Ceyear 思仪

6939

矢量信号发生器

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矢量信号发生器，固件版本 Version 1.0 及以上。

- 6939 矢量信号发生器

版本: B.2 2022年12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 6939 矢量信号发生器！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

AV2.654.1021 SC

版本

B.2　2022.12

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单位授权，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。

检定周期

本产品检定周期为 1 年。

目 录

1 手册导航..... 1

1.1 关于手册.....1
1.2 关联文档.....1

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..... 13
2.1.5 状态报告系统..... 15
2.1.6 编程注意事项..... 28
2.2 仪器程控端口与配置.....29
2.2.1 LAN..... 29
2.3 I/O 库.....29
2.3.1 I/O 库概述..... 29
2.3.2 I/O 库安装与配置..... 30

3 程控命令..... 32

3.1 命令说明.....32
3.2 通用命令.....32
*CAL?.....33
*CLS.....33
*ESE <Value>.....33
*ESR?.....33
*IDN?.....33
*IST?.....33
*OPC.....33
*RST.....34
3.3 仪器子系统命令.....34
3.3.1 OUTPut 子系统.....34
3.3.2 FREQuency 子系统.....35
3.3.3 POWer 子系统.....35
3.3.4 Digtal Modulation 子系统.....36
3.3.5 AMPLitude Modulation 子系统.....39
3.3.6 FREQuency Modulation 子系统.....40
3.3.7 PHASe Modulation 子系统.....42

目录

3.3.7 PULM 子系统.....43
3.4 仪器兼容命令.....45
4 编程示例.....46
4.1 基本操作示例.....46
4.1.1 VISA 库.....46
4.1.2 示例运行环境.....47
4.1.3 初始化和设置默认状态.....47
4.1.4 发送设置命令.....48
4.1.5 查询状态命令.....48
4.2 高级操作示例.....49
4.2.1 LAN 接口设置点频并查询.....49
4.2.2 示例.....50
5 系统集成.....51
5.1 初始化函数.....51
5.2 配置函数.....52
5.2.1 通用命令.....53
5.2.2 频率、功率控制.....54
5.2.3 射频开关控制.....55
5.2.4 调制控制.....55
6 错误说明.....63
6.1 错误信息.....63
6.1.1 错误信息文件.....63
6.1.2 错误信息说明.....63
6.2 返修方法.....65
6.2.1 联系我们.....65
6.2.2 包装与邮寄.....66
附 录.....67
附录 A SCPI 命令按子系统分类速查表.....67
附录 B 错误信息速查表.....68

1 手册导航

本章介绍了 6939 矢量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.....1

1.1 关于手册

本手册介绍了 6939 矢量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 6939 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 6939 矢量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6939矢量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

6939 矢量信号发生器的产品文档包括：

- 快速使用指南
- 用户手册

1.2 关联文档

- 程控手册
- 在线帮助

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：

- 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
- 附录

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：

- 远程控制
- 程控命令
- 编程示例
- 错误说明
- 附录

在线帮助

在线帮助集成在仪器产品中，提供快速的文本导航帮助，方便用户本地和远控操作。仪器前面板硬键或用户界面工具条都有对应的快捷键激活该功能。包含的主要章节同用户手册。

2 远程控制

本章介绍了 6939 矢量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.....29
- I/O库.....30

2.1 远程控制基础

- 程控接口.....3
- 消息.....5
- SCPI命令.....5
- 命令序列与同步.....13
- 状态报告系统.....15
- 编程注意事项.....29

2.1.1 程控接口

具备远程控制功能的仪器一般支持接口为：LAN，具体型号仪器支持的端口类型由仪器本身功能决定。

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)	**VXI-11协议:** TCPIP::host_address[::LAN_device_name][::INSTR] **原始套接字协议:** TCPIP::host_address::port::SOCKET	控者通过仪器后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

- LAN接口.....3

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：

2.1 远程控制基础

- 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 192.168.12.0，则信号发生器的 IP 地址应设为 192.168.12.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[::LAN device name][::INSTR] 或

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；
- LAN device name 定义了协议和子设备的句柄号（该项可选）；
  - 0号设备选择VXI-11协议；
  - 0号高速LAN仪器选择较新的高速LAN仪器协议；
- INSTR 表示仪器资源类型（该项可选）；
- port 标识套接字端口号；
- SOCKET 表示原始网络套接字资源类。

举例：

- 仪器的IP地址是192.1.2.3，VXI-11协议的有效资源字符串是：
  TCPIP::192.1.2.3::INSTR
- 建立原始套接字连接时可使用：
  TCPIP::192.1.2.3::5025::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) VXI-11 协议

VXI-11 标准基于 ONC RPC(Open Network Computing Remote Procedure Call)协议，它是 TCP/IP 协议的网络/传输层。TCP/IP 网络协议和相关的网络服务被预先配置好，通信时，这种面向连接的通讯，即遵循按序交换并能识别连接的中断，保证了不丢失信息。

3) 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。

2.1.2 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“2.1.3SCPI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……6

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.

6939 矢量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”;
3) 本手册“附录 B 错误信息速查表”;

2.1.3.2 SCPI 命令说明

- 通用术语……6
- 命令类型……7
- 仪器专用命令语法……8
- 命令树……9

- 命令参数和响应.....9
- 命令中数值的进制.....12
- 命令行结构.....13

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

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 5.2 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。

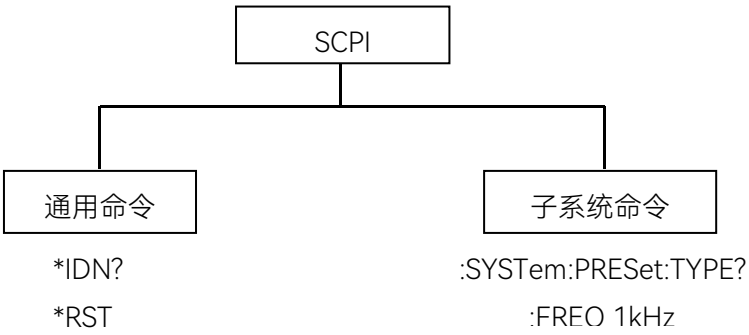

图 2.1 SCPI 命令类型

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]:EXPon ential? 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：

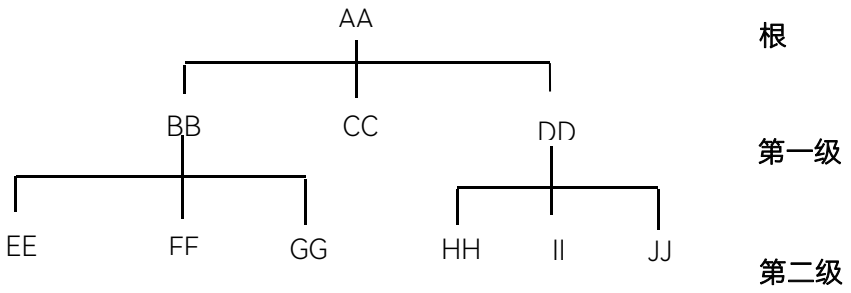

图 2.2 简化的命令树示意图

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：

2.1 远程控制基础

: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

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.....15
- SCPI状态寄存器结构.....16
- 状态寄存器说明.....18
- 状态报告系统的应用.....26
- 复位状态报告系统.....28

2.1.5.1 状态寄存器组织结构

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。

2.1 远程控制基础

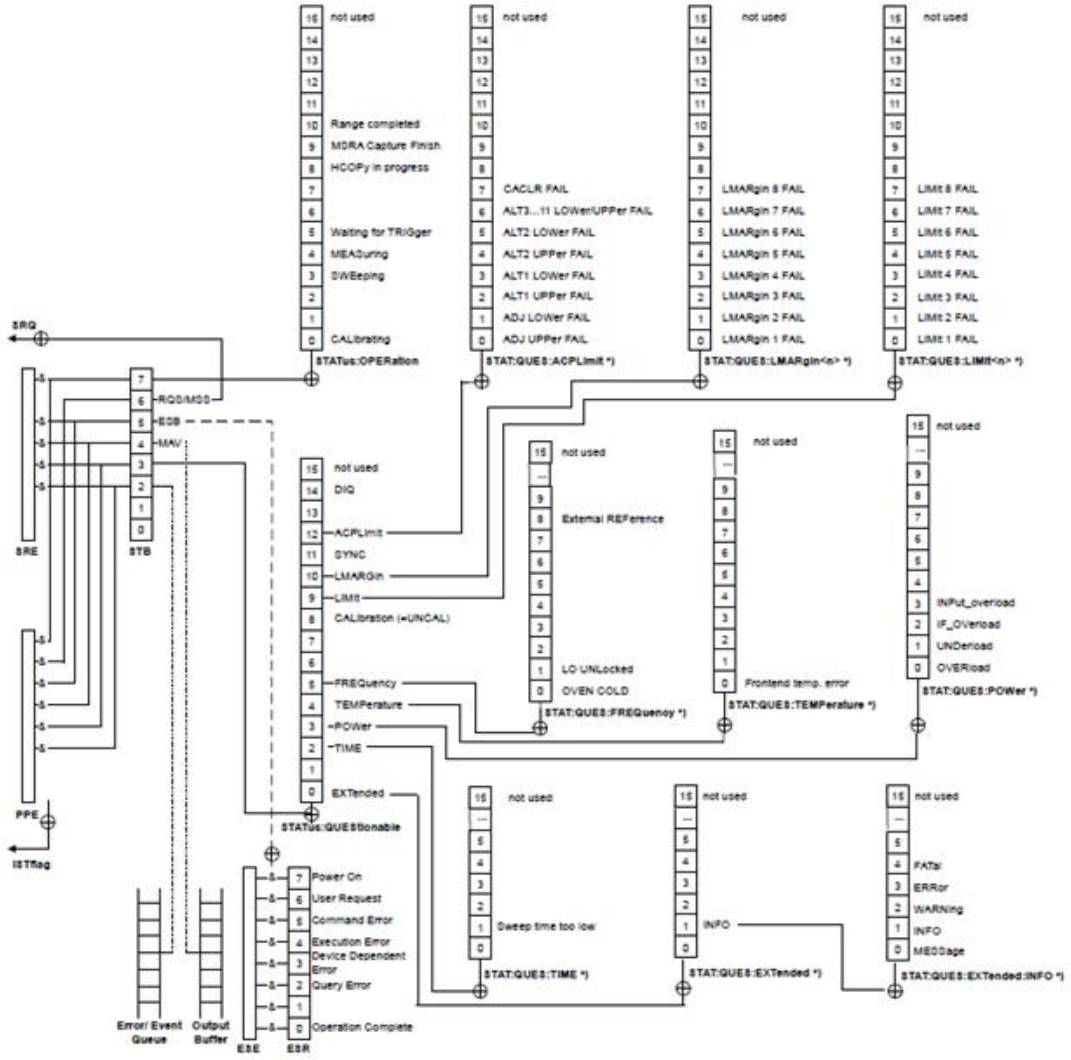

图 2.3 状态寄存器分层结构图

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。

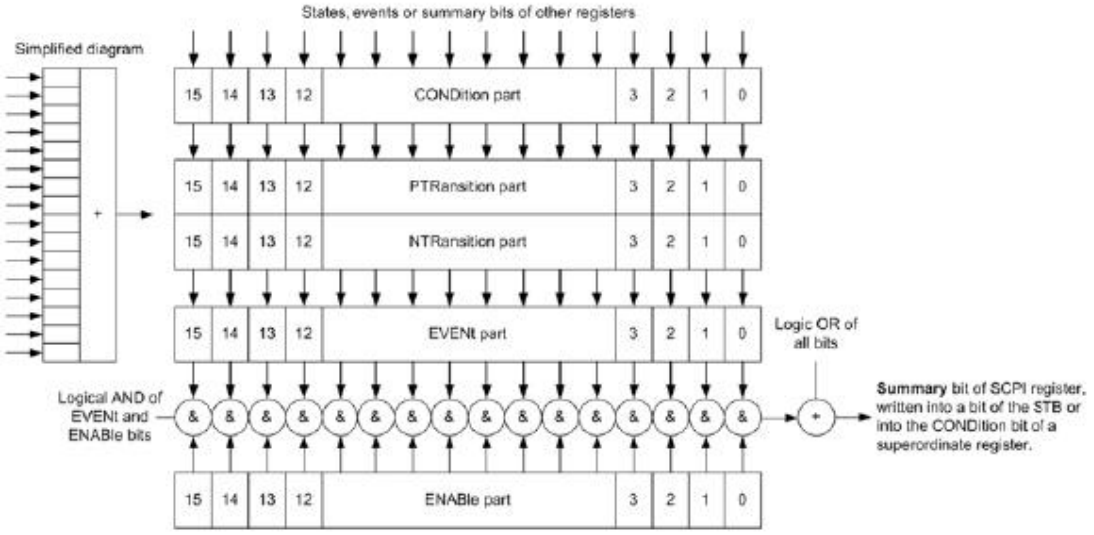

图 2.4 状态寄存器结构

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。

4) 状态:操作寄存器

状态：操作寄存器包含当前仪器的操作信息，及之前执行的操作信息。可通过命令“STATus:OPERation:CONDition?”或者“STATus: OPERation[: EVENt]?”读取该操作寄存器数值。寄存器说明如下表2.8。

表2.8 状态：操作寄存器说明

数据位	意 义
0	正在校准 仪器校准时，设置该位。
1-2	未使用。
3	正在扫描 仪器扫描时，设置该位。
4	正在测量 仪器测量时，设置该位。
5	等待触发 仪器准备触发并等待触发信号，设置该位。
6-7	未使用。
8	正在打印屏幕快照。 仪器正在打印屏幕快照时，设置该位。
9	MSRA模式下，用于数据采集：MSRA捕获完成。 MSRA操作时，若完成数据采集测量，设置该位。具体可参考用户手册中关于MSRA操作模式说明。
10	扫描范围完成 当扫描完成扫描列表的扫描范围时，并且激活“扫描完成时停扫”选项，设置该位。
11-14	未使用。
15	该位始终为 0。

5) 状态:询问寄存器

该寄存器包含不符合规范要求的仪器状态。可通过命令“STAT:QUES:COND”或“STAT:QUES:EVEN”，查询该寄存器数值。寄存器说明如下表2.9。

提 示

查询寄存器

状态：询问寄存器集合了所有低层子寄存器的信息（例如：其中的 bit2 集合了所有与时间相关的信息）。其中，因为每个通道对应单独的子寄存器，因此，若询问寄存器的一个状态位指示有错误，那么需要回溯通道的子寄存器中，检查具体的错误源头。默认状态下，查询的子寄存器状态属于当前选择的通道。

表2.9 状态：询问寄存器说明

数据位	意 义
0-1	未使用。
2	时间 当前通道出现时间错误时，设置该位。 参考 STATus:QUEStionable:TIMe Register 可查询具体错误类型。
3	功率电平 任意活动通道的测量功率电平出错时，设置该位。 参考STATus:QUEStionable:POWer Register可查询具体错误类型。
4	温度 若温度出错时，设置该位。
5	频率 若本地振荡器频率出错或者任意活动通道的参考频率出错时，设置该位。 参考 STATus:QUEStionable:FREQuency Register 可查询具体错误类型。
6-7	未使用。
8	校准 若仪器未校准（界面显示“未校准”提示信息），设置该位。
9	**门限（仪器定义）** 若任意窗口的活动通道超出门限值，设置该位。 参考STATus:QUEStionable:LIMit Register可查询具体错误类型。
10	LMARgin（仪器定义） 若任意窗口的活动通道超出了余量值，设置该位。 参考 STATus:QUEStionable:LMARgin Register 可查询具体错误类型。
11	SYNC（仪器定义） 若仪器未与使用的信号同步，设置该位。 FSW 不同步，如果： ➢ 测量或与测量时，fsw 不能同步 midamble。 ➢ 测量或与测量时，未找到突发脉冲信号。 ➢ 预测量时，结果偏离预期值太多。
12	ACPLimit (仪器定义) 若在任意活动通道中进行 ACLR 测量时，超出了限定值，设置该位。 参考 STATus:QUEStionable:ACPLimit Register 可查询具体错误类型。
13	未使用。
14	数字 I/Q（仪器定义） 若数字基带接口出现连接错误，设置该位。具体请参考用户手册中关于信号发生器的说明。
15	该位始终为 0。

6) 状态:询问:ACPLimit 寄存器

该寄存器存储了ACLR测量时极限检测结果。每个活动通道对应一个单独的ACPLimit寄

存器。可通过命令STATus: QUEStionable: ACPLimit: CONDition? 或STATus:QUEStionable:ACPLimit[: EVENt]? 读取寄存器值。

寄存器说明如下表2.10。

表2.10 状态:询问:ACPLimit寄存器说明

数据位	意 义
0	ADJ UPPer FAIL 上邻通道中超过极限值时设置该位。
1	ADJ LOWer FAIL 下邻通道中超过极限值时设置该位。
2	ALT1 UPPer FAIL 上邻第一可选通道中超过极限值时设置该位。
3	ALT1 LOWer FAIL 下邻第一可选通道中超过极限值时设置该位。
4	ALT2 UPPer FAIL 上邻第二可选通道中超过极限值时设置该位。
5	ALT2 LOWer FAIL 下邻第二可选通道中超过极限值时设置该位。
6	ALT3 … LOWer/UPPer FAIL 若3..11下邻或上邻可选通道中超过极限值时设置该位。
7	CACLR FAIL 若间隙通道中超过极限值时设置该位。
8..14	未使用。
15	该位始终为 0。

7) 状态:询问:扩展寄存器

该寄存器作为补充状态寄存器，存储了仪器其它状态寄存器未覆盖的状态信息。每个活动通道对应一个单独的扩展寄存器。可通过命令STATus:QUEStionable:EXTended:CONDtion? 或STATus:QUEStionable:EXTended[:EVENt]? 读取寄存器值。

寄存器说明如下表2.11。

表2.11 状态:询问:扩展寄存器说明

数据位	意 义
0	未使用。
1	INFO 若产生了一个状态信息，设置该位。具体信息类型可参考状态：询问：扩展：信息寄存器。
2..14	未使用。
15	该位始终为0。

8) 状态:询问:扩展:信息寄存器

该寄存器存储了操作仪器时产生的信息类型。每个活动通道对应一个单独的信息寄存器。可通过命令STATus:QUEStionable:EXTended:INFO:CONDition? 或 STATus:QUEStionable:EXTended:INFO[:EVENt]? 读取寄存器值。可通过命令 SYSTem:ERRor:EXTended? 查询特定通道产生的所有信息。

寄存器说明如下表2.12。

表2.12 状态:询问:扩展:信息寄存器说明

数据位	意 义
0	消息 操作中的事件或者状态可能会引起错误时，设置该位。
1	信息 若应用中产生了一个报告状态信息，设置该位。
2	警告 若仪器测量过程中产生了一个非法的状况，设置该位。例如：设置状态与显示结果不匹配或者与外部仪器的连接暂时中断。
3	错误 测量过程中出现错误时，设置该位。例如：数据丢失或者错误的设置导致无法正确完成测量。
4	严重错误 若应用中出现了严重错误，导致操作失败时，设置该位。
5..14	未使用
15	该位始终为 0。

9) 状态:询问:频率寄存器

该寄存器存储了本振和参考频率信息。每个活动通道对应一个单独的频率寄存器。可通过命令STATus:QUEStionable:FREQuency:CONDition? 或 STATus:QUEStionable: FREQuency [:EVENt]? 读取寄存器值。

寄存器说明如下表2.13。

表2.13 状态:询问:频率寄存器说明

数据位	意 义
0	OVEN COLD 若参考振荡器未到达其操作温度时，设置该位。同时用户界面显示提示信息：“OCXO”。
1	本振失锁 若本振失锁，设置该位。同时用户界面显示提示信息：“LO UNL”。
2..7	未使用。
8	外参考 若设置了外参考振荡器，但实际并未连接可用的外参考信号，设置该位。

	此时，频率合成器失锁，频率准确度低。
9..14	未使用。
15	**该位始终为0。**

10) 状态:询问:限定寄存器

该寄存器包含打开限定线时，限定测量结果信息。每个窗口每个活动通道对应一个单独的限定寄存器。可通过命令STATus:QUEStionable:LIMit<n>:CONDition? 或STATus:QUEStionable:LIMit<n>[:EVENt]? 读取寄存器值。

寄存器说明如下表2.14。

表2.14 状态:询问:限定寄存器说明

数据位	意 义
0	超出限定线 1 超出限定线 1 时，设置该位。
1	超出限定线 2 超出限定线 2 时，设置该位。
2	超出限定线 3 超出限定线 3 时，设置该位。
3	超出限定线 4 超出限定线 4 时，设置该位。
4	超出限定线 5 超出限定线 5 时，设置该位。
5	超出限定线 6 超出限定线 6 时，设置该位。
6	超出限定线 7 超出限定线 7 时，设置该位。
7	超出限定线 8 超出限定线 8 时，设置该位。
8..14	未使用。
15	该位始终为 0。

11) 状态:询问:限定范围寄存器

该寄存器包含打开限定范围时，限定测量结果信息。每个窗口每个活动通道对应一个单独的限定范围寄存器。可通过命令STATus:QUEStionable: LMARgin:CONDition? 或STATus:QUEStionable: LMARgin [:EVENt]? 读取寄存器值。

寄存器说明如下表2.15。

表2.15 状态:询问:限定范围寄存器说明

数据位	意 义
0	超出限定范围 1 超出限定范围 1 时，设置该位。
1	超出限定范围 2 超出限定范围 2 时，设置该位。
2	超出限定范围 3 超出限定范围 3 时，设置该位。
3	超出限定范围 4 超出限定范围 4 时，设置该位。
4	超出限定范围 5 超出限定范围 5 时，设置该位。
5	超出限定范围 6 超出限定范围 6 时，设置该位。
6	超出限定范围 7. 超出限定范围 7 时，设置该位。
7	超出限定范围 8. 超出限定范围 8 时，设置该位。
8..14	未使用。
15	**该位始终为 0。**

12) 状态:询问:功率寄存器

该寄存器包含操作仪器时，功率过载的信息。每个活动通道对应一个单独的功率寄存器。可通过命令STATus:QUEStionable: POWer:CONDition? 或 STATus:QUEStionable: POWer[:EVENt]? 读取寄存器值。

寄存器说明如下表2.16。

表2.16 状态:询问:功率寄存器说明

数据位	意 义
0	过载 若射频输入信号功率过载，设置该位。这会引起信号失真但不会损坏仪器，同时用户界面显示提示信息：“射频过载”。
1	低负载 若射频输入低负载，设置该位。同时用户界面显示提示信息:“低负载”。
2	中频过载 若中频通路过载，设置该位。同时用户界面显示提示信息：“中频过载”。
3	输入过载 若射频输入连接器的信号功率超出额定范围，设置该位。同时用户界面显示提示信息：“输入过载”。射频输入与输入混频器隔开以保护仪器。重复测量时，降低射频输入连接器的功率电平。
4..14	未使用。
15	**该位始终为 0。**

13) 状态:询问:温度寄存器

该寄存器包含操作仪器时，温度偏移信息。每个活动通道对应一个单独的温度寄存器。

2.1 远程控制基础

可通过命令STATus:QUEStionable:TEMPerature:CONDition? 或 STATus:QUEStionable:TEMPerature [:EVENt]? 读取寄存器值。

寄存器说明如下表2.17。

表2.17 状态:询问:温度寄存器说明

数据位	意 义
0	若前端温度传感器偏离自校准温度，设置该位。 开机预热时，该位始终为0。
1..14	未使用。
15	该位始终为0。

14) 状态:询问:时间寄存器

该寄存器包含操作仪器时，可能的时间错误信息。每个活动通道对应一个单独的时间寄存器。可通过命令STATus:QUEStionable: TIME:CONDition? 或 STATus:QUEStionable: TIME [:EVENt]? 读取寄存器值。寄存器说明如下表2.18。

表2.18 状态:询问:时间寄存器说明

数据位	意 义
0	未使用。
1	扫描时间太短 扫描时间太短校准失败，设置该位。 注：该错误不影响状态：询问寄存器有关校准的数据位。
2..14	未使用。
15	该位始终为0。

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：

数据位	7	6	5	4	3	2	1	0
权 重	128	64	32	16	8	4	2	1

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
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.....29

2.2.1 LAN

LAN（Local Area Network）程控系统采用SICL-LAN控制6939矢量信号发生器。

- 建立连接.....29
- 接口配置.....29

2.2.1.1 建立连接

使用网线将6939矢量信号发生器与外部控者（计算机）连接到局域网即可。

2.2.1.2 接口配置

通过局域网对矢量信号发生器进行远程控制时，应保证网络的物理连接畅通。

注 意

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

由于该矢量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。

2.3 I/O 库

- I/O库概述.....30
- I/O库安装与配置.....31

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 2.6 所示。

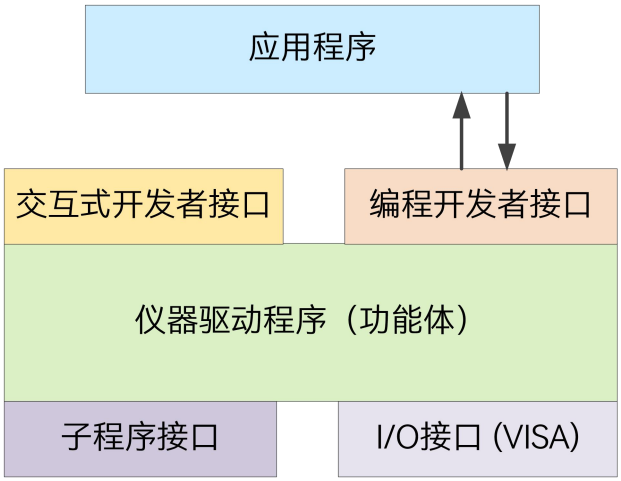

图 2.6 仪器驱动器结构模型

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 32
- 通用命令 32
- 仪器子系统命令 34

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 6939。
- 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。

远程控制时，本手册相关的命令后缀说明如下表：

表 3.1 命令后缀说明

后 缀	取值范围	描 述
<ch>	1..11	通道
<k>	1..11（门限线） 1 | 2 （显示线）	线
<m>	1..10	光标
<n>	1..10	窗口
<t>	1..5	轨迹

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。

- *CAL? 33
- *CLS 33
- *ESE<Value> 33
- *ESR? 33
- *IDN? 33
- *IST? 34

- *OPC.....34
- *RST.....34

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。

*CAL?

功能描述：校准查询。启动依次仪器校准并查询校准结果。

返回值：>0 表明校准错误。

说明：仅查询。

*CLS

功能描述：清除状态。设置状态字节（STB）、标准事件寄存器（ESR）和问题操作寄存的事件部分为零。该命令不改变屏蔽和转移寄存器的值，并清除输出缓冲区。

说明：仅设置。

*ESE <Value>

功能描述：事件状态使能。设置事件状态使能寄存器。查询该寄存器返回十进制数值。

参数说明：范围［0, 255］。

*ESR?

功能描述：读取事件状态寄存器十进制数值，接着设置寄存器数值为零。

返回值：范围［0，255］

说明：仅查询。

*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,6939,1312.8000K26/100005,1.30

说明：仅查询。

*IST?

功能描述：单独状态查询。十进制格式返回IST 标志。IST标志是并行查询时发送的状态位。

返回值：<ISTflag> ： 0 | 1。

说明：仅查询。

*OPC

功能描述：操作完成。所有待处理的命令执行完成后，设置事件状态寄存器的bit0，该位可用于启动服务请求。所有命令执行完毕后，该命令查询格式向输出缓冲区写入1，用于命令同步。

*RST

功能描述：复位命令。该命令将复位至用户指定模式。

3.3 仪器子系统命令

该章节详细说明6939矢量信号发生器的子系统命令。

- OUTPut.....34
- FREQuency.....35
- POWer.....36
- Digtal Modulation.....37
- AMPLitude Modulation.....40
- FREQuency Modulation.....41
- PHASe Modulation.....43
- PULM 44

3.3.1 OUTPut 子系统

输出子系统命令用于控制射频输出信号的开关。

下列命令选择操作模式，包括：

- OUTPut:[STATe] 34
- :OUTPut:MODulation[:STATe] 35

:OUTPut:[STATe]

功能描述：　该命令用于设置信号发生射频输出开关。

设置格式：　:OUTPut:[STATe] <val>

查询格式：　: OUTPut[:STATe]?

返回值为 0 到 1，分别对应于 OFF 和 ON。

参数说明：

<val>　布尔型数据，设置射频输出。取值如下：

ON | 1：　射频开。

OFF | 0：射频关。

举例：　:OUTP　1　　射频开。

:OUTP　?　　查询射频开关状态。

复位状态：　OFF

按键路径：　【射频】

:OUTPut:MODulation[:STATe]

功能描述：　该命令设置和查询输出信号调制开关。

设置格式：　:OUTPut:MODulation[:STATe] <val>

查询格式：　:OUTPut:MODulation[:STATe]?

参数说明：

< val >　布尔型数据，设置射频输出。取值如下：

ON | 1：　调制开。

OFF | 0：调制关。

举例：　:OUTP:MOD　1　　调制开。

:OUTP:MOD ? 查询调制状态。

复位状态: OFF '

按键路径: 【调制】

3.3.2 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] 35
- [:SOURce]:ROSCillator:SOURce 35

[:SOURce]:FREQuency[:CW|FiXed]

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: 250kHz~6GHz

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】

[:SOURce]:ROSCillator:SOURce

功能描述: 设置和查询内外参考。

设置格式: [:SOURce]:ROSCillator:SOURce <val>

查询格式: [:SOURce]:ROSCillator:SOURce?

参数说明:

<val> INT内参考

EXT外参考

设置举例: :ROSC:SOUR INT 设置内参考

查询举例: :ROSC:SOUR? 查询内外参考。

按键路径: 【频率】—>【参考】

3.3.3 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] 36
- [:SOURce]:POWer:ALC[:STATe] 36

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]

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>

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> 设置范围: -120.00dBm~10.00dBm

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】

[:SOURce]:POWer:ALC[:STATe]

功能描述: 设置和查询环路开关状态。

设置格式: [:SOURce]:POWer:ALC[:STATe] <State>

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> OFF | 0: 环路状态为闭环。

ON | 1: 环路状态为开环。

举例: :POW:ALC 1 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】 —> 【ALC开关】

3.3.4 Digtal Modulation 子系统

下列命令选择数字调制操作模式，包括:

- [:SOURce]:RADio:CUSTom:ALPHa 37
- [:SOURce]:RADio:CUSTom:DATA 37
- [:SOURce]:RADio:CUSTom:FILTer 37
- [:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] 38
- [:SOURce]:RADio:CUSTom:MODulation[:TYPE] 38
- [:SOURce]:RADio:CUSTom:SRATe 38
- [:SOURce]:RADio:ARB[:STATe] 39
- [:SOURce]:RADio:ARB:SEQuence 39
- [:SOURce]:RADio:ARB:SCLock:RATE 39
- [:SOURce]:RADio:MTONe:ARB[:STATe] 39

[:SOURce]:RADio:CUSTom:ALPHa

功能描述: 该命令设置基带滤波器因子。

设置格式: : [:SOURce]:RADio:CUSTom:ALPHa <val>.

查询格式: : [:SOURce]:RADio:CUSTom:ALPHa?

参数说明:

< val > 滤波器因子。

范围: 0——1.000

举例: :RADio:CUSTom:ALPHa 0.350 设置滤波器滤波因子为0.35。

复位状态: 0.220

按键路径: 【模式】—>【矢量调制】

[:SOURce]:RADio:CUSTom:DATA

功能描述: 该命令设置基带数据源类型。

设置格式: :RAD:CUST:DATA <Mode>

查询格式: :RAD:CUST:DATA ?

参数说明:

<Mode> 离散型数据，数据源类型。

PN9|PN11|PN15|PN16|PN20|PN21|PN23

举例: :RAD:CUST:DATA PN9 设置数据源类型为 PN9。

复位状态: PN9

按键路径: 【模式】—>【矢量调制】

[:SOURce]:RADio:CUSTom:FILTer

功能描述: 该命令设置基带滤波器类型。

设置格式: [:SOURce]:RADio:CUSTom:FILTer <Mode>

查询格式: [:SOURce]:RADio:CUSTom:FILTer?

参数说明:

< Mode > 基带预调制滤波器类型，取值如下:

RNYQuist 根耐奎斯特滤波器,

NYQuist 耐奎斯特滤波器,

GAUSsian 高斯滤波器,

RECTangle 矩形滤波器。

举例: :RADio:CUSTom:FILTer RNYQuist 设置滤波器类型为根乃奎斯特滤波器。

复位状态: RNYQuist

按键路径: 【模式】—>【矢量调制】

[:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation]

功能描述: 该命令设置和查询基带调制类型为FSK模式时，设置FSK的调频频偏。

设置格式: [:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] <val>

查询格式: [:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] ?

参数说明:

< val > 调制类型为FSK模式时，FSK的调频频偏。

范围：1MHz[1Hz，10MHz]。

举例: :RADio:CUSTom:MODulation:FSK 1MHz 设置滤波器频率频偏为 1MHz。

复位状态: 100kHz

按键路径: 【模式】—>【矢量调制】

[:SOURce]:RADio:CUSTom:MODulation[:TYPE]

功能描述: 该命令设置基带调制制式。

设置格式: [:SOURce]:RADio:CUSTom:MODulation<Mode>

查询格式: [:SOURce]:RADio:CUSTom:MODulation[:TYPE]?

参数说明:

<Mode> 离散型数据。

设置命令格式：
BPSK|QPSK|MSK|2FSK|4FSK|16QAM|32QAM|64QAM|128QAM|256QAM|512QAM|2ASK|4ASK

举例： :RADio:CUSTom:MODulation 2FSK 设置滤波器调制制式为2FSK。

复位状态： QPSK

按键路径： 【模式】—>【矢量调制】

[:SOURce]:RADio:CUSTom:SRATe

功能描述： 该命令设置基带码元速率。

设置格式： [:SOURce]:RADio:CUSTom:SRATe <val>

查询格式： [:SOURce]:RADio:CUSTom:SRATe?

参数说明：

<val> 双精度型数据，码元速率。
范围：1MHz[0,100MHz]

举例： :RAD:CUST:MOD:SRAT 1MHz 设置滤波器码元速率为 1MHz。

复位状态： 1MHz

按键路径： 【模式】—>【矢量调制】

[:SOURce]:RADio:ARB[:STATe]

功能描述： 该命令设置任意波开关状态。

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

查询格式： [:SOURce]:RADio:ARB:STATe?

参数说明：

<State> 布尔型数据，取值如下：
ON | 1： 任意波开，
OFF | 0：任意波关。

举例： [:SOURce]:RADio:ARB:STATe 1 开启任意波。

复位状态： 0

按键路径： 【模式】—>【矢量调制】—> 【任意波设置】

[:SOURce]:RADio:ARB:SEQuence

功能描述： 该命令设置加载波形文件。

设置格式： [:SOURce]:RADio:ARB: :SEQuence <file_name>

参数说明：

<FileName> 字符串类型。存放波形段文件的绝对路径字符串。

举例： :RAD:MTON:ARB:SEQ “C:\CeYear\VST\TwoTone.txt” 打开任意波文件。

按键路径： 【模式】—>【矢量调制】—> 【任意波设置】

说明： 仅设置

[:SOURce]:RADio:ARB:SCLock:RATE

功能描述： 该命令设置信号发生器任意波模式下的采样时钟频率。

设置格式： [:SOURce]:RADio:ARB:SCLock:RATE <val>

查询格式： [:SOURce]:RADio:ARB:SCLock:RATE?

参数说明：

<val>　　范围：250MHz[200kHz，250MHz]，1GHz。
举例:　　[:SOURce]:RADio:ARB:SCLock:RATE 50MHz 任意波时钟频率为 50MHz。
按键路径:　　【模式】—>【矢量调制】—> 【任意波设置】
说明:　　采样时钟在200kHz~250MHz范围内任意设置，250MHz以上设置任意值输出采样时钟均为1GHz。

[:SOURce]:RADio:MTONe:ARB[:STATe]
功能描述:　　该命令设置信号发生器多音开关状态
设置格式:　　[:SOURce]:RADio:MTONe:ARB:STATe ON|OFF|1|0
查询格式:　　[:SOURce]:RADio:MTONe:ARB:STATe?
参数说明:
<State>　　布尔型数据。多音调制开关状态，取值如下:
ON | 1:　多音调制开，
OFF | 0:　多音调制关。
举例:　　[:SOURce]:RADio:MTONe:ARB:STATe 1 开启多音调制。
复位状态:　0
按键路径:　　【模式】—>【矢量调制】—> 【多音设置】

3.3.5 AMPLitude Modulation 子系统

用于控制射频输出信号的调幅操作模式，包括:

- [:SOURce]:AM:STATe 40
- [:SOURce]:AM:INTernal:FREQuency 40
- [:SOURce]:AM[:DEPTh]:LINear 41
- [:SOURce]:AM:INTernal:SHAPe 41

[:SOURce]:AM:STATe
功能描述:　　该命令设置信号发生器调幅信号输出状态。
设置格式:　　[:SOURce]:AM:STATe <State>
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:　0
按键路径:　　【模式】—>【模拟调制】—>【幅度调制】—>【调制开关】

[:SOURce]:AM:INTernal:FREQuency
功能描述:　　该命令设置信号发生器调幅内部调制率。
设置格式:　　[:SOURce]:AM:INTernal:FREQuency < AMFreq >
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：1Hz~100kHz

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: 【模式】—>【模拟调制】—>【幅度调制】—>【调制深度】]

[:SOURce]:AM[:DEPTh]:LINear

功能描述: 该命令设置以百分比为单位的调幅信号深度。

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

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:

<AmDepthLine > 调幅深度（线性）。
范围：30[5，100]。

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: 50%

按键路径: 【模式】—>【模拟调制】—>【幅度调制】—> 【线性调制深度】

[:SOURce]:AM:INTernal:SHAPe

功能描述: 该命令设置调幅信号输出波形，包括：正弦、方波、三角波、锯齿波4种波形。

设置格式: [:SOURce]:AM:INTernal:SHAPe <Mode>

查询格式: [:SOURce]:AM:INTernal:SHAPe?

参数说明:

<Mode> 离散型数据。调幅信号输出波形类型，取值如下:
SINE 正弦，
SQUare 方波，
TRIangle 三角波，
RAMP 锯齿波，

举例: : AM:INTernal:SHAP RAMP 调幅信号波形为锯齿波。

复位状态: SINE

按键路径: 【模式】—>【模拟调制】—>【幅度调制】—> 【调制波形】

3.3.6 FREQuency Modulation 子系统

用于控制射频输出信号的调频操作模式，包括:

- [:SOURce]:FM:STATe 41
- [:SOURce]:FM:INTernal:FREQuency 42
- [:SOURce]:FM[:DEViation] 42
- [:SOURce]:FM:INTernal:SHAPe 42

[:SOURce]:FM: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 <State>

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: 【模式】—>【模拟调制】—>【频率调制】—>【调制开关】

[:SOURce]:FM:INTernal:FREQuency

功能描述: 设置和查询频率调制率。

设置格式: [:SOURce]:FM:INTernal:FREQuency <FMFreq>

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> 调频调制率,

设置范围: 1Hz~1MHz

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: 100kHz

按键路径: 【模式】—>【模拟调制】—>【频率调制】—> 【调制率】

[:SOURce]:FM[:DEViation]

功能描述: 设置和查询调频频偏。

设置格式: [:SOURce]:FM[:DEViation] <FreqOffset>

查询格式: [:SOURce]:FM[:DEViation]?

参数说明:

<FreqOffset> 调频频偏

设置范围: 0Hz~10MHz

举例: :FM 1kHz 设置信号源调频频偏为1kHz。

:FM? 查询信号源调频频偏值。

复位状态: 10kHz

按键路径: 【模式】—>【模拟调制】—>【频率调制】—> 【调频频偏】

[:SOURce]:FM:INTernal:SHAPe

功能描述: 该命令设置调频信号输出波形，包括: 正弦、方波、三角波、锯齿波4种波形。

设置格式: [:SOURce]:FM:INTernal:SHAPe <Mode>

查询格式: [:SOURce]:FM:INTernal:SHAPe?

参数说明:

<Mode>　离散型数据。调频信号输出波形类型，取值如下:
SINE 正弦,
SQUare 方波,
TRIangle 三角波,
RAMP 锯齿波,

举例:　[:SOURce]:FM:INTernal:SHAP RAMP 调频信号波形为锯齿波。

复位状态:　SINE

按键路径:　【模式】—>【模拟调制】—>【频率调制】—>【调频波形】

3.3.7 PHASe Modulation 子系统

用于控制射频输出信号的调相操作模式，包括:

- [:SOURce]:PM:STATe 43
- [:SOURce]:PM:INTernal:FREQuency 43
- [:SOURce]:PM[:DEViation] 44
- [:SOURce]:PM:INTernal:SHAPe 44

[:SOURce]:PM:STATe

功能描述:　设置和查询调相开关状态。

设置格式:　[:SOURce]:PM:STATe <State>

查询格式:　[:SOURce]:PM: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>　调相开关,取值如下:
OFF | 0: 频率调制开关关。
ON | 1: 频率调制开关开。

举例:　:PM:STAT 1　设置频率调制开关为开。
:PM:STAT?　查询频率调制开关状态。

复位状态:　关

按键路径:　【模式】—>【模拟调制】—>【相位调制】—>【调制开关】

[:SOURce]:PM:INTernal:FREQuency

功能描述:　设置和查询频率调制率。

设置格式:　[:SOURce]:PM:INTernal:FREQuency <PMFreq>

查询格式:　[:SOURce]:PM:INTernal:FREQuency?
返回信号源当前频率调制率，单位为 Hz。

参数说明:

<PMFreq>　调相调制率。
设置范围: 1Hz~1MHz

举例:　:PM:INT:FREQ 10kHz　设置信号源频率调制率为10kHz。
:PM:INT:FREQ?　查询信号源频率调制率。

复位状态:　10kHz

按键路径:　【模式】—>【模拟调制】—> 【相位调制】 —> 【调制率】

[:SOURce]:PM[:DEViation]

功能描述: 设置和查询调相相偏。

设置格式: [:SOURce]:PM[:DEViation] <PhaseOffset>

查询格式: [:SOURce]:PM[:DEViation]?

参数说明:

<PhaseOffset> 调相相偏。

设置范围: 0Hz~3.14rad

举例: :PM:DEViation 3rad 调相信号调相相偏为3rad。

:PM? 查询信号源调频频偏值。

复位状态: 1rad

按键路径: 【模式】—>【模拟调制】—> 【相位调制】 —>【相偏】

[:SOURce]:PM:INTernal:SHAPe

功能描述: 该命令设置调相信号输出波形，包括: 正弦、方波、三角波、锯齿波4种波形。

设置格式: [:SOURce]:PM:INTernal:SHAPe SINE|SQUare|TRIangle|RAMP

查询格式: [:SOURce]:PM:INTernal:SHAPe?

参数说明:

[:Mode] 离散型数据。调相信号输出波形，取值如下:

SINE 正弦，

SQUare 方波，

TRIangle 三角波，

RAMP 锯齿波，

举例: [:SOURce]:PM:INTernal:SHAP RAMP 调相信号输出波形为锯齿波。

复位状态: SINE

按键路径: 【模式】—>【模拟调制】—> 【相位调制】—>【调相波形】

3.3.7 PULM 子系统

用于控制射频输出信号的脉冲调制模式，包括:

- [:SOURce]:PULM:STATe 44
- [:SOURce]:PULM:SOURce 45
- [:SOURce]:PULM:INTernal:PWIDth 45
- [:SOURce]:PULM:INTernal:PERiod 45

[:SOURce]:PULM: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 <State>

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> 脉冲调制开关,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: 【模式】—>【模拟调制】—> 【脉冲调制】—> 【调制开关】

[:SOURce]:PULM:SOURce

功能描述: 设置和查询脉冲输入类型。

设置格式: [:SOURce]:PULM:SOURce <Mod>

查询格式: [:SOURce]:PULM:SOURce?

返回值为0到4，分别对应于脉冲输入设置为内部自动(0)、内部触发(1)、内部门控（2)、双脉冲（3）和外部输入（4)。

参数说明:

<Mod>　数据类型为离散型。取值如下:

INTernal: 脉冲源为内部自动

EXTernal: 脉冲输入设置为外部。

举例: :PULM:SOUR INTernal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: INTernal

按键路径: 【模式】—>【模拟调制】—> 【脉冲调制】—> 【脉冲输入】

[:SOURce]:PULM:INTernal: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>

查询格式: [:SOURce]:PULM:INTernal:PWIDth?

参数说明:

<Pwidth>　10ns的整数倍，小于脉冲周期。

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: 【模式】—>【模拟调制】—> 【脉冲调制】—> 【脉冲宽度】

[:SOURce]:PULM:INTernal: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>

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>　内部脉冲发生器的脉冲周期。

设置范围：50ns~100ms

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: 【模式】—>【模拟调制】—> 【脉冲调制】—>【脉冲周期】

3.4 仪器兼容命令

无。

4 编程示例

- 基本操作示例.....47
- 高级操作示例.....50

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.....47
- 示例运行环境.....48
- 初始化和设置默认状态.....48
- 发送设置命令.....49
- 查询状态命令.....49

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”），可建立程控 LAN 端口的仪器的连接。

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

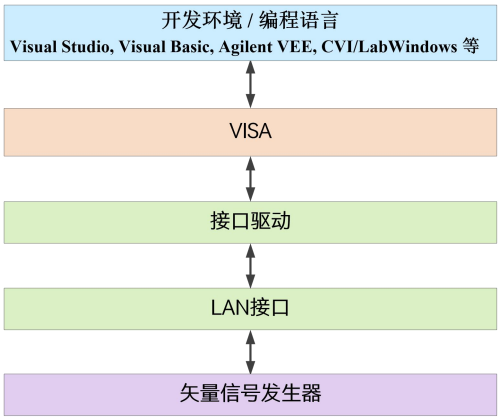

图4.1 程控软硬件层

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。矢量信号发生器目前支持的编程接口为 LAN 接口。该接口与 VISA 库和编程语言结合使用可以远程控制矢量信号发生器。目前常使用 NI 公司为用户提供的 VISA 库作为底层 I/O 库。

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 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 Visual Studio 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 visa64.lib 到源文件;
- 添加 visa.h 到头文件。

关于 VISA 库的更详细资料请分别在 NI 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 vsaHandle;

const char vsaHandleString[VI_FIND_BUFLEN] = "TCPIP::192.168.1.1::5025::SOCKET";

其中，常量vsaHandleString代表仪器描述符，仪器连接到LAN，IP地址是“172.141.10.10”，端口号是5025。

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vsaHandle
/*****************************************************************************/
void InitController()
{
    ViStatus status;
    status = viOpenDefaultRM(&defaultRM);
    status = viOpen(defaultRM, vsaHandleString, VI_NULL, VI_NULL, & vsaHandle);
```

```
}
```

4.1.4 发送设置命令

```
/*****************************************************************************/
下面的示例说明如何设置6939矢量信号发生器的频率参数。
/*****************************************************************************/
void ConfigureCenterFreq ()
{
    ViStatus status = 0;
    Unsigned int retCnt = 0;
    char buffer[256];
    //设置中心频率1GHz
    sprint(buffer, ":FREQ:CENT 1e9");
    status = viWrite(vsaHandle, reinterpret_case<unsigned char*>(&buffer),
    strlen(buffer),
                    &retCnt);
    //设置扫宽50MHz
    sprint(buffer, ":FREQ:SPAN 50e6");
    status = viWrite(vsaHandle, reinterpret_case<unsigned char*>(&buffer),
    strlen(buffer),
                    &retCnt);
    //设置分辨率带宽3MHz
    sprint(buffer, ":BAND 3e6");
    status = viWrite(vsaHandle, reinterpret_case<unsigned char*>(&buffer),
    strlen(buffer),
                    &retCnt);
}
```

4.1.5 查询状态命令

```
/*****************************************************************************/
下面的示例说明了如何读取仪器的设置状态。
/*****************************************************************************/
void ReadSettings()
{
    ViStatus status;
    Unsigned int retCnt;
    char rd_Buf_CW[VI_READ_BUFLEN]; // #define VI_READ_BUFLEN 20
    char rd_Buf_LVL[VI_READ_BUFLEN];
    //查询中心频率
    status = viWrite(vsaHandle, "FREQ:CENT?", reinterpret_case<unsigned
                    char*>(&buffer), &retCnt);
```

```
        Sleep(10);
        status = viRead(vsaHandle, rd_Buf_CW, 20, &retCnt);
        //查询参考电平
        status = viWrite(vsaHandle, ":DISP:WIND[1]:TRAC:Y:RLEV?",
        reinterpret_case<unsigned char*>(&buffer),, &retCnt);
        Sleep(10);
        status = viRead(vsaHandle, 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.....50
- 示例.....51

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 Visual Studio2015 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 示例

无。

5 系统集成

为了方便用户使用，我们对 SCPI 命令进行了封装，做成动态链接库，用户通过调用动态链接库可以很方便的对 6939 进行设置或查询，适合用户搭建自动测试系统。

- 初始化函数.....52
- 配置函数.....53

5.1 初始化函数

查找设备

int VSG_EiViFindDevs(ScyPXIDevInfo* vstdevInfo, int *pvstDevNum, ScyPXIDevInfo* UPdevInfo, int *pUPDevNum,int nMaxDevNum)

函数用途:

查找设备，获取设备数量以及设备信息。

参数列表:

vstdevInfo

变量类型：ScyPXIDevInfo

存储查找到的设备信息结构体，结构体内容定义如下:

```
struct ScyPXIDevInfo
{
    unsigned char ucBusNum;//总线号
    unsigned char ucSlotNum;//槽号
    unsigned char ucFuncNum;//功能号
    unsigned short usVendorId;//厂商号
    unsigned short usDeviceId;//设备号
    char pucCompanyname[256];//制造商
    char pucDRVDESC[256];//驱动描述
    char pucSrcName[256];//资源
    unsigned int   uiSerialID;//串号
}
```

pvstDevNum

变量类型：int

返回查找到的设备个数

nMaxDevNum

变量类型：int

搜索的最大设备个数，默认值取 32。

UPdevInfo

暂不使用，为空即可。

pUPDevNum

暂不使用，为空即可。

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

打开仪器

int VSG_InitWithOptions (char * resourceName,unsigned short IDQuery, unsigned short resetDevice, char * optionString, unsigned int *newVi)

函数用途:

打开仪器

该函数是控制仪器之前首先要调用的函数，它完成下面的初始化操作:

根据参数 resourceName 指定的接口和逻辑地址信息，打开模块的句柄，与 6939 连接建立数据通道；返回 vi 用于在后来的仪器驱动函数的调用中识别该模块。

参数列表:

resourceName

变量类型：char*

需要初始化的设备资源名字符串，对应查找函数中 vstdevInfo 结构体内 pucSrcName 成员。

例如：PCIe：PXI2::3::4::INSTR

2：总线号；3：设备号；4 槽号。若改变了仪器的插槽，则应以实际号为准。

extresourceName

扩展资源名，为空即可。

IDQuery

变量类型：unsigned short

ID 查询参数，若设置为 true，则该函数查询仪器 ID 并检查是否符合本驱动。

resetDevice

变量类型：unsigned short

该参数设为 true，则该函数复位仪器，作用与发送*RST 命令相同。

newVi

变量类型：unsigned int

函数返回的仪器句柄，与仪器通信。

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

5.2 配置函数

该类函数是用来设置 6939 矢量信号发生器，通过调用其中的函数设置、操作信号发生器。共包含 8 个子类：通用命令控制、频率控制、功率控制、扫描控制、调制控制、系统控制、其他命令控制，其中的每个子类包含各自相关的函数。下面依次说明每个子类包含的函数。

- 通用命令 54
- 频率功率控制 55
- 射频开关控制 56
- 调制控制 56

5.2.1 通用命令

该子类与通用命令相关，下面分别说明与通用命令收发相关函数：

复位仪器

int VSG_reset(unsigned int vi)
函数用途：

复位仪器，配置设备为初始上电状态。

参数列表：
vi
变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

返回值：
返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

关闭仪器

int VSG_close (unsigned int vi)
函数用途：

关闭仪器，在控制仪器结束后需调用该函数关闭仪器。

参数列表：
vi
变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。
返回值：
返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

参数配置生效

int VSG_Commit(unsigned int vi)
函数用途：

除通用命令外的其他参数配置程序之后，使用本函数配置设备，实现设置功能。

例如，设置频率功率为 1GHz，0dBm：
VSG_ConfigureRF(vi,1e9,0);
VSG_Commit(vi);
参数列表：
vi
变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

5.2.2 频率、功率控制

该子类函数与频率相关设置有关，主要包含点频设置、频率参考设置、ALC 环路开关设置等相关函数，下面说明这些函数:

设置信号源频率和功率

int VSG_ConfigureRF(unsigned int vi, double frequencyHz,double powerLeveldBm)

函数用途:

设置信号源输出频率和功率。

参数列表:

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

frequencyHz

变量类型：double

所要设置的频率值，单位：Hz，范围 250*1e3~6*1e9

powerLeveldBm

变量类型：double

所要设置的功率值，单位：dB，范围-120~10

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

设置环路开关状态

int VSG_ConfigurePowerAlcState(unsigned int vi, int AlcState)

函数用途:

设置信号源环路开关状态。

参数列表:

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

AlcState

变量类型：int

环路开关：0——开环，1——闭环。

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

设置信号源频率参考开关

int VSG_ConfigureReferenceOscillator(unsigned int vi, ViInt32 source)

函数用途:

设置信号源基带模块频率参考通路。

参数列表:

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

Source

变量类型：int

信号源频率参考：0——内参考，1——外参考。

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

5.2.3 射频开关控制

设置射频开关

int VSG_ConfigureOutputEnabled(unsigned int vi, unsigned short outputEnabled)

函数用途:

设置射频输出信号的开关。

参数列表:

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

outputEnabled

变量类型：int

射频开关：0——射频关，1——射频开

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

5.2.4 调制控制

该子类与调制相关，主要包含任意波、多音、数字调制、脉冲调制、幅度调制、频率调制相关函数。下面说明这些函数:

设置调制信号类型

int VSG_ConfigureGenerationMode(unsigned int vi,unsigned short nMode)

函数用途:

设置信号源调制信号类型。

参数列表:

5.2 配置函数

vi
变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。
nMode
变量类型：unsigned short
信号调制类型：
0——无调制
1——任意波
2——多音
3——矢量调制
4——幅度调制
5——频率调制
6——相位调制
7——脉冲调制
返回值：
返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

设置任意波文件

int VSG_SelectArbWaveform(unsigned int vi, char * name)
函数用途：
设置需要加载的任意波文件。
参数列表：
vi
变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。
name
变量类型：char*
存储任意波文件绝对路径
返回值：
返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

设置任意波采样频率

int VSG_SetSampClk(unsigned int vi, double nClkHz)
函数用途：
设置任意波采样频率。
参数列表：
vi
变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。
nClkHz

变量类型:double

任意波采样频率，单位：Hz，范围：200*1e3-250*1e6,1e9

200*1e3~250*1e6Hz 范围内可任意设置，采样率大于 250*1e6Hz 设备任意波采样率为 1e9Hz

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

播放任意波

int VSG_PlayArbWaveform(unsigned int vi, char * name, int nRepeatCount)

函数用途:

播放任意波。

参数列表:

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

name

变量类型：char*

存储任意波文件绝对路径

nRepeatCount

变量类型：int

暂不可用，设为 0

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

多音信号设置

int VSG_SetMultiTone(unsigned int vi, int nWaveNum, double* dFrqHz, double* dPwrDBm, double* dPhase)

函数用途:

设置多音信号。

参数列表:

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

nWaveNum

变量类型：int

多音信号音调数量，范围：1-1024

dFrqHz

变量类型：double*

存储多音信号各音调的频率值

dPwrDBm

5.2 配置函数

变量类型：double*

存储多音信号各音调的功率值

dPhase

变量类型：double*

存储多音信号各音调的相位值

返回值：

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

数字调制设置

int VSG_SetVectorMod(unsigned int vi, int FilterType, double RolloffFactor, int DataSourceType, int ModMode, double SymbolRate, double FreqFreBiasHz)

函数用途：

设置数字调制参数，包括调制制式、数据源类型、码元速率、滤波器参数和 FSK 调制频偏。

参数列表：

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

FilterType

变量类型：int

滤波器类型，定义如下：

0——奈奎斯特滤波

1——根奈奎斯特滤波

2——高斯滤波

3——矩形滤波

RolloffFactor

变量类型：double

所要设置的滤波器滚降系数

范围：0~1

DataSourceType

变量类型：int

调制信号数据源类型，定义如下：

0——PN9

1——PN11

2——PN15

3——PN16

4——PN20

5——PN21

6——PN23

ModMode

调制信号的调制制式，各调制制式对应的数字定义如下：

0——2FSK
1——4FSK
2——MSK
3——8PSK
6——BPSK
7——D8PSK
8——DQPSK
9——OQPSK
10——QPSK
11——16QAM
12——32QAM
13——64 QAM
14——128 QAM
15——256 QAM
16——512 QAM
17——2ASK
18——4ASK
SymbolRate
变量类型：double
所要设置调制信号的码元速率，单位：Hz，范围：0-100*1e6
FreqFreBiasHz
变量类型：double
FSK 信号所要设置的 FSK 频偏，单位：Hz，范围：0-10*1e6
返回值：
返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

调幅信号设置

int VSG_AMConfigureInternal(unsigned int vi,double depth,int waveform,double freqkHz)

函数用途：
设置调幅信号参数
参数列表：
vi
变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。
Depth
变量类型：double
幅度调制深度，单位：%，范围：0~100
freqkHz
变量类型：double
幅度调制频率，单位：kHz，范围：0~100。

5.2 配置函数

Waveform

变量类型：int

幅度调制波形:

0——正弦波

1——方波

2——三角波

3——锯齿波

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

调频信号设置

int VSG_FMConfigureInternal(unsigned int vi, double deviation,int waveform,double freqkHz)

函数用途:

设置调频信号参数

参数列表:

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

Deviation

调制频偏，单位:kHz,范围：0-100

Waveform

幅度调制信号波形:

0——正弦波

1——方波

2——三角波

3——锯齿波

freqkHz

变量类型：double

幅度调制频率，单位 kHz 范围：0-100

返回值:

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

调相信号设置

int VSG_PMConfigureInternal(unsigned int vi, double rad,int waveform,double freqkHz)

函数用途:

设置调相信号参数

参数列表:

vi

变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。
rad
变量类型：double
调相信号调制相偏，单位：rad，范围：0~3.14
waveform
变量类型：int
调相信号调制波形:
0——正弦波
1——方波
2——三角波
3——锯齿波
freqkHz
变量类型：double
调制频率，单位 kHz 范围：0-100
返回值:
返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

设置脉冲输入类型

int VSG_ConfigurePulseSource(unsigned int vi,int source)
函数用途:
设置脉冲输入类型。
参数列表:
vi
变量类型：unsigned int
初始化函数返回的仪器句柄，与仪器通信。
Source
变量类型：int
所要设置的脉冲输入类型
0——内部
1——外部
返回值:
返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

内部脉冲调制设置

int VSG_ConfigurePulseInternalTrigger(unsigned int vi,double periodus,double widthus)
函数用途:
设置内部脉冲发生器的脉冲宽度和脉冲周期。
参数列表:

5.2 配置函数

vi

变量类型：unsigned int

初始化函数返回的仪器句柄，与仪器通信。

widthus

变量类型：double

脉冲宽度，单位：us，设置范围：小于脉冲周期

periodus

脉冲周期，单位：us，范围：0.05~100000

变量类型：double

返回值：

返回值说明了函数执行的情况：等于 0 表示成功，小于 0 表示失败。

6 错误说明

本章将告诉您如何发现问题并接受售后服务。并说明矢量信号发生器出错信息。

- 错误信息.....64
- 返修方法.....66

6.1 错误信息

矢量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。

6.1.1 错误信息文件

随机光盘存储文件errormessage.pdf，其中记录了完整的错误信息。错误列表以错误序号和错误说明组成。对于同一错误序号对应的多个错误信息，则采用字母顺序以区别不同错误信息。

6.1.2 错误信息说明

- 本地错误信息.....64
- 程控错误信息.....65

6.1.2.1 本地错误信息

- 错误信息查看.....64
- 错误信息说明.....64

1) 错误信息查看

通过界面操作方法：

如果使用过程中在软件界面的正下方状态栏显示有错误提示信息，则说明矢量信号发生器软件运行或硬件出现问题。您根据错误代码和错误提示可以大致判断问题类型，并采取相应措施排除故障。

2) 错误信息说明

矢量信号发生器测量过程中若检测到错误，状态栏会显示告警或错误信息，并以信息图标的不同颜色标注错误级别：

6.1 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	严重错误	测量过程中的严重错误，出现此类错误时，仪器无法正常工作。
	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。

表5.1 字体颜色区分状态指示区错误级别（续表）

颜色	错误级别	错误说明
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。
灰色	信息	步骤设置或测量信息。
	消息	导致下一步操作错误的事件或者状态。

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
数据错误	只针对数字基波段接口：数字I/Q输入数据错误。
FIFO 过载	只针对数字基波段接口：外部输入采样率过高。
输入过载	RF输入端口的信号功率超过规定范围。
无参考	矢量信号发生器未检测到需要的外参考信号输入。
OVEN	OCXO 未达到工作温度，一般开机后几分钟后，OCXO进入正常工作状态后，该错误信息会消失。

6.1.2.2 程控错误信息

1) 错误信息格式及说明

远程控制模式下，错误信息记录在状态报告系统中的错误/事件队列中，可由命令“SYSTem:ERRor?”查询错误信息，格式如下：

“<错误代码>, "<错误队列中错误信息>; <详细错误信息描述>”

举例:

“-110,"数据超界；输入参数超出下界。”

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
1052	Frontend LO is Unlocked This message is displayed when the phase regulation of the local
1060	Trigger-Block Gate Delay Error- gate length < Gate Delay This message is displayed when the gate signal length is not sufficient

表5.3 仪器特性错误信息说明列表（续表）

错误代码	错误说明
1064	Tracking LO is Unlocked This message is displayed when the phase regulation of the local

2) 执行错误信息

错误事件只对应一种错误信息，执行错误信息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。

6.2 返修方法

- 联系我们.....67
- 包装与邮寄.....67

6.2.1 联系我们

若6939矢量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

6.2.2 包装与邮寄

当您的矢量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。如果经联系确认是矢量信号发生器需要返修时，请您用原包装材料和包装箱包装矢量信号发生器，并按下面的步骤进行包装：

1) 写一份有关矢量信号发生器故障现象的详细说明，与矢量信号发生器一同放入包装箱；
2) 用原包装材料将矢量信号发生器包装好，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

包装矢量信号发生器需注意

使用其它材料包装矢量信号发生器，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.....68
- 附录B 错误信息速查表.....69

附录 A SCPI 命令按子系统分类速查表

附表 1 信号发生 SCPI 命令速查表

索引	命令	功能
1	*IDN?	通用指令
2	*RST	通用指令
3	:OUTPut:[STATe]	设置和查询射频输出开关
4	:OUTPut:MODulation[:STATe]	设置和查询调制总开关
5	[:SOURce]:FREQuency[:CW|FiXed]	设置和查询信号发生器输出频率
6	[:SOURce]:ROSCillator:SOURce	设置和查询信号发生器内外参考
7	POWer[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]	设置和查询射频输出信号功率电平
8	[:SOURce]:POWer:ALC[:STATe]	设置和查询环路开关状态
9	[:SOURce]:RADio:CUSTom:ALPHa	设置和查询基带滤波器因子
10	[:SOURce]:RADio:CUSTom:DATA	设置和查询基带调制信号数据源
11	[:SOURce]:RADio:CUSTom:FILTer	设置和查询基带滤波器类型
12	[:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation]	设置和查询基带调制类型为 FSK 模式时调频频偏
13	[:SOURce]:RADio:CUSTom:MODulation[:TYPE]	设置和查询基带调制格式
14	[:SOURce]:RADio:CUSTom:SRATe	设置和查询基带码元速度
15	[:SOURce]:RADio:ARB[:STATe]	查询与设置任意波开关
16	[:SOURce]:RADio:ARB:SEQuence	加载任意波文件
17	[:SOURce]:RADio:ARB:SCLock:RATE	设置和查询任意波采样率
18	[:SOURce]:RADio:MTONe:ARB[:STATe]	查询与设置多音开关
19	[:SOURce]:AM:STATe	设置和查询幅度调制开关状态
20	[:SOURce]:AM:INTernal:FREQuency	设置和查询幅度调制率
21	[:SOURce]:AM[:DEPTh]:LINear	设置和查询线性调制深度
22	[:SOURce]:AM:INTernal:SHAPe	该命令设置调幅信号调制波形
23	[:SOURce]:FM:STATe	设置和查询频率调制开关状态
24	[:SOURce]:FM:INTernal:FREQuency	设置和查询频率调制率
25	[:SOURce]:FM[:DEViation]	设置和查询调频频偏
26	[:SOURce]:FM:INTernal:SHAPe	设置调频信号调制波形
27	[:SOURce]:PM:STATe	设置和查询调相开关状态
28	[:SOURce]:PM:INTernal:FREQuency	设置和查询频率调制率
29	[:SOURce]:PM[:DEViation]	设置和查询调制相偏
30	[:SOURce]:PM:INTernal:SHAPe	设置调频信号调制波形

31	[:SOURce]:PULM:STATe	设置和查询脉冲调制开关状态
32	[:SOURce]:PULM:SOURce	设置和查询脉冲输入类型
33	[:SOURce]:PULM:INTernal:PWIDth	设置和查询内部脉冲发生器的脉冲宽度
34	[:SOURce]:PULM:INTernal:PERiod	设置和查询内部脉冲发生器的脉冲周期

附录 B 错误信息速查表

附表 2 错误信息表

错误代码	错误关键字段	详细错误说明
-65535	ERR_COMMON_INVALIDPARAM1	第 1 个输入参数无效
-65534	ERR_COMMON_INVALIDPARAM2	第 2 个输入参数无效
-65533	ERR_COMMON_INVALIDPARAM3	第 3 个输入参数无效
-65532	ERR_COMMON_INVALIDPARAM4	第 4 个输入参数无效
-65531	ERR_COMMON_INVALIDPARAM5	第 5 个输入参数无效
-65530	ERR_COMMON_INVALIDPARAM6	第 6 个输入参数无效
-65529	ERR_COMMON_INVALIDPARAM7	第 7 个输入参数无效
-65528	ERR_COMMON_INIT	初始化失败
-65527	ERR_COMMON_OPENED	模块已经打开
-65526	ERR_COMMON_RESCNAMEERR	资源名错误
-65525	ERR_COMMON_CANNT_OPEN	获得模块句柄错误
-65524	ERR_COMMON_GET_SERIALID	获得串号失败
-65523	ERR_COMMON_GET_DEV_INFO	获取模块信息失败
-65522	ERR_COMMON_RESET	复位失败
-65521	ERR_COMMON_SELF_TEST	自测试失败
-65520	ERR_COMMON_GETTEMPERATURE	获取温度信息失败
-65519	ERR_COMMON_GET_DATA	采集数据异常
-65518	ERR_COMMON_READCALDATA	获取校准数据失败
-65517	ERR_SELF_CAL	自校准失败
-65516	ERR_FILE_OPEN	打开文件失败
-65515	ERR_BASEBAND_INIT	基带板初始化失败
-65514	ERR_LOBAND_INIT	本振板初始化失败
-65513	ERR_RFBAND_INIT	射频板初始化失败
-65510	ERR_LEVEL_ATT	衰减器设置失败
-65508	ERR_RF_CHANNEL_SET	射频通道设置失败
-65507	ERR_LO_SET	本振设置失败
-65505	ERR_SAMPLE_FREQ_SET	采样频率设置失败
-65502	ERR_MOD_SET	调制输出设置失败
-65500	ERR_ARB_NOTFOUND	查找任意波名字失败
-65497	ERR_DEMOD_MODUTYPE	调制类型设置失败

-65496	ERR_DEMOD_SYMBOLRATE	码元速率设置失败
-65490	ERR_DEMOD_FILTERALPHA	滤波器因子设置失败
-65481	ERR_DEMOD_EQUALIZER_USER	用户自定义滤波器系数设置失败