Ceyear 思仪

1433 系列

信号发生器

程控手册



中电科思仪科技股份有限公司

该手册适用下列型号信号发生器，基于固件版本 Version 1.0 及以上。

- 1433D 信号发生器（1MHz-20GHz）
- 1433D 信号发生器（1MHz-26.5GHz）
- 1433D 信号发生器（1MHz-40GHz）
- 1433D 信号发生器（1MHz-50GHz）

版　　本: C.2　2023年7月，中电科思仪科技股份有限公司
地　　址: 山东省青岛市黄岛区香江路98号
服务咨询: 0532-86889847　400-1684191
技术支持: 0532-86880796
质量监督: 0532-86886614
传　　真: 0532-86889056
网　　址: www.ceyear.com
电子信箱: techbb@ceyear.com
邮　　编: 266555

前　言

非常感谢您选择使用中电科思仪科技股份有限公司研制、生产的 1433 系列信号发生器！本所产品集高、精、尖于一体，在同类产品中有较高的性价比。

我们将以最大限度满足您的需求为己任，为您提供高品质的测量仪器，同时带给您一流的售后服务。我们的一贯宗旨是“质量优良，服务周到”，提供满意的产品和服务是我们对用户的承诺。

手册编号

AV2.827.1346SCCN

版本

C.2　2023.7

中电科思仪科技股份有限公司

手册授权

本手册中的内容如有变更，恕不另行通知。本手册内容及所用术语最终解释权属于中电科思仪科技股份有限公司。

本手册版权属于中电科思仪科技股份有限公司，任何单位或个人非经本所授权，不得对本手册内容进行修改，并且不得以赢利为目的对本手册进行复制、传播，中电科思仪科技股份有限公司保留对侵权者追究法律责任的权利。

产品质保

本产品从出厂之日起保修期为 18 个月。质保期内仪器生产厂家会根据用户要求及实际情况维修或替换损坏部件。具体维修操作事宜以合同为准。

产品质量证明

本产品从出厂之日起确保满足手册中的指标。校准测量由具备国家资质的计量单位予以完成，并提供相关资料以备用户查阅。

质量/环境管理

本产品从研发、制造和测试过程中均遵守质量和环境管理体系。中电科思仪科技股份有限公司已经具备资质并通过 ISO 9001 和 ISO 14001 管理体系。

安全事项

注　意

注意标识代表重要的信息提示，但不会导致危险。它提示用户注意某一操作过程、操作方法或者类似情况。若不能遵守规则或者正确操作，则可能引起的仪器损坏或丢失重要数据。在完全理解和满足所指出的注意条件之后，才可继续下一步。

提　示

提示标识代表信息提示。它提示用户注意仪器或者某一操作过程、操作方法或者类似情况。以引导仪器操作者正确的使用仪器。

目 录

程控手册 I

1 手册导航 1

1.1 关于手册 1
1.2 关联文档 2

2 远程控制 3

2.1 远程控制基础 3
2.1.1 程控接口 3
2.1.2 消息 5
2.1.3 SCPI 命令 6
2.1.4 命令序列与同步 14
2.1.5 状态报告系统 16
2.1.6 编程注意事项 24
2.2 仪器程控端口与配置 25
2.2.1 LAN 25
2.2.2 USB 26
2.3 I/O 库 27
2.3.1 I/O 库概述 27
2.3.2 I/O 库安装与配置 28

3 程控命令 31

3.1 命令说明 31
3.2 通用命令 31
*IDN? 32
*RST 32
3.3 仪器子系统命令 32
3.3.1 System 子系统 32
3.3.1 Frequency 子系统 39
3.3.1 Power 子系统 41

目录

3.3.4 Modulation 子系统 49
3.3.5 Sweep 系统 56
3.3.6 其他命令 65
3.4 仪器兼容命令 66
4 编程示例 69
4.1 基本操作示例 69
4.1.1 VISA 库 69
4.1.2 示例运行环境 70
4.1.3 初始化和设置默认状态 70
4.1.4 发送设置命令 71
4.1.5 读取测量仪器状态 72
4.2 高级操作示例 72
4.2.1 LAN 接口设置点频并查询 73
4.2.2 LAN 接口设置点频并查询（Visa） 74
4.2.3 USB 接口设置点频并查询 75
5 错误说明 77
5.1 错误信息 77
5.1.1 错误信息说明 77
5.2 返修方法 79
5.2.1 联系我们 79
5.2.2 包装与邮寄 80
附 录 81
附录 A SCPI 命令按子系统分类速查表 81
附录 B 错误信息速查表 84

1 手册导航

本章介绍了 1433 系列信号发生器的程控手册功能、章节构成和主要内容，并介绍了提供给用户使用的仪器关联文档。

- 关于手册.....1
- 关联文档.....2

1.1 关于手册

本手册介绍了 1433 系列信号发生器的远程控制和 SCPI 命令使用方法，同时，为方便用户快速掌握程控编程方法，列举了编程示例并介绍了 I/O 函数库的基本概念。为方便您熟练使用该仪器，请在操作仪器前，仔细阅读本手册，然后按手册指导正确操作。

SCPI（Standard Commands for Programmable Instruments）定义了如何远程控制仪器的标准和方法，也是程控电子测试和测量仪器使用的程控编程语言。SCPI 标准基于 IEEE-488.2 标准和形式，具体请浏览 http://www.scpiconsortium.org。该手册详细说明了 1433 程控命令。

程控手册的章节包括:

- **远程控制**

  概述了仪器远程控制操作方法，目的使用户可以对远程控制操作快速上手。分三部分介绍：程控基础，介绍与程控有关的概念、软件配置、程控端口、SCPI 命令等；仪器端口配置方法，介绍 1433 系列信号发生器程控端口的连接方法和软件配置方法；I/O 函数库，介绍仪器驱动器基本概念及 IVI-COM/IVI-C 驱动的基本安装配置说明。

- **程控命令**

  分类介绍了通用命令、仪器命令及兼容命令，逐条说明SCPI命令的功能、参数并示例。

- **编程示例**

  以文字说明和示例代码的方式给出基本编程示例和高级编程示例，并进行说明，方便用户快速掌握信号发生器的程控编程方法。

- **错误说明**

  包括错误信息说明及返修方法。

- **附录**

  给出了1433系列信号发生器关于程控的必要的参考信息，包括：SCPI命令速查表及错误信息速查表等。

1.2 关联文档

1433 系列信号发生器的产品文档包括：

- 快速使用指南
- 用户手册
- 程控手册

快速使用指南

本手册介绍了仪器的配置和启动测量的基本操作方法，目的是：使用户快速了解仪器的特点、掌握基本设置和基础的操作方法。包含的主要章节是：

- 手册当行
- 准备使用
- 典型应用
- 获取帮助

用户手册

本手册详细介绍了仪器的功能和操作使用方法，包括：配置、测量、程控和维护等信息。目的是：指导用户如何全面的理解产品功能特点及掌握常用的仪器测试方法。包含的主要章节是：

- 手册导航
- 概述
- 使用入门
- 操作指南
- 菜单
- 远程控制
- 故障诊断与返修
- 技术指标和测试方法

程控手册

本手册详细介绍了远程编程基础、SCPI 基础、SCPI 命令、编程示例和 I/O 驱动函数库等。目的是：指导用户如何快速、全面的掌握仪器的程控命令和程控方法。包含的主要章节是：

- 手册导航
- 远程控制
- 程控命令
- 编程示例
- 错误说明
- 附录

2 远程控制

本章介绍了 1433 系列信号发生器的程控基础、程控接口与配置方法，并简要的介绍了 I/O 仪器驱动库的概念及分类。以方便用户起步实现远程控制操作。具体内容包括：

- 远程控制基础.....3
- 仪器程控端口与配置.....25
- I/O库.....27

2.1 远程控制基础

- 程控接口.....3
- 消息.....5
- SCPI命令.....6
- 命令序列与同步.....14
- 状态报告系统.....16
- 编程注意事项.....24

2.1.1 程控接口

1433 系列信号发生器支持两种程控接口：LAN 与 USB。
程控接口及关联 VISA 寻址字符串说明，如下表：

表 2.1　远程控制接口类型和 VISA 寻址字符串

程控接口	VISA 寻址字符串	说 明
LAN (Local Area Network)	**原始套接字协议：** TCPIP::host_address::port::SOCKET	控者通过仪器顶部面板网络端口连接仪器实现远程控制。 具体协议请参考： 2.1.1.1 LAN 接口
USB (Universal Serial Bus)	USB::<vendor ID>::<product_ID>::<serial_number>[::RAW]	仪器后面板端口。 具体请参考： 2.1.1.4 USB 接口

- LAN接口.....4
- USB接口.....5

2.1.1.1 LAN 接口

信号发生器可通过 10Base-T 和 100Base-T 局域网内计算机进行远程控制，各种仪器在局域网内组合成系统，并统一由网内计算机控制。信号发生器为实现局域网内远程控制，需事先安装端口连接器、网卡和相关网络协议，并配置相关的网络服务，同时网内控者计算机也需事先安装仪器控制软件和 VISA 库。网卡的三种工作模式是：

- 10Mbit/s 以太网 IEEE802.3；
- 100Mbit/s 以太网 IEEE802.3u；
- 1Gbit/s 以太网 IEEE802.3ab。

控者计算机和信号发生器需通过网口连接到共同的 TCP/IP 协议网络上。连接计算机和信号发生器之间的电缆是商用 RJ45 电缆（带屏蔽或无屏蔽的 5 类双绞线）。数据传输时，采用数据分组传输方式，LAN 传输速度较快。通常，计算机和信号发生器之间的电缆长度不应超过 100 米（100Base-T 和 10Base-T）。关于 LAN 通信的更多信息，请参考：http://www.ieee.org。下面介绍 LAN 接口相关知识：

1） IP 地址

通过局域网对信号发生器进行远程控制时，应保证网络的物理连接畅通。通过信号发生器的菜单“本机 IP”将地址设置到主控计算机所在的子网内即可。例如：主控计算机的 IP 地址是 172.141.11.0，则信号发生器的 IP 地址应设为 172.141.11.XXX，其中 XXX 为 1 ~ 255 之间的数值。

建立网络连接时只需IP地址，VISA寻址字符串形式如下：

TCPIP::host address::port::SOCKET

其中：

- TCPIP 表示使用的网络协议；
- host address 表示仪器的IP地址或者主机名称，用于识别和控制被控仪器；
- port 标识套接字端口号；
- SOCKET 表示原始网络套接字资源类。

举例：

- 建立原始套接字连接时可使用：
  TCPIP::172.141.11.202::5000::SOCKET

提 示

程控系统中多仪器识别方法

若网络中连接多台仪器，采用仪器单独的IP地址和关联的资源字符串区分。主控计算机使用各自的VISA资源字符串识别仪器。

2) **套接字通信**

TCP/IP 协议通过局域网套接字在网络中连接信号源。套接字是计算机网络编程中使用的一个基本方法，它使得使用不同硬件和操作系统的应用程序得以在网络中进行通信。这种方法通过端口（port）使信号发生器与计算机实现双向通信。

套接字是专门编写的一个软件类，里面定义了 IP 地址、设备端口号等网络通信所必需的信息，整合了网络编程中的一些基本操作。在操作系统中安装了打包的库就可以使用套接字。两个常用的套接字库是 UNIX 中应用的伯克利（Berkeley）套接字库和 Windows 中应用的 Winsock 库。

信号发生器中的套接字通过应用程序接口（API）兼容 Berkeley socket 和 Winsock。此外，还兼容其他标准套接字 API。通过 SCPI 命令控制信号发生器时，程序中建立的套接字程序发出命令。在使用局域网套接字之前，必须先设置信号发生器的套接字端口号。信号发生器的套接字端口号为 5000。

2.1.1.2 USB 接口

为实现 USB 程控，需要通过 USB B 型口连接计算机和信号发生器，并事先安装 VISA 库，VISA 自动检测和配置仪器以建立 USB 连接，而不需要输入仪器地址字符串或安装单独的驱动程序。

USB 地址:

寻址字符串格式：USB::<vendor ID>::<product ID>::<serial number>[::RAW]

其中:

- <vendor ID> 代表生产厂家代号;
- <product ID> 代表仪器代号;
- <serial number> 代表仪器序列号;

示例:

USB0::0x3399::0x4E02::100001::RAW

0 x3399：**生产厂家代号**;

0 x4E02：仪**器代号**;

100001： 是仪器的序列号。

2.1.2 消息

数据线上传输的消息分为以下两类:

1) **接口消息**

仪器与主控计算机间通信时，首先需要拉低attention线，然后接口消息才能通过数据线传送给仪器。只有具备GPIB总线功能的仪器才能发送接口消息。

2) **仪器消息**

有关仪器消息的结构和语法，具体请参考章节“5.1.4 SCPI命令”。根据传输方向的不同，仪器消息可分为命令和仪器响应。如不特别声明，所有程控接口使用仪器消息的

方法相同。

a) **命令：**

命令（编程消息）是主控计算机发送给仪器的消息，用于远程控制仪器功能并查询状态信息。命令被划分为以下两类：

- 根据对仪器的影响：
  - 设置命令：改变仪器设置状态，例如：复位或设置频率等。
  - 查询命令：查询并返回数据，例如：识别仪器或查询参数值。查询命令以后缀问号结束。
- 根据标准中的定义：
  - 通用命令：由IEEE488.2定义功能和语法，适用所有类型仪器（若实现）用于实现：管理标准状态寄存器、复位和自检测等。
  - 仪器控制命令：仪器特性命令，用于实现仪器功能。例如：设置频率。语法同样遵循SCPI规范。

b) **仪器响应：**

仪器响应（响应消息和服务请求）是仪器发送给计算机的查询结果信息。该信息包括测量结果、仪器状态等。

2.1.3 SCPI 命令

- SCPI命令简介 6
- SCPI命令说明 7

2.1.3.1 SCPI 命令简介

SCPI(Standard Commands for Programmable Instruments——可程控设备的标准命令)是一个基于标准 IEEE488.2 建立的，适合所有仪器的命令集。其主要目的是为了使相同功能具有相同的程控命令，以实现程控命令的通用性。

SCPI 命令由命令头和一个或多个参数组成，命令头和参数之间由空格分开，命令头包含一个或多个关键字段。命令直接后缀问号即为查询命令。命令分为通用命令和仪器专用命令，它们的语法结构不同。SCPI 命令具备以下特点：

1) 程控命令面向测试功能，而不是描述仪器操作；
2) 程控命令减少了类似测试功能实现过程的重复，保证了编程的兼容性；
3) 程控消息定义在与通信物理层硬件无关的分层中。
4) 程控命令与编程方法和语言无关，SCPI 测试程序易移植。
5) 程控命令具有可伸缩性，可适应不同规模的测量控制。
6) SCPI 的可扩展性，使其成为“活”标准。

如果有兴趣了解更多关于 SCPI 的内容，可参考：

IEEE Standard 488.1-1987, IEEE Standard Digital Interface for Programmable Instrumentation. New York, NY, 1998。

IEEE Standard 488.2-1987, IEEE Standard Codes, Formats, Protocols and Comment Commands for Use with ANSI/IEEE Std488.1-1987. New York, NY, 1998

Standard Commands for Programmable Instruments(SCPI) VERSION 1999.0.

1433 系列信号发生器的程控命令集合、分类及说明，具体请参考：

1) 本手册“3 程控命令”;

2) 本手册“附录 A SCPI 命令按子系统分类速查表”。

2.1.3.2 SCPI 命令说明

- 通用术语.....7
- 命令类型.....8
- 仪器专用命令语法.....8
- 命令树.....10
- 命令参数和响应.....10
- 命令中数值的进制.....13
- 命令行结构.....14

1) 通用术语

下面这些术语适用本节内容。为了更好的理解章节内容，您需要了解这些术语的确切定义。

控制器

控制器是任何用来与 SCPI 设备通讯的计算机。控制器可能是个人计算机、小型计算机或者卡笼上的插卡。一些人工智能的设备也可作为控制器使用。

设备

设备是任何支持 SCPI 的装置。大部分的设备是电子测量或者激励设备，并使用 GPIB 接口通讯。

程控消息

程控消息是一个或者多个正确格式化过的 SCPI 命令的组合。程控消息告诉设备怎样去测量和输出信号。

响应消息

响应消息是指定 SCPI 格式的数据集合。响应消息总是从设备到控制器或者侦听设备。响应消息告诉控制器关于设备的内部状态或测量值。

命令

命令是指满足 SCPI 标准的指令。控制设备命令的组合形成消息。通常来说，命令包括关键字、参数和标点符号。

事件命令

2.1 远程控制基础

事件型程控命令不能被查询。一个事件命令一般没有与之相对应的前面板按键设置，它的功能就是在某个特定的时刻触发一个事件。

查询

查询是一种特殊类型的命令。查询控制设备时，返回适合控制器语法要求的响应消息。查询语句总是以问号结束。

2) 命令类型

SCPI 命令分为两种类型：通用命令和仪器专用命令。图 2.1 显示了两种命令的差异。通用命令由 IEEE 488.2 定义，用来管理宏、状态寄存器、同步和数据存储。因通用令均以一个星号打头，因此很容易辨认。例如*IDN?、*OPC、*RST 都是通用命令。通用命令不属于任何仪器专用命令，仪器采用同一种方法解释该类命令，而不用考虑命令的当前路径设置。

仪器专用命令因包含冒号（:），因此容易辨认。冒号用在命令表达式的开头和关键字的中间，例如：FREQuency[:CW?]。根据仪器内部功能模块，将仪器专用命令划分为对应的子系统命令子集合。例如，功率子系统（:POWer）包含功率相关命令，而状态子系统（:STATus）包含状态控制寄存器的命令。

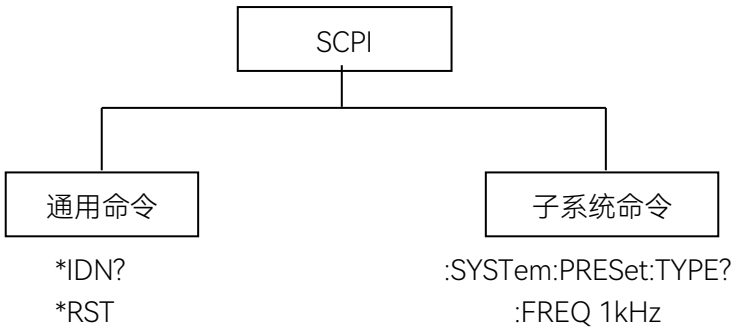

图 2.1 SCPI 命令类型

3) 仪器专用命令语法

一个典型的命令是由前缀为冒号的关键字构成。关键字后面跟着参数。下面是一个语法声明的例子。

[:SOURce]:POWer[:LEVel] MAXimum|MINimum

在上面的例子中，命令中的[:LEVel]部分紧跟着:POWer，中间没有空格。紧跟着[:LEVel]的部分：MINimum|MAXimum 是参数部分。在命令与参数之间有一个空格。语法表达式的其它部分说明见表 2.2 和 2.3。

表 2.2　命令语法中的特殊字符

符号	含义	举例
|	在关键字和参数之间的竖号代表多种选项。	[:SOURce]:AM: SOURce EXTernal|INTernal EXTernal 和 INTernal 是选项
[]	方括号表示被包含的关键字或者参数在构成命令时是可选的。这些暗含的关键字或者参数甚至在它们被忽略时命令也会被执行。	[:SOURce]:AM[:DEPTh]:EXPonential? SOURce 和 DEPTh 是可选项。
< >	尖括号内的部分表示在命令中并不是按照字面的含义使用。它们代表必需包含的部分。	[:SOURce]:FREQ:STOP <val><unit> 该命令中，<val>和<unit> 必须用实际的频率和单位替代。 例如：:FREQ:STOP 3.5GHz
{ }	大括号内的部分表示其中的参数可选。	[:SOURce]:LIST:POWer <val>{,<val>} 例如：LIST:POWer 5

表 2.3　命令语法

字符、关键字和语法	举例
大写的字符代表执行命令所需要的最小字符集合。	[:SOURce]:FREQuency[:CW]?, FREQ 是命令的短格式部分。
命令的小写字符部分是可选择的；这种灵活性的格式被称为“灵活地听”。更多信息请参照“命令参数和响应”部分。	:FREQuency :FREQ,:FREQuency 或者:FREQUENCY, 其中任意一个都是正确的。
当一个冒号在两个命令助记符之间，它将命令树中的当前路径下移一层。更多消息请参照“命令树”的命令路径部分。	:TRIGger:OUTPut:POLarity? TRIGger 是这个命令的最顶层关键字。
如果命令包含多个参数，相邻的参数间由逗号分隔。参数不属于命令路径部分，因此它不影响路径层。	[:SOURce]:LIST:DWELl <val>{,<val>}
分号分隔相邻的 2 条命令，但不影响当前命令路径。	:FREQ 2.5GHZ; :POW 10dBm
空白字符，例如<space>或者<tab>，只要不出现在关键字之间或者关键字之中，通常是被忽略的。然而，你必须用空白字符将命令和参数分隔开来，且不影响当前路径。	:FREQ uency 或者:POWer :LEVel6.2 是不允许的。 在:LEVel 和 6.2 之间必须由空格隔开。 即 :POWer:LEVel 6.2

4) 命令树

大部分远程控制编程会使用仪器专用命令。解析该类命令时，SCPI 使用一个类似于文件系统的结构，这种命令结构被称为命令树，如图 2.2 所示:

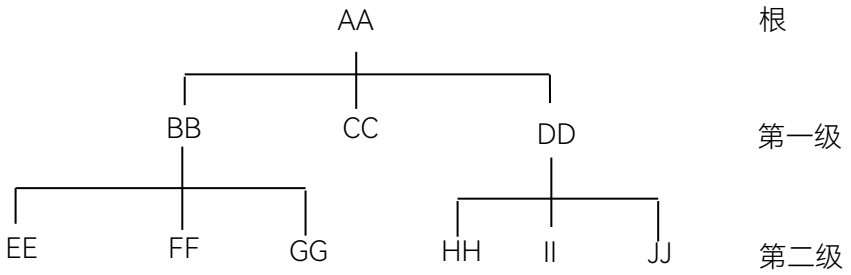

图 2.2 简化的命令树示意图

顶端命令是根命令，简称“根”。命令解析时，依据树结构遵循特定的路径到达下一层命令。例如：:POWer:ALC:SOURce? ，其中，:POWer 代表 AA，:ALC 代表 BB，:SOURce 代表 GG，整个命令路径是（:AA:BB:GG）。

仪器软件中的一个软件模块——**命令解释器**，专门负责解析每一条接收的 SCPI 命令。命令解释器利用一系列的分辨命令树路径的规则，将命令分成单独的命令元。解析完当前命令后，保持当前命令路径不变，这样做的好处是，因为同样的命令关键字可能出现在不同的路径中，更加快速有效的解析后续命令。开机或*RST（复位）仪器后，重置当前命令路径为根。

5) 命令参数和响应

SCPI 定义了不同的数据格式在程控和响应消息的使用中以符合“**灵活地听**”和“**精确地讲**”的原则。更多的信息请参照 IEEE488.2。“**灵活地听**”指的是命令和参数的格式是灵活的。

例如信号发生器设置频率偏移状态命令 :FREQuency:OFFSet:STATe ON|OFF|1|0，

以下命令格式都是设置频率偏移功能开:

:FREQuency:OFFSet:STATe ON，:FREQuency:OFFSet:STATe 1，

:FREQ:OFFS:STAT ON，:FREQ:OFFS:STAT 1。

不同参数类型都有一个或多个对应的响应数据类型。查询时，数值类型的参数将返回一种数据类型，响应数据是精确的，严格的，被称为“**精确地讲**”。

例如，查询功率状态（:POWer:ALC:STATe?），当其为开时，不管之前发送的设置命令是:POWer:ALC:STATe 1 或者 :POWer:ALC:STATe ON，查询时，返回的响应数据总是 1。

表 2.4　SCPI 命令参数和响应类型

参数类型	响应数据类型
数值型	实数或者整数
扩展数值型	整数
离散型	离散型
布尔型	数字布尔型
字符串	字符串
块	确定长度的块
	不确定长度的块
非十进制的数值类型	十六进制
	八进制
	二进制

数值参数

仪器专用命令和通用命令中都可使用数值参数。数值参数接收所有的常用十进制计数法，包括正负号、小数点和科学记数法。如果某一设备只接收指定的数值类型，例如整数，那么它自动将接收的数值参数取整。

以下是数值类型的例子：

0	无小数点
100	可选小数点
1.23	带符号位
4.56e<space>3	指数标记符 e 后可以带空格
-7.89E-01	指数标记符 e 可以大写或小写
+256	允许前面加正号
5	小数点可先行

扩展的数值参数

大部分与仪器专用命令有关的测量都使用扩展数值参数来指定物理量。扩展数值参数接收所有的数值参数和另外的特殊值。所有的扩展数值参数都接收 MAXimum 和 MINimum 作为参数值。其它特殊值，例如：UP 和 DOWN 是否接收由仪器解析能力决定，其 SCPI 命令表中会列出所有有效的参数。

注意：扩展数值参数不适用于通用命令或是 STATus 子系统命令。

扩展数值参数举例：

101	数值参数
1.2GHz	GHz 可以被用作指数（E009）
200MHz	MHz 可以被用作指数（E006）
-100mV	-100 毫伏
10DEG	10 度
MAXimum	最大的有效设置

MINimum	最小的有效设置
UP	增加一个步进
DOWN	减少一个步进

离散型参数

当需要设置的参数值为有限个时，使用离散参数来标识。离散参数使用助记符来表示每一个有效的设置。象程控命令助记符一样，离散参数助记符有长短两种格式，并可使用大小写混合的方式。

下面的例子，离散参数和命令一起使用。

:TRIGger[:SEQuence]:SOURce BUS|IMMediate|EXTernal

BUS	GPIB,LAN,RS-232 触发
IMMediate	立刻触发
EXTernal	外部触发

布尔型参数

布尔参数代表一个真或假的二元条件，它只能有四个可能的值。

布尔参数示例：

ON	逻辑真
OFF	逻辑假
1	逻辑真
0	逻辑假

字符串型参数

字符串型参数允许 ASCII 字符串作为参数发送。单引号和双引号被用作分隔符。

下面是字符串型参数的例子。

'This is Valid'　"This is also Valid"　'SO IS THIS'

实型响应数据

大部分的测试数据是实数型，其格式可以为基本的十进制计数法或科学计数法，大部分的高级程控语言均支持这两种格式。

实数响应数据示例：

1.23E+0
-1.0E+2
+1.0E+2
0.5E+0
0.23
-100.0
+100.0
0.5

整型响应数据

整数响应数据是包括符号位的整数数值的十进制表达式。当对状态寄存器进行查询时，大多返回整数型响应数据。

整数响应数据示例：

0	符号位可选
+100	允许先行正号
-100	允许先行负号
256	没有小数点

离散响应数据

离散型响应数据和离散型参数基本一样，主要区别是离散型响应数据的返回格式只为大写的短格式。

离散响应数据示例：

INTernal	稳幅方式为内部
EXTernal	稳幅方式为外部
MMHead	稳幅方式为毫米波源模块

数字布尔型响应数据

布尔型的响应数据返回一个二进制的数值 1 或者 0。

字符串型响应数据

字符串响应数据和字符串参数是同样的。主要区别是字符串响应数据的分隔符使用双引号，而不是单引号。字符串响应数据还可嵌入双引号，并且双引号间可以无字符。

下面是一些字符串型响应数据的例子：

“This is a string”

“one double quote inside brackets: (“”)”

6） 命令中数值的进制

命令的值可以用二进制，十进制，十六进制或者八进制的格式输入。当用二进制，十六进制或者八进制时，数值前面需要一个合适的标识符。十进制（默认格式）不需要标识符，

当输入一个数值前面没有表示符时，设备会确保其是十进制格式。下面的列表显示了各个格式需要的表示符：

- #B 表示这个数字是一个二进制数值。
- #H 表示这个数字是一个十六进制数值。
- #Q 表示这个数字是一个八进制数值。

下面是 SCPI 命令中十进制数 45 的各种表示：

#B101101

#H2D

#Q55

2.1 远程控制基础

下面的例子用十六进制数值 000A 设置 RF 输出功率为 10dBm（或者当前选择单位的等数值的值，如 DBUV 或者 DBUVEMF）。

:POW #H000A

在使用非十进制格式时，一个测量单位，如 DBM 或者 mV，不和数值一起使用。

7） 命令行结构

一条命令行或许包含多条SCPI命令，为表示当前命令行结束，可采用下面的方法：

- 回车；
- 回车与EOI；
- EOI与最后一个数据字节。

命令行中的命令由分号隔开，属于不同子系统的命令以冒号开头。例如：

MMEM:COPY "Test1", "MeasurementXY";:HCOP:ITEM ALL

该命令行包含两条命令，第一条命令属于MMEM子系统，第二条命令属于HCOP子系统。若相邻的命令属于同一个子系统，命令路径部分重复，命令可缩写。例如：

HCOP:ITEM ALL;:HCOP:IMM

该命令行包含两条命令，两条命令均属于HCOP子系统，一级相同。所以第二条命令可从HCOP的下级开始，并可省略命令开始的冒号。可以缩写为如下命令行：

HCOP:ITEM ALL;IMM

2.1.4 命令序列与同步

IEEE488.2 定义了交迭命令和连续命令之间的区别：

- 连续命令是指连续执行的命令序列。通常各条命令执行速度较快。
- 交迭命令是指下条命令执行前，前条命令未自动执行完成。通常交迭命令的处理时间较长并允许程序在此期间可同步处理其它事件。

即使一条命令行中的多条设置命令，也不一定按照接收的顺序依次执行。为了保证命令按照一定的顺序执行，每条命令必须以单独的命令行发送。

举例：命令行包含设置和查询命令

一条命令行的多条命令若包含查询命令，查询结果不可预知。下面的命令返回固定值：

:FREQ:STAR 1GHZ;SPAN 100;:FREQ:STAR?

返回值：1000000000（1GHz）

下面的命令返回值不固定：

:FREQ:STAR 1GHz;STAR?;SPAN 1000000

返回结果可能是当前的起始频率值，因为主机程序会推迟执行命令。若主机程序接收命令后执行，返回结果也可能是1GHz。

提 示

设置命令与查询命令分开发送

一般规则：为保证查询命令的返回结果的正确性，设置命令和查询命令应在不同的程控消息中发送。

2.1.4.1 防止命令交迭执行

为了防止命令的交迭执行，可采用多线程或者命令：*OPC、*OPC?或者*WAI，只有硬件设置完成后，才执行这三种命令。编程时，计算机可强制等待一段时间以同步某些事件。下面分别予以说明：

- **控者程序使用多线程**
  多线程被用于实现等待命令完成和用户界面及程控的同步，即单独的线程中等待*OPC? 完成，而不会阻塞GUI 或程控线程的执行。
- **三种命令在同步执行中的用法如下表**：

表 2.5 命令语法

方法	执行动作	编程方法
*OPC	命令执行完后，置位 ESR 寄存器中的操作完成位。	置位 ESE BIT0; 置位 SRE BIT5; 发送交迭命令和*OPC; 等待服务请求信号（SRQ） 服务请求信号代表交迭命令执行完成。
*OPC?	停止执行当前命令，直到返回 1。只有 ESR 寄存器中的操作完成位置位时，该命令才返回，表明前面命令处理完成。	执行其它命令前终止当前命令的处理，在当前命令后直接发送该命令。
*WAI	执行*WAI 前，等待发送完所有命令，再继续处理未完成的命令。	执行其它命令前终止当前命令的处理，在当前命令后直接发送该命令。

对于交迭命令的处理时间较短情况，可采用交迭命令后使用命令*WAI 或*OPC 以实现命令同步。为了实现计算机或仪器等待交迭命令执行完成时，同步执行其它任务，可采用下面的同步技术：

- **OPC与服务请求**

1) 设置ESE的OPC屏蔽位（bit0）：*ESE 1;
2) 设置SRE的bit5：*SRE 32 使能ESB服务请求;
3) 发送交迭命令和*OPC;

4) 等待服务请求信号。

服务请求信号代表交迭命令执行完成。

➢ **OPC? 与服务请求**

1) 设置SRE的bit4：*SRE 16 使能M服务请求;
2) 发送交迭命令和*OPC? ;
3) 等待服务请求信号。

服务请求信号代表交迭命令执行完成。

➢ **事件状态寄存器（ESE）**

1) 设置ESE的OPC屏蔽位（bit0）：*ESE 1;
2) 只发送交迭命令，不发送*OPC、*OPC或*WAI;
3) 在定时器中发送“*OPC;*ESR?”循环查询操作完成状态。

返回值（LSB）等于1表示交迭命令执行完成。

➢ ***OPC?与短超时**

1) 只发送交迭命令，不发送*OPC、*OPC或*WAI
2) 在定时器中发送“<short timeout>; *OPC?”循环查询操作完成状态;
3) 返回值（LSB）等于1表示交迭命令执行完成。超时时，表示在操作过程中;
4) 复位超时值到旧值;
5) 发送命令“SYStem:ERRor?”清除错误队列，并删除“-410，查询中断”信息。

返回值（LSB）等于1表示交迭命令执行完成。

2.1.5 状态报告系统

状态报告系统存储当前仪器所有的操作状态信息，包括错误信息。它们分别存储在状态寄存器和错误队列中，并可通过程控接口查询。

- 状态寄存器组织结构.....16
- SCPI状态寄存器结构.....18
- 状态寄存器说明.....19
- 状态报告系统的应用.....21
- 复位状态报告系统.....23

2.1.5.1 状态寄存器组织结构

请参考下面的状态寄存器的等级结构图:

As shown in the following figure, the status information is of hierarchical structure.

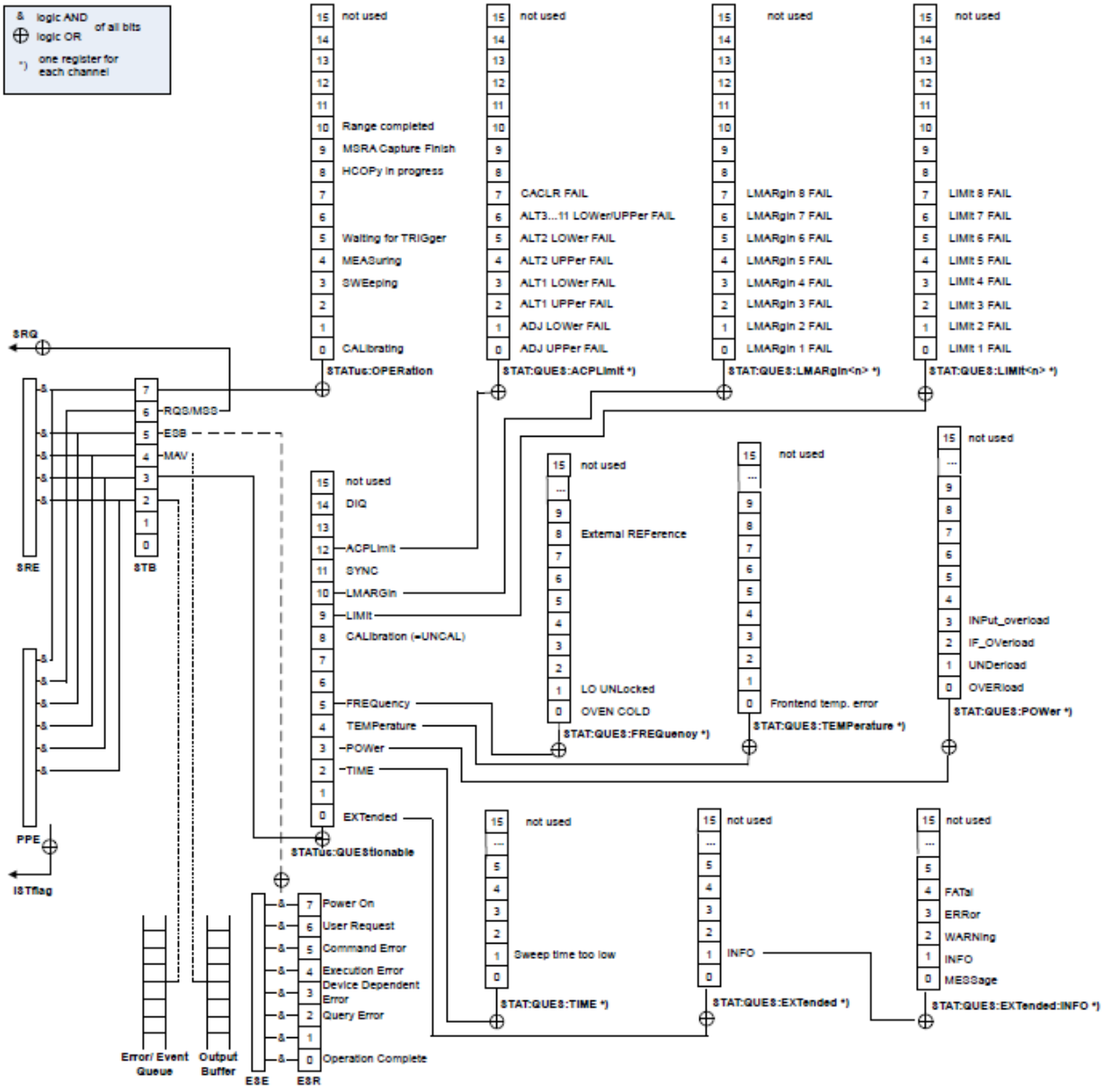

图 2.3　状态寄存器分层结构图

寄存器分类说明如下:

1)　STB, SRE

状态字节 (STB) 寄存器和与之关联的屏蔽寄存器——服务请求使能寄存器 (SRE) 组成了状态报告系统的最高层寄存器。STB通过收集低层寄存器信息，保存了仪器的大致工作状态。

2)　ESR, SCPI状态寄存器

STB接收下列寄存器的信息:

- 事件状态寄存器（ESR）与事件状态使能（ESE）屏蔽寄存器两者相与的值。
- SCPI状态寄存器包括：STATus:OPERation 与 STATus:QUEStionable 寄存器

（SCPI定义），它们包含仪器的具体操作信息。所有的SCPI状态寄存器具备相同的内部结构（具体请参考程控手册2.1.5.2“SCPI状态寄存器结构”章节部分）。

3) IST,PPE

类似SRQ，IST标志（“Individual STatus”）单独的一位，由仪器全部状态组合而成。关联的并行查询使能寄存器（PPE（parallel poll enable register））决定了STB的哪些数据位作用于IST标志。

4) 输出缓冲区

存储了仪器返回给控者的消息。它不属于状态报告系统，但是决定了STB的M位的值。

以上寄存器具体说明请参考程控手册“2.1.6 状态报告系统”章节部分。

提 示

SRE, ESE

服务请求使能寄存器 SRE 可被用作 STB 的使能部分。同理，ESE 可被用作 ESR 的使能部分。

2.1.5.2 SCPI 状态寄存器结构

每一个标准的 SCPI 寄存器包括 5 部分。每部分包含 16 个数据位并功能独立。例如：每个硬件状态分配一个数据位，该数据位对寄存器的所有 5 部分均有效。Bit15 若设置为 0，说明该寄存器的数值是正整型数据。

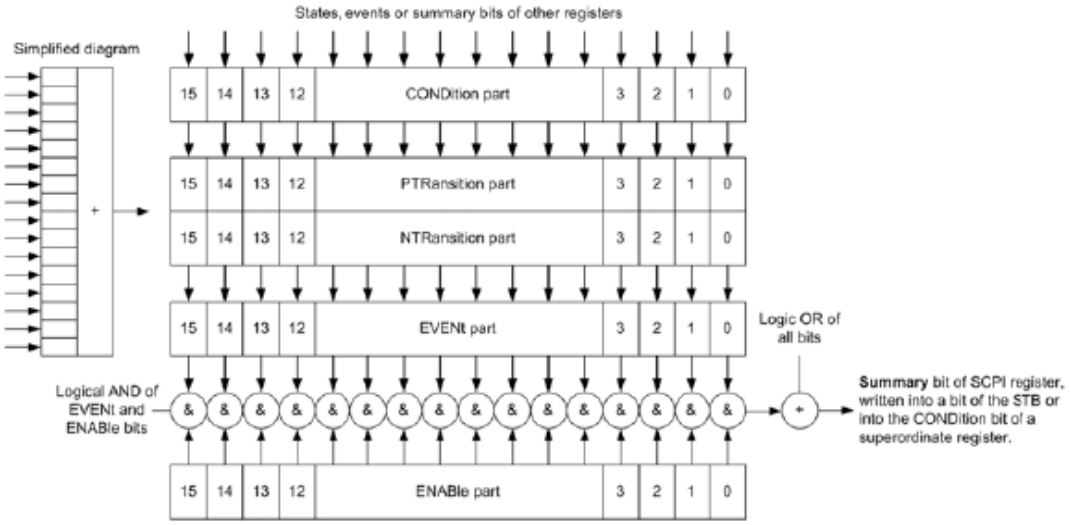

图 2.4 状态寄存器结构

由上图看出状态寄存器由 5 部分组成，分别说明如下：

- **条件寄存器**

该部分由硬件或者低级寄存器的数位和直接写入，反映了当前仪器的工作状态。该寄存器只读，不能写，读取不清除数值。

- **正负转换寄存器**

两个转移寄存器定义了存储在事件寄存器中的条件寄存器的状态转移位。

正转换寄存器类似于转换滤波器，当条件寄存器的某数据位从 0 变换到 1，相关的 PTR 位决定事件位是否设置为 1，说明如下：

—PTR 位=1：设置事件位。

—PTR 位=0：不设置事件位。

正转换寄存器可读可写，并且读取不清除数值。

负转换寄存器类似于转换滤波器，当条件寄存器的某数据位从 1 变换到 0，相关的 NTR 位决定事件位是否设置为 1，说明如下：

—NTR 位=1：设置事件位。

—NTR 位=0：不设置事件位。

正转换寄存器可读可写，并且读取不清除数值。

- **事件寄存器**

该部分表明从上次读取后该事件是否发生，存储了条件寄存器的内容。它仅代表经转移寄存器传递的事件，只能由仪器改变，由用户读取，读取后清除数值。该部分的值常等于整个寄存器的值。

- **使能寄存器**

该部分决定了关联的事件位是否作用于最终的数据和。每一个使能部分的数据位与关联的使能位相与。该部分的逻辑操作结果与数据和位相或。

—使能位=0：关联的事件位不作用于数据和。

—使能位=1：关联的事件位作用于数据和。

该部分可读可写，并且读取不清除。

- **数据位和**

每个寄存器的数据和位由事件和使能部分组成。结果进入高位寄存器的条件部分。仪器自动产生每个寄存器的数据和位，这样事件可引起不同层次的服务请求。

2.1.5.3 状态寄存器说明

下面依次详细介绍各状态寄存器，说明如下：

1） 状态字节（STB）和服务请求使能寄存器（SRE）

IEEE488.2 中定义了状态字节（STB），通过采集低位寄存器的信息反映粗略的仪器状态。其中 bit6 等于其它状态字节位的数据和。状态字节与 SCPI 寄存其中的条件部分比较后的结果可假设为 SCPI 层次中的最高层。通用命令“*STB?”或者串行查询可读取状态字节数值。

状态字节连接到服务请求使能寄存器（SRE）。状态字节的每一个数据位对应 SRE 中的一位。SRE 的 bit6 忽略不用。若 SRE 其中的某数据位设置后，且关联的 STB 位从 0 变化到

1，就会产生一个服务请求（SRQ）。通用命令“*SRE”用于设置 SRE，通用命令“*SRE? ”用于读取 SRE。状态字节说明，具体如下表 2.6 状态字节说明：

表 2.6　状态字节说明

数据位	意 义
0..1	未使用。
2	错误队列非空 若错误队列插入新的错误，设置该位。若关联的 SRE 位使能该位，错误队列中产生新的错误时，产生一次服务请求，这样可识别错误并查询错误信息。这种方法有效的减少了程控中的错误。
3	询问状态寄存器数据和位 若设置询问状态寄存器的事件位以及关联的使能位设置为 1，才能够设置该位。该位代表一个可查询的的仪器状态，可通过查询状态寄存器的询问状态寄存器得到具体的仪器状态信息。
4	M 位（message ailable） 若输出队列的信息可读时设置该位。控者查询仪器信息时使用该位。
5	ESB 位 事件状态寄存器的数据和位。若设置了事件状态寄存器中的其中的一位并且使能事件状态使能寄存器中的相应位，可设置该位。该位置位为 1 表明仪器出现了严重错误，并可通过查询事件状态寄存器获取具体错误信息。
6	MSS 位（master status summary bit） 若仪器触发了服务请求设置该位。
7	操作状态寄存器数据和位 若设置操作状态寄存器的事件位且对应的使能位设置为 1，则可设置该位。该位表明仪器执行了一个操作，具体操作类型可通过查询操作状态寄存器获取。

2) IST 标志和并行查询使能寄存器（PPE）

IST 以一个单独的数据位标识仪器整体状态的组合。可通过并行查询或者发送命令“*IST?”获取该标志。关联的并行查询使能寄存器（PPE（parallel poll enable register））决定了 STB 的哪些数据位作用于 IST 标志。STB 数据位与 PPE 数据位相与，且 bit6 的用法与 SRE 中的 bit6 相反，IST 标志等于所有结果的或值。设置 PPE 可通过命令“*PRE”，读取 PPE 可通过命令“*PRE?”。

3) 事件状态寄存器（ESR）和事件状态使能寄存器（ESE）

IEEE488.2 定义了 ESR。读取事件状态寄存器（ESR（event status register））可通过命令“*ESR?”。ESE 属于 SCPI 寄存器的使能部分，若其中的一位置 1 并且响应的 ESR 其中的某

个数据位从 0 变化到 1，那么 STB 的 ESB 位设置为 1。设置 ESE 可通过命令“*ESE”，可通过命令“*ESE?”读取 ESE。

表 2.7　事件状态字节说明

数据位	意 义
0	操作完成 当前面的命令都已执行完成且收到了命令*OPC，设置该位。
1	未使用。
2	查询错误 若控者未发送查询命令就读取仪器数据，或者还未读取查询数据就发送了了新的命令时，设置该位。表明产生了错误的查询而不能执行查询。
3	仪器产生的错误 若产生了仪器错误，设置该位。错误代码范围：-300~-399，或者正的错误代码。具体错误信息可查询错误队列中的相关信息。
4	执行错误 若收到语法正确的命令但是不能执行设置该位。同时错误队列中生成一个错误代号属于范围-200~-300 的错误。
5	命令错误 若接收的命令语法错误，设置该位。错误代号范围：-100~-200，具体错误信息可查询错误队列中的相关信息。
6	用户请求 若仪器切换到手动控制模式时，设置该位。
7	电源开 打开仪器电源时，设置该位。

2.1.5.4 状态报告系统的应用

状态报告系统用于监测测试系统中的一个或多个仪器状态。为了正确实现状态报告系统的功能，测试系统中的控者必须接收并评估所有仪器的信息，使用的标准方法包括：

1) 仪器发起的服务请求（SRQ）；
2) 串行查询总线系统中的所有的仪器，由系统中的控者发起，目的是找到服务请求发起者及原因。
3) 并行查询所有仪器；
4) 程控命令查询特定仪器状态；
5) 查询错误队列。

1) 服务请求

某些情况下，仪器为获取控者的服务，向控者发送服务请求（SRQ），控者发起一个中断进入相应的中断处理程序。从图 2.4 得知，一个 SRQ 通常由一个或多个状态字节及关联

使能寄存器（SRE）的第 2，3，4，5 或 7 位发起。这些数据位又进一步组成了高级寄存器、错误队列或输出缓冲区。为了尽可能的使用所有的服务请求，使能寄存器 SRE 和 ESE 的所有数据位需要设置为 1。

举例：使用命令*OPC 在扫描结束后，产生 SRQ 信号。

a) 调用写函数 InstrWrite 写命令“*ESE 1”，设置 ESE bit0（操作完成）。

b) 调用写函数 InstrWrite 写命令“*SRE 32”，设置 SRE bit5（ESB）。

c) 调用写函数 InstrWrite 写命令“*INIT;*OPC”，操作完成后产生 SRQ 信号。

仪器设置完成后，仪器产生一次 SRQ。

SRQ 只能由仪器本身发起，控者程序应允许在仪器产生错误时，向其发起服务请求，并由专门的中断服务程序处理。具体例程请参考章节 4.2.1.1. “服务请求”。

2) 串行查询

类似于命令*STB，串行查询用来查询仪器的状态字节。串行查询采用接口消息的方式，因此查询速度较快。IEEE 488.2 定义了串行查询的具体方法。该方法主要用于快速获取测试系统中与控者连接的一个或多个仪器的状态。

3) 并行查询

测试系统中，控者通过一条命令向数据线上发送一个信息位，能够同时查询 8 个仪器。配置到仪器的数据线上的数据是逻辑“0”或“1”。除了 SRE 寄存器决定产生 SRQ 的条件，并行查询使能寄存器（PPE）与 STB 寄存器数据位相与运算。得到的结果经过或运算，再经过位取反，作为响应结果发送给并行查询的控者，也可通过命令*IST 获取该结果。

并行查询时，首先通过命令 PPC 设置仪器到并行查询状态，该命令分配仪器一根数据线并决定响应时数据位是否反转。执行并行查询时使用 PPE 寄存器。并行查询主要用于控者快速定位哪台仪器发送了服务请求。为此寄存器 SRE 和 PPE 应设置相同的数值。

4) 查询仪器状态

可采用下面两种命令，查询状态状态寄存器的每一部分：

- 命令*ESR?, *IDN?, *IST?, *STB? 查询高级寄存器；
- 状态系统命令查询 SCPI 寄存器（例如：STATus：QUEStionable…）。

被查询的寄存器的返回值通常为十进制格式，由控者程序检测使用。为获取更加详细产生 SRQ 的原因，通常 SRQ 后才进行并行查询。

响应数据位说明

STB 与 ESR 寄存器包含 8 位，SCPI 寄存器包含 16 位。查询状态寄存器的返回值采用十进制格式。该十进制数值等于各数据位与各自的权重运算，相加后的结果。

数据位与其权重的关系如下图：

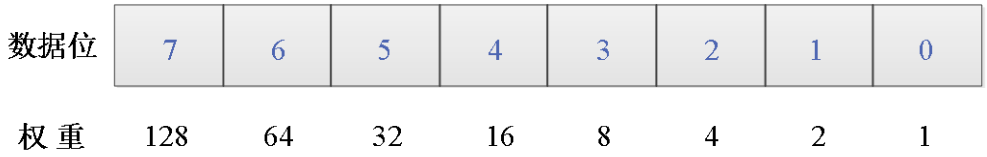

图 2.5　数据位与权重关系图

5)　错误队列

仪器的每个错误状态对应错误队列中的一个入口，其中包含具体的错误信息文本，可通过错误日志查看，或者通过程控命令查询：SYSTem:ERRor[:NEXT]? 或 SYSTem:ERRor:ALL?。若错误队列中无错误，查询返回 0，“无错误”。

在控者服务请求处理程序中应查询错误队列，因为可得到比状态寄存器更加精确的错误原因描述。尤其在控者程序测试阶段，应经常查询错误队列，以明确控者发送给仪器的错误命令记录。

2.1.5.5 复位状态报告系统

下面的列表列出了复位状态报告系统的命令和事件。除了命令*RST 和 SYSTem:PRESet，其它命令不改变仪器功能设置。同样，DCL 也不改变仪器设置状态。具体说明如下表：

表2.19　复位状态报告系统

事件 / 作用	电源开/关（加电状态清除）		DCL, SDC（仪器清除，选择仪器清除）	*RST 或 SYSTem:PRESet	STATus:PRESet	*CLS
	0	1				
清除 STB, ESR	—	是	—	—	—	是
清除 SRE,ESE	—	是	—	—	—	—
清除 PPE	—	是	—	—	—	—
清除寄存器的事件部分	—	是	—	—	—	是

表2.19（续1） 复位状态报告系统

清除操作和询问寄存器中的使能部分。其它寄存器的使能部分填充 1。	—	是	—	—	是	—
正转移部分填充 1。清除负转移部分。	—	是	—	—	是	—
清除错误队列	是	是	—	—	—	是
清除输出缓冲区	是	是	是	—	—	—
清除命令处理和输入缓冲区	是	是	是	—	—	—

2.1.6 编程注意事项

1） 改变设置前请初始化仪器状态

远程控制设置仪器时，首先需要初始化仪器状态（例如发送”*RST”）,然后再实现需要的状态设置。

2） 命令序列

一般来说，需要分开发送设置命令和查询命令。否则，查询命令的返回值会根据当前仪器操作顺序而变化。

3） 故障反应

服务请求只能由仪器自己发起。测试系统中的控者程序应指导仪器在出现错误时主动发起服务请求，进而进入相应的中断服务程序中进行处理。

4） 错误队列

控者程序每次处理服务请求时，应查询仪器的错误队列而不是状态寄存器，来获取更加精确的错误原因。尤其在控者程序的测试阶段，应经常查询错误队列以获取控者发送给仪器的错误命令。

2.2 仪器程控端口与配置

- LAN.....25
- USB.....26

2.2.1 LAN

LAN（Local Area Network）程控系统采用SOCKET控制1433系列信号发生器。

- 建立连接.....25
- 接口配置.....25

2.2.1.1 建立连接

使用网线将1433系列信号发生器与外部控者（计算机）连接到局域网，如图2.6所示:

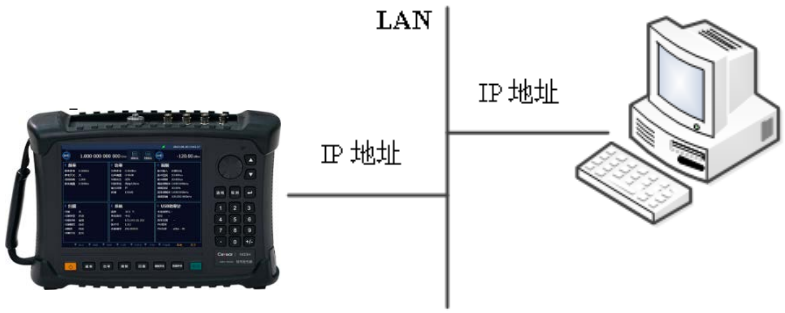

图2.6　LAN接口连接图

2.2.1.2 接口配置

通过局域网对信号发生器进行远程控制时，应保证网络的物理连接畅通。由于不支持DHCP、域名访问以及广域网络连接，因此信号发生器的网络程控设置相对简单。按[系统] [系统设置>>] [网络配置>>]，通过图 2.7 所示的菜单，将其中“IP 地址”,“子网地址”,“默认网关”设置到主控制器所在的子网内即可。



图 2.7　LAN 接口设置

注　意

确保信号发生器通过 10Base-T LAN 或 100Base-T LAN 电缆物理连接正常

由于该信号发生器只支持单一局域网络控制系统的搭建，且只支持静态 IP 地址的设置，不支持 DHCP，也不支持通过 DNS 和域名服务器访问主机，因此不需要用户修改子网掩码，仪器内将其固定设置为：255.255.255.0。

2.2.2 USB

- 建立连接.....26
- 接口配置.....27

2.2.2.1 建立连接

在测试仪的顶部面板中，USB 接口分为 A 型（内嵌 4 触点：触点 1 在左边）和 mini B 型两种，A 型接口可以连接 USB 鼠标、键盘或其他 USB 接口设备。B 型接口用于实现程控功能。使用 USB 电缆连接 1433 系列信号发生器的 mini B USB 接口与外部控者（计算机），如图 2.8 所示：

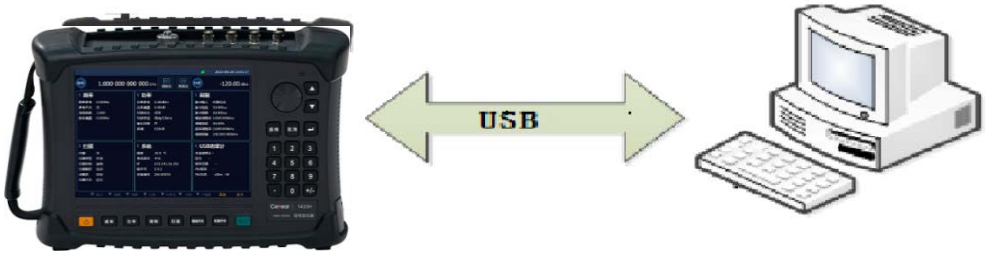

图 2.8　GPIB 接口连接图

注 意

顶部面板 Type-A 型 USB 主控端口连接器的使用

顶部面板的 Type-A 连接器是 USB 主控端口连接器，在 1433 系列信号发生器中，该端口用来连接 USB 接口的闪存盘，以实现仪器驻机软件的升级，也可以连接 USB 键盘和鼠标对信号发生器进行控制。不能通过该端口程控仪器。

2.2.2.2 接口配置

用户在通过仪器的 USB 接口搭建系统时，需要获取测试仪的 USB 地址的 VID、PID 及一起序列号。默认情况下，VID 为 0x3399，PID 为 0x4E00，序列号跟仪器出厂设置的串号相同。用户可以在主控计算机上通过计算机设备管理器，进入[USB Test and Measurement Device(IVI)属性]—>[详细信息]—>[设备实例路径]查看 USB 口相关信息。在 USB 程控条件下，一般不更改 VID、PID 及序列号信息。

2.3 I/O 库

- I/O库概述.....27
- I/O库安装与配置.....28

2.3.1 I/O 库概述

I/O 库是为仪器预先编写的一些软件程序库被称为仪器驱动程序，即：仪器驱动器（Instrument driver），它是介于计算机与仪器硬件设备之间的软件中间层，由函数库、实用程序、工具套件等组成，是一系列软件代码模块的集合，该集合对应于一个计划的操作，如配置仪器、从仪器读取、向仪器写入和触发仪器等。它驻留在计算机中，是连接计算机和仪器的桥梁和纽带。通过提供方便编程的高层次模块化库，用户不再需要学习复杂的针对某个仪器专用的低层编程协议，采用仪器驱动器是快速开发测试测量应用的关键。

从功能上看，一个通用的仪器驱动器一般由功能体、交互式开发者接口、编程开发者接

口、子程序接口和 I/O 接口五部分组成，如图 2.12 所示。

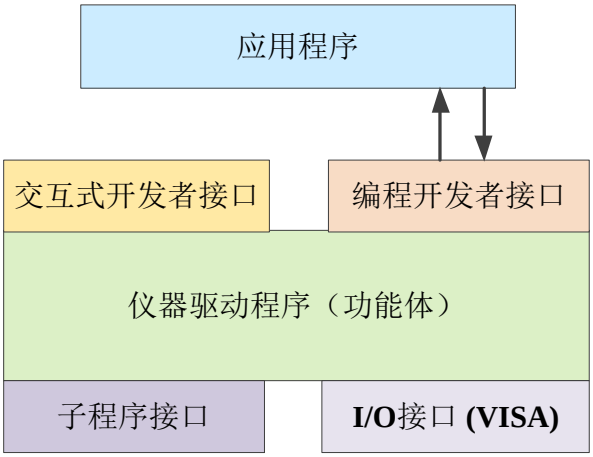

图 2.12　仪器驱动器结构模型

具体说明如下:

1) 功能体。这是仪器驱动器的主功能部分，可以理解为仪器驱动器的框架程序。。
2) 交互开发者接口。为方便用户使用，支持仪器驱动器开发的应用开发环境通常提供图形化的交互开发接口。例如，Labwindows/CVI 中，函数面板就是一种交互开发接口。函数面板中，仪器驱动器函数的各个参数都是以图形化的控件形式表示。
3) 编程开发者接口。它是应用程序调用仪器驱动器函数的软件接口，例如 Windows 系统下仪器驱动器的动态链接库文件.dll。
4) I/O 接口。它完成仪器驱动器与仪器间的实际通信。可以使用总线专用 I/O 软件，如 GPIB、RS-232;也可以使用跨多个总线使用的通用的标准 I/O 软件:VISA I/O。
5) 子程序接口。它是仪器驱动器访问其它一些支持库的软件接口，例如数据库、FFT 函数等。当仪器驱动器为完成其任务而需调用其它软件模块、操作系统、程控代码库及分析函数库时，将用到子程序接口。

2.3.2 I/O 库安装与配置

伴随着测试领域的应用经历了从传统仪器到虚拟仪器等不同的发展阶段,并且为了解决自动测试系统中仪器可互换性和测试程序的可重用性,仪器驱动程序经历了不同的发展过程。目前比较流行通用的驱动器是 IVI（Interchangeable Virtual Instruments）仪器驱动器，它基于 IVI 规范，定义了新的仪器编程接口，以及插入类驱动程序和 VPP 架构到 VISA 上，使测试应用程序与仪器硬件完全独立，并增加了独有的仪器仿真、范围检测、状态缓存等功能，提高了系统运行的效率与真正实现了仪器互换。

IVI 驱动分为两种类型：IVI-C 与 IVI-COM，IVI-COM 基于微软组件对象模型（COM）技术，采用 COM API 的形式；IVI-C 基于 ANSI C，采用 C API 的方式。这两种驱动类型都是遵照 IVI 规范定义的仪器类来设计的，它们的应用开发环境也都相同，包括 Visual Studio，Visual Basic，Agilent VEE，LabVIEW，CVI/LabWindows 等。

为满足不同用户在不同开发环境下的需求，目前需要提供两种驱动形式。信号发生器的 IVI 驱动利用 Nimbus Driver Studio 开发，直接生成 IVI-COM 与 IVI-C 驱动及程序安装包，

具体安装配置请参阅您所选择的控制卡及 I/O 库的随机文档资料。

安装后的 IVI 驱动分为：IVI 固有功能组与仪器类功能组（基本功能组和扩展功能组）。具体功能分类、函数和属性说明可参考驱动自带的帮助文档。

提 示

配置端口以及安装 IO 库

在使用计算机控制信号发生器之前，请确认您已正确安装且配置必要的端口和 I/O 库。

提 示

I/O 库的使用

安装随机的 IVI-COM/C 驱动程序安装包，会自动安装驱动函数面板、帮助文档及驱动函数的示例程序，方便用户开发集成程控功能。

3 程控命令

- 命令说明.....31
- 通用命令.....31
- 仪器子系统命令.....32

3.1 命令说明

该章节提供了详细的命令参考信息，便于实现远程控制，具体包括：

- 完整的语法格式及参数列表；
- 对于 SCPI 非标准命令，列出语法图示；
- 详细的功能描述及关联的命令说明；
- 支持的命令格式（设置 或 查询）；
- 参数说明，包含：数据类型、取值范围及默认值（单位）；
- 若支持查询命令格式，返回值说明；
- SCPI 命令使用示例；
- 开机/复位参数状态（值）；
- SCPI 命令对应的仪器按键路径；
- 命令兼容的同类仪器中的仪器型号，若不注明则说明当前命令只适用于 1433。
- 其它说明事项。

为方便用户查询使用，通用命令和仪器子系统命令章节的起始部分，首先以引导小目录方式列出命令列项。

3.2 通用命令

通用命令用来控制仪器状态寄存器、状态报告、同步、数据存储及其它通用功能，通用命令的用法和作用适用不同的仪器。所有的通用命令都可以通过命令字中的第一个“*”被识别，IEEE488.2 中详细定义了这些通用命令。

以下是 IEEE488.2 通用命令的解释和说明。

- *IDN?.....32
- *RST.....32

提 示

命令使用：

如不特别说明，命令可用于设置或者查询。

若一个命令仅用于设置或查询，或者启动一个事件，命令说明中会单独进行说明。

***IDN?**

功能描述：返回仪器标识。
返回值： <ID>："生产厂家,<仪器型号>,<串号>,<固件版本号>"
举例：Ceyear Technologies,1433H,ZAL00165,1.0.2
说明：仅查询。

***RST**

功能描述：该命令完成1433信号发生器复位功能。
说明：仅设置。

3.3 仪器子系统命令

该章节详细说明1433系列信号发生器的子系统命令。

- System子系统.....32
- Frequency子系统.....39
- Power子系统.....41
- Modulation子系统.....49
- Sweep子系统.....56
- 其他命令.....65

3.3.1 System 子系统

用于系统参数设置和查询，包括：

- [:SOURce]:ROSCillator:SOURce.....33
- [:SOURce]:ROSCillator: EXTReference.....33
- :SYStem:PANEl:LANguage.....33
- :SYStem:FAN:STATe.....34
- :SYStem:COMMunicate:LAN:IP.....34
- :SYStem:COMMunicate:LAN:SUBNET.....34
- :SYStem:COMMunicate:LAN:GATEway.....35
- :SYStem:TEMPerature.....35
- :SYStem:DORMancy:STATe.....35
- :SYStem:DORMancy:TIME.....36
- :SYStem:SHUTdown:STATe.....36
- :SYStem:SHUTdown:TIME.....36
- :SYStem:DISPlay:MODE.....37
- :SYStem:BRIGht:STATe.....37
- :SYStem:PRESet:TYPE.....38

- :MEMory:STORe:STATus.....38
- :MEMory:LOAD:STATus.....39

[:SOURce]:ROSCillator:SOURce <Type>

功能描述: 设置和查询信号发生器的振荡源。
设置格式: [:SOURce]:ROSCillator:SOURce INTernal|EXTernal
查询格式: [:SOURce]:ROSCillator:SOURce?
返回值为 0 到 1，分别对应于系统参考设置为内参考和外参考。
参数说明:
< Type > 离散数据。取值如下:
INTernal: 系统参考设置为内参考。
EXTTernal: 系统参考设置为外参考。

举例: :ROSC:SOUR INTernal 设置系统振荡源为内参考。

:ROSC:SOUR? 查询系统振荡源类型。
复位状态: 内部
按键路径: [系统]—>[系统设置] —>[参考选择]

[:SOURce]:ROSCillator: EXTReference <FreqRef>

功能描述: 设置和查询信号源外参考频率。
设置格式: [:SOURce]:ROSCillator:EXTReference < FreqRef ><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:ROSCillator:EXTReference?
返回信号源外参考频率值，单位为 Hz。
参数说明:
< FreqRef > 数据类型为double类型。
设置范围：1MHz~100MHz

举例: :ROSCillator:EXTR 10MHz 设置信号源外参考频率为 10MHz。

:ROSCillator:EXTR? 查询信外参考频率。
复位状态: 0Hz
按键路径: [系统]—>[系统设置] —>[外参考频率]

:SYStem:PANEl:LANguage <Type>

功能描述: 设置和查询系统语言。
设置格式: :SYSTem:PANEl:LANGuage Chinese|English
查询格式: :SYSTem:PANEl:LANGuage?
返回值为 0 和 1，分别对应于中文和英文。
参数说明:
< Type > 离散数据。取值如下:
Chinese: 中文。

Chinese: 英文。

举例: : SYST:PANE:LANG Chinese 设置系统语言为中文。

: SYST:PANE:LANG? 查询系统语言类型。

复位状态: 中文

按键路径: [系统]—>[系统设置] —>[系统语言]

:SYStem:FAN:STATe <State>

功能描述: 设置风扇控制方式，仅设置。

设置格式: :SYStem: FAN:STATe AUTO|OFF|ON

参数说明:

< State > 数据类型为离散型。取值如下:

AUTO: 风扇控制方式为自动。

OFF: 风扇控制方式为关。

ON: 风扇控制方式为开。

举例: : SYStem: FAN:STATe AUTO 设置风扇控制方式为自动。

复位状态: AUTO

按键路径: [系统]—> [系统设置] —> [风扇]

:SYSTem:COMMunicate:LAN:IP <Ip>

功能描述: 设置和查询IP地址。

设置格式: :SYSTem:COMMunicate:LAN:IP <Ip>

查询格式: :SYSTem:COMMunicate:LAN:IP?

返回 ip 地址。

参数说明:

< Ip > string型。

举例: :SYSTem:COMMunicate:LAN:IP“172.141.11.202“ 设 置 ip 地 址 为 172.141.11.202。

:SYSTem:COMMunicate:LAN:IP? 查询ip地址。

复位状态: 172.141.11.202

按键路径: [系统]—> [系统设置] —> [网络配置] —> [IP地址]

:SYSTem:COMMunicate:LAN:SUBNET <Subnet>

功能描述: 设置和查询子网掩码。

设置格式: :SYSTem:COMMunicate:LAN: SUBNET< subnet>

查询格式: :SYSTem:COMMunicate:LAN: SUBNET?

返回子网掩码。

参数说明:
< Subnet > string型。

举例: :SYSTem:COMMunicate:LAN: SUBNET “255.255.255.0“ 设子网掩码为255.255.255.0。

:SYSTem:COMMunicate:LAN: SUBNET? 查询子网掩码。
复位状态: 255.255.255.0
按键路径: [系统]—> [系统设置] —> [网络配置] —> [子网掩码]

:SYSTem:COMMunicate:LAN:GATEway <Gateway>

功能描述: 设置和查询默认网关。
设置格式: :SYSTem:COMMunicate:LAN: GATEway <Gateway>
查询格式: :SYSTem:COMMunicate:LAN: GATEway?
返回默认网关。
参数说明:
< Gateway > string型。

举例: :SYSTem:COMMunicate:LAN: GATEway“172.141.11.254“ 设置默认网关为172.141.11.254。

:SYSTem:COMMunicate:LAN: GATEway? 查询默认网关。
复位状态: 172.141.11.254
按键路径: [系统]—> [系统设置] —> [网络配置] —> [默认网关]

:SYStem:TEMPerature?

功能描述: 设置和查询默认网关。
查询格式: :SYStem:TEMPerature?
返回值为十进制温度值。

举例: :SYS:TEMP? 查询系统温度。

说明: 仅查询。

:SYStem:DORMancy:STATe<State>

功能描述: 设置和查询自动休眠状态。
设置格式: :SYStem:DORMancy:STATe ON|OFF|1|0
查询格式: :SYStem:DORMancy:STATe?
返回值为 0 和 1, 分别对应于自动休眠状态为关闭状态(OFF)和打开状态(ON)。
参数说明:
< State >数据类型为布尔型。取值如下:

OFF | 0: 休眠状态为关。
ON | 1: 休眠状态为开。

举例: :SYStem:DORMancy:STATe 1 设置自动休眠状态为开。

:SYStem:DORMancy:STATe? 查询自动休眠状态。

复位状态: 关

按键路径: [系统]—> [显示] —> [休眠时间]

:SYStem:DORMancy:TIME<Val>

功能描述: 设置和查询自动休眠时间。

设置格式: :SYStem:DORMancy:TIME<Val>

查询格式: :SYStem:DORMancy:TIME?
返回自动休眠时间,单位为 min。

参数说明:

< Val > 数据类型为整型。
设置范围: 1~240

举例: :SYStem:DORMancy:TIME 1 设置自动休眠时间为 1min。

:SYStem:DORMancy:TIME? 查询自动休眠时间。

复位状态: 1

按键路径: [系统]—> [显示] —> [休眠时间]

: SYStem:SHUTdown:STATe<State>

功能描述: 设置和查询定时自动关机状态。

设置格式: :SYStem:SHUTdown:STATe ON|OFF|1|0

查询格式: :SYStem:SHUTdown:STATe?
返回值为 0 和 1，分别对应于定时自动关机状态为关闭状态(OFF)和打开状态(ON)。

参数说明:

< State >数据类型为布尔型。取值如下:
OFF | 0: 定时关机状态为关。
ON | 1: 定时关机状态为开。

举例: :SYStem:SHUTdown:STATe 1 设置定时自动关机状态为开。

:SYStem:SHUTdown:STATe? 查询定时自动关机状态。

复位状态: 关

按键路径: [系统]—> [显示] —> [定时关机]

:SYStem:SHUTdown:TIME<Val>

功能描述: 设置和查询自动关机时间。

设置格式: :SYStem:SHUTdown:TIME<Val>
查询格式: :SYStem:SHUTdown:TIME?
返回定时自动关机时间,单位为 min。
参数说明:
< Val > 数据类型为整型。
设置范围: 1~240

举例: :SYStem:SHUTdown:TIME 1 设置自动关机时间为 1min。

:SYStem:SHUTdown:TIME? 查询自动关机时间。
复位状态: 30
按键路径: [系统]—> [显示] —> [关机时间]

:SYStem:DISPlay:MODE <Mod>

功能描述: 设置和查询显示模式。
设置格式: :SYStem:DISPlay:MODE NORMal|OUTDoor
查询格式: SYStem:DISPlay:MODE?
返回值为 0 和 1，分别对应于显示模式为正常模式(NORMal)和户外模式(OUTDoor)。
参数说明:
<Mod> 数据类型为离散型。取值如下:
NORMal: 显示模式为正常模式。
OUTDoor: 显示模式为户外模式。

举例: :SYStem:DISPlay:MODE NORMal 设置显示模式为正常模式。

:SYStem:DISPlay:MODE? 查询显示模式。
复位状态: 正常模式
按键路径: [系统]—> [显示] —> [模式切换]

:SYStem:BRIGht:STATe <State>

功能描述: 设置和查询背光控制方式。
设置格式: :SYStem:BRIGht:STATe MAN|AUTO
查询格式: :SYStem:BRIGht:STATe?
返回值为 0 和 1，分别对应于背光控制方式为手动模式(MAN)和自动模式(AUTO)。
参数说明:
< State >数据类型为离散型。取值如下:
MAN: 背光控制方式为手动模式。
AUTO: 背光控制方式为自动模式。

举例: :SYStem:BRIGht:STATe MAN 设置背光控制方式为手动模式。

:SYStem:BRIGht:STATe? 查询背光控制方式。

复位状态: 手动

按键路径: [系统]—> [显示] —> [亮度调节]

: SYStem:BRIGht:DATA <Val>

功能描述: 设置和查询背光强度。

设置格式: :SYStem:BRIGht:DATA <Val >

查询格式: :SYStem:BRIGht:DATA?
返回背光强度。

参数说明:

< Val > 数据类型为double型。
设置范围：60~250

举例: :SYStem:BRIGht:DATA 200 设置背光强度为 200。

:SYStem:BRIGht:DATA? 查询背光强度。

复位状态: 201

按键路径: [系统]—> [显示] —> [亮度调节]

:SYStem:PRESet:TYPE<Type>

功能描述: 设置和查询复位方式。

设置格式: :SYStem:PRESet:TYPE NORMal|USER

查询格式: :SYStem:PRESet:TYPE?
返回值为 0 和 1，分别对应于厂家和用户。

参数说明:

<Type> 离散数据。取值如下:
NORMal: 厂家。
USER: 用户。

举例: :SYStem:PRESet:TYPE NORMal 设置复位方式为厂家。

:SYStem:PRESet:TYPE? 查询复位方式。

复位状态: 厂家

按键路径: [系统]—>[文件] —>[复位类型]

:MEMory:STORe:STATus <Name>

功能描述: 保存用户状态。

设置格式: :MEMory:STORe: STATus <Name>

参数说明:

< Name > string**型。**

举例: :MEMory:STORe: STATus“data“ 保存用户状态,文件名为“data“。

说明: 仅设置。
按键路径: [系统] —> [文件] —> [存储状态]

:MEMory:LOAD:STATus <Name>

功能描述: 调用用户状态。
设置格式: :MEMory:LOAD: STATus <Name >
参数说明:
< Name > string**型。**

举例: :MEMory:LOAD: STATus“data“ 调用用户状态,文件名为“data“。

说明: 仅设置。
按键路径: [系统] —> [文件] —> [调用状态]

3.3.1 Frequency 子系统

用于控制射频输出信号的频率功能，包括:

- [:SOURce]:FREQuency[:CW|FIXed]....39
- [:SOURce]:FREQuency:REFerence....39
- [:SOURce]:FREQuency:REFerence:STATe....40
- [:SOURce]:FREQuency:OFFSet....40
- [:SOURce]:FREQuency:MULTiplier....41

[:SOURce]:FREQuency[:CW|FIXed] <Frequency>

功能描述: 设置和查询信号源频率。
设置格式: [:SOURce]:FREQuency[:CW|FIXed] <Frequency><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:FREQuency[:CW|FIXed]?
返回信号源当前频率值，单位为 Hz。
参数说明:
<Frequency> 数据类型为double型。
设置范围:
1MHz~20GHz(1433D)/26.5GHz(1433E)/40GHz(1433F)/50GHz(1433H)。

举例: :FREQ 10MHz 设置信号源输出频率为 10MHz。

:FREQ? 查询信号源输出频率。
复位状态: 1GHz
按键路径: 【频率】 —> [设置点频]

[:SOURce]:FREQuency:REFerence <FreqRef >

功能描述: 设置和查询信号源频率参考。

设置格式: [:SOURce]:FREQuency:REFerence <FreqRef><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:FREQuency:REFerence?
返回信号源当前频率参考值，单位为 Hz。
参数说明:
<FreqRef> 数据类型为double类型。
设置范围: -500GHz~500GHz

举例: :FREQ:REF 10GHz 设置信号源频率参考为 10GHz。

:FREQ:REF? 查询信号源频率参考。
复位状态: 0Hz
按键路径: 【频率】—> [频率参考]

: SOURce]:FREQuency:REFerence:STATe <State>

功能描述: 设置和查询信号源频率参考开关。
设置格式: [:SOURce]:FREQuency:REFerence:STATe ON|OFF|1|0
查询格式: [:SOURce]:FREQuency:REFerence:STATe?
返回值为 0 和 1, 分别对应于频率参考状态为关闭状态(OFF)和打开状态(ON)。
参数说明:
<State> 数据类型为布尔型。取值如下:
OFF | 0：频率参考关。
ON | 1： 频率参考开。

举例: :FREQ:REF:STAT 1 设置信号源频率参考开。

:FREQ:REF:STAT? 查询信号源频率参考开关状态。
复位状态: 关
按键路径: 【频率】—> [参考开关]

[:SOURce]:FREQuency:OFFSet <FreqOffset>

功能描述: 设置和查询频率偏置。
设置格式: [:SOURce]:FREQuency:OFFSet <FreqOffset><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:FREQuency:OFFSet?
返回信号源当前频率偏置值，单位为 Hz。
参数说明:
<FreqOffset> 数据类型为double类型。
设置范围: -500GHz~500GHz

举例: :FREQ:OFFS 10GHz 设置信号源频率偏置考为 10GHz。

:FREQ:OFFS? 查询信号源频率偏置值。
复位状态: 0Hz
按键路径: 【频率】—> [频率偏置]

[:SOURce]:FREQuency:MULTiplier <Multipl>

功能描述: 设置和查询倍频系数。
设置格式: [:SOURce]:FREQuency:MULTiplier <Multipl>
查询格式: [:SOURce]:FREQuency:MULTiplier?
返回信号源当前倍频系数值。
参数说明:
<Multipl> 数据类型为double类型。
设置范围: -100.0~100.0

举例: :FREQ:MULT 10.0 设置信号源倍频系数为 10.0。

:FREQ:MULT? 查询信号源倍频系数。
复位状态: 1.0
按键路径: 【频率】 —> [倍频系数]

注: 显示频率和实际输出频率、频率参考、倍频系数、频率偏置满足如下关系式:
频率参考开: 显示频率=输出频率*频率倍乘+频率偏置-频率参考
频率参考关: 显示频率=输出频率*频率倍乘+频率偏置。

3.3.1 Power 子系统

用于控制射频输出信号的频率功能，包括:

- [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] 42
- [:SOURce]:POWer:ALC[:STATe] 42
- [:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet 42
- [:SOURce]:POWer:REFerence 43
- [:SOURce]:POWer:REFerence:STATe 43
- [:SOURce]:POWer:ALC:BANDwidth|BWIDth 43
- [:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO 44
- [:SOURce]:OUTPut:BLANking[:STATe] 44
- [:SOURce]:POWer:ALC:SEARch 45
- [:SOURce]:POWer:ATTenuation:AUTO 45
- [:SOURce]:POWer:ALC:Level 45
- [:SOURce]:POWer:ATTenuation 46
- [:SOURce]:CORRection:STATe 46
- [:SOURce]:CORRection:STARt:FREQuency 47
- [:SOURce]:CORRection:STOP:FREQuency 47
- [:SOURce]:CORRection:STARt:AMPLitude 47
- [:SOURce]:CORRection:STOP:AMPLitude 48
- [:SOURce]:CORRection:POINts 48
- [:SOURce]:CORRection:FILL 48
- [:SOURce]:CORRection:LOAD 49

- [:SOURce]:CORRection:STORe.....49

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <Ampl>

功能描述: 设置和查询射频输出信号的功率电平。
设置格式: [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <Ampl><dBm>
查询格式: [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?
返回信号源当前功率电平值。
参数说明:
<Ampl> 数据类型为double类型。
设置范围: -135.00dBm~25.00dBm

举例: :POW 10.0dBm 设置信号源功率为 10.0dBm。

:POW? 查询信号源功率。
复位状态: -120dBm
按键路径: 【功率】 —> [功率设置] —>[功率]

[:SOURce]:POWer:ALC[:STATe] <State>

功能描述: 设置和查询环路开关状态。
设置格式: [:SOURce]:POWer:ALC[:STATe] ON|OFF|1|0
查询格式: [:SOURce]:POWer:ALC[:STATe]?
返回值为 0 和 1，分别对应于数据类型为布尔型。取值如下:
参数说明:
<State> 数据类型为布尔型。取值如下:
OFF | 0: 环路状态为开环。
ON | 1: 环路状态为闭环。

举例: :POW:ALC 0 设置信号源环路为开环。

:POW:ALC? 查询信号源环路状态。
复位状态: 闭环
按键路径: 【功率】 —> [环路控制] —>[环路开关]

[:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet <PowOffset>

功能描述: 设置和查询功率偏置。
设置格式: [:SOURce]: POWer:OFFSet <PowOffset><dB>
查询格式: [:SOURce]: POWer:OFFSet?
返回信号源当前功率偏置值，单位为 dB。
参数说明:
<PowOffset> 数据类型为double类型。
设置范围: -500.0dB~500.0dB

举例: :POW:OFFS 10dB 设置信号源功率偏置考为 10dB。

:POW:OFFS? 查询信号源功率偏置值。

复位状态: 0dB

按键路径: 【功率】—> [功率设置] —>[功率偏置]

[:SOURce]:POWer:REFerence <PowRef>

功能描述: 设置和查询功率参考。

设置格式: [:SOURce]:POWer:REFerence <PowRef><dBm>

查询格式: [:SOURce]:POWer:REFerence?

返回信号源当前功率参考值。

参数说明:

<PowRef> 数据类型为double类型。

设置范围: -500.0dBm~500.0dBm

举例: :POW:REF 10dBm 设置信号源功率参考为 10dBm。

:POW:REF? 查询信号源功率参考。

复位状态: 0dBm

按键路径: 【功率】—> [功率设置] —>[功率参考]

[:SOURce]:POWer:REFerence:STATe <State>

功能描述: 设置和查询信号源功率参考开关。

设置格式: [:SOURce]:POWer:REFerence:STATe ON|OFF|1|0

查询格式: [:SOURce]:POWer:REFerence:STATe?

返回值为 0 和 1, 分别对应于功率参考状态为关闭状态(OFF)和打开状态(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:

OFF | 0: 功率参考关。

ON | 1: 功率参考开。

举例: :POW:REF:STAT 1 设置信号源功率参考开。

:POW:REF:STAT? 查询信号源功率参考开关状态。

复位状态: 关

按键路径: 【功率】—> [参考开关]

[:SOURce]:POWer:ALC:BANDwidth|BWIDth[:DATA] <Width>

功能描述: 设置和查询ALC带宽。

设置格式:

[:SOURce]:POWer:ALC:BANDwidth|BWIDth:DATA AUTO|1kHz|10kHz|100kHz

查询格式: [:SOURce]:POWer:ALC:BANDwidth|BWIDth:DATA?

返回值为 0、1 和 2，分别对应于环路带宽 1kHz、10kHz 和 100kHz。

参数说明:

<Width> 数据类型为离散类型。取值如下:

AUTO: 环路带宽设置为自动。

1kHz： 环路带宽设置为1kHz。

10kHz： 环路带宽设置为10kHz。

100kHz：环路带宽设置为100kHz。

举例: :POW:ALC:BAND 1kHz 设置环路带宽为 1kHz。

:POW:REF? 查询环路带宽。

复位状态: 10kHz

按键路径: 【功率】—> [环路控制] —>[环路带宽]

[:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO <State>

功能描述: 设置和查询ALC带宽手自动状态。

设置格式: [:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO ON|OFF|1|0

查询格式: [:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO?

返回值为 0 和 1，分别对应于手动选择 ALC 带宽(OFF)和自动选择 ALC 带宽(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:

OFF | 0: 手动选择ALC带宽。

ON | 1: 自动选择ALC带宽。

举例: :POW:ALC:BAND:AUTO 1 自动选择 ALC 带宽。

:POW:ALC:BAND:AUTO? 查询ALC带宽的选择模式。

复位状态: 自动

按键路径: 【功率】—> [环路控制] —>[环路带宽]

[:SOURce]:OUTPut:BLANking[:STATe] <State >

功能描述: 设置和查询输出消隐状态。

设置格式: [:SOURce]:OUTPut:BLANking[:STATe] ON|OFF|1|0

查询格式: [:SOURce]:OUTPut:BLANking[:STATe]?

返回值为 0 和 1，分别对应于输出消隐关(OFF)和输出消隐开(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:

OFF | 0: 输出消隐设置为关闭状态。

ON | 1: 输出消隐设置为打开状态。

举例: :OUTP:BLAN 1 输出消隐设置为打开。

:OUTP:BLAN? 查询输出消隐状态开关。

复位状态: 开

按键路径: 【功率】—> [环路控制] —>[输出消隐]

[:SOURce]:POWer:ALC:SEARch<Type>

功能描述: 激活功率搜索方式，这种方式将信号发生器稳幅在用户选定的输出功率上，然后断开ALC环并保持对内部调制器的驱动状态。

设置格式: [:SOURce]:POWer:ALC:SEARch ON|OFF|ONCE

查询格式: [:SOURce]:POWer:ALC:SEARch?

返回值为 0、1，分别对应于自动搜索关和自动搜索开。

参数说明:

<Type> 数据类型为离散类型。取值如下:

ON(1) 该命令使功率随着射频输出功率或频率的变化而自动进行搜索。

OFF(0) 该命令终止功率自动搜索程序。

ONCE 该命令在当前射频输出频率下执行一次功率搜索。

举例: :POWer:ALC:SEARch ON 设置功率搜索方式为自动。

:POWer:ALC:SEARch? 查询功率搜索方式。

复位状态: 10kHz

按键路径: 【功率】—> [环路控制] —>[功率搜索]

[:SOURce]:POWer:ATTenuation:AUTO <State>

功能描述: 设置和查询衰减器状态。

设置格式: [:SOURce]:POWer:ATTenuation:AUTO ON|OFF|1|0

查询格式: [:SOURce]:POWer:ATTenuation:AUTO?

返回值为 0 和 1，分别对应于衰减器手动状态(OFF)和衰减器自动状态(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:

OFF | 0: 衰减器设置为手动状态。

ON | 1: 衰减器设置为自动状态。

举例: :POW:ATT:AUTO 1 衰减器设置为自动状态。

:POW:ATT:AUTO? 查询衰减器设置状态。

复位状态: 开

按键路径: 【功率】—> [衰减控制] —>[衰减器]

[:SOURce]:POWer:ALC:Level <Level>

功能描述: 设置和查询ALC功率。

设置格式: [:SOURce]:POWer:ALC:Level <Level>

查询格式: [:SOURce]:POWer:ALC:Level?

返回信号源当前 ALC 功率值。

参数说明:

<PowRef> 数据类型为double类型。
设置范围: -25dBm~25dBm

举例: :POWer:ALC:Level 10dBm 设置信号源 ALC 功率值为 10dBm。

:POWer:ALC:Level? 查询信号源ALC功率值。

复位状态: 0dBm

按键路径: 【功率】—> [衰减控制] —>[ALC功率]

[:SOURce]:POWer:ATTenuation <Att>

功能描述: 设置和查询功率衰减值，设置时需衰减器处于手动状态。

设置格式: [:SOURce]:POWer:ATTenuation <Att><dB>

查询格式: [:SOURce]:POWer:ATTenuation?
返回信号源当前功率衰减值，单位为 dB。

参数说明:

<Att> 数据类型为int类型，且是10的整数倍。
设置范围: 0 ~ 110

举例: :POW:ATT 10dB 设置信号源功率衰减值为 10dB。

:POW:ATT? 查询信号源功率衰减值。

复位状态: 110dB

按键路径: 【功率】—> [衰减控制] —>[衰减]

[:SOURce]:CORRection:STATe <State>

功能描述: 设置和查询用户校准数据开关。

设置格式: [:SOURce]:CORRection:STATe ON|OFF|1|0

查询格式: [:SOURce]:CORRection:STATe?
返回值为 0 和 1，分别对应于用户校准数据开关为关闭状态(OFF)和打开状态(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:
:CORR:STAT 1 设置用户校准数据开关开。
:CORR:STAT? 查询用户校准数据开关状态。

举例: :CORR:STAT 1 设置用户校准数据开关开。

:CORR:STAT? 查询用户校准数据开关状态。

复位状态: 关

按键路径: 【功率】—>[用户校准]—>[用户补偿]

[:SOURce]:CORRection:STARt:FREQuency <Frequency>

功能描述: 设置和查询用户校准起始填充频率。
设置格式:
[:SOURce]:CORRection:STARt:FREQuency <Frequency><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:CORRection:STARt:FREQuency?
返回用户校准起始填充频率值，单位为 Hz。
参数说明:
<Frequency> 数据类型为double型。
设置范围：1MHz~20GHz（1433D）/26.5GHz（1433E）/40GHz（1433F）/50GHz（1433H）

举例: :CORR:STAR:FREQ 10MHz 设置信号源用户校准起始填充频率为 10MHz。
:CORR:STAR:FREQ? 查询信号源用户校准起始填充频率。
复位状态: 1MHz
按键路径: 【功率】—>[用户校准]—>[起始填充频率]

[:SOURce]:CORRection: STOP:FREQuency <Frequency>

功能描述: 设置和查询用户校准终止填充频率。
设置格式:
[:SOURce]:CORRection:STOP:FREQuency <Frequency><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:CORRection:STOP:FREQuency?
返回用户校准终止填充频率值，单位为 Hz。
参数说明:
<Frequency> 数据类型为double型。
设置范围：1MHz~20GHz（1433D）/26.5GHz（1433E）/40GHz（1433F）/50GHz（1433H）

举例: :CORR:STOP:FREQ 10MHz 设置信号源用户校准终止填充频率为 10MHz。
:CORR:STOP:FREQ? 查询信号源用户校准终止填充频率。
复位状态: 1MHz
按键路径: 【功率】—>[用户校准]—>[终止填充频率]

[:SOURce]:CORRection:STARt:AMPLitude <Ampl>

功能描述: 设置和查询用户校准起始补偿功率。
设置格式: [:SOURce]:CORRection:STARt:AMPLitude <Ampl><dB>
查询格式: [:SOURce]:CORRection:STARt:AMPLitude?
返回用户校准起始补偿功率值，单位为 dB。
参数说明:
<Ampl> 数据类型为double型。
设置范围：-3dB~3dB

举例: :CORR:STAR:AMPL 1dB 设置用户校准起始补偿功率为 1dB。

:CORR:STAR:AMPL? 查询用户校准起始补偿功率。

复位状态: 0dBm

按键路径: 【功率】—>[用户校准]—>[起始功率补偿]

[:SOURce]:CORRection:STOP:AMPLitude <Ampl>

功能描述: 设置和查询用户校准终止补偿功率。

设置格式: [:SOURce]:CORRection:STOP:AMPLitude <Ampl><dB>

查询格式: [:SOURce]:CORRection:STOP:AMPLitude?

返回用户校准终止补偿功率值，单位为 dB。

参数说明:

<Ampl> 数据类型为double型。

设置范围：-3dB~3dB

举例: :CORR:STOP:AMPL 1dB 设置用户校准终止补偿功率为 1dB。

:CORR:STOP:AMPL? 查询用户校准终止补偿功率。

复位状态: 0dBm

按键路径: 【功率】—>[用户校准]—>[终止功率补偿]

[:SOURce]:CORRection:POINts <PointsNum>

功能描述: 设置和查询用户校准填充点数。

设置格式: [:SOURce]:CORRection:POINts <PointsNum>

查询格式: [:SOURce]:CORRection:POINts?

返回用户校准填充点数。

参数说明:

<PointsNum> 数据类型为int型。

设置范围：1~999

举例: :CORR:POIN 100 设置用户校准填充点数 100。

:CORR:POIN? 查询用户校准填充点数。

复位状态: 1

按键路径: 【功率】—>[用户校准]—>[填充点数]

[:SOURce]:CORRection:FILL

功能描述: 填充用户校准数据，无参数，只设置。

设置格式: [:SOURce]:CORRection:FILL

举例: :CORR:FILL 填充用户校准数据。

按键路径: 【功率】—>[用户校准]—>[确认填充]

[:SOURce]:CORRection:LOAD

功能描述: 加载用户校准数据，无参数，只设置。
设置格式: [:SOURce]:CORRection:LOAD

举例: :CORR:LOAD 加载用户校准数据。

按键路径: 【功率】—>[用户校准]—>[加载数据]

[:SOURce]:CORRection:STORe

功能描述: 保存用户校准数据，无参数，只设置。
设置格式: [:SOURce]:CORRection:STORe

举例: : CORR:STOR 保存用户校准数据。

按键路径: 【功率】—>[用户校准]—>[存储数据]

3.3.4 Modulation 子系统

用于控制射频输出信号的调制功能，包括:

- [:SOURce]:PULM:STATe.....49
- [:SOURce]:PULM:SOURce.....50
- [:SOURce]:PULM:INTernal:PWIDth.....50
- [:SOURce]:PULM:INTernal:PERiod.....51
- [:SOURce]:PULM:INTernal:DELay.....51
- [:SOURce]:PULM:POLarity.....51
- [:SOURce]:AM:STATe.....52
- [:SOURce]:AM:INTernal:FREQuency.....52
- [:SOURce]:AM[:DEPTh]:LINear.....52
- [:SOURce]:AM[:DEPTh]:EXPonential.....53
- [:SOURce]:AM:TYPE.....53
- [:SOURce]:AM:MODE.....54
- [:SOURce]:FM:STATe.....54
- [:SOURce]:FM:INTernal:FREQuency.....55
- [:SOURce]:FM[:DEViation].....55
- [:SOURce]:PM:STATe <State>.....55
- [:SOURce]:PM:INTernal:FREQuency <PMFreq>.....56
- [:SOURce]:PM:DEViation <PhaseOffset>.....56

[:SOURce]:PULM:STATe <State>

功能描述: 设置和查询脉冲调制开关状态。
设置格式: [:SOURce]:PULM:STATe ON|OFF|1|0

查询格式: [:SOURce]:PULM:STATe?
返回值为 0 和 1，分别对应于脉冲调制开关为关闭状态(OFF)和打开状态(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:
OFF | 0: 脉冲调制开关关。
ON | 1: 脉冲调制开关开。

举例: :PULM:STAT 1 设置脉冲调制开关为开。

:PULM:STAT? 查询信号源脉冲调制开关状态。

复位状态: 关

按键路径: 【调制】 —> [脉冲调制]—> [调制开关]

[:SOURce]:PULM:SOURce <Mod>

功能描述: 设置和查询脉冲输入类型。

设置格式: [:SOURce]:PULM:SOURce
AUTO|TRIG|GATE|DPULse|EXTernal|PTRain|STAGger|JITTered|SLIDing

查询格式: [:SOURce]:PULM:SOURce?
返回字符串，分别对应于脉冲输入设置为内部自动(AUTO)、内部触发(TRIG)、内部门控(GATE)、双脉冲(DPULse)、外部输入(EXTernal)、脉冲串(PTRain)、重频参差（STAGger）、重频抖动（JITTered）、重频滑变（SLIDing）。

参数说明:

<Mod> 数据类型为离散型。取值如下:
AUTO: 脉冲输入设置为内部自动。
TRIG: 脉冲输入设置为内部触发。
GATE: 脉冲输入设置为内部门控。
DPULse: 脉冲输入设置为双脉冲。
EXTernal: 脉冲输入设置为外部。
PTRain: 脉冲串。
STAGger: 重频参差。
JITTered: 重频抖动。
SLIDing: 重频滑变。

举例: :PULM:SOUR AUTO 设置脉冲输入为内部自动。

:PULM:SOUR? 查询信号源脉冲输入设置。

复位状态: 内部自动

按键路径: 【调制】 —> [脉冲调制]—> [脉冲设置]—> [脉冲输入]

[:SOURce]:PULM:INTernal:PWIDth <Pwidth>

功能描述: 设置和查询内部脉冲发生器的脉冲宽度。

设置格式: [:SOURce]:PULM:INTernal:PWIDth <Pwidth><ns|us|ms|s>

查询格式: [:SOURce]:PULM:INTernal:PWIDth?
返回内部脉冲发生器的脉冲宽度，单位为 ns。

参数说明:

<Pwidth> 数据类型为int型，是20ns的整数倍。
设置范围：80ns~60s-20ns

举例: :PULM:INT:PWID 100ns 脉冲发生器的脉冲宽度设为 100ns。

:PULM:INT:PWID? 查询脉冲发生器的脉冲宽度。

复位状态: 10us

按键路径: 【调制】—> [脉冲调制]—> [脉冲宽度]

[:SOURce]:PULM:INTernal:PERiod <Period>

功能描述: 设置和查询内部脉冲发生器的脉冲周期。

设置格式: :SOURce]:PULM:INTernal:PERiod <Period><ns|us|ms|s>

查询格式: [:SOURce]:PULM:INTernal:PERiod?
返回内部脉冲发生器的脉冲周期，单位为 ns。

参数说明:

<Period> 数据类型为int型，是20ns的整数倍。
设置范围：100ns~60s

举例: :PULM:INT:PER 100ns 脉冲发生器的脉冲周期设为 100ns。

:PULM:INT:PER? 查询脉冲发生器的脉冲周期。

复位状态: 20us

按键路径: 【调制】—> [脉冲调制]—> [脉冲周期]

[:SOURce]:PULM:INTernal:DELay <Delay>

功能描述: 设置和查询内部脉冲发生器的脉冲延时。

设置格式: [:SOURce]:PULM:INTernal:DELay <Delay><ns|us|ms|s>

查询格式: [:SOURce]:PULM:INTernal:DELay?
返回内部脉冲发生器的脉冲延时，单位为 ns。

参数说明:

<Delay> 数据类型为int型，是20ns的整数倍。
设置范围：0ns~60s

举例: :PULM:INT:DEL 10ns 脉冲发生器的脉冲延时设为 10ns。

:PULM:INT:DEL? 查询脉冲发生器的脉冲延时。

复位状态: 0ns

按键路径: 【调制】—> [脉冲调制]—> [脉冲延时]

[:SOURce]:PULM:POLarity <Mod>

功能描述: 设置和查询脉冲输入反相状态。
设置格式: [:SOURce]:PULM:POLarity INVerted|NORMal
查询格式: [:SOURce]:PULM: POLarity?
返回值为 0 和 1，分别对应于脉冲输入反相为关闭状态(NORMal)和打开状态(INVerted)。
参数说明:
<Mod> 数据类型为离散型。取值如下:
INVerted: 脉冲输入反相设置为打开状态。
NORMal: 脉冲输入反相设置为关闭状态。

举例: :PULM:POL INVerted 设置脉冲输入反相状态为开。

:PULM:POL? 查询信号源脉冲输入反相状态。
复位状态: 关
按键路径: 【调制】 —> [脉冲调制]—> [输入反相]

[:SOURce]:AM:STATe <State>

功能描述: 设置和查询幅度调制开关状态。
设置格式: [:SOURce]:AM:STATe ON|OFF|1|0
查询格式: [:SOURce]:AM:STATe?
返回值为 0 和 1, 分别对应于幅度调制开关为关闭状态(OFF)和打开状态(ON)。
参数说明:
<State> 数据类型为布尔型。取值如下:
OFF | 0: 幅度调制开关关。
ON | 1: 幅度调制开关开。

举例: :AM:STAT 1 设置幅度调制开关为开。

:AM:STAT? 查询信号源幅度调制开关状态。
复位状态: 关
按键路径: 【调制】 —> [幅度调制]—> [调制开关]

[:SOURce]:AM:INTernal:FREQuency <AMFreq>

功能描述: 设置和查询幅度调制率。
设置格式: [:SOURce]:AM:INTernal:FREQuency <AMFreq><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:AM:INTernal:FREQuency?
返回信号源当前幅度调制率，单位为 Hz。
参数说明:
<AMFreq> 数据类型为double型。
设置范围：0Hz~50kHz

举例: :AM:INT:FREQ 1kHz 设置信号源幅度调制率为 1kHz。

:AM:INT:FREQ? 查询信号源幅度调制率。

复位状态: 1kHz

按键路径: 【调制】 —> [幅度调制] —> [调制率]

[:SOURce]:AM[:DEPTh]:LINear <Depth>

功能描述: 设置和查询线性调制深度。

设置格式: [:SOURce]:AM[:DEPTh]:LINear <Depth><%>

查询格式: [:SOURce]:AM[:DEPTh]:LINear?

返回信号源当前线性调制深度。

参数说明:

<Depth> 数据类型为double型。

设置范围：0~100%

举例: :AM:LIN 30% 设置幅度调制的线性调制深度为 30%。

:AM:LIN? 查询幅度调制的线性调制深度。

复位状态: 30%

按键路径: 【调制】 —> [幅度调制] —> [线性调制深度]

[:SOURce]:AM[:DEPTh]:EXPonential <Depth>

功能描述: 设置和查询指数调制深度。

设置格式: [:SOURce]:AM[:DEPTh]:EXPonential <Depth><dB>

查询格式: [:SOURce]:AM[:DEPTh]:EXPonential?

返回信号源当前指数调制深度。

参数说明:

<Depth> 数据类型为double型。

设置范围：0~50dB

举例: :AM:EXP 10dB 设置幅度调制的指数调制深度为 10dB。

:AM:EXP? 查询幅度调制的指数调制深度。

复位状态: 10dB

按键路径: 【调制】 —> [幅度调制] —> [指数调制深度]

[:SOURce]:AM:TYPE <Type>

功能描述: 设置和查询调幅类型。

设置格式: [:SOURce]:AM:TYPE EXPonential|LINear

查询格式: [:SOURce]:AM:TYPE?

返回值为 0 和 1，分别对应于调幅类型为线性调幅和指数调幅。

参数说明:

<Type> 离散数据。取值如下:
EXPonential: 调幅类型为指数调幅。
LINear: 调幅类型为线性调幅。

举例: :AM:TYPE LINear 设置调幅类型为线性调幅。

:AM:TYPE? 查询调幅类型。

复位状态: 线性

按键路径: 【调制】 —> [幅度调制] —> [调幅类型]

[:SOURce]:AM:MODE <Mod>

功能描述: 设置和查询深度调幅开关。

设置格式: [:SOURce]:AM:MODE DEEP|NORMal

查询格式: [:SOURce]:AM:MODE?

返回值为 0 和 1, 分别对应于深度调幅开关为关闭状态(NORMal)和打开(DEEP)。

参数说明:

<Mod> 数据类型为离散型。取值如下:
NORMal: 深度调幅开关为关闭状态。
DEEP: 深度调幅开关为打开状态。

举例: :AM:MODE NORMal 设置深度调幅开关为关。

:AM:MODE? 查询深度调幅开关。

复位状态: 关

按键路径: 【调制】 —> [脉冲调制]—> [深度调幅]

[:SOURce]:FM:STATe <State>

功能描述: 设置和查询频率调制开关状态。

设置格式: [:SOURce]:FM:STATe ON|OFF|1|0

查询格式: [:SOURce]:FM:STATe?

返回值为 0 和 1, 分别对应于频率调制开关为关闭状态(OFF)和打开状态(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:
OFF | 0: 频率调制开关关。
ON | 1: 频率调制开关开。

举例: :FM:STAT 1 设置频率调制开关为开。

:FM:STAT? 查询频率调制开关状态。

复位状态: 关

按键路径: 【调制】 —> [频率调制]—> [调制开关]

[:SOURce]:FM:INTernal:FREQuency <FMFreq>

功能描述: 设置和查询频率调制率。
设置格式: [:SOURce]:FM:INTernal:FREQuency <FMFreq><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:FM:INTernal:FREQuency?
返回信号源当前频率调制率，单位为 Hz。
参数说明:
<FMFreq> 数据类型为double型。
设置范围：0Hz~20kHz

举例: :FM:INT:FREQ 10kHz 设置信号源频率调制率为 10kHz。

:FM:INT:FREQ? 查询信号源频率调制率。
复位状态: 1kHz
按键路径: 【调制】—> [频率调制] —> [调制率]

[:SOURce]:FM:DEViation <FreqOffset>

功能描述: 设置和查询调频频偏。
设置格式: [:SOURce]:FM:DEViation <FreqOffset><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:FM:DEViation?
返回信号源当前调频频偏，单位为 Hz。
参数说明:
<FreqOffset> 数据类型为double型。
设置范围：0Hz~4MHz

举例: :FM:DEV 1kHz 设置信号源调频频偏为 1kHz。

:FM:DEV? 查询信号源调频频偏值。
复位状态: 10kHz
按键路径: 【调制】—> [频率调制] —> [调频频偏]

[:SOURce]:PM:STATe <State>

功能描述: 设置和查询相位调制开关状态。
设置格式: [:SOURce]:PM:STATe ON|OFF|1|0
查询格式: [:SOURce]:PM:STATe?
返回值为 0 和 1，分别对应于相位调制开关为关闭状态(OFF)和打开状态(ON)。
参数说明:
<State> 数据类型为布尔型。取值如下:
OFF | 0: 相位调制开关关。
ON | 1: 相位调制开关开。
举例: :PM:STAT 1 设置相位调制开关为开。
:PM:STAT? 查询相位调制开关状态。
复位状态: 关

按键路径: 【调制】 —> [相位调制]—> [相位调制开关]

[:SOURce]:PM:INTernal:FREQuency <PMFreq>

功能描述: 设置和查询相位调制率。
设置格式: [:SOURce]:PM:INTernal:FREQuency <PMFreq><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:PM:INTernal:FREQuency?
返回信号源当前相位调制率，单位为Hz。
参数说明:
<PMFreq> 数据类型为double型。
设置范围：0Hz~20kHz
举例: :PM:INT:FREQ 10kHz 设置信号源相位调制率为10kHz。
:PM:INT:FREQ? 查询信号源相位调制率。
复位状态: 1kHz
按键路径: 【调制】 —> [相位调制] —> [调制率]

[:SOURce]:PM:DEViation <PhaseOffset>

功能描述: 设置和查询调相相偏。
设置格式: [:SOURce]:PM:DEViation <PhaseOffset><rad>
查询格式: [:SOURce]:PM:DEViation?
返回信号源当前调相相偏，单位为rad。
参数说明:
<PhaseOffset> 数据类型为double类型。
不同频段设置调相相偏时，对应不同的相偏范围:

频率范围	调相相偏范围
[1MHz, 2.35GHz)	[0rad, 12.5rad]
[2.35GHz, 2.5GHz)	[0rad, 3.125rad]
[2.5GHz, 5GHz)	[0rad, 6.25rad]
[5GHz, 10GHz]	[0rad, 12.5rad]
(10GHz, 20GHz]	[0rad, 25rad]
(20GHz, 40GHz]	[0rad, 50rad]
(40GHz, 50GHz]	[0rad, 100rad]

举例: :PM:DEV 1 设置信号源调相相偏为1rad。
:PM:DEV? 查询信号源调相相偏值。
复位状态: 1rad
按键路径: 【调制】 —> [相位调制] —> [调相相偏]

3.3.5 Sweep 系统

用于控制扫描功能，包括:

- [:SOURce]:SWEep:MODE.....57

- [:SOURce]:LIST:TYPE 58
- :TRIGger[:SEQuence]:SOURce 58
- [:SOURce]:LIST:TRIGger:SOURce 58
- [:SOURce]:LIST:DIRection 59
- :INITiate:CONTinuous[:ALL] 59
- :INITiate[:IMMediate][:ALL] 59
- [:SOURce]:LIST:MODE 60
- [:SOURce]:LIST:MANual 60
- [:SOURce]:FREQuency:STARt 60
- [:SOURce]:FREQuency:STOP 61
- [:SOURce]:POWer:STARt 61
- [:SOURce]:POWer:STOP 61
- [:SOURce]:SWEep:POINts 62
- [:SOURce]:SWEep:DWELl 62
- [:SOURce]:LIST:FREQuency:DATA 63
- [:SOURce]:LIST:FREQuency:POINts? 63
- [:SOURce]:LIST:POWer:DATA 63
- [:SOURce]:LIST:POWer:POINts? 64
- [:SOURce]:LIST:DWELl:DATA 64
- [:SOURce]:LIST:DWELl:POINts? 64
- :MEMory:STORe:LIST 64
- :MEMory:LOAD:LIST 65

[:SOURce]:SWEep:MODE <Mod>

功能描述: 设置和查询扫描模式。

设置格式: [:SOURce]:SWEep:MODE OFF|FREQ|AMP|FREQAMP

查询格式: [:SOURce]:SWEep:MODE?

返回值为 0 到 3，分别对应于扫描模式为关 (0)、频率扫描 (1)、功率扫描 (2) 和频率功率扫描 (3)。

参数说明:

<Mod> 数据类型为离散型。取值如下:

OFF:	扫描模式设置为关。
FREQ:	扫描模式设置为频率扫描。
AMP:	扫描模式设置为功率扫描。
FREQAMP:	扫描模式设置为频率功率扫描。

举例: :SWE:MODE FREQ 设置扫描模式为频率扫描。

:SWE:MODE? 查询扫描模式。

复位状态: 关

按键路径: 【扫描】 —> [扫描模式]—> [扫描]

[:SOURce]:LIST:TYPE <Type>

功能描述: 设置和查询扫描类型。
设置格式: [:SOURce]:LIST:TYPE STEP|LIST
查询格式: [:SOURce]:LIST:TYPE?
返回值为 0 和 1，分别对应于扫描类型为步进扫描和列表扫描。
参数说明:
<Type> 离散数据。取值如下:
STEP: 扫描类型设置为步进扫描。
LIST: 扫描类型设置为列表扫描。

举例: :LIST:TYPE STEP 设置扫描类型为步进扫描。

:LIST:TYPE? 查询扫描类型。
复位状态: 步进
按键路径: 【扫描】—> [扫描模式]—> [扫描类型]

:TRIGger[:SEQuence]:SOURce <Type>

功能描述: 设置和查询扫描触发类型。
设置格式: :TRIGger[:SEQuence]:SOURce IMMediate|KEY
查询格式: :TRIGger[:SEQuence]:SOURce?
返回值为 0 或 1，分别对应于扫描触发类型为自动和手动。
参数说明:
<Type> 离散数据。取值如下:
IMMediate: 扫描触发类型设置为自动。
KEY: 扫描触发类型设置为手动。

举例: :TRIG:SOUR IMMediate 设置扫描触发类型为自动。

:TRIG:SOUR? 查询扫描触发类型。
复位状态: 自动
按键路径: 【扫描】—> [扫描模式]—> [扫描触发]

[:SOURce]:LIST:TRIGger:SOURce <Type>

功能描述: 设置和查询点触发类型。
设置格式: [:SOURce]:LIST:TRIGger:SOURce IMMediate|KEY
查询格式: [:SOURce]:LIST:TRIGger:SOURce?
返回值为 0 或 1，分别对应于点触发类型为自动和手动。
参数说明:
<Type> 离散数据。取值如下:
IMMediate: 点触发类型设置为自动。
KEY: 点触发类型设置为手动。

举例: :LIST:TRIG:SOUR IMMediate 设置点触发类型为自动。

:LIST:TRIG:SOUR? 查询点触发类型。

复位状态: 自动

按键路径: 【扫描】—> [扫描模式]—> [点触发]

[:SOURce]:LIST:DIRection <Type>

功能描述: 设置和查询扫描方向。

设置格式: [:SOURce]:LIST:DIRection UP|DOWN

查询格式: [:SOURce]:LIST:DIRection?

返回值为 0 和 1，分别对应于扫描方向为正向和反向。

参数说明:

<Type> 离散数据。取值如下:

UP: 扫描方向为正向。

DOWN: 扫描方向为反向。

举例: :LIST:DIR UP 设置扫描方向为正向。

:LIST:DIR? 查询扫描方向。

复位状态: 正向

按键路径: 【扫描】—> [扫描模式]—> [扫描方向]

:INITiate:CONTinuous[:ALL] <State>

功能描述: 设置和查询扫描控制类型。

设置格式: :INITiate:CONTinuous[:ALL] ON|OFF|1|0

查询格式: :INITiate:CONTinuous[:ALL]?

返回值为 0 和 1，分别对应于扫描控制类型设置为单次(OFF)和连续(ON)。

参数说明:

<State> 数据类型为布尔型。取值如下:

OFF | 0: 扫描控制类型设置为单次。

ON | 1: 扫描控制类型设置为连续。

举例: :INIT:CONT 1 扫描控制类型设置为单次。

:INIT:CONT? 查询扫描控制类型。

复位状态: 连续

按键路径: 【扫描】—> [扫描执行]—> [扫描控制]

:INITiate[:IMMediate][:ALL]

功能描述: 执行单次扫描。

设置格式: :INITiate[:IMMediate][:ALL]

举例: :INIT 执行单次扫描。

复位状态: 连续
说明: 仅设置。

[:SOURce]:LIST:MODE <Mod>

功能描述: 设置和查询扫描手动模式状态。
设置格式: [:SOURce]:LIST:MODE AUTO|MANual
查询格式: [:SOURce]:LIST:MODE?
返回值为 0 和 1，分别对应于扫描手动模式设置为关(AUTO)和开(MANual)。
参数说明:
<Mod> 数据类型为离散型。取值如下:
AUTO: 扫描手动模式设置为关。
MANual: 扫描手动模式设置为开。

举例: :LIST:MODE AUTO 设置扫描手动模式为关。

:LIST:MODE? 查询扫描手动模式开关状态。
复位状态: 关
按键路径: 【扫描】—> [扫描执行]—> [手动模式]

[:SOURce]:LIST:MANual <PointsNum>

功能描述: 设置和查询手动扫描点数。
设置格式: [:SOURce]:LIST:MANual <PointsNum>
查询格式: [:SOURce]:LIST:MANual?
返回信号源当前手动扫描点数。
参数说明:
<PointsNum> 数据类型为int类型。
设置范围: 1~160

举例: :LIST:MAN 1000 设置手动扫描点数为 1000。

:LIST:MAN? 查询手动扫描点数。
复位状态: 1
按键路径: 【扫描】—> [扫描执行]—> [手动点]

[:SOURce]:FREQuency:STARt <Frequency>

功能描述: 设置和查询步进扫描起始频率。
设置格式: [:SOURce]:FREQuency:STARt <Frequency><Hz|kHz|MHz|GHz>
查询格式: [:SOURce]:FREQuency:STARt?
返回步进扫描起始频率，单位为 Hz。
参数说明:

<Frequency>　数据类型为double类型。
设置范围：　1MHz~20GHz（1433D）/26.5GHz（1433E）/40GHz（1433F）/50GHz（1433H）

举例：　:FREQ:STAR 10MHz　设置步进扫描起始频率为 10MHz。

:FREQ:STAR?　查询步进扫描起始频率。

复位状态：1GHz

按键路径：【扫描】—> [步进扫描]—> [起始频率]

[:SOURce]:FREQuency: STOP<Frequency>

功能描述：设置和查询步进扫描终止频率。

设置格式：[:SOURce]:FREQuency:STOP　<Frequency><Hz|kHz|MHz|GHz>

查询格式：[:SOURce]:FREQuency:STOP?
返回步进扫描终止频率，单位为 Hz。

参数说明：

<Frequency>　数据类型为double类型。
设置范围：　1MHz~20GHz（1433D）/26.5GHz（1433E）/40GHz（1433F）/50GHz（1433H）

举例：　:FREQ:STOP 10MHz　设置步进扫描终止频率为 10MHz。

:FREQ:STOP?　查询步进扫描终止频率。

复位状态：20GHz（1433D）/26.5GHz（1433E）/ 40GHz（1433F）/50GHz（1433H）

按键路径：【扫描】—> [步进扫描]—> [终止频率]

[:SOURce]:POWer:STARt <Ampl>

功能描述：设置和查询步进扫描起始功率。

设置格式：[:SOURce]:POWer:STARt <Ampl><dBm>

查询格式：[:SOURce]:POWer:STARt?
返回步进扫描起始功率。

参数说明：

<Ampl>　数据类型为double类型。
设置范围：　-135.0dBm~25.0dBm

举例：　:POW:STAR 10dBm　设置步进扫描起始功率为 10dBm。

:POW:STAR?　查询步进扫描起始功率。

复位状态：0dBm

按键路径：【扫描】—> [步进扫描]—> [起始功率]

[:SOURce]:POWer: STOP <Ampl>

功能描述：设置和查询步进扫描终止功率。

设置格式: [:SOURce]:POWer:STOP <Ampl><dBm>
查询格式: [:SOURce]:POWer:STOP?
返回步进扫描终止功率。
参数说明:
<Ampl> 数据类型为double类型。
设置范围: -135.0dBm~25.0dBm

举例: :POW:STOP 10dBm 设置步进扫描终止功率为 10dBm。

:POW:STOP? 查询步进扫描终止功率。
复位状态: 0dBm
按键路径: 【扫描】—> [步进扫描]—> [终止功率]

[:SOURce]:SWEep:POINts <PointsNum>

功能描述: 设置和查询步进扫描扫描点数。
设置格式: [:SOURce]:SWEep:POINts <PointsNum>
查询格式: [:SOURce]:SWEep:POINts?
返回信号源当前步进扫描点数。
参数说明:
<PointsNum> 数据类型为int类型。
设置范围: 2~1601

举例: :SWE:POIN 100 设置步进扫描点数为 100。

:SWE:POIN? 查询步进扫描点数。
复位状态: 2
按键路径: 【扫描】—> [步进扫描]—> [扫描点数]

[:SOURce]:SWEep:DWELl <Val>

功能描述: 设置和查询步进扫描中每个点的驻留时间。
设置格式: [:SOURce]:SWEep:DWELl <Val><ns|us|ms|s>
查询格式: [:SOURce]:SWEep:DWELl?
返回信号源当前步进扫描中每个点的驻留时间，单位是 ns。
参数说明:
<Val> 数据类型为double类型。
设置范围: 10ms~100s

举例: :SWE:DWEL 1.0s 设置步进扫描中每个点的驻留时间为 1s。

:SWE:DWEL? 查询步进扫描中每个点的驻留时间。
复位状态: 10ms
按键路径: 【扫描】—> [步进扫描]—> [驻留时间]

[:SOURce]:LIST:FREQuency:DATA <Frequency>

功能描述: 设置和查询列表扫描频率参数，可多参数。

设置格式: [:SOURce]:LIST:FREQuency:DATA <Frequency><freq unit>{,<Frequency><freq unit>}

查询格式: [:SOURce]:LIST:FREQuency:DATA?
返回信号源当前列表扫描频率参数，单位是 Hz。

参数说明:

<Frequency> 数据类型为double类型。
设置范围: 1MHz~20GHz（1433D）/26.5GHz（1433E）/40GHz（1433F）/50GHz（1433H）

举例: :LIST:FREQuency:DATA 1GHz,2GHz 设置列表扫描前两个点的频率为 1GH 和 2GHz。

:LIST:FREQuency:DATA? 查询列表扫描所有点的频率。

按键路径: 【扫描】—> [列表扫描]

[:SOURce]:LIST:FREQuency:POINts?

功能描述: 查询列表扫描频率点数。

查询格式: [:SOURce]:LIST:FREQuency: POINts?
返回列表扫描频率点数。

举例: :LIST:FREQuency: POINts? 查询列表扫描频率点数。

说明: 仅查询.

[:SOURce]:LIST:POWer:DATA <Ampl>

功能描述: 设置和查询列表扫描功率参数，可多参数。

设置格式: [:SOURce]:LIST:POWer:DATA <Ampl><dBm>{,<Ampl><dBm>}

查询格式: [:SOURce]:LIST:POWer:DATA?
返回信号源当前列表扫描功率参数，单位是 dBm。

参数说明:

<Ampl> 数据类型为double类型。
设置范围: -135dBm~25dBm

举例: :LIST:POWer:DATA 1dBm,2dBm 设置列表扫描前两个点的功率为 1dBm 和 2dBm。

:LIST:POWer:DATA? 查询列表扫描所有点的功率。

复位状态: -120dBm

按键路径: 【扫描】—> [列表扫描]

[:SOURce]:LIST:POWer:POINts?

功能描述: 查询列表扫描功率点数。
查询格式: [:SOURce]:LIST: POWer: POINts?
返回列表扫描功率点数。

举例: :LIST: POWer: POINts? 查询列表扫描功率点数。

说明: 仅查询。

[:SOURce]:LIST:DWELl:DATA <Val>

功能描述: 设置和查询列表扫描驻留时间参数，可多参数。
设置格式: [:SOURce]:LIST:DWELl:DATA <Val><time unit>{,<val><time unit>}
查询格式: [:SOURce]: LIST:DWELl:DATA?
返回信号源当前列表扫描驻留时间参数，单位是 ns。
参数说明:
<Val> 数据类型为double类型。
设置范围：10ms~100s

举例: :LIST:DWELl:DATA 100ms,100ms 设置列表扫描前两个点的驻留时间为 100ms。

:LIST:DWELl:DATA? 查询列表扫描所有点的驻留时间。
复位状态: 300ms
按键路径: 【扫描】—> [列表扫描]

[:SOURce]:LIST:DWELl:POINts?

功能描述: 查询列表扫描驻留时间点数。
查询格式: [:SOURce]:LIST: DWELl: POINts?
返回列表扫描驻留时间点数。

举例: :LIST: DWELl: POINts? 查询列表扫描驻留时间点数。

说明: 仅查询。

:MEMory:STORe:LIST<Name>

功能描述: 保存列表扫描文件。
设置格式: :MEMory:STORe:LIST< Name>
参数说明:
<Name> string型。

举例: :MEMory:STORe:LIST“data“ 保存列表扫描文件,文件名为“data“。

说明: 仅设置。
按键路径: 【扫描】—> [列表扫描] —> [存储]

:MEMory:LOAD:LIST<Name>

功能描述: 调用列表扫描文件。
设置格式: :MEMory:LOAD:LIST< Name>
参数说明:
<Name> string型。

举例: :MEMory:LOAD:LIST“data“ 调用列表扫描文件,文件名为“data“。

说明: 仅设置。
按键路径: 【扫描】—> [列表扫描] —> [调用]

3.3.6 其他命令

用于系统功能开关设置和查询，包括:

- [:SOURce]:OUTPut[:STATe]……65
- [:SOURce]:OUTPut:MODulation[:STATe]……65

[:SOURce]:OUTPut[:STATe] <State>

功能描述: 设置和查询射频功能开关。
设置格式: [:SOURce]:OUTPut[:STATe] ON|OFF|1|0
查询格式: [:SOURce]:OUTPut[:STATe]?
返回值为 0 和 1，分别对应于射频功能关(OFF)和射频功能开(ON)。
参数说明:
<State> 数据类型为布尔型。取值如下:
ON | 1: 射频功能开。
OFF | 0: 射频功能关。

举例: :OUTP 1 打开射频功能。

:OUTP? 查询射频功能开关状态。
复位状态: 关
按键路径: 【射频开关】

[:SOURce]:OUTPut:MODulation[:STATe] <State>

功能描述: 设置和查询调制功能开关。
设置格式: [:SOURce]:OUTPut:MODulation[:STATe] ON|OFF|1|0
查询格式: [:SOURce]:OUTPut:MODulation[:STATe]?
返回值为 0 和 1，分别对应于调制功能关(OFF)和调制功能开(ON)。
参数说明:
<State> 数据类型为布尔型。取值如下:
ON | 1: 调制功能开。

OFF | 0: 调制功能关。

举例: :OUTP:MOD 1 打开调制功能。

:OUTP:MOD? 查询调制功能开关状态。

复位状态: 关

按键路径: 【调制开关】

3.4 仪器兼容命令

1433 系列信号发生器兼容 1431 系列信号发生器的 SCPI 命令，简要说明如下表（表 3.1），具体命令说明请参阅章节“3.3 仪器子系统命令”。

表 3.2 1433 兼容命令说明

SCPI 命令	兼容仪器型号	支持	说明
[:SOURce]:FREQuency[:CW|FIXed]	1431/1431A	是	范围为：10MHz~18GHz(1431) 和 250kHz~4GHz(1431A)
[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]	1431/1431A	是	范围为：-125dBm~25dBm(1431) 和 -130dBm~25dBm(1431A)
[:SOURce]:PULM:INTernal:PWIDth	1431/1431A	是	范围为：10ns~(60s-10ns)
[:SOURce]:PULM:INTernal:PERiod	1431/1431A	是	范围为：20ns~60s
[:SOURce]:PULM:INTernal:DELay	1431/1431A	是	范围为：-60s~60s
[:SOURce]:AM:INTernal:FREQuency	1431/1431A	是	范围为：0~50kHz
[:SOURce]:AM[:DEPTh]:EXPonential	1431/1431A	是	范围为：0~40dB
[:SOURce]:AM[:DEPTh]:LINear	1431/1431A	是	
[:SOURce]:FM:INTernal:FREQuency	1431/1431A	是	范围为：0~70kHz
[:SOURce]:FM[:DEViation]	1431/1431A	是	范围为：0~800kHz
[:SOURce]:FREQuency:STARt	1431/1431A	是	范围为：10MHz~18GHz(1431) 和 250kHz~4GHz(1431A)
[:SOURce]:FREQuency:STOP	1431/1431A	是	范围为：10MHz~18GHz(1431) 和 250kHz~4GHz(1431A)
[:SOURce]:POWer:STARt	1431/1431A	是	范围为：-125dBm~25dBm(1431) 和 -130dBm~25dBm(1431A)

表 3.2（续 1） 1433 兼容命令说明

SCPI 命令	兼容仪器型号	支持	说明
[:SOURce]:POWer:STOP	1431/1431A	是	范围为: -125dBm~25dBm(1431) 和 -130dBm~25dBm(1431A)
[:SOURce]:SWEep:DWELl	1431/1431A	是	范围为：10ms~60s

- 上表中支持列项选项包含：是、否和接受。
  “是” 说明仪器支持该条 SCPI 命令，实施正常语法解析、功能设置和查询。
  “否” 说明仪器不支持该条 SCPI 命令，不实施正常语法解析、功能设置和查询。
  “接受”说明仪器仅接收该条 SCPI 命令，不实施正常语法解析、功能设置和查询。

4 编程示例

- 基本操作示例.....69
- 高级操作示例.....73

4.1 基本操作示例

下面举例说明如何使用 VISA 库实现仪器程控编程的基本方法。以 C++语言为例。

- VISA库.....69
- 示例运行环境.....70
- 初始化和设置默认状态.....70
- 发送设置命令.....71
- 读取信号发生器状态.....72

4.1.1 VISA 库

VISA 是标准的 I/O 函数库及其相关规范的总称。其中，VISA 库函数是一套可方便调用的函数，其核心函数能够控制各种类型器件，无需考虑器件的接口类型和不同 I/O 接口软件的使用方法。这些库函数用于编写仪器的驱动程序，完成计算机与仪器间的命令和数据传输，以实现对仪器的程控。通过初始化寻址字符串（“VISA 资源字符串”），可建立具备程控端口（LAN、USB、GPIB 及 RS-232 等）的仪器的连接。

为实现远程控制首先需要安装 VISA 库。其中，VISA 库封装了底层的 VXI、GPIB、LAN 及 USB 接口的底层传输函数，方便用户直接调用。信号发生器支持的编程接口为：GPIB、LAN 和 RS-232。这些接口与 VISA 库和编程语言结合使用可以远程控制信号发生器。目前常使用 Agilent 公司为用户提供的 Agilent I/O Library 作为底层 I/O 库。

图 4.1 以 GPIB 接口为例显示了程控接口、VISA 库、编程语言和信号发生器之间的关系。

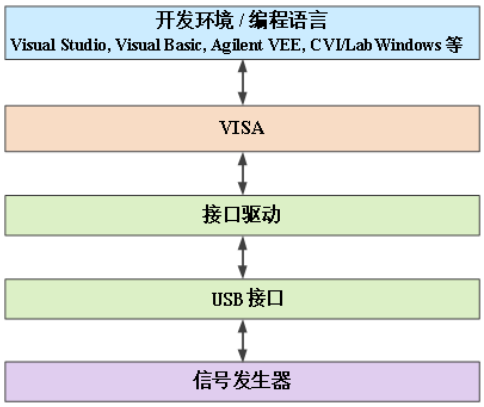

图4.1　程控软硬件层

4.1.2 示例运行环境

4.1.2.1 配置要求

本章所描述的编程示例，已在如下配置的计算机上运行成功。

- IBM 兼容、奔腾级以上的 PC 机;
- Windows 2000 或 Windows XP 操作系统;
- Visual Studio 2010/2012 集成开发环境;
- NI 公司的 PCI-GPIB 接口卡或 Agilent 公司的 GPIB 接口卡;
- NI 的 VISA 库或者 Agilent 的 VISA 库;
- 网卡;
- 可用的串口 COM1 和 COM2。

4.1.2.2 包含的文件

运行 C/C++编写的程序示例，您必须在 VC6.0 的项目中包含必须的文件
如果您使用 VISA 库必须进行以下步骤:

- 添加 visa32.lib 到源文件;
- 添加 visa.h 到头文件。

如果您使用 NI-488.2 库必须进行以下步骤:

- 添加 windows.h 文件到头文件;

关于 NI-488.2 库和 VISA 库的更详细资料请分别在 NI 公司和 Agilent 公司的网站查阅。

4.1.3 初始化和设置默认状态

程序开始时首先需要初始化 VISA 资源管理器,打开并建立 VISA 库与仪器的通信连接。具体步骤如下:

4.1.3.1 生成全局变量

首先生成其它程序模块需要调用的全局变量，例如：仪器句柄变量。以下示例程序需要包含下面的全局变量:

```
ViSession analyzer;
ViSession defaultRM;
const char analyzerString [VI_FIND_BUFLEN] = “TCPIP0::172.141.11.202::SOCKET “;
const analyzerTimeout = 10000;
```

其中，常量analyzerString代表仪器描述符,“TCPIP0”代表控者，IP地址是“172.141.11.202”。若仪器连接到USB，USB地址是“0x3399::0x4E02::00000000-0000-0000-0000-00005df2fb48”，那么该变量值是:

const char analyzerString [VI_FIND_BUFLEN] = “USB0::
0x3399::0x4E02::00000000-0000-0000-0000-00005df2fb48::RAW “;

4.1.3.2 初始化控者

/***
*****/
下面的示例说明了如何打开并建立VISA库与仪器（仪器描述符指定）的通信连接。
//初始化控者：打开默认资源管理器并且返回仪器句柄analyzer
/***
*****/

```
void InitController()
{
    ViStatus status;
    status = viOpenDefaultRM(&defaultRM);
    status = viOpen(defaultRM, analyzerString, VI_NULL, VI_NULL, &analyzer);
}
```

4.1.3.3 初始化仪器

/***
*****/
下面的示例初始化仪器默认状态，并且清空状态寄存器。
/***
*****/

```
void InitDevice()
{
    ViStatus status;
    long retCnt;
    status = viWrite(analyzer, “*CLS\n”, 5, &retCnt); //复位状态寄存器
    status = viWrite(analyzer, “*RST\n”, 5, &retCnt); //复位仪器
}
```

4.1.4 发送设置命令

/***
****/
下面的示例说明如何设置1433系列信号发生器的点频，幅度。
/***
****/

```
void SimpleSettings()
{
    ViStatus status;
    long retCnt;
    //设置信号源频率10MHz
    status = viWrite(analyzer, ":FREQ 10MHz\n ", 12, &retCnt);
    //设置信号源电平幅度-10dBm
     status = viWrite(analyzer, ":POW -10dBm\n ", 12, &retCnt);
}
```

4.1.5 读取测量仪器状态

```
/*****************************************************************************************
***/
```

下面的示例说明了如何读取仪器的设置状态。

```
/*****************************************************************************************
***/
void ReadSettings()
{
    ViStatus status;
    long retCnt;
    char rd_Buf_CW[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 20
    char rd_Buf_LVL[VI_READ_BUFLEN];

    //查询频率
    status = viWrite(analyzer, ":FREQ?\n", 7, &retCnt);
    Sleep(10);
    status = viRead(analyzer, rd_Buf_CW, 20, &retCnt);
    //查询幅度
    status = viWrite(analyzer, ":POW?\n", 6, &retCnt);
    Sleep(10);
    status = viRead(analyzer, rd_Buf_LVL, 20, &retCnt);
    //打印调试信息
    sprint("Cw is %s", rd_Buf_CW);
    sprint("LEVel is %s", rd_Buf_ LVL);
}
```

4.2 高级操作示例

- LAN接口设置点频并查询(Socket) 73
- LAN接口设置点频并查询(Visa) 74

- USB接口设置点频并查询.....75

4.2.1 LAN 接口设置点频并查询

```
/*****************************************************************************************
*****/
```

为了能正确使用以下示例，您必须将您的主机地址与信号源的 IP 地址相匹配。(本手册网络设计示例在 VC6.0 下利用 WINSOCK 组件建立 socket 实现)。

```
/*****************************************************************************************
*****/
#include “stdafx.h”
#include <afxsock.h>
#include <stdio.h>
#include <stdlib.h>
Csocket sockClient;
void main()
{
 bool flag;
 char buff[100];
 if (!AfxSocketInit())              //初始化网口
 {
        AfxMessageBox(“初始化失败”,”ok”，MB_OK);
 }
 else
 {
flag = sockClient.Create();
if(flag)
        {
                AfxMessageBox(“套接字创建成功”,”ok”，MB_OK);
        }
        else
        {
AfxMessageBox(“套接字创建失败”,”ok”，MB_OK);
sockClient.Close();
        }
 }
 flag=sockClient.Connect(name，5001);              //连接网络端口
 flag=sockClient.Send(“FREQ 1GHz\n”，12，0);    //设置点频为 1GHz
 if(!flag)
 {
MessageBox(“发送失败”,”ERROR”，MB_OK);
```

```
exit(0);
 }
 cout<<"显示点频频率值"<<end1
 flag=sockClient.Send("FREQ?\n", 6, 0);              //查询当前频率值
 if(!flag)
 {
MessageBox("发送失败","ERROR", MB_OK);
exit(0);
 }
 flag= sockClient.Receive(buff, 100, 0);              //将查询值放入数组
 if(!flag)
 {
MessageBox("Receive Failed!","ERROR", MB_OK);
Exit(0);
 }
 sockClient.Close();
}
```

4.2.2 LAN 接口设置点频并查询（Visa）

通过网络控制仪器时，建议使用 VISA 库作为底层 I/O 库。但是网络是串行通路，没有表示结束的硬连接。因此在使用 VISA 时，必须设置结束字节使能。

viSetAttribute(*instrSession, VI_ATTR_TERMCHAR_EN, VI_TRUE));

该设置可以使得网络通信收到'\n'时，自动结束。同时设置

viSetAttribute(*instrSession, VI_ATTR_SUPPRESS_END_EN, VI_FALSE));

/**
*****/

本例通过 LAN 接口，设置信号源输出 500MHz 信号的点频和-2dBm 的功率，并查询当前频率和功率，在 VS2010 开发环境中编译通过。本例使用 VISA 库对 1433 信号发生器进行远程控制，需要将 visa.h、visa32.lib、visatype.h 三个文件添加到工程中，把下述代码输入您的.cpp 文件。

/**
*****/

本例使用 VISA 库的功能，设置信号源输出 500MHz 信号的点频和-2dBm 的功率，并查询当前频率和功率，启动 VC6.0，添加必须的文件，把下述代码输入您的.cpp 文件

/**
*****/

```
#include "stdAfx.h"
#include <visa.h>
#include <iostream>
#include <stdlib.h>
```

```
#include <conio.h>
void main()
{
ViSession defaultRM，vi;              //声明类型为 VISession 的变量
ViStatus vistatus = 0;                //为了设备通信
Char buff[256];                       //声明存储字符数据的变量
int num;                              //声明存储整形数据的变量
vistatus = viOpenDefaultRM(&defalutRM);
 vistatus=viOpen(defaultRM，"TCPIP0::172.141.11.202::SOCKET", VI_NULL，VI_NULL，
&vi);
if(vistatus)
{
        printf("任务无法打开，请重新检查设备并连接\n");
        exit(0);
}
viPrintf(vi，"*RST\n");               //复位信号源
viPrintf(vi，"FREQ 500MHZ\n");        //设置点频 500MHz
viPrintf(vi，"FREQ?\n");                  //查询点频频率
viScanf(vi，"%t", buff);                  //把查询结果放入数组
printf("source CW freq is: %s\n", buff);     //显示点频频率
viPrintf(vi，"POW -2dBm\n");          //设置功率电平为-2dBm
viPrintf(vi，"POW?\n");               //查询当前功率
viScanf(vi，"%t", buff);                  //把查询结果放入数组
printf("source POWER is: %s\n", buff);     //显示功率
viClear(vi);
viClose(vi);
viClose(defaultRM);
}
```

4.2.3 USB 接口设置点频并查询

```
/*****************************************************************************************
****/
```

本例使用 VISA 库的功能，设置信号源输出 500MHz 信号的点频和-2dBm 的功率，并查询当前频率和功率，启动 VC6.0，添加必须的文件，把下述代码输入您的.cpp 文件

```
/*****************************************************************************************
*****/
#include "stdAfx.h"
#include <visa.h>
#include <iostream>
#include <stdlib.h>
```

4.2 高级操作示例

```
#include <conio.h>
void main()
{
ViSession defaultRM，vi;              //声明类型为 VISession 的变量
ViStatus vistatus = 0;                 //为了设备通信
Char buff[256];                        //声明存储字符数据的变量
int num;                               //声明存储整形数据的变量
vistatus = viOpenDefaultRM(&defalutRM);
      vistatus=viOpen(defaultRM，”
USB0::0x3399::0x4E02::00000000-0000-0000-0000-00005df2fb48::RAW”，VI_NULL，
VI_NULL，&vi);
if(vistatus)
{
          printf(“任务无法打开，请重新检查设备并连接\n”);
          exit(0);
}
viPrintf(vi，”*RST\n”);                //复位信号源
viPrintf(vi，”FREQ 500MHZ\n”);     //设置点频 500MHz
viPrintf(vi，”FREQ?\n”);               //查询点频频率
viScanf(vi，”%t”，buff);               //把查询结果放入数组
printf(“source CW freq is: %s\n”，buff);     //显示点频频率
viPrintf(vi，”POW -2dBm\n”);          //设置功率电平为-2dBm
viPrintf(vi，”POW?\n”);                //查询当前功率
viScanf(vi，”%t”，buff);               //把查询结果放入数组
printf(“source POWER is: %s\n”，buff);     //显示功率
viClear(vi);
viClose(vi);
viClose(defaultRM);
}
```

5 错误说明

本章将告诉您如何发现问题并接受售后服务。并说明信号发生器出错信息。

- 错误信息.....77
- 返修方法.....79

5.1 错误信息

信号发生器采用两种途径记录测量过程中出现的错误：前面板显示错误信息队列和SCPI（远程控制模式）错误信息队列，两种错误信息队列分别存储管理。

- 错误信息说明.....77

5.1.1 错误信息说明

- 本地错误信息.....77
- 程控错误信息.....78

5.1.1.1 本地错误信息

1) 错误信息查看

通过界面操作方法:

如果使用过程中在分析仪的右下脚显示有错误提示信息，则说明信号源分析仪软件运行或硬件出现问题。您根据错误代码可以大致判断问题类型，并采取相应措施排除故障。

在一个时刻，信号源分析仪错误显示区只能显示一条错误提示信息。由于仪器可能同时存在若干问题，执行下面的操作就可以看到所有错误提示信息:

步骤 1. 按[系统]，然后按［日志]，将会弹出日志列表。

步骤 2. 提示信息会显示在窗口中。

步骤 3. 用鼠标可以浏览错误信息，关闭对话窗口。

步骤 4. 选择清除错误列表按钮可以清除历史错误信息。

2) 错误信息说明

信号发生器测量过程中若检测到错误，状态指示区会显示告警或错误信息，并以状态条的不同颜色标注错误级别，如下图:



图5.1　状态指示区显示错误信息

表5.1　字体颜色区分状态指示区错误级别

颜色	错误级别	错误说明
红色	错误	测量过程中出现的错误，例如因数据丢失或错误的设置导致测量无法正常完成。
黄色	警告	测量过程中不规律出现的工作状况,例如：仪器设置与显示结果不匹配，或者与外部连接的仪器暂时断开连接，无法继续测量。
白色	信息	步骤设置或测量信息。

表5.2　本地错误信息说明列表

错误关键字段	详细错误说明
32769	频率补偿文件加载失败!
32770	功率补偿文件加载失败!
32771	调制补偿文件加载失败!
32773	保存状态文件失败!
32774	加载状态文件失败!
32775	保存默认状态失败!
32776	仪器存储空间已满，保存失败!

5.1.1.2 程控错误信息

1)　**错误信息格式及说明**

远程控制模式下，错误信息记录在日志中，格式如下:
“<错误代码>, <详细错误信息描述>”
举例:
“-108,”程控指令不合适的参数！”
程控错误信息包括两种类型:

- SCPI标准定义的负值错误代码，该类错误信息在此不做具体说明。

➢ 仪器特性正值错误代码，具体说明如下表：

表5.3 仪器特性错误信息说明列表

错误代码	错误说明
-100	程控指令错误!
-108	程控指令不合适的参数!
-109	程控指令丢失参数!
-115	程控指令参数个数不合适!
-130	程控指令后缀错误!
-131	程控指令无效的后缀!
-134	程控指令后缀太长!

2) 错误信息分类

错误事件只对应一种错误信息，下面分类说明错误信息类型（8257）：

- **查询错误 (–499 to –400)**：表明仪器的输出队列控制检测到 IEEE 488.2, Chapter 6 中描述的消息交换协议错误。此时，事件状态寄存器的查询错误位（bit2）置位（具体请参阅 IEEE 488.2, 6.5。）此时不能成功从输出队列中读取数据。
- **仪器特性错误 (–399 to –300, 201 to 703, and 800 to 810)**：表明仪器操作未成功，原因可能是不正常的硬件或固件状态，该类错误代码常用于仪器自检测操作。此时，事件状态寄存器的仪器特性错误位（bit3）置位。
- **执行错误（–299 to –200）**：表明仪器执行测量过程中检测到错误。此时，事件状态寄存器的执行错误位（bit4）置位。
- **命令错误（–199 to –100）**：表明仪器命令解析过程中检测到的语法错误，一般是错误的命令格式导致。此时，事件状态寄存器的命令错误位（bit5）置位。

5.2 返修方法

- 联系我们.....79
- 包装与邮寄.....80

5.2.1 联系我们

若1433系列信号发生器出现问题，首先观察错误信息并保存，分析可能的原因并参考用户手册中章节“7.2 故障诊断与排除”中提供的方法，予以先期排查解决问题。若未解决，请根据下面的联系方式与我所服务咨询中心联系并提供收集的错误信息，我们将以最快的速度协助您解决问题。

联系方式:

服务咨询: 0532-86889847 400-1684191
技术支持: 0532-86880796
传　　真: 0532-86889056
网　　址: www.ceyear.com
电子信箱: techbb@ceyear.com
邮　　编: 266555
地　　址: **中国山东省青岛市黄岛区香江路98号**

5.2.2 包装与邮寄

当您的信号发生器出现难以解决的问题时，可通过电话或传真与我们联系。如果经联系确认是信号发生器需要返修时，请您用原包装材料和包装箱包装信号发生器，并按下面的步骤进行包装:

1) 写一份有关信号发生器故障现象的详细说明，与信号发生器一同放入包装箱;
2) 用原包装材料将信号发生器包装好，以减少可能的损坏;
3) 在外包装纸箱四角摆放好衬垫，将仪器放入外包装箱;
4) 用胶带密封好包装箱口，并用尼龙带加固包装箱;
5) 在箱体上标明“易碎！ 勿碰！ 小心轻放！”字样;
6) 请按精密仪器进行托运;
7) 保留所有运输单据的副本。

注 意

包装信号发生器需注意

使用其它材料包装信号发生器，可能会损坏仪器。禁止使用聚苯乙烯小球作为包装材料，它们一方面不能充分保护仪器，另一方面会被产生的静电吸入仪器风扇中，对仪器造成损坏。

提 示

仪器的包装和运输

运输或者搬运本仪器（例如，因发货期间的损坏）时，请严格遵守用户手册章节“3.1.1.1 开箱”中描述的注意事项。

附 录

- 附录A SCPI命令按子系统分类速查表.....81
- 附录B 错误信息速查表.....84

附录 A SCPI 命令按子系统分类速查表

附表 1 SCPI 命令按子系统分类速查表

索引	命令	功能
1	*IDN?	查询功率计的标识串
2	*RST	复位信号发生器
3	[:SOURce]:OUTPut[:STATe]	设置和查询射频功能开关
4	[:SOURce]:OUTPut:MODulation[:STATe]	设置和查询调制功能开关
5	[:SOURce]:ROSCillator:SOURce	设置和查询信号源的振荡源
6	[:SOURce]:ROSCillator:EXTReference	设置和查询外参考频率
7	:SYStem:PANEl:LANguage	设置和查询系统语言
8	:SYStem:FAN:STATe	设置风扇控制方式
9	:SYSTem:COMMunicate:LAN:IP	设置和查询 IP 地址
10	:SYSTem:COMMunicate:LAN:SUBNET	设置和查询子网掩码
11	**:SYSTem:COMMunicate:LAN:GATEway**	设置和查询默认网关
12	**:SYStem:TEMPerature?**	查询系统温度
13	**:SYStem:DORMancy:STATe**	设置和查询自动休眠状态
14	**:SYStem:DORMancy:TIME**	设置和查询自动休眠时间
15	**:SYStem:SHUTdowm:STATe**	设置和查询自动关机状态
16	**:SYStem:SHUTdowm:TIME**	设置和查询自动关机时间
17	**:SYStem:DISPlay:MODE**	设置和查询显示模式
18	**:SYStem:BRIGht:STATe**	设置和查询背光控制方式
19	**:SYStem:BRIGht:DATA <Val>**	设置和查询背光强度
20	**:SYStem:PRESet:TYPE**	设置和查询复位方式
21	**:MEMory:STORe:STATus**	保存用户状态
22	:MEMory:LOAD: STATus	调用用户状态
23	[:SOURce]:FREQuency[:CW|FIXed]	设置和查询信号源频率
24	[:SOURce]:FREQuency:REFerence	设置和查询信号源频率参考
25	[:SOURce]:FREQuency:REFerence:STATe	设置和查询信号源频率参考开关
26	[:SOURce]:FREQuency:OFFSet	设置和查询频率偏置道
27	[:SOURce]:FREQuency:MULTiplier	设置和查询倍频系数
28	[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude	设置和查询射频输出信号的

附录 A SCPI 命令按子系统分类速查表

	]	功率电平
29	[:SOURce]:POWer:ALC[:STATe]	设置和查询环路状态开关
30	[:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet	设置和查询频率偏置
31	[:SOURce]:POWer:REFerence	设置和查询功率参考
32	[:SOURce]:POWer:REFerence:STATe	设置和查询信号源功率参考开关
33	[:SOURce]:POWer:ALC:BANDwidth|BWIDth[:DATA]	设置和查询 ALC 带宽
34	[:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO	设置和查询 ALC 带宽手自动状态
35	[:SOURce]:OUTPut:BLANking[:STATe]	设置和查询输出消隐状态
36	[:SOURce]:POWer:ALC:SEARch	激活功率搜索方式
37	[:SOURce]:POWer:ATTenuation:AUTO	设置和查询衰减器状态
38	[:SOURce]:POWer:ALC:Level	设置和查询 ALC 功率
39	[:SOURce]:POWer:ATTenuation	设置和查询功率衰减值
40	[:SOURce]:CORRection:STATe	设置和查询用户校准数据开关
41	[:SOURce]:CORRection:STARt:FREQuency	设置和查询用户校准起始填充频率
42	[:SOURce]:CORRection:STOP:FREQuency	设置和查询用户校准终止填充频率
43	[:SOURce]:CORRection:STARt:AMPLitude	设置和查询用户校准起始填充功率
44	[:SOURce]:CORRection:STOP:AMPLitude	设置和查询用户校准终止填充功率
45	[:SOURce]:CORRection:POINts	设置和查询用户校准点数
46	[:SOURce]:CORRection:FILL	填充用户校准数据
47	[:SOURce]:CORRection:LOAD	加载用户校准数据
48	[:SOURce]:CORRection:STORe	保存用户校准数据
49	[:SOURce]:PULM:STATe	设置和查询脉冲调制开关状态
50	[:SOURce]:PULM:SOURce	设置和查询脉冲输入类型
51	[:SOURce]:PULM:INTernal:PWIDth	设置和查询内部脉冲发生器的脉冲宽度
52	[:SOURce]:PULM:INTernal:PERiod	设置和查询内部脉冲发生器的脉冲周期
53	[:SOURce]:PULM:INTernal:DELay	设置和查询内部脉冲发生器的脉冲延时
54	[:SOURce]:PULM:POLarity	设置和查询脉冲输入反相状态
55	[:SOURce]:AM:STATe	设置和查询幅度调制开关状态

56	[:SOURce]:AM:INTernal:FREQuency	设置和查询幅度调制率
57	[:SOURce]:AM[:DEPTh]:LINear	设置和查询线性调制深度
58	[:SOURce]:AM[:DEPTh]:EXPonential	设置和查询指数调制深度
59	[:SOURce]:AM:TYPE	设置和查询调幅类型
60	[:SOURce]:AM:MODE	设置和查询深度调幅开关
61	[:SOURce]:FM:STATe	设置和查询频率调制开关状态
62	[:SOURce]:FM:INTernal:FREQuency	设置和查询频率调制率
63	[:SOURce]:FM[:DEViation]	设置和查询调频频偏
64	[:SOURce]:PM:STATe	设置和查询相位调制开关状态
65	[:SOURce]:PM:INTernal:FREQuency	设置和查询相位调制率
66	[:SOURce]:PM:DEViation	设置和查询调相相偏
67	[:SOURce]:SWEep:MODE	设置和查询扫描模式
68	[:SOURce]:LIST:TYPE	设置和查询扫描类型
69	:TRIGger[:SEQuence]:SOURce	设置和查询扫描触发类型
70	[:SOURce]:LIST:TRIGger:SOURce	设置和查询点触发类型
71	[:SOURce]:LIST:DIRection	设置和查询扫描方向
72	:INITiate:CONTinuous[:ALL]	设置和查询扫描控制类型
73	:INITiate[:IMMediate][:ALL]	执行单次扫描
74	[:SOURce]:LIST:MODE	设置和查询扫描手动模式状态
75	[:SOURce]:LIST:MANual	设置和查询手动扫描点数
76	[:SOURce]:FREQuency:STARt	设置和查询步进扫描起始频率
77	[:SOURce]:FREQuency:STOP	设置和查询步进扫描终止频率
78	[:SOURce]:POWer:STARt	设置和查询步进扫描起始功率
79	[:SOURce]:POWer:STOP	设置和查询步进扫描终止功率
80	[:SOURce]:SWEep:POINts	设置和查询步进扫描扫描点数
81	[:SOURce]:SWEep:DWELl	设置和查询步进扫描中每个点的驻留时间
82	[:SOURce]:LIST:FREQuency:DATA	设置和查询列表扫描起始频率
83	[:SOURce]:LIST:FREQuency: POINts	查询列表扫描频率点数
84	[:SOURce]:LIST:POWer:DATA	设置和查询列表扫描起始功率
85	[:SOURce]:LIST:POWer: POINts	查询列表扫描功率点数

86	[:SOURce]:LIST:DWELl:DATA	设置和查询列表扫描驻留时间
87	[:SOURce]:LIST:DWELl: POINts	查询列表扫描驻留时间点数
88	:MEMory:STORe:LIST	保存列表扫描文件
89	:MEMory:LOAD:LIST	加载列表扫描文件

附录 B 错误信息速查表

附表2 本地错误信息说明列表

错误关键字段	详细错误说明
32769	频率补偿文件加载失败!
32770	功率补偿文件加载失败!
32771	调制补偿文件加载失败!
32773	保存状态文件失败!
32774	加载状态文件失败!
32775	保存默认状态失败!
32776	仪器存储空间已满，保存失败!

附表3 程控错误信息表

错误代码	错误说明
-100	程控指令错误!
-108	程控指令不合适的参数!
-109	程控指令丢失参数!
-115	程控指令参数个数不合适!
-130	程控指令后缀错误!
-131	程控指令无效的后缀!
-134	程控指令后缀太长!