0 或 1。

➤ **举例:**

:SPEaker ON

打开音频播放

:SPEaker?

查询返回 1

**[[:SENSe]:AM|FM:LPFilter**

➤ **命令格式:**

[[:SENSe]:AM|FM:LPFilter {OFF|LPF300|LPF3K|LPF15K|LPF30K|LPF80K|LPF100K|LPF300K}

[[:SENSe]:AM|FM:LPFilter?

➤ **功能描述:**

模拟解调不同解调模式下选择低通滤波器。

OFF: 关

LPF300: 300Hz

LPF3K: 3kHz

LPF15K: 15kHz

LPF30K: 30kHz

LPF80K: 80kHz

LPF100K: 100kHz

LPF300K: 300kHz

➤ **返回格式:**

查询返回低通滤波器类型, OFF、LPF300、LPF3K、LPF15K、LPF30K、LPF80K、LPF100K 或 LPF300K。

➤ **举例:**

:AM:LPFilter LPF3K

调幅模式选择 3kHz 低通滤波器

:AM:LPFilter?

查询返回 LPF3K

**[[:SENSe]:AM|FM:HPFilter**

➤ **命令格式:**

[[:SENSe]:AM|FM:HPFilter {OFF|HPF20|HPF50|HPF300|HPF400}

[[:SENSe]:AM|FM:HPFilter?

➤ **功能描述:**

模拟解调不同解调模式下选择高通滤波器。

OFF: 关

HPF20: 20Hz

HPF50: 50Hz

HPF300: 300Hz

HPF400: 400Hz

➤ **返回格式:**

查询返回高通滤波器类型, OFF、HPF20、HPF50、HPF300 或 HPF400。

➤ **举例:**

:AM:HPFilter HPF50

调幅模式选择 50Hz 高通滤波器

:AM:HPFilter?

查询返回 HPF50

### [[:SENSe]:AM|FM:BPFiLter

#### ➤ 命令格式:

[[:SENSe]:AM|FM:BPFiLter {OFF|AWEighting|CWEighting|CCIR1k|CCIR2k|CUNWeighting}

[[:SENSe]:AM|FM:BPFiLter?

#### ➤ 功能描述:

模拟解调不同解调模式下选择带通滤波器。

OFF: 关

AWEighting: A 记权

CWEighting: C 记权

CCIR1k: CCIR-1 记权

CCIR2k: CCIR-2 记权

CUNWeighting: CCIR 无记权

#### ➤ 返回格式:

查询返回带通滤波器类型, OFF、AWEighting、CWEighting、CCIR1k、CCIR2k 或 CUNWeighting。

#### ➤ 举例:

:AM:BPFiLter AWEighting

调幅模式选择 A 记权带通滤波器

:AM:BPFiLter?

查询返回 AWEighting

## 编程说明

描述在编程操作过程中可能出现的一些问题及解决方法。当您遇到如下这些问题时，请按照相应的说明进行处理。

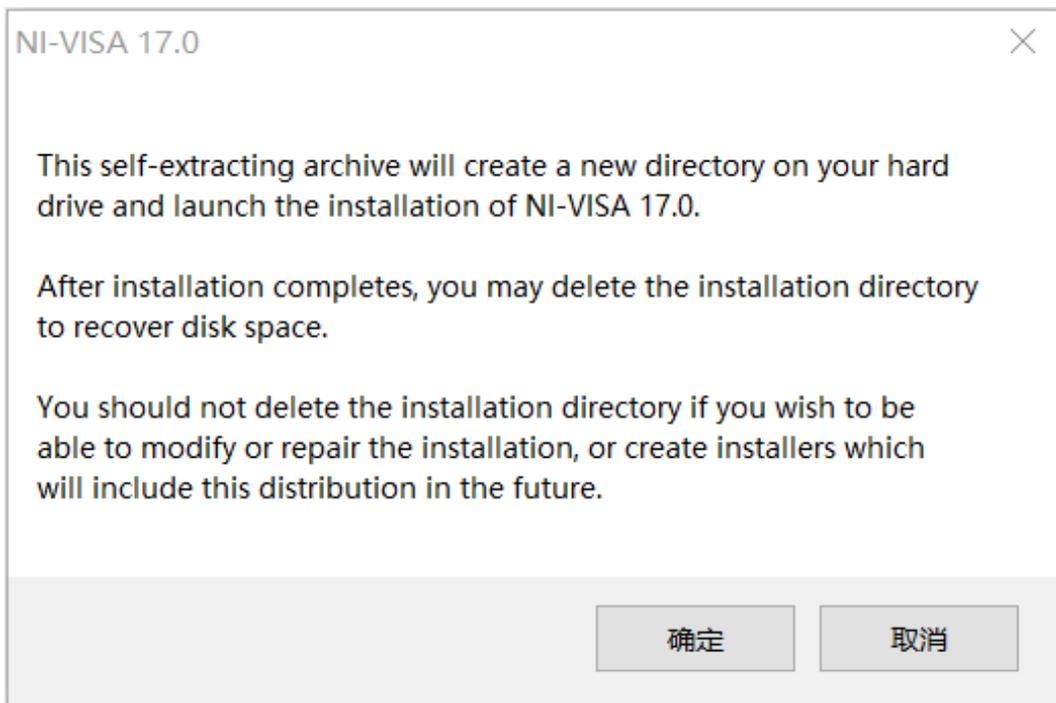
### 编程准备

用户可以通过使用频谱分析仪的 USB 或 LAN 端口，并结合 NI-VISA 和程序语言，远程控制频谱分析仪。编程工作仅适用于在 Windows 操作系统下使用 Visual Studio 和 LabVIEW 开发工具进行编程。

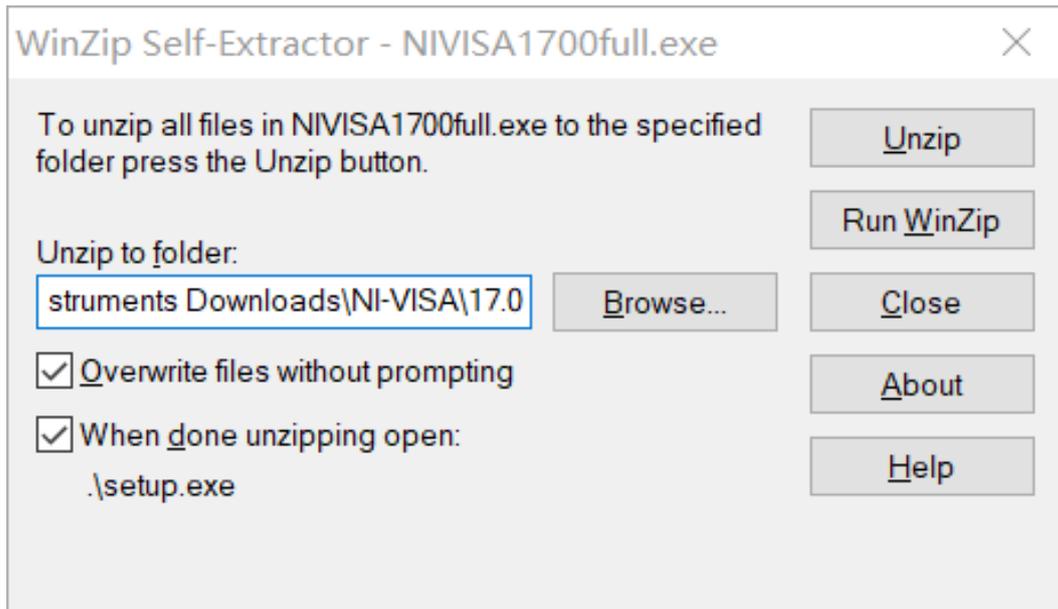
## 1. 建立通信

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-VISA17.0 完整版）：

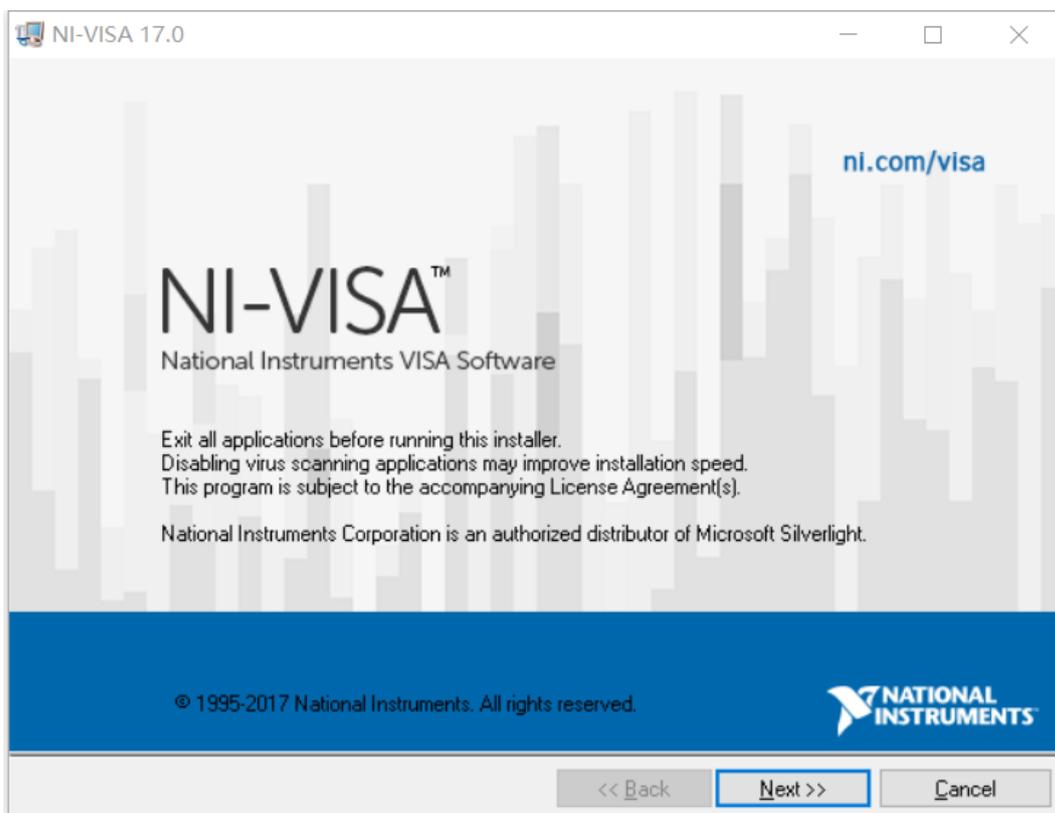
- a. 下载合适版本的 NI-VISA
- b. 双击 NIVISA1700full.exe，弹出对话框如下：



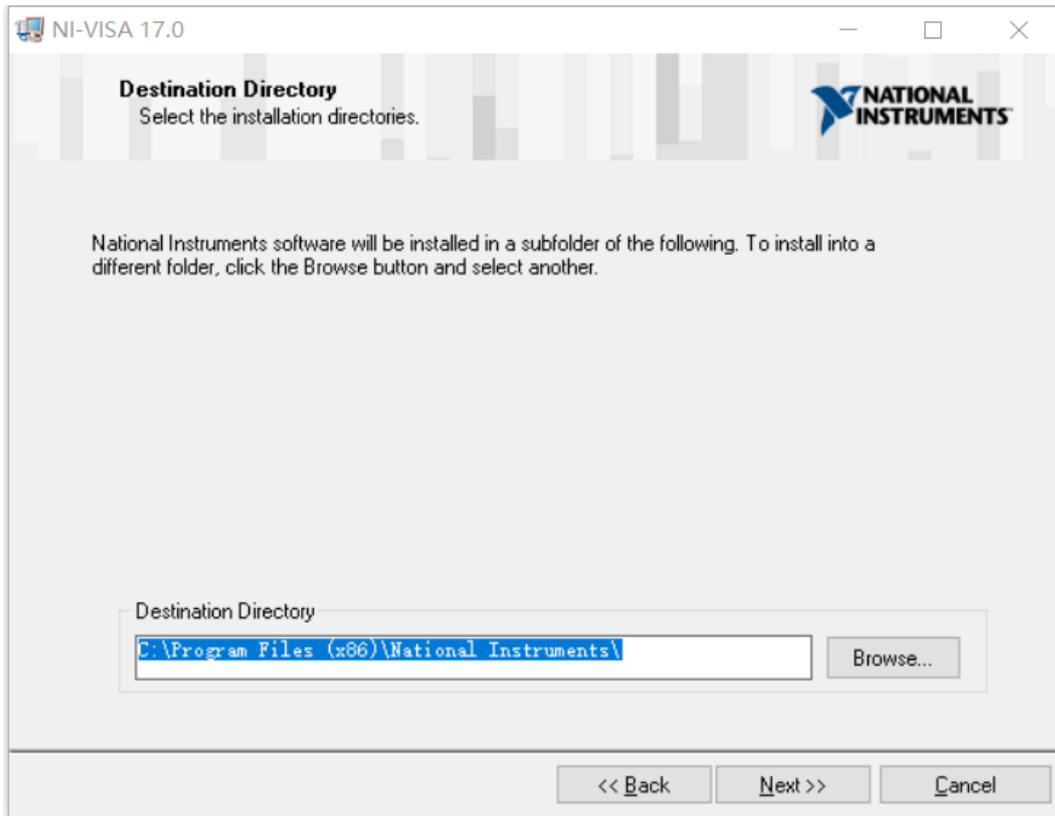
c. 点击确认弹出对话框如下：



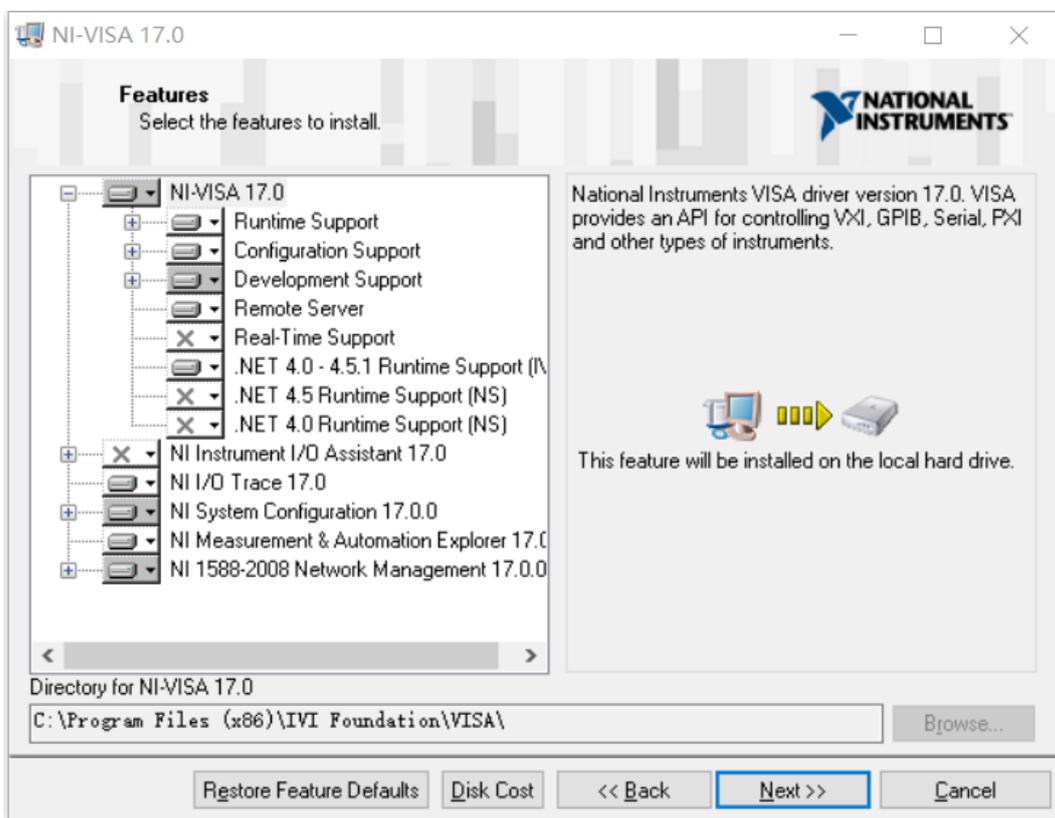
d. 点击 Unzip 解压文件，当解压完成后，安装程序将自动执行。若你的计算机需要安装 .NET Framework4 ，则在安装过程会自动安装 .NET Framework4 。



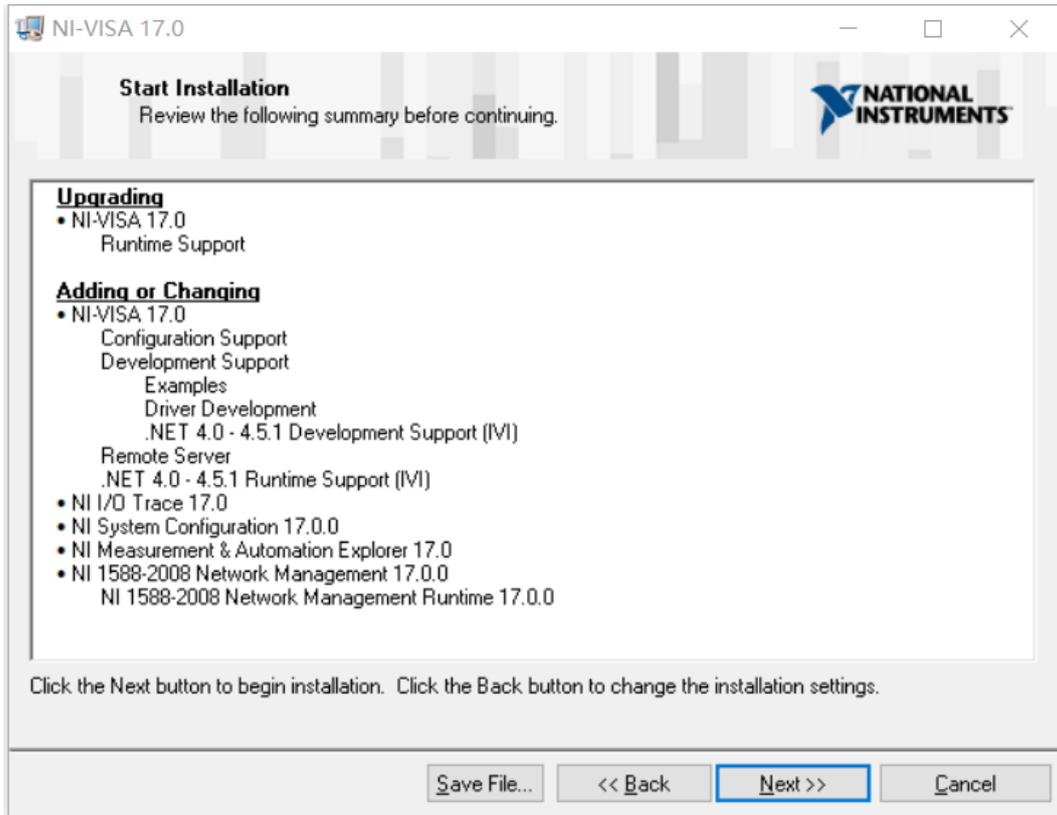
e. NI-VISA 安装对话框如上图所示，点击 Next 开始安装过程。



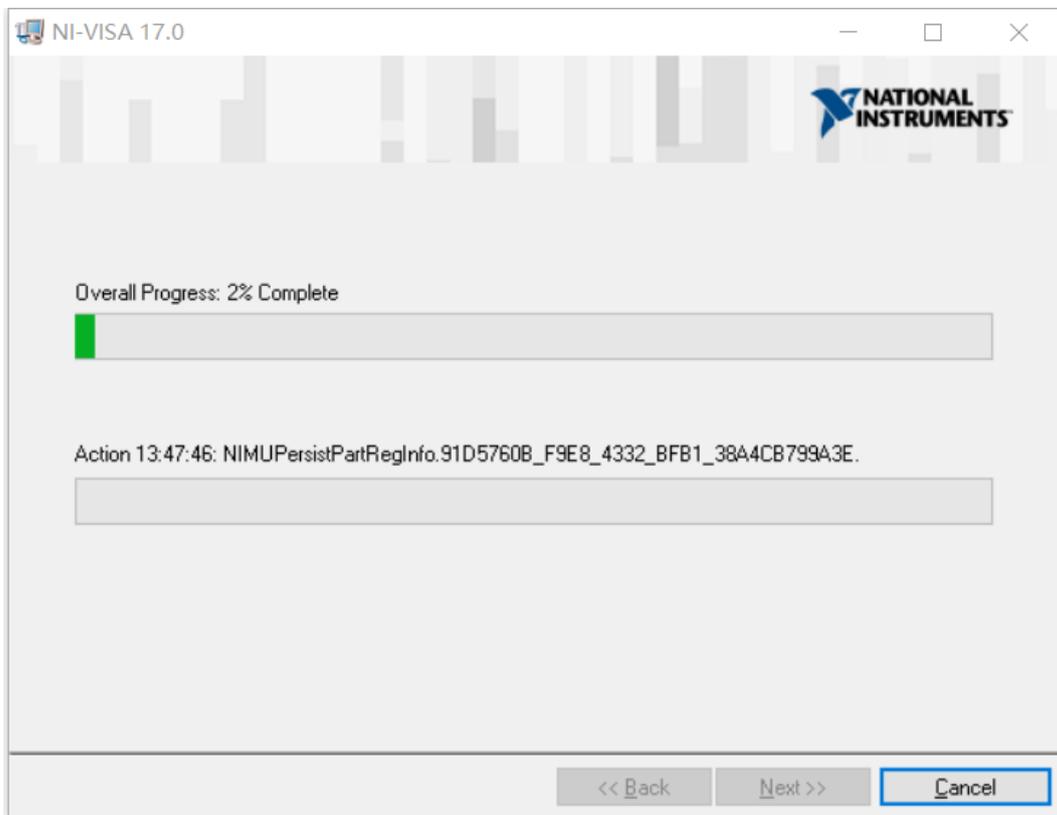
f. 设置安装路径，默认路径为“C:\Program Files (x86)\National Instruments\”。你也可以修改安装路径。点击 Next ，对话框如下图所示：



g. 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：



h. 点击 Next 开始安装：

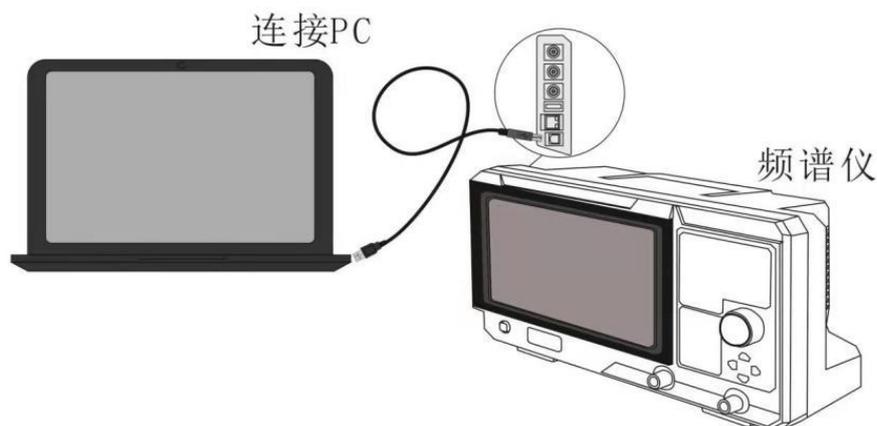


j. 安装完成后，重启电脑

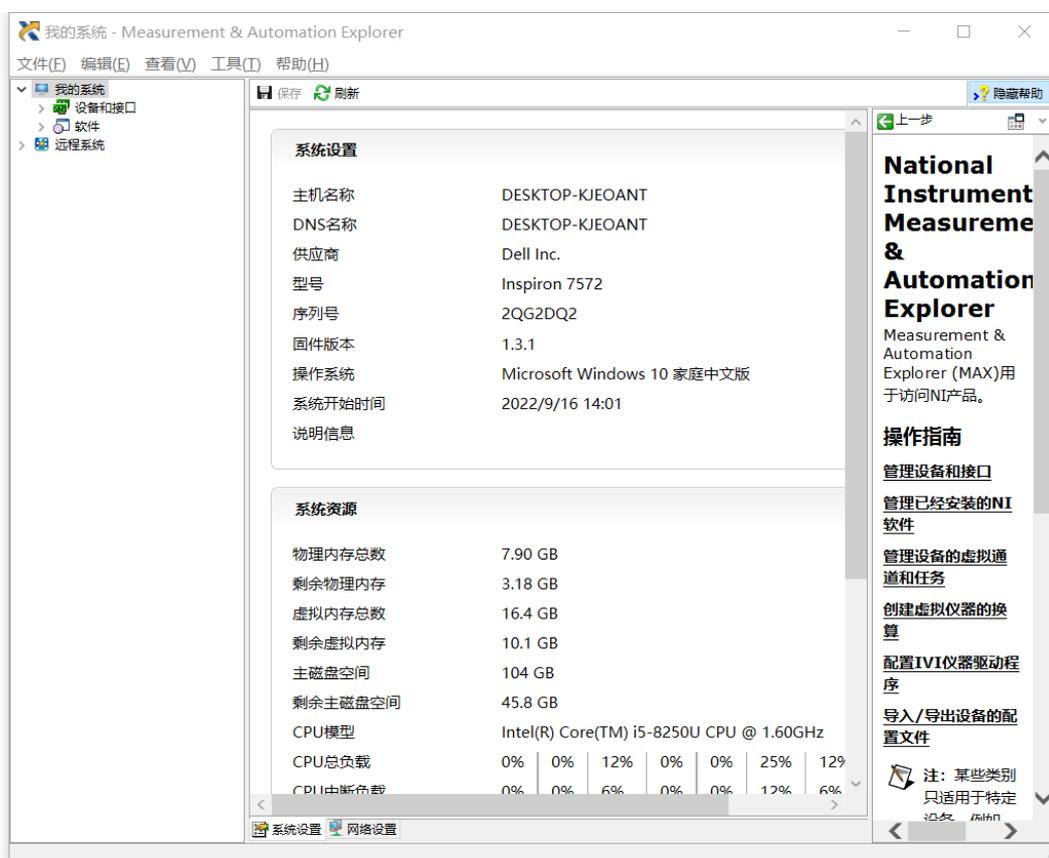
## 2. 连接仪器

以下通过 USB 方式连接进行介绍。

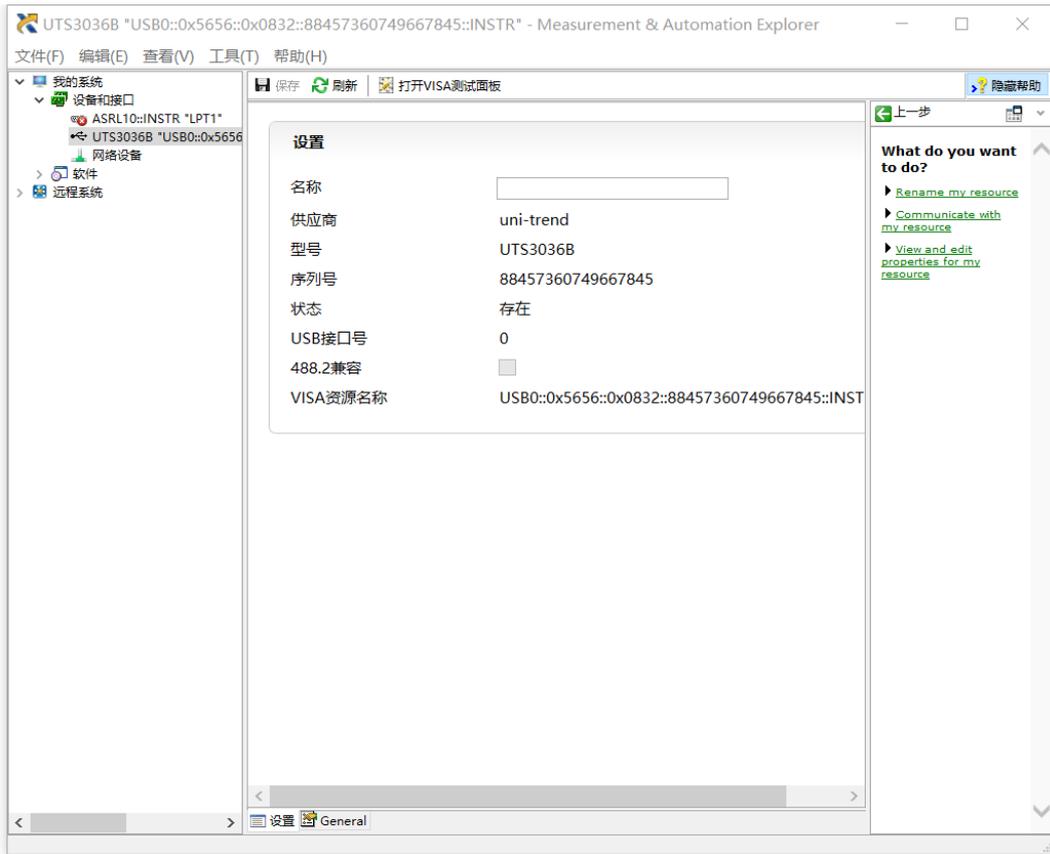
- 打开频谱分析仪；
- 使用 USB 线将频谱分析仪的 USB Device 端口和计算机的 USB Host 端口连接起来，如下图所示：



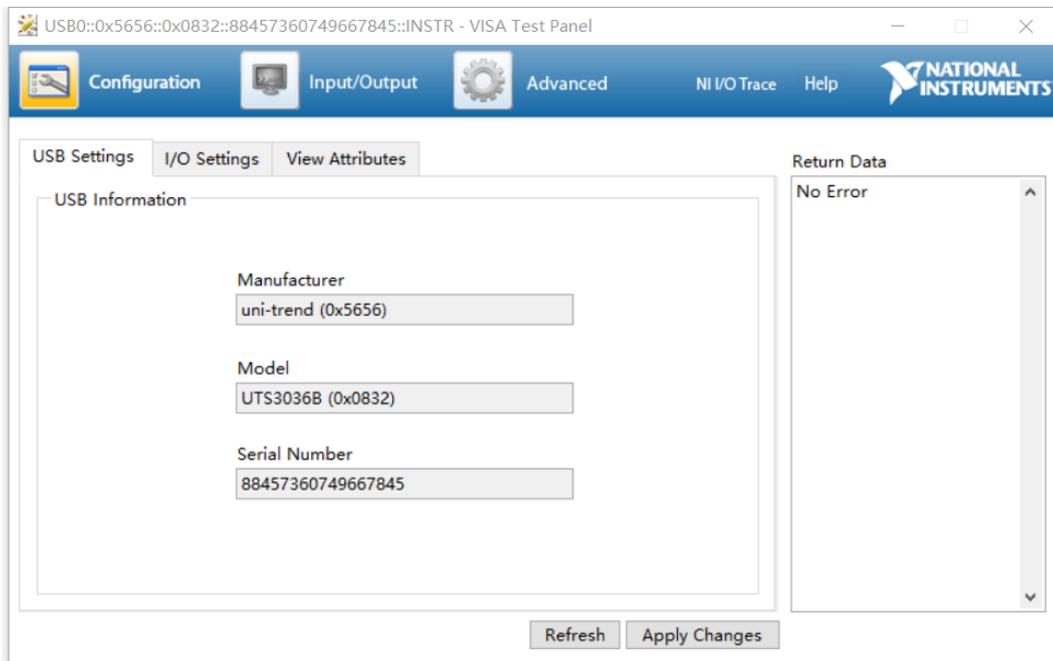
- 在计算机上打开 NI MAX，弹出如下对话框：



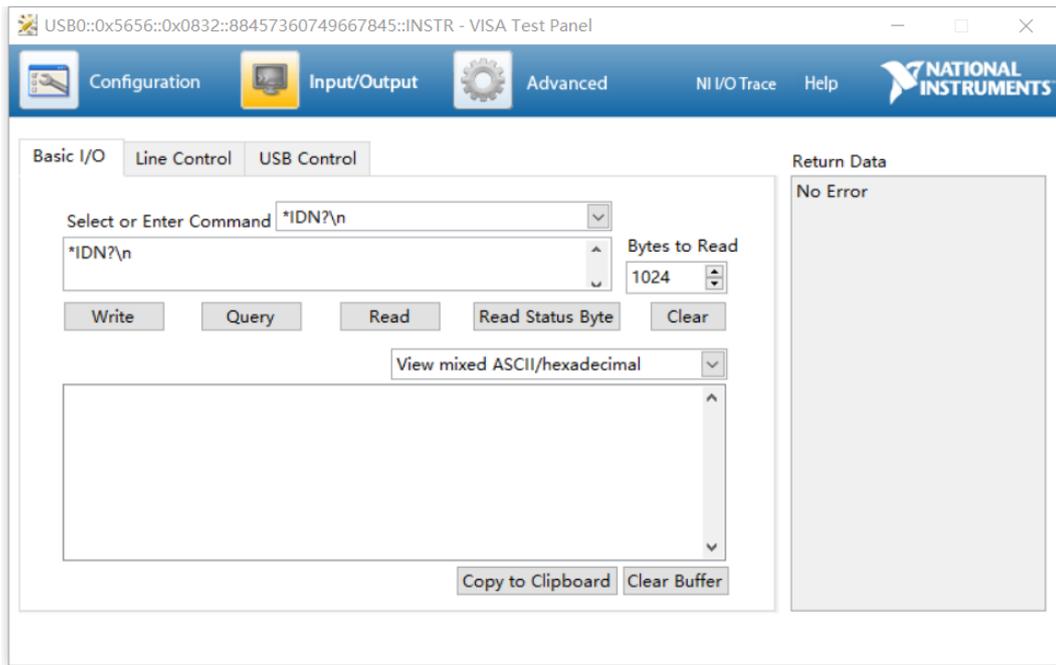
d. 打开设备和接口下拉选项，并选中频谱分析仪驱动，如下图所示：



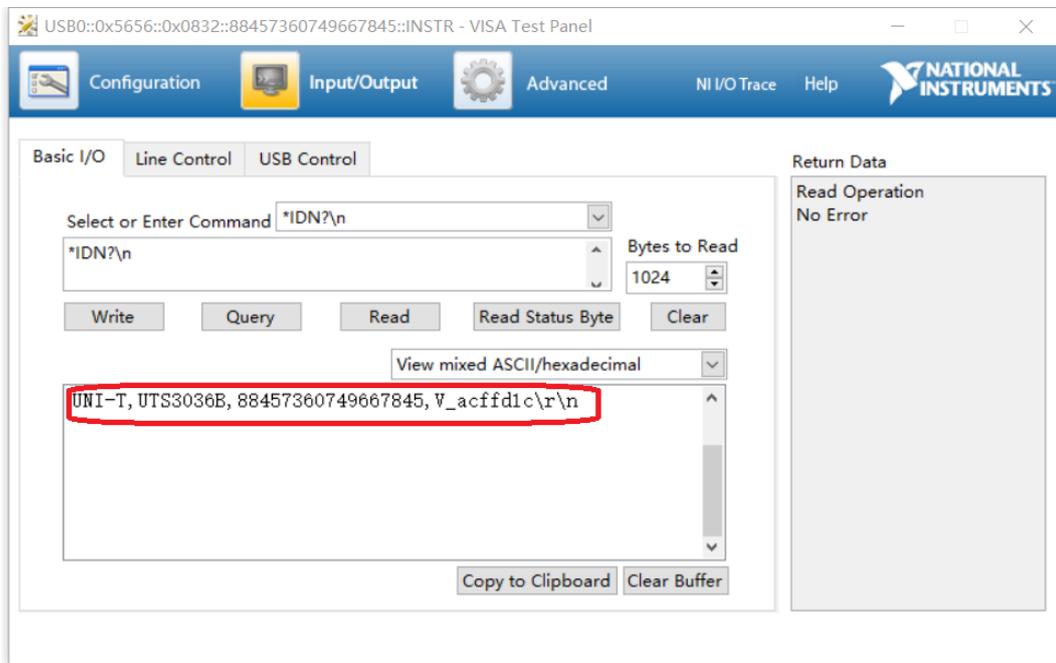
e. 鼠标点击 VISA 测试面板，弹出如下对话框：



f. 鼠标点击 Input/Output 选项，如图下图所示：



g. 鼠标点击 Query 按钮，查询频谱分析仪的 IDN，查询结果如下图红色区域所示：



h. 如果能够查询到频谱分析仪相关信息，就表示频谱分析仪已经可以与计算机通信

## VISA 编程示例

本节给出了一些编程示例。通过这些例子，可以了解如何使用 VISA，并结合编程手册的命令实现对仪器设备的控制。通过下面的例子，你可以开发更多应用。

### VC++示例

- 环境：Window 系统，Visual Studio。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，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，分别为静态库和动态库。

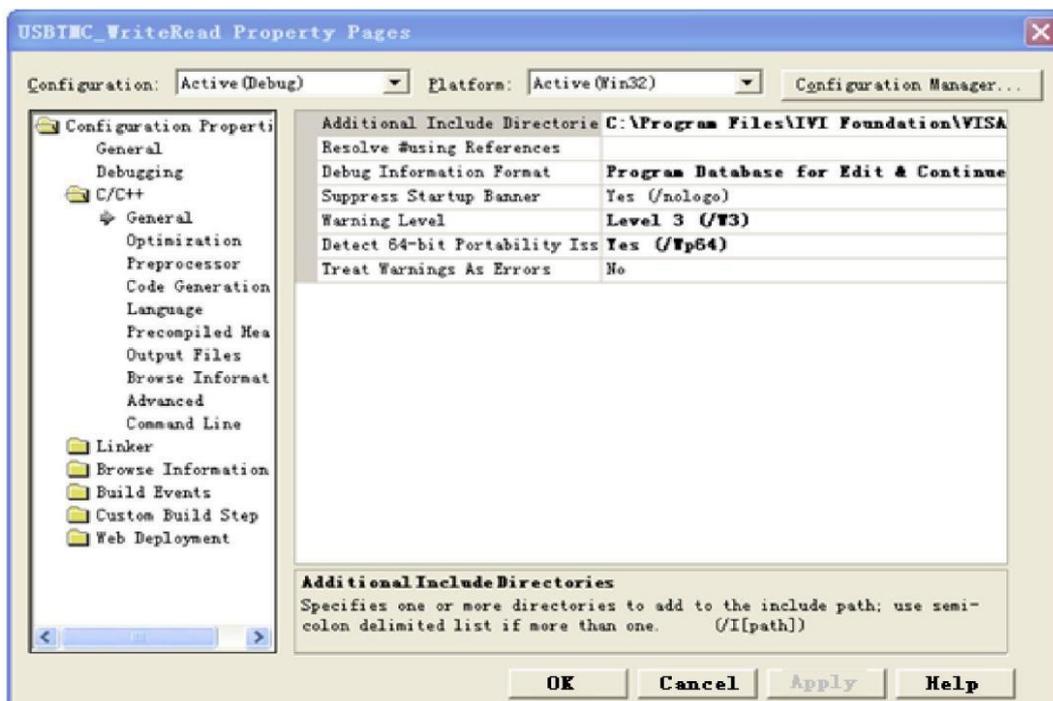
#### a) 静态库：

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：

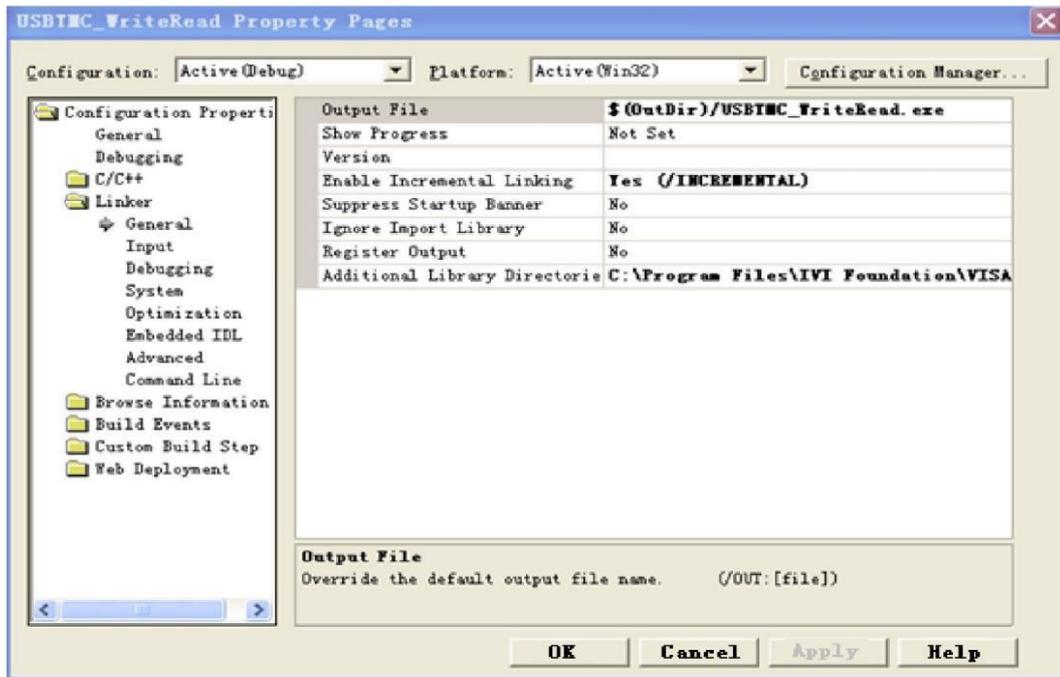
```
#include "visa.h"
#pragma comment(lib, "visa32.lib")
```

#### b) 动态库：

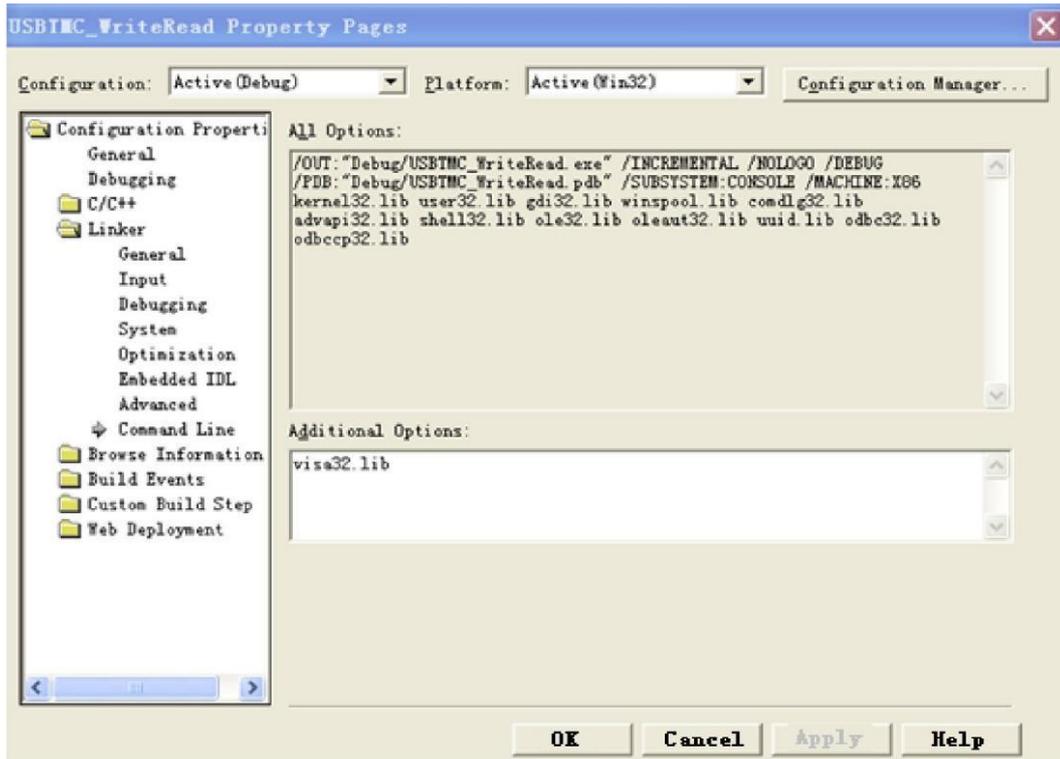
点击"project>>properties"，在属性对话框左侧选择"c/c++---General"中，将 "Additional Include Directories" 项 的 值 设 置 为 NI-VISA 的 安 装 路 径 ，（ 例 如 ： C:\ProgramFiles\IVI Foundation\VISA\WinNT\include），如下图所示：



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>
```

1. 源码:
  - a) USBTMC 示例
 

```
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.
    * 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_BUFLen];
unsigned char buffer[100];
int i;
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 * cmand = "*IDN?\n";
    status = viPrintf(instr, cmand);
    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;
}

```

## b) TCP/IP 示例

```

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[] = "::inst0::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("\nMessage 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;
}

```

## C#示例

- 环境：Window 系统, Visual Studio。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，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 C# console project。
  2. 添加 VISA 的 C#引用 Ivi.Visa.dll 和 NationalInstruments.Visa.dll。
  3. 源码：
    - a) USBTMC 示例

```

class Program
{
    void usbtmc_test()
    {
        using (var rmSession = new ResourceManager())
        {
            var resources = rmSession.Find("USB?*INSTR");
            foreach (string s in resources)
            {
                try
                {
                    var mbSession = (MessageBasedSession)rmSession.Open(s);
                    mbSession.RawIO.Write("*IDN?\n");
                    System.Console.WriteLine(mbSession.RawIO.ReadString());
                }
                catch (Exception ex)
            }
        }
    }
}

```

```

        {
            System.Console.WriteLine(ex.Message);
        }
    }
}

void Main(string[] args)
{
    usbtmc_test();
}
}

```

## b) TCP/IP 示例

```

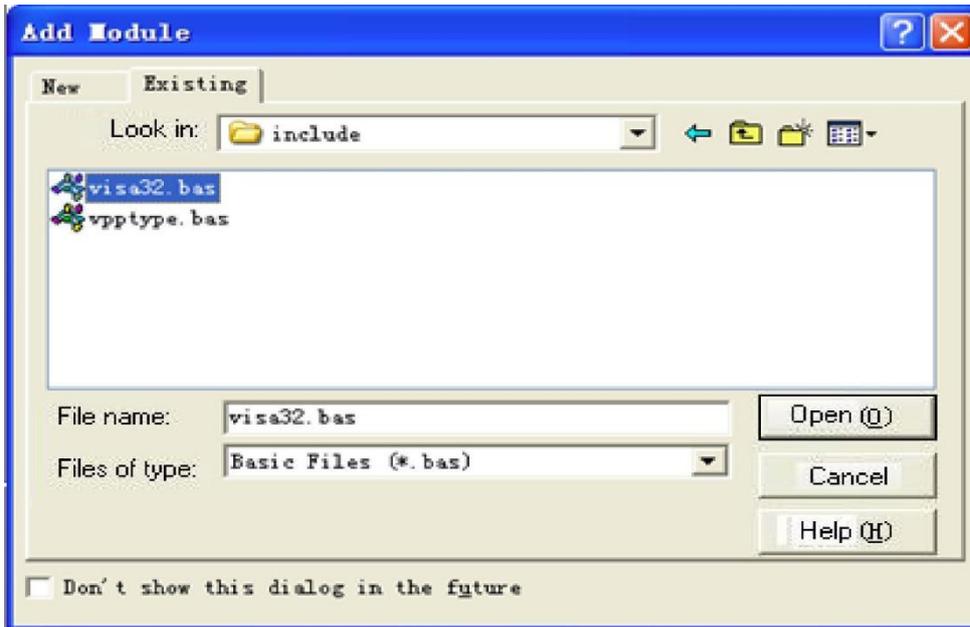
class Program
{
    void tcp_ip_test(string ip)
    {
        using (var rmSession = new ResourceManager())
        {
            try
            {
                var resource = string.Format("TCP/IP0::{0}::inst0::INSTR", ip);
                var mbSession = (MessageBasedSession)rmSession.Open(resource);
                mbSession.RawIO.Write("*IDN?\n");
                System.Console.WriteLine(mbSession.RawIO.ReadString());
            }
            catch (Exception ex)
            {
                System.Console.WriteLine(ex.Message);
            }
        }
    }

    void Main(string[] args)
    {
        tcp_ip_test("192.168.20.11");
    }
}

```

## VB 示例

- 环境：Window 系统，Microsoft Visual Basic 6.0。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，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。
  2. 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：



3. 源码：
  - a) USBTMC 示例

```
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_BUFLen
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 示例

```

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

```

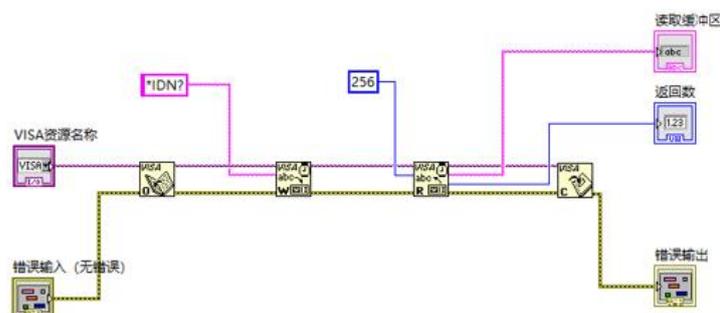
```

' Now we will open a session via TCP/IP device
status = viOpen(defaultRM, "TCPIP0::" + ip + "::inst0::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_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
EndIf
status = viClose(instrsesn)
status = viClose(defaultRM)
tcp_ip_test = 0
EndFunction

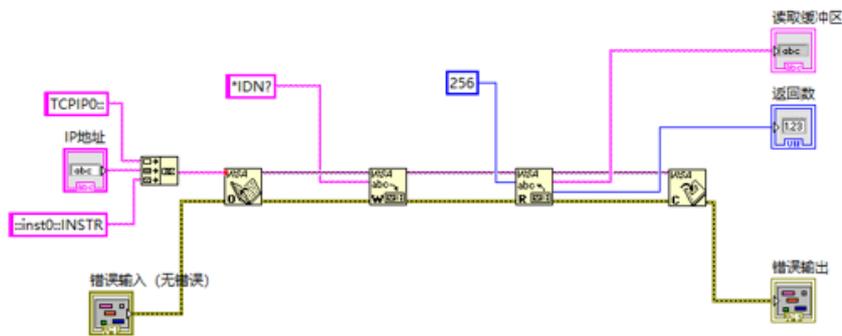
```

## LabVIEW 示例

- 环境：Window 系统，LabVIEW。
- 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，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 面板中选择和添加下列功能：VISA Write、VISA Read、VISA Open 和 VISA Close。
  4. VI 打开了一个 USBTMC 设备的 VISA 会话，并向设备写\*IDN?命令并回读的响应值。当所有通信完成时，VI 将关闭 VISA 会话，如下图所示：



5. 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 Mode>>Synchronous" 以实现同步写入或读取数据, 如下图所示:



## MATLAB 示例

- 环境: Window 系统, MATLAB。
- 描述: 通过 USBTMC 和 TCP/IP 访问仪器设备, 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, 点击在 Matlab 界面的 File>>New>>Script 创建一个空的 M 文件。

### 2. 源码:

#### a) USBTMC 示例

```
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::0x5345::0x1234::SN20220718::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
b) TCP/IP 示例
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', ['TCP/IP0::', '192.168.20.11', '::inst0::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

```

## Python 示例

- 环境：Window 系统，Python3.8，PyVISA 1.11.0。
  - 描述：通过 USBTMC 和 TCP/IP 访问仪器设备，并在 NI-VISA 上发送"\*IDN?"命令来查询设备信息。
  - 步骤：
    1. 首先安装 python，然后打开 Python 脚本编译软件，创建一个空的 test.py 文件。
    2. 使用 pip install PyVISA 指令安装 PyVISA，如无法安装，请参考此链接使用说明 (<https://pyvisa.readthedocs.io/en/latest/>)
    3. 源码：
- a) USBTMC 示例

```

import pyvisa

rm = pyvisa.ResourceManager()
rm.list_resources()

my_instrument = rm.open_resource('USB0::0x5345::0x1234::SN20220718::INSTR')

```

```
print(my_instrument.query('*IDN?'))
```

## b) TCP/IP 示例

```
import pyvisa  
rm = pyvisa.ResourceManager()  
rm.list_resources()  
my_instrument = rm.open_resource('TCP/IP0::192.168.20.11::inst0::INSTR')  
print(my_instrument.query('*IDN?'))
```

# 编程应用实例

本部分将介绍运用 SCPI 指令在频谱分析模式下如何对稳定信号进行精准测量。

## 1 信号源准备：

频谱分析仪的射频输入端口输入连续正弦波信号，信号频率 100MHz，信号功率-20dBm。

## 2 设置频谱分析仪参数：

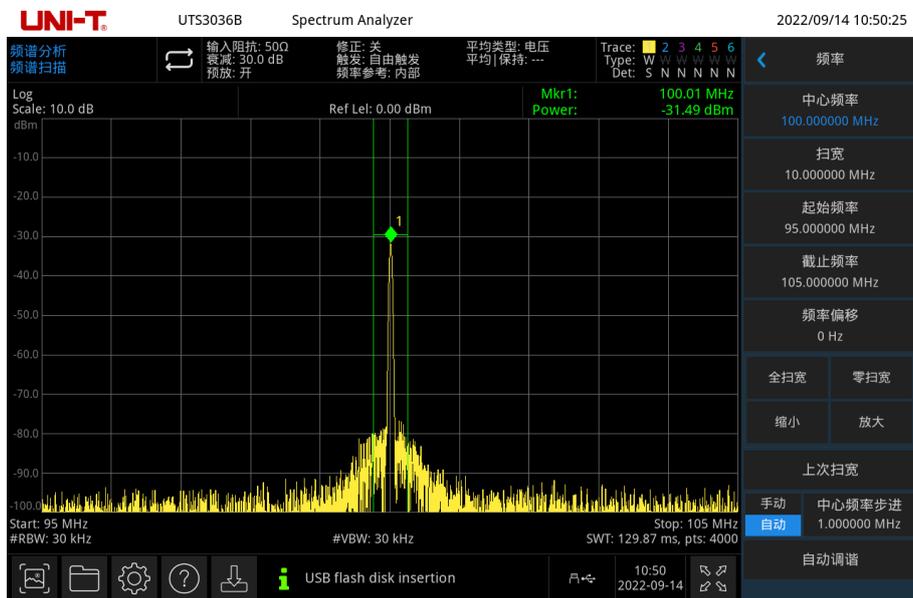
首先需要选择工作模式为频谱分析；

然后 default，让参数恢复到默认设置；

后续根据输入的信号设置中心频率、扫宽、分辨率带宽、参考电平、输入衰减、前置放大、显示刻度等参数，让信号在可视范围内，合理显示在屏幕中央；

再根据实际需求设置扫描时间、扫描模式、扫描点数、及迹线类型、迹线检波器类型等参数，以精确显示信号；

最后进行信号峰值搜索、标记测量等操作。



以下命令可进行如上所属的操作，获得上图所示精准测量。

```
:instrument:select SA //选择频谱分析工作模式  
:key:default //恢复默认设置  
:sense:frequency:center 100000000 //设置中心频率为 100MHz  
:sense:frequency:span 10000000 //设置扫宽 10MHz
```

```
:sense:bw:resolution 30000 //设置分辨率带宽 30kHz
:sense:bw:video 30000 //设置视频带宽 30kHz
:display:window:trace:y:scale:rlevel 0 //设置参考电平为 0
:sense:power:rf:attenuation:auto 1 //设置输入衰减自动
:sense:power:rf:gain:state 1 //打开前置放大
:display:window:trace:Y:scale:spacing LOG //显示刻度对数
:sense:sweep:time:auto 1 //扫描时间自动
:sense:sweep:type:auto 1 //扫描模式自动
:sense:sweep:points 4000 //扫描点数 4000
:trace1:mode write //迹线 1 迹线类型刷新
:detector:trace1 sample //迹线 1 检波器采样
:calculate:marker1:maximum //峰值搜索
:calculate:marker1:function bpower //打开标记功能带内功率
:calculate:marker1:function:band:span 500000 //设置带内宽度 500kHz, 然后频谱图右上角光标测量结果区域查看测量结果, 包含光标频率和带内功率。
```

## 附录 1: <key>列表

按键命令关键字	功能描述	LED 灯
STUp	测量设置	
Meas	测量	
Mode	模块	
FREQ	频率	
AMPT	幅度	
BW	带宽	
Sweep	扫描	
Trace	迹线	
Marker	标记	
Peak	峰值	
Auto	自动	
Save	保存	
System	系统设置	
Default	恢复出厂设置	
TG	跟踪源	√
Single	单次	√
TLOCK	触屏锁	√
UP	方向键上	
DOWN	方向键下	
LEFT	方向键左	
RIGHT	方向键右	
FKNLeft	旋钮左旋	
FKNRight	旋钮右旋	
NUM0	数字键 0	
NUM1	数字键 1	
NUM2	数字键 2	
NUM3	数字键 3	
NUM4	数字键 4	
NUM5	数字键 5	
NUM6	数字键 6	
NUM7	数字键 7	
NUM8	数字键 8	
NUM9	数字键 9	
DOT	数字键小数点	
SYMBOL	数字键符号	
ESC	退出	
BACKspace	退格	
Enter	回车	

## 附录 2：按键锁定状态

位序	按键	状态
0	Setup	0 表示未锁定, 1 表示锁定
1	Meas	0 表示未锁定, 1 表示锁定
2	Mode	0 表示未锁定, 1 表示锁定
3	FREQ	0 表示未锁定, 1 表示锁定
4	AMPT	0 表示未锁定, 1 表示锁定
5	BW	0 表示未锁定, 1 表示锁定
6	Sweep	0 表示未锁定, 1 表示锁定
7	Trace	0 表示未锁定, 1 表示锁定
8	Marker	0 表示未锁定, 1 表示锁定
9	Peak	0 表示未锁定, 1 表示锁定
10	Auto	0 表示未锁定, 1 表示锁定
11	Save	0 表示未锁定, 1 表示锁定
12	System	0 表示未锁定, 1 表示锁定
13	Default	0 表示未锁定, 1 表示锁定
14	TG	0 表示未锁定, 1 表示锁定
15	Single	0 表示未锁定, 1 表示锁定
16	TLOCK	0 表示未锁定, 1 表示锁定
17	UP	0 表示未锁定, 1 表示锁定
18	DOWN	0 表示未锁定, 1 表示锁定
19	LEFT	0 表示未锁定, 1 表示锁定
20	RIGHT	0 表示未锁定, 1 表示锁定
21	FKNLeft	0 表示未锁定, 1 表示锁定
22	FKNRight	0 表示未锁定, 1 表示锁定
23	NUM0	0 表示未锁定, 1 表示锁定
24	NUM1	0 表示未锁定, 1 表示锁定
25	NUM2	0 表示未锁定, 1 表示锁定
26	NUM3	0 表示未锁定, 1 表示锁定
27	NUM4	0 表示未锁定, 1 表示锁定
28	NUM5	0 表示未锁定, 1 表示锁定
29	NUM6	0 表示未锁定, 1 表示锁定
30	NUM7	0 表示未锁定, 1 表示锁定
31	NUM8	0 表示未锁定, 1 表示锁定
32	NUM9	0 表示未锁定, 1 表示锁定
33	DOT	0 表示未锁定, 1 表示锁定
34	SYMBOL	0 表示未锁定, 1 表示锁定
35	ESC	0 表示未锁定, 1 表示锁定
36	BACKspace	0 表示未锁定, 1 表示锁定
37	Enter	0 表示未锁定, 1 表示锁定