1465 series
Signal Generator
Programming Manual

mcstest

Approved UK Distributor for Ceyear

Visit: www.mcs-testequipment.com
Email: sales@mcs-testequipment.com
Call: 01492 550 398

MCS Test
New Vision Business Park
Glascoed Road
St Asaph
LL17 0LP

This manual applies to signal generators of models below and based on firmware version 1.0 and above.

- 1465A signal generator (100kHz ~ 3GHz)
- 1465B signal generator (100kHz ~ 6GHz)
- 1465C signal generator (100kHz ~ 10GHz)
- 1465D signal generator (100kHz ~ 20GHz)
- 1465F signal generator (100kHz ~ 40GHz)
- 1465H signal generator (100kHz ~ 50GHz)
- 1465L signal generator (100kHz ~ 67GHz)
- 1465A-V signal generator (100kHz ~ 3GHz)
- 1465B-V signal generator (100kHz ~ 6GHz)
- 1465C-V signal generator (100kHz ~ 10GHz)
- 1465D-V signal generator (100kHz ~ 20GHz)
- 1465F-V signal generator (100kHz ~ 40GHz)
- 1465H-V signal generator (100kHz ~ 50GHz)
- 1465L-V signal generator (100kHz ~ 67GHz)

Preface

Thank you for choosing the 1465 series signal generator developed and manufactured by China Electronics Technology Instruments Co., Ltd (CETI). Our product is high-end, precise and sophisticated, and embraces a high cost performance among the competitors of the same class.

We are devoted to providing for you high-quality products and first-class after-sales service with your most concerns and demands in mind. Our consistent aim is providing excellent quality and good service, and this is our sincere commitment for all users.

Manual No.

AV2.827.1202SCEN

Version

A.3 2018.9

China Electronics Technology Instruments Co., Ltd

Manual Authorization

This manual may be subject to change without notice. CETI reserves all the rights to the final explanation for all the information and terminologies referred to in this manual.

This manual is the property of CETI. Without CETI's permission, any organizations or individuals shall neither alter/temper nor duplicate/transmit this manual for profits; otherwise, CETI reserves the right to pursue any liabilities therefrom.

Product Warranty

The warranty period of this product is 18 months from the date of delivery. Instrument manufacturer will repair or replace the damaged parts according to the actual situation in the warranty period. In order to ship the product back to the manufacturer for repairs, the user must pay shipping and handling fees. After maintenance and repair, the manufacturer will ship the product back to the user along with reimbursement of the shipping and handling fees.

Product Quality Certificate

This product is certified to fulfill the standards indicated in this manual from the day of delivery. Calibration measurements have been carried out based on national standards. Related information is available to the user for reference.

Quality/Environmental Management

The quality and environmental management systems have always been implemented during development, manufacturing and test of this product. China Electronics Technology Instruments Co., Ltd (CETI) has been properly qualified and certified ISO 9001 and ISO 14001 management system standards.

Safety Precautions

CAUTION

CAUTION indicates important information rather than a danger. It reminds the user to be cautious of a certain operation process, operation method or the similar. Failure to follow the rules or operate correctly may cause the damage to the instrument or loss of important data. The conditions indicated by CAUTION should be fully understood and met before the next operation.

NOTE

NOTE indicates an information prompt. It reminds the user to pay attention to the instrument or a certain operation process, operation method or the similar, so as to guide the instrument operator to correctly use the instrument.

Table of Contents

1 About This Manual 1
1.1 About This Manual 1
1.2 Related Documents 1
2 Remote Control 3
2.1 Remote control basis 3
2.1.1 Remote Interface 3
2.1.2 Message 6
2.1.3 SCPI 7
2.1.4 Command sequence and synchronization 15
2.1.5 Status reporting system 17
2.1.6 Programming considerations 26
2.2 Remote interface and its configuration 27
2.2.1 LAN 27
2.2.2 GPIB 28
2.3 I/O library 29
2.3.1 Overview of I/O library 29
2.3.2 Installation and configuration of I/O library 30
3 SCPI 33
3.1 Command instruction 33
3.2 Common command 33
*IDN? 34
*RCL <Value> 34
*RST 34
*SAV <Value> 34
*TRG 34
3.3 Instrument subsystem command 34
3.3.1 OUTPut subsystem 35
3.3.2 FREQuency subsystem 36
3.3.3 POWer subsystem 41
3.3.4 LIST subsystem 47
3.3.5 LFOutput subsystem 52
3.3.6 SWEep subsystem 56

3.3.7 PULM subsystem 59
3.3.8 AMPLitude MODulation subsystem 67
3.3.9 FREQuency MODulation subsystem 72
3.3.10 PHASe MODulation subsystem 76
3.3.11 Digital MODulation subsystem 81
3.3.12 MEMory subsystem 107
3.3.13 ROSCillator subsystem 109
3.3.14 SYSTem subsystem 109
4 Programming example 113
4.1 Basic operation example 113
4.1.1 VISA library 113
4.1.2 Example running environment 114
4.1.3 Initialization and default status setting 115
4.1.4 Sending of setting command 116
4.1.5 Reading of instrument status 116
4.2 Advanced operation example 117
4.2.1 Setting and checking the frequency of LAN interface 117
4.2.2 Setting and checking the frequency of GPIB interface 118
5 Error Description 121
5.1 Error information 121
5.1.1 Description of error information 121
5.2 Repair Method 123
5.2.1 Contact us 123
5.2.2 Packaging and mailing 123
Appendixes 125
Appendix A Lookup Table of SCPI by Subsystem 125
Appendix B Lookup Table of Error Information 136

1 About This Manual

This chapter introduces the function, compositions and main content of the Programming Manual of the 1465 series signal generator as well as other related documents provided to the user.

- About this manual 1
- Related Documents.....1

1.1 About This Manual

This manual introduces the remote control and the SCPI operation method of the 1465 series signal generator, as well as the programming examples and the basic concept of the I/O function library to facilitate the user to quickly master the programming method. To facilitate your familiarity with the instrument, please read this manual carefully before operating the instrument, and then follow the instructions of manual.

SCPI (Standard Commands for Programmable Instruments) defines standards and methods for remote control of the instruments, and it is also the programming language for programmable instruments for electronic test and measurement. The SCPI is based on the specifications and types in IEEE-488.2. For details, please visit http://www.scpiconsortium.org. This manual describes in detail the SCPIs of the 1465 series signal generator.

The chapters of the Programming Manual include:

- **Remote Control**

  This chapter introduces the remote control methods of the instrument so that the user can rapidly master the method to control the instrument in a remote way. It is further divided into the following three sections: remote control basis, which introduces the concepts related to remote control, software configuration, remote interface, SCPI, etc.; instrument interface configuration method, which introduces the connection method and software configuration method of the remote interface of the 1465 series signal generator; the I/O function library, which introduces the basic concept of the instrument driver and the basic installation and configuration of the IVI-COM/IVI-C driver.
- **SCPI**

  The common command, instrument-specific command and compatibility command are introduced by category, and functions, parameters, and examples of the SCPI are described one by one.
- **Programming Examples**

  The basic programming examples and advanced programming examples are given and described in the form of explanatory note and example code, so as to facilitate the user to quickly master the programming method of the signal generator.
- **Error Description**

  This chapter includes error information description and repair methods.
- **Appendixes**

  This chapter provides the necessary remote control reference information of the 1465 series signal generator, including the SCPI lookup table and error information.

1.2 Related Documents

The product documents related to 1465 series signal generator include:

1.2 Related Documents

- Quick Start Guide
- User Manual
- Programming Manual
- Online support

Quick Start Guide

This manual introduces the set-up of the instrument as well as the basic operation methods of measurement with the aim of enabling users to quickly understand the features and operational procedures of the instrument. Main chapters included in this manual are as follows:

- Preparation before Use
- Typical Applications
- Getting Help

User Manual

This manual gives a detailed introduction of features and operation methods of the instrument, including information about configuration, measurement, remote control, maintenance, etc. for the purpose of guiding the user to fully understand the features of the product and master the common instrument test methods. Main chapters included in this manual are as follows:

- About This Manual
- Overview
- Start Guide
- Operation Guide
- Menu
- Remote Control
- Fault Diagnosis and Repair
- Specifications and Test Methods
- Appendixes

Programming Manual

This manual describes in detail the basics of remote programming, SCPI basics, SCPI, programming examples, I/O driver library, etc. for the purpose of for the purpose of guiding the user to master the SCPIs and methods of the instrument quickly and comprehensively. Main chapters included in this manual are as follows:

- Remote Control
- SCPI
- Programming Examples
- Error Description
- Appendixes

Online support

Online help is integrated in the instrument product to provide quick text navigation help for user local and remote control operation. The hard keys on the instrument front panel or the user interface toolbars may be activated with their corresponding shortcut keys. The contents are the same as those in the user manual.

2 Remote Control

This chapter introduces the remote control basis as well as the remote interface and its configuration method of the 1465 series signal generator, and also briefly describes the concept and classification of the I/O driver library so that the user can have a preliminary knowledge about the remote control of this instrument. The specific content includes:

- Remote control basis 3
- Remote interface and its configuration 27
- I/O Function Library 29

2.1 Remote control basis

- Remote Interface 3
- Message 6
- SCPI 7
- Command Sequence and Synchronization 15
- Status Reporting System 17
- Programming Considerations 26

2.1.1 Remote Interface

The instrument with remote control function generally supports four kinds of remote control interfaces, i.e. LAN and GPIB, and the type of port supported by the instrument is determined by the function of the instrument.

The description of the remote interface and associated VISA addressing string is as shown in the following table:

Table 2.1 Types and VISA Addressing Strings of Remote Control Interfaces

Remote Interface	VISA Addressing String	Description
LAN (Local Area Network)	**VXI-11 protocol:** TCPIP::host_address[::LAN_device_name][::INSTR] **Raw socket protocol:** TCPIP::host_address::port::SOCKET	Controller realizes remote control by connecting the instrument via the network port on the rear panel of the instrument. For the specific protocol, please refer to: 2.1.1.1 LAN interface
GPIB (IEC/IEEE Bus Interface)	GPIB::primary address[::INSTR] 4	Controller realizes remote control by connecting the instrument via the

		port on the rear panel of the instrument. Compliance with the bus interface standard of IEC 625.1/IEEE 418. For details, please refer to: 2.1.1.2 GPIB interface

- LAN port 4
- GPIB interface 6

2.1.1.1 LAN interface

The signal generator is available for remote control via computer in 10Base-T and 100Base-T LAN, in which various instruments are integrated into a system and controlled by network computer. In order to realize the remote control within the LAN, the signal generator shall be preinstalled with the port connector, network card and relevant network protocol, and configured with relevant network service. And, the controller computer within the LAN shall also be preinstalled with the instrument control software and VISA library. The three working modes of the network card include:

- 10 Mbit/s Ethernet IEEE802.3;
- 100 Mbit/s Ethernet IEEE802.3u;
- 1 Gbit/s Ethernet IEEE802.3ab.

Control computer and signal generator need to be connected to a common TCP/IP protocol network via network interface. The cable between the computer and the signal generator is a commercial RJ45 cable (shielded or unshielded Category 5 twisted pair). During data transmission, data packet transmission will be adopted, and LAN transmission is faster. The cable length between the computer and the signal generator shall generally not exceed 100 m (100Base-T and 10Base-T). For more information about LAN communication, refer to: http://www.ieee.org。 The knowledge of LAN interface is introduced hereinafter.

1) IP address

When the signal generator is remotely controlled via LAN, the physical network connection shall be guaranteed to be smooth. The address of the signal generator is set to the subnet where the main control computer is located via menu “local IP”. For example: If the IP address of main control computer is 192.168.12.0, the IP address of the signal generator shall be set to 192.168.12.XXX, where XXX is a value between 1 and 255, and the default network port number for the signal generator communication is 5001.

Only the IP address is required to establish a network connection. The VISA addressing string is as follows:

TCPIP::host address[::LAN device name][::INSTR] or

TCPIP::host address::port::SOCKET

Where,

- TCPIP - network protocol used;
- host address - IP address or host name of the instrument, for identification and control of the controlled instrument;
- The LAN device name defines the handle number of the protocol and sub-device (optional);
  - The VXI-11 protocol is adopted for the 0# equipment;
  - The newer high speed LAN instrument protocol is adopted for the 0# high speed LAN instrument;
- The INSTR represents the instrument resource type (optional);
- The port represents the socket port number;
- SOCKET is resource class of original network socket.

Example:

- The IP address of the instrument is 192.1.2.3, and the valid resource string of the VXI-11 protocol is:
  TCPIP::192.1.2.3::INSTR
- When the raw socket connection is created, the following addressing string can be used:
  TCPIP::192.1.2.3::5001::SOCKET

NOTE

Method for identification of multiple instruments in the remote control system

If multiple instruments are connected to the network, they can be identified by their individual IP address and associated resource string. The main control computer uses the respective VISA resource string for instrument identification.

2) VXI-11 protocol

The VXI-11 standard is based on the ONC RPC (Open Network Computing Remote Procedure Call) protocol, which is the network/transport layer of the TCP/IP protocol. TCP/IP network protocol and related network services are pre-configured. In communication, such connection-oriented communication follows sequential exchange and can identify the interruption of connection to ensure no information is lost.

3) Socket communication

The TCP/IP protocol connects the signal sources in the network through the LAN socket. As a basic computer network programming method, the socket enables applications with different hardware and operating systems to communicate in the network, This method enables two-way communication between the signal generator and the computer via port.

The socket is a special software class that defines the necessary information for network communication such as IP address and device port number and integrates some basic network programming operations. Sockets can be used in the operating system installed with a packaged library. UNIX Berkeley socket and Winsock are commonly used.

2.1 Remote control basis

The socket in the signal generator is compatible with Berkeley socket and Winsock through Application Program Interface (API). In addition, other standard sockets are also compatible through the API. When the signal generator is controlled using SCPI, the socket program created in the program issues command. The socket port number of the signal generator must be set before LAN socket is used. For the signal generator, the socket port number is set to 5,000.

2.1.1.2 GPIB interface

The GPIB interface is a widely-used instrument remote interface currently, which can be connected with different kinds of instruments through the GPIB cable and can establish the test system with the main control computer. To realize remote control, the main control computer shall be preinstalled with the GPIB bus card, driver and VISA library. During communication, the main control computer will address the controlled instrument through the GPIB bus address firstly. The user can set the GPIB address and ID for querying strings, and the GPIB communication language can be set to the SCPI form by default.

The operation of the GPIB and its relevant interface is defined and described in details in the ANSI/IEEE standard 488.1-1987 and the ANSI/IEEE standard 488.2-1992. For details, refer to IEEE website: http://www.ieee.org。

As the GPIB processes information in bytes and the data transmission rate can reach 8 MBps, the GPIB data transmission is faster. The data transmission rate is limited by the distance between the device/system and the computer. When GPIB is connected, the following points shall be noted:

- Up to 15 instruments may be set up through the GPIB interface;
- The total length of the transmission cable shall not exceed 15 m, or shall not exceed twice of number of instruments in the system. The maximum length of transmission cable between equipment generally does not exceed 2 m.
- If you connect multiple instruments in parallel, you need to use “OR” connectors.
- The terminal of the IEC bus cable shall be connected with the instrument or the controller computer.

2.1.2 Message

Messages transmitted by data cable fall into the following two categories:

1) Interface message

During communication between the instrument and the main control computer, it is necessary to pull down the attention line and then the interface message can be transmitted to the instrument through the data line. Only the instrument with the GPIB bus functions can send the interface message.

2) Instrument message

For the structure and syntax of instrument message, refer to Section "5.1.4 SCPI". Instrument message can be divided into command and instrument response according to the different transmission directions. All remote interfaces use instrument message in the same method unless otherwise stated.

a) Commands:

A command (programming message) is a message transmitted from the main control computer to the instrument for remote control of instrument functions and query of status information. It falls into the following two categories:

- Based on the impact on the instrument:
  — Set command: change the instrument setup status, for example: reset or set the frequency.

— Query command: query and return data, for example: identify instrument or query parameter values. The query command is always ended with a question mark.

➢ Based on the definition in the standard:

— Common command: with functions and syntax defined in IEEE488.2, applicable to all types of instrument (if implemented) for management of standard status register, reset, self-test, etc.

— Instrument control command: Instrument characteristic command for instrument functions. For example: set the frequency.

The syntax also follows SCPI specification.

b) Instrument response:

The instrument response (response message and service request) is the query result information sent by the instrument to the computer. This information includes measurement result and instrument status.

2.1.3 SCPI

● Brief introduction to SCPI..... 6
● SCPI description 7

2.1.3.1 Brief introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) is a set of commands established for all instruments based on IEEE488.2 mainly to achieve the universality of SCPI, i.e.the same SCPI is generated and issued for the same function.

The SCPI consists of a command header and one or more parameters which are separated by a space. The command header contains one or more key fields. The command with question mark as postfix is a query command. Commands are divided into common commands and instrument-specific commands that are different in syntactic structure. SCPI has the following features:

1) The SCPI is established for the test functions rather than instrument operation description.
2) The SCPI reduces the repetition of the realization process of similar test functions, thus ensuring the programming compatibility;
3) Remote control message is defined in a layer that is independent of the communication physical layer hardware.
4) The SCPI is irrelated with the programming methods and languages, and the SCPI test program is easy to be transplanted;
5) The SCPI is scalable so that it is applicable to measurement control on different scales.
6) Scalability makes SCPI a “Live” standard.

If you are interested in learning more about SCPI, please refer to:

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.

For the collection of SCPIs, classification and description of 1465 series signal generator, please refer to:

1) “3 SCPI” of this manual;
2) Appendix B Lookup Table of SCPI in this manual.

2.1.3.2 SCPI description

- General terms 7
- Command type 8
- Instrument-specific command syntax 9
- Command tree 10
- Command parameter and response 11
- Systems of Values in Commands 13
- Command line structure 14
- Unit Description 14

1) General terms

For the purpose of this section, the following terms should apply. It is necessary to know about the exact definitions of these terms for a better understanding of the content in various chapters.

Controller

The controller is any computer used to communicate with the SCPI equipment. The controller may be a personal computer, a small computer or a card inserted onto a cage. Some artificial intelligence equipment can also be used as a controller.

Equipment

The equipment is any device that supports SCPI. Most equipment is electronic measuring or excitation equipment and uses the GPIB interface for communication.

Remote control message

The remote control message is a combination of one or more correctly formatted SCPIs. It guides the equipment to measure and output the signal.

Response message

The response message is a data set that specifies the SCPI format. It is always sent from the equipment to the controller or listener to remind the controller of the internal condition or measured value of the equipment.

Command

A command is an instruction in compliance with the SCPI standard. The combination of controller commands forms a message. In general, a command includes the keyword, parameter and punctuation.

Event command

An event-type SCPI can't be queried. An event command generally has no corresponding key settings on front panel. Its function is to trigger an event at a particular moment.

Query

Query is a special command. When the controller is queried, it is necessary to return to the response message in conformity with syntax requirement of the controller. The query statement is always ended with a question mark.

2) Command type

There are two types of SCPIs: common commands and instrument-specific commands. Figure 5.2 shows the difference between two commands. Common commands are defined in IEEE 488.2 to manage macros, status registers, synchronization, and data storage. Common commands are easy to

recognize as they all begin with an asterisk. For example, *IDN? , *OPC and *RST are common commands. Common commands don't belong to any instrument-specific command. The instrument uses the same method to interpret them without consideration to the current path setting.

It is very easy to identify instrument-specific commands because they contain a colon (:). Colons are used between the beginning of command expression and keywords, e.g.: FREQuency[:CW？]. Instrument-specific commands are divided into command subsets of corresponding subsystem according to the functional block inside the instrument. For example, the power subsystem (:POWer) contains the power-related command while the status subsystem (:STATus) contains the command for the status control register.

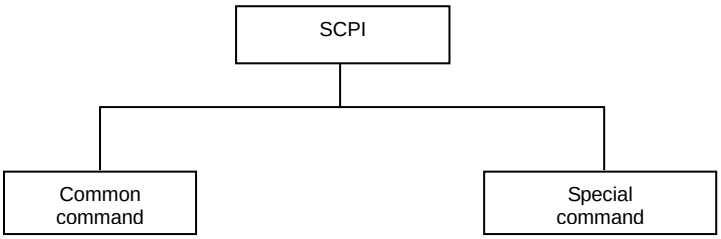

Figure 2.1 SCPI Type

3) Instrument-specific command syntax

A typical command consists of keywords with colon as prefix. These keywords are followed by parameters. An example of syntax statement is shown below.

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

In the above example, the [:LEVel] portion of the command immediately follows :POWer with no separating space. The portion following [:LEVel] is MINimum|MAXimum that is a parameter. There is a space between the command and the parameter. The description of other parts of the syntax expression is as shown in Table 2.2 and Table 2.3.

Table 2.2 Special Characters in Command Syntax

Symbol	Meaning	Example
|	The vertical line between the keyword and the parameter represents a variety of options.	[:SOURce]:AM:SOURce EXTernal|INTernal EXTernal 和 INTernal alternative choices
[]	Keywords or parameters in square brackets are optional when they form a command. These implied keywords or parameters are executed even when they are ignored.	[:SOURce]:AM[:DEPTh]:EXPonential? SOURce and DEPTh are dispensable.
< >	The part inside the angle brackets can't be used literally in the command, instead, it represents the part that must be contained.	[:SOURce]:FREQ:STOP <val> In this command, <val>must be replaced by an actual frequency and unit. For example: :FREQ:STOP 3.5GHz
{ }	The part inside the braces indicates that the parameter is optional.	[:SOURce]:LIST:POWer <val>{,<val>}

		For example: LIST:POWer 5

Table 2.3 Command Syntax

Character, Keyword and Syntax	Example
Upper-case characters represent the minimum character set required by command execution.	[:SOURce]:FREQuency[:CW]?, FREQ is the short-format part of the command.
The lower-case characters portion of command is optional; Such flexible format is called “flexible listening”. For more information, please refer to “Command Parameter and Response”.	:FREQuency :FREQ, :FREQuency or :FREQUENCY, any of which is correct.
When a colon is placed between two command mnemonics, it moves the current path down one level in the command tree. For more information, please refer to the command path part of “Command Trees”.	:TRIGger[:SEQuence]:SOURce? TRIGger is the topmost keyword of this command.
If a command requires more than one parameter, you must separate adjacent parameters using a comma. Parameters do not affect path layers as they are not the portion of the command path.	[:SOURce]:LIST:DWELl <val>{,<val>}
A semicolon separates 2 adjacent commands without affecting the current path.	:FREQ 2.5GHZ; :POW 10DBM
White space characters, such as <space> or <tab>, are generally ignored so long as they don’t appear between or among . However, the command and parameter must be separated by a blank character, without affecting the current path.	:FREQ uency or :POWer :LEVel6.2 is not allowed. :LEVel and 6.2 must be separated by a space. i.e. :POWer:LEVel 6.2

4) Command tree

Most remote control programming tasks involve instrument-specific commands. When such a command is parsed, the SCPI will use a structure similar to the file structure, and it is called as a command tree, as shown in Figure 2.2:

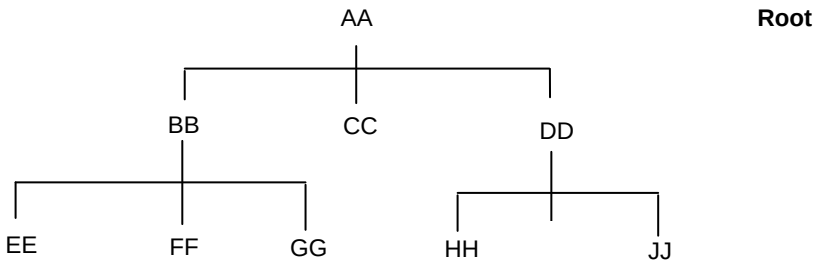

Figure 2.2 Simplified Command Tree Diagram

The top command is root command, or simply “root”. In the case of command parsing, the command at the next layer is reached by following a specific route based on the tree structure. For example: :POWer:ALC:SOURce？ where, POWer stands for AA,: ALC stands for BB, :SOURce stands for GG, and the whole command path is (:AA:BB:GG).

A software module in the instrument software—the command interpreter—is responsible for parsing

each received SCPI. The command interpreter breaks up the command into individual command element using a series of rules for identifying the command tree path. After the current command is parsed, the current command path remains unchanged. In this way, the subsequent commands can be parsed more quickly and efficiently because the same command keyword may appear in different paths. After the power-on or *RST (reset) operation of the instrument, the current command path is reset as the root.

5) Command parameter and response

The SCPI defines different data formats in the use of the remote control and response messages to conform to the principles of “flexible listening” and “accurate speaking”. For more information, please refer to IEEE 488.2. “Flexible Listening” means that the formats of commands and parameters are flexible.

For example, the signal generator is set to a frequency reference status command, i.e. FREQuency:REFerence:STATe ON|OFF|1|0.

The following command formats are used to set the frequency reference function as “On”:

:FREQuency:REFerence:STATe ON, :FREQuency:REFerence:STATe 1,

:FREQ:REF:STAT ON, :FREQ:REF:STAT 1.

Each parameter type corresponding to one or more response data types. In the case of query, a numeric parameter will return a data type, and the response data is exact and strict, which is called “Accurate Speaking”.

For example, if you query the power state (:POWer:ALC:STATe?), when it is turned on, the response is always 1, regardless of whether you previously sent :POWer:ALC:STATe 1 or :POWer:ALC:STATe ON.

Table 2.4 Types of SCPI Parameter and Response

Parameter Type	Response Data Type
Numeric	Real or Integer
Extended Numeric	Integer
Discrete	Discrete
Boolean	Numeric Boolean
String	String
Block	Definite Length Block
	Indefinite Length Block
Non-decimal numeric	Hexadecimal
	Octal
	Binary

Numeric parameter

Numeric parameters can be used in both instrument-specific commands and common commands. It receives all common decimal systems including signs, decimal point and scientific notation. If a certain piece of equipment only receives a specified type of numeric parameter such as an integer, it will automatically round off the received numeric parameter.

Examples of numeric parameter:

0	no decimal point
100	decimal point optional
1.23	with sign digit
4.56e<space>3	space allowed after exponent marker e

2.1 Remote control basis

-7.89E-01	exponent marker e may be upper or lower case
+256	leading + allowed
5	decimal point may be leading

Extended numeric parameter

Most measurements related to Instrument-specific commands use extended numeric parameters to specify the physical quantities. Extended numeric parameters receive all numeric parameters and additional special values. All extended numeric parameters receive MAXimum and MINimum as parameter values。 Other special values, such as: UP and DOWN are received by the instrument parsing capability. SCPI table will list all valid parameters.

Note: Extended numeric parameters are not applicable to common commands or STATus subsystem commands.

Examples of extended numeric parameters:

101	Numeric parameter
1.2GHz	GHz may be used an exponent (E009)
200MHz	MHz may be used an exponent (E006)
-100mV	-100 millivolt
10DEG	10 degrees
MAXimum	maximum valid setting
MINimum	minimum valid setting
UP	increase a step
DOWN	decrease a step

Discrete parameter

When there are a finite number of parameter values to be set, discrete parameters are used for identification. A discrete parameter uses mnemonics to represent each valid setting. Like the SCPI mnemonics, the discrete parameter mnemonics can be set in long and short formats, with both capitalized and lowercase characters.

The following example illustrates the combined use of discrete parameter and command.

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

BUS	GPIB, LAN, RS-232 trigger
IMMediate	Immediate trigger
EXTernal	External trigger

Boolean parameter

Boolean parameters represent a single binary condition that is either true or false. There are only four possible representations for a Boolean parameter.

Samples of Boolean parameters:

ON	Logical true
OFF	Logical false
1	Logical true
0	Logical false

String parameter

A string parameter allows the ASCII string to be sent as a parameter. Single and double quotes are used as separators.

Examples of string parameter:

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

Real response data

Large portions of measurement data are real. They are formatted as basic decimal notation or scientific notation. Most advanced programming languages all support these two formats.

Examples of real response data:

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

Integer response data

The integer response data are a decimal expression of an integer with the sign bit. When the status register is queried, the integer response data will be mostly returned.

Examples of integer response data:

0	Sign digit optional
+100	Leading + allowed
-100	Leading – allowed
256	No decimal point

Discrete response data

The discrete response data and discrete parameters are basically the same. The main difference is that the discrete response data can only be returned in the short format with capitalized characters.

Examples of discrete response data:

INTernal	Amplitude level control mode is internal
EXTernal	Amplitude level control mode is external
MMHead	Amplitude level control mode is millimeter wave source module

Numeric Boolean response data

The Boolean response data returns a binary value of 1 or 0.

String response data

The string response data and string parameters are the same. The main difference is that the string response data use double quotes rather than single quotes as the separator. The string response data can also be inserted with double quotes inside which there can be no characters.

Examples of string response data:

"This is a string"
"one double quote inside brackets: ("")"

6) Systems of Values in Commands

The value of the command can be entered in binary, decimal, hexadecimal or octal format. In the binary, hexadecimal, or octal format, a suitable identifier should be added in front of the value. In the decimal (default) format, an identifier isn't required.

2.1 Remote control basis

When the value without an indicator is entered, the equipment will ensure that it is entered in decimal format. The identifiers required in all formats are listed as follows:

- #B indicates that this digit is a binary value.
- #H indicates that this digit is a hexadecimal value.
- #Q indicates an octal number.

The representations of the decimal value 45 in the SCPI are given as follows:

#B101101

#H2D

#Q55

The following example shows setting of the RF output power as 10 dBm (or the value equivalent to the current selected unit including DBUV or DBUVEMF) with the hexadecimal value 000A.

:POW #H000A

In a non-decimal format, the measurement unit such as DBM or mV isn't used together with the value.

7) Command line structure

A command line may contain multiple SCPIs. To indicate the end of the current command line, the following methods can be used:

- Enter;
- Enter and EOI;
- EOI and the last data byte.

Commands in command line are separated by semicolons, and commands for different subsystems begin with a colon. For example:

MEM:COPY:NAME Test1, MeasurementXY;FREQ:CW 5GHz.

This command line contains two commands, in which the first one belongs to the MEM subsystem and the second one belongs to the FREQ subsystem. If the adjacent commands belong to the same subsystem, the command path will be partially repeated and the command can be abbreviated. For example: For example:

FREQ:CW 5GHZ;FREQ:OFFSet 3GHz

This command line contains two commands, both of which belong to the FREQ subsystem and have the same level. Therefore, the second command can begin with the subordinate to FREQ and may not begin with a colon, which can be abbreviated to the following command line:

FREQ:CW 5GHz;OFFS 3GHz

8) Unit Description

The programmable commands provided herein support the frequency units Hz, kHz, MHz and GHz. The programmable commands related to the frequency support parameters without unit, and if the frequency unit is not provided, the default unit is Hz; the power unit dBm is supported currently, and if the parameter unit is not provided, the default unit is dBm; the gain and attenuation unit dB is supported, and if the parameter unit is not provided, the default unit is dB; the time units ns, us, ms and s are supported, and if the parameter unit is not provided, the default unit is s. See the range of parameters in each command for the unit type of programmable command parameters provided herein.

2.1.4 Command sequence and synchronization

IEEE488.2 defines the difference between overlapped commands and sequential commands:

- Sequential commands are sequences of commands that are executed continuously. Usually, each command is executed fast.
- Overlapped commands indicate that the previous command is not executed automatically before the next command is executed. Normally overlapped commands take longer to process and allow the program to process other events synchronously.

Even if multiple commands are set in a command line, they are not necessarily executed in the order in which they are received. In order to ensure that the commands are executed in a certain order, each command must be sent as a separate command line.

Example: Command line contains set and query commands

If multiple commands in a command line contain query commands, the query result is unpredictable.

The following command returns a fixed value:

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

Return value: 1000000000 (1GHz)

The following command returns an unfixed value:

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

The returned result may be the current frequency start value, because the host program will delay execution of the command. If the host program executes the command after receiving it, the returned value may also be 1 GHz. **This signal generator has not supported the overlapped command yet.**

NOTE

Set command and query command are sent separately

General rules: In order to ensure the correctness of the returned result from the query command, the setting command and the query command shall be sent in different program control messages.

2.1.4.1 Preventing overlapping execution of the command

In order to prevent the overlapped execution of commands, multiple threads or commands: *OPC, *OPC? or *WAI can be used. These three commands can be executed only after the hardware is set. During programming, the computer can be forced to wait for some time to synchronize certain events. The details are separately described below:

- **Controller program uses multiple threads**

  Multi threads are used to wait for completion of the command and achieve synchronization of GUI and program control, that is, a single thread waits for completion of *OPC?, without impeding the execution of the GUI or remote control thread.

- **The usage of the three commands in synchronous execution is shown in the table below:**

Table 2.5 Command Syntax

Method	Actions to be Executed	Programming Method
*OPC	Set the operation completion bit is set.	Set ESE BIT0; Set SRE BIT5; Send the overlapped command and *OPC; Wait for service request (SRQ) SRQ represents the completion of execution of the overlapped command
*OPC?	Stop executing the current command 1。 The command is returned only when the operation completion bit in the ESR is set, which indicates that the previous command is processed.	Terminate the processing of the current command before executing other commands. Send this command directly after the current command.
*WAI	Before executing *WAI, wait until all commands are sent and continue processing the uncompleted commands.	Terminate the processing of the current command before executing other commands. Send this command directly after the current command.

If the processing time of the overlapped command is short, the command *WAI or *OPC can be used after use of the overlap command to achieve command synchronization. In order to synchronously execute other tasks when the computer or instrument is waiting for the completion of execution of overlapped commands, the following synchronization technologies can be adopted:

- **OPC and service request**

1) Set the OPC mask bit (bit0) of ESE to *ESE 1;
2) Set the bit5 of SRE to *SRE 32 to enable the ESB service request;
3) Send the overlapped command and *OPC;
4) Wait for the service request signal.
   SRQ represents the completion of execution of the overlapped command

- **OPC? and service request**

1) Set the bit4 of SRE to *SRE 16 to enable the MAV service request;
2) Send overlapped commands and *OPC? ；
3) Wait for the service request signal.
   SRQ represents the completion of execution of the overlapped command

- **Event Status Register (ESE)**

1) Set the OPC mask bit (bit0) of ESE to *ESE 1;
2) Send the overlapped command only and do not send *OPC, *OPC or *WAI;
3) Send “*OPC;*ESR?” in the timer for cyclic query of completion status of operation.
   If the return value (LSB) is equal to 1, this indicates that the overlap command has been executed.

- ***OPC? and short timeout**

1) Send the overlapped command only and do not send *OPC, *OPC or *WAI
2) Send “<short timeout>; *OPC?” in the timer, so as to query the completion status of operation circularly;
3) If the return value (LSB) is equal to 1, this indicates that the overlap command has been executed. In case of timeout, the device is in the operational process.
4) Reset the timeout value to the old value;
5) Send the command “SYStem:ERRor?” to clear the error queue and delete the “-410, Query Interrupted” message.
   If the return value (LSB) is equal to 1, this indicates that the overlap command has been executed.

2.1.5 Status reporting system

The status reporting system will save all operation status information of the current instrument, including error information. Such information is stored in the status register and error queue respectively and can be queried through the remote interface.

- Structure of Status Register 17
- Structure of SCPI Status Register 19
- Instruction on Status Register 20
- Application of Status Reporting System 23
- Reset status reporting system 25

2.1.5.1 Structure of the status register

Please refer to the hierarchical structure of status register below:

2.1 Remote control basis

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

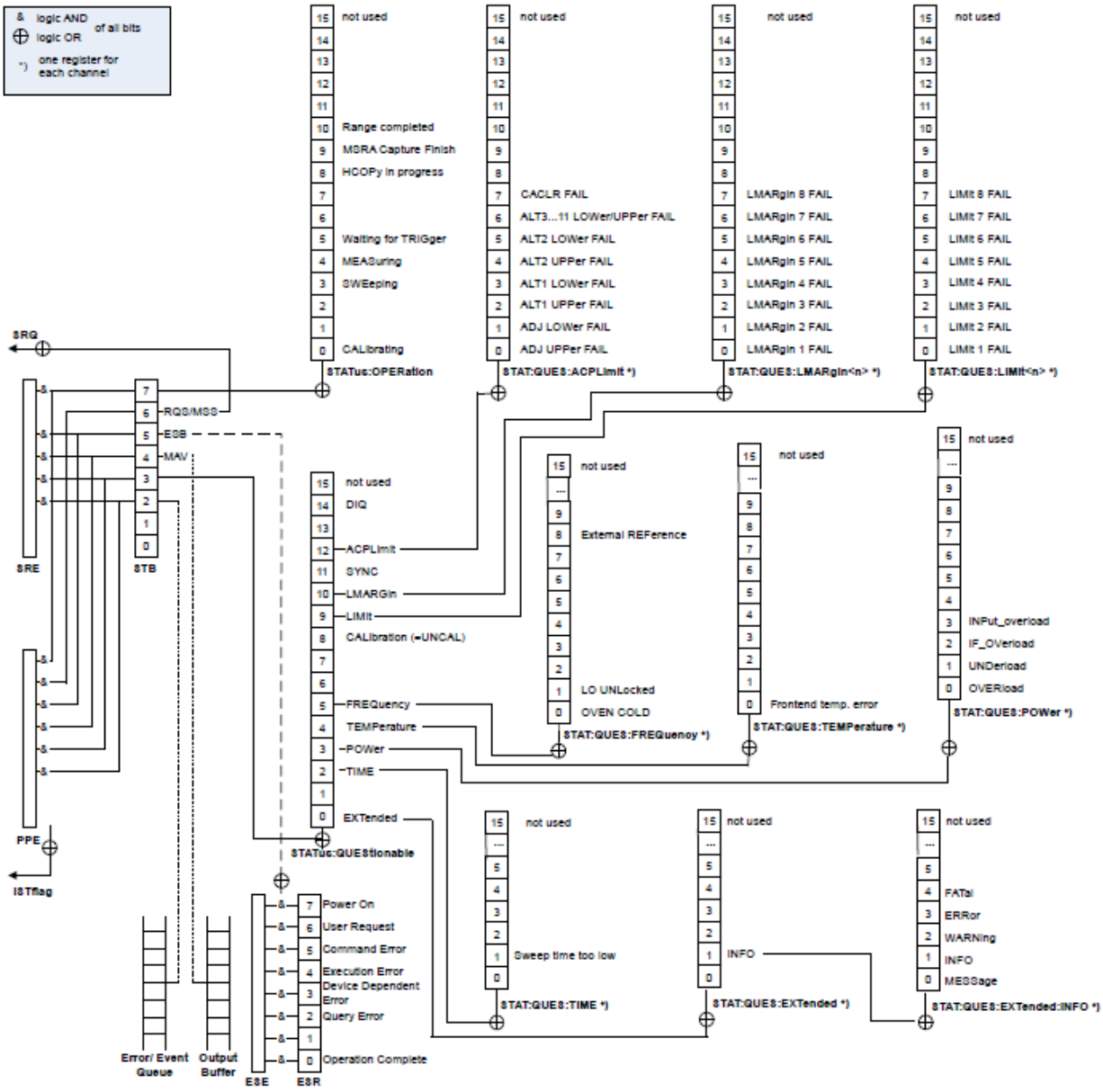

Fig. 2.3 Hierarchical Structure of Status Register

The register classification is described as follows:

1) STB, SRE

   Status Byte (STB) register and its associated mask register, Service Request Enable (SRE) register, constitute the top-level register of the status reporting system. The STB saves the general working status of the instrument by collecting low-level register information.

2) ESR, SCPI status register

   STB receives the information of the following registers:

   - The value of Event Status Register (ESR) and Event Status Enable (ESE) mask register.
   - SCPI status registers include: STATus:OPERation and STATus:QUEStionable registers

(SCPI definition), they contain the specific operating information of the instrument. All SCPI status registers have the same internal structure (refer to “Section 2.1.5.2 Structure of SCPI Status Register” in this Programming Manual for details)

3) IST, PPE

Similar to the SRQ, an individual bit of the IST mark ("Individual STatus") is a combination of all statuses of the instrument. The associated parallel query enable register (PPE) determines which data bits of the STB act on the IST mark.

4) Output buffer

The output buffer stores the message returned by the instrument to the controller. It doesn’t belong to the status reporting system but determines the value of the MAV bit of STB.

For detailed description of the above register, refer to “Section 2.1.5 Status Reporting System” in this Programming Manual.

NOTE

SRE, ESE

The SRE can be used as an enable part of the STB. Similarly, the ESE can be used as an enable part of the ESR.

2.1.5.2 Structure of SCPI Status Register

Each standard SCPI register contains five parts. Each part contains 16 data bits and is functionally independent. For example: Each hardware status is allocated with a data bit that is effective for all the five parts of the register. If the Bit15 is set to 0, it means that the value of the register is a positive integer.

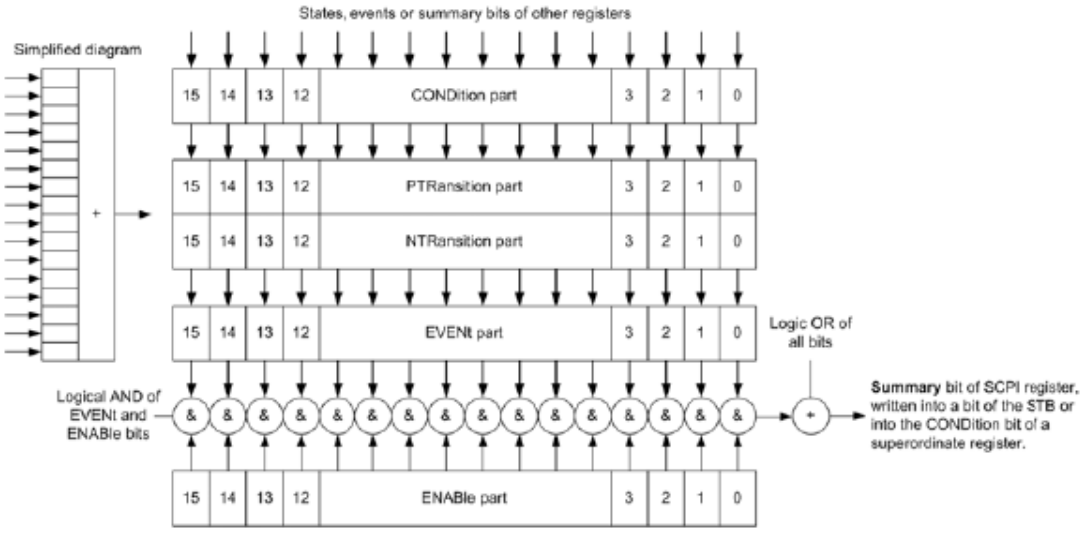

Fig. 2.4 Structure of Status Register

The above Figure shows that the status register is composed of 5 parts, which are described as follows:

- **Condition register**

This part will be directly written by hardware or low-level register digit, which will reflect the current working state of the instrument. This register is read-only and cannot be written. Reading will not clear any value.

2.1 Remote control basis

- **Positive and negative conversion register**

Two transfer registers define the status transfer bit of the condition register stored in the event register.

The positive conversion register is similar to a conversion filter. When an event bit of the condition register is changed from 0 to 1, the associated PTR bit determines if the event bit should be set to 1, as described below:

- If the PTR bit is equal to 1, the event bit should be set.
- If the PTR bit is equal to 0, the event bit should not be set.

The positive conversion register is readable and writable, and its reading will not clear any value.

The negative conversion register is similar to a conversion filter. When an event bit of the condition register is changed from 1 to 0, the associated NTR bit determines if the event bit should be set to 1, as described below:

- If the NTR bit is equal to 1, the event bit should be set.
- If the NTR bit is equal to 0, the event bit should not be set.

The positive conversion register is readable and writable, and its reading will not clear any value.

- **Event register**

This part indicates whether the event occurs after the last reading, and whether the content of the condition register is saved. It only represents the event passed by the transfer register and can only be changed by the instrument, read by the user, and cleared after reading. The value of this part is equal to the value of whole register generally.

- **Enable register**

This part determines whether the associated event bit acts on the final data sum. The data bit of each enable part has a And relation with the associated enable bit. The logical operation result of this part has a OR relation with the data sum bit.

- Enable bit=0: the associated event bit does not act on the data sum.
- Enable bit=1: the associated event bit acts on the data sum.

This part is read-write, and no value will be cleared after reading.

- **Data bit sum**

The data sum bit of each register consists of event and enable parts. The result gets into the condition part of the high level register. The instrument automatically generates data sum bit for each register so that events can cause different levels of service requests.

2.1.5.3 Status Register Description

The following describes the status registers in turn, as shown below:

1) Status byte (STB) and service request enable register (SRE)

The IEEE488.2 defines the status byte (STB) that reflects the rough instrument status by collecting information from the low level registers. The bit6 is equal to the data sum of other status byte bits. The result after comparing the status byte with the condition part of the SCPI register can be assumed to be the highest level in the SCPI level. The common command “*STB?” or the serial query can read the status byte value.

The status byte is connected with the service request enable register (SRE). Each data bit of the status byte corresponds to one bit in the SRE. The SRE bit6 is ignored. If one data bit in the SRE is set and the

associated STB bit changes to 1 from 0, a service request (SRQ) will be generated. The common command “*SRE” is used to set the SRE, and the common command “*SRE? ” is used to read the SRE. The status byte is described in the following Table 2.6 Description of the Status Byte:

Table 2.6 Description of Status Bytes

Data Bit	Meaning
0..1	Not used.
2	The error queue is non-null Set the bit if a new error is inserted into the error queue. If the associated SRE bit enables the bit and a a new error is generated in the error queue, a service request will be generated to identify the error and query the error information. This method effectively reduces the error in remote control.
3	Data sum bit of the status query register The bit can be set if the event bit of the status query register and the associated enable bit are set to 1. The bit represents a queriable status of the instrument, and the specific status information of the instrument can be obtained by querying the status query register of the status register.
4	MAV bit (message available) Set the bit if the output queue information is readable. Use the bit when the controller queries the instrument information.
5	ESB bit Data sum bit of the event status register. The bit can be set if one bit of the event status register is set and the enable event enables the corresponding bit in the register. If the position bit is 1, it means that the instrument has a severe error, and the specific error information can be obtained by querying the event status register.
6	MSS bit (master status summary bit) Set the bit if the instrument triggers the service request.
7	Data sum bit of the operation status register The bit can be set if the event bit of the operation status register and the corresponding enable bit are set to 1. This bit indicates that the instrument executes an operation, and the specific operation type can be obtained by querying the operation status register.

2) Event status register (ESR) and event status enable register (ESE)

For definition of ESR, refer to IEEE488.2. The event status register (ESR) can be read through the command “*ESR?”. The ESE is an enable part of the SCPI register. If one position is set to 1 and one data bit in the responsive ESR changes to 1 from 0, the ESB bit of the STB will be set to 1. Set and read the ESE through the command “*ESE” and the command “*ESE?” respectively.

Table 2.7 Description of Event Status Bytes

Data Bit	Meaning
0	Operation completed The bit can be set when the the previous commands have been executed and the command *OPC has been received.
1	Not used.
2	Query error This bit is set if the controller reads the instrument data without sending a query command or sends a new command without reading the query data. It means that a wrong query is generated and the query can’t be executed.
3	Instrument error Set the bit if an instrument error is generated. Range of error code: -300~-399, or positive error code. For details of specific error information, query relevant information in the error queue.
4	Execution error

Data Bit	Meaning
	Set the bit if a command with correct syntax is received but can't be executed. In addition, an error with the code within the range of -200 ~ -300 is generated in the error queue.
5	Command error Set the bit if the received command has a syntax error. Range of error code: -100~-200; query the relevant information in error queue for specific error information.
6	User request Set the bit if the instrument is switched to the manual control mode.
7	Power on Set the bit when the instrument is powered on.

3) STAT: QUES register

The register includes the status of the instrument not conforming to requirements of the specification. The value of this register can be queried via the command "STAT:QUES:COND" or "STAT:QUES:EVEN". The description of the status register is as shown in the following Table 2.9.

Table 2.8 STAT: QUES Register Description

Data Bit	Meaning
0-2	Not used.
3	If the local power setting is wrong, set this bit.
4	If the time base is not hot, set this bit.
5	If the frequency of local oscillator or the reference frequency of any active channel is wrong, set this bit.
6	Not used.
7	If the local modulation setting is wrong, set this bit.
8	If the instrument is uncalibrated (the "Uncalibrated" prompt message is displayed on the interface), set this bit.
9	In case of self-test error, set this bit.
10-14	Unused.
15	This bit is always 0.

NOTE

Query register

STAT: QUES register collects information about all low-level sub-registers (for example, bit2 in it collects all time-related information). Because each channel corresponds to a separate sub-register, it's required to check the specific error source in the sub-register corresponding to the channel if a status bit of the QUES register indicates an error. The status of the sub-register state to be queried belongs to the currently selected channel by default.

4) STAT: QUES: FREQ register

The information about local oscillator and reference frequency is stored in this register. Each active channel corresponds to a separate frequency register. The value of this register can be read via the command "STATus:QUEStionable:FREQuency:CONDition?" or "STATus:QUEStionable: FREQuency [:EVENt]?".

The description of the status register is as shown in the following Table 2.13.

Table 2.9 STAT: QUES: FREQ Register Description

Data Bit	Meaning
0	OVEN COLD If the reference oscillator does not reach its operating temperature, set this bit. At the same time, the user interface shows the prompt message “OCXO”.
1	Local oscillator UNL If the local oscillator is unlocked, set this bit. At the same time, the user interface shows the prompt message “LO UNL”.
2..7	Not used.
8	External reference If an external reference oscillator is set, but the available external reference signal is not actually connected, set this bit. At this time, the frequency synthesizer is unlocked and the frequency accuracy is low.
9..14	Not used.
15	**This bit is always 0.**

5) STAT: QUES: POW register

This register contains information about power overload when the instrument is operated. Each active channel corresponds to a separate power register. The value of this register can be read via the command “STATus:QUEStionable: POWer:CONDition?” or “STATus:QUEStionable: POWer [:EVENt]?”. The description of the status register is as shown in the following Table 2.16.

Table 2.10 STAT: QUES: POW Register Description

Data Bit	Meaning
0	Overload If the power of RF input signal is overloaded, set this bit. This will cause signal distortion without damaging the instrument, and at the same time, the user interface will show the prompt message “RF overload”.
3	Input overload If the signal power of RF input connector is out of the rated range, set this bit. At the same time, the user interface shows the prompt message “Input overload”. The RF input and the input mixer are separated to protect the instrument. In case of repeated measurement, reduce the power level of RF input connector.
4..14	Not used.
15	**This bit is always 0.**

2.1.5.4 Application of status reporting system

The status reporting system is used to monitor the status of one or more instruments in the test system. To correctly realize the function of the status reporting system, the controller in the test system must receive and evaluate the information of all instruments. The standard methods applied include:

1) Query the service request (SRQ) initiated by the instrument;
2) Perform serial poll of all instruments in the bus system, which is initiated by the controller in the system in order to find out the initiator and reason of the service request.
3) Perform parallel query of all instruments;
4) Query of the status of specific instrument by remote control command;
5) Query of the error queue.

1) Service request

In some cases, the instrument will send a service request (SRQ) to the controller to obtain the

controller's service, and then the controller will initiate an interruption to enter the corresponding interruption handler. As shown in Figure 2.4, an SRQ is usually initiated by one or more status bytes and the 2nd, 3rd, 4th, 5th or 7th bits of the associated enable register (SRE). These data bits are composed of the advanced register, error queue, or output buffer area further. In order to use all the service requests as much as possible, all data bits of the enable registers SRE and ESE shall be set to 1.

Example: When the command "*OPC" is used after sweep, the SRQ signal will be generated.

a) Recall the write function InstrWrite and write command "*ESE 1", and set the ESE bit0 (operation is completed).
b) Recall the write function InstrWrite and write command "*ESE 32", and set the SRE bit5 (ESB).
c) Recall the write function InstrWrite and write command "*INIT;*OPC", and generate the SRQ signal after operation is completed.

The instrument will generate an SRQ after settings are completed.

The SRQ can only be initiated by the instrument, and the controller program shall allow sending the service request to it when the instrument has an error, and it will be processed by a special interruption handler. For specific routine, please refer to Section 4.2.1.1. Service Request.

2) Series query

Similar to the command *STB, the series query can be used to query the status byte of the instrument. The series query adopts the interface message mode, therefore, the query is fast. The IEEE 488.2 defines the specific series query method. This method is mainly used to quickly obtain the status of one or more instruments connected with the controller in the test system.

3) Parallel query

The controller can send an information bit to the USB cable through a command, and query 8 instruments in the test system. The data configured on the USB cable of the instrument is logic "0" or "1". Except that the SRE register determines the conditions generating the SRQ, perform parallel query to check AND operation of data bit of the enable register (PPE) and the STB register. The result will be made as the response result and sent to the parallel query controller through OR operation and bit reverse, and it can also be obtained through the command *IST.

During parallel query, set the instrument to the parallel query status through the command PPC firstly, and this command will allocate a USB cable to the instrument and determine whether the data bit is reversed in response. Use the PPE register during execution of the parallel query. The parallel query is mainly used to quickly position which instrument sends the service request by the controller. Accordingly, the same value shall be set for the registers SRE and PPE.

4) Instrument status query

Query each part of the status register through following two commands:

- Command "*IDN?" for querying the high-level register;
- Status system command for querying the SCPI register (e.g. STATus:QUEStionable...).

The returned value of the queried register is usually in decimal format, which is used by the controller program for detection. In order to obtain a more detailed description of the SRQ cause, the parallel query will be carried out after the SRQ generally.

Description of response data bit

The STB and ESR registers include 8 bits, and the SCPI register includes 16 bits. The returned value of the queried status register is in decimal format. The decimal value is equal to the sum of the data bits and their own weights after operation.

The relationship between the data bit and its weight is as shown in the following Figure:

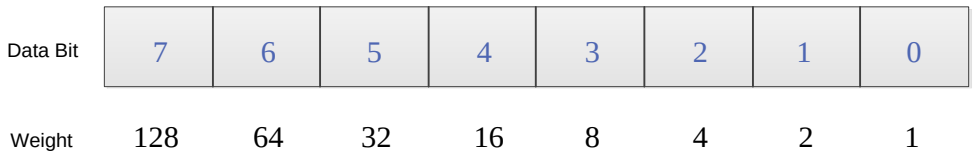

Fig. 2.5 Relationship between Data Bits and their Weights

5) Error queue

Each error status of the instrument corresponds to an entry in the error queue, and the entry contains the specific error message text which can be viewed through the error log or queried via the programmable command “SYSTem:ERRor[:NEXT]?”. If there is no error in the error queue, the query will return 0, “No Error”.

As the obtained error cause description is more accurate than the status register, the error queue shall be queried in the controller service request handler. The error queue shall be frequently queried especially during the controller program test stage, so as to clarify the error command record sent to the instrument by the controller.

2.1.5.5 Reset status reporting system

The following list shows the commands and events for resetting the status report system. Except for the commands *RST and SYSTem:PRESet, other commands do not change the instrument function settings. Similarly, the DCL will not change setting status of the instrument. The specific description is as shown in the following table:

Table 2.11 Resetting Status Report System

Event / Function	Power On/Off (Power-on status cleared)		DCL, SDC (Instrument cleared, selected instrument cleared)	*RST or SYSTem: PRESet	STATus: PRESet	*CLS
	0	1				
Clear STB and ESR	—	Yes	—	—	—	Yes
Clearing SRE, ESE	—	Yes	—	—	—	—
Clearing PPE	—	Yes	—	—	—	—
Clearing the event part of the register	—	Yes	—	—	—	Yes

2.1 Remote control basis

Event / Function	Power On/Off (Power-on status cleared)		DCL, SDC (Instrument cleared, selected instrument cleared)	*RST or SYSTem: PRESet	STATus: PRESet	*CLS
	0	1				
Clear the enable part in the operation and inquiry registers. Filling 1 in the enable part of other registers.	—	Yes	—	—	Yes	—
The positive transfer part is filled with 1. Clear the negative transfer part.	—	Yes	—	—	Yes	—
Clearing the error queue	Yes	Yes	—	—	—	Yes
Clearing the output buffer area	Yes	Yes	Yes	—	—	—
Clearing the command processing and input buffer area	Yes	Yes	Yes	—	—	—

2.1.6 Programming considerations

1) Please initialize the instrument status before changing the settings

When setting the instrument through remote control, it is necessary to initialize the instrument status (e.g. send “*RST”) and then set the desired status.

2) Command sequence

In general, the set and query commands should be sent separately; otherwise the returned value of the query command will change according to the current instrument operation sequence.

3) Failure response

The service request can only be initiated by the instrument itself. The controller program in the test system should instruct the instrument to initiate a service request when an error occurs, and then enter the corresponding interrupt service routine for processing.

4) Error queue

Each time the controller program processes a service request, the error queue rather than the status register of the instrument should be queried to obtain a more accurate error reason. The error queue should be frequently queried to obtain the wrong command sent by the controller to the instrument especially during testing of the controller program.

2.2 Remote interface and its configuration

- LAN 27
- GPIB 28

2.2.1 LAN

The programmed control system of LAN (Local Area Network) uses SICL-LAN to control the 1465 series signal generator.

CAUTION

Use of USB main control port connector on front panel

The Type-A connector on the front panel is for USB master control port. In the 1465 series signal generator, this port is used to be connected to the flash disk of USB 2.0 interface to realize the instrument's resident software upgrade, and can also be connected to the USB keyboard and mouse to control the signal generator. It is not possible to remotely control the instrument via this port.

- Connection Establishment 27
- Interface Configuration 27

2.2.1.1 Connection

Connect the 1465 series signal generator and the external controller (computer) to LAN with network cable, as shown in Fig. 2.6:

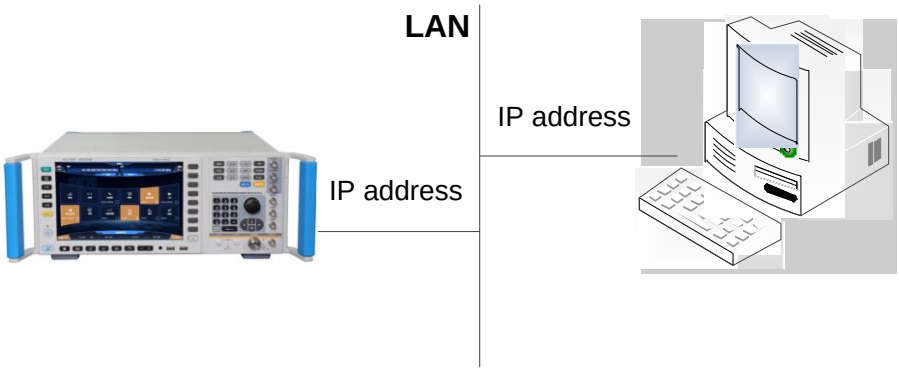

Fig. 2.6 LAN Interface Connection

2.2.1.2 Interface configuration

When the signal generator is remotely controlled via LAN, the physical network connection shall be guaranteed to be smooth. Because DHCP, domain name access and wide area network connection are not supported, the network program control setting of the signal generator is relatively simple.

Press 【System】→[LAN Interface] to enter the interface shown in Fig. 2.7 and set the "IP address", "Subnet mask" and "Default gateway" to the subnet of main controller.

2.2 Remote interface and its configuration

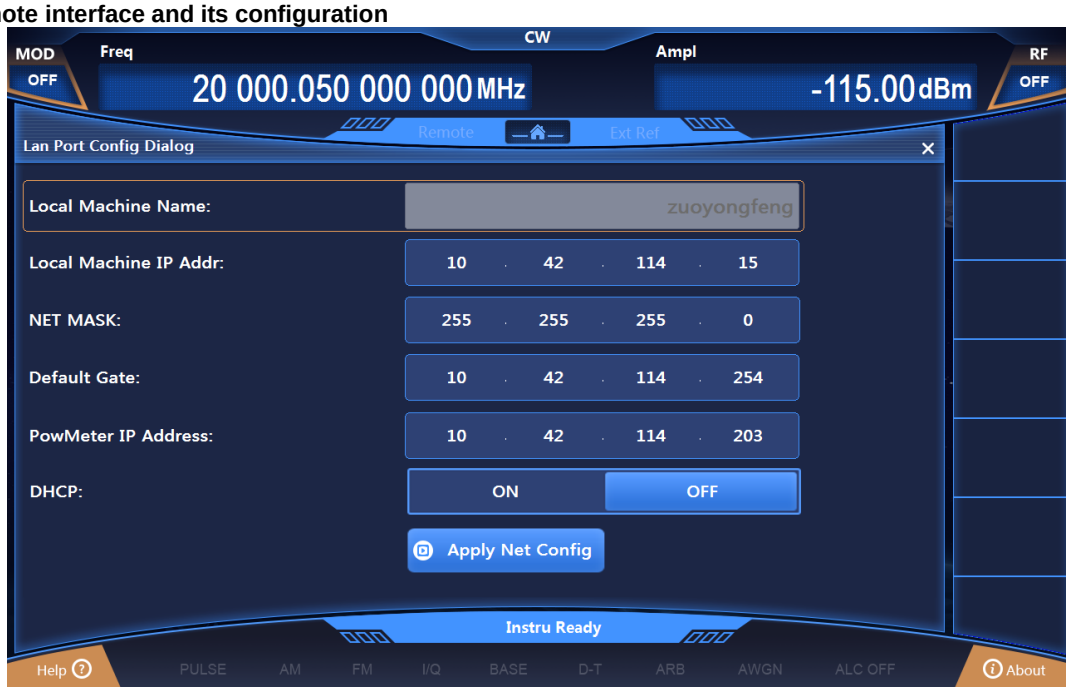

Figure 2.7 LAN Interface Settings

CAUTION

Ensure that the signal generator is physically connected properly by 10Base-T LAN or 100Base-T LAN cable

As the signal generator only supports the construction of a single LAN control system and the setting of static IP address, rather than DHCP and the access to the host via DNS and domain name server, users do not need to modify the subnet mask. In the instrument, it is fixedly set to: 255.255.255.0。

2.2.2 GPIB

- Connection Establishment 28
- Interface Configuration 29

2.2.2.1 Connection

Connect the 1465 series signal generator to the external controller (computer) with GPIB cable, as shown in Fig. 2.8:

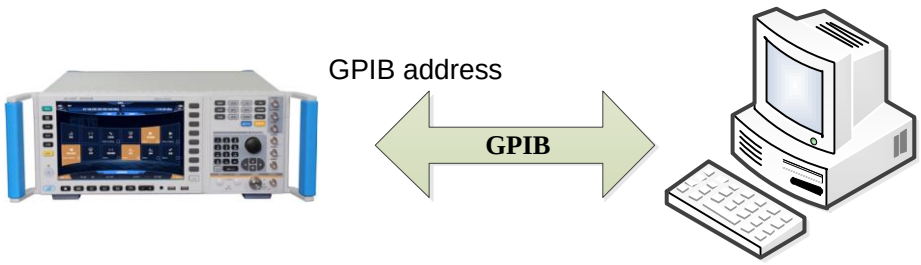

Fig. 2.8 GPIB Interface Connection

2.2.2.2 Interface configuration

The user may need to modify the GPIB address when building a system with a signal generator. The GPIB address of the machine is 19 by default. The method to change the GPIB address is as follows: Press 【System】→[GPIB Interface] to enter the interface shown in Fig. 2.9, so as to make changes in the local GPIB address input box by using the numeric keys on the front panel.

Figure 2.9 GPIB Interface Settings

2.3 I/O library

- General 29
- Installation and Configuration 30

2.3.1 Overview of I/O library

As a library of software programs pre-written for the instrument, the I/O library is called an instrument driver. It is considered as the intermediate layer of the software between the computer and the instrument hardware equipment, composed of function library, utility program and tool kit, and used as a software code module set that corresponds to a planned operation, e.g. configuring, reading from, writing to or triggering the instrument. It resides in the computer as the bridge and link between the computer and the instrument, and provides an easily programmed high-level modular library so that the user no longer needs to learn complex low-level programming protocols specific to an instrument. The instrument driver is the key to rapid development and test of measurement applications.

From the aspect of function, a general instrument driver usually consists of a functional body, an interactive developer interface, a program developer interface, a subprogram interface and an I/O interface as shown in Fig. 2.12.

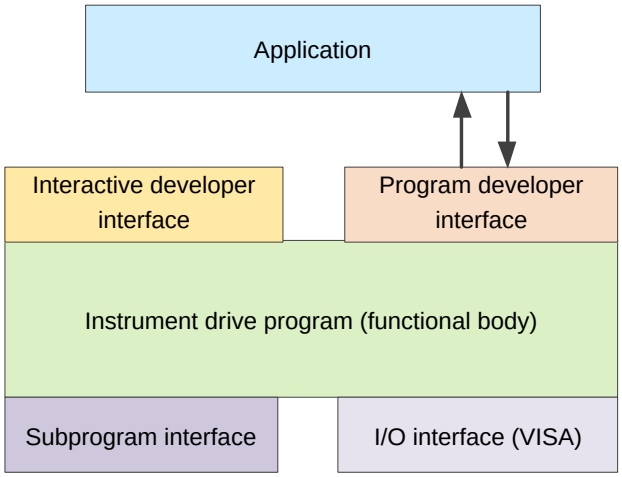

Figure 2.12 Instrument Driver Structure Model

The detailed description is given as follows:

1) Functional body. It is the main functional part of the instrument driver and can be understood as the framework program of the instrument driver. 。
2) Interactive developer interface. For user's convenience, a graphical interactive developer interface is generally provided in the application development environment that supports the development of the instrument driver. For example, the function panel in Labwindows/CVI is an interactive developer interface. In the function panel, each parameter of the instrument driver function is represented by a graphical control.
3) Program developer interface. It is a software interface for recalling of the instrument driver function by the application, such as the dynamic link library file .dll of the instrument driver of the Windows system.
4) I/O interface. It is used to complete the actual communication between the instrument driver and the instrument. The special bus I/O software can be used, such as GPIB and RS-232; the general standard I/O software across multiple buses can also be used, such as VISA I/O。
5) Subroutine interface. It is a software interface for the instrument driver to access other support libraries including database and FFT function. When the instrument driver needs to recall other software modules, operating systems, program code libraries and analysis function libraries to complete its task, the subprogram interface will be used.

2.3.2 Installation and configuration of I/O library

With the development of the test field application from the traditional instrument to the virtual instrument, the instrument driver has experienced different development processes in order to solve the instrument interchangeability and test program reusability of the automatic test system. Currently, the IVI (Interchangeable Virtual Instruments) driver is popularly applied. Based on the IVI specification, a new instrument programming interface is defined, the class driver and VPP architecture are inserted onto the VISA so that the test application is completely independent of the instrument hardware, and unique instrument simulation, range detection and status buffer functions are added, which improves the system operation efficiency and truly achieves the instrument interchange.

IVI driver is divided into two types: IVI-C and IVI-COM. IVI-COM is based on Microsoft Component

Object Model (COM) and adopts COM API; IVI-C is based on ANSI C and adopts C API. Both drivers are designed according to the instrument class defined in the IVI specification, with the same application development environments including Visual Studio, Visual Basic, Agilent VEE, LabVIEW and CVI/LabWindows.

Currently, it is necessary to provide two types of drivers in order to meet the demands of different users in different development environments. The IVI driver of the signal generator uses Nimbus Driver Studio to produce IVI-COM and IVI-C drivers as well as program installation package. For specific installation and configuration, please refer to documents accompanied with the control card and I/O Library of your choice.

The IVI drivers installed are divided into: IVI intrinsic function group and instrument class function group (basic function group and extended function group). For details about functional classification, functions and attributes, please refer to the accompanied help document of the driver.

NOTE

Configuration of ports and installation of IO library

When a computer is to be used to control the signal generator, please confirm that the necessary ports and I/O library have been properly installed and configured.

NOTE

Use of I/O library

Once installed, the attached IVI-COM/C driver installation package will automatically install the driver function panel, help documents, and sample programs of the driver functions to facilitate the users to develop and integrate the program control functions.

3 SCPI

- Command instruction 33
- Common command 33
- Instrument subsystem command 34

3.1 Command instruction

This chapter provides detailed reference information of commands for achieving remote control, which includes:

- Complete syntax format and parameter list;
- Syntax diagram for SCPI non-standard command;
- Specific function description and related command instruction;
- Supported command format (setting or query);
- Parameter description including: data type, data range and default (unit);
- Key path;
- Model of similar instruments to which the commands are compatible, or the indication that the current command is only applicable to 1465 model.
- Other descriptions.

In the sections about common commands and instrument subsystem commands, the commands in sequence are listed first for convenient query and use.

3.2 Common command

Common commands are used for the control of instrument status register, status report, synchronization, data storage and other general functions. They are applicable to different instruments in respect of usage and functions. All common commands can be identified by the first“*”in the command characters. These common commands are defined in detail in IEEE488.2.

The followings are the explanation and instruction of the common commands in IEEE488.2.

- *IDN? 34
- *RCL 34
- *RST 34
- *SAV 34
- *TRG 34

NOTE

Use of command:

Unless otherwise specified, the command may be used for setting and query.

The condition that one command is only used for setting or querying, or initiating an event, if any, will be indicated separately in the command instruction.

*IDN?

Function: This command is used for returning to instrument identification.
Return value: <ID>: "manufacturer, <instrument model>, <serial number>, <firmware version number>"
Example: CETC41,1465,2017008,1.0.11
Note: for query only.

*RCL <Value>

Function: This command is used for recalling instrument status from the designated internal register of the signal generator.
Parameter: range [0, 99].
Note: for setting only.

*RST

Function: This command is used for accomplishing reset function of the signal generator.
Note: for setting only.

*SAV <Value>

Function: This command is used for storing the current status of the signal generator in the designated internal register.
Parameter: range [0, 99].
Note: for setting only.

*TRG

Function: This command is used for selecting the bus in the signal generator as the trigger source.
Note: for setting only.

3.3 Instrument subsystem command

This chapter details the subsystem commands in 1465 series signal generator.

- OUTPut 35
- FREQuency 36
- POWer 41
- PHASe 47
- LIST 47
- LFOutput 52
- SWEep 56
- PULM 59
- AMPLitude Modulation 65
- FREQuency Modulation 72
- PHASe Modulation 76
- Digital Modulation.....81
- MEMory 107

- ROSCillator 109
- SYSTem 109

3.3.1 OUTPut subsystem

OUTPut subsystem commands are used for controlling the switch of RF output signal.

The following commands are used for setting operation modes:

- OUTPut:BLANking[:STATe] ····· 35
- OUTPut[:STATe] ····· 35
- OUTPut:MODulation[:STATe] ····· 35

:OUTPut:BLANking[:STATe] <State>

Function: This command sets the state of RF Blank. When RF Blank is turned on, if the signal generator is in spot frequency state, RF output signal will be turned off during frequency switching; if the signal generator is in sweep state, RF output signal will be turned off during frequency band switching and RTC.

Setting format: OUTPut:BLANking[:STATe] ON|OFF|1|0

Query format: OUTPut:BLANking[:STATe]?

Parameter:

<State> Boolean data is as follows:

ON | 1: Blank is turned on

OFF | 0: Blank is turned off.

Example: OUTPut:BLANking 1

The Preceding example sets the signal generator Blank to ON state.

Reset status: 0

:OUTPut[:STATe] <State>

Function: This command enables RF output of the signal generator.

Setting format: :OUTPut[:STATe] ON|OFF|1|0

Query format: :OUTPut[:STATe]?

Parameter:

<State> Boolean data is as follows:

ON | 1: RF output

OFF | 0: RF OFF

Example: OUTPut: 1

The Preceding example sets RF output of the signal generator..

Reset status: 0

Key Entry 【RF ON/OFF】

:OUTPut:MODulation[:STATe] <State>

Function: This command is used for setting the state of the modulation main switch.

Setting format: :OUTPut:MODulation[:STATe] ON|OFF|1|0

Query format: :OUTPut:MODulation[:STATe]?

Parameter:

<State> Boolean data is as follows:
ON | 1: modulation ON,
OFF | 0: modulation OFF.

Example: :OUTPut:MODulation 1 Set the state of the modulation main switch as ON.

Reset status: 0

Key Entry: 【Modulation ON/OFF】

3.3.2 FREQuency subsystem

FREQuency subsystem commands are used for controlling the general functions of RF output signal frequency.

The following commands are used for setting operation modes:

- [:SOURce]:FREQuency[:CW|FIXed] 36
- [:SOURce]:FREQuency:MODE 37
- [:SOURce]:FREQuency:MULTiplier..... 37
- [:SOURce]:FREQuency:OFFSet 37
- [:SOURce]:FREQuency:REFerence 38
- [:SOURce]:FREQuency:REFerence:STATe 38
- [:SOURce]:FREQuency:STEP 39
- [:SOURce]:FREQuency:STARt 39
- [:SOURce]:FREQuency:STOP 40
- [:SOURce]:FREQuency[:CW|FIXed]:AUTO 40

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

Function: This command is used for setting the output frequency of the signal generator in CW mode. For setting other frequency modes, please refer to the command ":FREQuency:MODE".

Setting format: [:SOURce]:FREQuency[:CW|FIXed] <val>

Query format: [:SOURce]:FREQuency[:CW|FIXed]?

Parameter:

<Frequency> Output frequency in CW mode.

Model Range

1465A [100kHz~3GHz]
1465B [100kHz~6GHz]
1465C [100kHz~10GHz]
1465D [100kHz~20GHz]
1465E [100kHz~40GHz]
1465H [100kHz~50GHz]
1465L [100kHz~67GHz]

Example: [:SOURce]:FREQuency[:CW|FIXed] 10GHz Set the frequency of the signal generator to 10 GHz.

Reset status: Freq Start + (Freq Stop - Freq Start)/2
Key Entry: 【Frequency】—>[CW]

[:SOURce]:FREQuency:MODE <Mode>

Function: This command sets the frequency mode of the signal generator.
Setting format: [:SOURce]:FREQuency:MODE FIXed|CW|SWEep|LIST
Query format: [:SOURce]:FREQuency:MODE?
Parameter:
<Mode> Discrete data, Current Sweep Type to be configured. The values are as follows:

FIXed|CW	The setting of these two discrete parameters is the same in the signal generator in meaning, that is, the controlled signal generator outputs CW (spot frequency) signal, which will terminate the frequency sweep signal currently output by the instrument. For setting spot frequency, please refer to the commands ":FREQuency[:CW]" and ":FREQuency[:FIXed]".
SWEep	This parameter sets current frequency mode as sweep mode. Sweep mode in the software, whether step sweep or ramp sweep, is determined by: "SWEep:GENeration". The default is step sweep.
LIST	Current Sweep Type is set to List mode. If the current list is empty, the signal generator will prompt the list is empty. The signal generator can start sweep only when there is at least one sweep point in the list.

Example: :FREQuency:MODE LIST Set the signal generator to List Sweep mode.
Reset status: CW

[:SOURce]:FREQuency:MULTiplier <FreqMult>

Function: This command sets Freq Mul for signal source. When set Freq Mul is greater than 1, a multiplier indicator "*" is displayed in frequency display area. At this time, displayed frequency value= RF output frequency value ×multiplier. However, the actual frequency output is still the frequency with no multiplier added. When Freq Mul is set to 1, the indicator disappears.
Setting format: [:SOURce]:FREQuency:MULTiplier <val>
Query format: [:SOURce]:FREQuency:MULTiplier?
Parameter:
<FreqMult> Freq Mul.
Range: 1[1，36].
Example: :FREQuency: MULTiplier 8 Freq Mul of the signal generator is 8.
Reset status: 1
Key Entry: 【Frequency】—>[Basic Config]->[Freq Mul]

[:SOURce]:FREQuency:OFFSet <FreqOffs>

Function: When Frequency Offset is not set to zero, an offset indicator "*" is displayed in frequency display area, and the display value becomes the frequency value plus offset.

At this time, the displayed frequency value= RF output frequency ×multiplier+ frequency offset. However, the actual frequency output is still the frequency with no multiplier and frequency offset added. When Freq Offset is set to zero, the indicator disappears.

Setting format: [:SOURce]:FREQuency:OFFSet <val>

Query format: [:SOURce]:FREQuency:OFFSet?

Parameter:

<FreqOffs> Freq Offset

Range: 0Hz[-325GHz，+325GHz].

Example: :FREQuency:OFFSetr 10GHz Freq Offset of the signal generator is 10 MHz.

Reset status: 0Hz

Key Entry: 【Frequency】—>[Basic Config]->[Freq Offset]

[:SOURce]:FREQuency:REFerence <FreqRef>

Function: This command is used for setting the frequency reference function. The set value can function well when the frequency reference switch is on. See the command ":FREQuency:REFerence:STATe". Frequency reference indicator"*" is displayed in the frequency display area. In this case, the frequency reference value will be subtracted from the set CW output signal. For example: The current CW output frequency is 1 GHz. If Freq Ref is set to 1 GHz, the CW output frequency displayed at this time will be based on 0 Hz frequency reference, so frequency display area will display 1 MHz; if CW frequency is set to 1 MHz, frequency display area will display 1 MHz and the actual output frequency is 1.001 GHz.

Setting format: [:SOURce]:FREQuency:REFerence <val>

Query format: [:SOURce]:FREQuency:REFerence?

Parameter:

<FreqRef> Freq Ref.

Model Range

1465A [0Hz~3GHz]
1465B [0Hz~6GHz]
1465C [0Hz~10GHz]
1465D [0Hz~20GHz]
1465E [0Hz~40GHz]
1465H [0Hz~50GHz]
1465L [0Hz~67GHz]

Example: :FREQuency:REFerence 10GHz Set the relative frequency of the signal generator to 10 GHz.

Reset status: 0Hz

Key Entry: 【Frequency】—>[Base Config]->[Freq Ref]

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

Function: This command sets Freq Ref Switch to ON or OFF state. After the frequency reference

switch is ON, when the CW frequency of the signal generator is changed, the frequency value displayed in the frequency display area is based on this frequency reference. For setting frequency reference, please see the command":FREQuency:REFerence"; when the switch is OFF, the frequency value displayed in the frequency display area is the actual CW frequency of the signal generator.

Setting format: [:SOURce]:FREQuency:REFerence:STATe ON|OFF|1|0

Query format: [:SOURce]:FREQuency:REFerence:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: Freq Ref Switch is turned on,

OFF | 0: Freq Ref Switch is turned off.

Example: :FREQuency:REFerence:STATe 1 Freq Ref of the signal generator is turned on.

Reset status: 0

Key Entry: 【Frequency】—>[Basic Config]->[Freq Ref Switch]

[:SOURce]:FREQuency:STEP <FreqStep>

Function: Set the step value of the frequency. After setting this value, for setting CW frequency of the signal generator, please see the command ":FREQuency:CW" or ":FREQuency:FIXed". When changing the frequency by UP|DOWN, the frequency variation will change based on the current set step value.

Setting format: [:SOURce]:FREQuency:STEP <val>

Query format: [:SOURce]:FREQuency:STEP?

Parameter:

<FreqStep> Freq Step.

Model Range

1465A [0Hz~2.9999GHz]

1465B [0Hz~5.9999GHz]

1465C [0Hz~9.9999GHz]

1465D [0Hz~19.9999GHz]

1465E [0Hz~39.9999GHz]

1465H [0Hz~49.9999GHz]

1465L [0Hz~66.9999GHz]

Example: :FREQuency: STEP 1MHz Freq Step of the signal generator is 1 MHz.

Reset status: 100MHz

Key Entry: 【Frequency】—>[Basic Config]->[Freq Step]

[:SOURce]:FREQuency:STARt <StartFreq>

Function: This command sets Freq Start of Step Sweep. See the command ":FREQuency:STOP".

Setting format: [:SOURce]:FREQuency:STARt <val>

Query format: [:SOURce]:FREQuency:STARt?

Parameter:

<StartFreq> Freq Start of sweep.

Model Range

1465A [100kHz~3GHz]
1465B [100kHz~6GHz]
1465C [100kHz~10GHz]
1465D [100kHz~20GHz]
1465E [100kHz~40GHz]
1465H [100kHz~50GHz]
1465L [100kHz~67GHz]

Example: :FREQuency:STARt 1MHz Freq Start of Step/Analog Sweep of the signal generator is 1 MHz.

Reset status: 100kHz

Key Entry: 【Sweep】—>[Step Sweep]/[List Sweep]-> [Freq Start]

[:SOURce]:FREQuency:STOP <StopFreq>

Function: This command sets Freq Stop of Step Sweep. See the command "FREQuency:STARt".

Setting format: [:SOURce]:FREQuency:STOP <val>

Query format: [:SOURce]:FREQuency:STOP?

Parameter:

<StopFreq> Freq Stop of sweep.

Model Range

1465A [100kHz~3GHz]
1465B [100kHz~6GHz]
1465C [100kHz~10GHz]
1465D [100kHz~20GHz]
1465E [100kHz~40GHz]
1465H [100kHz~50GHz]
1465L [100kHz~67GHz]

Example: :FREQuency:STOP 100MHz Freq Stop of Step/Analog Sweep of the signal generator is 100 MHz.

Reset status: It is related to the model. For 67GHz model, it is 67GHz.

Key Entry: 【Sweep】—>[Step Sweep]/[List Sweep]-> [Freq Stop]

[:SOURce]:FREQuency[:CW|FIXed]:AUTO <State>

Function: This command is used for set frequency follow-up. When this function is enabled, the output CW frequency of the signal generator will change according to the editing of corresponding frequency value in the internal user flatness list. See the commands ":CORRection:FLATness:PAIR", ":CORRection:FLATness:FILL:STARt" and ":CORRection:FLATness:FILL:STOP".

Setting format: [:SOURce]:FREQuency[:CW|FIXed]:AUTO ON|OFF|1|0

Query format: [:SOURce]:FREQuency[:CW|FIXed]:AUTO?

Parameter:

<State> Boolean data is as follows:

ON | 1: Frequency follow-up ON,

OFF | 0: Frequency follow-up OFF.

Example: [:SOURce]:FREQuency[:CW|FIXed]:AUTO 1 Frequency follow-up of the signal generator is ON.

Reset status: 0

Key Entry: 【Calibration】—>[Power Accuracy Calibration] —>[Frequency Follow-up]

3.3.3 POWer subsystem

POWer subsystem commands are used for controlling the general functions of RF output signal power level.

The following commands are used for setting operation modes:

- [:SOURce]:POWer:ALC:LEVel 41
- [:SOURce]:POWer:ALC:SEARch 42
- [:SOURce]:POWer:ALC:SOURce 42
- [:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling 42
- [:SOURce]:POWer:ALC[:STATe]..... 43
- [:SOURce]:POWer:ATTenuation 43
- [:SOURce]:POWer:ATTenuation:AUTO 43
- [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] 44
- [:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet 44
- [:SOURce]:POWer:REFerence..... 44
- [:SOURce]:POWer:REFerence:STATe 45
- [:SOURce]:POWer:STEP 45
- [:SOURce]:POWer:ALC:BANDwidth|BWIDth 46
- [:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTo 46

[:SOURce]:POWer:ALC:LEVel <AlcLevel>

Function: This command sets ALC level when Attenuation Style is set to Manual mode. For selection of manual/automatic mode of attenuator, see the command ":POWer:ATTenuation:AUTO".

Setting format: [:SOURce]:POWer:ALC:LEVel <value>

Query format: [:SOURce]:POWer:ALC:LEVel?

Parameter:

<AlcLevel> ALC level.

Range: 0dBm[-20dBm，+30dBm].

Example: :POWer:ALC:LEVel 5dBm ALC level is 5 dBm.

Reset status: 0dBm

Key Entry: 【Amplitude】—>[Atten Config]—>[ALC]

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

Function: This command activates or deactivates the internal amplitude auto search of the signal generator when ALC loop is turned on. Amplitude search will enable amplitude to stabilize the signal generator on the output amplitude selected by users and keep the internal modulator in drive state when ALC Loop is turned off. For ALC loop state, see the command ":POWer:ALC[:STATe]".

Setting format: [:SOURce]:POWer:ALC:SEARch ON|OFF|1|0|ONCE

Query format: [:SOURce]:POWer:ALC:SEARch?

Parameter:

<Mode> Discrete data. Auto amplitude search state is as follows:

OFF | 0: This command terminates Auto mode, and Search Style is Manual.

ON | 1: Amplitude is automatically searched with the change of RF output amplitude or frequency. Search Style is then in Auto mode

ONCE Do Search once at current RF output frequency.

Example: :POWer:ALC:SEARch 1 Search Style is Auto.

Reset status: Manual

Key Entry: 【Amplitude】—> [ALC Loop Config]—>[Search Style Auto Manual]/[Search Out]

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

Function: This command allows users to select ALC Level Control for the signal generator as appropriate, including INT, EXT and Source Module.

Setting format: [:SOURce]:POWer:ALC:SOURce INTernal|DIODe

Query format: [:SOURce]:POWer:ALC:SOURce?

Parameter:

<State > Discrete data. Level Control is as follows:

INTernal ： Level Control is INT,

DIODe external diode detector level control,

Example: :POWer:ALC:SEARch:REFerence INT Level Control is INT.

Reset status: Internal

Key Entry: 【Amplitude】—> [Level Control]->[Level Control]

[:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling <CouplingValue>

Function: This command sets EXT Detector Couple. When Level Control is EXT diode detector, this command enables you to set the coupling value for external level control. For level control, see the command ":POWer:ALC:SOURce".

Setting format: [:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling <value>

Query format: [:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling?

Parameter:

<CouplingValue> EXT Detector Couple

Range: 16dBm[-90dBm，+90dBm].

Example: :POWer:ALC:SOURce:EXTernal:COUPling 16dBm Ext Detector Couple is 16dBm.

Reset status: 16.00dBm

Key Entry: 【Amplitude】—> [Level Control]->[Ext Detector Couple]

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

Function: This command enables you to turn ALC loop ON or OFF. ALC Loop is mainly to correct amplitude drift and to make the signal generator output amplitude level not change with time and temperature.

Setting format: [:SOURce]:POWer:ALC[:STATe] ON|OFF|1|0

Query format: [:SOURce]:POWer:ALC[:STATe]?

Parameter:

<State> Boolean data is as follows:

ON | 1: ALC Loop is turned off,

OFF | 0: ACL Loop is turned on.

Example: :POWer:ALC 1 This example indicates ALC Loop state is set as ALC-OFF.

Reset status: 1

Key Entry: 【Amplitude】—> [ALC Loop Config]—>[ALC Loop State ALC-OFF ALC-ON]

[:SOURce]:POWer:ATTenuation <Atten>

Function: This command is used for setting power amplitude attenuation value of the mechanical attenuator of the signal generator. The value set by this command will be valid only when the attenuator is set to Manual. For setting auto/manual mode of attenuator, see the command ":POWer:ATTenuation:AUTO".

The minimum attenuation step set by this command is 5 dB, that is, users can only set attenuation to 0 dB, 5 dB, 10 dB, 15 dB by step of 5 dB. After Attenuation is set, the signal generator output amplitude is the current ALC minus the currently set attenuation.

Setting format: [:SOURce]:POWer:ATTenuation <value>

Query format: [:SOURce]:POWer:ATTenuation?

Parameter:

<Atten> Amplitude attenuation.

Range: 115dB[0dB，115dB].

Example: :POWer:ATTenuation 15dB Attenuation is 15 dB.

Reset status: 115dB

Key Entry: 【Amplitude】—>[Atten Config]—>[Attenuation]

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

Function: This command sets the control state of internal programmable step attenuator: Auto or Manual. In Auto mode, the signal generator automatically sets the value of amplitude attenuator based on the current output amplitude; in Manual mode, the current amplitude attenuation of attenuator does not change during the process of changing amplitude output level.

Setting format: [:SOURce]:POWer:ATTenuation:AUTO ON|OFF|1|0

Query format: [:SOURce]:POWer:ATTenuation:AUTO?

Parameter:

<State> Boolean data is as follows:

ON | 1: Attenuation is AUTO,

OFF | 0: Attenuation is Manual.

Example: :POWer:ATTenuation:AUTO 0 Attenuation Style is Manual.

Reset status: 1

Key Entry: 【Amplitude】—>[Attenuation Config]—>[Attenuation Style Auto Manual]

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

Function: This command sets the output amplitude level of the signal generator.

Setting format: [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <value>

Query format: [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]?

Parameter:

<Ampl> Amplitude level value.

Range: -135dBm [-135dBm，+30dBm].

Example: :POWer 0dBm The output amplitude level is 0 dBm.

Reset status: -115dBm

Key Entry: 【Amplitude】

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

Function: The command is the actual output amplitude offset of the signal generator. When the value is non-zero, an offset indicator "*" is displayed in amplitude display area and the displayed amplitude value is the actual amplitude power plus the amplitude offset. Such amplitude offset does not change the actual output amplitude of the signal generator. Only the displayed amplitude value is changed.

Setting format: [:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet <value>

Query format: [:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet?

Parameter:

<PowOffset> Ampl Offset.

Range: 0dB [-100dB，+100dB].

Example: :POWer:OFFS -10dB Ampl Offset is -10 dB.

Reset status: 0dB

Key Entry: 【Amplitude】—>[Basic Config]->[Ampl Offset]

[:SOURce]:POWer:REFerence <PowRef>

Function: Ampl Ref can be set when Ampl Ref Switch is turned on. For the state of power reference switch, see the command ":POWer:REFerence:STATe". When Ampl Ref Switch is turned on, an indicator "*" is displayed in amplitude display area and the displayed amplitude value = actual output amplitude - amplitude reference value.

For example: When current CW output power is 1dBm, if the power reference is set to 1dBm, the displayed CW output power is based on this power reference. Therefore, the power display area shows 0dBm, and the actual output frequency of the signal

generator remains 1dBm.

Setting format: [:SOURce]:POWer:REFerence <value>

Query format: [:SOURce]:POWer:REFerence?

Parameter:

<PowRef> Ampl Ref.

Range: 0dBm [-135dBm，+30dBm].

Example: :POWer:REFerence -10dBm Ampl Ref is -10 dBm.

Reset status: 0dBm

Key Entry: 【Amplitude】—>[Basic Config]->[Ampl Ref]

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

Function: This command sets the state of Ampl Ref Switch. When the power reference switch is ON and the power reference value is not zero, if the power level of the signal generator is changed, the power value displayed in the power display area will be based on this power reference. For setting power reference, see the command ":POWer:REFerence"; when the power reference switch is OFF, the power value displayed in the power display area will be the actual CW output power of the signal generator.

Setting format: [:SOURce]:POWer:REFerence:STATe ON|OFF|1|0

Query format: [:SOURce]:POWer:REFerence:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: Ampl Ref is turned on,

OFF | 0: Ampl Ref is turned off.

Example: :POWer:REFerence:STATe 1 Ampl Ref is turned on.

Reset status: 0

Key Entry: 【Amplitude】—>[Basic Config]->[Ampl Ref Switch ON OFF]

[:SOURce]:POWer:STEP <PowStep>

Function: Set the step value of the power. After setting this value, for setting CW power of the signal generator, see the command ":POWer[:LEVel][:IMMediate][:AMPLitude]". When changing the power by UP|DOWN, the power variation will change based on the variation of the current set step value.

Setting format: [:SOURce]:POWer:STEP <value>

Query format: [:SOURce]:POWer:STEP?

Parameter:

<PowStep> Ampl Ref.

Range: 0.10dB [0.01dB，20dB].

Example: :POWer:STEP 1dB Ampl Step is 1 dB.

Reset status: 0.10dB

Key Entry: 【Amplitude】—>[Basic Config]->[Ampl Step]

[:SOURce]:POWer:ALC:BANDwidth|BWIDth <AlcBandWidth>

Function: This command sets ALC (automatic leveling control) Band which enables the signal generator to output different frequency bands. There are four ALC loop bandwidth settings in different states, 100 Hz, 1 kHz, 10 kHz and 100 kHz.

Notes:

1. When ALC Band is set to Auto, see the commands ":POWer:ALC:BANDwidth:AUTO"and ":POWer:ALC:BANDwidth:AUTO"; when it is improperly set, this setting is invalid;
2. When the internal baseband of the instrument is turned on, ACL Band is invalid, and the appropriate bandwidth should to be selected by baseband. For details, please refer to Section "5.2.5 Baseband" of the User Manual of AV1465 series Microwave Synthesized Signal Generator.

Setting format: [:SOURce]:POWer:ALC:BANDwidth|BWIDth 100Hz|1kHZ|10kHz|100kHz

Query format: [:SOURce]:POWer:ALC:BANDwidth|BWIDth?

Parameter:

<AlcBandWidth > Discrete data. ALC Band is as follows:

100Hz | 0: ALC Band is 100 Hz,

1kHz | 1: ALC Band is 1 kHz,

10kHz | 2: ALC Band is 10 kHz,

100kHz | 3: ALC Band is 100 kHz.

Example: [:SOURce]:POWer:ALC:BANDwidth|BWIDth 100Hz

ALC loop bandwidth is 100Hz.

Reset status: 10kHz

Key Entry: 【Amplitude】—>[ALC Band]

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

Function: This command is used for setting ALC(automatic leveling control) loop bandwidth mode. At auto mode, the signal generator selects proper ALC loop bandwidth automatically; at manual mode, ALC loop bandwidth is set by the user. For details, see the command "[:SOURce]:POWer:ALC:BANDwidth|BWIDth ".

Setting format: [:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO ON|OFF|1|0

Query format: [:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO?

Parameter:

<State> Boolean data is as follows:

ON | 1: ALC Band is in Auto mode,

OFF | 0: ALC Band is in Manu mode.

Example: [:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTO 1 ALC bandwidth mode is set to Manual.

Reset status: 1

Key Entry: 【Amplitude】—>[ALC Band Manual Auto]

3.3.4 LIST subsystem

LIST subsystem commands are used for setting the list sweep function of RF output signal. The commands and parameters of the subsystem are:

The following commands are used for setting operation modes:

- [:SOURce]:LIST:DIRection 47
- [:SOURce]:LIST:DWELl 47
- [:SOURce]:LIST:FREQuency 48
- [:SOURce]:LIST:FILL:POINts 48
- [:SOURce]:LIST:FILL:STARt 49
- [:SOURce]:LIST:FILL:STOP 49
- [:SOURce]:LIST:POWer 50
- [:SOURce]:LIST:RETRace 50
- [:SOURce]:LIST:TRIGger:SOURce 50
- [:SOURce]:LIST:FILL:POWer 51
- [:SOURce]:LIST:FILL:DWELl 51
- [:SOURce]:LIST:FILL:EXECute 51
- [:SOURce]:LIST:DELete 52

[:SOURce]:LIST:DIRection <Direc>

Function: This command sets Step Sweep Indirection. Users can select both Forward and Backward modes: Forward indicates sweep from the first point to the last point in list, and Backward indicates sweep from the last point to the first point in current list.

Setting format: [:SOURce]:LIST:DIRection UP|DOWN

Query format: [:SOURce]:LIST:DIRection?

Parameter:

<Direc> Discrete data. Step Sweep Indirection is as follows:

UP Start Forward sweep from the first point in list,

DOWN Start Backward sweep from the last point in list.

Example: :LIST:DIRection UP Set Step Sweep Indirection to Forward.

Reset status: UP

Key Entry: 【Sweep】—>[List Sweep]—>[Step Sweep Indirection Forward Backward]

[:SOURce]:LIST:DWELl <Val>{,{Val}}

Function: This command sets the dwell time of each sweep point in current list. If users need to set different dwell time, it is necessary to enter the corresponding dwell time for each point in list, i.e. the dwell time parameter value of list sweep points in turn, separated by commas. If the number of points input by users is less than that of the current list points, the number of points where the dwell time is not entered uses the current default.

Setting format: [:SOURce]:LIST:DWELl <val>{,{val}}

Query format: [:SOURce]:LIST:DWELl?

Parameter:

<Val> Dwell Time of List Sweep Points.

Range: 1ms [1ms，60s].

Example: :LIST:DWELl 30ms，20ms Set the dwell time of the first point in list to 30 ms, and the second point to 20 ms.

Key Entry: 【Sweep】—>[List Sweep]—>[Edit List...]—>[Time(ms)]

[:SOURce]:LIST:FREQuency <Val>{,{Val}}

Function: This command is used for setting the CW frequency of each sweep point in the current list. If the user needs to set different frequency value, every frequency point must be set with the corresponding frequency value. To this end, it is only necessary to input the frequency values of the sweep points successively with a comma between values. If the number of points input by the user is less than that of the current list, the points without list frequency will take the current default value.

Setting format: [:SOURce]:LIST:FREQuency <val>{,{val}}

Query format: [:SOURce]:LIST:FREQuency?

Parameter:

<Val> Frequency of list sweep point.

Model Range

1465A [100kHz~3GHz]
1465B [100kHz~6GHz]
1465C [100kHz~10GHz]
1465D [100kHz~20GHz]
1465E [100kHz~40GHz]
1465H [100kHz~50GHz]
1465L [100kHz~67GHz]

Example: :LIST:FREQuency The CW frequency in 300MHz，1GHz，500MHz lists are set to 300MHz, 1GHz, 500MHz successively.

Key Entry: 【Sweep】—>[List Sweep]—>[Edit List...]—>[Freq(MHz)]

[:SOURce]:LIST:FILL:POINts <Num>

Function: After this command is set, frequency point will be inserted according to Freq Start, Freq Stop and Equal Freq Interval. It should be noted that, if you don’t want the frequency points in the list to increase or decrease in sequence, please don't use this command. For related commands, see “:LIST:FILL:STARt”,“:LIST:FILL:STOP”.

Setting format: [:SOURce]:LIST:FILL:POINts <num>

Query format: [:SOURce]:LIST:FILL:POINts?

Parameter:

<Num> Insert Counts

Range: 3[2，801].

Example: :LIST:FILL:POINts 100 Set frequency points to 100 in list.

Reset status: 3

Key Entry: 【Sweep】—>[List Sweep]—>[Insert Counts]

[:SOURce]:LIST:FILL:STARt <FreqStart>

Function: This command sets Freq Start of List Sweep, which is used with Freq Stop and Insert Counts to automatically generate list sweep points. For setting the stop frequency and sweep points of the list, see the commands ":LIST:FILL:STOP",":LIST:FILL:POINts",:LIST:FILLL:POWer","LIST:FILL:DWELl".

Setting format: [:SOURce]:LIST:FILL:STARt <val>

Query format: [:SOURce]:LIST:FILL:STARt?

Parameter:

<FreqStart> Freq Start of List Sweep.

Model Range

1465A [100kHz~3GHz]
1465B [100kHz~6GHz]
1465C [100kHz~10GHz]
1465D [100kHz~20GHz]
1465E [100kHz~40GHz]
1465H [100kHz~50GHz]
1465L [100kHz~67GHz]

Example: :LIST:FILL:STARt 300MHz Set Freq Start of List Sweep to 300 MHz.

Key Entry: 【Sweep】—>[List Sweep]-> [Freq Start]

[:SOURce]:LIST:FILL:STOP <FreqStop>

Function: This command is used for setting the stop frequency for list sweep. It is used together with the start frequency and number of points of the list to generate the list sweep point automatically. For setting start frequency and number of sweep points of the list, see the commands ":LIST:FILL:STARt",":LIST:FILL:POINts",":LIST:FILLL:POWer","LIST:FILL:DWELl".

Setting format: [:SOURce]:LIST:FILL:STOP <val>

Query format: [:SOURce]:LIST:FILL:STOP?

Parameter:

<FreqStop> Freq Stop of List Sweep.

Model Range

1465A [100kHz~3GHz]
1465B [100kHz~6GHz]
1465C [100kHz~10GHz]
1465D [100kHz~20GHz]
1465E [100kHz~40GHz]
1465H [100kHz~50GHz]
1465L [100kHz~67GHz]

Example: :LIST:FILL:STOP 1GHz Set Freq Stop of List Sweep to 1 GHz.

Key Entry: 【Sweep】—>[List Sweep]-> [Freq Stop]

[:SOURce]:LIST:POWer <Val>{,{Val}}

Function: This command sets the amplitude of each sweep point in current list. If users need to set different offset for each list point, it is necessary to enter the corresponding offset value for each point in list, i.e. the amplitude offset of list sweep points in turn, separated by commas. If the number of points input by users is less than that of the current list points, the number of points where the list power offset is not entered uses the current default.

Setting format: [:SOURce]:LIST:POWer <val>{,{val}}

Query format: [:SOURce]:LIST:POWer?

Parameter:

<Val> Ampl Offset of List Sweep Points.

Range: 0dBm [-100dB，+100dB].

Example: :LIST:POWer 1dB，0.2dB，1.3dB，2.5dB，-3.6dB

Set the power offset in the list to 1dB，0.2dB，1.3dB，2.5dB，-3.6dB successively.

Key Entry: 【Sweep】—>[List Sweep]—>[Edit List...]—>[Offset(dB)]

[:SOURce]:LIST:RETRace <State>

Function: This command is used for setting RTC switch. After accomplishing the single list sweep, the output frequency of the signal generator stays on the first or the last point of the list. This command is only available in the single sweep mode.

Setting format: [:SOURce]:LIST:RETRace ON|OFF|1|0

Query format: [:SOURce]:LIST:RETRace?

Parameter:

<State> Boolean data is as follows:

ON | 1: RTC Switch ON; after accomplishing sweep, the output frequency stays on the first frequency point of the list.

OFF | 0: RTC Switch OFF; after accomplishing sweep, the output frequency stays on the last frequency point of the list.

Example: :LIST:RETRace 0 RTC Switch OFF; after accomplishing the single list sweep, the output CW frequency of the signal generator will stay on the last frequency point of the list.

Reset status: 0

Key Entry: 【Sweep】—>[Sweep Mode]->[RTC Switch ON OFF]

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

Function: This command is used for setting the trigger source for initiating list sweep. The trigger sources are available in four modes, namely, Auto, Bus, Ext and Key. For related command, see the command "TRIGger[:SEQuence]:SOURce".

Setting format: [:SOURce]:LIST:TRIGger:SOURce IMMediate|BUS|EXTernal|KEY

Query format: [:SOURce]:LIST:TRIGger:SOURce?

Parameter:

<Source> Discrete data. List Trig is as follows:

IMMediate, the trigger signal is always true, when a sweep is completed, the system

automatically triggers the next sweep.

BUS Bus, the trigger source is from GPIB group execute trigger, or triggered by “*TRG” command.

EXTernal Ext, the trigger signal is from the trigger input connector on the rear panel.

KEY Key, the trigger signal is from the trigger key on the front panel.

Example: :LIST:TRIGger:SOURce BUS Set the trigger source for list sweep to Bus.

Reset status: IMM

Key Entry: 【Sweep】—>[List Sweep]—>[List Trig]

[:SOURce]:LIST:FIll:POWer <Val>

Function: This command is used for setting the list sweep power offset. It is used together with the start frequency and number of points of the list to generate the list sweep point. For setting start frequency and number of sweep points of the list, see the commands “:LIST:FILL:STARt”,“:LIST:FILL:POINts”,“:LIST:FILLL:STOP”,“LIST:FILL:DWELl”

Setting format: [:SOURce]:LIST:FILL:POWer <val>

Query format: [:SOURce]:LIST:FILL:POWer?

Parameter:

<val> Power offset for all list sweep points.

Range: 0dBm [-100dB，+100dB]，

Example: :LIST:FILL:POWer 10dB Set the list sweep power offset to 10dB.

Key Entry: 【Sweep】—> [List Sweep]—>[All List Ampl Offset]

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

Function: This command is used for setting the dwell time for all list sweep points. It is used together with the start frequency and number of points of the list to generate the list sweep point automatically. For setting start frequency and number of sweep points of the list, see the commands “:LIST:FILL:STARt”,“:LIST:FILL:POINts”,“:LIST:FILLL:STOP”,“LIST:FILL:POWer”.

Setting format: [:SOURce]:LIST:FILL:DWELl <val>

Query format: [:SOURce]:LIST:FILL:DWELl?

Parameter:

<val> Dwell time for all list sweep points.

Range: 1ms [1ms，60s]. ，

Example: :LIST:FILL:POWer 10ms Set the dwell time for all list sweep points to 10ms.

Key Entry: 【Sweep】—> [List sweep]—>[All List Dwell Time]

[:SOURce]:LIST:FIll:EXECute

Function: This command is used for generating list sweep points according to set start frequency, stop frequency, number of points of list, power offset for all points and dwell time for all points; see the commands “:LIST:FILL:STARt”,“:LIST:FILL:POINts”,“:LIST:FILLL:STOP”,“LIST:FILL:POWer”,“LIST:FILL:STOP”.

Setting format: [:SOURce]:LIST:FILL:EXECute

Parameter:

Example: :LIST:FILL:EXECute Accomplish automatic filling of list.

Key Entry: 【Sweep】—>[List Sweep]—>[Auto Fill]

Note: For setting only.

[:SOURce]:LIST:DELete <Mode>

Function: This command is used for deleting the points in the list of List Sweep.

Setting format: [:SOURce]:LIST:FILL:EXECute

Parameter: Discrete data. Options for deleting the list points are:

CURRent Current point

ALL All points

Example: :LIST:DELete ALL Delete all points in the list.

Key Entry: 【Sweep】—>[List Sweep]—>[Edit List]—>[Del All]

Note: For setting only.

3.3.5 LFOutput subsystem

The following commands are used for setting operation modes:

- [:SOURce]:LFOutput:AMPLitude 52
- [:SOURce]:LFOutput:FREQuency 52
- [:SOURce]:LFOutput:FREQuency:ALTernate 53
- [:SOURce]:LFOutput:FREQuency:ALTernate:AMPLitude:PERCent 54
- [:SOURce]:LFOutput:NOISe 54
- [:SOURce]:LFOutput:RAMP 54
- [:SOURce]:LFOutput:SHAPe 55
- [:SOURce]:LFOutput:STATe 55
- [:SOURce]:LFOutput:SWEep:TIME 55

[:SOURce]:LFOutput:AMPLitude <Ampl>

Function: This command is used for setting the output signal amplitude from the LF output BNC (Bayonet Nut Connector) of the signal generator

Setting format: [:SOURce]:LFOutput:AMPLitude <val>(unit:VPP|Mvpp|VRMS)

Query format: [:SOURce]:LFOutput:AMPLitude?

Parameter:

<Ampl> LF output signal amplitude.

Range: 2.000Vpp[0.002Vpp,4.000Vpp].

Example: :LFOutput:AMPLitude 1Vpp LF output signal amplitude is set to 1 Vpp.

Reset status: 2.000Vpp

Key Entry: **【Frequency】—>[LF Out]—>[LF Ampl]**

[:SOURce]:LFOutput:FREQuency <Frequency>

Function: This command is used for setting LF output. It should be noted that, when LF Waveform

is set to SweepSinc or DualSinc, this command will set the start frequency of sweep sine and the frequency 1 of dual sine. For setting LF waveform, see the command "LFOutput:SHAPe".

Setting format: [:SOURce]:LFOutput:FREQuency <val>

Query format: [:SOURce]:LFOutput:FREQuency?

Parameter:

<Frequency> LF output signal frequency.

Range: 400Hz[0.01Hz，10MHz].

Freq Start of Sweep Sin

Range: 400Hz[0.01Hz,9.99999999MHz]. When inputting 10MHz, there will be no error indication in the software, and the start frequency of sweep sine will be set to 9.99999999MHz automatically.

DualSinc Frequency1

Range: 400Hz[0.01Hz,10MHz].

Example: :LFOutput:FREQuency 1MHz Set LF output signal frequency to 1 MHz.

Reset status: 400Hz

Key Entry:

- **【Frequency】—>[LF Out]—>[LF]**
- **【Frequency】—>[Sweep Sinc]—>[Freq Start]**
- **【Frequency】—>[Dual Sinc>>—>[Frequency1]**

[:SOURce]:LFOutput:FREQuency:ALTernate <Frequency>

Function: This command is used for setting the stop frequency of sweep sine or the frequency 2 of dual sine when LF Waveform is set to SweepSinc or DualSinc. For setting LF waveform, see the command "LFOutput:SHAPe". If LF Waveform is not set to SweepSinc or DualSinc, the value of frequency 2 of dual sine will be modified as default.

Setting format: [:SOURce]:LFOutput:FREQuency:ALTernate <val>

Query format: [:SOURce]:LFOutput:FREQuency:ALTernate?

Parameter:

<Frequency> Frequency2 of dual sine.

Range: 400Hz[0.01Hz，10MHz].

Stop frequency of sweep sine or

Range: 1MHz[0.02Hz，10MHz]. If the stop frequency of sweep sine is set to 0.01Hz, there will be no error indication in the software, and the stop frequency of sweep sine will be set to 0.02Hz automatically.

Example: :LFOutput:FREQuency:ALTernate 1MHz Set the stop frequency of sweep sine or the frequency 2 of dual sine to 1MHz.

Reset status: 400Hz

Key Entry:

- **【Frequency】—>[Sweep Sinc]—>[Freq Stop]**
- **【Frequency】—>[Dual Sinc]—>[Frequency2]**

[:SOURce]:LFOutput:FREQuency:ALTernate:AMPLitude:PERCent <Percent>

Function: This command is used for setting the percentage of amplitude of second tone to that of total output signal in dual sine waveform when LF Waveform is set to DualSinc; for example, if the second tone accounts for 20% of the total waveform power, the first tone will account for 80% of the total power output.

Setting format: [:SOURce]:LFOutput:FREQuency:ALTernate:AMPLitude:PRECent <val>

Query format: [:SOURce]:LFOutput:FREQuency:ALTernate:AMPLitude:PRECent?

Parameter:

<Percent> Amplitude percentage of frequency 2 of dual sine.

Range: 50 [0，100].

Example: :LFOutput:FREQuency:ALTernate:AMPLitude:PRECent 20

Set the percentage of second waveform of dual sine to the total signal output power to 20%.

Reset status: 50%

Key Entry: 【Frequency】—> [Dual Sinc]—>[Freq 2 Ampl Percent]

[:SOURce]:LFOutput:NOISe <Mode>

Function: This command is used for setting signal output types when LF Waveform is set to White noise, which includes: While noise and Gauss. For setting LF waveform, see the command ":LFOutput:SHAPe".

Setting format: [:SOURce]:LFOutput: NOISe UNIForm|GAUSsian

Query format: [:SOURce]:LFOutput: NOISe?

Parameter:

<Mode> Discrete data. Options for type of LF noise signal are:

UNIForm White noise,

GAUSsian Gauss.

Example: :LFOutput: NOISe GAUSsian LF noise is of Gauss type.

Reset status: UNIF

Key Entry: 【Frequency】—>[LF Out]—>[LF Waveform>>]—>[White noise]

[:SOURce]:LFOutput:RAMP <Mode>

Function: This command is used for setting signal output types when LF Waveform is set as Zigzag, which include Zigzag-up and Zigzag-down. For setting LF waveform, see the command ":LFOutput:SHAPe".

Setting format: [:SOURce]:LFOutput:RAMP POSitive|NEGative

Query format: [:SOURce]:LFOutput:RAMP?

Parameter:

<Mode> Discrete data. Options for type of Zigzag signal are:

POSitive Zigzag-up,

NEGative Zigzag-down.

Example: :LFOutput:RAMP NEGative LF Zigzag signal is of Zigzag-down type.

Reset status: POS

Key Entry: 【Frequency】—>[LF Out]—>[LF Waveform]—>[Zigzag>>]

[:SOURce]:LFOutput:SHAPe <Mode>

Function: This command sets LF signal output waveform. There are seven waveforms for selection: Sinc, Square, Triangle, Zigzag, White noise, SweepSinc and DualSinc.

Setting format: [:SOURce]:LFOutput:SHAPe SINE|SQUare|TRIangle|RAMP |NOISe|SWEPtsine|DUALsine

Query format: [:SOURce]:LFOutput:SHAPe?

Parameter:

<Mode> Discrete data. LF Waveform is as follows:

SINE Sine,
SQUare Square,
TRIangle Triangle,
RAMP Zigzag,
NOISe Noise,
SWEPtsine SweepSinc,
DUALsine DualSinc.

Example: :LFOutput:SHAPe TRIangle LF waveform of the signal generator is triangle.

Reset status: SINE

Key Entry: 【Frequency】—>[LF Out]—>[LF Waveform>>]

[:SOURce]:LFOutput:STATe <State>

Function: This command is used for setting LF Output of the signal generator.

Setting format: [:SOURce]:LFOutput:STATe ON|OFF|1|0

Query format: [:SOURce]:LFOutput:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: LF Output is turned on, and LF signal output is turned on accordingly.
OFF | 0: LF Output is turned off, and LF signal output is turned off accordingly.

Example: :LFOutput:STATe OFF Turn off LF signal output.

Reset status: 0

Key Entry: 【Frequency】—>[LF Out]—>[LF Output ON OFF]

[:SOURce]:LFOutput:SWEep:TIME <Time>

Function: This command is used for setting sweep time when LF Waveform is set to SweepSinc.

Setting format: [:SOURce]:LFOutput:SWEep:TIME <val>

Query format: [:SOURce]:LFOutput:SWEep:TIME?

Parameter:

<Time> Sweep time of sweep sine.

Range: 10ms[100us，2s].

Example: :LFOutput:SWEep:TIME 200ms Sweep time of sweep sine is set to 200ms.

Reset status: 10ms

Key Entry: 【Frequency】—>[Sweep Sinc>>]—>[Sweep Time]

3.3.6 SWEep subsystem

SWEep subsystem commands are used for controlling the sweep frequency functions of RF output signal. The commands and parameters of the subsystem are:

The following commands are used for setting operation modes:

- [:SOURce]:SWEep:DIRection 56
- [:SOURce]:SWEep:DWELl 56
- [:SOURce]:SWEep:POINts 57
- [:SOURce]:SWEep:STEP 57
- [:SOURce]:SWEep:TIME:AUTO 57
- [:SOURce]:SWEep:TIME 57
- [:SOURce]:SWEep:GENeration 57
- [:SOURce]:SWEep:TRIGger:SOURce 57

[:SOURce]:SWEep:DIRection <Direction>

Function: This command sets Step Sweep Indirection, including: Forward and Backward. Forward indicates Step Sweep from Freq Start to Freq Stop, and Backward indicates sweep from Freq Stop to Freq Start.

Setting format: [:SOURce]:SWEep:DIRection UP|DOWN

Query format: [:SOURce]:SWEep:DIRection?

Parameter:

<Direction> Discrete data. Step Sweep Indirection is as follows:

UP : Forward,

DOWN : Backward.

Example: :SWEep:DIRection DOWN Step sweep direction is set as backward.

Reset status: UP

Key Entry: 【Sweep】—>[Sweep Mode>>]—>[Step Sweep>>]—>[Step Sweep Indirection Forward Backward]

[:SOURce]:SWEep:DWELl <DwellTime>

Function: This command is used for setting the dwell time of step sweep which refers to the pause time during the sweep process of current step frequency points. The dwell time set by the user will be valid when the trigger source of step sweep is set as Auto. For setting trigger source, see the command “SWEep:TRIGger:SOURce”.

Setting format: [:SOURce]:SWEep:DWELl <value>

Query format: [:SOURce]:SWEep:DWELl?

Parameter:

<Val> Step dwell.

Range: 10.000ms[1ms，60s].

Example: :SWEep:DWELl 1s Set All Step Dwell Time to 1 s.

Reset status: 10.000ms

Key Entry: **【Sweep】—>[Step Sweep]—>[Step Dwell]**

[:SOURce]:SWEep:POINts <Num>

Function: This command sets the current step counts.

Setting format: [:SOURce]:SWEep:POINts <val>

Query format: [:SOURce]:SWEep:POINts?

Parameter:

<Num> Step Counts

Range: 11[2，801].

Example: :SWEep:POINts 101 Set Step Counts to 101.

Reset status: 11

Key Entry: **【Sweep】—>[Step Sweep]—>[Step Counts]**

[:SOURce]:SWEep:STEP <FreqStep>

Function: This command is used for setting the frequency step of step sweep. After the start and stop frequency of step sweep as well as the step value are set by the user, step sweep will start according to the step value. For setting start and stop frequency of step sweep, see the commands “:FREQuecny:STARt”,“:FREQuenct:STOP” .

Setting format: [:SOURce]:SWEep:STEP <value>

Query format: [:SOURce]:SWEep:STEP?

Parameter:

<FreqStep> Frequency step of step sweep.

Range: (Stop frequency – Start frequency)/ (Max. step points – 1) — (Stop frequency – Start frequency) / (Min. step points – 1).

Example: :SWEep:STEP 1MHz Set Step Sweep interval to 1MHz.

Reset status: (Stop frequency – Start frequency)/ (Min. step points - 1).

Key Entry: None

CAUTION

Frequency step of step sweep

To ensure equispaced step points, the set frequency step value will generally be slightly adjusted. It will not be adjusted only when it can be divided exactly by the sweep width.

[:SOURce]:SWEep:TIME:AUTO <state>

Function: This command is used for setting ramp sweep time type

Setting format: [:SOURce]:SWEep:TIME:AUTO ON|OFF|1|0

Query format: [:SOURce]:SWEep:TIME:AUTO?

Parameter:

<State> Boolean data is as follows:

ON | 1: Sweep ime type is set to Auto.

OFF | 0: Sweep time type is set to Manual.

Example: :SWEep:TIME:AUTO 1 Set sweep time type to Auto

Reset status: 1

Key Entry: 【Sweep】—>[Ramp Sweep] —>[Sweep Dwell Time Type] —>[Auto Manual]

[:SOURce]:SWEep:TIME <state>

Function: When ramp sweep time type is set to Manual, this command is used for setting sweep time

Setting format: [:SOURce]:SWEep:TIME <val>

Query format: [:SOURce]:SWEep:TIME?

Parameter:

<Time> Sweep time.

Range: 100ms[100ms，200s].

Example: :SWEep:TIME 200ms Set sweep time to Auto

Reset status: 2346.667ms

Key Entry: 【Sweep】—>[Ramp Sweep] —>[Sweep Dwell Time]

[:SOURce]:SWEep:GENeration <Mode>

Function: When sweep mode is set, this command is used for setting sweep mode

Setting format: [:SOURce]:SWEep:GENeration ANALog|STEPped

Query format: [:SOURce]:SWEep:GENeration?

Parameter:

<Mode> Discrete data. Options for sweep mode are:

ANALog Ramp Sweep,

STEPped Step Sweep.

Example: :SWEep:GENeration STEPped Set sweep mode to step sweep.

Reset status: STEP

Key Entry: None

[:SOURce]:SWEep:TRIGger:SOURce <Mode>

Function: This command sets Step Trig. It includes Auto, Bus, Ext and Key. For related command, see the command “TRIGger[:SEQuence]:SOURce”.

Setting format: [:SOURce]:SWEep:TRIGger:SOURce IMMediate|BUS|EXTernal|KEY

Query format: [:SOURce]:SWEep:TRIGger:SOURce?

Parameter:

<Mode> Discrete data. Step Trig is as follows:

IMMediate Auto, the trigger signal is always true, when a sweep is completed, the system automatically triggers the next sweep.

BUS Bus, the trigger source is from GPIB group execute trigger, or triggered by *TRG command.

EXTernal Ext, the trigger signal is from the trigger input connector on the rear panel.
KEY |Trigger key; trigger source comes from the trigger key of the front panel.

Example: :SWEep: TRIGger:SOURce BUS Set Step Trig to Bus mode.

Reset status: IMM

Key Entry: 【Sweep】—>[Step Sweep]—>[Step Trig]

3.3.7 PULM subsystem

PULM subsystem commands are used for controlling the pulse modulation functions of RF output signal.

The commands and parameters of the subsystem are:

The following commands are used for setting pulse modulation mode:

- [:SOURce]:PULM:EXTernal:POLarity 58
- [:SOURce]:PULM:INTernal:DELay 59
- [:SOURce]:PULM:INTernal:FREQuency 60
- [:SOURce]:PULM:INTernal:PERiod 60
- [:SOURce]:PULM:INTernal:PWIDth 61
- [:SOURce]:PULM:SOURce 61
- [:SOURce]:PULM:STATe 62
- [:SOURce]:PULM:INTernal:JITTered:MODE 62
- [:SOURce]:PULM:INTernal:JITTered:PERCent 62
- [:SOURce]:PULM:INTernal:PTRain:DATA 63
- [:SOURce]:PULM:INTernal:PTRain:DELete 63
- [:SOURce]:PULM:INTernal:PTRain:POINts 64
- [:SOURce]:PULM:INTernal:PTRain:PRESet 64
- [:SOURce]:PULM:INTernal:SLIDing:STEP 64
- [:SOURce]:PULM:INTernal:SLIDing:POINts 65
- [:SOURce]:PULM:INTernal:STAGger:INSert 65
- [:SOURce]:PULM:INTernal:STAGger:POINts 65
- [:SOURce]:PULM:INTernal:STAGger:DELete 66
- [:SOURce]:PULM:INTernal:STAGger:PRESet 66
- [:SOURce]:PULM:LFM:BWIDth 66
- [:SOURce]:PULM:LFM:DIRection 67
- [:SOURce]:PULM:LFM:STATe 67

[:SOURce]:PULM:EXTernal:POLarity <Mode>

Function: The command logically inverts the external input pulse signal, i.e. when Pulse Source is EXT, the pulse signal input from pulse input port on the front panel of the signal generator is TTL high level signal or is inverted to TTL low level signal. For selecting pulse source, see the command ":PULM:SOURce".

Setting format: [:SOURce]:PULM:EXTernal:POLarity INVerted|NORMal

Query format: [:SOURce]:PULM:EXTernal:POLarity?

Parameter:

<Mode> Discrete data. Input Direction is as follows:

NORMal Input Direction is turned off and the input pulse signal is TTL high level.
INVerted Input Direction is turned on and the input pulse signal is TTL low level.

Example: :PULM:ENTernal:POLarity INV External input pulse signal is inverted to TTL low level.

Reset status: NORM

Key Entry: 【Pulse】—>[Basic Config]—>[Input Direction ON OFF]

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

Function: This command is used for setting the pulse delay of the pulse modulation. The actual settable maximum delay depends on the current pulse period set by the user. In addition, it should be noted that, the set pulse delay is valid only when the pulse source is set to Auto, Square, D-Pulse or Trig. And when it is set to Trig, there is 100ns inherent delay in the pulse delay. For related commands, see the commands ":PULM:INTernal:PERiod","[:PULM:SOURce](#)".

Setting format: [:SOURce]:PULM:INTernal:DELay <val>

Query format: [:SOURce]:PULM:INTernal:DELay?

Parameter:

<DelayTime> Pulse delay time of pulse modulation.

Non-Trig Mode: 0s[0ns，42.000000000s],

Trig Mode: 0s[100ns，42.000000000s].

Example: :PULM:INTernal:DELay 1ms Set pulse delay to 1 ms

Reset status: 0s

Key Entry: 【Pulse】—>[Basic Config]—>[Delay]

[:SOURce]:PULM:INTernal:FREQuency <Frequency>

Function: This command sets PRF of pulse modulation. When the pulse source is set to Square, the duty ratio of the pulse signal output is 50%. This command may be used for changing the frequency of the square signal. For setting pulse source, see the command ":PULM:SOURce".

Setting format: [:SOURce]:PULM:INTernal:FREQuency <val>

Query format: [:SOURce]:PULM:INTernal:FREQuency?

Parameter:

<Frequency> PRF of pulse modulation.

Range: 1kHz [0.023Hz，25MHz]

Example: :PULM:INTernal:FREQuency 1MHz Set PRF to 1 MHz.

Reset: 1kHz

Key Entry: 【Pulse】—>[Basic Config]—>[PRF]

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

Function: This command sets the period of pulse signal generated inside the signal generator. If the set period is smaller than or equal to the current pulse width, the pulse width will be automatically adjusted to be smaller than pulse period. For setting pulse width, see the command ":PULM:INTernal:PWIDth".

Setting format: [:SOURce]:PULM:INTernal:PERiod <value>

Query format: [:SOURce]:PULM:INTernal:PERiod?

Parameter:

<Percent> Pulse period.

Range: 1.000000ms[40ns，42.000000000s].

Example: :PULM:INTernal:PERiod 10ms Pulse signal period is 10 ms.

Reset status: 1.000000ms

Key Entry: 【Pulse】—>[Basic Config]—>[Period]

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

Function: This command sets Width of pulse signal generated inside the signal generator. If the set pulse width is greater than or equal to the current pulse period, the pulse width will be automatically adjusted to be smaller than the current pulse period. Besides, if the set pulse width is less than 1 us, it is recommended to execute power search function. When pulse source is set to Staggered, the pulse width in staggered list will be a unified value, and should also be changed through this command.

Setting format: [:SOURce]:PULM:INTernal:PWIDth <val>

Query format: [:SOURce]:PULM:INTernal:PWIDth?

Parameter:

<PWidth> Pulse signal width.

Range: 50.000us [20ns，41.999999990s].

Example: :PULM:INTernal:PWIDth 10us Set the pulse signal width to 10 us.

Reset status: 50.000us

Key Entry: 【Pulse】—>[Basic Config]—>[Width]

[:SOURce]:PULM:SOURce <Mode>

Function: This command sets Source of Pulse Modulation, including: EXT, Scalar, Auto, Square, D-Pulse, Pulse Train, Gate, Trig, Jittered, Staggered and Sliding. In Scalar mode, it is not allowed to change associated pulse parameters, and the signal generator will automatically output pulse signal of 18-microsecond pulse width and 36-microsecond period.

Setting format: [:SOURce]:PULM:SOURce EXTernal|SCALar|INTernal|SQUare|DOUBlet |PTRain|GATEd|TRIGgered|JITTered|STAGger|SLIDing

Query format: [:SOURce]:PULM:SOURce?

Parameter:

<Mode> Discrete data. Source is as follows:

EXTernal	Source is EXT,
SCALar	Source is Scalar and outputs 27.8 kHz square wave,
INTernal	Pulse Source is Auto,
SQUare	Source is Square.
DOUBlet	Source is D-Pulse.
PTRain	Source is Pulse Train.

GATEd	Source is Gate.
TRIGgered	Auto mode is activated. In this mode, the period is the external sync pulse period, and the pulse width is the local pulse width setting.
JITTered	Source is Jittered.
STAGger	Source is Staggered.
SLIDing	Source is Sliding.

Example: :PULM:SOURce SQUare Set Source to Square.

Reset status: INT

Key Entry: 【Pulse】—>[Basic Config]—>[Source]

[:SOURce]:PULM:STATe <State>

Function: This command sets whether to output the pulse modulation signal of the signal generator.

Setting format: [:SOURce]:PULM:STATe ON|OFF|1|0

Query format: [:SOURce]:PULM:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: Pulse Modulation is turned on,

OFF | 0: Pulse Modulation is turned off.

Example: :PULM:STATe 1 Pulse Modulation is turned on.

Reset: 0

Key Entry: 【Pulse】—>[Basic Config]—>[Pulse Modulation ON OFF]

[:SOURce]:PULM:INTernal:JITTered:MODE <Mode>

Function: This command is used for setting the jitter modes of pulse modulation period, which include Random and Gaussion. At Random mode, the pulse period changes from 0 to 10% at random; at Gaussion mode, it changes from 0 to 10% in Gaussian distribution.

Setting format: [:SOURce]:PULM:INTernal:JITTered:MODE UNIForm|GAUSsian

Query format: [:SOURce]:PULM:INTernal:JITTered:MODE?

Parameter:

<Mode> Discrete data. Options for PRF jitter mode for pulse modulation are:

UNIForm | 0: Random,

GAUSsian | 1: Gauss.

Example: [:SOURce]:PULM:INTernal:JITTered:MODE GAUS Set PRF jitter mode to Gaussion.

Reset status: GAUS

Key Entry: 【PULSE】—>[Jittered]—>[Dither Style]

[:SOURce]:PULM:INTernal:JITTered:PERCent <Percent>

Function: This command is used for setting PRF jitter percentage of pulse modulation signal. When the pulse source is set to Jittered, this percentage will change the pulse period while the pulse width will remain unchanged.

Setting format: [:SOURce]:PULM:INTernal:JITTered:PERCent <val>

Query format: [:SOURce]:PULM:INTernal:JITTered:PERCent?

Parameter:

<Percent> PRF jitter percentage of pulse modulation.

Range: 0[0，10].

Example: [:SOURce]:PULM:INTernal:JITTered:PERCent 5 Pulse jitter percentage is 5%.

Reset status: 10

Key Entry: 【PULSE】—>[Jittered]—>[Dither Percent]

[:SOURce]:PULM:INTernal:PTRain:DATA <PlsWidth>,<PlsPerd>{ PlsWidth , PlsPerd …}

Function: When the pulse source for pulse modulation of the signal generator is set to M-Pulse, this command is used for setting the pulse train function, which includes the following parametric components: Pulse width and period. It supports at most 1024 pulse trains. Pulse width and period should be set in pairs; otherwise, this command parameter will be invalid.

Setting format: [:SOURce]:PULM:INTernal:PTRain:DATA <pls_width>,<pls_period>

{pls_width，pls_period…}

Parameter:

<PlsWidth> Pulse width of pulse train.

Range: 50us[0.02us，42s].

< PlsPerd> Pulse period of pulse train.

Range: 1ms[0.03us，42s].

Example: [:SOURce]:PULM:INTernal:PTRain:DATA 100us,1ms,200us,2ms

Set pulse trains with pulse width of 100us and pulse period of 1ms, and those with pulse width of 200us and pulse period of 2ms.

Key Entry: 【PULSE】—>[Pulse Train]—>[Edit Pulse Train]

Note: for setting only.

[:SOURce]:PULM:INTernal:PTRain:DELete <Index>

Function: This command is used for deleting any index point in the pulse trains list of the signal generator. If the index number to be deleted exceeds the range of list points, this operation will be invalid. In this case, the user may query the current list points before deletion operation. If the queried number of points is 1, for valid deletion of the pulse trains list, the index value should be set to zero. For setting the points of pulse train, see the command "PULM:INTernal:PTRain:POINts".

Setting format: [:SOURce]:PULM:INTernal:PTRain:DELete <num>

Parameter:

<Index> Index of pulse trains list.

Range: 0[0，1023].

Example: [:SOURce]:PULM:INTernal: PTRain:DELete 1 Delete the point with index number of 1 in the current pulse trains list.

Key Entry: None

Note: for setting only.

[:SOURce]:PULM:INTernal:PTRain:POINts?

Function: This command is used for querying points of current pulse train. "Zero" indicates that the current list is empty, and "Nonzero" indicates the actual points in the current list. It should be noted that, if the user operates the interface manually, the index in the pulse trains list will start from 0; that is, if the queried number of points in the list is 1, the index actually shown in the list is 0.

Query format: [:SOURce]:PULM:INTernal:PTRain:POINts?

Return value:

<Num> Integer data; returned points of pulse trains list.

Range: 0[0，1023].

Example: [:SOURce]:PULM:INTernal:PTRain:POINts? Query points of current pulse trains list.

Reset status: 0

Key Entry: None

Note: for query only.

[:SOURce]:PULM:INTernal:PTRain:PRESet

Function: This command is used for deleting all points in the pulse trains list. If the current list is empty, no operation will be conducted. The user may query the points in the current list before deletion operation. For the command related to pulse train points, see "PULM:INTernal:PTRain:POINts".

Setting format: [:SOURce]:PULM:INTernal:PTRain:PRESet

Example: [:SOURce]:PULM:INTernal:PTRain:PRESet Delete all point in the pulse train list.

Key Entry: 【PULSE】—>[Pulse Train]—>[Edit Pulse Train] —>[Del All]

Note: for setting only.

[:SOURce]:PULM:INTernal:SLIDing:STEP <StepTime>

Function: When the pulse source is set to Sliding, this command is used for setting pulse sliding step. Based on the current pulse period, the pulse signal will increase progressively with the set sliding step. For example, when the step point is set to 1024, current pulse period is 1ms and sliding step is 100us, the pulse period will change between 1ms and 103.4ms.

Setting format: [:SOURce]:PULM:INTernal:SLIDing:STEP <val><time unit>

Query format: [:SOURce]:PULM:INTernal:SLIDing:STEP?

Parameter:

<StepTime> Sliding step.

Range: 100ns[0s，42ms].

Example: [:SOURce]:PULM:INTernal:SLIDing:STEP 100us Set pulse sliding step to 100us.

Reset: 100ns

Key Entry: 【PULSE】—>[Sliding]—>[Sliding Step]

[:SOURce]:PULM:INTernal:SLIDing:POINts <Num>

Function: When the pulse source is set to Sliding, this command is used for setting pulse sliding points.

Setting format: [:SOURce]:PULM:INTernal:SLIDing:POINts <Num>

Query format: [:SOURce]:PULM:INTernal:SLIDing:POINts?

Parameter:

<StepTime> Sliding points.

Range: [2，1024].

Example: [:SOURce]:PULM:INTernal:SLIDing:POINts 10 Set sliding points to 10.

Reset: 1024

Key Entry: 【PULSE】—>[Sliding]—>[Sliding Counts]

[:SOURce]:PULM:INTernal:STAGger:INSert <Index>,<PlsPerd>

Function: This command is used for inserting pulse staggered point. The user should input index and pulse period successively. The multiple of at most five is supported for the staggered pulse of the signal generator. Therefore, the user may query the current staggered points before using this command. When the maximum number of staggered points is exceeded, current set staggered point will not be inserted to this staggered list. In addition, pulse width in the list will be a unified value. For related commands, see "PULM:INTernal:STAGger:POINts","[:PULM:INTernal:PWIDth](#)".

Setting format: [:SOURce]:PULM:INTernal:STAGger:INSert <index>,<pls_period>

Parameter:

<Index> Index of PRF staggered list points.

Range: 0[0，4].

< PlsPerd> Pulse period.

Range: 1.000000ms[40ns，42.000000000s].

Example: [:SOURce]:PULM:INTernal:STAGger:INSert 1，1ms

Insert a staggered point with 1ms period in index 1 of the current PRF staggered list.

Key Entry: 【PULSE】—>[Staggered]—>[EditStagg List…] —>[Insert]

Note: for setting only.

[:SOURce]:PULM:INTernal:STAGger:POINts?

Function: This command is used for querying current PRF staggered points. "Zero" indicates that there is no staggered point in the current list, and "Nonzero" indicates the actual points in the current list. It should be noted that, if the user operates the interface manually, the index in the Stagg List will start from 0; that is, if the queried number of points in the list is 1, the index actually shown in the list is 0.

Query format: [:SOURce]:PULM:INTernal:STAGger:POINts?

Return value:

<Num> Integer data; returned points of Stagg List.

Range: 0[0，5].

Example: [:SOURce]:PULM:INTernal:STAGger:POINts?

Query current PRF staggered points.

Reset status: 0

Key Entry: None

Note: for query only.

[:SOURce]:PULM:INTernal:STAGger:DELete <Index>

Function: This command is used for deleting any index in Stagg Listof the signal generator. If the index point to be deleted exceeds the range of list points, this operation will be invalid. In this case, user may query the current list points before deletion operation. It should be noted that, if the queried staggered point number is 1, for valid deletion, the index shall be set to zero. For points in Stagg List, see the command "PULM:INTernal:STAGger:POINts".

Setting format: [:SOURce]:PULM:INTernal:STAGger:DELete <num>

Parameter:

<Index> Index of Stagg List.

Range: 0[0，4].

Example: [:SOURce]:PULM:INTernal:STAGger:DELete 1 Delete the point with index 1 in the current Stagg List.

Reset status: 0

Key Entry: 【PULSE】—>[Staggered]—>[EditStagg List…] —>[Del Current]

Note: for setting only.

[:SOURce]:PULM:INTernal:STAGger:PRESet

Function: This command is used for deleting all points in Stagg List. If there is no staggered point in current list, this operation will be invalid. In this case, user may query the staggered points in the current list before deletion operation. For querying points in Stagg List, see the command "PULM:INTernal:STAGger:POINts".

Setting format: [:SOURce]:PULM:INTernal:STAGger:PRESet

Example: [:SOURce]:PULM:INTernal:STAGger:PRESet Delete all points in Stagg List.

Key Entry: 【PULSE】—>[Staggered]—>[EditStagg List…] —>[Del All]

Note: for setting only.

[:SOURce]:PULM:LFM:BWIDth <LfmBwidth>

Function: This command is used for setting LFM bandwidth. After setting, the output signal centers on the output carrier frequency of the signal generator, with the FM range equal to carrier frequency ± 1/2 bandwidth.

Setting format: [:SOURce]:PULM:LFM:BWIDth <val><freq unit>

Query format: [:SOURce]:PULM:LFM:BWIDth?

Parameter:

<LfmBwidth> LFM bandwidth.

Range: 4MHz[0Hz，200MHz].

Example: [:SOURce]:PULM:LFM:BWIDth 20MHz LFM bandwidth is 20MHz.

Key Entry: 【Signal】—>[LFM]—> [BandWidth]

[:SOURce]:PULM:LFM:DIRection <Mode>

Function: This command is used for setting LFM direction. When LFM Direction is set to Increase, the FM signal output of the signal generator will increase progressively from negative to positive bandwidth; when the direction is set to Decrease, it will change from positive to negative bandwidth. After turning off LFM, the instrument will turn off pulse modulation and IQ modulation automatically.

Setting format: [:SOURce]:PULM:LFM:DIRection POSitive|NEGative

Query format: [:SOURce]:PULM:LFM:DIRection?

Parameter:

<Mode> Discrete data. Options for LFM Direction are:

POSitive | 0: Increase,

NEGative | 1: Decrease.

Example: [:SOURce]:PULM:LFM:DIRection POS LFM Direction is Increase.

Reset status: POS

Key Entry: 【Signal】—>[LFM]—> [LFM Direction]

[:SOURce]:PULM:LFM:STATe <State>

Function: This command is used for setting LFM. When LFM is set to ON, the signal generator will initiate pulse modulation and IQ modulation automatically. After turning off LFM, the instrument will turn off pulse modulation and IQ modulation automatically.

Setting format: [:SOURce]:PULM:LFM:STATe ON|OFF|1|0

Query format: [:SOURce]:PULM:LFM:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: LFM ON,

OFF | 0: LFM OFF.

Example: [:SOURce]:PULM:LFM:STATe 1 LFM is ON.

Reset status: 0

Key Entry: 【Signal】—>[LFM]—> [LFM ON OFF]

3.3.8 AMPLitude MODulation subsystem

The following commands are used for setting AM operation mode:

- [:SOURce]:AM[:DEPTh]:EXPonential 68
- [:SOURce]:AM[:DEPTh][:LINear] 68
- [:SOURce]:AM:INTernal:FREQuency 68
- [:SOURce]:AM:INTernal:FREQuency:ALTernate 69
- [:SOURce]:AM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent 69
- [:SOURce]:AM:INTernal:NOISe 70
- [:SOURce]:AM:INTernal:RAMP 70

- [:SOURce]:AM:INTernal:SHAPe 70
- [:SOURce]:AM:INTernal:SWEep:TIME 71
- [:SOURce]:AM:MODE 71
- [:SOURce]:AM:SOURce 71
- [:SOURce]:AM:STATe 72
- [:SOURce]:AM:TYPE 72

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

Function: When AM Type is EXP, set AM Depth of AM signal in dB. For AM type, see the command ":AM:TYPE".

Setting format: [:SOURce]:AM:DEPTh:EXPonential <.val>

Query format: [:SOURce]:AM: DEPTh:EXPonential?

Parameter:

<AmDepthExp> AM Depth (EXP).

Range: 0.00dB[0.00dB，40.00dB].

Example: :AM: DEPTh:EXPonential 10dB AM Depth is 10 dB.

Reset status: 30dB

Key Entry: 【AM】—>[Basic Config]—>[AM Depth]

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

Function: This command is used for setting AM depth in percent. The set value is valid only when AM type is set to Linear. See the command ":AM:TYPE".

Setting format: [:SOURce]:AM: DEPTh [:LINear] <.val>

Query format: [:SOURce]:AM: DEPTh [:LINear]?

Parameter:

<AmDepthExp> AM depth (linear)

Range: 30[0，100].

Example: :AM:DEPTh 10 This example indicates that the set linear AM depth is 10%

Reset status: 30

Key Entry: 【AM】—>[Basic Config]—>[AM Depth]

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

Function: This command is used for setting the internal modulation rate for AM in the signal generator. In addition, through this command, the first tone can be set when AM waveform is set to DualSinc, and the start frequency can be set when AM waveform is set to SweepSinc. For AM signal output waveform, see the command ":AM:INTernal:SHAPe".

Setting format: [:SOURce]:AM:INTernal:FREQuency <val>

Query format: [:SOURce]:AM:INTernal:FREQuency?

Parameter:

<Frequency> AM rate.

Range: 1kHz[5mHz，1MHz].

Freq Start of SweepSinc.

Range: 10mHz[10mHz,0.99999999MHz]

DualSinc frequency1.

Range: 1kHz[10mHz,1MHz].

Example: :AM:INTernal:FREQuency 100kHz Internal modulation rate for AM is 100kHz.

Reset status: AM rate: 1kHz.

Start frequency of sweep sine: 10mHz.

Frequency 1 of dual sine: 1kHz

Key Entry: 【AM】—>[Basic Config]—>[AM Rate]

[:SOURce]:AM:INTernal:FREQuency:ALTernate <Frequency>

Function: This command is used for setting the second tone when AM waveform is set to DualSinc, or the stop frequency when AM waveform is set to SweepSinc. For AM signal output waveform, see the command ":AM:INTernal:SHAPe".

Setting format: [:SOURce]:AM:INTernal:FREQuency:ALTernate <.val>

Query format: [:SOURce]:AM:INTernal:FREQuency:ALTernate?

Parameter:

<Frequency> Frequency2 when AM waveform is set to DualSinc, or the stop frequency when AM waveform is set to SweepSinc.

Range: Dual sine 1kHz[0.02Hz，1.000000000MHz].

Sweep sine 1kHz[0.02Hz，1.000000000MHz].

Example: :AM:INTernal:FREQuency:ALTernate 500kHz Set alternate frequency to 500kHz.

Key Entry: 【AM】—>[Sweep Sinc]—>[Freq Stop],

【AM】—>[Dual Sinc]—>[Frequency2].

[:SOURce]:AM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent<Pert>

Function: This command is used for setting the percentage of amplitude of second tone to that of total output signal in dual sine waveform when AM Waveform is set to DualSinc; for example, if the second tone accounts for 20% of the total waveform power, the first tone will account for 80% of the total power output.

Setting format: [:SOURce]:AM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent <val>

Query format: [:SOURce]:AM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent?

Parameter:

<Pert> Amplitude percentage of frequency 2 when AM Waveform is set to DualSinc.

Range: 50[0，100].

Example: :AM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent 20

Set the percentage of second waveform of dual sine to the total signal output power to 20%.

Reset status: 50

Key Entry: 【AM】—>[Dual Sinc]—> [Freq 2 Ampl Percent]

[:SOURce]:AM:INTernal:NOISe <Mode>

Function: This command is used for setting signal output types when AM Waveform is set to White noise, which includes: White noise and Gauss. For AM signal output waveform, see the command ":AM:INTernal:SHAPe".

Setting format: [:SOURce]:AM:INTernal:NOISe UNIForm|GAUSsian

Query format: [:SOURce]:AM:INTernal:NOISe?

Parameter:

<Mode> Discrete data. Options for signal output type when AM Waveform is set to White noise:

UNIForm White noise,

GAUSsian Gauss.

Example: :AM:INTernal:NOISe GAUS AM noise is set as Gauss.

Reset status: UNIF

Key Entry: 【AM】—>[Basic Config]—>[AM Waveform]

[:SOURce]:AM:INTernal:RAMP <Mode>

Function: This command is used for setting signal output types when AM Waveform is set to Zigzag, which includes: Zigzag-up and Zigzag-down. For AM signal output waveform, see the command ":AM:INTernal:SHAPe".

Setting format: [:SOURce]:AM:INTernal:RAMP POSitive|NEGative

Query format: [:SOURce]:AM:INTernal:RAMP?

Parameter:

<Mode> Discrete data. Options for signal output type when AM Waveform is set to Zigzag:

POSitive Zigzag-up,

NEGative Zigzag-down.

Example: :AM:INTernal:RAMP NEG Positive increase of AM Zigzag waveform signal.

Reset status: POS

Key Entry: 【AM】—>[Basic Config]—>[AM Waveform]

[:SOURce]:AM:INTernal:SHAPe <Mode>

Function: This command is used for setting AM signal output waveform, which includes: Sinc, Square, Triangle, Zigzag, White noise, SweepSinc, and DualSinc.

Setting format: [:SOURce]:AM:INTernal:SHAPe SINE|SQUare|TRIangle|RAMP NOISe|SWEPtsine|DUALsine

Query format: [:SOURce]:AM:INTernal:SHAPe?

Parameter:

<Mode> Discrete data. AM Waveform is as follows:

SINE Sine,

SQUare Square,

TRIangle Triangle,

RAMP Zigzag,

NOISe Noise,

SWEPtsine SweepSinc,

DUALsine DualSinc.

Example: :AM:INTernal:SHAP RAMP AM signal waveform is Zigzag.

Reset status: SINE

Key Entry: 【AM】—>[Basic Config]—>[AM Waveform]

[:SOURce]:AM:INTernal:SWEep:TIME <Time>

Function: This command is used for setting sweep time when AM Waveform is set to SweepSinc.

Setting format: [:SOURce]:AM:INTernal:SWEep:TIME <val>

Query format: [:SOURce]:AM:INTernal:SWEep:TIME?

Parameter:

<Time> Sweep time when AM Waveform is set to SweepSinc.

Range: 10.000ms [10.000us，2s].

Example: :AM:INTernal:SWEep:TIME 1s

Sweep time of sweep sine for AM signal is 1s.

Reset status: 10.000ms

Key Entry: 【AM】—>[Sweep Sinc]—>[Sweep Time]

[:SOURce]:AM:MODE <Mode>

Function: This command sets AM mode. In DEEP mode, AM Depth of the signal generator is dynamically greater than that when ALC Loop is turned off, and AM index is superior to the index in data manual; in NORMal mode, AM index is the same as that in data manual. Please refer to data index of 1465 series signal generator.

Setting format: [:SOURce]:AM:MODE DEEP|NORMal

Query format: [:SOURce]:AM:MODE?

Parameter:

<Mode> Discrete data. AM Mode is as follows:

DEEP AM Depth is turned on,

NORMal AM Depth in turned off.

Example: :AM:MODE NORM Depth AM OFF

Reset status: NORM

Key Entry: 【AM】—>[Basic Config]—>[AM Depth ON OFF]

[:SOURce]:AM:SOURce <Mode>

Function: This command is used for setting AM Sourcemodes, which include: Internal and External. When it is set to Internal, the external AM signal should be connected to the AM input interface on the front panel of the signal generator.

Setting format: [:SOURce]:AM:SOURce EXTernal|INTernal

Query format: [:SOURce]:AM:SOURce?

Parameter:

<Mode> Discrete data. Options for AM Source mode are:

EXTernal Internal,

INTernal External.

Example: :AM:SOURce INT AM Source is set to Internal.

Reset status: INT

Key Entry: 【AM】—>[AM Source]—>[AM Source]

[:SOURce]:AM:STATe <State>

Function: This command sets the AM signal output state of the signal generator.

Setting format: [:SOURce]:AM:STATe ON|OFF|1|0

Query format: [:SOURce]:AM:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: AM output is turned on,

OFF | 0: AM output is turned off.

Example: :AM:STATe 1 AM is turned on.

Reset status: 0

Key Entry: 【AM】—>[Basic Config]—>[AM ON OFF]

[:SOURce]:AM:TYPE <Mode>

Function: This command sets AM Type of the signal generator to EXP or Linear. AM Depth will be in dB in EXP mode; AM Depth will be in percent in Linear mode.

Setting format: [:SOURce]:AM:TYPE EXPonential|LINear

Query format: [:SOURce]:AM:TYPE?

Parameter:

<Mode> Discrete data. AM Type is as follows:

EXPonential EXP AM,

LINear Linear AM.

Example: :AM:TYPE EXP EXP AM.

Reset status: LIN

Key Entry: 【AM】—>[Basic Config]—>[AM Type EXP Linear]

3.3.9 FREQuency MODulation subsystem

The following commands are used for setting FM operation modes:

- [:SOURce]:FM:DEViation 73
- [:SOURce]:FM:INTernal:FREQuency 73
- [:SOURce]:FM:INTernal:FREQuency:ALTernate 74
- [:SOURce]:FM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent 74
- [:SOURce]:FM:INTernal:NOISe 74
- [:SOURce]:FM:INTernal:RAMP 75
- [:SOURce]:FM:INTernal:SHAPe 75
- [:SOURce]:FM:INTernal:SWEep:TIME 75
- [:SOURce]:FM:SOURce 76
- [:SOURce]:FM:STATe 76

[:SOURce]:FM:DEViation <Deviation>

Function: This command sets FM Dev of the signal generator. It should be noted that FM Dev varies in different frequency band.

Setting format: [:SOURce]:FM:DEViation <val>

Query format: [:SOURce]:FM:DEViation?

Parameter:

<Deviation> The relationship between current frequency and FM Dev is as follows:

1465

Current frequency	FM Dev
9kHz - 250MHz	0 - 2MHz
250MHz -500MHz	0 - 1MHz
500MHz - 1GHz	0 - 2MHz
1GHz - 2GHz	0 - 4MHz
2GHz-3.2GHz	0 - 8MHz
3.2GHz - 10GHz	0 - 16MHz
10GHz-20GHz	0 - 32MHz
20GHz-40GHz	0 - 64MHz
40GHz-67GHz	0 - 128MHz

Example: :FM:DEViation 500kHz FM Dev of FM signal is 500 kHz.

Key Entry: 【FM/ΦM】—>[Basic Config]—>[FM Dev]

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

Function: This command is used for setting the internal modulation rate for FM in the signal generator. Through this command, the first tone can be set when FM waveform is set to DualSinc, and the start frequency can be set when FM waveform is set to SweepSinc. It should be noted that, when FM Source is set to External, the internal modulation rate can not be set. For related commands, see ":FM:INTernal:SHAPe",":FM:SOURce".

Setting format: [:SOURce]:FM:INTernal:FREQuency <val>.

Query format: [:SOURce]:FM:INTernal:FREQuency?

Parameter:

<Frequency> The FM modulation rate ranges of different waveforms are:

Sine wave: [0.005Hz，10.000000000MHz],
Square wave: [0.005Hz，10.000000000MHz],
Triangular wave: [0.005Hz，10.000000000MHz],
Zigzagwave: [0.005Hz，10.000000000MHz],
Noise: [0.005Hz，10.000000000MHz],
Sweep sine: [0.01Hz，10.000000000MHz],
Dual sine: [0.01Hz，10.000000000MHz].

Example: :FM:INTernal:FREQuency 300kHz FM signal modulation rate is 300kHz.

Reset status: 0.001MHz

Key Entry: 【FM/ΦM】—>[Basic Config]—>[FM Rate]

【FM/ΦM】—>[SweepSinc]—> [Freq Start]

【FM/ΦM】—>[Dual Sinc]—>[Frequency1]

[:SOURce]:FM:INTernal:FREQuency:ALTernate <Frequency>

Function: This command is used for setting the second tone when FM waveform is set to DualSinc, or the stop frequency when FM waveform is set to SweepSinc. For FM waveform, see the command ":FM:INTernal:SHAPe".

Setting format: [:SOURce]:FM:INTernal:FREQuency:ALTernate <.val>

Query format: [:SOURce]:FM:INTernal:FREQuency:ALTernate?

Parameter:

<Frequency> The frequency ranges of different FM waveforms are:

Sweep sine: [0.02Hz，1.000000000MHz],

Dual sine: [0.02Hz，1.000000000MHz],

Example: :FM:INTernal:FREQuency:ALTernate 500kHz

Set alternate frequency to 500kHz.

Key Entry: 【FM/ΦM】—>[SweepSinc]—> [Freq Stop]

【FM/ΦM】—>[Dual Sinc]—>[Frequency2]

[:SOURce]:FM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent<Pert>

Function: This command is used for setting the percentage of amplitude of second tone to that of total output signal in dual sine waveform when FM Waveform is set to DualSinc. For example, if the second tone accounts for 20% of the total waveform power, the first tone will account for 80% of the total power output.

Setting format: [:SOURce]:FM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent <val>

Query format: [:SOURce]:FM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent?

Parameter:

<Pert> Amplitude percentage of frequency 2 when FM Waveform is set to DualSinc.

Range: 50[0，100].

Example: :FM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent 20

Set the percentage of second waveform of dual sine to the total signal output power to 20%.

Reset status: 50

Key Entry: 【FM/ΦM】—>[Dual Sinc]—> [Freq 2 Ampl Percent]

[:SOURce]:FM:INTernal:NOISe <Mode>

Function: This command is used for setting signal output types when FM Waveform is set to White noise, which includes: While noise and Gauss. For setting FM Waveform, see the command ":FM:INTernal:SHAPe".

Setting format: [:SOURce]:FM:INTernal:NOISe UNIForm|GAUSsian

Query format: [:SOURce]:FM:INTernal:NOISe?

Parameter:

<Mode> Discrete data. Options for signal output type when FM Waveform is set to White noise:

UNIForm White noise,

GAUSsian Gauss.

Example: :FM:INTernal:NOISe GAUS FM noise is set as Gauss.

Reset status: UNIF

Key Entry: 【FM/ΦM】—>[Basic Config]—>[FM Waveform]—>[Noise]

[:SOURce]:FM:INTernal:RAMP <Mode>

Function: This command is used for setting signal output types when FM Waveform is set to Zigzag, which includes: Zigzag-up and Zigzag-down. For FM signal output waveform, see the command ":FM:INTernal:SHAPe".

Setting format: [:SOURce]:FM:INTernal:RAMP POSitive|NEGative

Query format: [:SOURce]:FM:INTernal:RAMP?

Parameter:

<Mode> Discrete data. Options for signal output type when FM Waveform is set to Zigzag:

POSitive Zigzag-up,

NEGative Zigzag-down.

Example: :FM:INTernal:RAMP NEG FM Zigzag waveform is set as Zigzag-up.

Reset status: POS

Key Entry: 【FM/ΦM】—>[Basic Config]—>[FM Waveform]—>[Zigzag]

[:SOURce]:FM:INTernal:SHAPe <Mode>

Function: This command is used for setting FM signal output waveform, which includes: Sinc, Square, Triangle, Zigzag, White noise, SweepSinc, and DualSinc.

Setting format: [:SOURce]:FM:INTernal:SHAPe SINE|SQUare|TRIangle|RAMP |NOISe|SWEPtsine|DUALsine

Query format: [:SOURce]:FM:INTernal:SHAPe?

Parameter:

<Mode> Discrete data. Options for FM output waveform are:

SINE	Sine,
SQUare	Square,
TRIangle	Triangle,
RAMP	Zigzag,
NOISe	Noise,
SWEPtsine	SweepSinc,
DUALsine	DualSinc.

Example: [:SOURce]:FM:INTernal:SHAP RAMP FM waveform is zigzag.

Reset status: SINE

Key Entry: 【FM/ΦM】—>[Basic Config]—>[FM Waveform]

[:SOURce]:FM:INTernal:SWEep:TIME <Time>

Function: This command is used for setting sweep time when FM Waveform is set to SweepSinc.

Setting format: [:SOURce]:FM:INTernal:SWEep:TIME <val>

Query format: [:SOURce]:FM:INTernal:SWEep:TIME?

Parameter:

<Time> Sweep time when FM Waveform is set to SweepSinc.

Range: 10.000ms [10.000us，2s].

Example: :FM:INTernal:SWEep:TIME 1s

Sweep time of sweep sine for FM signal is 1s.

Reset status: 10.000us

Key Entry: 【FM/ΦM】—>[Sweep Sinc]—>[Sweep Time]

[:SOURce]:FM:SOURce <Mode>

Function: This command is used for setting FM Sourcemodes, which include: Internal and External. When it is set to External, the external FM signal should be connected to the FM input interface on the front panel of the signal generator.

Setting format: [:SOURce]:FM:SOURce EXTernal|INTernal

Query format: [:SOURce]:FM:SOURce?

Parameter:

<Mode> Discrete data. Options for FM Source mode are:

EXTernal Internal,

INTernal External.

Example: :FM:SOURce INT FM Source is set to Internal.

Reset status: INT

Key Entry: 【FM/ΦM】—>[FM Source]

[:SOURce]:FM:STATe <State>

Function: This command sets the FM signal output state of the signal generator.

Setting format: [:SOURce]:FM:STATe ON|OFF|1|0

Query format: [:SOURce]:FM:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: FM output in turned on,

OFF | 0: FM output in turned off.

Example: :FM:STATe 0 FM OFF.

Reset status: 0

Key Entry: 【FM/ΦM】—>[Basic Config]—>[FM ON OFF]

3.3.10 PHASe MODulation subsystem

The following commands are used for setting PM operation modes:

- [:SOURce]:PM:BANDwidth|BWIDth 77
- [:SOURce]:PM:DEViation 77
- [:SOURce]:PM:INTernal:FREQuency 77
- [:SOURce]:PM:INTernal:FREQuency:ALTernate 78
- [:SOURce]:PM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent 78
- [:SOURce]:PM:INTernal:NOISe 79
- [:SOURce]:PM:INTernal:RAMP 79

- [:SOURce]:PM:INTernal:SHAPe 79
- [:SOURce]:PM:INTernal:SWEep:TIME 80
- [:SOURce]:PM:SOURce 80
- [:SOURce]:PM:STATe 80

[:SOURce]:PM:BANDwidth|BWIDth

Function: This command is used for setting PM signal bandwidth of the signal generator. In NORMal mode, PM bandwidth is normal; in HIGH mode, it is from broadband.

Setting format: [:SOURce]:PM:BANDwidth|BWIDth NORMal|HIGH

Query format: [:SOURce]:PM:BANDwidth|BWIDth?

Parameter:

<Mode> Discrete data. Options for PM bandwidth type are:

NORMal PM bandwidth, normal,

HIGH PM bandwidth, broadband.

Example: :PM:BWIDth NORM PM bandwidth is from broadband.

Reset status: NORM

Key Entry: 【FM/ΦM】—>[FM/PM]—>[PM BandWidth Normal HighBand]

[:SOURce]:PM:DEViation <Deviation>

Function: This command sets PM Bias of the signal generator. It should be noted that PM Bias varies in different frequency band.

Setting format: [:SOURce]:PM:DEViation <val>

Query format: [:SOURce]:PM:DEViation?

Parameter:

<Deviation> The relationship between PM Bias and PM BandWidth is as follow:

1465

Current frequency	PM BandWidth Normal	PM BandWidth HighBand
100kHz - 250MHz	0 - 2.000rad	0 – 0.200rad
250MHz-500MHz	0 - 1.000rad	0 – 0.100rad
500MHz - 1GHz	0 - 2.000rad	0 – 0.200rad
1GHz - 2GHz	0 - 4.000rad	0 – 0.400rad
2GHz-3.2GHz	0 - 8.000rad	0 – 0.400rad
3.2GHz - 10GHz	0 - 16.000rad	0 – 1.600rad
10GHz-20GHz	0 - 32.000rad	0 – 3.200rad
20GHz-40GHz	0 – 64.000rad	0 – 6.400rad
40GHz-67GHz	0 - 128.000rad	0 – 12.800rad

Example: :PM:DEViation 3rad PM deviation of the PM signal is 3rad.

Reset status: 0.001rad

Key Entry: 【FM/ΦM】—>[Basic Config]—>[PM Bias]

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

Function: This command is used for setting the internal modulation rate of PM in the signal generator. Through this command, the first tone can be set when PM waveform is set

to DualSinc, and the start frequency can be set when FM waveform is set to SweepSinc. It should be noted that, when PM Source is set to External, the internal modulation rate can not be set. For related commands, see ":PM:INTernal:SHAPe","PM:SOURce".

Setting format: [:SOURce]:PM:INTernal:FREQuency <val>

Query format: [:SOURce]:PM:INTernal:FREQuency?

Parameter:

<Frequency> The PM modulation rate ranges of different waveforms are:

Sine wave:	[0.005Hz，10.000000000MHz],
Square wave:	[0.005Hz，10.000000000MHz],
Triangular wave:	[0.005Hz，10.000000000MHz],
Zigzagwave:	[0.005Hz，10.000000000MHz],
Noise:	[0.005Hz，10.000000000MHz],
Sweep sine:	[0.01Hz，10.000000000MHz],
Dual sine:	[0.01Hz，10.000000000MHz].

Example: :PM:INTernal:FREQuency 300kHz

PM modulation rate is 300kHz.

Reset status: 0.001MHz

Key Entry: 【FM/ΦM】—>[Basic Config]—>[FM Rate]

[:SOURce]:PM:INTernal:FREQuency:ALTernate <Frequency>

Function: This command is used for setting the second tone when PM waveform is set to DualSinc, or the stop frequency when PM waveform is set to SweepSinc. For PM waveform, see the command ":AM:INTernal:SHAPe".

Setting format: [:SOURce]:PM:INTernal:FREQuency:ALTernate <.val>

Query format: [:SOURce]:PM:INTernal:FREQuency:ALTernate?

Parameter:

<Frequency> The frequency ranges of different PM waveforms are:

Sweep sine: [0.02Hz，1.000000000MHz],

Dual sine: [0.02Hz，1.000000000MHz],

Example: :PM:INTernal:FREQuency:ALTernate 500kHz

Alternate frequency is 500kHz.

Key Entry: 【FM/ΦM】—>[Sweep Sinc]—> [Freq Stop]

【FM/ΦM】—>[Dual Sinc]—>[Frequency2]

[:SOURce]:PM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent <Pert>

Function: This command is used for setting the percentage of amplitude of second tone to that of total output signal in dual sine waveform when PM Waveform is set to DualSinc. For example, if the second tone accounts for 20% of the total waveform power, the first tone will account for 80% of the total power output.

Setting format: [:SOURce]:PM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent <val>

Query format: [:SOURce]:PM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent?

Parameter:

<Pert> Amplitude percentage of frequency 2 when PM Waveform is set to DualSinc.
Range: 50[0，100].

Example: :PM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent Set the percentage of second waveform of dual sine to the total signal output power to 20%..

Reset status: 50

Key Entry: 【FM/ΦM】—>[Dual Sinc]—> [Freq 2 Ampl Percent]

[:SOURce]:PM:INTernal:NOISe <Mode>

Function: This command is used for setting signal output types when PM Waveform is set to White noise, which includes: White noise and Gauss. For PM waveform, see the command ":AM:INTernal:SHAPe".

Setting format: [:SOURce]:PM:INTernal:NOISe UNIForm|GAUSsian

Query format: [:SOURce]:PM:INTernal:NOISe?

Parameter:

<Mode> Discrete data. Options for signal output type when FM Waveform is set to White noise:
UNIForm White noise,
GAUSsian Gauss.

Example: :PM:INTernal:NOISe GAUS PM noise is set to Gauss.

Reset status: UNIF

Key Entry: 【FM/ΦM】—>[Basic Config]—>[PM Waveform]

[:SOURce]:PM:INTernal:RAMP <Mode>

Function: This command is used for setting signal output types when PM Waveform is set to Zigzag, which includes: Zigzag-up and Zigzag-down. For PM waveform, see the command ":AM:INTernal:SHAPe".

Setting format: [:SOURce]:PM:INTernal:RAMP POSitive|NEGative

Query format: [:SOURce]:PM:INTernal:RAMP?

Parameter:

<Mode> Discrete data. Options for signal output type when PM Waveform is set to Zigzag:
POSitive Zigzag-up,
NEGative Zigzag-down.

Example: :PM:INTernal:RAMP NEG PM Zigzag is set to Zigzag-up.

Reset status: POS

Key Entry: 【FM/ΦM】—>[Basic Config]—>[PM Waveform]

[:SOURce]:PM:INTernal:SHAPe <Mode>

Function: This command is used for setting PM signal output waveform, which includes: Sinc, Square, Triangle, Zigzag, White noise, SweepSinc, and DualSinc.

Setting format: [:SOURce]:PM:INTernal:SHAPe SINE|SQUare|TRIangle|RAMP NOISe|SWEPtsine|DUALsine

Query format: [:SOURce]:PM:INTernal:SHAPe?

Parameter:

<Mode> Discrete data. Options for PM output waveform are:

SINE	Sine,
SQUare	Square,
TRIangle	Triangle,
RAMP	Zigzag,
NOISe	Noise,
SWEPtsine	SweepSinc,
DUALsine	DualSinc.

Example: [:SOURce]:FM:INTernal:SHAP RAMP PM waveform is zigzag.

Reset status: SINE

Key Entry: 【FM/ΦM】—>[Basic Config]—>[PM Waveform]

[:SOURce]:PM:INTernal:SWEep:TIME <Time>

Function: This command is used for setting sweep time when FM Waveform is set to SweepSinc.

Setting format: [:SOURce]:PM:INTernal:SWEep:TIME <val>

Query format: [:SOURce]:PM:INTernal:SWEep:TIME?

Parameter:

<Time> Sweep time when PM Waveform is set to SweepSinc.

Range: 10.000ms [10.000us，2s].

Example: :PM:INTernal:SWEep:TIME 1s

Sweep time of sweep sine for PM signal is 1s.

Reset status: 10.000us

Key Entry: 【FM/ΦM】—>[Sweep Sinc]—>[Sweep Time]

[:SOURce]:PM:SOURce <Mode>

Function: This command is used for setting PM Sourcemodes, which include: Internal and External. When it is set to External, the external PM signal should be connected to the FM/PM input interface on the front panel of the signal generator.

Setting format: [:SOURce]:PM:SOURce EXTernal|INTernal'

Query format: [:SOURce]:PM:SOURce?

Parameter:

<Mode> Discrete data. Options for PM Source mode are:

EXTerna	External,
INTernal	Internal.

Example: :PM:SOURce INT PM Source is set to Internal.

Reset: INT

Key Entry: 【FM/ΦM】—>[PM Source]—>[PM Source]

[:SOURce]:PM:STATe <State>

Function: This command sets the PM signal output state of the signal generator.

Setting format: [:SOURce]:PM:STATe ON|OFF|1|0

Query format: [:SOURce]:PM:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: PM output is turned on,

OFF | 0: PM output is turned off.

Example: :PM:STATe 0 PM OFF.

Reset status: 0

Key Entry: 【FM/ΦM】—>[Basic Config]—>[PM ON OFF]

3.3.11 Digital MODulation subsystem

The following commands are used for setting DM operation modes:

- [:SOURce]:DM:IQADjustment:GAIN 82
- [:SOURce]:DM:IQADjustment:IOFFset 83
- [:SOURce]:DM:IQADjustment:QOFFset 83
- [:SOURce]:DM:IQADjustment:QSKew 83
- [:SOURce]:DM:IQADjustment[:STATe] 84
- [:SOURce]:DM:MODulation:ATTenuation 84
- [:SOURce]:DM:MODulation:ATTenuation:AUTO 85
- [:SOURce]:DM:STATe 85
- [:SOURce]:DM:EXTernal:BWIDth[:STATe] 86
- [:SOURce]:DM:IQADjustment:OUTPut[:STATe] 86
- [:SOURce]:DM:IQADjustment:OUTPut:ATTen 86
- [:SOURce]:DM:IQADjustment:OUTPut:GAIN 86
- [:SOURce]:DM:IQADjustment:OUTPut:IOFFset 87
- [:SOURce]:DM:IQADjustment:OUTPut:UIOFfset 87
- [:SOURce]:DM:IQADjustment:OUTPut:QOFFset 87
- [:SOURce]:DM:IQADjustment:OUTPut:UQOFfset 88
- [:SOURce]:DM:IQADjustment:OUTPut:SKEW 88
- [:SOURce]:RADio:CUSTom:ALPHa 88
- [:SOURce]:RADio:CUSTom:DATA 89
- [:SOURce]:RADio:CUSTom:DATA:PRAM 89
- [:SOURce]:RADio:CUSTom:FILTer 89
- [:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] 90
- [:SOURce]:RADio:CUSTom:MODulation:MSK:PHASe 90
- [:SOURce]:RADio:CUSTom:MODulation[:TYPE] 90
- [:SOURce]:RADio:CUSTom:SRATe 91
- [:SOURce]:RADio:CUSTom:STATe 91
- [:SOURce]:RADio:CUSTom:POLarity[:ALL] 92
- [:SOURce]:RADio:CUSTom:DENCode 92
- [:SOURce]:RADio:CUSTom:VCO:CLOCk 92
- [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay 93
- [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay:STATe 93

- [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:SLOPe 93
- [:SOURce]:RADio:CUSTom:TRIGger:SOURce 94
- [:SOURce]:RADio:CUSTom:TRIGger:TYPE 94
- [:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous:TYPE 95
- [:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive 95
- [:SOURce]:RADio:MTONe:ARB:SETup 95
- [:SOURce]:RADio:MTONe:ARB:SETup:STORe 96
- [:SOURce]:RADio:MTONe:ARB: SETup:TABLe 96
- [:SOURce]:RADio:MTONe:ARB: SETup:TABLe:FSPacing 97
- [:SOURce]:RADio:MTONe:ARB: SETup:TABLe:NTONes 97
- [:SOURce]:RADio:MTONe:ARB: SETup:TABLe:PHASe:INITialize 98
- [:SOURce]:RADio:MTONe:ARB: SETup:TABLe:PHASeINITialize:SEED 98
- [:SOURce]:RADio:MTONe:ARB: SETup:TABLe:ROW 98
- [:SOURce]:RADio:MTONe:ARB[:STATe] 99
- [:SOURce]:RADio:TTONe:ARB:ALIGnment 99
- [:SOURce]:RADio:TTONe:ARB:FSPacing 100
- [:SOURce]:RADio:TTONe:ARB[:STATe] 100
- [:SOURce]:RADio:ARB:MODE 100
- [:SOURce]:RADio:ARB[:STATe] 101
- [:SOURce]:RADio:ARB:SEQuence 101
- [:SOURce]:RADio:ARB:SEQuence:CLOCk 102
- [:SOURce]:RADio:ARB:SCLock:RATE 102
- [:SOURce]:RADio:ARB:TRIGger:TYPE 103
- [:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous[:TYPE] 104
- [:SOURce]:RADio:ARB:TRIGger:TYPE:SINGle 104
- [:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance[:TYPE] 105
- [:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive 106
- [:SOURce]:RADio:ARB:TRIGger:SOURce 106
- [:SOURce]:RADio:ARB:VCO:CLOCk 107
- [:SOURce]:RADio:ARB:EXTernal:CLOCk:RATE 107

[:SOURce]:DM:IQADjustment:GAIN <Gain>

Function: When I/Q Adjust is turned on, set the gain of the signal generator I signal relative to Q signal. For enabling or disabling I/Q adjustment function, see the command ":DM:IQADjustment[:STATe]".

Setting format: [:SOURce]:DM:IQADjustment:GAIN <val>

Query format: [:SOURce]:DM:IQADjustment:GAIN?

Parameter:

<Gain> I/Q Signal Gain Balance.

Range: 0dB[-4.00dB，+4.00dB].

Example: :DM:IQADjustment:GAIN 0dB

Set Gain Balance of I/Q signal to 0dB.

Reset status: 0dB

Key Entry: 【I/Q】—>[I/Q Input Adj]—>[Gain Balance]

[:SOURce]:DM:IQADjustment:IOFFset <Offset>

Function: When I/Q Adjust is turned on, set the offset of the signal generator I channel. The set parameters are in percent. The maximum corresponds to 1.5 V DC and the minimum resolution is 0.025%. The parameter completes the suppression of carrier leakage signal. Carrier leakage increases after users complete other adjustments, such as orthority and modulation attenuation. It is necessary to adjust DC offset after other adjustments are completed.

Setting format: [:SOURce]:DM:IQADjustment:IOFFset <val>

Query format: [:SOURce]:DM:IQADjustment:IOFFset?

Parameter:

<Offset> I/Q signal I offset.

Range: 0 [-50，+50].

Example: :DM:IQADjustment:IOFfset 30

Set I Offset to 30%.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Input Adj]—>[I Offset]

[:SOURce]:DM:IQADjustment:QOFFset <Offset>

Function: This command is used for setting Q channel offset of the signal generator in percent. The maximum value corresponds to DC 1.5V. The minimum resolution is 0.025%. The parameter completes the suppression of carrier leakage signal. Carrier leakage increases after users complete other adjustments, such as orthority and modulation attenuation. It is necessary to adjust DC offset after other adjustments are completed.

Setting format: [:SOURce]:DM:IQADjustment:QOFFset <val>

Query format: [:SOURce]:DM:IQADjustment:QOFFset?

Parameter:

<Offset> I/Q signal Q offset.

Range: 0 [-50，+50].

Example: :DM:IQADjustment:QOFFset 30

Set Q Offset to 30%.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Input Adj]—>[Q Offset]

[:SOURce]:DM:IQADjustment:QSKew <Offset>

Function: When I/Q Adjust is turned on, this command adjusts the phase angle between I and Q vectors by increasing or decreasing the phase angle of I or Q. If the current carrier frequency exceeds 3.2 GHz, the error of orthority offset may exceed that specified in the sample specification of 1465 series signal generator.

Setting format: [:SOURce]:DM:IQADjustment:QSKew <val>

Query format: [:SOURce]:DM:IQADjustment:QSKew?

Parameter:

<Offset> I/Q adjustment orthority offset.

Range: 0deg [-10.00deg，+10.00deg].

Example: :DM:IQADjustment:QSKew 30deg

Set the Orthority Offset for I/Q adjustment to 30deg.

Reset status: 0deg

Key Entry: 【I/Q】—>[I/Q Input Adj]—>[Orthority Offset]

[:SOURce]:DM:IQADjustment[:STATe] <State>

Function: This command enables I/Q Adjust ON/OFF. After this function is turned on, the parameters adjusted: Gain Balance, I Offset, Q Offset and Orthority Offset are superimposed on adjustment circuit; after this function is turned off, the above parameter values are not used, but the modulation attenuation is not affected by the ON/OFF state of I/Q Adjust. For related commands, see ":DM:MODulation:ATTenuation","[:DM:IQADjustment:EXTernal:IQATten](#)".

Setting format: [:SOURce]:DM:IQADjustment[:STATe] ON|OFF|1|0

Query format: [:SOURce]:DM:IQADjustment[:STATe] ?

Parameter:

<State> Boolean data is as follows:

ON | 1: I/Q Adjust is turned on,

OFF | 0: I/Q Adjust is turned off.

Example: :DM:IQADjustment 1 Turn on I/Q Adjust.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Input Adj]—>[I/Q Adjust ON OFF]

[:SOURce]:DM:MODulation:ATTenuation <Atten>

Function: This command sets the attenuation of I/Q signal modulated by the signal generator RF channel. The output of the output attenuation can be set when Attenuation is in Manual state. Attenuation is valid even if I/Q Adjust is turned off. For related commands, see ":DM:MODulation:ATTenuation:AUTO","[:DM:IQADjustment[:STATe]](#)".

Setting format: [:SOURce]:DM:MODulation:ATTenuation <val>

Query format: [:SOURce]:DM:MODulation:ATTenuation?

Parameter:

<Atten> I/Q Modulation Attenuation.

Range: 12.00dB[0.00dB，40.00dB].

Example: :DM:MODulation:ATTenuation 10dB Attenuation of I/Q modulator is 10dB.

Reset status: 0dB

Key Entry: 【I/Q】—>[Attenuation]—>[Attenuation]

[:SOURce]:DM:MODulation:ATTenuation:AUTO <State>

Function: This command sets the Manual state of the signal generator I/Q channel attenuator. When Manual mode is enabled, current attenuation remains unchanged. For modulator attenuation, see the command ":DM:MODulation:ATTenuation"; after Manual mode is disabled, the user can not change attenuation, and the signal generator will select the optimum attenuation value for its current status automatically.

Setting format: [:SOURce]:DM:MODulation:ATTenuation:AUTO ON|OFF|1|0

Query format: [:SOURce]:DM:MODulation:ATTenuation:AUTO?

Parameter:

<State> Boolean data is as follows:

ON | 1: Modulation Attenuation is in Manual state,

OFF | 0: Modulation Attenuation is in Auto state.

Example: :DM:MODulation:ATTenuation:AUTO 1

Modulation Attenuation is in Manual mode.

Reset status: 0

Key Entry: 【I/Q】—>[Attenuation]->[Modulation Attenuation Manual Auto]

[:SOURce]:DM:STATe <State>

Function: This command enables the internal I/Q modulator to ON/OFF state.

Setting format: [:SOURce]:DM:STATe ON|OFF|1|0

Query format: [:SOURce]:DM:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: I/Q Modulate is turned on,

OFF | 0: I/Q Modulate is turned off.

Example: :DM:STATe 1 Turn on I/Q modulator.

Reset status: 0

Key Entry: 【I/Q】—>[Base Config]->[I/Q Modulate ON OFF]

[:SOURce]:DM:SOURce <Mode>

Function: This command selects I/Q Data Source entering IQ modulator of the signal generator. There are two modes for selection, EXTernal and INTernal.

Setting format: [:SOURce]:DM:SOURce EXTernal| INTernal

Query format: [:SOURce]:DM:SOURce?

Parameter:

<Mode> Discrete data. I/Q filter is selected as follows in Manual mode:

EXTernal | 0: I/Q signal input matching EXT50Ω,

INTernal | 1: Internal I/Q signal input to I/Q modulator,

Example: [:SOURce]:DM:SOURce EXT Set I/Q Data Source as EXT.

Reset status: EXT

Key Entry: 【I/Q】—>[Base Config]->[Data Source]

[:SOURce]:DM:EXTernal:BWIDth[:STATe] <State>

Function: This command sets the ON/OFF state of Ext WideBand I/Q Input.
Setting format: [:SOURce]:DM:EXTernal:BWIDth[:STATe] ON|OFF|1|0
Query format: [:SOURce]:DM:EXTernal:BWIDth[:STATe]?
Parameter:
<State> Boolean data is as follows:
ON | 1: Ext WideBand I/Q Input is turned on,
OFF | 0: Ext WideBand I/Q Input is turned off.
Example: :DM:EXT:BWID:STATe 1 Ext WideBand I/Q Input is turned on.
Reset status: 0
Key Entry: 【I/Q】—>[Base Config]->[Ext WideBand I/Q Input ON OFF]

[:SOURce]:DM:IQADjustmentOUTPut[:STATe] <State>

Function: This command sets I/Q Output Adj to ON/OFF state.
Setting format: [:SOURce]:DM:IQADjustment:OUTPut[:STATe] ON|OFF|1|0
Query format: [:SOURce]:DM:IQADjustment:OUTPut [:STATe]?
Parameter:
<State> Boolean data is as follows:
ON | 1: I/Q Output Adj is turned on,
OFF | 0: I/Q Output Adj is turned off.
Example: DM:IQADjustment:OUTPut 1 I/Q Output Adj is turned on.
Reset status: 0
Key Entry: 【I/Q】—>[I/Q Output Adj]—>[I/Q Output Adj ON OFF]

[:SOURce]:DM:IQADjustment:OUTPut:ATTen <Atten >

Function: This command is used for setting the attenuation value for I/Q output adjustment. This command will be valid when I/Q Output Adj is set to ON. For enabling or disabling I/Q output adjustment function, see the command "DM:IQADjustment:OUTPut".
Setting format: [:SOURce]:DM:IQADjustment:OUTPut:ATTen <val>.
Query format: [:SOURce] :DM:IQADjustment:OUTPut:ATTen?
Parameter:
<Atten> Attenuation of I/Q Output Adj.
Range: 0dB [0dB，94.5dB].
Example: :DM:IQADjustment:OUTPut:ATTen 10dB Set I/Q output attenuation value to 10dB.
Reset status: 0
Key Entry: 【I/Q】—>[I/Q Output Adj]—>[Attenuation]

[:SOURce]:DM:IQADjustment:OUTPut:GAIN <Gain >

Function: This command is used for setting Gain Balance of I/Q output adjustment. This command will be valid when I/Q Output Adj is set to ON. For enabling or disabling I/Q Output adjustment function, see the command "DM:IQADjustment:OUTPut".
Setting format: [:SOURce]:DM:IQADjustment:OUTPut:GAIN <val>.

Query format: [:SOURce] :DM:IQADjustment:OUTPut:GAIN?

Parameter:

<Gain> Gain Balance of I/Q Output Adj.

Range: 0dB [-4dB，4dB].

Example: :DM:IQADjustment:OUTPut:GAIN 2dB Set Gain Balance of I/Q Output Adj to 2 dB.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Output Adj]—>[Gain Balance]

[:SOURce]:DM:IQADjustment:OUTPut:IOFFset <offset >

Function: This command is used for setting I offset for I/Q output adjustment. This command will be valid when I/Q Output Adj is set to ON. For enabling or disabling I/Q output adjustment function, see the command "DM:IQADjustment:OUTPut".

Setting format: [:SOURce]:DM:IQADjustment:OUTPut:IOFFset <val>.

Query format: [:SOURce] :DM:IQADjustment:OUTPut: IOFFset?

Parameter:

<offset> I Offset of I/Q Output Adj.

Range: 0V [-1V，1V].

Example: :DM:IQADjustment:OUTPut:IOFFset 1V Set I offset for I/Q output to 1 V.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Output Adj]—>[I Offset]

[:SOURce]:DM:IQADjustment:OUTPut:UIOFfset <offset >

Function: This command is used for setting I/ offset for I/Q output adjustment. This command will be valid when I/Q Output Adj is set to ON. For enabling or disabling I/Q output adjustment function, see the command "DM:IQADjustment:OUTPut".

Setting format: [:SOURce]:DM:IQADjustment:OUTPut:UIOFfset <val>.

Query format: [:SOURce] :DM:IQADjustment:OUTPut:UIOFfset?

Parameter:

<offset> I/ Offset of I/Q Output Adj.

Range: 0V [-1V，1V].

Example: :DM:IQADjustment:OUTPut:UIOFfset 1V Set I/ Offset for I/Q output to 1V.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Output Adj]—>[I/ Offset]

[:SOURce]:DM:IQADjustment:OUTPut:QOFFset <offset >

Function: This command is used for setting Q offset for I/Q output adjustment. This command will be valid when I/Q Output Adj is set to ON. For enabling or disabling I/Q output adjustment function, see the command "DM:IQADjustment:OUTPut".

Setting format: [:SOURce]:DM:IQADjustment:OUTPut:QOFFset <val>.

Query format: [:SOURce] :DM:IQADjustment:OUTPut:QOFFset?

Parameter:

<offset> Q Offset of I/Q Output Adj.

Range: 0V [-1V，1V].

Example: :DM:IQADjustment:OUTPut:QOFFset 1V Set Q Offset for I/Q output to 1V.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Output Adj]—>[Q Offset]

[:SOURce]:DM:IQADjustment:OUTPut:UQOFfset <offset >

Function: This command is used for setting Q/ offset for I/Q output adjustment. This command will be valid when I/Q Output Adj is set to ON. For enabling or disabling I/Q Output adjustment function, see the command "DM:IQADjustment:OUTPut".

Setting format: [:SOURce]:DM:IQADjustment:OUTPut:UQOFfset <val>.

Query format: [:SOURce] :DM:IQADjustment:OUTPut:UQOFfset?

Parameter:

<offset> Q/ Offset of I/Q Output Adj.

Range: 0V [-1V，1V].

Example: :DM:IQADjustment:OUTPut:UQOFfset 1V Set I/ Offset for I/Q output to 1V.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Output Adj]—>[Q/ Offset]

[:SOURce]:DM:IQADjustment:OUTPut:SKEW <skew >

Function: This command is used for setting Orthority Offset for I/Q output adjustment. This command will be valid when I/Q Output Adj is set to ON. For enabling or disabling I/Q Output adjustment function, see the command "DM:IQADjustment:OUTPut".

Setting format: [:SOURce]:DM:IQADjustment:OUTPut:SKEW <val>.

Query format: [:SOURce] :DM:IQADjustment:OUTPut:SKEW?

Parameter:

<skew> Orthority Offset of I/Q Output Adj.

Range: 0V [-10deg，10deg].

Example: :DM:IQADjustment:OUTPut:SKEW 1deg Set Orthority Offset for I/Q output to 1deg.

Reset status: 0

Key Entry: 【I/Q】—>[I/Q Output Adj]—>[Orthority Offset]

[:SOURce]:RADio:CUSTom:ALPHa <FilterAlpha>

Function: This command is used for setting the alpha values of Nyquist filter, Root Nyquist filter and Gauss filter. If the user changes these values, the occupied bandwidth of the spectrum of baseband signal will be changed. For changing filter type, see the command ":RADio:CUSTom:FILTer".

Setting format: [:SOURce]:RADio:CUSTom:ALPHa <val>.

Query format: [:SOURce]:RADio:CUSTom:ALPHa?

Parameter:

<FilterAlpha> Filter factor.

Range: 0.350 [0，1.000].

Example: :RADio:CUSTom:ALPHa 0.350 Set filter factor to 0.35.

Reset status: 0.350

Key Entry: 【Base】—>[Filter]—>[Filter Factor α]

[:SOURce]:RADio:CUSTom:DATA <Mode>

Function: This command is used for setting the data sources of the baseband modulation signal, which include PN9, PN11, PN15, PN16, PN20, PN21, PN23, FIX4, P4, P8, P16, P32, P64 and PRAM.

Setting format: [:SOURce]:RADio:CUSTom:DATA PN9|PN11|PN15 |PN16|PN20|PN21|PN23|FIX4|P4|P8|P16|P32|P64PRAM

Query format: [:SOURce]:RADio:CUSTom:DATA?

Parameter:

<Mode> Discrete data. For data source of baseband modulation signal, please refer to the format of set command.

Example: :RADio:CUSTom:DATA FIX4

The baseband data source is set to Fix 4.

Reset status: PN9

Key Entry: 【Base】—>[Base Config]->[Data Source]

[:SOURce]:RADio:CUSTom:DATA:PARM <S>

Function: When the data source of real-time baseband of the signal generator is set to File, this command is used for selecting the data source code file, which must be stored under D:\1465data\user\DataSrc, with filename extension of ".src". This command is used for setting only.

Setting format: [:SOURce]:RADio:CUSTom:DATA

Parameter:

<S > Name of selected data source code file, including filename extension.

Example: :RADio:CUSTom:DATA:PRAM "Test.src"

Selected data source code file is "Test.src".

Key Entry: 【Base】—>[Base Config]->[Data Source File]-[Select File]

[:SOURce]:RADio:CUSTom:FILTer <Mode>

Function: This command selects the baseband pre-modulation filter types of the signal generator, including: RNYQuist, NYQuist, GAUSsian and RECTangle. RECTangle is applicable to digital FM signal, such as FSK and MSK. See the command ":RADio:CUSTom:MODulation[:TYPE]".

Setting format: [:SOURce]:RADio:CUSTom:FILTer RNYQuist|NYQuist |GAUSsian| RECTangle

Query format: [:SOURce]:RADio:CUSTom:FILTer?

Parameter:

<Mode> Discrete data. Baseband pre-modulation filter types are as follows:

RNYQuist Root Nyquist filter,

NYQuist Nyquist filter,
GAUSsian Gauss filter,
RECTangle Rectangle filter.

Example: :RADio:CUSTom:FILTer RNYQuist
The baseband premodulation filter is of Root Nyquist type.

Reset status: RNYQuist

Key Entry: 【Base】—>[Filter]—>[Filter Select]

[:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] <Dev>

Function: This command is used for setting FM deviation of FSK when the modulation type is set to FSK. This value will be valid after the FSK mode is selected by the user. For setting Modulation Type, see the command ":RADio:CUSTom:MODulation[:TYPE]".

Setting format: [:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation] <val>

Query format: [:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation]?

Parameter:

<Dev> FM deviation of FSK when Modulation Type is set to FSK.
Range: 0.4kHz [0.4kHz，20MHz].

Example: :RADio:CUSTom:MODulation:FSK 1MHz
FM deviation of FSK is 1MHz.

Reset status: 0.4kHz

Key Entry: 【Base】—>[Module Type]—>[FM Dev]

[:SOURce]:RADio:CUSTom:MODulation:MSK:PHASe <Phase>

Function: This command is used for setting PM bias of MSK when Modulation Type is set to MSK. This value will be valid after the MSK mode is selected by the user. For setting Modulation Type, see the command ":RADio:CUSTom:MODulation[:TYPE]".

Setting format: [:SOURce]:RADio:CUSTom:MODulation:MSK:PHASe <val>

Query format: [:SOURce]:RADio:CUSTom:MODulation:MSK:PHASe?

Parameter:

<Phase> PM bias of MSK when Modulation Type is set to MSK.
Range: 90rad [0rad，90rad].

Example: :RADio:CUSTom:MODulation:MSK:PHASe 30rad
PM bias is 30rad.

Reset status: 90.000rad

Key Entry: 【Base】—>[Module Type]—>[PM Bias]

[:SOURce]:RADio:CUSTom:MODulation[:TYPE] <Mode>

Function: This command sets Modulation Type. 。

Setting format: [:SOURce]:RADio:CUSTom:MODulation[:TYPE] BPSK|QPSK
|IS95QPSK|GRAYQPSK|OQPSK|IS95OQPSK|P4DQPSK|8PSK
|16PSK|D8PSK|MSK|2FSK|4FSK|8FSK|16FSK|C4FM|4QAM
|16QAM|32QAM|64QAM|128QAM|256QAM|512QAM|1024Q AM|ASK

Query format: [:SOURce]:RADio:CUSTom:MODulation[:TYPE]?

Parameter:

<Mode> Discrete data. For Modulation Type, please refer to the format of set command:

Example: :RADio:CUSTom:MODulation 8PSK

Modulation Type is set to 8PSK.

Reset status: QPSK

Key Entry: 【Base】—>[Module Type]—>[Modulation Type]

[:SOURce]:RADio:CUSTom:SRATe <Val>

Function: This command sets the baseband signal symbol rate of the signal generator in sps, ksps, Msps and Gsps.

Setting format: [:SOURce]:RADio:CUSTom:SRATe <val>

Query format: [:SOURce]:RADio:CUSTom:SRATe?

Parameter:

<Val> Baseband signal symbol rate.

The relationship between Modulation Type, Symbo Bits and Symbo Rate is as follows:

Modulation Type	Symbo Bits	Symbo Rate
BPSK	1	0.00005Msps – 50Msps
MSK	1	0.00005Msps – 50Msps
2FSK	1	0.00005Msps – 50Msps
OQPSK	2	0.00005Msps – 50Msps
QPSK	2	0.00005Msps – 50Msps
8FSK	3	0.00005Msps – 50Msps
QAM16	4	0.00005Msps – 50Msps

Example: :RADio:CUSTom:SRATe 3Msps Symbo rate is 3 Msps.

Reset status: 24.300000ksps

Key Entry: 【Base】—>[Base Config]->[Symbo Rate]

[:SOURce]:RADio:CUSTom:STATe <State>

Function: This command enables BaseBand of the signal generator to ON/OFF state. When BaseBand is turned on, the main information display area in user interface of the signal generator will display the indication of BaseBand and IQ Modulate.

Setting format: [:SOURce]:RADio:CUSTom:STATe ON|OFF|1|0

Query format: [:SOURce]:RADio:CUSTom:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: BaseBand in turned on,

OFF | 0: BaseBand in turned off.

Example: :RADio:CUSTom:STATe 1 Turn on BaseBand.

Reset status: 0

Key Entry: 【Base】—>[BaseBand ON OFF]

[:SOURce]:RADio:CUSTom:POLarity[:ALL] <Mode>

Function: This command sets the phase rotation direction of baseband signal, including: Normal and Reverse, signal is normally modulated in Normal mode; Q channel signal is reversed to complete the reverse of carrier signal in Reverse mode.

Setting format: [:SOURce]:RADio:CUSTom:POLarity[:ALL] NORMal|INVert

Query format: [:SOURce]:RADio:CUSTom:POLarity[:ALL]?

Parameter:

<Mode> Discrete data. Phase rotation mode of baseband signal is as follows:

NORMal | 0: Normal,

INVert | 1: Reverse.

Example: [:SOURce]:RADio:CUSTom:POLarity[:ALL] INV

Reverse phase of baseband signal.

Reset status: NORM

Key Entry: 【Base】—>[Base Config]—>[Phase Polarity Normal Reverse]

[:SOURce]:RADio:CUSTom:DENCode <State>

Function: This command enables Differential Encode ON/OFF. When Differential Encode is turned on, modulated bit is set to 1 if data bit is different from its previous bit; if data bits are the same, modulated bit will be set to 0. For example, when data bit is 1010 and Differential Encode is turned on, modulated bit is 1111.

Setting format: [:SOURce]:RADio:CUSTom:DENCode ON|OFF|1|0

Query format: [:SOURce]:RADio:CUSTom:DENCode?

Parameter:

<State> Boolean data is as follows:

ON | 1: Turn on Differential Encode,

OFF | 0: Turn off Differential Encode.

Example: [:SOURce]:RADio:CUSTom:DENCode 1 Differential Code ON.

Reset status: 0

Key Entry: 【Base】—>[Base Config]—>[Differential Code ON OFF]

[:SOURce]:RADio:CUSTom:VCO:CLOCk <Mode>

Function: This command is used for setting baseband sampling clock.

Setting format: [:SOURce]:RADio:CUSTom:VCO:CLOCk INTernal|EXTernal

Query format: [:SOURce]:RADio:CUSTom:VCO:CLOCk?

Parameter:

<Mode> Discrete datathat take following values:

INTernal: Internal sampling clock,

EXTernal: External sampling clock.

Example: [:SOURce]:RADio:CUSTom:VCO:CLOCk INT Internal sampling clock is selected.

Reset status: 0

Key Entry: 【Base】—>[Clock]—>[Base Sample Clock]

[:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay <Val>

Function: This command is used for setting the delay time (in bit) for the baseband signal to respond to the trigger signal after receipt of external trigger signal when Trig Source is set to EXT. This command is valid only when the external delay switch is turned on. For setting the state of external delay switch, see the command ":RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay:STATe" .

Setting format: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay <val>

Query format: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay?

Parameter:

<Val> Delay time when Trig Source is set to EXT.

Range: 0[0，1048575].

Example: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay 3000

The external trigger delay is 3000 bits.

Reset status: 0

Key Entry: 【Base】—>[Trigger]—>[Trig Source]—>[EXT]—>

[Trig Source-EXT]—>[DelayTime]

[:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay:STATe <State>

Function: This command is used for setting the state of the external trigger delay switch when the Trig Source of the signal generator is set to EXT. When the switch is ON, the set delay time will be valid. For setting external delay time, see the command ":RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay".

Setting format: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay:STATe ON|OFF|1|0

Query format: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay:STATe?

Parameter:

<State> Boolean data. Options for state of delay switch when Trig Source is set to EXT are:

ON | 1: Delay Switch ON,

OFF | 0: Delay Switch OFF.

Example: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay:STATe 1 Delay Switch is ON.

Reset status: 0

Key Entry: 【Base】—>[Trigger]—>[Trig Source]—>[EXT]

—>[Trig Source-EXT]—>[Delay Switch ON OFF]

[:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:SLOPe <Mode>

Function: This command is used for setting the polarity of trigger input signal of the rear panel to effective triggering of baseband output at high or low level when Trig Source is set to EXT.

Setting format: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:SLOPe POSitive|NGEative

Query format: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:SLOPe?

Parameter:

<Mode> Discrete data. Options for external trigger polarity are:

POSitive | 0: Pos,

NEGative | 1: Neg.

Example: [:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:SLOPe NEGative External trigger polarity is effective at low level.

Reset status: POS

Key Entry: 【Base】—>[Trigger]—>[Trig Source]—>[EXT]
—>[Trig Source-EXT] —> [Ext Trig Polar Pos Neg]

[:SOURce]:RADio:CUSTom:TRIGger:SOURce <Mode>

Function: This command sets the baseband signal trigger source of the signal generator, including: KEY, BUS and EXT. For more details。

Setting format: [:SOURce]:RADio:CUSTom:TRIGger:SOURce KEY|BUS|EXT

Query format: [:SOURce]:RADio:CUSTom:TRIGger:SOURce?

Parameter:

<Mode> Discrete data. Baseband signal trigger source is as follows:

KEY | 0: The trigger source is from the trigger key on the front panel of the instrument,

BUS | 1: The trigger source is from GPIB group execute trigger, or triggered when "*TRG" command is received;

EXT | 2: The trigger source is from the trigger input of ports on the rear panel of the instrument.

Example: [:SOURce]:RADio:CUSTom:TRIGger:SOURce BUS

Trig Source is set to BUS.

Reset status: KEY

Key Entry: 【Base】—>[Trigger]—>[Trig Source]

[:SOURce]:RADio:CUSTom:TRIGger:TYPE <Mode>

Function: This command sets the baseband signal trigger mode that controls data transmission, including: Continue, Single and Gate

Setting format: [:SOURce]:RADio:CUSTom:TRIGger:TYPE CONTinuous|SINGle|GATE

Query format: [:SOURce]:RADio:CUSTom:TRIGger:TYPE?

Parameter:

<Mode> Discrete data. Baseband signal trigger style is as follows:

CONTinuous | 0: Trig Style is set to Continue,

SINGle | 1: Trig Style is set to Single,

GATE | 2: Trig Style is set to Gate.

Example: [:SOURce]:RADio:CUSTom:TRIGger:TYPE SING

Baseband trigger mode is set to Single.

Reset status: CONT

Key Entry: 【Base】—>[Trigger]—>[Trig Style]

[:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous:TYPE <Mode>

Function: This command is used for setting the response mode of baseband data to trigger signal when the baseband trigger mode is set to Continue. Three modes, namely Auto, Trig and Realtime, are available for the user. For setting baseband signal trigger mode, see the command ":RADio:CUSTom:TRIGger:TYPE"

Setting format: [:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous:TYPE FREE|TRIGger|RESet

Query format: [:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous:TYPE?

Parameter:

<Mode> Discrete data. Options for response mode to trigger signal when baseband trigger mode is set to Continue are:

FREE | 0: At Continue Auto mode, the baseband signal will continue output until the baseband switch is turned off or the trigger mode is changed.

TRIGger | 1: At Trig mode, the baseband signal will start output only after a trigger signal is received.

RESet | 2: At Realtime mode, the baseband signal will stop current output, then recalculate the baseband data and load the output.

Example: [:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous:TYPE FREE

Set trigger mode to Continue Auto.

Reset status: FREE

Key Entry: **【Base】—>[Trigger]—>[Trig Style]—>[Continue>>]**

[:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive <Mode>

Function: This command is used for setting the gate trigger types at baseband trigger mode, which include Low and High. When the external trigger signal is at high or low level, the trigger baseband signal will be outputted. For setting baseband signal trigger mode, see the command ":RADio:CUSTom:TRIGger:TYPE"

Setting format: [:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive LOW|HIGH

Query format: [:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive?

Parameter:

<Mode> Discrete data. Options for types of trigger signal when baseband trigger mode is set to Gate are:

LOW | 0: Low,

HIGH | 1: Hight.

Example: [:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive HIGH

When the external trigger signal is at high level, the trigger signal will be outputted.

Reset status: LOW

Key Entry: **【Base】—>[Trigger]—>[Trig Style]—>[Gate>>]**

[:SOURce]:RADio:MTONe:ARB:SETup <FileName>

Function: This command selects multi tone file and loads it into the signal generator memory to play. For command parameters, it is only necessary to set multi tone file name without

specifying an absolute path.

Setting format: [:SOURce]:RADio:MTONe:ARB:SETup <file_name>

Parameter:

<FileName> String type, multi tone file name.

Example: [:SOURce]:RADio:MTONe:ARB:SETup “mtone1.mtn”

Load file mtone1.mtn to the memory of the signal generator.

Key Entry: 【Dual/Multi Tone】—>[Multi Tone]—>[Base Config]—>[Load]

Note: for setting only.

[:SOURce]:RADio:MTONe:ARB:SETup:STORe <FileName>

Function: This command stores the waveform data in the current multi tone table to the multi tone file of the signal generator. For command parameters, it is only necessary to set multi tone file name without specifying an absolute path.

Setting format: [:SOURce]:RADio:MTONe:ARB:STORe <file_name>

Parameter:

<FileName> String type, multi tone file name.

Example: [:SOURce]:RADio:MTONe:ARB:STORe “mtone1.mtn”

Save multi tone table to the signal generator multi tone file mtone1.mtn.

Key Entry: 【Dual/Multi Tone】—>[Multi Tone]—>[Base Config]—>[Save]

Note: for setting only.

[:SOURce]:RADio:MTONe:ARB:SETup:TABLe <FreqSpacing>,<NumTones>,<Pow>,<Phase>,<State>

Function: This command creates and configures a multi tone waveform. The command parameters include: <freq_offset>, <num_tones>, <pow>, <phase> and <state>. <Freq_offset>, this parameter is limited by 200 M baseband bandwidth and tone count in multi tone table. Such value between tones is the same as frequency interval.

Setting format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe <freq_spacing>,<num_tones>,{<pow>,<phase>,<state>…}

Parameter: String Type Parameter

<FreqSpacing> Freq Interval.

Range: 1MHz[100Hz，200MHz].

<NumTones> Tone Count

Range: 2[2，64].

<Pow> Atten Ampl

Range: 0dB[-100dB，0dB].

<Phase> Initial Phase.

Range: 0deg[0deg，359deg].

<State> Status. Boolean data is as follows:

1： ON,

0： off.

Example: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe “1000000，3，-10，90，0，-20，0，1，

-30，45，1”

This example shows that the multi tone frequency interval is set to 1 MHz. There are three tones. The first has an Atten Ampl of 10 dB and Phase of 90 degrees, in OFF status; the second has an Atten Ampl of 20 dB and Phase of 0 degrees, in ON status; the third has an Atten Ampl of 30 dB and Phase of 45 degrees, in ON status.

Reset status: 1MHz，2，0dB，0deg，1

Key Entry: None

Note: for setting only.

[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing <FreqSpacing>

Function: This command sets Freq Interval, which takes effect only after Multi Tone is turned on. For enabling or disabling the multitone modulation function, see the command “:RADio:MTONe:ARB:STATe”. For designating other parameters of multitone modulation or setting the multitone list, see the command “:RADio:MTONe:ARB:SETup:TABLe”.

Setting format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing <val><freq unit>

Query format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing?

Parameter:

<FreqSpacing> Freq Interval.

Range: 1MHz[100Hz，200MHz].

Example: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:FSPacing 200kHz

Set frequency interval in multi tone table to 200 kHz.

Reset status: 1MHz

Key Entry: **【Dual/Multi Tone】—>[Multi Tone]—>[Base Config]—>[Freq Interval]**

[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes <NumTones>

Function: This command is used for setting the number of tone in the multitone modulation list. This command is valid only when multitone modulation function is enabled. For enabling or disabling the multitone modulation function, see the command “:RADio:MTONe:ARB:STATe”. For designating other parameters of multitone modulation or setting the multitone list, see the command “:RADio:MTONe:ARB:SETup:TABLe”.

Setting format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes <val>

Query format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NTONes?

Parameter:

NumTones> Number of tone in the multitone modulation list.

Range: 2[2，64].

Example: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:NOTNes 3

Set number of tone in multitone modulation list to 3.

Reset status: 2

Key Entry: **【Dual/Multi Tone】—>[Multi Tone]—>[Base Config]—>[Tone Count]**

[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize <Mode>

Function: This command initializes initial phase mode in multi tone modulation table, including: Random and Fixed. In Fixed mode, the phase of all tones in multi tone table will be set to a Fixed value (0 degree); in Random mode, the phase of all tones in multi tone table will be set to different random values based on random seed. For setting phase relationship of tone for multitone modulation, see the command ":RADio:MTONe:ARB:SETup:TABle:PHASe:INITialize:SEED".

Setting format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize RANDom|FIXed

Query format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize?

Parameter:

<Mode> Discrete data. Initial phase mode in multi tone modulation table is as follows:

RANDom : Set all tones to random value,

FIXed ： Set all tones to fixed value.

Example: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize FIX

Set the tone phase in multi tone table to a fixed value.

Reset status: FIXed

Key Entry: **【Dual/Multi Tone】—>[Multi Tone]—>[Base Config]—>[Initial Phase]**

[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize:SEED <Mode>

Function: This command sets Phase Rela, including: Random and Fixed.

Setting format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize:SEED RANDom|FIXed

Query format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize:SEED?

Parameter:

<Mode> Discrete data. Phase Rela is as follows:

RANDom: Phase Rela is Random,

FIXed ： Phase Rela is Fixed.

Example: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:PHASe:INITialize:SEED FIX Set Phase Rela to Fixed.

Reset status: FIX

Key Entry: **【Dual/Multi Tone】—>[Multi Tone]—>[Base Config]—>[Phase Rela]**

[:SOURce]:RADio:MTONe:ARB:SETup:TABLe:ROW <RowIndex>,<Pow>,<Phase>,<State>

Function: This command is used for changing the parameters in a certain row of the multitone modulation list, which include <row_index>, <pow>, <phase> and <state>. If the user needs to change the whole list, see the command "RADio:MTONe:ARB:SETup:TABLe".

Setting format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:ROW <row_index>,<pow>,<phase>,<state>

Query format: [:SOURce]:RADio:MTONe:ARB:SETup:TABLe:ROW? <row_Index>

Parameter:

<RowIndex> Row index of the multitone modulation list.

Range: 0[0，63].

<Pow> Atten Ampl

Range: 0dB[-100dB，0dB].

<Phase> Initial Phase.

Range: 0deg[0deg，359deg].

<State> Status. Boolean data is as follows:

1： ON,

0： off.

Example: :RADio:MTONe:ARB:SETup:TABLe:ROW “2，-10，40，0” This example indicates that, in the second row of the list, Atten Ampl is set to -10dB, Phase set to 40deg and Status set to OFF.

:RADio:MTONe:ARB:SETup:TABLe:ROW? 2 indicates query of index 2, namely the information in the third row, including Freq Offset, Atten Ampl, Phase and Status.

Reset status: 0，0dB，0deg，0

Key Entry: None

[:SOURce]:RADio:MTONe:ARB[:STATe] <State>

Function: This command sets Multi Tone ON/OFF state of the signal generator. When Multi Tone is turned on, the main information display area in user interface of the signal generator will display the indication of IQ Modulate and Multi Tone.

Setting format: [:SOURce]:RADio:MTONe:ARB:STATe ON|OFF|1|0

Query format: [:SOURce]:RADio:MTONe:ARB:STATe?

Parameter:

<State> Boolean data. Multi Tone ON/OFF state is as follows:

ON | 1: Multi Tone is turned on,

OFF | 0: Multi Tone is turned off.

Example: [:SOURce]:RADio:MTONe:ARB:STATe 1 Multitone modulation function is enabled.

Reset status: 0

Key Entry: 【Dual/Multi Tone】—>[Multi Tone]—>[Base Config]—>[Multi Tone ON OFF]

[:SOURce]:RADio:TTONe:ARB:ALIGnment <Mode>

Function: This command sets Dual Tone Alignment, including: Left, Centre and Right, which takes effect only after Dual Tone is turned on. For enabling or disabling dual tone modulation function, see the command “:RADio:TTONe:ARB:STATe”.

Setting format: [:SOURce]:RADio:TTONe:ARB:ALIGnment LEFT|CENTer|RIGHt

Query format: [:SOURce]:RADio:TTONe:ARB:ALIGnment?

Parameter:

<Mode> Discrete data. Dual Tone Alignment is as follows:

LEFT | 0: Left,

CENTer | 1: Center,

RIGHt | 2: Right.

Example: [:SOURce]:RADio:TTONe:ARB:ALIGnment RIGHt

Display Dual Tone Alignment to be displayed to the right of carrier.

Reset status: CENT

Key Entry: 【Dual/Multi Tone】—>[Dual Tone]—>[Alignment>>]

[:SOURce]:RADio:TTONe:ARB:FSPacing <FreqSpacing>

Function: This command is used for setting Freq Separation of dual tone. This command is valid only when Dual Tone is set as ON. For enabling or disabling the dual tone modulation function, see the command ":RADio:TTONe:ARB:STATe".

Setting format: [:SOURce]:RADio:TTONe:ARB:FSPacing <val><freq unit>

Query format: [:SOURce]:RADio:TTONe:ARB:FSPacing?

Parameter:

<FreqSpacing>Frequency spacing of dual tone.

Range: 10MHz[0Hz，40MHz].

Example: [:SOURce]:RADio:TTONe:ARB:FSPacing 30MHz

Set the frequency spacing of dual tone to 30MHz.

Reset status: 10MHz

Key Entry: 【Dual/Multi Tone】—>[Dual Tone]—>[Freq Separation]

[:SOURce]:RADio:TTONe:ARB[:STATe] <State>

Function: This command is used for enabling the dual tone modulation function of the signal generator. When this function is enabled, the main information display area of the signal generator will display IQ modulation and dual tone indication.

Setting format: [:SOURce]:RADio:TTONe:ARB:STATe ON|OFF|1|0

Query format: [:SOURce]:RADio:TTONe:ARB:STATe?

Parameter:

<State> Boolean data. Options for state of dual tone modulation are:

ON | 1: Dual Tone ON,

OFF | 0: Dual Tone OFF.

Example: [:SOURce]:RADio:MTONe:ARB:STATe 1 Multi Tone is ON.

Reset status: 0

Key Entry: 【Dual/Multi Tone】—>[Dual Tone]—>[Dual Tone ON OFF]

[:SOURce]:RADio:ARB:MODE <Mode>

Function: This command is used for setting arbitrary waveform modes, which include Arb and Seq. At Arb mode, the user can load and play any arbitrary waveform data file in the user-defined format; at Seq mode, the user can generate waveform segment files as required and combine waveform segments for sequence play. See the command ":RADio:ARB:SEQuence".

Setting format: [:SOURce]:RADio:ARB:MODE ARB|SEQuence

Query format: [:SOURce]:RADio:ARB:MODE?

Parameter:

<Mode> Discrete data. Arb mode is as follows:

ARB | 0: Arb mode,

SEQuence | 1: Seq mode.

Example: [:SOURce]:RADio:ARB:MODE SEQ Operation mode of arbitrary waveform is set to Seq.

Reset status: SEQ

Key Entry: 【Arb】—>[Base Config] —>[Work Pattern Arb Seq]

[:SOURce]:RADio:ARB[:STATe] <State>

Function: This command enables ARB generator state of the signal generator. When ARB is turned on, the main information display area in user interface of the signal generator will display an indication.

Setting format: [:SOURce]:RADio:ARB:STATe ON|OFF|1|0

Query format: [:SOURce]:RADio:ARB:STATe?

Parameter:

<State> Boolean data is as follows:

ON | 1: ARB is turned on,

OFF | 0: ARB is turned off.

Example: [:SOURce]:RADio:ARB:STATe 1 Arbitrary waveform is enabled.

Reset status: 0

Key Entry: 【Arb】—>[Base Config] —>[Arb Seq ON OFF]

[:SOURce]:RADio:ARB:SEQuence <FileName>,<WaveForm>,<Reps>,<Marks>

Function: The command indicates that a waveform sequence is loaded. A waveform sequence can be composed of multiple waveform segments. Parameters are composed of file_name, waveform, reps and M1M2M3M4. File_name refers to the folder where waveform segment file is stored. The user-specified folder can only be that under relative path. In this parameter, the user has no right to specify an absolute path. For example, if the user names "D:\\USER\\SEQ" for file_name, these folders cannot be created under D drive, and the file name is considered wrong; waveform refers to the specific waveform segment file. The maximum number of waveform segment files supported by this command is 64; Reps refers to the number of times each waveform segment is played cyclically, and one waveform segment file can be played back up to 65,535 times; M1M2M3M4 is a mark switch of each waveform segment files, for example: If users do not want to output any mark of waveform segment, they can select NONE, or ALL if they want to output all marks of waveform segment.

Setting format: [:SOURce]:RADio:ARB

:SEQuence <file_name>,<waveform>,<reps>,NONE

|M1|M2|M3|M4|M1M2|M1M3|M1M4|M2M3|M2M4

|M3M4|M1M2M3|M1M2M4|M1M3M4|M2M3M4|ALL,{

<waveform2>,<reps>,NONE|M1|M2|M3|M4

|M1M2|M1M3|M1M4|M2M3|M2M4|M3M4|M1M2M3

|M1M2M4|M1M3M4|M2M3M4|ALL

Parameter:

<FileName> String type.

The folder where waveform segment file is stored, the folder specified by users can only be that under relative path.

<WaveForm> String type.

Name of waveform segment file, the maximum number of waveform segment file supported by this command is 64.

<Reps> Integer. Times of looped playback of each waveform segment.

Range: 1[1，65535].

<Marks> Discrete data type, mark switch of each waveform segment file, specific option is in command format.

Example: [:SOURce]:RADio:ARB:SEQuence Seq1，waveform1，12，NONE，vaveform2，300，M1M2

Load a waveform sequence, which is located in Seq1 folder which contains waveform1 and waveform2. The former is played 12 times with no mark output; the latter is played 300 times, with mark 1 and mark 2 of each symbol output.

Key Entry: 【Arb】—>[Base Config] —>[Add Wave Seg]

Note: for setting only.

[:SOURce]:RADio:ARB:SEQuence:CLOCk <Mode>

Function: This command sets Clock Type when the signal generator is in Arb mode. Arb mode allows users to use CUSTom mode only. Other modes cannot be set; Seq mode enables users to select CURRent, HIGH or CUSTom. For setting operation mode of arbitrary waveform, see the command ":RADio:ARB:MODE".

Setting format: [:SOURce]:RADio:ARB:SEQuence:CLOCk CURRent|HIGH|CUSTom

Query format: [:SOURce]:RADio:ARB:SEQuence:CLOCk?

Parameter:

<Mode> Discrete data. Clock Type in Arb mode is as follows:

CURRent	| 0: Play waveform segment files in Seq mode at sampling rate of each waveform segment;
HIGH	| 1: Play waveform segment files in Seq mode at maximum sampling rate of waveform segments each waveform segment;
CUSTom	| 2: Play waveform segment files in Seq mode at the current clock frequency setting of the signal generator.

Example: [:SOURce]:RADio:ARB:SEQuence:CLOCk HIGH

Set Clock Type to Highest.

Reset status: CURR

Key Entry: 【Arb】—>[Base Config] —>[Clock Type]

[:SOURce]:RADio:ARB:SCLock:RATE <ClockRate>

Function: This command is used for setting the signal sampling rate of arbitrary waveform. The set

value will be valid only when Clock Type is set to User Defined. For setting Clock Type, see the command ":RADio:ARB:SEQuence:CLOCk".

Setting format: [:SOURce]:RADio:ARB:SCLock:RATE <val><freq unit>

Query format: [:SOURce]:RADio:ARB:SCLock:RATE?

Parameter:

<ClockRate> I/Q signal Q offset.

Range: 100MHz[0.01MHz，250MHz].

Example: [:SOURce]:RADio:ARB:SCLock:RATE 50MHz ARB Clock Freq is 50 MHz.

Reset status: 100MHz

Key Entry: 【Arb】—>[Base Config] —>[Clock Freq]

[:SOURce]:RADio:ARB:TRIGger:TYPE <Mode>

Function: This command is used for setting the trigger modes for control of waveform play, which include CONTinuous, SINGle, GATE and SADVance. For single trigger function, the system may receive effective trigger event only after the play of current code pattern sequence has been accomplished. In addition, multiple status can be set at CONTinuous, SINGle, GATE and SADVance modes. See the command ":RADio:ARB:TRIGger:TYPE:CONTinuous", ":RADio:ARB:TRIGger:TYPE:SINGle", ":RADio:ARB:TRIGger:TYPE:SADVance:TYPE", ":RADio:ARB:TRIGger:TYPE:GATE:ACTive".

Setting format: [:SOURce]:RADio:ARB:TRIGger:TYPE CONTinuous |SINGle|SADVance|GATE

Query format: [:SOURce]:RADio:ARB:TRIGger:TYPE?

Parameter:

<Mode> Discrete data. At Arb mode, options for trigger mode for control of waveform play are:

CONTinuous	| 0: When Trig Mode is set to Continue, the waveform sequence will be played repeatedly after the instrument has received effective trigger event;
SINGle	| 1: When Trig Mode is set to Single, the waveform sequence will be played only once after the instrument has received effective trigger event;
SADVance	| 2: When Trig Mode is set to Wave Segment, one waveform segment will be played each time the instrument receives an effective trigger event;
GATE	| 3: When Trig Mode is set to Gate, the waveform sequence will be played continuously during the valid time of the gate signal.

Example: [:SOURce]:RADio:ARB:TRIGger:TYPE SING Trig Mode is set to Single.

Reset status: CONT

Key Entry: 【Arb】—>[Trigger]—>[Trig Mode]

[:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous[:TYPE] <Mode>

Function: This command sets the mode in which sequence file responds to trigger signal when ARB is in Continue or Single mode. There are three modes for selection, FREE, TRIGger and RESet. For setting Trig Mode, see the command ":RADio:ARB:TRIGger:TYPE".

Setting format: [:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous FREE|TRIGger|RESet

Query format: [:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous?

Parameter:

<Mode> Discrete data. The mode in which sequence file responds to trigger signal when ARB is in Continue mode,

The values are as follows:

FREE	0: When Auto mode is selected, sequence is triggered automatically to start playing after sequence waveform data is downloaded. All trigger events are ignored during playback
TRIGger	1: When Trig mode is selected, no current waveform sequence will be played until a valid trigger event is received. After a valid trigger event is received, the system starts to play the current waveform sequence. After the sequence is played, no current waveform sequence is replayed until a valid trigger event occurs
RESet	2: When Realtime mode is selected, no modulation source data will be generated until a valid trigger event is received, with no code pattern signal generated; after a valid trigger event is received, the system starts to generate the selected modulation source data, thereby generating corresponding code pattern data and signals. After the current source data is generated, the system will automatically restart to generate the currently set modulation source data. During the generation process of modulation source data, if a valid trigger event is received, the system will immediately abort the modulation source data currently being generated and restart to generate the set modulation source data

Example: [:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous TRIG

When ARB is in Continue mode, set the mode in which sequence file responds to trigger signal to Trig.

Reset status: FREE

Key Entry: 【Arb】—>[Trigger]—>[Trig Mode]—>[Continue]

[:SOURce]:RADio:ARB:TRIGger:TYPE:SINGle <Mode>

Function: This command sets the mode in which sequence file responds to trigger signal when ARB is in Single mode. There are three modes for selection, FREE, TRIGger and

RESet. For setting Trig Mode, see the command ":RADio:ARB:TRIGger:TYPE".

Setting format: [:SOURce]:RADio:ARB:TRIGger:TYPE:SINGle FREE|TRIGger|RESet

Query format: [:SOURce]:RADio:ARB:TRIGger:TYPE:SINGle?

Parameter:

<Mode> Discrete data. The mode in which sequence file responds to trigger signal when ARB is in Single mode,

The values are as follows:

FREE | 0: When Auto mode is selected, sequence is triggered automatically to start playing after sequence waveform data is downloaded. All trigger events are ignored during playback

TRIGger | 1: When Trig mode is selected, no current waveform sequence will be played until a valid trigger event is received. After a valid trigger event is received, the system starts to play the current waveform sequence. After the sequence is played, no current waveform sequence is replayed until a valid trigger event occurs

RESet | 2: When Realtime mode is selected, no modulation source data will be generated until a valid trigger event is received, with no code pattern signal generated; after a valid trigger event is received, the system starts to generate the selected modulation source data, thereby generating corresponding code pattern data and signals. After the current source data is generated, the system will automatically restart to generate the currently set modulation source data. During the generation process of modulation source data, if a valid trigger event is received, the system will immediately abort the modulation source data currently being generated and restart to generate the set modulation source data

Example: [:SOURce]:RADio:ARB:TRIGger:TYPE:SINGle TRIG

When ARB is in Single mode, set the mode in which sequence file responds to trigger signal to Trig.

Reset status: FREE

Key Entry: 【Arb】—>[Trigger]—>[Trig Mode]—>[Single]

[:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance[:TYPE] <Mode>

Function: This command sets the mode in which sequence file responses to trigger signal when ARB is in Wave Segment mode. There are two modes for selection, Single and Continue. Wave segment play trigger function is not for the entire waveform sequence, but a single wave segment in the sequence. For setting Trig Mode, see the command ":RADio:ARB:TRIGger:TYPE".

Setting format: [:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance:TYPE SINGle|CONTinuous

Query format: [:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance:TYPE?

Parameter:

<Mode> Discrete data. The mode in which sequence file responses to trigger signal when ARB is in Wave Segment mode,

The values are as follows:

SINGle | 0: Single waveform segment trigger,

CONTinuous | 1: Continue waveform segment, trigger.

Example: [:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance:TYPE SING

When ARB is in Wave Segment mode, set the mode in which sequence file responds to trigger signal to Single.

Reset status: SING

Key Entry: 【Arb】—>[Trigger]—>[Trig Mode]—>[Wave Segment >>]

[:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive <Mode>

Function: This command sets the mode in which sequence file responds to trigger signal when ARB is in Gate mode. There are two modes for selection, Low/Hight. For setting Trig Mode, see the command ":RADio:ARB:TRIGger:TYPE".

Setting format: [:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive LOW|HIGH

Query format: [:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive?

Parameter:

<Mode> Discrete data. The mode in which sequence file responds to trigger signal when ARB is in Gate mode,

The values are as follows:

LOW | 0: Low,

HIGH | 1: Hight.

Example: [:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive LOW

When ARB is in Gate mode, set the mode in which sequence file responds to trigger signal to Low.

Reset status: LOW

Key Entry: 【Arb】—>[Trigger]—>[Trig Mode>>]—>[Gate>>]

[:SOURce]:RADio:ARB:TRIGger:SOURce <Mode>

Function: This command sets ARB Trig Source.

Setting format: [:SOURce]:RADio:ARB:TRIGger:SOURce KEY|BUS|EXT|INT

Query format: [:SOURce]:RADio:ARB:TRIGger:SOURc e?

Parameter:

<Mode> Discrete data. ARB Trig Source,

The values are as follows:

KEY: Trigger key,

BUS: Bus,

EXT: External,

INT: Internal.

Example: [:SOURce]:RADio:ARB:TRIGger:SOURce BUS

Set Trig Source to Bus.

Reset status: KEY

Key Entry: 【Arb】—>[Trigger]—>[Trig Source]

[:SOURce]:RADio:ARB:VCO:CLOCk <Mode>

Function: This command sets ARB Samp Clock. When Samp Clock is Int, the clock frequency is 200 MHz. And it can not be changed. When Samp Clock is set to Ext, the clock frequency can be set by the external clock frequency command. This command is "RADio:ARB:EXTernal:CLOCk:RATE".

Setting format: [:SOURce]:RADio:ARB:VCO:CLCOk INTernal|EXTernal

Query format: [:SOURce]:RADio:ARB:VCO:CLOCk?

Parameter:

<Mode> Discrete data. ARB Samp Clock is as follows:

INTernal : Internal,

EXTernal : EXT.

Example: [:SOURce]:RADio:ARB:VCO:CLOCk EXTernal

Set Trig Source to Bus.

Reset status: LOW

Key Entry: 【Arb】—>[Trigger]—>[Samp Clock]

[:SOURce]:RADio:ARB:EXternal:CLOCk:RATE <ClockRate>

Function: This command is used for setting the external clock frequency. It is valid when Samp Clock is set to Ext. For setting Samp Clock, see the command "RADio:ARB:VCO:CLOCk".

Setting format: [:SOURce]:RADio:ARB:EXTernal:CLCOk:RATE <val><freq unit>

Query format: [:SOURce]:RADio:ARB:EXTernal:CLOCk:RATE?

Parameter:

<Mode> ARB Sampl Clock is as follows:

Range [100Hz,250MHz]

Example: [:SOURce]:RADio:ARB:EXTernal:CLOCk:RATE 100MHz

Set Trig Source to Bus.

Reset status: 200MHz

Key Entry: 【Arb】—>[Trigger]—>[Ext Clock Freq]

3.3.12 MEMory subsystem

The following commands are used for setting memory operation modes:

- :MEMory:COPY:NAME 108
- :MEMory:DELete:NAME 108
- :MEMory:MOVE 108
- :MEMory:DATA 108

:MEMory:COPY:NAME <SrcName>,<DestName>

Function: This command is used for copying the data of one file to another file. If the source file and destination file are not in the same folder, the destination file can be named as that of the source file. When copying arbitrary waveform file, the marked files related to this file will be copied at the same time. It should be noted that the file must be named with absolute path.

Setting format: :MEMory:COPY:NAME <src_name>,<dest_name>

Parameter:

<SrcName> String type, name of the source file.

<DestName> String type, name of the destination file.

Example: :MEMory:COPY:NAME c:\\data\\user\\seq1.dat,c:\\data\\user\\seq2.dat

Copy the data in seq1.dat to seq2.dat.

Key Entry: 【File】—>[Copy]

Note: for setting only.

:MEMory:DELete:NAME <FileName>

Function: This command is used for deleting the user file in the signal generator,

Setting format: :MEMory:DLEete:NAME <file_name>

Parameter:

<FileName> String type. The user file stored in the signal generator.

Example: :MEMory:DELete:NAME c:\\arb1.seq

Delete File arb1.seq in Disk C.

Key Entry: None

Note: for setting only.

:MEMory:MOVE <FileName>

Function: This command is used for renaming the file in the signal generator. It should be noted that the file must be named with absolute path.

Setting format: :MEMory:MOVE <Sourfile_name>，<Desfile_name>

Parameter:

<FileName> String type. The file name stored in the signal generator.

Example: :MEMory:MOVE c:\data\arb1.seq,c:\data\arb2.seq

Rename file arb1.seq to arb2.seq.

Note: for setting only.

:MEMory:DATA <FileName>,<#AB\n><DataBlock>

Function: This command is used for downloading arbitrary waveform data to the signal generator through the communication interface in the form of data block, and naming it as <file_name> to store it in the instrument. This command can only transfer binary data with the number of bytes less than 1000000000 bytes. In other words, the transfered data has less than 10 digits.

Setting format: :MEMory:DATA <file_name>，<data_block>

Parameter:

<FileName> String type. For the user-defined arbitrary waveform file name stored in the signal generator, the user is not authorized to designate its absolute path.

<#AB\n> "#" and "\n" are fixed formats. "#" refers to beginning of data, "\n" refers to placeholder, "A" refers to data length, and "B" refers to data length. Take "#210\n" for example. "3" indicates the data has total 2 digits, "10" refers to the total number of bytes, <DataBlock> behind "10" refers to the data with 10 bytes.

<DataBlock> Data block needed to be transfered.

Example: [:SOURce]:MEMory:DATA arb1.dat，#41024\n jklasdj…

It indicates data with 1024 bytes are transfered to the signal generator and named as arb1.dat to store in the generator.

Note: for setting only.

3.3.13 ROSCillator subsystem

Oscillator subsystem commands are used for the functions related to the time base of the signal generator.

- [:SOURce]:ROSCillator:ADJust:REFerence 109

[:SOURce]:ROSCillator:ADJust:REFerence <Val>

Function: This command adjusts the internal reference of the signal generator by setting internal calibration parameters to make the frequency output accuracy more accurate. It should be noted that the instrument needs to be warmed up within 2 hours after the signal generator is started. Do not change the internal reference value. For more details, please refer to the User Manual of 1465 Vector Signal Generator.

Setting format: [:SOURce]:ROSCillator:ADJust:REFerence <val>

Query format: [:SOURce]:ROSCillator:ADJust:REFerence?

Parameter:

<Val> Internal calibration parameter data.

Range: [0，32767].

Example: :ROSCillator:ADJust:REFerence 30000

Adjust internal reference accuracy to 30000.

3.3.14 SYSTem subsystem

These subsystem commands are used for the functions related to overall performance of the signal generator.

The following commands are used for setting operation modes:

- :DIAGnostic:INFormation:CCOunt:PON 110
- :DIAGnostic:INFormation:OTIMe 110
- :DIAGnostic:SNUM 110
- :SYSTem:COMMunicate:GPIB:ADDRess 110
- :SYSTem:COMMunicate:GTLocal 111
- :SYSTem:DEVice:LANGuage 111

- :SYSTem:COMMunicate:LAN:IP 111
- :SYSTem:COMMunicate:LAN:SUBNet 111
- :SYSTem:COMMunicate:LAN:GATeway 112
- :SYSTem:ERRor[:NEXT] 112
- :SYSTem:PRESet:TYPE 112

:DIAGnostic:INFormation:CCOunt:PON

Function: This command is used to query the cumulative number of times the instrument is powered on

Query format: :DIAGnostic:INFormation:CCOunt:PON？

Example: :DIAGnostic:INFormation:CCOunt:PON？ This example indicates query of cumulative startup times of the signal generator.

:DIAGnostic:INFormation:OTIMe

Function: This command is used to query the instrument firmware date and time stamp

Query format: :DIAGnostic:INFormation:OTIMe？

Example: :DIAGnostic:INFormation:OTIMe？ This example indicates query of cumulative startup hours of the signal generator.

:DIAGnostic:SNUM?

Function: This command is used for reading the system serial number of the signal generator.

Query format: :DIAGnostic:SNUM？

Return value: Serial number

Example: :DIAGnostic:SNUM This example indicates query of the system serial number of the signal generator.

Note: for query only.

:SYSTem:COMMunicate:GPIB:ADDRess <Address>

Function: This command is used for setting GPIB address of the signal generator. Default address is 19. To ensure normal communication, local GPIB address shall be different from other GPIB addresses in the same test system.

Setting format: :SYSTem:COMMunicate:GPIB:ADDRess <val>

Query format: :SYSTem:COMMunicate:GPIB:ADDRess?

Parameter:

<Address> Integer data, GPIB address.

Range: 19[0，30].

Example: :SYSTem:COMMunicate:GPIB:ADDRess 19

Set GPIB address of the signal generator to 19.

Reset status: 19

Key Entry: 【System】—>[GPIB Port] —>[Local GPIB Addr]

:SYSTem:COMMunicate:GTLocal

Function: This command is used for switching the signal generator to local operation mode. In this mode, the user can operate front panel buttons of the instrument. At the same time, remote operation mode indication in the instrument operation interface will disappear.

Setting format: :SYSTem:COMMunicate:GTLocal

Example: :SYSTem:COMMunicate:LAN:IP 172.141.114.114

:SYSTem:COMMunicate:GTLocal

Switch the signal generator from remote status to local status.

Note: for setting only.

:SYSTem:DEVice:LANGuage <Mode>

Function: This command is used for setting the language displayed in the interface. At present, the instrument supports interface in both Chinese and English. The interface in Chinese is set as default.

Setting format: :SYSTem:DEVice:LANGuage CHINese|ENGLish

Query format: :SYSTem:DEVice:LANGuage?

Parameter description

<Mode> Discrete data. Options for interface language are:

CHINeses Chinese

ENGLish English

Example: SYSTem:DEVice:LANGuage ENGLish Set Language as English.

Reset status: Keep current language unchanged.

Key Entry: 【System】—>[Base Config]—>[Language/LANG]

:SYSTem:COMMunicate:LAN:IP <Address>

Function: This command is used for setting the IP address of the signal generator, which applies dotted decimal notation.

Setting format: :SYSTem:COMMunicate:LAN:IP <ipstring>

Query format: :SYSTem:COMMunicate:LAN:IP?

Parameter description

<Address> String type, IP address in dotted decimal notation

Example: :SYSTem:COMMunicate:LAN:IP “172.141.114.114”

Set IP address of signal generator to 172.141.114.114.

Key Entry: 【System】—>[LAN Port]—>[Local Machine IP Addr]

:SYSTem:COMMunicate:LAN:SUBNet <Address>

Function: This command is used for setting the subnet mask address of the signal generator, which applies dotted decimal notation.

Setting format: :SYSTem:COMMunicate:LAN:SUBNet <ipstring>

Query format: :SYSTem:COMMunicate:LAN: SUBNet?

Parameter description

<Address> String type, IP address in dotted decimal notation

Example: :SYSTem:COMMunicate:LAN: SUBNet “255.255.255.0”
Set the subnet mask address of signal generator to 255.255.255.0.

Key Entry: 【System】—>[LAN Port]—>[NET MASK]

:SYSTem:COMMunicate:LAN:GATeway <Address>

Function: This command is used for setting network gateway address of the signal generator in external LAN, which applies dotted decimal notation.

Setting format: :SYSTem:COMMunicate:LAN:GATeway <ipstring>

Query format: :SYSTem:COMMunicate:LAN:GATeway?

Parameter description

<Address> String type, IP address in dotted decimal notation

Example: :SYSTem:COMMunicate:LAN: GATeway “172.141.114.254”
Set network gateway address of the signal generator to 172.141.114.254.

Key Entry: 【System】—>[LAN Port]—>[Default Gate]

:SYSTem:ERRor[:NEXT]?

Function: This command is used for querying error information in the error list of the signal generator. Error information will be deleted from the list each time it is queried. If there is no error information in the list, the user will find the message “+0，No ERROR”.

Query format: :SYSTem:ERRor[:NEXT]?

Return value: <ErrorInfo>: “Error Code, Error Information”.

Example: :SYSTem:ERRor[:NEXT]? This example indicates query of error information in the signal generator

Note: for query only.

:SYSTem:PRESet:TYPE <Mode>

Function: This command is used for setting reset modes of the signal generator, which include: Factory, User and Last State. At Factory mode, the instrument will be reset to the default status of the factory; at User mode, the instrument will return to the status designated by the user; at Last State mode, the instrument will return to the last status before resetting.

Setting format: :SYSTem:PRESet:TYPE NORMal|USER|LAST

Query format: :SYSTem:PRESet:TYPE?

Parameter:

<Mode> Discrete data. Options for reset mode are:
NORMal | 0: Factory,
USER | 1: User,
LAST | 2: Last State.

Example: :SYSYem:PRESet:TYPE USER Set the reset mode of the signal generator to User.

Reset status: NORMal

Key Entry: 【System】—>[Reset] —>[Reset Type]

4 Programming example

- Basic operation example 113
- Advanced operation example 117

4.1 Basic operation example

The following demonstrates the use of the VISA library to implement the basic programming of the device, using C++ as an example.

- VISA library 113
- Sample runtime environment 114
- Initialization and configuring default status 115
- Sending set-up command 116
- Reading the status of the measuring device 116

4.1.1 VISA library

VISA is a general term of standard I/O function libraries and their related specifications. VISA library function is a set of functions that can be easily recalled. Its core function enables you to control various devices, regardless of their interface types and the usage of different I/O interface software. These library functions are used to write the driver of the instrument as well as complete the command and data transmission between the computer and the instrument, so as to realize the remote control of the instrument. By initializing the address string (“VISA resource string”), a connection can be established with a device that has a programmable port (LAN, USB, GPBI, RS-232, etc.)

At first, it is necessary to install the VISA library so as to achieve remote control. The VISA library packages the underlying transfer functions of underlying VXI, GPIB, LAN and USB interfaces so that the user can recall them directly. Programming interfaces supported by the signal generator: GPIB, LAN, and RS-232. The use of these interfaces in conjunction with VISA library and programming languages allows remote control of the signal generator. Agilent I/O Library provided by Agilent for users is often used as the underlying I/O library.

Figure 4.1 demonstrates the relationship between the program-controlled interface, the VISA library, the programming language, and the signal generator, using the GPIB interface as an example.

–

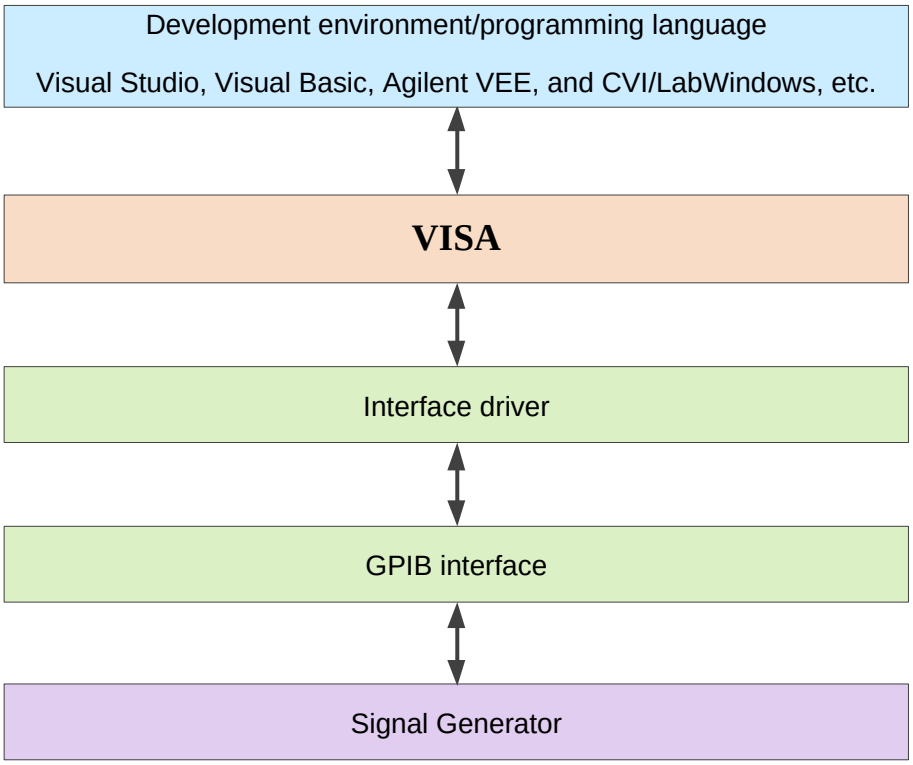

Figure 4.1 Programmable Software and Hardware Layers

4.1.2 Example running environment

4.1.2.1 Configuration requirements

The programming examples described in this chapter have been successfully run on the computers with the following configuration.

- IBM-compatible, Pentium and above PC;
- Windows 2000 or Windows XP operating system;
- Visual Studio 2010/2012 integrated development environment;
- NI's PCI-GPIB interface card or Agilent's GPIB interface card;
- NI's VISA library or Agilent's VISA library;
- GPIB card;
- Network card;
- Available serial ports COM1 and COM2.

4.1.2.2 Included Files

To run a program example written in C/C++, you must contain the required files in the VC6.0 project
If you use VISA library, you must perform the following steps:

- Add visa32.lib to the source file;
- Add visa.h to the header file.

If using the NI-488.2 library, you must do the following operations:

- Add GPIB-32.OBJ file to the source file;
- Add windows.h file to the header file;
- Add Deci-32.h file to the header file.

More detailed information on the NI-488.2 library and the VISA library is available on National Instruments and Agilent websites respectively.

4.1.3 Initialization and default status setting

When the program starts, firstly initialize VISA resource manager, and then enable and establish the communication connection between VISA library and the instrument. The specific steps are as follows:

4.1.3.1 Generation of global variable

First generate global variables to be recalled by other program modules, for example: Instrument handle variable. The following example program shall contain the following global variables:

ViSession source;

ViSession defaultRM;

Const char sourceString [VI_FIND_BUFLEN] = "GPIB0::20::INSTR";

Const sourceTimeout = 10000;

The constant sourceString represents the instrument descriptor, “GPIB0” represents the controller and “20” represents the instrument connected to the controller. If supposing that the instrument is connected with LAN and IP address is “192.168.1.1”, the variable will be:

Const char sourceString [VI_FIND_BUFLEN] = " TCPIP::192.168.1.1::INSTR ";

4.1.3.2 Controller initialization

/***/

The following example shows how to open and establish a communication connection between VISA library and the instrument (specified by instrument descriptor).

//Initialize controller: Open the default resource manager and return to the device handle source

/***/

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

4.1.3.3 Instrument initialization

/***/

The following example shows how to initialize the instrument default status and clear status registers.

/***/

```
void InitDevice()
{
```

```
    ViStatus status;
    long retCnt;
    status = viWrite(source, "*CLS", 4, &retCnt); //Reset status register
    status = viWrite(source, "*RST", 4, &retCnt); //Reset device
    status = viWrite(source, "freq 1ghz", 9, &retCnt); //Set the continuous wave frequency of the signal generator to 1 GHz
}
```

4.1.4 Sending of setting command

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

The following example shows how to set the frequency and amplitude of the 1465 Series signal generator.

```
/*************************************************************************/
void SimpleSettings()
{
    ViStatus status;
    long retCnt;
    Set the frequency to 128 MHz
    status = viWrite(source, "FREQENCY:CW 128MHz", 18, &retCnt);
    //Set the amplitude to -10 dBm
    status = viWrite(source, "POW -10dBm", 10, &retCnt);
}
```

4.1.5 Reading of instrument status

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

The following example shows how to read the setting status of the instrument.

```
/*************************************************************************/
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];

    Check the frequency
    status = viWrite(source, "FREQ:CW?", 8, &retCnt);
    Sleep(10);
    status = viRead(source, rd_Buf_CW, 20, &retCnt);
    //Query amplitude
    status = viWrite(source, "POW?", 4, &retCnt);
    Sleep(10);
    status = viRead(source, rd_Buf_LVL, 20, &retCnt);
```

```
        //Print debugging information
        sprint("Cw is %s", rd_Buf_CW);
        sprint("LEVel is %s", rd_Buf_ LVL);
}
```

4.2 Advanced operation example

- Setting and checking the frequency of LAN interface ····· 117
- Setting and checking the frequency of GPIB interface ·····118

4.2.1 Setting and checking the frequency of LAN interface

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

To use the following examples correctly, you must match your host address with the IP address of the signal source. (The examples of network design in this manual use WINSOCK components to establish socket under VC6.0).

```
/*************************************************************************/
#include "stdafx.h"
#include <afxsock.h>
#include <stdio.h>
#include <stdlib.h>
CSocket sockClient;
  void main()
  {
bool flag;
char buff[100];
if (!AfxSocketInit()) //initialize network port
{
AfxMessageBox("Initialization failed", "ok", MB_OK);
}
else
{
    flag = sockClient.Create();
    if(flag)
{
AfxMessageBox("Socket created successfully", "ok", MB_OK);
}
else
{
            AfxMessageBox("Socket creation failed", "ok", MB_OK);
            sockClient.Close();
}
}
flag=sockClient.Connect(name, 5001);  //Connect network port
```

```
flag=sockClient.Send(“FREQ 1GHz\n”，12，0)； //Set frequency to 1 GHz
if(!flag)
{
    MessageBox("Sending failed"，"ERROR"，MB_OK)；
    exit(0)；
}
cout<<”Display Point Frequency Value”<<end1
flag=sockClient.Send(“FREQ?\n”，6，0)； //Check current frequency
if(!flag)
{
    MessageBox("Sending failed"，"ERROR"，MB_OK)；
    exit(0)；
}
flag= sockClient.Receive(buff，100，0)； //Insert query value into array
if(!flag)
{
    MessageBox("Receive Failed!"，"ERROR"，MB_OK)；
    Exit(0)；
}
sockClient.Close()；
  }
```

4.2.2 Setting and checking the frequency of GPIB interface

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

This example uses the functions of the VISA library to set the point frequency of the signal source outputting 500 MHZ signals and the power of -2 dBm, and query the current frequency and power. Start VC6.0, add the required files, and enter the following code into your .cpp file

```
/*************************************************************************/
#include “stdAfx.h”
#include <visa.h>
#include <iostream>
#include <stdlib.h>
#include <conio.h>
  void main()
  {
  ViSession defaultRM，vi；                    //Declare VISession type variables
  ViStatus vistatus = 0；                      //For device communication
  Char buff[256]；                             //Declare variables that store character data
  int num；              //Declare variables that store integer data
  vistatus = viOpenDefaultRM(&defalutRM); //Open GPIB task, address 19
```

```
vistatus=viOpen(defaultRM，”GPIB::19::INSTR”，VI_NULL，VI_NULL，&vi)；
  if(vistatus)
  {
printf(“Cannot open task, please check device and connect again\n”)；
exit(0)；
  }
  viPrintf(vi，”*RST\n”)；                    //Reset signal source
  viPrintf(vi，”FREQ 500MHZ\n”)；             //Set frequency to 500 MHz
  viPrintf(vi，”FREQ?\n”)；                   //Check frequency
  viScanf(vi，"%t"，buff)；                   //Insert query value into array
  printf(“source CW freq is: %s\n”，buff)；   //Display frequency
  viPrintf(vi，”POW -2dBm\n”)；               //Set power level to -2 dBm
  viPrintf(vi，”POW?\n”)；                    //Check current frequency
  viScanf(vi，"%t"，buff)；                   //Insert query value into array
  printf(“source POW is: %s\n”，buff)；       //Display power
  viClear(vi)；
  viClose(vi)；
  viClose(defaultRM)；
  }
```

5 Error Description

This chapter aims to help you to identify a problem and receive after-sales service. Error information of the signal generator is also introduced.

- Error information 121
- How to fix errors 123

5.1 Error information

The signal generator uses two methods to record the errors during measurement: the front panel operation interface displays the error message queue and the SCPI (remote control mode) error message queue, which are separately stored and managed.

- Description of error messages 121

5.1.1 Description of error information

- Local error information 121
- Remote error information 122

5.1.1.1 Local error information

- Error information view 121
- Description of error messages 121

1) Error information view

Operation method through interface:

If any error information is displayed in the lower right corner of the signal generator during use, it indicates that the operation of the software or hardware of the instrument is defective. You can determine the type of the fault according to the error code and then take appropriate measures for troubleshooting.

The error display area of the signal generator can only display one piece of error information at one time. As the instrument may have a number of problems at the same time, all error information can be seen by executing the following operations:

Step 1. Press 【System】, then press [Error List]. A window appears with a list of errors.

Step 2. The prompt information is displayed in the window.

Step 3. Use the mouse to browse the error information and close the dialog window.

Step 4. Select the Clear List button to clear the history error information.

2) Description of error messages

If errors are detected during measurement by the signal generator, the warning or error information (error abbreviation + detailed description of error) will be displayed on the right side of the status indication area.

Table 5.2 Local Error Information List

Error Key Field	Description
Data error	Only for digital-based band interface: Digital I/Q input data error.

5.1 Error information

FIFO overload	Only for digital-based band interface: High external input sampling rate.
Input overload	Signal power of RF input port is out of specified range.
No reference	The signal generator does not detect any required external reference single input.
OVEN	OCXO has not reached the working temperature. Generally, after turning on the machine for several minutes, OCXO will enter normal working condition and the error message will disappear.
No ALC	Excess power or no power.
Loss of lock of reference loop	The signal generator's internal loop signal involves loss of lock.
Loss of lock of decimal loop	The signal generator's fractional loop signal involves loss of lock.
Loss of lock of high purity loop	The signal generator's fractional loop signal involves loss of lock.
Loss of lock of YO	The signal generator's YO loop signal involves loss of lock.

5.1.2.2 Remote error messages

1) Format and description of error information

Under the remote control mode, the error information will be recorded in the error/event queue of the status reporting system. The error information can be queried with the command "SYSTem:ERRor?", and the format is as follows:

"<Error code>, "<Error information in error queue>; <Detailed description of error information>"

E.g.:

"-110, "Data out of Range; Inputted Parameter out of Lower Bound."

The information of the negative error code defined in SCPI standard will not be described in details here.

2) Type of error information

Each error corresponds to only one type of error message. The following classification describes the types of error messages (8257):

- **Query error (–499 to –400):** Instrument output queue control detects the message exchange protocol error described in IEEE 488.2, Chapter 6. Now the event state register's query error bit (bit2) is set (refer to IEEE 488.2, 6.5 for details). Now data can not be read successfully from output queue.
- **Device-specific error（-399 to –300, 201 to 703, and 800 to 810）:** Instrument operation fails, which may be caused by abnormal hardware/firmware state, and this error code is often used in instrument self test. Now the event state register's instrument feature error bit (bit3) is set.
- **Execution error (-299 to –200):** Indicates that the device has encountered an error during the measurement process. Now the event state register's execution error bit (bit4) is set.
- **Command error (-199 to –100):** The instrument detects syntactic error during command parsing, which is caused by incorrect command format. Now the event state register's command error bit (bit5) is set.

5.2 Repair Method

- Contact us 123
- Packaging and mailing 123

5.2.1 Contact us

If there is a problem with the 1465 vector signal generator, first observe and save the error message, and proceed to solve the problem. Please contact our service center according to the contact information below in the case of fault unsolved, and provide collected error information, so that we can help you to solve your problem as early as possible.

Contact information:

Tel.:	(86) 0532-86896691
E-mail:	sales@ceyear.com
Website:	**www.ceyear.com**
Postal code:	**266555**
Address:	**No.98, Xiangjiang Road, Qingdao City, China**

5.2.2 Packaging and mailing

If you have any intricate problem with your signal generator, contact us by phone or fax. After necessary repair is confirmed, pack the signal generator with original packaging materials and put it in a box before the following steps.

1) A detailed description of fault symptom should be provided and attached to the signal generator.
2) The signal generator should be packed by the original packaging materials, so as to minimize damages.
3) Four corners of external packaging box should be cushioned before the instrument is put in the box.
4) The packaging box should be sealed up by adhesive tape, and then fastened by nylon tape.
5) Words “Fragile！ Do not touch！ Handle with care！”should be marked on the box.
6) The instrument should be consigned as a precise instrument.
7) The copies of all transport documents should be retained.

CAUTION

Precautions for packaging

Packaging the signal generator with other materials may damage the instrument. It is forbidden to use polystyrene beads as the packaging material because they can't fully protect the instrument and may damage the instrument after being sucked into the instrument fan by the static electricity.

NOTE

8)

Packaging and transport of instrument

When transporting or handling the instrument (for example, damage during shipment), you shall strictly observe the precautions described in “3.1.1.1 Unpacking” of the User Manual.

Appendixes

- Appendix A: Lookup Table of SCPI by Subsystem 125
- Appendix B: Lookup Table of Error Information 136

Appendix A Lookup Table of SCPI by Subsystem

Attached Table 1: Lookup Table of SCPI for 1465 Series Signal Generator

Index	Command	Function
1	*IDN?	General instruction
2	*RCL	General instruction
3	*RST	General instruction
4	*SAV	General instruction
5	*TRG	General instruction
6	:OUTPut:BLANking[:STATe](?)	Set the output blanking
7	:OUTPut[:STATe](?)	Set the RF output on-off
8	OUTPut:MODulation[:STATe](?)	Set modulation master switch
9	[:SOURce]:FREQuency[:CW|FIXed](?)	Set the signal generator output frequency
10	[:SOURce]:FREQuency:MODE(?)	Set the frequency generation mode
11	[:SOURce]:FREQuency:MULTiplier（?）	Set the frequency multiplication
12	[:SOURce]:FREQuency:OFFSet（?）	Set the frequency offset
13	[:SOURce]:FREQuency:REFerence（?）	Set the relative frequency
14	[:SOURce]:FREQuency:REFerence:STATe（?）	Set the relative frequency on-off
15	[:SOURce]:FREQuency:STEP(?)	Set the frequency step
16	[:SOURce]:FREQuency:STARt(?)	Set the step sweep start frequency
17	[:SOURce]:FREQuency:STOP(?)	Set the step sweep stop frequency
18	[:SOURce]:FREQuency[:CW|FIXed]:AUTO(?)	Set frequency follower switch
19	[:SOURce]:POWer:ALC:LEVel(?)	Set the ALC level
20	[:SOURce]:POWer:ALC:SEARch(?)	Set the power search mode

Appendix A: Lookup Table of SCPI by Subsystem

21	[:SOURce]:POWer:ALC:SOURce(?)	Set the power amplitude level control mode
22	[:SOURce]:POWer:ALC:SOURce:EXTernal:COUPling(?)	Set the external detection coupling coefficient
23	[:SOURce]:POWer:ALC[:STATe](?)	Set the ALC loop state
24	[:SOURce]:POWer:ATTenuation(?)	Set the power attenuation
25	[:SOURce]:POWer:ATTenuation:AUTO(?)	Set the power attenuation on-off
26	[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude](?)	Set the power level
27	[:SOURce]:POWer[:LEVel][:IMMediate]:OFFSet(?)	Set the power offset
28	[:SOURce]:POWer:REFerence(?)	Set the relative power
29	[:SOURce]:POWer:REFerence:STATe(?)	Set the relative power on-off
30	[:SOURce]:POWer:STEP(?)	Set the power step
31	[:SOURce]:POWer:ALC:BANDwidth|BWIDth	Set ALC loop bandwidth
32	[:SOURce]:POWer:ALC:BANDwidth|BWIDth:AUTo	Set ALC loop bandwidth selection mode
33	[:SOURce]:LIST:DIRection(?)	Set the list sweep direction
34	[:SOURce]:LIST:DWELl	Set dwell times of all points
35	[:SOURce]:LIST:FREQuency	Set the list sweep frequency
36	[:SOURce]:LIST:FILL:POINts(?)	Set the list sweep points
37	[:SOURce]:LIST:FILL:STARt(?)	Set the list sweep start frequency
38	[:SOURce]:LIST:FILL:STOP(?)	Set the list sweep stop frequency
39	[:SOURce]:LIST:POWer	Set the list sweep power
40	[:SOURce]:LIST:RETRace(?)	Set the list sweep retrace on-off
41	[:SOURce]:LIST:TRIGger:SOURce(?)	Set the list sweep trigger source
42	[:SOURce]:LIST:FILL:POWer(?)	Set the power

		offsets of all points in the list
43	[:SOURce]:LIST:FILL:DWELl(?)	Set the dwell times of all points in the list
44	[:SOURce]:LIST:FILL:EXECute	Complete the list fill
45	[:SOURce]:LIST:DELete	Delete the list point
46	[:SOURce]:LFOutput:AMPLitude(?)	Set the low frequency amplitude
47	[:SOURce]:LFOutput:FREQuency(?)	Set the low frequency
48	[:SOURce]:LFOutput:FREQuency:ALTernate(?)	Set the sweep frequency sine stop frequency and double-sine frequency 2
49	[:SOURce]:LFOutput:FREQuency:ALTernate:AMPLitude:PRECent(?)	Set the percent of amplitude occupied by frequency 2 in the case of double-sine
50	[:SOURce]:LFOutput: NOISe(?)	Set the low frequency output noise type
51	[:SOURce]:LFOutput:RAMP(?)	Set the direction of the low frequency zigzagform
52	[:SOURce]:LFOutput:SHAPe(?)	Set the low frequency waveform
53	[:SOURce]:LFOutput:STATe(?)	Set the low frequency on-off
54	[:SOURce]:LFOutput:SWEep:TIME(?)	Set the sweep frequency sine sweep time
55	[:SOURce]:SWEep:DIRection(?)	Set the step sweep direction
56	[:SOURce]:SWEep:DWELl(?)	Set the step dwell time
57	[:SOURce]:SWEep:POINts(?)	Set the step sweep points
58	[:SOURce]:SWEep:STEP(?)	Set the step amount of the step sweep frequency
59	[:SOURce]:SWEep:TIME:AUTO(?)	Set the ramp

Appendix A: Lookup Table of SCPI by Subsystem

		sweep time auto manual
60	[:SOURce]:SWEep:TIME (?)	Set the ramp sweep time
61	[:SOURce]:SWEep:GENeration(?)	Set the sweep type
62	[:SOURce]:SWEep:TRIGger:SOURce(?)	Set the step sweep trigger source
63	[:SOURce]:PULM:EXTernal:POLarity(?)	Pulse input polarity status
64	[:SOURce]:PULM:INTernal:DELay(?)	Set the pulse delay
65	[:SOURce]:PULM:INTernal:FREQuency（？）	Set the pulse repetition frequency
66	[:SOURce]:PULM:INTernal:PERiod(?)	Set the pulse modulation period
67	[:SOURce]:PULM:INTernal:PWIDth(?)	Set the pulse width of the pulse modulation
68	[:SOURce]:PULM:SOURce(?)	Set the pulse source
69	[:SOURce]:PULM:STATe(?)	Set the pulse modulation on-off
70	[:SOURce]:PULM:INTernal:JITTered:MODE(?)	Set pulse period jittered mode
71	[:SOURce]:PULM:INTernal:JITTered:PERCent(?)	Set pulse dither percent
72	[:SOURce]:PULM:INTernal:PTRain:DATA	Set pulse train
73	[:SOURce]:PULM:INTernal:PTRain:DELete	Delete any index point from pulse train list
74	[:SOURce]:PULM:INTernal:PTRain:POINts(?)	Check current pulse train points
75	[:SOURce]:PULM:INTernal:PTRain:PRESet	Delete all points from the pulse train list
76	[:SOURce]:PULM:INTernal:SLIDing:STEP(?)	Set pulse sliding step
77	[:SOURce]:PULM:INTernal:SLIDing:POINts(?)	Set pulse sliding point
78	[:SOURce]:PULM:INTernal:STAGger:INSert	Insert staggered pulse
79	[:SOURce]:PULM:INTernal:STAGger:POINts(?)	Check number of points in the staggered pulse list

80	[:SOURce]:PULM:INTernal:STAGger:DELete(?)	Delete any index point from staggered pulse list
81	[:SOURce]:PULM:INTernal:STAGger:PRESet	Clear all points from the staggered pulse repetition frequency list
82	[:SOURce]:PULM:LFM:BWIDth(?)	Set linear frequency modulation bandwidth
83	[:SOURce]:PULM:LFM:DIRection(?)	Set linear frequency modulation direction
84	[:SOURce]:PULM:LFM:STATe(?)	Set linear frequency modulation status
85	[:SOURce]:AM[:DEPth:]EXPonential(?)	Set the exponential AM depth
86	[:SOURce]:AM[:DEPth][:LINear](?)	Set the linear AM depth
87	[:SOURce]:AM:INTernal:FREQuency(?)	Set the AM modulation rate
88	[:SOURce]:AM:INTernal:FREQuency:ALTernate(?)	Set the sweep frequency sine stop frequency or double-sine frequency 2
89	[:SOURce]:AM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent(?)	Set the percent of amplitude occupied by double-sine frequency 2
90	[:SOURce]:AM:INTernal:NOISe(?)	Set the AM waveform noise type
91	[:SOURce]:AM:INTernal:RAMP(?)	Set the AM zigzag direction
92	[:SOURce]:AM:INTernal:SHAPe(?)	Set the internal AM source waveform
93	[:SOURce]:AM:INTernal:SWEep:TIME(?)	Set the AM sweep frequency sine sweep time
94	[:SOURce]:AM:MODE(?)	Set the depth modulation on-off
95	[:SOURce]:AM:SOURce(?)	Set the AM input

Appendix A: Lookup Table of SCPI by Subsystem

		selection
96	[:SOURce]:AM:STATe(?)	Set the amplitude modulation on-off
97	[:SOURce]:AM:TYPE(?)	Set the AM type
98	[:SOURce]:FM:DEViation(?)	Set the internal FM offset
99	[:SOURce]:FM:INTernal:FREQuency(?)	Set the FM modulation rate or sweep frequency sine start frequency and double-sine frequency 1
100	[:SOURce]:FM:INTernal:FREQuency:ALTernate(?)	Set the sweep frequency sine stop frequency or double-sine frequency 2
101	[:SOURce]:FM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent(?)	Set the percent of amplitude occupied by double-sine frequency
102	[:SOURce]:FM:INTernal:NOISe(?)	Set the FM noise type
103	[:SOURce]:FM:INTernal:RAMP(?)	Set the FM zigzag direction
104	[:SOURce]:FM:INTernal:SHAPe(?)	Set the FM waveform
105	[:SOURce]:FM:INTernal:SWEep:TIME(?)	Set the sweep frequency sine sweep time
106	[:SOURce]:FM:SOURce(?)	Set the FM source
107	[:SOURce]:FM:STATe(?)	Set the FM on-off
108	[:SOURce]:PM:BANDwidth|BWIDth(?)	Set the phase modulation bandwidth
109	[:SOURce]:PM:DEViation(?)	Set the phase modulation offset
110	[:SOURce]:PM:INTernal:FREQuency(?)	Set the phase modulation rate and sweep frequency sine start frequency or double-sine frequency 1
111	[:SOURce]:PM:INTernal:FREQuency:ALTernate(?)	Set the sweep frequency sine stop

		frequency or double-sine frequency 2
112	[:SOURce]:PM:INTernal:FREQuency:ALTernate:AMPLitude:PERCent(?)	Set the percent of amplitude occupied by double-sine frequency 2
113	[:SOURce]:PM:INTernal:NOISe(?)	Set the phase modulation noise type
114	[:SOURce]:PM:INTernal:RAMP(?)	Set the phase modulation zigzag direction
115	[:SOURce]:PM:INTernal:SHAPe(?)	Set the phase modulation waveform
116	[:SOURce]:PM:INTernal:SWEep:TIME(?)	Set the phase modulation sweep frequency sine sweep time
117	[:SOURce]:PM:SOURce(?)	Set the phase modulation source
118	[:SOURce]:PM:STATe(?)	Set the phase modulation on-off
119	[:SOURce]:DM:IQADjustment::GAIN(?)	Set internal IQ gain balance
120	[:SOURce]:DM:IQADjustment:IOFFse(?))	Set the I channel offset value
121	[:SOURce]:DM:IQADjustment:QOFFse(?)	Set the Q channel offset value
122	[:SOURce]:DM:IQADjustment:QSKew(?)	Set phase angle between IQ vectors
123	[:SOURce]:DM:IQADjustment[:STATe](?)	Set IQ adjustment state
124	[:SOURce]:DM:MODulation:ATTenuation(?)	Set IQ signal attenuation
125	[:SOURce]:DM:MODulation:ATTenuation:AUTO(?)	Set IQ signal attenuation state
126	[:SOURce]:DM:STATe(?)	Set IQ modulation state
127	[:SOURce]:DM:EXTernal:BWIDth[:STATe](?)	Set external bandwidth IQ input state
128	[:SOURce]:DM:IQADjustment:OUTPut[:STATe](?)	Set I/Q input modulation state
129	[:SOURce]:DM:IQADjustment:OUTPut:ATTen(?)	Set I/Q output

Appendix A: Lookup Table of SCPI by Subsystem

		adjustment attenuation
130	[:SOURce]:DM:IQADjustment:OUTPut:GAIN(?)	Set I/Q output adjustment gain balance
131	[:SOURce]:DM:IQADjustment:OUTPut:IOFFset(?)	Set I/Q output adjustment I offset
132	[:SOURce]:DM:IQADjustment:OUTPut:UIOFfset(?)	Set I/Q output adjustment I/ offset
133	[:SOURce]:DM:IQADjustment:OUTPut:QOFFset(?)	Set I/Q output adjustment Q offset
134	[:SOURce]:DM:IQADjustment:OUTPut:UQOFfset(?)	Set I/Q output adjustment Q/ offset
135	[:SOURce]:DM:IQADjustment:OUTPut:SKE W (?)	Set I/Q output adjustment orthority offset
136	[:SOURce]:RADio:CUSTom:ALPHa(?)	Set the baseband filter factor
137	[:SOURce]:RADio:CUSTom:DATA(?)	Set the baseband modulation signal data source
138	[:SOURce]:RADio:CUSTom:FILTer(?)	Set the baseband filter type
139	[:SOURce]:RADio:CUSTom:MODulation:FSK[:DEViation](?)	Set the FM offset when the baseband modulation type is FSK mode
140	[:SOURce]:RADio:CUSTom:MODulation:MSK:PHASe(?)	Set the phase modulation offset when the baseband modulation type is MSK mode
141	[:SOURce]:RADio:CUSTom:MODulation[:TYPE](?)	Set the baseband modulation format
142	[:SOURce]:RADio:CUSTom:SRATe(?)	Set the baseband symbol rate
143	[:SOURce]:RADio:CUSTom:STATe(?)	Set the baseband on-off
144	[:SOURce]:RADio:CUSTom:POLarity[:ALL](?)	Set phase polarity of baseband signal
145	[:SOURce]:RADio:CUSTom:DENCode(?)	Set differential encoding command status
146	[:SOURce]:RADio:CUSTom:VCO:CLOCk(?)	Set baseband sample clock type

147	[:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay(?)	Set delay time of external trigger source
148	[:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:DELay:STATe(?)	Set delay status of external trigger source
149	[:SOURce]:RADio:CUSTom:TRIGger:EXTernal:SOURce:SLOPe(?)	Set polarity of external trigger source
150	[:SOURce]:RADio:CUSTom:TRIGger:SOURce(?)	Set baseband trigger source type
151	[:SOURce]:RADio:CUSTom:TRIGger:TYPE(?)	Set baseband trigger source trigger type
152	[:SOURce]:RADio:CUSTom:TRIGger:TYPE:CONTinuous:TYPE(?)	Set baseband continuous trigger type
153	[:SOURce]:RADio:CUSTom:TRIGger:TYPE:GATE:ACTive(?)	Set baseband control trigger mode
154	[:SOURce]:RADio:MTONe:ARB:SETup	Select multi-tone file loading
155	[:SOURce]:RADio:MTONe:ARB:SETup:STORe	Store multi-tone file
156	[:SOURce]:RADio:MTONe:ARB: SETup:TABLe	Set multi-tone waveform sequence
157	[:SOURce]:RADio:MTONe:ARB: SETup:TABLe:FSPacing(?)	Set multi-tone frequency spacing
158	[:SOURce]:RADio:MTONe:ARB: SETup:TABLe:NTONes(?)	Set counts of multi-tones
159	[:SOURce]:RADio:MTONe:ARB: SETup:TABLe:PHASe:INITialize(?)	Set initial phase type of multi-tone
160	[:SOURce]:RADio:MTONe:ARB: SETup:TABLe:PHASeINITialize:SEED(?)	Set relationship between multi-tone phases
161	[:SOURce]:RADio:MTONe:ARB: SETup:TABLe:ROW(?)	Set specific row of multi-tone parameter in the multi-tone modulation list
162	[:SOURce]:RADio:MTONe:ARB[:STATe](?)	Set multi-tone state
163	[:SOURce]:RADio:TTONe:ARB:ALIGnment(?)	Set bi-tone alignment
164	[:SOURce]:RADio:TTONe:ARB:FSPacing(?)	Set bi-tone frequency spacing

Appendix A: Lookup Table of SCPI by Subsystem

165	[:SOURce]:RADio:TTONe:ARB[:STATe](?)	Set bi-tone status
166	[:SOURce]:RADio:ARB:MODE(?)	Set ARB mode
167	[:SOURce]:RADio:ARB[:STATe](?)	Set ARB state
168	[:SOURce]:RADio:ARB:SEQuence	Load arbitrary wave file
169	[:SOURce]:RADio:ARB:SEQuence:CLOCk(?)	Set ARB clock type
170	[:SOURce]:RADio:ARB:SCLock:RATE(?)	Set ARB clock frequency rate
171	[:SOURce]:RADio:ARB:TRIGger:TYPE(?)	Set ARB trigger type
172	[:SOURce]:RADio:ARB:TRIGger:TYPE:CONTinuous[:TYPE](?)	Set arbitrary wave continuous trigger type
173	[:SOURce]:RADio:ARB:TRIGger:TYPE:SINGle(?)	Set arbitrary wave single-trigger type
174	[:SOURce]:RADio:ARB:TRIGger:TYPE:SADVance[:TYPE](?)	Set ARB segmental trigger type
175	[:SOURce]:RADio:ARB:TRIGger:TYPE:GATE:ACTive(?)	Set ARB gate trigger mode
176	[:SOURce]:RADio:ARB:TRIGger:SOURce(?)	Set ARB trigger source
177	[:SOURce]:RADio:ARB:VCO:CLOCk(?)	Set ARB sample clock
178	[:SOURce]:RADio:ARB:EXTernal:CLOCk:RATE(?)	Set ARB external clock trigger rate
179	:MEMory:COPY:NAME	Copy the files in the signal generator
180	:MEMory:DLEete:NAME	Delete the user files
181	:MEMory:MOVE	Rename the signal generator file name
182	:MEMory:DATA	Transfer data files
183	:ROSCillator:ADJust:REFerence(?)	Set the signal generator internal reference
184	:DIAGnostic:INFormation:CCOunt:PON?	This command is used to query the cumulative number of times the instrument is powered on
185	:DIAGnostic:INFormation:OTIMe?	This command is used to query the instrument

		firmware date and time stamp
186	:DIAGnostic:SNUM?	Read the signal generator system serial number
187	:SYSTem:COMMunicate:GPIB:ADDRess(?)	Set the signal generator GPIB address
188	:SYSTem:COMMunicate:GTLocal	Set the signal generator to local mode
189	:SYSTem:DEVice:LANGuage(?)	Set interface language of device
190	:SYSTem:COMMunicate:LAN:IP(?)	Set IP address of device
191	:SYSTem:COMMunicate:LAN:SUBNet(?)	Set subnet mask of device
192	:SYSTem:COMMunicate:LAN:GATeway(?)	Set default gateway of device
193	:SYSTem:ERRor[:NEXT](?)	Check device error information
194	:SYSTem:PRESet:TYPE(?)	Set device reset status

Appendix B Lookup Table of Error Information

Attached Table 3 Table of Local Error Information

Error Key Field	Error Description
No ALC	Excess power or no power.
Time base not heated	The internal 10 MHz time base of the signal generator is not at the working temperature.
Loss of lock of reference loop	The signal generator's internal loop signal involves loss of lock.
Loss of lock of decimal loop	The signal generator's fractional loop signal involves loss of lock.
Loss of lock of high purity loop	The signal generator's fractional loop signal involves loss of lock.
Loss of lock of YO	The signal generator's YO loop signal involves loss of lock.
External reference	The signal generator is at an external reference state. This is not an error.