



RIGOL



# DSG5000 Series

## Microwave Signal Generator

### Programming Guide

Jun.2022

## **Guaranty and Declaration**

### **Copyright**

© 2022 RIGOL TECHNOLOGIES CO., LTD. All Rights Reserved.

### **Trademark Information**

RIGOL® is the trademark of RIGOL TECHNOLOGIES CO., LTD.

### **Notices**

- RIGOL products are covered by P.R.C. and foreign patents, issued and pending.
- RIGOL reserves the right to modify or change parts of or all the specifications and pricing policies at the company's sole decision.
- Information in this publication replaces all previously released materials.
- Information in this publication is subject to change without notice.
- RIGOL shall not be liable for either incidental or consequential losses in connection with the furnishing, use, or performance of this manual, as well as any information contained.
- Any part of this document is forbidden to be copied, photocopied, or rearranged without prior written approval of RIGOL.

### **Product Certification**

RIGOL guarantees that this product conforms to the national and industrial standards in China as well as the ISO9001:2015 standard and the ISO14001:2015 standard. Other international standard conformance certifications are in progress.

### **Contact Us**

If you have any problem or requirement when using our products or this manual, please contact RIGOL.

E-mail: [service@rigol.com](mailto:service@rigol.com)

Website: <http://www.rigol.com>

| Section  | Description                                      | Page |
|----------|--|------|
| 1        | Document Overview.....                           | 1    |
| 2        | SCPI Command Overview.....                       | 3    |
| 3        | Command System.....                              | 6    |
| 3.1      | :SOURce Commands.....                            | 6    |
| 3.1.1    | [:SOURce][:RF]:FREQuency Commands.....           | 6    |
| 3.1.1.1  | [:SOURce][:RF]:FREQuency.....                    | 6    |
| 3.1.1.2  | [:SOURce][:RF]:FREQuency:OFFSet.....             | 7    |
| 3.1.2    | [:SOURce][:RF]:LEVel Commands.....               | 8    |
| 3.1.2.1  | [:SOURce][:RF]:LEVel.....                        | 8    |
| 3.1.2.2  | [:SOURce][:RF]:LEVel:LIMit.....                  | 9    |
| 3.1.2.3  | [:SOURce][:RF]:LEVel:OFFSet.....                 | 10   |
| 3.1.3    | [:SOURce][:RF]:PHASe Commands.....               | 11   |
| 3.1.3.1  | [:SOURce][:RF]:PHASe.....                        | 11   |
| 3.1.3.2  | [:SOURce][:RF]:PHASe:RESet.....                  | 12   |
| 3.1.4    | [:SOURce][:RF]:CORRection Commands.....          | 13   |
| 3.1.4.1  | [:SOURce][:RF]:CORRection:FLATness[:STATe].....  | 13   |
| 3.1.4.2  | [:SOURce][:RF]:CORRection:FLATness:COUNt.....    | 13   |
| 3.1.4.3  | [:SOURce][:RF]:CORRection:FLATness:LIST.....     | 14   |
| 3.1.4.4  | [:SOURce][:RF]:CORRection:FLATness:DELeTe.....   | 15   |
| 3.1.4.5  | [:SOURce][:RF]:CORRection:FLATness:PAIR.....     | 16   |
| 3.1.4.6  | [:SOURce][:RF]:CORRection:FLATness:LOAD.....     | 16   |
| 3.1.4.7  | [:SOURce][:RF]:CORRection:FLATness:SAVe.....     | 17   |
| 3.1.5    | [:SOURce][:RF]:Sweep Commands.....               | 18   |
| 3.1.5.1  | [:SOURce][:RF]:SWEep:DIRection.....              | 18   |
| 3.1.5.2  | [:SOURce][:RF]:SWEep:EXECute.....                | 18   |
| 3.1.5.3  | [:SOURce][:RF]:SWEep:LIST:ADDList.....           | 19   |
| 3.1.5.4  | [:SOURce][:RF]:SWEep:LIST:CPOint.....            | 20   |
| 3.1.5.5  | [:SOURce][:RF]:SWEep:LIST:DELeTe.....            | 21   |
| 3.1.5.6  | [:SOURce][:RF]:SWEep:LIST:LIST.....              | 21   |
| 3.1.5.7  | [:SOURce][:RF]:SWEep:LIST:INITialize:FSTep.....  | 22   |
| 3.1.5.8  | [:SOURce][:RF]:SWEep:LIST:INITialize:PRESet..... | 23   |
| 3.1.5.9  | [:SOURce][:RF]:SWEep:LIST:LOAD.....              | 24   |
| 3.1.5.10 | [:SOURce][:RF]:SWEep:LIST:SAVe.....              | 24   |

|          |  |    |
|----------|--|----|
| 3.1.5.11 | [:SOURce][:RF]:SWEep:STATe.....                | 25 |
| 3.1.5.12 | [:SOURce][:RF]:SWEep:TYPE.....                 | 26 |
| 3.1.5.13 | [:SOURce][:RF]:SWEep:MODE.....                 | 26 |
| 3.1.5.14 | [:SOURce][:RF]:SWEep:RESet[:ALL].....          | 27 |
| 3.1.5.15 | [:SOURce][:RF]:SWEep:POINt:TRIGger:TYPE.....   | 28 |
| 3.1.5.16 | [:SOURce][:RF]:SWEep:SWEep:TRIGger:TYPE.....   | 29 |
| 3.1.5.17 | [:SOURce][:RF]:SWEep:STEP:DWELL.....           | 31 |
| 3.1.5.18 | [:SOURce][:RF]:SWEep:STEP:POINts.....          | 31 |
| 3.1.5.19 | [:SOURce][:RF]:SWEep:STEP:START:FREQuency..... | 32 |
| 3.1.5.20 | [:SOURce][:RF]:SWEep:STEP:START:LEVel.....     | 33 |
| 3.1.5.21 | [:SOURce][:RF]:SWEep:STEP:STOP:FREQuency.....  | 34 |
| 3.1.5.22 | [:SOURce][:RF]:SWEep:STEP:STOP:LEVel.....      | 35 |
| 3.1.5.23 | [:SOURce][:RF]:SWEep:STEP:SHAPE.....           | 36 |
| 3.1.5.24 | [:SOURce][:RF]:SWEep:STEP:SPACing.....         | 37 |
| 3.1.6    | [:SOURce][:RF]:AM Commands.....                | 38 |
| 3.1.6.1  | [:SOURce][:RF]:AM:DEPTH.....                   | 38 |
| 3.1.6.2  | [:SOURce][:RF]:AM:FREQuency.....               | 38 |
| 3.1.6.3  | [:SOURce][:RF]:AM:WAVEform.....                | 39 |
| 3.1.6.4  | [:SOURce][:RF]:AM:STATe.....                   | 40 |
| 3.1.6.5  | [:SOURce][:RF]:AM:SOURce.....                  | 41 |
| 3.1.6.6  | [:SOURce][:RF]:AM:EQUal:LEVel.....             | 42 |
| 3.1.6.7  | [:SOURce]:AM:EXT:COUP.....                     | 42 |
| 3.1.6.8  | [:SOURce]:AM:EXT:IMP.....                      | 43 |
| 3.1.7    | [:SOURce][:RF]:FM Commands.....                | 44 |
| 3.1.7.1  | [:SOURce][:RF]:FM:DEVIation.....               | 44 |
| 3.1.7.2  | [:SOURce][:RF]:FM:FREQuency.....               | 45 |
| 3.1.7.3  | [:SOURce][:RF]:FM:WAVEform.....                | 46 |
| 3.1.7.4  | [:SOURce][:RF]:FM:STATe.....                   | 46 |
| 3.1.7.5  | [:SOURce][:RF]:FM:SOURce.....                  | 47 |
| 3.1.7.6  | [:SOURce]:FM:EXT:COUP.....                     | 48 |
| 3.1.7.7  | [:SOURce]:FM:EXT:IMP.....                      | 49 |
| 3.1.8    | [:SOURce][:RF]:PM Commands.....                | 50 |
| 3.1.8.1  | [:SOURce][:RF]:PM:DEVIation.....               | 50 |
| 3.1.8.2  | [:SOURce][:RF]:PM:FREQuency.....               | 50 |
| 3.1.8.3  | [:SOURce][:RF]:PM:WAVEform.....                | 51 |
| 3.1.8.4  | [:SOURce][:RF]:PM:STATe.....                   | 52 |
| 3.1.8.5  | [:SOURce][:RF]:PM:SOURce.....                  | 53 |

|          |   |    |
|----------|---|----|
| 3.1.8.6  | [:SOURce]:PM:EXT:COUP.....                              | 54 |
| 3.1.8.7  | [:SOURce]:PM:EXT:IMP.....                               | 54 |
| 3.1.9    | [:SOURce]:[RF]:PULM Commands.....                       | 55 |
| 3.1.9.1  | [:SOURce]:[RF]:PULM:MODE.....                           | 55 |
| 3.1.9.2  | [:SOURce]:[RF]:PULM:OUT:STATe.....                      | 56 |
| 3.1.9.3  | [:SOURce]:[RF]:PULM:PERiod.....                         | 57 |
| 3.1.9.4  | [:SOURce]:[RF]:PULM:POLarity.....                       | 58 |
| 3.1.9.5  | [:SOURce]:[RF]:PULM:SOURce.....                         | 58 |
| 3.1.9.6  | [:SOURce]:[RF]:PULM:STATe.....                          | 59 |
| 3.1.9.7  | [:SOURce]:[RF]:PULM:TRAI:n:LIST:COUNT.....              | 60 |
| 3.1.9.8  | [:SOURce]:[RF]:PULM:TRAI:n:LIST:INSERt.....             | 61 |
| 3.1.9.9  | [:SOURce]:[RF]:PULM:TRAI:n:LIST:DELeTe.....             | 61 |
| 3.1.9.10 | [:SOURce]:[RF]:PULM:TRAI:n:LIST:GET.....                | 62 |
| 3.1.9.11 | [:SOURce]:[RF]:PULM:TRAI:n:LIST:RUN.....                | 63 |
| 3.1.9.12 | [:SOURce]:[RF]:PULM:TRAI:n:LIST:LOAD.....               | 63 |
| 3.1.9.13 | [:SOURce]:[RF]:PULM:TRAI:n:LIST:SAVe.....               | 64 |
| 3.1.9.14 | [:SOURce]:[RF]:PULM:TRIGGer:DELaY.....                  | 65 |
| 3.1.9.15 | [:SOURce]:[RF]:PULM:TRIGGer:EXTeRnal:GATE:POLarity..... | 65 |
| 3.1.9.16 | [:SOURce]:[RF]:PULM:TRIGGer:EXTeRnal:SLOPe.....         | 66 |
| 3.1.9.17 | [:SOURce]:[RF]:PULM:TRIGGer:MODE.....                   | 67 |
| 3.1.9.18 | [:SOURce]:[RF]:PULM:WIDTh.....                          | 69 |
| 3.1.10   | [:SOURce]:[RF]:OUTPut Commands.....                     | 70 |
| 3.1.10.1 | [:SOURce]:[RF]:OUTPut[:STATe].....                      | 70 |
| 3.1.10.2 | [:SOURce]:RFALL:OUTPut[:STATe].....                     | 70 |
| 3.1.11   | [:SOURce]:[RF]:MODulation Commands.....                 | 71 |
| 3.1.11.1 | [:SOURce]:[RF]:MODulation:STATe.....                    | 71 |
| 3.1.11.2 | [:SOURce]:RFALL:MODulation:STATe.....                   | 72 |
| 3.1.12   | :SOURce:ROSCillator:HCLock Commands.....                | 72 |
| 3.1.12.1 | :SOURce:ROSCillator:HCLock:OUTPut:STATe.....            | 72 |
| 3.1.12.2 | :SOURce:ROSCillator:HCLock:TYPE.....                    | 73 |
| 3.1.13   | [:SOURce]:[RF]:INPut:TRIGGer:SLOPe.....                 | 74 |
| 3.2      | :TRIGGer Commands.....                                  | 74 |
| 3.2.1    | :TRIGGer[:RF]:PULM[:IMMediate].....                     | 75 |
| 3.2.2    | :TRIGGer:RFALL:PULM[:IMMediate].....                    | 75 |
| 3.2.3    | :TRIGGer[:RF][:SWEep][:IMMediate].....                  | 76 |
| 3.2.4    | :TRIGGer:RFALL[:SWEep][:IMMediate].....                 | 76 |
| 3.2.5    | :TRIGGer[:RF]:PULM:KEY.....                             | 77 |

|        |  |    |
|--------|--|----|
| 3.2.6  | :TRIGger:RFAL:PULM:KEY.....                    | 77 |
| 3.2.7  | :TRIGger[:RF][:SWEep]:KEY.....                 | 78 |
| 3.2.8  | :TRIGger:RFAL[:SWEep]:KEY.....                 | 78 |
| 3.3    | :UNIT Command.....                             | 79 |
| 3.3.1  | :UNIT[:RF]:POWER.....                          | 79 |
| 3.4    | :SYSTem Commands.....                          | 79 |
| 3.4.1  | :SYSTem:COMMunication:LAN:DHCP.....            | 80 |
| 3.4.2  | :SYSTem:COMMunication:LAN:IP:ADDRESS.....      | 80 |
| 3.4.3  | :SYSTem:COMMunication:LAN:IP:AUTO.....         | 81 |
| 3.4.4  | :SYSTem:COMMunication:LAN:IP:GATEway.....      | 82 |
| 3.4.5  | :SYSTem:COMMunication:LAN:IP:MANual.....       | 83 |
| 3.4.6  | :SYSTem:COMMunication:LAN:IP:SET.....          | 84 |
| 3.4.7  | :SYSTem:COMMunication:LAN:IP:SUBnet:MASK.....  | 84 |
| 3.4.8  | :SYSTem:COMMunication:LAN:RESet.....           | 85 |
| 3.4.9  | :SYSTem:COMMunication:LAN[:SELF]:PREFered..... | 85 |
| 3.4.10 | :SYSTem:COMMunication:LAN:MDNS.....            | 86 |
| 3.4.11 | :SYSTem:COMMunication:LAN:HOST:NAME.....       | 87 |
| 3.4.12 | :SYSTem:COMMunication:LAN:MAC.....             | 87 |
| 3.4.13 | :SYSTem:PRESet.....                            | 88 |
| 3.4.14 | :SYSTem:PRESet:TYPE.....                       | 88 |
| 3.4.15 | :SYSTem:PRESet:SAVE.....                       | 89 |
| 3.4.16 | :SYSTem:FSWitch:STATE.....                     | 89 |
| 3.4.17 | :SYSTem:LANGuage.....                          | 90 |
| 3.4.18 | :SYSTem:DATE.....                              | 91 |
| 3.4.19 | :SYSTem:TIME.....                              | 92 |
| 3.4.20 | :SYSTem:STIME.....                             | 92 |
| 3.4.21 | :SYSTem:CLEAr.....                             | 93 |
| 3.4.22 | :SYSTem:DISPlay:UPDate[:STATE].....            | 94 |
| 3.4.23 | :SYSTem:LKEY.....                              | 94 |
| 3.4.24 | :SYSTem:LKEY:DELeTe.....                       | 95 |
| 3.4.25 | :SYSTem:OPTion:LIST?.....                      | 96 |
| 3.4.26 | :SYSTem:CONFigure:INFormation?.....            | 96 |
| 3.5    | :MMEMory Commands.....                         | 97 |
| 3.5.1  | :MMEMory:CATalog.....                          | 97 |
| 3.5.2  | :MMEMory:CATalog:LENGth.....                   | 97 |
| 3.5.3  | :MMEMory:COpy.....                             | 98 |
| 3.5.4  | :MMEMory:DELeTe.....                           | 99 |

|        |                                     |     |
|--------|-------------------------------------|-----|
| 3.5.5  | :MMEMory:DISK:FORMat.....           | 100 |
| 3.5.6  | :MMEMory:DISK:INFormation.....      | 100 |
| 3.5.7  | :MMEMory:FILEtype.....              | 101 |
| 3.5.8  | :MMEMory:CHANnel:SElect.....        | 101 |
| 3.5.9  | :MMEMory:LDISK:SPACe.....           | 102 |
| 3.5.10 | :MMEMory:LOAD.....                  | 103 |
| 3.5.11 | :MMEMory:MDIRectory.....            | 103 |
| 3.5.12 | :MMEMory:MOVE.....                  | 104 |
| 3.5.13 | :MMEMory:PNAME:EDIT.....            | 105 |
| 3.5.14 | :MMEMory:PNAME:STATe.....           | 105 |
| 3.5.15 | :MMEMory:SAVe.....                  | 106 |
| 3.6    | IEEE488.2 Common Commands.....      | 107 |
| 3.6.1  | *CLS.....                           | 107 |
| 3.6.2  | *ESE.....                           | 107 |
| 3.6.3  | *IDN?.....                          | 108 |
| 3.6.4  | *PSC.....                           | 109 |
| 3.6.5  | *RST.....                           | 109 |
| 3.6.6  | *SRE.....                           | 110 |
| 3.6.7  | *STB?.....                          | 110 |
| 3.6.8  | *TRG.....                           | 111 |
| 3.6.9  | *WAI.....                           | 111 |
| 3.7    | :STATus Commands.....               | 112 |
| 3.7.1  | :STATus:OPERation:ENABLE.....       | 113 |
| 3.7.2  | :STATus:OPERation[:EVENT].....      | 113 |
| 4      | Application Examples.....           | 115 |
| 4.1    | To Output the RF Signal.....        | 115 |
| 4.2    | To Output the RF Sweep Signal.....  | 115 |
| 5      | Programming Examples.....           | 117 |
| 5.1    | Programming Preparations.....       | 117 |
| 5.2    | Excel Programming Examples.....     | 118 |
| 5.3    | Matlab Programming Example.....     | 122 |
| 5.4    | LabVIEW Programming Example.....    | 123 |
| 5.5    | Visual C++ Programming Example..... | 128 |
| 6      | Appendix.....                       | 133 |
| 6.1    | Appendix A: Factory Setting.....    | 133 |
| 6.2    | Appendix B: Warranty.....           | 135 |





# 1 Document Overview

This manual is your guide to programming the DSG5000 series microwave signal generator by using SCPI commands through remote interface.



## TIP

For the latest version of this manual, download it from the official website of **RIGOL** (<http://www.rigol.com>).

## Publication Number

PGG04100-1110

## Software Version

Software upgrade might change or add product features. Please acquire the latest version of the manual from **RIGOL** website or contact **RIGOL** to upgrade the software.

## Format Conventions in this Manual

### 1. Menu

The menu item is denoted by the format of "Menu Name (Bold) + Character Shading" in the manual. For example, **Freq** indicates clicking or tapping the "Freq" menu to enter the "Freq" setting configuration menu.

### 2. Operation Procedures

">" denotes the next step of operation. For example, **Interface** > **LAN** indicates that first clicking/tapping **Interface** and then clicking/tapping **LAN**.

### 3. Connector

The connectors on the front or rear panel are usually denoted by the format of "Connector Name (Bold) + Square Brackets (Bold)". For example, **[RF 50Ω]**.

## Content Conventions in this Manual

The DSG5000 series includes the following models. Unless otherwise specified, this manual takes DSG5208 as an example to illustrate how to use the DSG5000 series.

| Model   | Frequency Range | No. of Channels |
|---------|-----------------|-----------------|
| DSG5202 | 9 kHz to 20 GHz | 2               |
| DSG5204 | 9 kHz to 20 GHz | 4               |
| DSG5206 | 9 kHz to 20 GHz | 6               |
| DSG5208 | 9 kHz to 20 GHz | 8               |

| Model   | Frequency Range | No. of Channels |
|---------|-----------------|-----------------|
| DSG5122 | 9 kHz to 12 GHz | 2               |
| DSG5124 | 9 kHz to 12 GHz | 4               |
| DSG5126 | 9 kHz to 12 GHz | 6               |
| DSG5128 | 9 kHz to 12 GHz | 8               |

### Manuals of this Product

The manuals of this product mainly include the quick guide, user guide, programming guide, data sheet, and etc. For the latest version of this manual, download it from the official website of RIGOL (<http://www.rigol.com>).

## 2 SCPI Command Overview

SCPI (Standard Commands for Programmable Instruments) is a standardized instrument programming language that is built upon the existing standard IEEE 488.1 and IEEE 488.2 and conforms to various standards, such as the floating point operation rule in IEEE 754 standard, ISO 646 7-bit coded character set for information interchange (equivalent to ASCII programming). The SCPI commands provide a hierarchical tree structure, and consist of multiple subsystems. Each command subsystem consists of one root keyword and one or more sub-keywords.

### Syntax

The command line usually starts with a colon; the keywords are separated by colons, and following the keywords are the parameter settings available. The command ending with a quotation mark indicates querying a certain function and returns the query results. The keywords of the command and the first parameter are separated by a space. For example,

```
:SYSTem:COMMunication:LAN:IP:ADDDress <value>
```

**SYSTem** is the root keyword of the command, **COMMunication** is the second-level keyword, **LAN**, **IP**, **ADDDress** are the second-level, third-level, forth-level and fifth-level keywords respectively. The command line starts with a colon, and different levels of keywords are also separated by colons. *<value>* indicates a settable parameter. The command ending with a quotation mark "?" indicates querying. The command keywords **:SYSTem:COMMunication:LAN:IP:ADDDress** and the parameter *<value>* are separated by a space.

In some commands with parameters, ", " is often used to separate multiple parameters. For example,

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:ADDList <freq>,<level>,<time>
```

### Symbol Description

The following symbols are not sent with the commands.

#### 1. Braces { }

The parameters enclosed in the braces are optional. You do not have to set it, also you can set for one or more times.

#### 2. Vertical Bar |

The vertical bar is used to separate multiple parameters. When using the command, you must select one of the parameters. For example,

```
:SYSTem:LANGuage <type>
```

Wherein, *<type>* can be set to CHINese|ENGLISH.

#### 3. Square Brackets [ ]

The contents in the square brackets can be omitted. If the parameter is omitted, the instrument will set the parameter to its default. For example, for the command

```
[ :SOURce ] :RFALl :OUTPut [ :STATe ] ?
```

sending any of the four commands below can generate the same effect:

```
:SOURce:RFALl:OUTPut:STATe?
```

```
:SOURce:RFALl:OUTPut?
```

```
:RFALl:OUTPut:STATe?
```

```
:RFALl:OUTPut?
```

#### 4. Angle Brackets < >

The parameter enclosed in the angle brackets must be replaced by an effective value. For example,

```
:FREQuency 4 MHz is the correct command form for sending the command  
[ :SOURce ] [ :RF< channel/> ] :FREQuency < freq> [ < unit> ].
```

### Parameter Type

The command parameters introduced in this manual include 5 types: bool, discrete, integer, real number, and ASCII string.

#### 1. Bool

The parameter can be set to ON, OFF, 1, or 0. For example,

```
:SYSTem:COMMunication:LAN:MDNS < bool/>
```

```
:SYSTem:COMMunication:LAN:MDNS?
```

Wherein, < bool/> can be set to 1|ON|0|OFF. The query returns 1 or 0.

#### 2. Discrete

The parameter can be any of the values listed. For example,

```
:SYSTem:LANGuage < type>
```

```
:SYSTem:LANGuage?
```

Wherein, < type> can be set to CHINese|ENGLish. The query returns an abbreviated form: CHIN or ENGL.

#### 3. Integer

Unless otherwise specified, the parameter can be any integer (NR1 format) within the effective value range.

#### CAUTION

**Do not set the parameter to a decimal, otherwise, errors will occur.**

For example,



**:SYSTem:BRIGhtness <value>**

**:SYSTem:BRIGhtness?**

Wherein, <value> can be set to an integer ranging from 0 to 7. The query returns an integer ranging from 0 to 7.

#### 4. Real

The parameter can be any real number within the effective value range, and this command accepts parameter input in decimal (NR2 format) and scientific notation (NR3 format). For example,

**[ :SOURce ] [ :RF<channel> ] :FREQuency <freq> [<unit>]**

**[ :SOURce ] [ :RF<channel> ] :FREQuency?**

Wherein, <freq> can be set to a real number ranging from 9 kHz to 20 GHz.

#### 5. ASCII String

The parameter can be the combinations of ASCII characters. For example,

**:MMEMory:SAVe <File\_name>**

Wherein, <File\_name> is the filename of the file to be saved and can include Chinese characters (one Chinese character occupies two bytes), English characters and numbers. The filename cannot exceed 28 bytes.

#### Command Abbreviation

All the commands are case-insensitive. They can all be in upper case or in lower case. If abbreviation is used, you must input all the capital letters in the command. For example,

**:SYSTem:COMMunication:LAN:IP:SET**

can be abbreviated to

**:SYST:COMM:LAN:IP:SET.**

## 3 Command System

This chapter introduces the syntax, function, parameters, and usage of each command.

### 3.1 :SOURce Commands

The :SOURce commands are used to set the related parameters of the main functions of the microwave signal generator, including the frequency, level, phase, flatness correction, Pulse, SWEEP, and so on.

#### 3.1.1 [:SOURce][:RF]:FREQUENCY Commands

##### 3.1.1.1 [:SOURce][:RF]:FREQUENCY

###### Syntax

```
[[:SOURce]][:RF<channel>]:FREQUENCY <freq>[<unit>]
```

```
[[:SOURce]][:RF<channel>]:FREQUENCY?
```

###### Description

Sets the RF frequency of the specified channel.

Queries the RF frequency of the specified channel.

###### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <freq>    | Real     | 9 kHz to 20 GHz | 20 GHz  |
| <unit>    | Discrete | Hz kHz MHz GHz  | Hz      |

###### Remarks

- When <freq> is set in "Number" form, the default unit is Hz. For example, 4000000. Besides, <freq> can also be set in "Number + Unit" form, For example, 4MHz.
- The default unit of the return value is Hz.
- When the frequency offset is 0 Hz, the frequency displayed in the interface (namely the setting frequency) is equal to the actual output frequency.

- When the frequency offset is not 0 Hz, the output frequency is determined by the setting frequency and frequency offset. The three parameters satisfy the equation: setting frequency (display frequency) = output frequency + frequency offset. You can query or set the RF frequency offset by using the `[[:SOURce] [:RF]:FREQuency:OFFSet]` command.

### Return Format

The query returns the RF frequency of the specified channel.

### Example

```
:RF2:FREQ 4MHz /*Sets the RF frequency of CH2 to 4MHz.*/
:RF2:OFFS? /*The query returns 4000000Hz.*/
```

### 3.1.1.2 [[:SOURce]][:RF]:FREQuency:OFFSet

#### Syntax

`[[:SOURce] [:RF<channel>] :FREQuency:OFFSet <freq>[<unit>]]`

`[[:SOURce] [:RF<channel>] :FREQuency:OFFSet?]`

#### Description

Sets the RF frequency offset of the specified channel.

Queries the RF frequency offset of the specified channel.

#### Parameter

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <freq>    | Real     | -49 GHz to 49 GHz | 0 Hz    |
| <unit>    | Discrete | Hz kHz MHz GHz    | Hz      |

#### Remarks

- When <freq> is set in "Number" form, the default unit is Hz. For example, 20000. Besides, <freq> can also be set in "Number + Unit" form, For example, 20kHz.
- The default unit of the return value is Hz.

- When the frequency offset is 0 Hz, the frequency displayed in the interface (namely the setting frequency) is equal to the actual output frequency.
- When the frequency offset is not 0 Hz, the output frequency is determined by the setting frequency and frequency offset. The three parameters satisfy the equation: setting frequency (display frequency) = output frequency + frequency offset. You can query or set the RF frequency by using the `[[:SOURce]`  
`[[:RF]:FREQuency` command.

### Return Format

The query returns the RF frequency offset of the specified channel.

### Example

```
:RF2:FREQ:OFFS 20kHz /*Sets the RF frequency offset of CH2 to
20kHz*/
:RF2:FREQ:OFFS? /*The query returns 20000Hz.*/
```

## 3.1.2 [:SOURce][:RF]:LEVel Commands

### 3.1.2.1 [:SOURce][:RF]:LEVel

#### Syntax

```
[[:SOURce]][:RF]<channel>]:LEVel <level>[<unit>]
```

```
[[:SOURce]][:RF]<channel>]:LEVel?
```

#### Description

Sets the RF output amplitude of the specified channel.

Queries the RF output amplitude of the specified channel.

#### Parameter

| Name      | Type     | Range              | Default |
|-----------|----------|--------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8    | 1       |
| <level>   | Real     | -30 dBm to +25 dBm | -30 dBm |
| <unit>    | Discrete | dBm V W dBmV dBuV  | dBm     |



### Remarks

- When < value> is set in "Number" form (for example, 2), the default unit is dBm. When set in "Number + Unit" form (for example, 2dBm), the level displayed in the interface of the microwave signal generator is related to the setting of the level unit.
  - When the level unit is "dBm", 2.00dBm is displayed.
  - When the level unit is "dBmV", 48.99dBmV is displayed.
  - When the level unit is "dBuV", 108.99dBuV is displayed.
  - When the level unit is "Volts", 281.50mV is displayed.
  - When the level unit is "Watts", 1.58mW is displayed.
- The default unit of the return value is dBm.
- When the amplitude offset is 0 dB, the display amplitude (namely the setting amplitude) is equal to the actual output amplitude.
- When the amplitude offset is not 0 dB, the output amplitude is determined by the setting amplitude and amplitude offset. The three parameters satisfy the equation: setting amplitude (display amplitude) = output amplitude + amplitude offset. You can query or set the RF amplitude offset by using the `[[:SOURce][:RF]:LEVel:OFFSet]` command.

### Return Format

The query returns the RF output amplitude of the specified channel.

### Example

```
:LEV 2dBm /*Sets the RF output amplitude of CH1 to 2dBm (by
default, CH1 is selected).*/
:LEV? /*The query returns 2.00.*/
```

### 3.1.2.2 [:SOURce][:RF]:LEVel:LIMit

#### Syntax

```
[[:SOURce] [:RF<channel>] :LEVel:LIMit <value> [<unit>]
[:SOURce] [:RF<channel>] :LEVel:LIMit?
```

#### Description

Sets the limit of the RF output amplitude for the specified channel.

Queries the limit of the RF output amplitude for the specified channel.

#### Parameter

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <value>   | Real     | -30 dBm to 25 dBm | 25 dBm  |
| <unit>    | Discrete | dBm V W dBmV dBuV | dBm     |

#### Remarks

- When < value> is set in "Number" form (for example, 2), the default unit is dBm. When set in "Number + Unit" form (for example, 2dBm), the level displayed in the interface of the microwave signal generator is related to the setting of the level unit.
  - When the level unit is "dBm", 2.00dBm is displayed.
  - When the level unit is "dBmV", 48.99dBmV is displayed.
  - When the level unit is "dBuV", 108.99dBuV is displayed.
  - When the level unit is "Volts", 281.50mV is displayed.
  - When the level unit is "Watts", 1.58mW is displayed.
- The default unit of the return value is dBm.

#### Return Format

The query returns the limit of the RF output amplitude for the specified channel.

#### Example

```
:LEV:LIM 5.6 /*Sets the limit of the RF output amplitude of CH1 to
5.6dBm (by default, CH1 is selected).*/
:LEV:LIM? /*The query returns 5.60.*/
```

### 3.1.2.3 [:SOURce][:RF]:LEVel:OFFSet

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :LEVel:OFFSet <value>
```

```
[ :SOURce ] [ :RF<channel> ] :LEVel:OFFSet?
```

#### Description

Sets the amplitude offset of the RF output for the specified channel.

Queries the amplitude offset of the RF output for the specified channel.

#### Parameter

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <value>   | Real     | -200 dB to 200 dB | 0 dB    |

#### Remarks

- When <value> is set in "Number" form, the default unit is dB. Besides, <value> can also be set in "Number + Unit" form. For example, 10dB.
- The default unit of the return value is dB.
- When the amplitude offset is 0 dB, the display amplitude (namely the setting amplitude) is equal to the actual output amplitude.
- When the amplitude offset is not 0 dB, the output amplitude is determined by the setting amplitude and amplitude offset. The three parameters satisfy the equation: setting amplitude (display amplitude) = output amplitude + amplitude offset. You can query or set the RF amplitude by using the `[[:SOURce][[:RF]:LEV]` command.

#### Return Format

The query returns the amplitude offset of the RF output for the specified channel.

#### Example

```
:LEV:OFFS 10 /*Sets the amplitude offset of the RF output of CH1
to 10dB (by default, CH1 is selected).*/
:LEV:OFFS? /*The query returns 10.00.*/
```

### 3.1.3 [[:SOURce][[:RF]:PHASe Commands

#### 3.1.3.1 [[:SOURce][[:RF]:PHASe

##### Syntax

```
[[:SOURce][[:RF:<channel>]:PHASe <value>
```

```
[[:SOURce][[:RF:<channel>]:PHASe?
```

##### Description

Sets the phase deviation of the RF output for the specified channel.

Queries the phase deviation of the RF output for the specified channel.

#### Parameter

| Name      | Type     | Range               | Default |
|-----------|----------|---------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8     | 1       |
| <value>   | Real     | -720 deg to 720 deg | 0 deg   |

#### Remarks

- When <value> is set in "Number" form, the default unit is deg. Besides, <value> can also be set in "Number + Unit" form. For example, 35deg.
- The default unit of the return value is deg.

#### Return Format

The query returns the phase deviation of the RF output for the specified channel.

#### Example

```
:PHAS 35 /*Sets the phase deviation of the RF output of CH1 to
35deg (by default, CH1 is selected).*/
:PHAS? /*The query returns 35.00.*/
```

### 3.1.3.2 [:SOURce][:RF]:PHASe:RESet

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :PHASe:RESet
```

#### Description

Resets the current phase deviation to 0deg for the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

#### Remarks

N/A

#### Return Format

N/A

#### Example

N/A

### 3.1.4 [:SOURce][:RF]:CORRection Commands

#### 3.1.4.1 [:SOURce][:RF]:CORRection:FLATness[:STATe]

##### Syntax

```
[ :SOURce ] [ :RF<channel> ] :CORRection:FLATness [ :STATe ] <bool>
```

```
[ :SOURce ] [ :RF<channel> ] :CORRection:FLATness [ :STATe ] ?
```

##### Description

Enables or disables the flatness correction for the specified channel.

Queries the on/off status of the flatness correction for the specified channel.

##### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | 0 OFF   |

##### Remarks

N/A

##### Return Format

The query returns 1 or 0.

##### Example

```
:CORR:FLAT ON /*Enables the flatness correction for CH1 (by
default, CH1 is selected).*/
:CORR:FLAT? /*The query returns 1.*/
```

#### 3.1.4.2 [:SOURce][:RF]:CORRection:FLATness:COUNT

##### Syntax

```
[ :SOURce ] [ :RF<channel> ] :CORRection:FLATness:COUNT?
```

##### Description

Queries the number of points in the current flatness correction table for the specified channel.

##### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

**Remarks**

You can use the `[[:SOURce][:RF]:CORRection:FLATness:DELeTe` command to reduce the number of points in the table or use the `[[:SOURce][:RF]:CORRection:FLATness:PAIR` command to increase the number of points in the table.

**Return Format**

The query returns the number of points in the current flatness correction table for the specified channel in integer. For example, 5.

**Example**

N/A

**3.1.4.3 [:SOURce][:RF]:CORRection:FLATness:LIST****Syntax**

`[[:SOURce]][:RF]<channel>]:CORRection:FLATness:LIST? <Start>,<Count>`

**Description**

Queries the flatness correction list within the specified range for the specified channel.

**Parameter**

| Name      | Type     | Range  | Default |
|-----------|----------|--|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8                                | 1       |
| <Start>   | Integer  | 1 to total number of lines in the current list | -       |
| <Count>   | Integer  | 1 to total number of lines in the current list | -       |

**Remarks**

- **<Start>**: indicates the start line No. of the list to be acquired.
- **<Count>**: indicates the total number of lines of the list to be acquired.

**Return Format**

The query returns the newly acquired flatness correction list. For example,

NO.1:2000000.000000 , 9.000000

NO.2:2500000.000000 , 10.000000

**Example**

```
:CORR:FLAT:LIST? 2,3 /*Queries three lines of flatness correction
data starting from the 2nd line of the flatness correction list
for CH1 (by default, CH1 is selected).*/
```

**3.1.4.4****[[:SOURce]][:RF]:CORRection:FLATness:DELeTe****Syntax**

```
[[:SOURce]][:RF]<channel>]:CORRection:FLATness:DELeTe <Row>
```

**Description**

Deletes one correction point from the flatness correction list for the specified channel.

**Parameter**

| Name      | Type     | Range  | Default |
|-----------|----------|--|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8                                | 1       |
| <Row>     | Integer  | 1 to total number of lines in the current list | -       |

**Remarks**

- <Row> represents the line number in the flatness correction list. The total number of lines is the total number of correction points in the flatness correction list.
- Each correction point corresponds to one frequency value and amplitude value.
- You can use this command to reduce the number of lines from the list or use the `[[:SOURce]][:RF]:CORRection:FLATness:PAIR` command to increase the number points in the list.

**Return Format**

N/A

**Example**

```
:CORR:FLAT:DEL 2 /*Deletes the second line from the flatness
correction list for CH1 (by default, CH1 is selected).*/
```

### 3.1.4.5 [:SOURce][:RF]:CORRection:FLATness:PAIR

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :CORRection:FLATness:PAIR <Freq>,<Ampt>
```

#### Description

Sets the correction point pair in the flatness correction list for the specified channel.

#### Parameter

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <Freq>    | Real     | 9 kHz to 20 GHz   | 20 GHz  |
| <Ampt>    | Real     | -100 dB to 100 dB | -       |

#### Remarks

If the frequency point exists in the flatness correction list, modify the amplitude value; if not exist, add it.

You can use this command to edit the frequency and amplitude values of the currently selected line in the flatness correction list.

- **<Freq>**: sets the frequency of the correction point.
- **<Ampt>**: sets the amplitude value at the frequency point.

#### Return Format

N/A

#### Example

```
:CORR:FLAT:PAIR 2.5M,10 /*Sets the frequency value of the
currently selected line to 2.5MHz and the amplitude value to 10dB
for CH1 (by default, CH1 is selected).*/
```

### 3.1.4.6 [:SOURce][:RF]:CORRection:FLATness:LOAD

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :CORRection:FLATness:LOAD <filename>
```

#### Description

Loads the specified flatness calibration list file to the specified channel.



**Parameter**

| Name       | Type         | Range           | Default |
|------------|--------------|-----------------|---------|
| <channel>  | Discrete     | 1 2 3 4 5 6 7 8 | 1       |
| <filename> | ASCII String | -               | -       |

**Remarks**

N/A

**Return Format**

N/A

**Example**

```
:RF2:CORR:FLAT:LOAD flat.csv /*Loads the flatness calibration list
file named "flat.csv" to CH2.*/
```

**3.1.4.7 [:SOURce][:RF]:CORRection:FLATness:SAVe****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :CORRection:FLATness:SAVe <filename>
```

**Description**

Saves the flatness calibration list file of the specified channel.

**Parameter**

| Name       | Type         | Range           | Default |
|------------|--------------|-----------------|---------|
| <channel>  | Discrete     | 1 2 3 4 5 6 7 8 | 1       |
| <filename> | ASCII String | -               | -       |

**Remarks**

N/A

**Return Format**

N/A

**Example**

```
:RF2:CORR:FLAT:SAV flat.csv /*Saves the flatness calibration list
file of CH2, with the filename "flat.csv".*/
```

## 3.1.5 [:SOURce][:RF]:Sweep Commands

### 3.1.5.1 [:SOURce][:RF]:SWEep:DIRection

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:DIRection <dir>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:DIRection?
```

#### Description

Sets the sweep direction for the specified channel.

Queries the sweep direction for the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <dir>     | Discrete | FWD REV         | FWD     |

#### Remarks

- **FWD:** selects "Fwd" sweep direction. At this point, the microwave signal generator sweeps from the start frequency or start level to the stop frequency or stop level.
- **REV:** selects "Down" sweep direction. At this point, the microwave signal generator sweeps from the stop frequency or stop level to the start frequency or start level.

#### Return Format

The query returns FWD or REV.

#### Example

```
:SWE:DIR FWD /*Sets the sweep direction to FWD (by default, CH1 is
selected).*/
:SWE:DIR? /*The query returns FWD.*/
```

### 3.1.5.2 [:SOURce][:RF]:SWEep:EXECute

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:EXECute
```

**Description**

Executes a sweep on the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

**Remarks**

- If the current sweep mode is "Continuous", sending this command will change the sweep mode to "Single". The instrument starts a sweep if the trigger condition is currently met.
- If the current sweep mode is "Single", the instrument starts a sweep if the trigger condition is met after sending this command.

**Return Format**

N/A

**Example**

N/A

### 3.1.5.3 [:SOURce][:RF]:SWEep:LIST:ADDList

**Syntax**

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:ADDList <freq>,<level>,<time>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:ADDList?
```

**Description**

Adds one line of sweep values to the sweep list for the specified channel.

**Parameter**

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <freq>    | Real     | 9 kHz to 20 GHz   | 20 GHz  |
| <level>   | Real     | -30 dBm to 25 dBm | -30 dBm |
| <time>    | Real     | 5 ms to 100 s     | 500 ms  |

**Remarks**

When editing the sweep list, you can use this command to add one line of sweep values. At this point, one line is added to the sweep list.

- **<freq>**: sets the frequency of the sweep point.
- **<level>**: sets the amplitude at the frequency point.
- **<time>**: sets the duration of a sweep step.

**Return Format**

N/A

**Example**

```
:SWEep:RF2:LIST:ADDList 1M,10,1s /*Adds one line of sweep values
(frequency 1MHz, amplitude 10dBm, dwell time 1s) to the sweep list
for CH2.*/
```

**3.1.5.4 [:SOURce][:RF]:SWEep:LIST:CPOint****Syntax**

```
[[:SOURce]][:RF]<channel>]:SWEep:LIST:CPOint?
```

**Description**

Queries the number of points in the current sweep list for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

**Remarks**

You can use the `[:SOURce][:RF]:SWEep:LIST:DELe` command to delete one line of sweep values or use the `[:SOURce][:RF]:SWEep:LIST:ADDList` command to add one line of sweep values.

**Return Format**

The query returns the total number of sweep points in the sweep list. For example, 5.

**Example**

N/A

### 3.1.5.5 [:SOURce][:RF]:SWEep:LIST:DELeTe

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:DELeTe <ListNum>
```

#### Description

Deletes a line of sweep list values for the specified channel.

#### Parameter

| Name      | Type     | Range  | Default |
|-----------|----------|--|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8                                | 1       |
| <ListNum> | Integer  | 1 to total number of lines in the current list | -       |

#### Remarks

- <ListNum> denotes the line number (namely the number of sweep points) in the sweep list. The number of sweep points in the sweep list can not exceed 6001.
- You can use this command to reduce the number of lines in the list and use the `[:SOURce][:RF]:SWEep:LIST:ADDList` command to increase the number of lines in the list.

#### Return Format

N/A

#### Example

```
:SWE:LIST:DEL 2 /*Deletes the second line of sweep values from the sweep list (by default, CH1 is selected).*/
```

### 3.1.5.6 [:SOURce][:RF]:SWEep:LIST:LIST

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:LIST? <Start>,<Count>
```

#### Description

Acquires the sweep list within the specified range for the specified channel.

**Parameter**

| Name      | Type     | Range  | Default |
|-----------|----------|--|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8                                | 1       |
| <Start>   | Integer  | 1 to total number of lines in the current list | -       |
| <Count>   | Integer  | 1 to total number of lines in the current list | -       |

**Remarks**

- **<Start>**: indicates the start line No. of the sweep list to be acquired.
- **<Count>**: indicates the total number of lines of the sweep list to be acquired.

**Return Format**

The query returns the newly acquired sweep list. For example,

NO.1:2000000 , 11.000000, 1

NO.2:3000000 , 12.000000, 1

NO.3:4000000 , 13.000000, 1

**Example**

```
:SWEep:LIST:LIST? 2,3 /*Acquires 3 lines of sweep values starting
from the second line in the sweep list (by default, CH1 is
selected).*/
```

**3.1.5.7 [:SOURce][:RF]:SWEep:LIST:INITialize:FSTep****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:INITialize:FSTep
```

**Description**

Recalculates the data points set in the current step sweep to generate a new sweep list.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

**Remarks**

In the new sweep list,

- "ListNum" depends on the "Points" of the step sweep.
- "Frequency" depends on the "Start Frequency" and "Stop Frequency" of the step sweep.
- "Amplitude" depends on the "Start Level" and "Stop Level" of the step sweep.
- "Time" depends on the "Dwell Time" of the step sweep.

#### Return Format

N/A

#### Example

N/A

### 3.1.5.8 [:SOURce][:RF]:SWEep:LIST:INITialize:PRESet

#### Syntax

```
[[:SOURce]][:RF]<channel>]:SWEep:LIST:INITialize:PRESet
```

#### Description

Resets the sweep list for the specified channel to factory setting.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

#### Remarks

- After resetting the sweep list using this command, the sweep list only contains one frequency point (20GHz) and one level point (-30dBm).
- You can use the [\[:SOURce\]\[:RF\]:SWEep:LIST:ADDList](#) and [\[:SOURce\]\[:RF\]:SWEep:LIST:DELeTe](#) commands to re-edit the current sweep list.

#### Return Format

N/A

#### Example

N/A

### 3.1.5.9 [:SOURce][:RF]:SWEep:LIST:LOAD

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:LOAD <filename>
```

#### Description

Loads the specified sweep list file to the specified channel.

#### Parameter

| Name       | Type         | Range           | Default |
|------------|--------------|-----------------|---------|
| <channel>  | Discrete     | 1 2 3 4 5 6 7 8 | 1       |
| <filename> | ASCII String | -               | -       |

#### Remarks

N/A

#### Return Format

N/A

#### Example

```
:RF2:SWE:LIST:LOAD sweep.csv /*Loads the specified sweep list file  
named "sweep.csv" to CH2.*/
```

### 3.1.5.10 [:SOURce][:RF]:SWEep:LIST:SAVe

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:LIST:SAVe <filename>
```

#### Description

Saves the sweep list file of the specified channel.

#### Parameter

| Name       | Type         | Range           | Default |
|------------|--------------|-----------------|---------|
| <channel>  | Discrete     | 1 2 3 4 5 6 7 8 | 1       |
| <filename> | ASCII String | -               | -       |

#### Remarks

N/A



**Return Format**

N/A

**Example**

```
:RF2:SWE:LIST:SAV sweep.csv /*Saves the sweep list file of CH2,
with the filename "sweep.csv".*/
```

**3.1.5.11 [:SOURce][:RF]:SWEep:STATe****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STATe <state>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STATe?
```

**Description**

Sets the sweep state for the specified channel.

Queries the sweep state for the specified channel.

**Parameter**

| Name      | Type     | Range                                   | Default |
|-----------|----------|---|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8                         | 1       |
| <state>   | Discrete | OFF FREQuency LEVel <br>LEVel,FREQuency | OFF     |

**Remarks**

- **OFF:** disables the sweep function.
- **FREQuency:** enables the frequency sweep function.
- **LEVel:** enables the level sweep function.
- **LEVel,FREQuency:** enables both the frequency and the level sweep functions.

**Return Format**

The query returns the sweep state.

**Example**

```
:SWE:STAT FREQ /*Sets the sweep state to FREQ (by default, CH1 is
selected).*/
:SWE:STAT? /*The query returns FREQ.*/
```

### 3.1.5.12 [:SOURce][:RF]:SWEep:TYPE

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:TYPE <type>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:TYPE?
```

#### Description

Sets the sweep type of the specified channel.

Queries the sweep type of the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <type>    | Discrete | LIST STEP       | STEP    |

#### Remarks

- **LIST:** selects "List" sweep type. At this point, the microwave signal generator sweeps according to the current sweep list.
- **STEP:** selects "step" sweep type. At this point, the microwave signal generator performs step sweep at the current setting.

#### Return Format

The query returns LIST or STEP.

#### Example

```
:SWE:TYPE STEP /*Sets the sweep type to STEP (by default, CH1 is
selected).*/
:SWE:TYPE? /*The query returns STEP.*/
```

### 3.1.5.13 [:SOURce][:RF]:SWEep:MODE

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:MODE <mode>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:MODE?
```

#### Description

Sets the sweep mode for the specified channel.

Queries the sweep mode for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default  |
|-----------|----------|-----------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1        |
| <mode>    | Discrete | CONTinue SINGle | CONTinue |

**Remarks**

- **CONTinue:** selects "Cont" mode. The instrument sweeps continuously according to the current setting when the trigger condition is met.
- **SINGle:** selects "Single" mode. The instrument performs a sweep according to the current setting and then stops when the trigger condition is met.

**Return Format**

The query returns CONT or SING.

**Example**

```
:SWE:MODE CONT /*Sets the sweep mode to CONT (by default, CH1 is
selected).*/
:SWE:MODE? /*The query returns CONT.*/
```

**3.1.5.14 [:SOURce][:RF]:SWEep:RESet[:ALL]****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :SWEep:RESet [ :ALL ]
```

**Description**

Resets all the sweeps of the specified channel to the start point.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

**Remarks**

- If the current sweep direction is "Fwd", the instrument will stop the current sweep and sweep from the start frequency or start level after sending this command.

- If the current sweep direction is "Down", the instrument will stop the current sweep and sweep from the stop frequency or stop level after sending this command.

#### Return Format

N/A

#### Example

N/A

### 3.1.5.15 [:SOURce][:RF]:SWEep:POINt:TRIGger:TYPE

#### Syntax

[[:SOURce]][:RF]<channel>]:SWEep:POINt:TRIGger:TYPE <type>

[[:SOURce]][:RF]<channel>]:SWEep:POINt:TRIGger:TYPE?

#### Description

Sets the point trigger type of the sweep for the specified channel.

Queries the point trigger type of the sweep for the specified channel.

#### Parameter

| Name      | Type     | Range            | Default |
|-----------|----------|------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8  | 1       |
| <type>    | Discrete | AUTO KEY BUS EXT | AUTO    |

#### Remarks

- **AUTO:** selects Auto trigger.

If the sweep mode is "Continuous", select one sweep type and the instrument sweeps each sweep point continuously within a sweep period. If the sweep mode is "Single", you need to send the *[:SOURce][:RF]:SWEep:EXECute* command first to meet the single sweep condition, and then it will launch one sweep period and then stops.

- **KEY:** selects Key trigger.

If the sweep mode is "Continuous", every time you click or tap **Key**, the instrument starts to sweep one point. If the sweep mode is "Single", you need to send the *[:SOURce][:RF]:SWEep:EXECute* command first to meet the single sweep condition. At this point, every time you click or tap **Key**, the instrument sweeps one point and then stops after completing one sweep period.

- **BUS:** selects Bus trigger.

If the sweep mode is "Continuous", every time you send the *\*TRG* command, the instrument starts to sweep one point. If the sweep mode is "Single", you need to send the *[:SOURce][:RF]:SWEep:EXECute* command first to meet the single sweep condition. At this point, each time you send the *\*TRG* command, the instrument sweeps one point and then stops after completing one sweep period.

- **EXT:** selects External trigger.

The instrument receives the trigger signal input from the front-panel **[TRIGGER]** connector. If the sweep mode is "Continuous", every time the instrument receives a TTL pulse signal with the specified polarity, the instrument starts to sweep one point. If the sweep mode is "Single", you need to send the *[:SOURce][:RF]:SWEep:EXECute* command first to meet the single sweep condition. At this point, each time the instrument receives a TTL pulse signal with the specified polarity, it sweeps one point and then stops after completing one sweep period.



#### TIP

The above remarks are only valid when the trigger mode in the sweep period is met.

When executing the sweep operation, the priority of the required conditions is: single sweep → trigger mode → point trigger mode.

#### Return Format

The query returns the point trigger type.

#### Example

```
:SWE:POIN:TRIG:TYPE AUTO /*Sets the point trigger type to AUTO (by
default, CH1 is selected).*/
:SWE:POIN:TRIG:TYPE? /*The query returns AUTO.*/
```

### 3.1.5.16 **[:SOURce][:RF]:SWEep:SWEep:TRIGger:TYPE**

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:SWEep:TRIGger:TYPE <type>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:SWEep:TRIGger:TYPE?
```

#### Description

Sets the trigger type of the sweep period for the specified channel.

Queries the trigger type of the sweep period for the specified channel.

**Parameter**

| Name      | Type     | Range            | Default |
|-----------|----------|------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8  | 1       |
| <type>    | Discrete | AUTO KEY BUS EXT | AUTO    |

**Remarks**

- **AUTO:** selects Auto trigger.

If the sweep mode is "Continuous", you only need to select one sweep type and start to sweep; if the sweep mode is "Single", you need to send the *[[:SOURce]][:RF]:SWEep:EXECute* command first to meet the single sweep condition, and then it will launch one sweep and then stops.

- **KEY:** selects Key trigger.

If the sweep mode is "Continuous", every time you click or tap **Key**, the instruments start to sweep one point. If the sweep mode is "Single", you need to send the *[[:SOURce]][:RF]:SWEep:EXECute* command first to meet the single sweep condition. At this point, every time you click or tap **Key**, the instrument launches one sweep and then stops.

- **BUS:** selects Bus trigger.

If the sweep mode is "Continuous", every time you send the *\*TRG* command, the instrument starts one sweep. If the sweep mode is "Single", you need to send the *[[:SOURce]][:RF]:SWEep:EXECute* command first to meet the single sweep condition. At this point, every time you send the *\*TRG* command, the instrument launches one sweep and then stops.

- **EXT:** selects External trigger.

The instrument receives the trigger signal input from the front-panel **[TRIGGER]** connector. If the sweep mode is "Continuous", every time the instrument receives a TTL pulse signal with the specified polarity, the instrument starts one sweep. If the sweep mode is "Single", you need to send the *[[:SOURce]][:RF]:SWEep:EXECute* command first to meet the single sweep condition. At this point, every time the instrument receives a TTL pulse signal with the specified polarity, it launches one sweep and then stops.

**TIP**

The above remarks are only valid when the trigger mode of each sweep point in the sweep period is met.

When executing the sweep operation, the priority of the required conditions is: single sweep → trigger mode → point trigger mode.

### Return Format

The query returns the sweep trigger type.

### Example

```
:SWE:SWE:TRIG:TYPE AUTO /*Sets the sweep trigger type to AUTO (by
default, CH1 is selected).*/
:SWE:SWE:TRIG:TYPE? /*The query returns AUTO.*/
```

## 3.1.5.17 [:SOURce][:RF]:SWEep:STEP:DWELL

### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STEP:DWELL <value>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STEP:DWELL?
```

### Description

Sets the dwell time of the step sweep for the specified channel.

Queries the dwell time of the step sweep for the specified channel.

### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Real     | 5 ms to 100 s   | 500 ms  |

### Remarks

- When <value> is set in "Number" form, the default unit is s. Besides, <value> can also be set in "Number + Unit" form. For example, 3000ms.
- The default unit of the return value is s.

### Return Format

The query returns the dwell time of the step sweep.

### Example

```
:SWE:STEP:DWELL 3 /*Sets the dwell time of the step sweep to 3s (by
default, CH1 is selected).*/
:SWE:STEP:DWELL? /*The query returns 3.*/
```

## 3.1.5.18 [:SOURce][:RF]:SWEep:STEP:POINTs

### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STEP:POINTs <value>
```

[ :SOURce] [ :RF<channel> ] :SWEep:STEP:POINTs?

### Description

Sets the number of points of step sweep.

Queries the number of points of step sweep.

### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Integer  | 2 to 1,001      | 11      |

### Remarks

The number of sweep points determines the time interval between two neighboring sweep points.

### Return Format

The query returns the number of sweep points.

### Example

```
:SWEep:STEP:POINTs 5 /*Sets the number of points of step sweep to
5 (by default, CH1 is selected).*/
:SWEep:STEP:POINTs? /*The query returns 5.*/
```

## 3.1.5.19 [:SOURce][:RF]:SWEep:STEP:START:FREQuency

### Syntax

[ :SOURce] [ :RF<channel> ] :SWEep:STEP:START:FREQuency <freq>[<unit>]

[ :SOURce] [ :RF<channel> ] :SWEep:STEP:START:FREQuency?

### Description

Sets the start frequency of the step sweep for the specified channel.

Queries the start frequency of the step sweep for the specified channel.

### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <freq>    | Real     | 9 kHz to 20 GHz | 20 GHz  |
| <unit>    | Discrete | Hz kHz MHz GHz  | Hz      |



**Remarks**

- When <freq> is set in "Number" form, the default unit is Hz. For example, 4000000. Besides, <freq> can also be set in "Number + Unit" form, For example, 4MHz.
- The default unit of the return value is Hz.

**Return Format**

The query returns the start frequency of the step sweep.

**Example**

```
:SWE:STEP:STAR:FREQ 4MHz /*Sets the start frequency of the step
sweep to 4MHz (by default, CH1 is selected).*/
:SWE:STEP:STAR:FREQ? /*The query returns 4000000.*/
```

**3.1.5.20 [:SOURce][:RF]:SWEep:STEP:STARt:LEVel****Syntax**

```
[[:SOURce]][:RF]<channel>]:SWEep:STEP:STARt:LEVel <level>[<unit>]
```

```
[[:SOURce]][:RF]<channel>]:SWEep:STEP:STARt:LEVel?
```

**Description**

Sets the start level of the step sweep for the specified channel.

Queries the start level of the step sweep for the specified channel.

**Parameter**

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <level>   | Real     | -30 dBm to 25 dBm | -10 dBm |
| <unit>    | Discrete | dBm V W dBmV dBuV | dBm     |

**Remarks**

- When <level> is set in "Number" form (for example, 2), the default unit is dBm. When set in "Number + Unit" form (for example, 2dBm), the start level displayed in the interface of the microwave signal generator is related to the setting of the level unit.
  - When the level unit is "dBm", 2.00dBm is displayed.

- When the level unit is "dBmV", 48.99dBmV is displayed.
- When the level unit is "dBuV", 108.99dBuV is displayed.
- When the level unit is "Volts", 281.50mV is displayed.
- When the level unit is "Watts", 1.58mW is displayed.
- The default unit of the return value is dBm.

### Return Format

The query returns the start level of the step sweep.

### Example

```
:SWE:STEP:STAR:LEV 2dBm /*Sets the start level of the step sweep
to 2dBm (by default, CH1 is selected).*/
:SWE:STEP:STAR:LEV? /*The query returns 2.00.*/
```

### 3.1.5.21 [:SOURce][:RF]:SWEep:STEP:STOP:FREQuency

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STEP:STOP:FREQuency <freq>[<unit>]
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STEP:STOP:FREQuency?
```

#### Description

Sets the stop frequency of the step sweep for the specified channel.

Queries the stop frequency of the step sweep for the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <freq>    | Real     | 9 kHz to 20 GHz | 2 GHz   |
| <unit>    | Discrete | Hz kHz MHz GHz  | Hz      |

#### Remarks

- When <value> is set in "Number" form, the default unit is Hz. Besides, <value> can also be set in "Number + Unit" form. For example, 4MHz.
- The default unit of the return value is Hz.

### Return Format

The query returns the stop frequency of the step sweep.

### Example

```
:SWE:STEP:STOP:FREQ 4MHz /*Sets the stop frequency of the step
sweep to 4MHz (by default, CH1 is selected).*/
:SWE:STEP:STOP:FREQ? /*The query returns 4000000.*/
```

### 3.1.5.22 [:SOURce][:RF]:SWEep:STEP:STOP:LEVel

#### Syntax

```
[[:SOURce]][:RF]<channel>]:SWEep:STEP:STOP:LEVel <level>[<unit>]
```

```
[[:SOURce]][:RF]<channel>]:SWEep:STEP:STOP:LEVel?
```

#### Description

Sets the stop level of the step sweep for the specified channel.

Queries the stop level of the step sweep for the specified channel.

#### Parameter

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <level>   | Real     | -30 dBm to 25 dBm | -20 dBm |
| <unit>    | Discrete | dBm V W dBmV dBuV | dBm     |

#### Remarks

- When <level> is set in "Number" form (for example, 2), the default unit is dBm. When set in "Number + Unit" form (for example, 2dBm), the start level displayed in the interface of the microwave signal generator is related to the setting of the level unit.
  - When the level unit is "dBm", 2.00dBm is displayed.
  - When the level unit is "dBmV", 48.99dBmV is displayed.
  - When the level unit is "dBuV", 108.99dBuV is displayed.
  - When the level unit is "Volts", 281.50mV is displayed.
  - When the level unit is "Watts", 1.58mW is displayed.
- The default unit of the return value is dBm.

### Return Format

The query returns the stop level of the step sweep.

### Example

```
:SWE:STEP:STOP:LEV 2dBm /*Sets the stop level of the step sweep to
2dBm (by default, CH1 is selected).*/
:SWE:STEP:STOP:LEV? /*The query returns 2.00.*/
```

### 3.1.5.23 [:SOURce][:RF]:SWEep:STEP:SHAPE

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STEP:SHAPE <shape>
```

```
[ :SOURce ] [ :RF<channel> ] :SWEep:STEP:SHAPE?
```

#### Description

Sets the step sweep shape for the specified channel.

Queries the step sweep shape for the specified channel.

#### Parameter

| Name      | Type     | Range           | Default  |
|-----------|----------|-----------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1        |
| <shape>   | Discrete | TRlangle RAMP   | TRlangle |

#### Remarks

The sweep shape determines the cycle mode of multiple sweeps.

- **TRlangle:** selects the "Triangle" waveform. The sweep period always starts from the start frequency or start level to the stop frequency or stop level and then returns back to the start frequency or start level (when the sweep direction is "Fwd").
- **RAMP:** selects the "Ramp" waveform. The sweep period always starts from the start frequency or start level to the stop frequency or stop level (when the sweep direction is "Fwd").

#### Return Format

The query returns TRI or RAMP.

**Example**

```
:SWE:STEP:SHAP TRI /*Sets the step sweep shape to TRI (by default,
CH1 is selected).*/
:SWE:STEP:SHAP? /*The query returns TRI.*/
```

**3.1.5.24 [:SOURce][:RF]:SWEep:STEP:SPACing****Syntax**

```
[[:SOURce]][:RF]<channel>]:SWEep:STEP:SPACing <type>
```

```
[[:SOURce]][:RF]<channel>]:SWEep:STEP:SPACing?
```

**Description**

Sets the step sweep spacing for the specified channel.

Queries the step sweep spacing for the specified channel.

**Parameter**

| Name      | Type     | Range              | Default |
|-----------|----------|--------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8    | 1       |
| <type>    | Discrete | LINear LOGarithmic | LINear  |

**Remarks**

The sweep spacing refers to the variation mode from one frequency or amplitude to another frequency or amplitude within one step.

- **LINear:** sets the sweep spacing to "Lin". The level sweep only supports Lin sweep spacing.
- **LOGarithmic:** sets the sweep spacing to "LOG".

**Return Format**

The query returns LIN or LOG.

**Example**

```
:SWE:STEP:SPAC LIN /*Sets the step sweep spacing to LIN (by
default, CH1 is selected).*/
:SWE:STEP:SPAC? /*The query returns LIN.*/
```

## 3.1.6 [:SOURce][:RF]:AM Commands

### 3.1.6.1 [:SOURce][:RF]:AM:DEPTH

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :AM:DEPTH <value>
```

```
[ :SOURce ] [ :RF<channel> ] :AM:DEPTH?
```

#### Description

Sets the AM modulation depth of the specified channel.

Queries the AM modulation depth of the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Real     | 0 to 100        | 50      |

#### Remarks

- When "internal" modulation source is selected, the AM modulation depth ( $m_a$ ) and amplitude difference ( $\Delta P_{sb}$ ) between the carrier and sidebands satisfy the following relations:  $\Delta P_{sb} = 6 - 20 \lg m_a$ .
- Besides, <value> can also be expressed in percentage. For example, 80%.

#### Return Format

The query returns the modulation depth. For example, 80.00

#### Example

```
:RF2:AM:DEPT 80 /*Sets the AM modulation depth of CH2 to 80%.*/
:RF2:AM:DEPT? /*The query returns 80.00.*/
```

### 3.1.6.2 [:SOURce][:RF]:AM:FREQUENCY

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :AM:FREQUENCY <freq> [<unit>]
```

```
[ :SOURce ] [ :RF<channel> ] :AM:FREQUENCY?
```

#### Description

Sets the AM modulation frequency of the specified channel.

Queries the AM modulation frequency of the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <freq>    | Real     | 0.1 Hz to 1 MHz | 10 kHz  |
| <unit>    | Discrete | Hz kHz MHz GHz  | Hz      |

#### Remarks

- When <freq> is set in "Number" form, the default unit is Hz. For example, 20000. Besides, <freq> can also be set in "Number + Unit" form, For example, 20kHz.
- The default unit of the return value is Hz.
- When the AM modulation source is set to "Ext", this command is invalid. You can run the `[[:SOURce]][:RF]:AM:SOURce` command to set or query the type of the AM modulation source.

#### Return Format

The query returns the AM modulation frequency.

#### Example

```
:RF2:AM:FREQ 20kHz /*Sets the AM modulation frequency of CH2 to
20kHz.*/*
:RF2:AM:FREQ? /*The query returns 20000.*/*
```

### 3.1.6.3 [[:SOURce]][:RF]:AM:WAVEform

#### Syntax

```
[[:SOURce]][:RF]<channel>]:AM:WAVEform <type>
```

```
[[:SOURce]][:RF]<channel>]:AM:WAVEform?
```

#### Description

Sets the AM modulating waveform of the specified channel.

Queries the AM modulating waveform of the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <type>    | Discrete | SINE SQUA       | SINE    |

**Remarks**

- **SINE:** sets the AM modulating waveform to Sine.
- **SQUA:** sets the AM modulating waveform to Square.

When the AM modulation source is set to "Ext", this command is invalid. You can run the `[[:SOURce][:RF]:AM:SOURce]` command to set or query the type of the AM modulation source.

**Return Format**

The query returns SINE or SQUA.

**Example**

```
:RF2:AM:WAVE SQUA /*Sets the AM modulating waveform of CH2 to SQUA.*/
:RF2:AM:WAVE? /*The query returns SQUA.*/
```

**3.1.6.4 [:SOURce][:RF]:AM:STATe****Syntax**

`[[:SOURce]][:RF]<channel>]:AM:STATe <bool>`

`[[:SOURce]][:RF]<channel>]:AM:STATe?`

**Description**

Turns on or off the AM switch to enable the AM function for the specified channel.

Queries the on/off status of the AM switch for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | OFF 0   |

**Remarks**

- **ON:** turns on the AM switch to enable the AM function.



- **OFF:** turns off the AM switch to disable the AM function.

#### Return Format

The query returns 1 or 0.

#### Example

```
:RF2:AM:STAT ON /*Turns on the AM switch to enable the AM
function for CH2.*/
:RF2:AM:STAT? /*The query returns 1.*/
```

### 3.1.6.5

#### **[[:SOURce]][:RF]:AM:SOURce**

##### Syntax

[[:SOURce]][:RF]<channel>]:AM:SOURce <type>

[[:SOURce]][:RF]<channel>]:AM:SOURce?

##### Description

Sets the AM modulation source type of the specified channel.

Queries the AM the modulation source type of the specified channel.

##### Parameter

| Name      | Type     | Range             | Default  |
|-----------|----------|-------------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1        |
| <type>    | Discrete | EXternal Internal | Internal |

##### Remarks

- **EXternal:** selects the external modulation source. At this point, the external modulating signal is input from the **[EXT MOD IN]** connector.
- **Internal:** selects the internal modulation source. At this point, the instrument provides the modulating signal. You can run the **[[:SOURce]][:RF]:AM:FREQuency** command and the **[[:SOURce]][:RF]:AM:WAVEform** command to set the modulation frequency and modulation waveform of the modulating signal.

#### Return Format

The query returns the AM modulation source type of the specified channel.

**Example**

```
:RF2:AM:SOUR EXT /*Sets the AM modulation source type of CH2 to
EXT./
:RF2:AM:SOUR? /*The query returns EXT.*/
```

**3.1.6.6 [:SOURce][:RF]:AM:EQual:LEVel****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :LEVel:AM:EQual:LEVel <value>
```

```
[ :SOURce ] [ :RF<channel> ] :LEVel:AM:EQual:LEVel?
```

**Description**

Sets the on/off status of the RF equal level output of the specified channel.

Queries the on/off status of the RF equal level output of the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Bool     | 1 ON 0 OFF      | 0 OFF   |

**Remarks**

N/A

**Return Format**

The query returns 1 or 0.

**Example**

```
:AM:EQual:LEVel ON /*Enables the RF equal level output of CH1 (by
default, CH1 is selected).*/
:AM:EQual:LEVel? /*The query returns 1.*/
```

**3.1.6.7 [:SOURce]:AM:EXT:COUP****Syntax**

```
[ :SOURce ] :AM:EXT:COUP <type>
```

```
[ :SOURce ] :AM:EXT:COUP?
```

**Description**

Sets the AM external coupling mode.

Queries the AM external coupling mode.

**Parameter**

| Name   | Type     | Range | Default |
|--------|----------|-------|---------|
| <type> | Discrete | AC DC | AC      |

**Remarks**

- AC: sets the AM external coupling mode to "AC".
- DC: sets the AM external coupling mode to "DC".
- When the modulation source of AM is set to "Int", this command is invalid. You can send the `[:SOURce][:RF]:AM:SOURce` command to set or query the AM modulation source.

**Return Format**

The query returns AC or DC.

**Example**

```
:AM:EXT:COUP AC      /*Sets the AM external coupling mode to AC.*/
:AM:EXT:COUP?        /*The query returns AC.*/
```

**3.1.6.8****[:SOURce]:AM:EXT:IMP****Syntax**

```
[ :SOURce ] :AM:EXT:IMP <value>
```

```
[ :SOURce ] :AM:EXT:IMP?
```

**Description**

Sets the impedance of AM external modulation.

Queries the impedance of AM external modulation.

**Parameter**

| Name    | Type     | Range       | Default |
|---------|----------|-------------|---------|
| <value> | Discrete | 50 600 100k | 100k    |

**Remarks**

- 50: sets the impedance of AM external modulation to "50Ω".
- 100k: sets the impedance of AM external modulation to "100kΩ".
- 600: sets the impedance of AM external modulation to "600Ω".

- When the modulation source of AM is set to "Int", this command is invalid. You can send the `[:SOURce][:RF]:AM:SOURce` command to set or query the AM modulation source.

#### Return Format

The query returns 50, 600, or 100k.

#### Example

```
:AM:EXT:IMP 600 /*Sets the impedance of the AM external
modulation to 600Ω.*/
:AM:EXT:IMP? /*The query returns 600.*/
```

### 3.1.7 [:SOURce][:RF]:FM Commands

#### 3.1.7.1 [:SOURce][:RF]:FM:DEVIation

##### Syntax

```
[[:SOURce]][:RF]<channel>]:FM:DEVIation <value>[<unit>]
```

```
[[:SOURce]][:RF]<channel>]:FM:DEVIation?
```

##### Description

Sets the FM frequency offset of the specified channel.

Queries the FM frequency offset of the specified channel.

##### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Real     | 0.1 Hz to 4 MHz | 10 kHz  |
| <unit>    | Discrete | Hz kHz MHz GHz  | Hz      |

##### Remarks

- When <value> is set in "Number" form, the default unit is Hz. For example, 20000. Besides, <value> can also be set in "Number + Unit" form, For example, 20kHz.
- The default unit of the return value is Hz.

### Return Format

The query returns the FM frequency offset.

### Example

```
:RF2:FM:DEV 20kHz /*Sets the FM frequency offset of CH2 to 20kHz*/
:RF2:FM:DEV? /*The query returns 20000.*/
```

## 3.1.7.2

### [[:SOURce]][:RF]:FM:FREQuency

#### Syntax

```
[[:SOURce]][:RF<channel>]:FM:FREQuency <freq>[<unit>]
```

```
[[:SOURce]][:RF<channel>]:FM:FREQuency?
```

#### Description

Sets the FM modulation frequency of the specified channel.

Queries the FM modulation frequency of the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <freq>    | Real     | 0.1 Hz to 1 MHz | 10 kHz  |
| <unit>    | Discrete | Hz kHz MHz GHz  | Hz      |

#### Remarks

- When <freq> is set in "Number" form, the default unit is Hz. For example, 20000. Besides, <freq> can also be set in "Number + Unit" form, For example, 20kHz.
- The default unit of the return value is Hz.
- When the FM modulation source is set to "Ext", this command is invalid.

### Return Format

The query returns the FM modulation frequency.

### Example

```
:RF2:FM:FREQ 20kHz /*Sets the FM modulation frequency of CH2 to
20kHz.*/
:RF2:FM:FREQ? /*The query returns 20000.*/
```

### 3.1.7.3 [:SOURce][:RF]:FM:WAVEform

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :FM:WAVEform <type>
```

```
[ :SOURce ] [ :RF<channel> ] :FM:WAVEform?
```

#### Description

Sets the FM modulating waveform of the specified channel.

Queries the FM modulating waveform of the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <type>    | Discrete | SINE SQUA       | SINE    |

#### Remarks

- **SINE:** sets the FM modulating waveform to Sine.
- **SQUA:** sets the FM modulating waveform to Square.

When the PM modulation source is set to "Ext", this command is invalid. You can run the `[:SOURce][:RF]:FM:SOURce` command to set or query the type of the FM modulation source.

#### Return Format

The query returns SINE or SQUA.

#### Example

```
:RF2:FM:WAVE SQUA /*Sets the FM modulating waveform of CH2 to SQUA./
:RF2:AM:WAVE? /*The query returns SQUA.*/
```

### 3.1.7.4 [:SOURce][:RF]:FM:STATe

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :FM:STATe <bool>
```

```
[ :SOURce ] [ :RF<channel> ] :FM:STATe?
```

#### Description

Turns on or off the FM switch to enable the AM function for the specified channel.

Queries the on/off status of the FM switch for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | 0 OFF   |

**Remarks**

- **ON:** turns on the FM switch to enable the FM function.
- **OFF:** turns off the FM switch to disable the FM function.

**Return Format**

The query returns 1 or 0.

**Example**

```
:RF2:FM:STAT ON /*Turns on the FM switch to enable the FM
function for CH2.*/
:RF2:FM:STAT? /*The query returns 1.*/
```

**3.1.7.5****[[:SOURce]][:RF]:FM:SOURce****Syntax**

```
[[:SOURce]][:RF:<channel>]:FM:SOURce <type>
```

```
[[:SOURce]][:RF:<channel>]:FM:SOURce?
```

**Description**

Sets the FM modulation source type of the specified channel.

Queries the FM the modulation source type of the specified channel.

**Parameter**

| Name      | Type     | Range             | Default  |
|-----------|----------|-------------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1        |
| <type>    | Discrete | EXternal INternal | INternal |

**Remarks**

- **EXternal:** selects the external modulation source. At this point, the external modulating signal is input from the **[EXT MOD IN]** connector.

- **INTernal:** selects the internal modulation source. At this point, the instrument provides the modulating signal. You can run the `[:SOURce][:RF]:FM:FREQuency` command and the `[:SOURce][:RF]:FM:WAVEform` command to set the modulation frequency and modulation waveform of the modulating signal.

#### Return Format

The query returns the FM modulation source type of the specified channel.

#### Example

```
:RF2:FM:SOUR INT /*Sets the FM modulation source type of CH2 to
INT./
:RF2:FM:SOUR? /*The query returns INT.*/
```

### 3.1.7.6 [:SOURce]:FM:EXT:COUP

#### Syntax

`[ :SOURce ] :FM:EXT:COUP <type>`

`[ :SOURce ] :FM:EXT:COUP?`

#### Description

Sets the FM external coupling mode.

Queries the FM external coupling mode.

#### Parameter

| Name   | Type     | Range | Default |
|--------|----------|-------|---------|
| <type> | Discrete | AC DC | AC      |

#### Remarks

- AC: sets the FM external coupling mode to "AC".
- DC: sets the FM external coupling mode to "DC".
- When the modulation source of AM is set to "Int", this command is invalid. You can send the `[:SOURce][:RF]:FM:SOURce` command to set or query the FM modulation source.

#### Return Format

The query returns AC or DC.



**Example**

```
:FM:EXT:COUP AC      /*Sets the FM external coupling mode to AC.*/
:FM:EXT:COUP?        /*The query returns AC.*/
```

**3.1.7.7****[[:SOURce]:FM:EXT:IMP****Syntax**

```
[[:SOURce]:FM:EXT:IMP <value>
```

```
[[:SOURce]:FM:EXT:IMP?
```

**Description**

Sets the impedance of FM external modulation.

Queries the impedance of FM external modulation.

**Parameter**

| Name    | Type     | Range       | Default |
|---------|----------|-------------|---------|
| <value> | Discrete | 50 600 100k | 100k    |

**Remarks**

- 50: sets the impedance of FM external modulation to "50Ω".
- 100k: sets the impedance of FM external modulation to "100kΩ".
- 600: sets the impedance of FM external modulation to "600Ω".
- When the modulation source of FM is set to "Int", this command is invalid. You can send the `[[:SOURce]:[:RF]:FM:SOURce` command to set or query the FM modulation source.

**Return Format**

The query returns 50, 600, or 100k.

**Example**

```
:FM:EXT:IMP 600      /*Sets the impedance of the FM external
modulation to 600Ω.*/
:FM:EXT:IMP?         /*The query returns 600.*/
```

### 3.1.8 [:SOURce][:RF]:PM Commands

#### 3.1.8.1 [:SOURce][RF]:PM:DEVIation

##### Syntax

```
[ :SOURce ] [ :RF<channel> ] :PM:DEVIation <value> [<unit>]
```

```
[ :SOURce ] [ :RF<channel> ] :PM:DEVIation?
```

##### Description

Sets the PM phase deviation for the specified channel.

Queries the PM phase deviation for the specified channel.

##### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Real     | 0 rad to 10 rad | 5 rad   |
| <unit>    | Discrete | rad             | rad     |

##### Remarks

- When <value> is set in "Number" form, the default unit is rad.
- The default unit of the return value is rad.

##### Return Format

The query returns the PM phase deviation for the specified channel.

##### Example

```
:RF2:PM:DEV 2 /*Sets the PM phase deviation of CH2 to 2 rad.*/
:RF2:PM:DEV? /*The query returns 2.00.*/
```

#### 3.1.8.2 [:SOURce][:RF]:PM:FREQuency

##### Syntax

```
[ :SOURce ] [ :RF<channel> ] :PM:FREQuency <freq> [<unit>]
```

```
[ :SOURce ] [ :RF<channel> ] :PM:FREQuency?
```

##### Description

Sets the PM modulation frequency of the specified channel.

Queries the PM modulation frequency of the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <freq>    | Real     | 0.1 Hz to 1 MHz | 10 kHz  |
| <unit>    | Discrete | Hz kHz MHz GHz  | Hz      |

**Remarks**

- When <freq> is set in "Number" form, the default unit is Hz. For example, 20000. Besides, <freq> can also be set in "Number + Unit" form, For example, 20kHz.
- The default unit of the return value is Hz.

**Return Format**

The query returns the PM modulation frequency.

**Example**

```
:RF2:PM:FREQ 20kHz /*Sets the PM modulation frequency of CH2 to 20 kHz.*/
:RF2:PM:FREQ? /*The query returns 20000.*/
```

**3.1.8.3****[ :SOURce ] [ :RF ] :PM:WAVEform****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :PM:WAVEform <type>
```

```
[ :SOURce ] [ :RF<channel> ] :PM:WAVEform?
```

**Description**

Sets the PM modulating waveform of the specified channel.

Queries the PM modulating waveform of the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <type>    | Discrete | SINE SQUA       | SINE    |

**Remarks**

- **SINE:** sets the PM modulating waveform to Sine.
- **SQUA:** sets the PM modulating waveform to Square.

When the PM modulation source is set to "Ext", this command is invalid. You can run the `[[:SOURce]][:RF]:PM:SOURce` command to set or query the type of the PM modulation source.

**Return Format**

The query returns SINE or SQUA.

**Example**

```
:RF2:PM:WAVE SQUA /*Sets the PM modulating waveform of CH2 to SQUA.*/
:RF2:AM:WAVE? /*The query returns SQUA.*/
```

**3.1.8.4 [:SOURce]][:RF]:PM:STATe****Syntax**

`[[:SOURce]][:RF]<channel>]:PM:STATe <bool>`

`[[:SOURce]][:RF]<channel>]:PM:STATe?`

**Description**

Turns on or off the PM switch to enable the AM function for the specified channel.

Queries the on/off status of the PM switch for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | 0 OFF   |

**Remarks**

- **ON:** turns on the PM switch to enable the PM function.
- **OFF:** turns off the PM switch to disable the PM function.

**Return Format**

The query returns 1 or 0.

**Example**

```
:RF2:PM:STAT ON /*Turns on the PM switch to enable the PM
function for CH2.*/
:RF2:PM:STAT? /*The query returns 1.*/
```

**3.1.8.5****[[:SOURce]][:RF]:PM:SOURce****Syntax**

```
[[:SOURce]][:RF]<channel>]:PM:SOURce <type>
```

```
[[:SOURce]][:RF]<channel>]:PM:SOURce?
```

**Description**

Sets the PM modulation source type of the specified channel.

Queries the PM the modulation source type of the specified channel.

**Parameter**

| Name      | Type     | Range             | Default  |
|-----------|----------|-------------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1        |
| <type>    | Discrete | EXternal INTernal | INTernal |

**Remarks**

- **EXternal:** selects the external modulation source. At this point, the external modulating signal is input from the **[EXT MOD IN]** connector.
- **INTernal:** selects the internal modulation source. At this point, the instrument provides the modulating signal. You can run the *[[:SOURce]][:RF]:PM:FREQuency* command and the *[[:SOURce]][:RF]:PM:WAVEform* command to set the modulation frequency and modulation waveform of the modulating signal.

**Return Format**

The query returns the PM modulation source type of the specified channel.

**Example**

```
:RF2:PM:SOUR INT /*Sets the PM modulation source type of CH2 to
INT.*/
:RF2:PM:SOUR? /*The query returns INT.*/
```

### 3.1.8.6 [:SOURce]:PM:EXT:COUP

#### Syntax

[ :SOURce ] : PM : EXT : COUP < type >

[ :SOURce ] : PM : EXT : COUP ?

#### Description

Sets the PM external coupling mode.

Queries the PM external coupling mode.

#### Parameter

| Name   | Type     | Range | Default |
|--------|----------|-------|---------|
| <type> | Discrete | AC DC | AC      |

#### Remarks

- AC: sets the PM external coupling mode to "AC".
- DC: sets the PM external coupling mode to "DC".
- When the modulation source of PM is set to "Int", this command is invalid. You can send the [:SOURce][:RF]:PM:SOURce command to set or query the PM modulation source.

#### Return Format

The query returns AC or DC.

#### Example

```
:PM:EXT:COUP AC      /*Sets the PM external coupling mode to AC.*/  
:PM:EXT:COUP?        /*The query returns AC.*/
```

### 3.1.8.7 [:SOURce]:PM:EXT:IMP

#### Syntax

[ :SOURce ] : PM : EXT : IMP < value >

[ :SOURce ] : PM : EXT : IMP ?

#### Description

Sets the impedance of the PM external modulation.

Queries the impedance of the PM external modulation.

**Parameter**

| Name    | Type     | Range       | Default |
|---------|----------|-------------|---------|
| <value> | Discrete | 50 600 100k | 100k    |

**Remarks**

- 50: sets the impedance of the PM external modulation to "50Ω".
- 100k: sets the impedance of the PM external modulation to "100kΩ".
- 600: sets the impedance of the PM external modulation to "600Ω".
- When the modulation source of PM is set to "Int", this command is invalid. You can send the `[[:SOURce]][:RF]:PM:SOURce` command to set or query the PM modulation source.

**Return Format**

The query returns 50, 600, or 100k.

**Example**

```
:PM:EXT:IMP 600 /*Sets the impedance of the PM external
modulation to 600Ω.*/
:PM:EXT:IMP? /*The query returns 600.*/
```

## 3.1.9 [:SOURce]][:RF]:PULM Commands

### 3.1.9.1 [:SOURce]][:RF]:PULM:MODE

**Syntax**

`[[:SOURce]][:RF]<channel>]:PULM:MODE <mode>`

`[[:SOURce]][:RF]<channel>]:PULM:MODE?`

**Description**

Sets the pulse type of the specified channel.

Queries the pulse type of the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <mode>    | Discrete | SINGLE TRAIn    | SINGLE  |

**Remarks**

- **SINGLE:** selects the "Single" pulse type.
- **TRAI**n: selects the "Train" pulse type.

This command is invalid when "Ext" modulation source is selected. You can run the `[[:SOURce]][:RF]:PULM:SOURce` command to set or query the pulse modulation source.

**Return Format**

The query returns SING or TRA.

**Example**

```
:PULM:MODE SING /*Sets the pulse type of the pulse modulation to
SING (by default, CH1 is selected).*/
:PULM:MODE? /*The query returns SING.*/
```

**3.1.9.2 [:SOURce]][:RF]:PULM:OUT:STATe****Syntax**

`[[:SOURce]][:RF]<channel>]:PULM:OUT:STATe <bool>`

`[[:SOURce]][:RF]<channel>]:PULM:OUT:STATe?`

**Description**

Enables or disables the pulse output for the specified channel.

Queries the on/off status of the pulse output for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | 0 OFF   |

**Remarks**

- **ON|1:** enables the pulse output. At this point, the microwave signal generator can output the pulse signal generated by the internal pulse generator via the front-panel **[PULSE]** connector. The output signal is related to the "Mode" setting.
- **OFF|0:** disables the pulse output.



This command is invalid when "Ext" modulation source is selected. You can run the `[[:SOURce][:RF]:PULM:SOURce]` command to set or query the pulse modulation source.

### Return Format

The query returns 1 or 0.

### Example

```
:PULM:OUT:STAT ON /*Enables the pulse output (by default, CH1 is
selected).*/
:PULM:OUT:STAT? /*The query returns 1.*/
```

## 3.1.9.3 [:SOURce][:RF]:PULM:PERiod

### Syntax

`[[:SOURce]][:RF:<channel>]:PULM:PERiod <value>`

`[[:SOURce]][:RF:<channel>]:PULM:PERiod?`

### Description

Sets the single pulse period for the specified channel.

Queries the single pulse period for the specified channel.

### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Real     | 40 ns to 170 s  | 500 us  |

### Remarks

- When <value> is set in "Number" form, the default unit is s. Besides, <value> can also be set in "Number + Unit" form. For example, 1000ms.
- The default unit of the return value is s.
- The period of the single pulse is related to the current "Mode" setting.
- This command is invalid when "Ext" modulation source is selected or "Train" pulse is selected. You can run the `[[:SOURce][:RF]:PULM:SOURce]` command to set or query the pulse modulation source; run the `[[:SOURce][:RF]:PULM:MODE]` command to set or query the pulse type of the pulse modulation.

**Return Format**

The query returns the period of the single pulse.

**Example**

```
:PULM:PER 1000ms /*Sets the period of the single pulse to 1s (by
default, CH1 is selected).*/
:PULM:PER? /*The query returns 1.*/
```

**3.1.9.4 [:SOURce][:RF]:PULM:POLarity****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :PULM:POLarity <type>
```

```
[ :SOURce ] [ :RF<channel> ] :PULM:POLarity?
```

**Description**

Sets the polarity of the pulse modulation for the specified channel.

Queries the polarity of the pulse modulation for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <type>    | Discrete | NORMAL INVerse  | NORMAL  |

**Remarks**

- **NORMAL:** sets the polarity of the current pulse modulation signal to "Normal".
- **INVerse:** sets the polarity of the current pulse modulation signal to "Inverse".

**Return Format**

The query returns NORM or INV.

**Example**

```
:PULM:POL INV /*Sets the polarity of the pulse modulation to INV
(by default, CH1 is selected).*/
:PULM:POL? /*The query returns INV.*/
```

**3.1.9.5 [:SOURce][:RF]:PULM:SOURce****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :PULM:SOURce <type>
```

```
[ :SOURce ] [ :RF<channel> ] :PULM:SOURce?
```

### Description

Sets the pulse modulation source type for the specified channel.

Queries the pulse modulation source type for the specified channel.

### Parameter

| Name      | Type     | Range             | Default  |
|-----------|----------|-------------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1        |
| <type>    | Discrete | INTernal EXTernal | INTernal |

### Remarks

- **INTernal:** selects the internal modulation source. At this point, the modulation signal is generated by the internal pulse generator. When "Pulse Out" is enabled, the pulse signal generated by the internal pulse generator is output via the front-panel **[PULSE]** connector.
- **EXTernal:** selects the external modulation source. At this point, the microwave signal generator receives the external pulse modulating signal input from the front-panel **[PULSE]** connector.

### Return Format

The query returns INT or EXT.

### Example

```
:PULM:SOUR INT /*Sets the pulse modulation source to INT (by
default, CH1 is selected).*/
:PULM:SOUR? /*The query returns INT.*/
```

## 3.1.9.6

### [[:SOURce]][:RF]:PULM:STATe

#### Syntax

```
[[:SOURce]][:RF]<channel>]:PULM:STATe <bool>
```

```
[[:SOURce]][:RF]<channel>]:PULM:STATe?
```

#### Description

Enables or disables the pulse modulation for the specified channel.

Queries the on/off status of the pulse modulation for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | 0 OFF   |

**Remarks**

- **ON:** enables the pulse modulation for the specified channel.
- **OFF:** disables the pulse modulation for the specified channel.

**Return Format**

The query returns 1 or 0.

**Example**

```
:PULM:STAT ON /*Enables the pulse modulation (by default, CH1 is
selected).*/
:PULM:STAT? /*The query returns 1.*/
```

**3.1.9.7 [:SOURce][:RF]:PULM:TRAI:n:LIST:COUNT**

[ :SOURce ] [ :RF<channel> ] :PULM:TRAI:n:LIST:COUNT?

**Description**

Acquires the total number of lines of the current pulse list for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

**Remarks**

You can use the `[:SOURce][:RF]:PULM:TRAI:n:LIST:DELe` command to reduce the number of lines of the list or use the `[:SOURce][:RF]:PULM:TRAI:n:LIST:INSERt` command to increase the number of lines of the list.

**Return Format**

The query returns an integer. For example, 2.

**Example**

N/A

### 3.1.9.8 [:SOURce][:RF]:PULM:TRAI:n:LIST:INSErt

#### Syntax

```
[ :SOURce] [ :RF<channel> ] :PULM:TRAI:n:LIST:INSErt
<OnTime>,<OffTime>,<Repeat>
```

#### Description

Inserts one line of pulse values into the pulse list for the specified channel.

#### Parameter

| Name      | Type     | Range                    | Default |
|-----------|----------|--------------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8          | 1       |
| <OnTime>  | Real     | 20 ns to (170 s - 20 ns) | 500 us  |
| <OffTime> | Real     | 20 ns to (170 s -20 ns)  | 500 us  |
| <Repeat>  | Integer  | 1 to 256                 | 1       |

#### Remarks

- **<OnTime>**: sets the duration of the positive pulse.
- **<OffTime>**: sets the duration of the negative pulse.
- **<Repeat>**: sets the repetition times of this pulse.

#### Return Format

N/A

#### Example

```
:PULM:TRA:LIST:INSE 5.5s,2.5s,2 /*Inserts one line of pulse values
before the currently selected line: OnTime 5.5s, OffTime 2.5s,
Repeat 2 (by default, CH1 is selected).*/
```

### 3.1.9.9 [:SOURce][:RF]:PULM:TRAI:n:LIST:DELEte

#### Syntax

```
[ :SOURce] [ :RF<channel> ] :PULM:TRAI:n:LIST:DELEte <Index>
```

#### Description

Deletes the pulse value of the specified line from the current pulse list for the specified channel.

**Parameter**

| Name      | Type     | Range  | Default |
|-----------|----------|--|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8                                      | 1       |
| <Index>   | Integer  | 1 to total number of lines in the current pulse list | -       |

**Remarks**

- <Index> denotes the line number of the pulse list.
- This command allows you to delete one line of pulse values. The `[[:SOURce] [:RF]:PULM:TRAI:n:LIST:INSErt` command allows you to add one line of pulse values.

**Return Format**

N/A

**Example**

```
:PULM:TRA:LIST:DEL 2 /*Deletes the second line of pulse values
from the current pulse list (by default, CH1 is selected).*/
```

**3.1.9.10 [[:SOURce] [:RF]:PULM:TRAI:n:LIST:GET****Syntax**

```
[[:SOURce] [:RF:<channel>] :PULM:TRAI:n:LIST:GET? <Start>,<Count>
```

**Description**

Acquires the pulse list within the specified range for the specified channel.

**Parameter**

| Name      | Type     | Range  | Default |
|-----------|----------|--|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8                                      | 1       |
| <Start>   | Integer  | 1 to total number of lines in the current pulse list | -       |
| <Count>   | Integer  | 1 to total number of lines in the current pulse list | -       |

**Remarks**

- **<Start>**: indicates the start line No. of the pulse list to be acquired.
- **<Count>**: indicates the total number of lines of the pulse list to be acquired.

**Return Format**

The query returns the newly acquired pulse list data. For example,

NO.2:200.00 us , 30.00 ms, 1, 30.20 ms

NO.3:300.00 us , 40.00 ms, 1, 40.30 ms

**Example**

```
:PULM:TRA:LIST:GET 2,3 /*Acquires 3 lines of pulse data from the
pulse list starting from the second line (by default, CH1 is
selected).*/
```

**3.1.9.11 [:SOURce][:RF]:PULM:TRAI:n:LIST:RUN****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :PULM:TRAI:n:LIST:RUN
```

**Description**

Executes the currently edited pulse list for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

**Remarks**

You can use this command to update the internal modulating signal for the current pulse list data.

**Return Format**

N/A

**Example**

N/A

**3.1.9.12 [:SOURce][:RF]:PULM:TRAI:n:LIST:LOAD**

```
[ :SOURce ] [ :RF<channel> ] :PULM:TRAI:n:LIST:LOAD<filename>
```

**Description**

Loads the specified pulse list file to the specified channel.

**Parameter**

| Name       | Type         | Range           | Default |
|------------|--------------|-----------------|---------|
| <channel>  | Discrete     | 1 2 3 4 5 6 7 8 | 1       |
| <filename> | ASCII String | -               | -       |

**Remarks**

N/A

**Return Format**

N/A

**Example**

```
:RF2:PULM:TRA:LIST:LOAD tra.csv /*Loads the specified pulse list
file named "tra.csv" to CH2.*/
```

**3.1.9.13 [:SOURce][:RF]:PULM:TRAI:n:LIST:SAVe**

```
[ :SOURce ] [ :RF<channel> ] :PULM:TRAI:n:LIST:SAVe<filename>
```

**Description**

Saves the pulse list file of the specified channel.

**Parameter**

| Name       | Type         | Range           | Default |
|------------|--------------|-----------------|---------|
| <channel>  | Discrete     | 1 2 3 4 5 6 7 8 | 1       |
| <filename> | ASCII String | -               | -       |

**Remarks**

N/A

**Return Format**

N/A

**Example**

```
:RF2:PULM:TRA:LIST:SAV tra.csv /*Saves the pulse list file of CH2
named "tra.csv".*/
```



### 3.1.9.14 [:SOURce][:RF]:PULM:TRIGger:DElay

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :PULM:TRIGger:DElay <value>[<unit>]
```

```
[ :SOURce ] [ :RF<channel> ] :PULM:TRIGger:DElay?
```

#### Description

Sets the pulse trigger delay for the specified channel.

Queries the pulse trigger delay for the specified channel..

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <value>   | Real     | 10 ns to 170 s  | 100 us  |
| <unit>    | Discrete | ns us ms s      | s       |

#### Remarks

- You can use this command to set the delay from the start of receiving the external trigger signal to the start of the pulse modulating signal when "Int" modulation source is selected and "Ext" trigger mode is selected.
- When <value> is set in "Number" form, the default unit is s. Besides, <value> can also be set in "Number + Unit" form. For example, 30ns.
- The default unit of the return value is s.

#### Return Format

The query returns the trigger delay.

#### Example

```
:PULM:TRIG:DEL 3 /*Sets the trigger delay to 3s (by default, CH1
is selected).*/
:PULM:TRIG:DEL? /*The query returns 3.*/
```

### 3.1.9.15 [:SOURce][:RF]:PULM:TRIGger:EXTErnal:GATE:POLarity

#### Syntax

```
[ :SOURce ] [ :RF<channel> ] :PULM:TRIGger:EXTErnal:GATE:POLarity <type>
```

```
[ :SOURce ] [ :RF<channel> ] :PULM:TRIGger:EXTErnal:GATE:POLarity?
```

### Description

Sets the polarity of the external gated signal for the specified channel.

Queries the polarity of the external gated signal for the specified channel.

### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <type>    | Discrete | NORMAL INVerse  | NORMAL  |

### Remarks

- **NORMAL:** sets the polarity of the current pulse modulation signal to "Normal".
- **INVerse:** sets the polarity of the current pulse modulation signal to "Inverse".

When the trigger mode of the pulse modulation is set to "Gated", the microwave signal generator receives the external gated signal input from the front-panel **[TRIGGER]** connector. At this point, you can use this command to set the polarity of the external gated signal. You can run the `[[:SOURce]:[RF]:PULM:TRIGger:MODE` command to set or query the trigger mode of the pulse modulation.

This command is invalid when "Ext" modulation source is selected. You can run the `[[:SOURce]:[RF]:PULM:SOURce` command to set or query the pulse modulation source.

### Return Format

The query returns NORM or INV.

### Example

```
:PULM:TRIG:EXT:GATE:POL INV /*Sets the polarity of the pulse
modulation to INV (by default, CH1 is selected).*/
:PULM:TRIG:EXT:GATE:POL? /*The query returns INV.*/
```

## 3.1.9.16 [:SOURce]:[RF]:PULM:TRIGger:EXternal:SLOPe

### Syntax

```
[[:SOURce]][:RF]<channel>]:PULM:TRIGger:EXternal:SLOPe <slope>
```

```
[[:SOURce]][:RF]<channel>]:PULM:TRIGger:EXternal:SLOPe?
```

### Description

Sets the slope of the valid edge of the external trigger pulse for the specified channel.

Queries the slope of the valid edge of the external trigger pulse for the specified channel.

#### Parameter

| Name      | Type     | Range             | Default  |
|-----------|----------|-------------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1        |
| <slope>   | Discrete | POSitive NEGative | POSitive |

#### Remarks

- **POSitive:** sets the slope of the valid edge of the external trigger pulse to "Pos".
- **NEGative:** sets the slope of the valid edge of the external trigger pulse to "Neg".

When the trigger mode of the pulse modulation is set to "Ext", the microwave signal generator receives the external trigger signal input from the front-panel **[TRIGGER]** connector. At this point, you can use this command to set the trigger edge of the external trigger signal.

This command is invalid when "Ext" modulation source is selected. You can run the `[[:SOURce][:RF]:PULM:SOURce]` command to set or query the pulse modulation source.

#### Return Format

The query returns POS or NEG.

#### Example

```
:PULM:RF2:TRIG:EXT:SLOP NEG /*Sets the slope of the valid edge of
the external trigger pulse for CH2 to NEG.*/
:PULM:RF2:TRIG:EXT:SLOP? /*The query returns NEG.*/
```

### 3.1.9.17 [:SOURce][:RF]:PULM:TRIGger:MODE

#### Syntax

```
[[:SOURce]][:RF:<channel>]:PULM:TRIGger:MODE <mode>
```

```
[[:SOURce]][:RF:<channel>]:PULM:TRIGger:MODE?
```

#### Description

Sets the trigger mode of the pulse modulation for the specified channel.

Queries the trigger mode of the pulse modulation for the specified channel.

**Parameter**

| Name      | Type     | Range                       | Default |
|-----------|----------|-----------------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8             | 1       |
| <mode>    | Discrete | AUTO EXTeRnal EGATe KEY BUS | AUTO    |

**Remarks**

- **AUTO:** selects Auto trigger. At this point, the microwave signal generator meets the trigger condition at any time and can perform pulse modulation continuously.
- **EXTeRnal:** selects Ext trigger. At this point, the microwave signal generator receives the external trigger signal input from the front-panel **[TRIGGER]** connector. The instrument starts a pulse modulation each time a TTL pulse with the specified polarity is received. To specify the polarity of the TTL pulse, use the *[[:SOURce]:RF]:PULM:TRIGger:EXTeRnal:SLOPe* command to select "Pos" or "Neg".
- **EGATe:** selects Gate trigger. At this point, the microwave signal generator receives the external gated signal input from the front-panel **[TRIGGER]** connector. The instrument starts a pulse modulation within its valid level each time a gated signal with the specified polarity is received. To specify the polarity of the external gated signal, use the *[[:SOURce]:RF]:PULM:TRIGger:EXTeRnal:GATE:POLarity* command to select "Normal" or "Inverse".
- **KEY:** selects Key trigger. At this point, the instrument starts a pulse modulation each time you click or tap **Key**.
- **BUS:** selects Bus trigger. At this point, the instrument starts a pulse modulation each time you send the *\*TRG* command.

This command is invalid when "Ext" modulation source is selected.

### Return Format

The query returns the trigger mode of the pulse modulation for the specified channel.

### Example

```
:PULM:TRIG:MODE EGAT /*Sets the trigger mode of the pulse
modulation to EGAT (by default, CH1 is selected).*/
:PULM:TRIG:MODE? /*The query returns EGAT.*/
```

### 3.1.9.18 [:SOURce][:RF]:PULM:WIDTH

#### Syntax

`[[:SOURce]][:RF]<channel>]:PULM:WIDTH <value>`

`[[:SOURce]][:RF]<channel>]:PULM:WIDTH?`

#### Description

Sets the pulse width of the single pulse for the specified channel.

Queries the pulse width of the single pulse for the specified channel.

#### Parameter

| Name      | Type     | Range                    | Default |
|-----------|----------|--------------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8          | 1       |
| <value>   | Real     | 10 ns to (170 s - 10 ns) | 500 us  |

#### Remarks

- When <value> is set in "Number" form, the default unit is s. Besides, <value> can also be set in "Number + Unit" form. For example, 2000ms.
- When the modulation source is "Internal" and the pulse mode is "Single", you can use this command to set the pulse width of the single pulse. Otherwise, this command is invalid for setting the pulse width.
- The width of the single pulse is limited by the minimum pulse width and pulse period, and the pulse width should meet the following conditions:  

$$\text{minimum width} \leq \text{pulse width} \leq (\text{pulse period}) - 10 \text{ ns}$$

You can run the `[:SOURce][:RF]:PULM:PERiod` command to set or query the pulse width.

**Return Format**

The query returns the pulse width of the single pulse.

**Example**

```
:PULM:WIDT 2 /*Sets the pulse width of the single pulse for CH1 to
2s (by default, CH1 is selected).*/
:PULM:WIDT? /*The query returns 2.*/
```

**3.1.10 [:SOURce][:RF]:OUTPut Commands****3.1.10.1 [:SOURce][:RF]:OUTPut[:STATe]****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :OUTPut [ :STATe ] <bool>
```

```
[ :SOURce ] [ :RF<channel> ] :OUTPut [ :STATe ] ?
```

**Description**

Enables or disables the RF output for the specified channel.

Queries the RF output status for the specified channel.

**Parameter**

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | 0 OFF   |

**Remarks**

- **ON|1**: enables the RF output of the channel.
- **OFF|0**: disables the RF output of the channel.

**Return Format**

The query returns 1 or 0.

**Example**

```
:RF2:OUTPut ON /*Enables the RF output of CH2.*/
:RF2:OUTPut? /*The query returns 1.*/
```

**3.1.10.2 [:SOURce]:RFALL:OUTPut[:STATe]****Syntax**

```
[ :SOURce ] :RFALL:OUTPut [ :STATe ] <bool>
```

`[ :SOURce ] :RFALl :OUTPut [ :STATe ] ?`

### Description

Enables or disables the RF output of all the channels.

Queries the RF output status of all the channels.

### Parameter

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 0 OFF   |

### Remarks

- **ON|1**: enables the RF output of all the channels.
- **OFF|0**: disables the RF output of all the channels.

### Return Format

The query returns 1 or 0.

### Example

```
:RFAL:OUTPut ON /*Enables the RF output of all the channels.*/  
:RFAL:OUTPut? /*The query returns 1.*/*
```

## 3.1.11 [:SOURce][:RF]:MODulation Commands

### 3.1.11.1 [:SOURce][:RF]:MODulation:STATe

#### Syntax

`[ :SOURce ] [ :RF<channel> ] :MODulation:STATe <bool>`

`[ :SOURce ] [ :RF<channel> ] :MODulation:STATe?`

#### Description

Enables or disables the modulation output for the specified channel.

Queries the on/off status of the modulation output for the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |
| <bool>    | Bool     | 1 ON 0 OFF      | 0 OFF   |

**Remarks**

N/A

**Return Format**

The query returns 1 or 0.

**Example**

```
:MOD:STATe ON /*Enables the modulation output for CH1 (by default,
CH1 is selected).*/
:MOD:STATe? /*The query returns 1.*/
```

**3.1.11.2 [:SOURce]:RFALl:MODulation:STATe****Syntax**

```
[[:SOURce]:RFALl:MODulation:STATe <bool>
```

```
[[:SOURce]:RFALl:MODulation:STATe?
```

**Description**

Enables or disables the modulation output for all the channels.

Queries the on/off status of modulation output for all the channels.

**Parameter**

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 0 OFF   |

**Remarks**

N/A

**Return Format**

The query returns 1 or 0.

**Example**

```
:RFAL:MOD:STATe ON /*Enables the modulation output for all the
channels.*/
:RFAL:MOD:STATe? /*The query returns 1.*/
```

**3.1.12 :SOURce:ROSCillator:HCLock Commands****3.1.12.1 :SOURce:ROSCillator:HCLock:OUTPut:STATe****Syntax**

```
:SOURce:ROSCillator:HCLock:OUTPut:STATe <bool>
```

```
:SOURce:ROSCillator:HCLock:OUTPut:STATe?
```



**Description**

Enables or disables the external 4.8 GHz output.

Queries the on/off status of external 4.8 GHz output.

**Parameter**

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 0 OFF   |

**Remarks**

N/A

**Return Format**

The query returns 1 or 0.

**Example**

```
:SOUR:ROSC:HCL:OUTP:STAT ON /*Enables the external 4.8 GHz  
output.*/  
:SOUR:ROSC:HCL:OUTP:STAT? /*The query returns 1.*//
```

**3.1.12.2 :SOURce:ROSCillator:HCLock:TYPE****Syntax**

```
:SOURce:ROSCillator:HCLock:TYPE <type>
```

```
:SOURce:ROSCillator:HCLock:TYPE?
```

**Description**

Sets the type of the 4.8 GHz reference clock.

Queries the the type of the 4.8 GHz reference clock.

**Parameter**

| Name   | Type     | Range             | Default |
|--------|----------|-------------------|---------|
| <type> | Discrete | INTernal EXTernal | -       |

**Remarks**

- **INTernal:** sets the type of the 4.8 GHz reference clock to "Internal".
- **EXTernal:** sets the type of the 4.8 GHz reference clock to "External".

**Return Format**

The query returns INT or EXT.

**Example**

```
:SOUR:ROSC:HCL:TYPE INT /*Sets the type of the 4.8 GHz reference
clock to "Internal".*/
:SOUR:ROSC:HCL:TYPE? /*The query returns INT.*/
```

**3.1.13 [:SOURce][:RF]:INPut:TRIGger:SLOPe****Syntax**

```
[ :SOURce ] [ :RF<channel> ] :INPut:TRIGger:SLOPe <slope>
```

```
[ :SOURce ] [ :RF<channel> ] :INPut:TRIGger:SLOPe?
```

**Description**

Sets the polarity of the input signal of the external trigger for the specified channel.

Queries the polarity of the input signal of the external trigger for the specified channel.

**Parameter**

| Name      | Type     | Range             | Default  |
|-----------|----------|-------------------|----------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1        |
| <slope>   | Discrete | POSitive NEGative | POSitive |

**Remarks**

- **POSitive:** sets the polarity of the input signal of the external trigger to "Positive".
- **NEGative:** sets the polarity of the input signal of the external trigger to "Negative".

This command is only valid when "Ext" is selected for the trigger type of sweep.

The external trigger signal is input from the front-panel **[TRIGGER]** connector.

**Return Format**

The query returns POS or NEG.

**Example**

```
:INP:TRIG:SLOP POS /*Sets the polarity of the input signal of the
external trigger for CH1 to "Positive" (by default, CH1 is
selected).*/
:INP:TRIG:SLOP? /*The query returns POS.*/
```

**3.2 :TRIGger Commands**

The :TRIGger commands are used to generate a trigger event.

### 3.2.1 :TRIGger[:RF]:PULM[:IMMediate]

#### Syntax

```
:TRIGger[:RF<channel>]:PULM[:IMMediate]
```

#### Description

When the "Trig Mode" of pulse modulation of the specified channel is set to "Bus", the instrument starts a pulse modulation each time this command is sent.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

#### Remarks

N/A

#### Return Format

N/A

#### Example

N/A

### 3.2.2 :TRIGger:RFALL:PULM[:IMMediate]

#### Syntax

```
:TRIGger:RFALL:PULM[:IMMediate]
```

#### Description

When the "Trig Mode" of pulse modulation is set to "Bus", the instrument starts one pulse modulation for all the channels immediately.

#### Parameter

N/A

#### Remarks

N/A

#### Return Format

N/A

#### Example

N/A

### 3.2.3 :TRIGger[:RF][:SWEep][:IMMediate]

#### Syntax

```
:TRIGger[:RF<channel>][:SWEep][:IMMediate]
```

#### Description

Triggers a RF sweep immediately on the specified channel.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

#### Remarks

When the "Trig Type" or "Point Trig" mode of SWEEP is set to "Bus" and the corresponding trigger condition is met, the instrument starts a RF sweep within the sweep period or sweeps a point and then stops each time this command is sent.

#### Return Format

N/A

#### Example

N/A

### 3.2.4 :TRIGger:RFALL[:SWEep][:IMMediate]

#### Syntax

```
:TRIGger:RFALL[:SWEep][:IMMediate]
```

#### Description

Triggers one RF sweep immediately on all the channels.

#### Parameter

N/A

#### Remarks

N/A

#### Return Format

N/A

#### Example

N/A

### 3.2.5 :TRIGger[:RF]:PULM:KEY

#### Syntax

**:TRIGger** [ **:RF** <channel> ] **:PULM:KEY**

#### Description

When the "Trig Mode" of pulse modulation of the specified channel is set to "Key", the instrument starts one pulse modulation for the specified channel immediately.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

#### Remarks

N/A

#### Return Format

N/A

#### Example

N/A

### 3.2.6 :TRIGger:RFALL:PULM:KEY

#### Syntax

**:TRIGger:RFALL:PULM:KEY**

#### Description

When the "Trig Mode" of pulse modulation of the specified channel is set to "Key", the instrument starts one pulse modulation for all the channels immediately.

#### Parameter

N/A

#### Remarks

N/A

#### Return Format

N/A

#### Example

N/A

### 3.2.7 :TRIGger[:RF][:SWEep]:KEY

#### Syntax

```
:TRIGger [ :RF<channel> ] [ :SWEep ] :KEY
```

#### Description

When the "Trig Mode" of pulse modulation of the specified channel is set to "Key", the instrument starts one RF sweep for the specified channel immediately.

#### Parameter

| Name      | Type     | Range           | Default |
|-----------|----------|-----------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8 | 1       |

#### Remarks

When the "Trig Type" or "Point Trig" mode of SWEEP is set to "Key" and the corresponding trigger condition is met, the instrument starts an RF sweep within the sweep period or sweeps a point and then stops each time this command is sent.

#### Return Format

N/A

#### Example

N/A

### 3.2.8 :TRIGger:RFALL[:SWEep]:KEY

#### Syntax

```
:TRIGger:RFALL [ :SWEep ] :KEY
```

#### Description

When the "Trig Mode" of pulse modulation of the specified channel is set to "Key", the instrument starts one RF sweep for all the channels immediately.

#### Parameter

N/A

#### Remarks

When the "Trig Type" or "Point Trig" mode of SWEEP is set to "Key" and the corresponding trigger condition is met, the instrument starts an RF sweep within the sweep period or sweeps a point and then stops each time this command is sent.

**Return Format**

N/A

**Example**

N/A

## 3.3 :UNIT Command

The :UNIT command is used to set the output and displayed units for the instrument.

### 3.3.1 :UNIT[:RF]:POWer

**Syntax**

```
:UNIT [ :RF<channel> ] :POWer <unit>
```

```
:UNIT [ :RF<channel> ] :POWer?
```

**Description**

Sets the unit for the output and displayed amplitude of the specified channel.

Queries the unit for the output and displayed amplitude of the specified channel.

**Parameter**

| Name      | Type     | Range             | Default |
|-----------|----------|-------------------|---------|
| <channel> | Discrete | 1 2 3 4 5 6 7 8   | 1       |
| <unit>    | Discrete | dBm dBmV dBuV V W | dBm     |

**Remarks**

N/A

**Return Format**

The query returns dBm, dBmV, dBuV, V, or W.

**Example**

```
:UNIT:POW V /*Sets the unit for the output and displayed amplitude
of CH1 to V (by default, CH1 is selected).*/
:UNIT:POW? /*The query returns V.*/
```

## 3.4 :SYSTem Commands

The :SYSTem commands are used to set a series of parameters relating to the system and the settings of these parameters do not affect the output signal of the microwave signal generator.

### 3.4.1 :SYSTem:COMMunication:LAN:DHCP

#### Syntax

`:SYSTem:COMMunication:LAN:DHCP <bool>`

`:SYSTem:COMMunication:LAN:DHCP?`

#### Description

Enables or disables the DHCP configuration mode.

Queries the status of the DHCP configuration mode.

#### Parameter

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 1 ON    |

#### Remarks

- In DHCP mode, the DHCP server in the current network assigns the network parameters (e.g. IP address) for the signal generator.
- When the three IP configuration modes are all enabled, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Manual IP".
- The three IP configuration modes cannot be all disabled at the same time.

#### Return Format

The query returns 1 or 0.

#### Example

```
:SYST:COMM:LAN:DHCP ON          /*Enables the DHCP mode.*/  
:SYST:COMM:LAN:DHCP?           /*The query returns 1.*/
```

### 3.4.2 :SYSTem:COMMunication:LAN:IP:ADDRESS

#### Syntax

`:SYSTem:COMMunication:LAN:IP:ADDRESS <value>`

`:SYSTem:COMMunication:LAN:IP:ADDRESS?`

#### Description

Sets the current IP address.

Queries the current IP address.



**Parameter**

| Name    | Type         | Range  | Default |
|---------|--------------|--|---------|
| <value> | ASCII String | The format of <value> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) of the value is from 1 to 223 (except 127); and the range for the other three segments is from 0 to 255. | -       |

**Remarks**

- This command is only valid when the Manual IP configuration mode is enabled.
- You are recommended to ask your network administrator for an address available.

**Return Format**

The query returns the IP address. For example, 172.16.3.199.

**Example**

```
:SYST:COMM:LAN:IP:ADD 172.16.3.199 /*Sets the IP address to
172.16.3.199.*/
:SYST:COMM:LAN:IP:ADD? /*The query returns 172.16.3.199*/
```

### 3.4.3 :SYSTem:COMMunication:LAN:IP:AUTO

**Syntax**

:SYSTem:COMMunication:LAN:IP:AUTO <bool>

:SYSTem:COMMunication:LAN:IP:AUTO?

**Description**

Enables or disables the auto IP configuration mode.

Queries the on/off status of the auto IP configuration mode.

**Parameter**

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 1 ON    |

**Remarks**

- In AutoIP mode, the instrument will get the IP address (ranging from 169.254.0.1 to 169.254.255.254) and the subnet mask (255.255.0.0) automatically according to the current network configuration.
- When the three IP configuration modes are all enabled, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Manual IP".
- The three IP configuration modes cannot be all disabled at the same time.

**Return Format**

The query returns 1 or 0.

**Example**

```
:SYST:COMM:LAN:IP:AUTO ON /*Enables the Auto IP configuration mode.*/
:SYST:COMM:LAN:IP:AUTO? /*The query returns 1.*/
```

### 3.4.4 :SYSTem:COMMunication:LAN:IP:GATeway

**Syntax**

:SYSTem:COMMunication:LAN:IP:GATeway <value>

:SYSTem:COMMunication:LAN:IP:GATeway?

**Description**

Sets the current default gateway.

Queries the current default gateway.

**Parameter**

| Name    | Type         | Range  | Default |
|---------|--------------|--|---------|
| <value> | ASCII String | The format of <value> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) of the value is from 1 to 223 (except 127); and the range for the other three segments is from 0 to 255. | -       |

**Remarks**

- This command is only valid when the Manual IP configuration mode is enabled.

- You are recommended to ask your network administrator for an address available.

#### Return Format

The query returns the default gateway. For example, 172.16.3.1

#### Example

```
:SYST:COMM:LAN:IP:GAT 172.16.3.1 /*Sets the default gateway to
172.16.3.1.*/
:SYST:COMM:LAN:IP:GAT? /*The query returns 172.16.3.1*/
```

### 3.4.5 :SYSTem:COMMunication:LAN:IP:MANual

#### Syntax

```
:SYSTem:COMMunication:LAN:IP:MANual <bool>
```

```
:SYSTem:COMMunication:LAN:IP:MANual?
```

#### Description

Enables or disables the manual IP configuration mode.

Queries the status of the manual IP configuration mode.

#### Parameter

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 0 OFF   |

#### Remarks

- In Manual IP mode, the network parameters of the generator, e.g. IP address, are defined by users.
- When the three IP configuration modes are all enabled, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Manual IP".
- The three IP configuration modes cannot be all disabled at the same time.

#### Return Format

The query returns 1 or 0.

#### Example

```
:SYST:COMM:LAN:IP:MAN ON /*Enables the manual IP
configuration mode.*/
:SYST:COMM:LAN:IP:MAN? /*The query returns 1.*/
```

### 3.4.6 :SYSTem:COMMunication:LAN:IP:SET

#### Syntax

```
:SYSTem:COMMunication:LAN:IP:SET
```

#### Description

Applies the currently set network parameters.

#### Parameter

N/A

#### Remarks

After setting the LAN related parameters, you must execute this command to apply the parameters, otherwise the settings will not take effect.

#### Return Format

N/A

#### Example

N/A

### 3.4.7 :SYSTem:COMMunication:LAN:IP:SUBnet:MASK

#### Syntax

```
:SYSTem:COMMunication:LAN:IP:SUBnet:MASK <value>
```

```
:SYSTem:COMMunication:LAN:IP:SUBnet:MASK?
```

#### Description

Sets the current subnet mask.

Queries the current subnet mask.

#### Parameter

| Name    | Type         | Range  | Default |
|---------|--------------|--|---------|
| <value> | ASCII String | The format of <value> is "nnn.nnn.nnn.nnn". The range of "nnn" is from 0 to 255. | -       |

#### Remarks

- This command is only valid when the Manual IP configuration mode is enabled.

- You are recommended to ask your network administrator for a subnet mask available.

**Return Format**

The query returns the subnet mask. For example, 255.255.255.0

**Example**

```
:SYST:COMM:LAN:IP:SUB:MASK 255.255.255.0 /*Sets the current subnet mask to 255.255.255.0.*/  
:SYST:COMM:LAN:IP:SUB:MASK? /*The query returns 255.255.255.0*/
```

### 3.4.8 :SYSTem:COMMunication:LAN:RESet

**Syntax**

```
:SYSTem:COMMunication:LAN:RESet
```

**Description**

Resets the currently set network parameters.

**Parameter**

N/A

**Remarks**

After resetting the current parameters, DHCP and Auto-IP are turned on and Manual-IP is turned off.

**Return Format**

N/A

**Example**

N/A

### 3.4.9 :SYSTem:COMMunication:LAN[:SELF]:PREFferred

**Syntax**

```
:SYSTem:COMMunication:LAN[:SELF]:PREFferred <value>
```

```
:SYSTem:COMMunication:LAN[:SELF]:PREFferred?
```

**Description**

Sets the DNS.

Queries the current DNS.

**Parameter**

| Name    | Type         | Range  | Default |
|---------|--------------|--|---------|
| <value> | ASCII String | The format of <value> is "nnn.nnn.nnn.nnn". The range for the first segment (nnn) of the value is from 1 to 223 (except 127); and the range for the other three segments is from 0 to 255. | -       |

**Remarks**

You are recommended to ask your network administrator for an address available.

**Return Format**

The query returns the DNS address. For example, 172.16.2.3.

**Example**

```
:SYST:COMM:LAN:PREF 172.16.2.3 /*Sets the DNS address to
172.16.2.3.*/
:SYST:COMM:LAN:PREF? /*The query returns 172.16.2.3*/
```

### 3.4.10 :SYSTem:COMMunication:LAN:MDNS

**Syntax**

```
:SYSTem:COMMunication:LAN:MDNS <bool>
```

```
:SYSTem:COMMunication:LAN:MDNS?
```

**Description**

Enables or disables mDNS.

Queries the on/off status of the mDNS port.

**Parameter**

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 0 OFF   |

**Remarks**

N/A

**Return Format**

The query returns 1 or 0.

**Example**

```
:SYSTem:COMMunication:LAN:MDNS ON /*Enables mDNS.*/  
:SYSTem:COMMunication:LAN:MDNS? /*The query returns 1.*/
```

### 3.4.11 :SYSTem:COMMunication:LAN:HOST:NAME

**Syntax**

```
:SYSTem:COMMunication:LAN:HOST:NAME <name>
```

```
:SYSTem:COMMunication:LAN:HOST:NAME?
```

**Description**

Sets the host name.

Queries the host name.

**Parameter**

| Name   | Type         | Range | Default |
|--------|--------------|-------|---------|
| <name> | ASCII String | -     | -       |

**Remarks**

N/A

**Return Format**

The query returns the host name.

**Example**

```
N/A
```

### 3.4.12 :SYSTem:COMMunication:LAN:MAC

**Syntax**

```
:SYSTem:COMMunication:LAN:MAC?
```

**Description**

Queries the MAC address.

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

The query returns the MAC address in strings.

**Example**

N/A

### 3.4.13 :SYSTem:PRESet

**Syntax**

:SYSTem:PRESet

**Description**

Resets the instrument to the preset state (the settings (FACtory or USER) defined by the :SYSTem:PRESet:TYPE command).

**Parameter**

N/A

**Remarks**

This command has the same function as clicking or tapping **Preset**, namely recalling the default values or user-preset values.

**Return Format**

N/A

**Example**

N/A

### 3.4.14 :SYSTem:PRESet:TYPE

**Syntax**

:SYSTem:PRESet:TYPE <type>

:SYSTem:PRESet:TYPE?

**Description**

Sets the preset type of the system.

Queries the preset type of the system.

**Parameter**

| Name   | Type     | Range        | Default |
|--------|----------|--------------|---------|
| <type> | Discrete | FACtory USER | -       |



**Remarks**

- **FACTory**: sets the preset type of the system to "Factory".
- **USER**: sets the preset type of the system to "User". At this point, you can run the `:SYSTem:PRESet:SAVE` command to save the current system configuration.

**Return Format**

The query returns FAC or USER.

**Example**

```
:SYST:PRES:TYPE USER /*Sets the preset type to USER.*/  
:SYST:PRES:TYPE? /*The query returns USER.*/
```

### 3.4.15 :SYSTem:PRESet:SAVE

**Syntax**

`:SYSTem:PRESet:SAVE`

**Description**

Saves the user setting.

**Parameter**

N/A

**Remarks**

- Using this command can save the current system configuration as user-defined setting to the internal non-volatile memory.
- When Preset Type is set to "User" (use the `:SYSTem:PRESet:TYPE` command), this configuration will be loaded.
- When Preset Type is set to "Factory", this command is invalid.

**Return Format**

N/A

**Example**

N/A

### 3.4.16 :SYSTem:FSWitch:STATe

**Syntax**

`:SYSTem:FSWitch:STATe <type>`

**:SYSTem:FSWitch:STATe?**

### Description

Sets the power status of the signal generator after power-on.

Queries the power status of the signal generator after power-on.

### Parameter

| Name   | Type     | Range        | Default |
|--------|----------|--------------|---------|
| <type> | Discrete | OPEN DEFault | -       |

### Remarks

- **OPEN:** selects "Open" state. The microwave signal generator starts automatically after power-on.
- **DEFault:** selects "Default" state. You have to press the power key on the front panel to start the microwave signal generator after the instrument is connected to AC power.

### Return Format

The query returns OPEN or DEF.

### Example

```
:SYST:FSW:STAT OPEN /*Sets the power status of the signal generator
to OPEN.*/
:SYST:FSW:STAT /*The query returns OPEN.*/
```

## 3.4.17 :SYSTem:LANGuage

### Syntax

**:SYSTem:LANGuage <type>**

**:SYSTem:LANGuage?**

### Description

Sets the system language.

Queries the system language.

### Parameter

| Name   | Type     | Range           | Default |
|--------|----------|-----------------|---------|
| <type> | Discrete | ENGLISH CHINEse | -       |

**Remarks**

- **ENGLISH:** indicates that the system language is English.
- **CHINESE:** indicates that the system language is Simplified Chinese.

**Return Format**

The query returns ENGL or CHIN.

**Example**

```
:SYST:LANG CHIN /*Sets the system language to "Simplified
Chinese".*/
:SYST:LANG /*The query returns CHIN.*/
```

### 3.4.18 :SYSTem:DATE

**Syntax**

:SYSTem:DATE <year>,<month>,<day>

:SYSTem:DATE?

**Description**

Sets the date of the instrument.

Queries the date of the instrument.

**Parameter**

| Name    | Type         | Range        | Default |
|---------|--------------|--------------|---------|
| <year>  | ASCII String | 2000 to 2099 | -       |
| <month> | ASCII String | 01 to 12     | -       |
| <day>   | ASCII String | 01 to 31     | -       |

**Remarks**

N/A

**Return Format**

The query returns the current date in the format of "YYYY,MM,DD". For example, "2020,08,31".

**Example**

```
:SYST:DATE 2020,08,31 /*Sets the system date to 2020/08/31.*/
:SYST:DATE? /*The query returns 2020,08,31.*/
```

### 3.4.19 :SYSTem:TIME

#### Syntax

:SYSTem:TIME <hour>,<min>,<sec>

:SYSTem:TIME?

#### Description

Sets the system time of the instrument.

Queries the system time of the instrument.

#### Parameter

| Name   | Type         | Range    | Default |
|--------|--------------|----------|---------|
| <hour> | ASCII String | 00 to 23 | -       |
| <min>  | ASCII String | 00 to 59 | -       |
| sec    | ASCII String | 00 to 59 | -       |

#### Remarks

N/A

#### Return Format

The query returns the current system time in the format of "hh,mm,ss". For example, "16,40,30".

#### Example

```
:SYST:TIME 16,40,30      /*Sets the system time to 16:40:30.*/
:SYST:TIME?              /*The query returns 16,40,30.*/
```

### 3.4.20 :SYSTem:STIME

#### Syntax

:SYSTem:STIME <bool>

:SYSTem:STIME?

#### Description

Enables or disables the display of the system date and time.

Queries whether to display the system date and time.

**Parameter**

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 1 ON    |

**Remarks**

N/A

**Return Format**

The query returns 1 or 0.

**Example**

```
:SYSTem:STIME ON          /*Enables the display of the system date and  
time.*/  
:SYSTem:STIME?            /*The query returns 1.*/
```

### 3.4.21 :SYSTem:CLEar

**Syntax**

:SYSTem:CLEar

**Description**

Clear all the user-defined data safely.

**Parameter**

N/A

**Remarks**

The instrument will be reset to the factory setting after the data are cleared.

- Format the NAND FLASH;
- Reset the user data saved in NVRAM and NorFlash to the factory setting;
- Reset the HOST NAME, IP address and password in LXI to the factory setting.

**Return Format**

N/A

**Example**

N/A

### 3.4.22 :SYSTem:DISPlay:UPDate[:STATe]

#### Syntax

`:SYSTem:DISPlay:UPDate[:STATe] <bool>`

`:SYSTem:DISPlay:UPDate[:STATe]?`

#### Description

Enables or disables the touch screen.

Queries the on/off status of the touch screen.

#### Parameter

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 0 OFF   |

#### Remarks

When enabled, the screen stops updating and is locked. At this point, the measurement speed is improved. Screen locking is mainly used in remote operation mode.

#### Return Format

The query returns 1 or 0.

#### Example

```
:SYST:DISP:UPD OFF /*Enables the touch screen.*/  
:SYST:DISP:UPD? /*The query returns 0.*/
```

### 3.4.23 :SYSTem:LKEY

#### Syntax

`:SYSTem:LKEY <option>@<license_key>`

`:SYSTem:LKEY? <option>`

#### Description

Used to install and activate the options of the instrument.

Queries the license of the specified option.

**Parameter**

| Name          | Type         | Range                                  | Default |
|---------------|--------------|--|---------|
| <option>      | ASCII String | DSG5000-PUL DSG5000-PUG DSG5000-AMD    | -       |
| <license_key> | ASCII String | License of the option you've purchased | -       |

**Remarks**

The license for each option is unique and can only be used by one instrument (namely, the license of the option corresponds to the serial number of the instrument you bought).

**Return Format**

N/A

**Example**

```
:SYSTem:LKEY:DSG5000-AMD@fbcd61febe039cfa3330036b8feFFc885a2210b4a31f620d370f0006931d6575a5cea7eee0f355f0adc33f936bfcbl3a /*Installs and activates the DSG5000-AMD option.*/
:SYSTem:LKEY? DSG5000-AMD /*The query returns fbcd61febe039cfa3330036b8feFFc885a2210b4a31f620d370f0006931d6575a5cea7eee0f355f0adc33f936bfcbl3a.*/
```

### 3.4.24 :SYSTem:LKEY:DElete

**Syntax**

**:SYSTem:LKEY:DElete**

**Description**

Uninstalls all options.

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

N/A

**Example**

N/A

### 3.4.25 :SYSTem:OPTion:LIST?

**Syntax**

:SYSTem:OPTion:LIST?

**Description**

Queries the installation status of all the options.

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

N/A

**Example**

N/A

### 3.4.26 :SYSTem:CONFigure:INFormation?

**Syntax**

:SYSTem:CONFigure:INFormation?

**Description**

Queries the configuration of the system, such as the software and hardware version number.

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

The query returns the system information of the current system.

**Example**

N/A



## 3.5 :MMEMory Commands

The :MMEMory commands are used to store the file to the internal or external memory of the instrument, read or delete the specified file as well as query the disk information, etc.

### 3.5.1 :MMEMory:CATalog

#### Syntax

**:MMEMory:CATalog?** *<path>*

#### Description

Queries all the files and folders under the specified path.

#### Parameter

| Name                | Type         | Range      | Default |
|---------------------|--------------|------------|---------|
| <i>&lt;path&gt;</i> | ASCII String | Valid path | -       |

#### Remarks

- The parameter *<path>* can be the local memory (Disk C) or external memory (Disk D, available when a USB storage device is connected to the front-panel USB Host interface), or the folders under Disk C or D.
- The query returns a list of all the files and folders under the path specified by *<path>*.

#### Return Format

NO.1 File Name: 1

NO.2 File Name: 4.STA

#### Example

```
:MMEM:CAT? C: /*The query returns all the files and folders under  
Disk C.*/
```

### 3.5.2 :MMEMory:CATalog:LENGth

#### Syntax

**:MMEMory:CATalog:LENGth?** *<path>*

**Description**

Queries the number of files and folders under the specified path.

**Parameter**

| Name   | Type         | Range      | Default |
|--------|--------------|------------|---------|
| <path> | ASCII String | Valid path | -       |

**Remarks**

The parameter <path> can be the local memory (Disk C) or external memory (Disk D, available when a USB storage device is connected to the front-panel USB Host interface), or the folders under Disk C or D.

**Return Format**

The query returns an integer. For example, 2.

**Example**

```
:MMEM:CAT:LENG? C: /*The query returns the number of files and
folders under Disk C.*/
```

### 3.5.3 :MMEMory:COPY

**Syntax**

```
:MMEMory:COPY <file_source>,<file_destination>
```

**Description**

Copies the file or folder specified by <file\_source> to the destination path specified by <file\_destination>.

**Parameter**

| Name               | Type         | Range                       | Default |
|--------------------|--------------|-----------------------------|---------|
| <file_source>      | ASCII String | File or folder to be copied | -       |
| <file_destination> | ASCII String | Valid destination path      | -       |

**Remarks**

- The parameter <file\_source> denotes the file or folder to be copied. The file or folder name must contain the path. The parameter <file\_destination> indicates the destination path and does not include the filename.

- If the file or folder specified by <file\_source> does not exist, the copy operation fails.
- If the destination path specified by <file\_destination> does not exist, the copy operation fails.

**Return Format**

N/A

**Example**

```
:MMEM:COPY C:\1.sta,C:\New /*Copies the file "1.sta" under Disk C to the "New" folder under Disk C.*/
```

### 3.5.4 :MMEMory:DElete

**Syntax**

```
:MMEMory:DElete <file_name>
```

**Description**

Deletes the specified file or folder under the specified operation path.

**Parameter**

| Name        | Type         | Range                                    | Default |
|-------------|--------------|--|---------|
| <file_name> | ASCII String | Name of the file or folder to be deleted | -       |

**Remarks**

- This command is only valid when the specified file or folder exists under the current operation path or the specified path.
- The parameter <file\_name> can be a file name or folder name under the current operation path or the specified path. For example, :MMEM:DEL C:\NEW\3.sta.

**Return Format**

N/A

**Example**

```
:MMEM:DEL C:\8.sta /*Deletes a file named "8.sta" from Disk C.*/
```

### 3.5.5 :MMEMory:DISK:FORMat

#### Syntax

**:MMEMory:DISK:FORMat**

#### Description

Formats the local disk (Disk C).

#### Parameter

N/A

#### Remarks

N/A

#### Return Format

N/A

#### Example

N/A

### 3.5.6 :MMEMory:DISK:INFormation

#### Syntax

**:MMEMory:DISK:INFormation? <disk>**

#### Description

Queries the information of the disk.

#### Parameter

| Name   | Type         | Range | Default |
|--------|--------------|-------|---------|
| <disk> | ASCII String | C:    | C:      |

#### Remarks

N/A

#### Return Format

The query returns the disk information, including Disk Name, File System, Used Space and Total Space. For example,

Disk Type:Local Disk(C)

File System:FAT32

Used Space:527.848MB

Total Space:113.619GB

### Example

```
:MMEM:DISK:INF? C: /*The query returns the information of Disk C.*/
```

## 3.5.7 :MMEMory:FILEtype

### Syntax

```
:MMEMory:FILEtype <type>
```

```
:MMEMory:FILEtype?
```

### Description

Sets the file type.

Queries the current file type.

### Parameter

| Name   | Type     | Range                            | Default |
|--------|----------|----------------------------------|---------|
| <type> | Discrete | STATe FLACsv SWPCsv TRNCsv IMAGe | STATe   |

### Remarks

- The file types include state file, flatness csv file, sweep csv file, pulse csv file, and image type.
- After selecting the appropriate file type, you can view all files or save a new file of this file type.

### Return Format

The query returns STAT, FLAC, SWPC, TRNC, or IMAG.

### Example

```
:MMEM:FILE TRNCsv /*Sets the file type to TRNCsv.*/
:MMEM:FILE? /*The query returns TRNC.*/
```

## 3.5.8 :MMEMory:CHANnel:SElect

### Syntax

```
:MMEMory:CHANnel:SElect<channel>
```

### Description

Selects the specified channel for saving or loading the file.

**Parameter**

| Name      | Type     | Range               | Default |
|-----------|----------|---------------------|---------|
| <channel> | Discrete | ALL 1 2 3 4 5 6 7 8 | ALL     |

**Remarks**

- When <channel> is set to ALL, it indicates all the channels.
- You need to run this command first to configure the specified channel, then run the *:MMEMory:LOAD* or *:MMEMory:SAVe* command to save or load the file of the specified channel.

**Return Format**

N/A

**Example**

N/A

### 3.5.9 :MMEMory:LDISK:SPACe

**Syntax**

**:MMEMory:LDISK:SPACe?**

**Description**

Queries the space information of the local disk (Disk C).

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

The query returns the space information of Disk C, including "Used space" and "Free space".

For example, Used:512 k,Free:1048064 k.

**Example**

N/A

### 3.5.10 :MMEMory:LOAD

#### Syntax

**:MMEMory:LOAD** <file\_name>

#### Description

Loads the specified file from the specified operation path.

#### Parameter

| Name        | Type         | Range                         | Default |
|-------------|--------------|-------------------------------|---------|
| <file_name> | ASCII String | Name of the file to be loaded | -       |

#### Remarks

- This command is only valid when the specified file exists under the current operation path or the specified path.
- The parameter <file\_name> can be a file name under the current operation path or a file name under the specified path. For example, :MMEM:LOAD C:\NEW\1.sta.

#### Return Format

N/A

#### Example

```
:MMEM:LOAD C:\1.trn /*Loads the file named 1.sta from Disk C.*/
```

### 3.5.11 :MMEMory:MDIRectory

#### Syntax

**:MMEMory:MDIRectory** <directory\_name>

#### Description

Creates a new folder under the specified operation path.

#### Parameter

| Name             | Type         | Range                            | Default |
|------------------|--------------|----------------------------------|---------|
| <directory_name> | ASCII String | Name of the folder to be created | -       |

**Remarks**

- The folder name can include Chinese characters (one Chinese character occupies two bytes), English characters, and numbers. The filename cannot exceed 28 bytes.
- If the name of the folder to be created already exists, this operation is invalid. At this point, "The filename already exists" is displayed in the user interface.
- The parameter <directory\_name> can be a new folder name without a path location being added, denoting creating a folder under the current operation path; or a folder name with the specified path being added, denoting creating a new folder under the specified path. For example, :MMEM:MDIR C:\1\NEW.

**Return Format**

N/A

**Example**

```
:MMEM:MDIR C:\NEW /*Creates a folder named NEW under Disk C.*/
```

### 3.5.12 :MMEMory:MOVE

**Syntax**

:MMEMory:MOVE <file\_source>,<file\_destination>

**Description**

Renames the file or folder specified by <file\_source> to the destination file or folder name specified by <file\_destination>.

**Parameter**

| Name               | Type         | Range                     | Default |
|--------------------|--------------|---------------------------|---------|
| <file_source>      | ASCII String | Valid file or folder name | -       |
| <file_destination> | ASCII String | Valid file or folder name | -       |

**Remarks**

- The file or folder name specified by <file\_source> and <file\_destination> must contain the path.



- If the file or folder specified by <file\_source> does not exist, the rename operation fails.
- If the destination file or folder name specified by <file\_destination> already exists under the current path, the rename operation fails.

#### Return Format

N/A

#### Example

```
:MMEM:MOVE C:\1.sta,C:\2.sta /*Renames the filename 1.sta in Disk C to 2.sta.*/
```

### 3.5.13 :MMEMory:PNAME:EDIT

#### Syntax

```
:MMEMory:PNAME:EDIT <pre_name>
```

```
:MMEMory:PNAME:EDIT?
```

#### Description

Edits and saves the filename prefix.

Queries the filename prefix saved.

#### Parameter

| Name       | Type         | Range                        | Default |
|------------|--------------|------------------------------|---------|
| <pre_name> | ASCII String | Filename prefix to be edited | -       |

#### Remarks

You can edit any prefix name.

#### Return Format

The query returns the filename prefix. For example, N.

#### Example

```
:MMEM:PNAME:EDIT N /*Sets the filename prefix to N.*/
:MMEM:PNAME:EDIT? /*The query returns N.*/
```

### 3.5.14 :MMEMory:PNAME:STATe

#### Syntax

```
:MMEMory:PNAME:STATe <bool>
```

**:MMEMory:PNAMe:STATe?**

### Description

Enables or disables the filename prefix.

Queries the current on/off state of the filename prefix.

### Parameter

| Name   | Type | Range      | Default |
|--------|------|------------|---------|
| <bool> | Bool | 1 ON 0 OFF | 0 OFF   |

### Remarks

- You can select "On" or "Off" to enable or disable the prefix edited.
- When "On" is selected, the edited prefix will be added to the filename input field automatically when saving the file.

### Return Format

The query returns 0 or 1.

### Example

```
:MMEM:PNAM:STATE ON /*Enables the filename prefix.*/  
:MMEM:PNAM:STAT? /*The query returns 1.*/*
```

## 3.5.15 :MMEMory:SAVe

### Syntax

**:MMEMory:SAVe** <File\_name>

### Description

Saves the file with the specified filename under the current operation path.

### Parameter

| Name        | Type         | Range                        | Default |
|-------------|--------------|------------------------------|---------|
| <File_name> | ASCII String | Name of the file to be saved | -       |

### Remarks

- The filename can include Chinese characters (one Chinese character occupies two bytes), English characters and numbers. The filename cannot exceed 28 bytes.

- When the current path already contains a file with the same name, this command will directly overwrite the original file.

**Return Format**

N/A

**Example**

```
:MMEM:SAV SET.sta /*Saves the current instrument state with the  
SET.sta filename under the current operation path.*/
```

## 3.6 IEEE488.2 Common Commands

The IEEE488.2 common commands are used to query the basic information of the instrument or executing basic operations. These commands usually start with "\*", and the command keywords contain 3 characters and are related with status registers.

### 3.6.1 \*CLS

**Syntax**

\*CLS

**Description**

Clears all the event registers, and also clears the error queue.

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

N/A

**Example**

N/A

### 3.6.2 \*ESE

**Syntax**

\*ESE <value>

\*ESE?

**Description**

Sets or queries the enable register of the standard event status register.

**Parameter**

| Name    | Type    | Range    | Default |
|---------|---------|----------|---------|
| <value> | Integer | 0 to 255 | 0       |

**Remarks**

Bit 1 and Bit 6 are not used and are always treated as 0; therefore, the range of <value> is a decimal number that corresponds to the binary numbers from 00000000 (0 in decimal 0) to 11111111 (255 in decimal) and of which Bit 1 and Bit 6 are 0.

**Return Format**

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register.

**Example**

```
*ESE 16      /*Enables the Bit 4 (16 in decimal) in the register.*/  
*ESE?       /*The query returns 16.*/
```

### 3.6.3 \*IDN?

**Syntax**

\*IDN?

**Description**

Queries the ID string of the instrument.

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

The query returns the ID string of the instrument. For example, RIGOL TECHNOLOGIES,DSG5208,00.00.01.00.13,00.00.01.

**Example**

N/A

### 3.6.4 \*PSC

#### Syntax

\*PSC <value>

\*PSC?

#### Description

Retains or resets the settings of the status register.

Queries the "power-on-status-clear" flag of the status register.

#### Parameter

| Name    | Type | Range | Default |
|---------|------|-------|---------|
| <value> | Bool | 0 1   | -       |

#### Remarks

- **0:** retains the settings of the status register.
- **1:** resets the settings of the status register.

#### Return Format

The query returns 1 or 0.

#### Example

N/A

### 3.6.5 \*RST

#### Syntax

\*RST

#### Description

Restores the microwave signal generator to the preset state (factory or user).

#### Parameter

N/A

#### Remarks

N/A

#### Return Format

N/A

**Example**

N/A

**3.6.6 \*SRE****Syntax****\*SRE** <value>**\*SRE?****Description**

Sets the enable register for the status byte register.

Queries the enable register for the status byte register.

**Parameter**

| Name    | Type    | Range    | Default |
|---------|---------|----------|---------|
| <value> | Integer | 0 to 255 | 0       |

**Remarks**

Bit 0 and Bit 1 are not used and are always treated as 0; therefore, the range of <value> is a decimal number that corresponds to the binary numbers from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which Bit 0 and Bit 1 are 0.

**Return Format**

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register.

**Example**

```
*SRE 16      /*Enables the Bit 4 (16 in decimal) in the register.*/
*SRE?        /*The query returns 16.*/
```

**3.6.7 \*STB?****Syntax****\*STB?****Description**

Queries the event register for the status byte register. After executing the command, the value in the status byte register is cleared.

**Parameter**

N/A

**Remarks**

Bit 0 and Bit 1 are not used and are always treated as 0; therefore, the range of <value> are the decimal numbers corresponding to the binary numbers ranging from 00000000 (0 in decimal) to 11111111 (255 in decimal) and of which Bit 0 and Bit 1 are 0.

**Return Format**

The query returns an integer. The integer equals to the binary-weighted sum of all the bits set in the register.

**Example**

N/A

**3.6.8****\*TRG****Syntax**

\*TRG

**Description**

Triggers a pulse modulation or RF sweep immediately.

**Parameter**

N/A

**Remarks**

N/A

**Return Format**

N/A

**Example**

N/A

**3.6.9****\*WAI****Syntax**

\*WAI

**Description**

Waits for all the pending operations to complete before executing any additional commands.

**Parameter**

N/A





### 3.7.1 :STATus:OPERation:ENABle

#### Syntax

`:STATus:OPERation:ENABle <value>`

`:STATus:OPERation:ENABle?`

#### Description

Sets the enable register of the operation status register.

Queries the enable register of the operation status register.

#### Parameter

| Name    | Type    | Range                   | Default |
|---------|---------|-------------------------|---------|
| <value> | Integer | Refer to <i>Remarks</i> | 0       |

#### Remarks

The range of <value> is a decimal value that corresponds to the binary number from 0000000000000000 (0 in decimal) to 1111111111111111 (32767 in decimal).

#### Return Format

The query returns the enable register of the operation status register in integer.

#### Example

```
:STAT:OPER:ENAB 100 /*Sets the enable register of the operation  
status register to 100.*/  
:STAT:OPER:ENAB? /*The query returns 100.*/
```

### 3.7.2 :STATus:OPERation[:EVENT]

#### Syntax

`:STATus:OPERation[:EVENT]?`

#### Description

Queries the event register of the operation status register.

#### Parameter

N/A

#### Remarks

N/A

#### Return Format

The query returns the event register in integer. For example, 32.

**Example**

N/A

## 4 Application Examples

This chapter provides the application instances of the SCPI commands. The main functions of the signal generator can be realized through a series of SCPI commands.



### NOTE

- The examples in this chapter are based on DSG5208. The range of certain parameters for other models may be different. Therefore, you need to adjust the parameter range for the model that you use if necessary.
- Before using the examples in this chapter, please select the desired communication interface (USB or LAN) and make correct connections. In addition, you have to install Ultra Sigma or other PC software that can be used to send commands.
- In each example, every command is followed by contents enclosed by two slashes ("/\*" and "\*/"). They are the descriptions of the command and not part of the command, which help you understand the command better.

### 4.1 To Output the RF Signal

#### Requirement

Use the SCPI commands to realize the following functions:

Output the RF signal with 1 GHz frequency and -20 dBm amplitude from the [RF 50Ω] connector of CH2.

#### Method

```
*IDN? /*Queries the ID string of the instrument to check whether
the remote communication works normally.*/
:SYST:PRES:TYPE FAC /*Sets the preset type to "Factory".*/
:SYST:PRES /*Restores the instrument to the factory setting.*/
:RF2:FREQ 1GHz /*Sets the frequency of the RF signal of CH2 to 1
GHz.*/
:RF2:LEV -20 /*Sets the amplitude of the RF signal of CH2 to -20
dBm.*/
:RF2:OUTPut ON /*Enables the RF output for CH2.*/
```

### 4.2 To Output the RF Sweep Signal

#### Requirement

Use the SCPI commands to realize the following functions:

Configure a continuous linear step sweep to output the RF sweep signal from the [RF 50Ω] connector of CH2 to make its frequency stay within a range from 1 GHz to 2 GHz, amplitude from -20 dBm to 0 dBm, the number of sweep points 10, and the dwell time 500 ms.

## Method

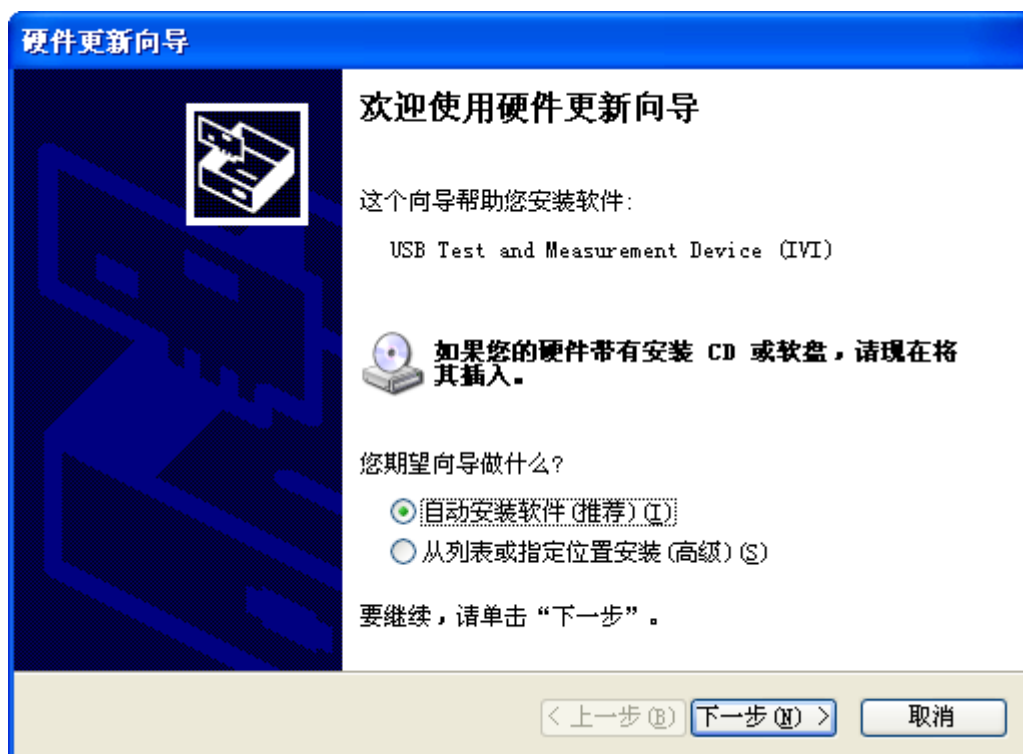
```
*IDN? /*Queries the ID string of the instrument to check whether
the remote communication works normally.*/
:SYST:PRES:TYPE FAC /*Sets the preset type to "Factory".*/
:SYST:PRES /*Restores the instrument to the factory setting (by
default, the sweep mode is continuous, the sweep type is step and
the sweep spacing is linear)*/
:RF2:SWE:STEP:STAR:FREQ 1GHz /*Sets the start frequency of step
sweep for CH2 to 1 GHz.*/
:RF2:SWE:STEP:STOP:FREQ 2GHz /*Sets the stop frequency of step
sweep for CH2 to 2 GHz.*/
:RF2:SWE:STEP:STAR:LEV -20 /*Sets the start level of step sweep
for CH2 to -20 dBm.*/
:RF2:SWE:STEP:STOP:LEV 0 /*Sets the stop level of step sweep for
CH2 to 0 dBm.*/
:RF2:SWE:STEP:POIN 10 /*Sets the number of step sweep points for
CH2 to 10.*/
:RF2:SWE:STEP:DWEL 500ms /*Sets the dwell time of step sweep for
CH2 to 500ms.*/
:RF2:SWE:STAT LEV,FREQ /*Enables the Freq&Level sweep for CH2.*/
:RF2:OUTPut ON /*Enables the RF output for CH2.*/
```

## 5 Programming Examples

### 5.1 Programming Preparations

Before programming, you need to prepare the following tasks:

1. Make sure that your PC has been installed with the NI-VISA library (available to download from the NI website <http://www.ni.com/visa>). In this manual, the default installation path is C:\Program Files\IVI Foundation\VISA.
2. In this manual, the signal generator communicates with the PC via the USB interface. Use the USB cable to connect the signal generator to the PC via the USB DEVICE interface on the rear panel of the signal generator.
3. After the signal generator is properly connected to the PC, connect the signal generator to power source and turn it on.
4. In this case, "Found New Hardware Wizard" dialog box appears on the PC. Please install "USB Test and Measurement Device (IVI)" according to the instructions.



5. Acquire the USB VISA descriptor of the microwave signal generator: run Ultra Sigma and search for the microwave signal generator resource currently connected

to the PC. The resource found is displayed under the "RIGOL Online Resource" directory, including the instrument model and the USB interface information (namely the VISA descriptor), The VISA descriptor of the microwave signal generator used is USB0::0x1AB1::0x0993::DSG5D242600001::INSTR.

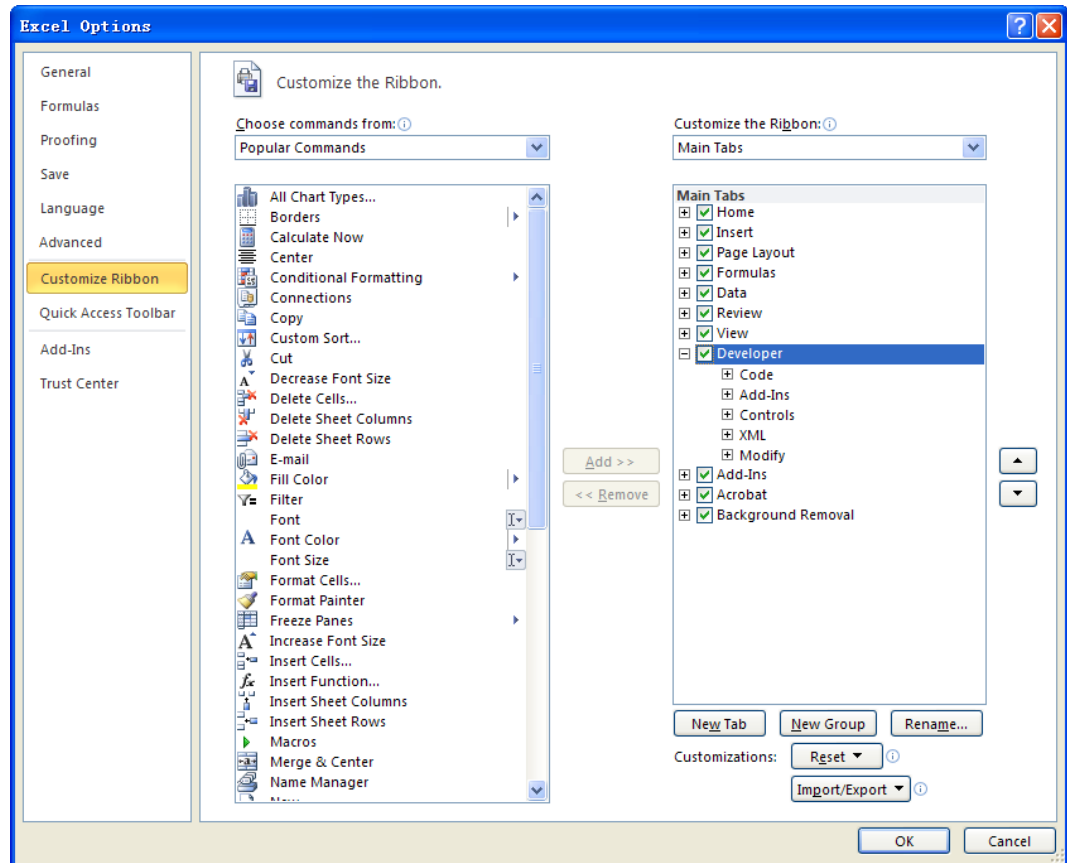
Then, the preparation work for the programming is completed.

## 5.2 Excel Programming Examples

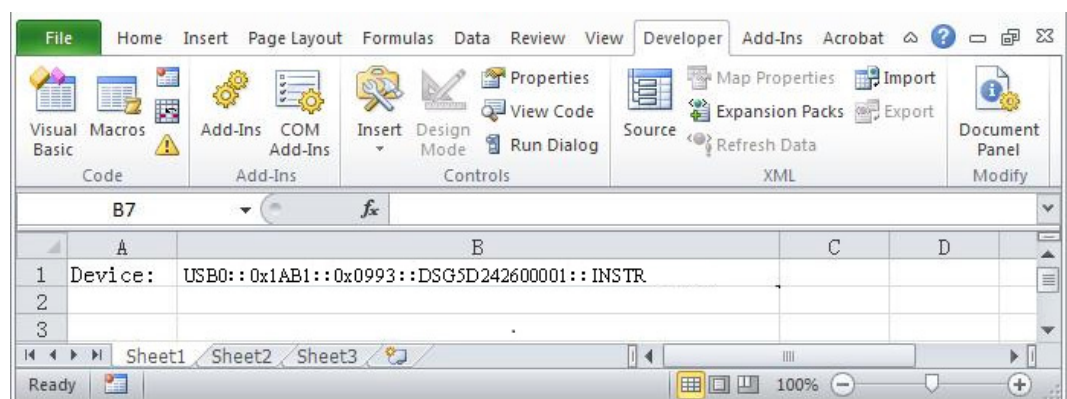
**Program used in this instance:** Microsoft Office Excel 2016

**Function realized in this example:** sending the "\*IDN" command and reading the instrument information.

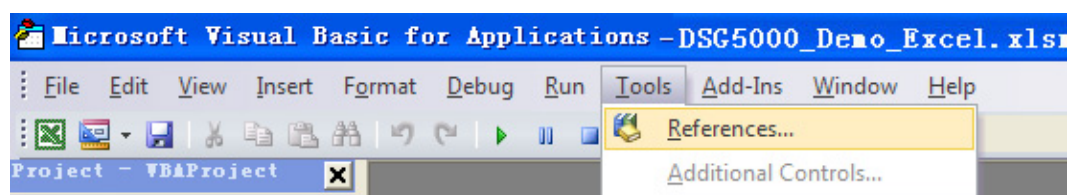
1. Create a new Excel file that enables the Macros and name it "DSG5000\_Demo\_Excel.xlsm".
2. Run DSG5000\_Demo\_Excel.xlsm. Click **File** > **Options** at the upper-left corner of the Excel file to open the interface as shown in the figure below. Click **Customize Ribbon** at the left, check **Developer** and click **OK**. At this point, the Excel menu bar displays the Developer menu.



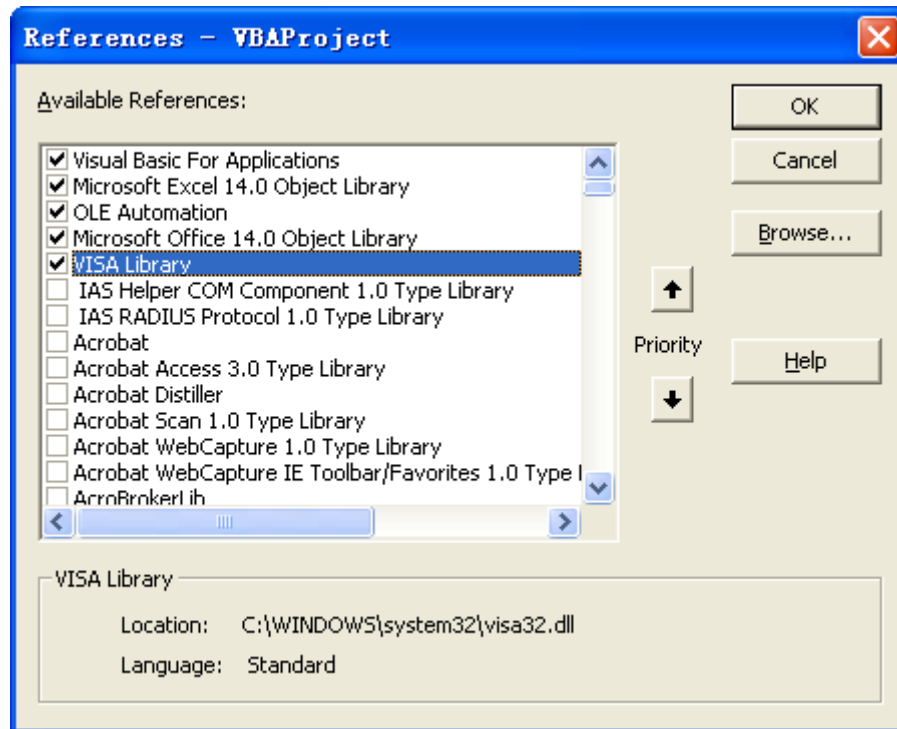
3. Input the VISA descriptor into a cell of the file, as shown in the figure below. Click the **Developer** menu and select the Visual Basic option to open the Microsoft Visual Basic.



4. Select "Tools(T)" in the Microsoft Visual Basic menu bar and click "References".



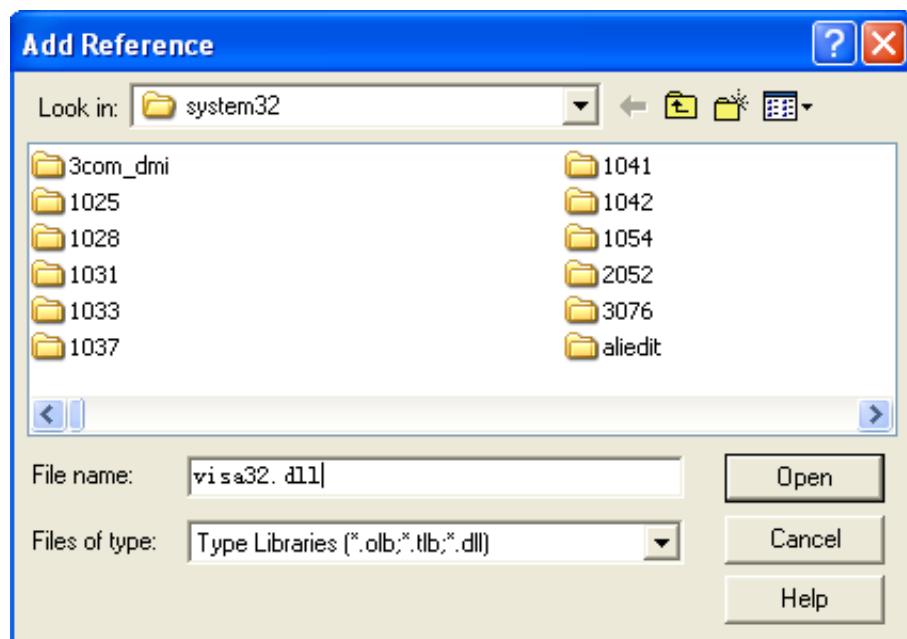
In the displayed dialog box, select "VISA Library", and click **OK** to refer to VISA Library.



#### TIP

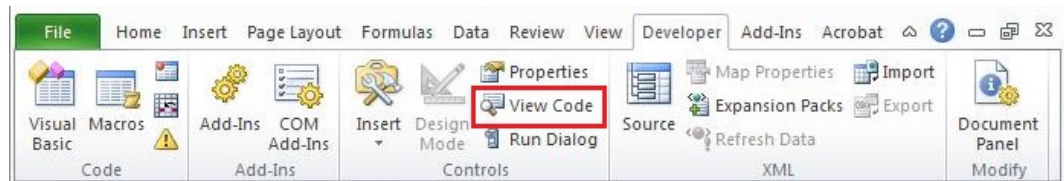
If you cannot find VISA Library in the left section of the above dialog box, please follow the method below to find it.

1. Make sure that your PC has installed the NI-VISA library.
2. Click Browse... at the right section to search visa32.dll from C:\WINDOWS\system32, as shown in the figure below.





5. Click **View Code** under Developer menu to enter the interface of Microsoft Visual Basic. Add the following codes and save it.



### NOTE

If the Excel file created at Step 2 does not enable the Macros, a prompt message "The following features cannot be saved in macro-free workbooks" will be displayed. In this case, please save the file as a macro-enabled file type (filename with a suffix of ".xlsm").

```
Sub QueryIdn ()

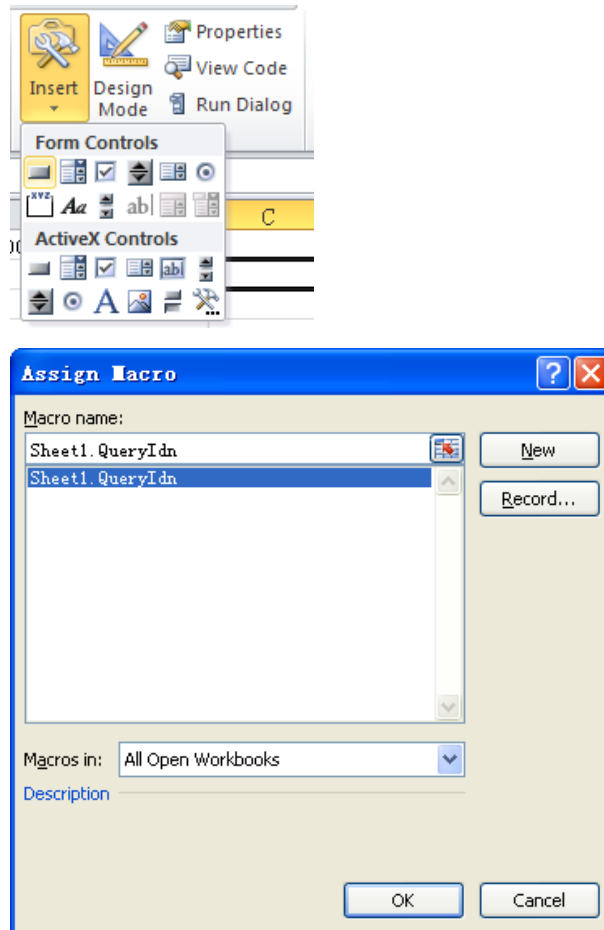
    Dim viDefRm As Long
    Dim viDevice As Long
    Dim viErr As Long
    Dim cmdStr As String
    Dim idnStr As String * 128
    Dim ret As Long
    'Turn on the device, and the device resource descriptor is in
    CELLS(1,2) of SHEET1'
    viErr = visa.viOpenDefaultRM(viDefRm)
    viErr = visa.viOpen(viDefRm, Sheet1.Cells(1,2), 0, 5000,
viDevice)

    'Send request, read the data, and the return value is in
    CELLS(2,2) of SHEET1'
    cmdStr = "*IDN?"
    viErr = visa.viWrite(viDevice, cmdStr, Len(cmdStr), ret)
    viErr = visa.viRead(viDevice, idnStr, 128, ret)
    Sheet1.Cells(2,2) = idnStr

    'Turn off the device'
    visa.viClose (viDevice)
    visa.viClose (viDefRm)

End Sub
```

6. Add the button control. Click **Insert** under the Developer menu, and select a button control under the Form Controls menu item and put it into the Excel cell. At this time, the Assign Macro dialog box is displayed, Select "Sheet1.QueryIdn" and click **OK**.



The default name of the button is "Button1". Right-click the button and select Edit Text in the pop-up menu to change the button name to "\*IDN?".

7. Click "\*IDN?" to run the program. The device information of the microwave signal generator is as shown in the figure below.

| A       | B  | C     |
|---------|--|-------|
| Device: | USB0::0x1AB1::0x0993::DSG5D242600001::INSTR        | *IDN? |
|         | RIGOL TECHNOLOGIES,DSG5208,DSG5D242600001,00.00.01 |       |

## 5.3 Matlab Programming Example

**Program used in this example:** MATLAB R2009a

**Functions realized in this demo:** read the current frequency and amplitude of the microwave signal generator.

1. Run the Matlab software and modify the current path (Current Directory). In this example, modify the current path to D:\DSG5000\_Demo.
2. Click **File** > **New** > **Blank M File** in the Matlab interface to create an empty M file.

### 3. Add the following codes to the M file:

```
DSG5000=visa('ni','USB0::0x1AB1::0x0993::DSG5D242600001::INSTR');
    %Creates the Visa object.

fopen( DSG5000 );    %Opens the visa object created.

fprintf(DSG5000,':FREQ?');    %Sends request to query the
frequency.

meas_RF_FREQ = fscanf(DSG5000);    %Reads the frequency data.

fprintf(DSG5000,':LEV?' );    %Sends request to query the
amplitude.

meas_RF_LEV = fscanf(DSG5000);    %Reads the amplitude data.

fclose(DSG5000);    %Closes the visa object.

display(meas_RF_FREQ)    %Displays the frequency read.

display(meas_RF_LEV)    %Display the amplitude read.
```

### 4. Save the M file to the current path. In this example, the M file is named as "DSG5000\_Demo\_MATLAB.m".

### 5. Run the M file and the running results are displayed as follows:

```
meas_RF_FREQ =
1.500 000 000 00

meas_RF_LEV =
-20.00
```

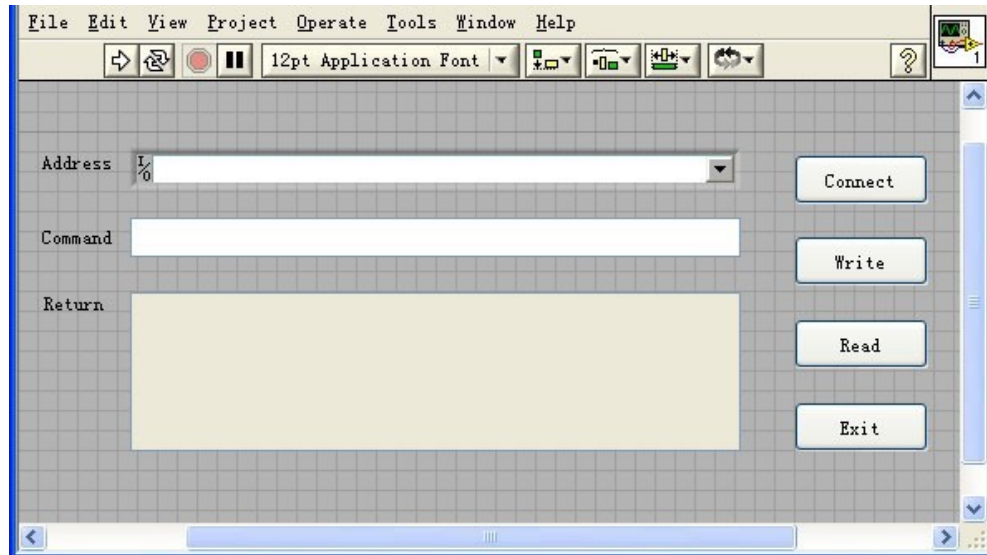
The results above denote that the current frequency of the microwave signal generator is 1.5 GHz and the amplitude is -20 dBm.

## 5.4 LabVIEW Programming Example

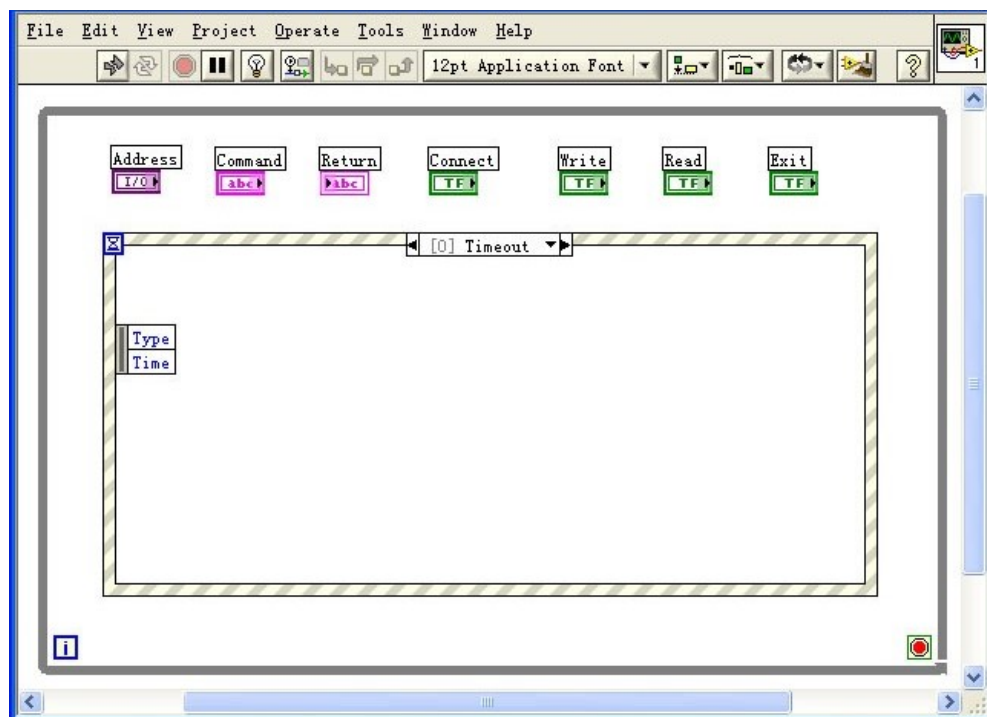
**Program used in this example:** LabVIEW 2009

**Function realized in this example:** search for the instrument address, connect to the instrument, send commands, and read return values.

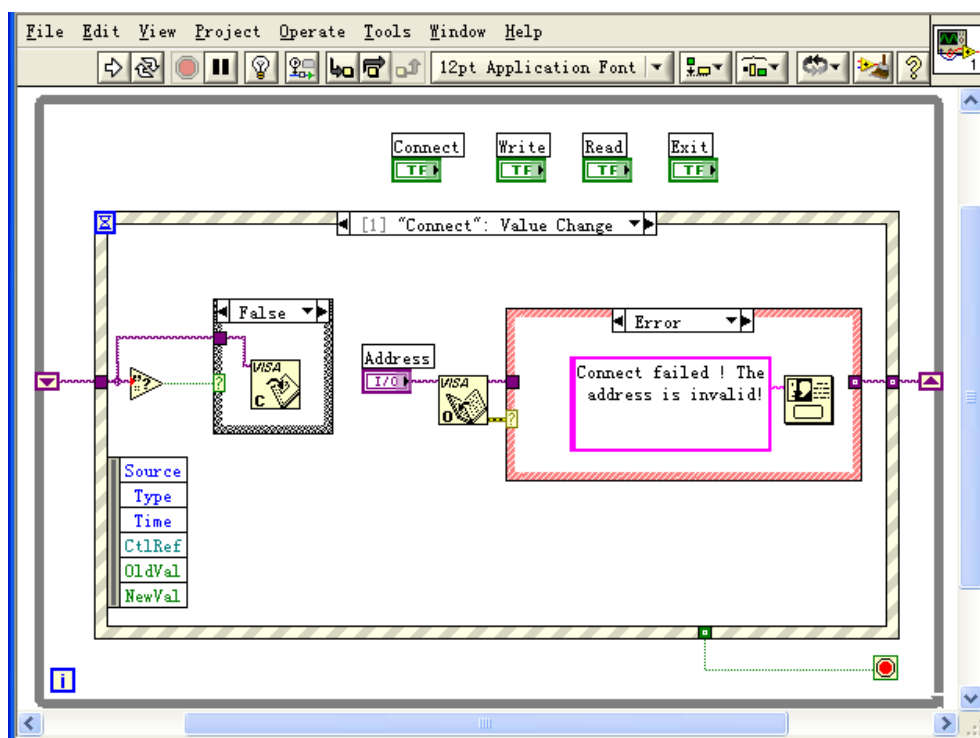
1. Run LabVIEW 2009, and then create a VI file named DSG5000\_Demo\_LABVIEW.
2. Add controls to the front panel interface, including the Address field, Command field, and Return field, the Connect button, the Write button, the Read button, and the Exit button.



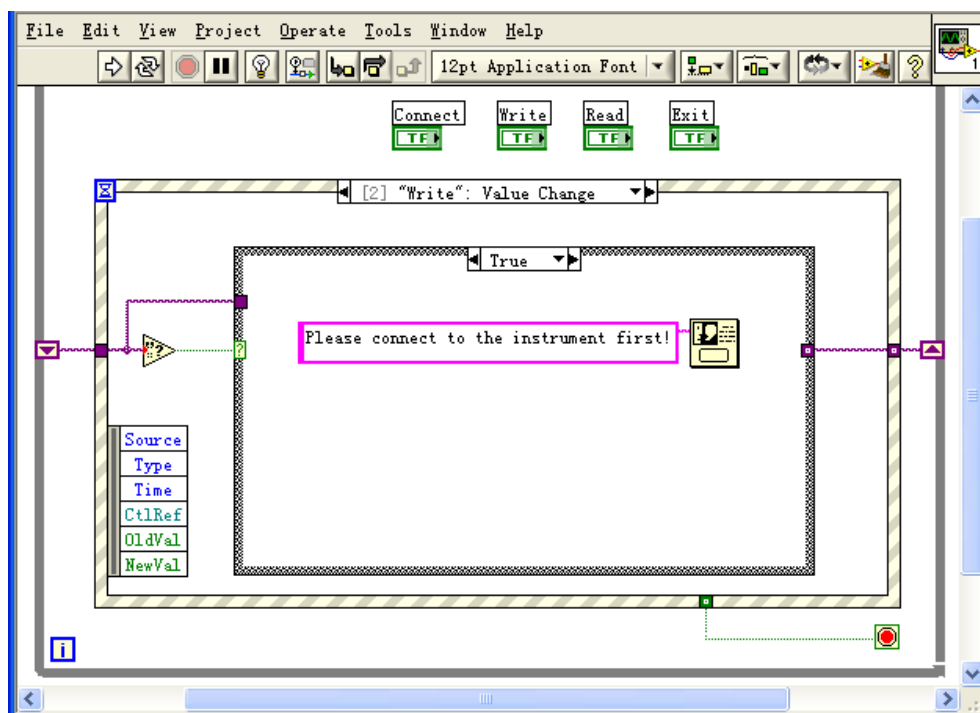
3. Click Show Block Diagram under the Window menu and add the While cycle to create an event structure.

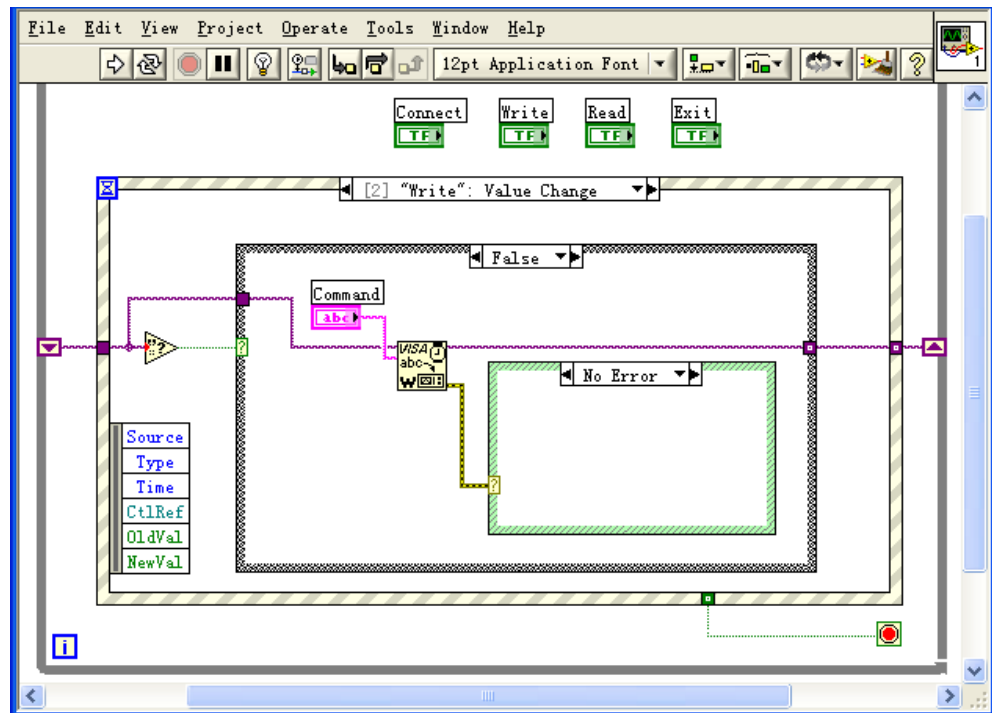


4. Add the events (including connecting to the instrument, write operation, read operation, and exit)
  - a. Connect (including error processing)

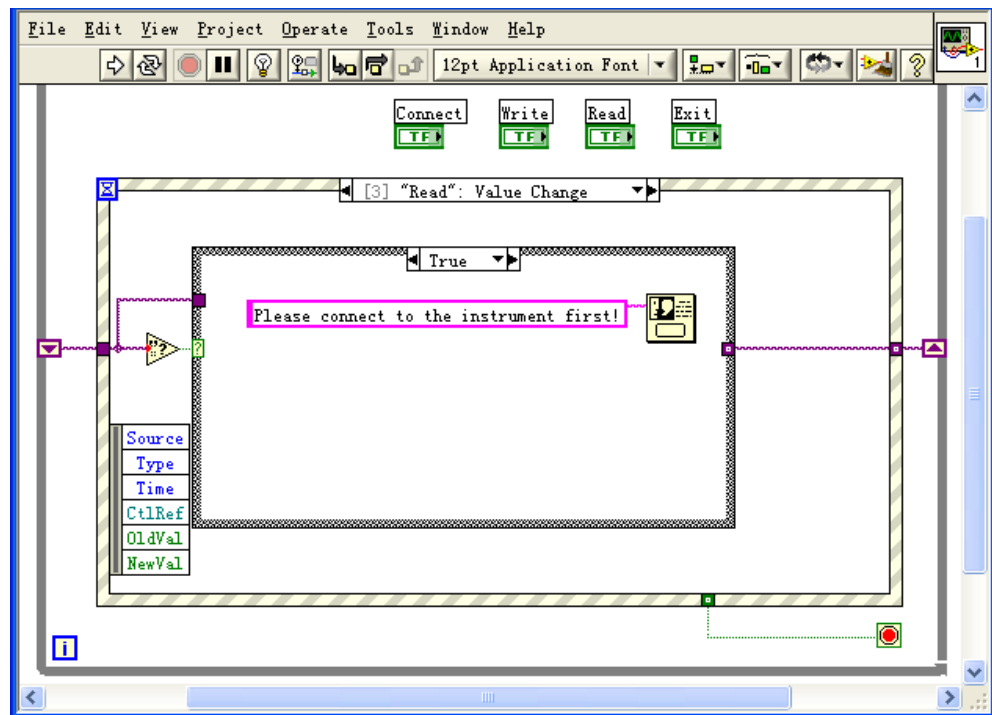


**b. Write operation (including error confirmation)**

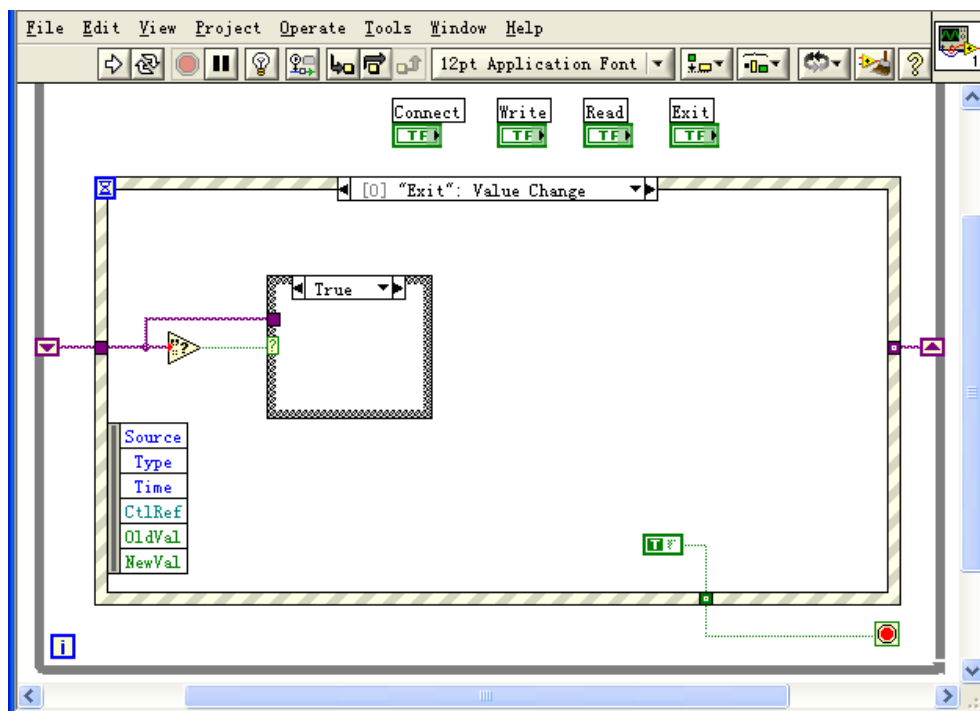




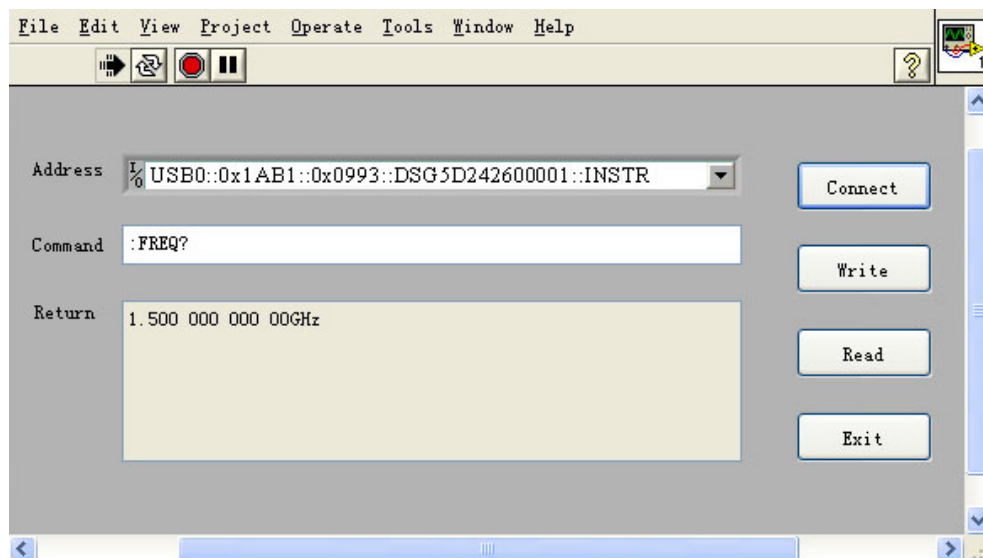
c. Read operation (including error correction advice)



d. Exit



5. Run the program, and then the following interface is displayed below. Click the VISA resource name from the drop-down list under Address, and click **Connect** to connect the instrument. Then, input \*IDN in the Command field. Click **Write** to write the command to the instrument. If the command is a query (e.g.:FREQ?), click **Write** to write the command into the instrument, and then click **Read**. The return value is displayed in the Return field. The return value 1.500 000 000 00GHz (denote that the current frequency of the RF signal is 1.5GHz) is displayed in the Return text box. Click **Exit** to exit the program.



## 5.5 Visual C++ Programming Example

**Program used in this example:** Microsoft Visual C++6.0

**Function realized in this example:** search for the instrument address, connect to the instrument, send commands, and read return values.

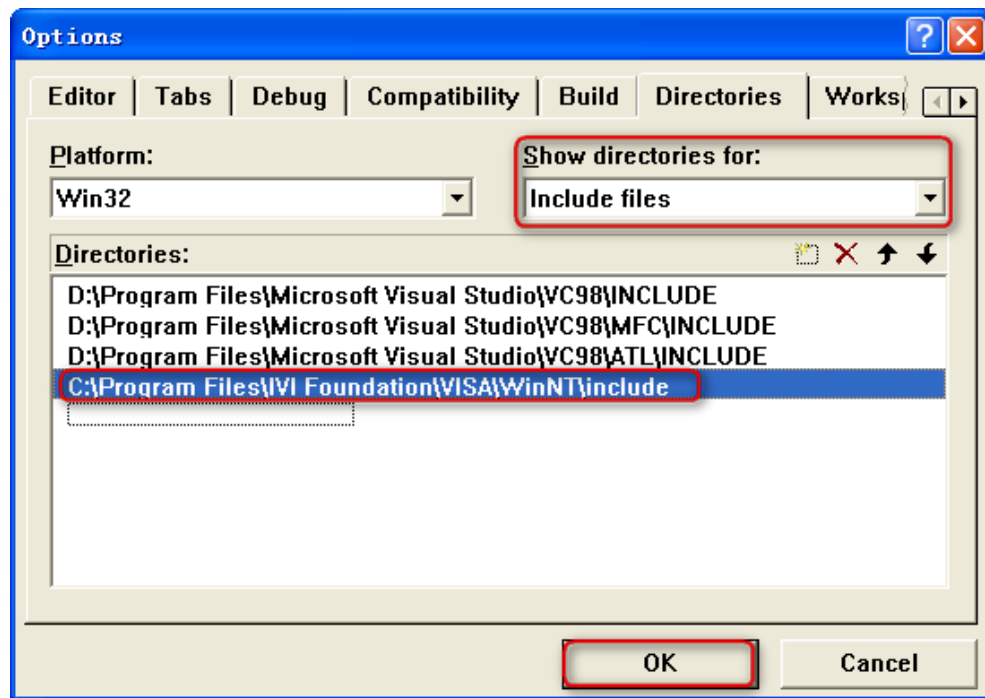
1. Run Microsoft Visual C++ 6.0. Create an MFC project based on a dialog box and name it "DSG5000\_DEMO\_VC".
2. Click **Project** > **Settings** and add visa32.lib in the "Link" tab in the pop-up interface manually.
3. Click **Tools** > **Options** and add the Include and Lib directories under the Directories tab in the pop-up interface.
  - Select Include files from the drop-down list under Show directories for. Double click the empty space under Directories to enter the specified path of Include files: C:\Program Files\IVI Foundation\VISA\WinNT\include. Click OK to close the dialog box.
  - Select Library files from the drop-down list under Show directories for. Double click the empty space under Directories to enter the specified path of Library files: C:\Program Files\IVI Foundation\VISA\WinNT\lib\msc. Click OK to close the dialog box.



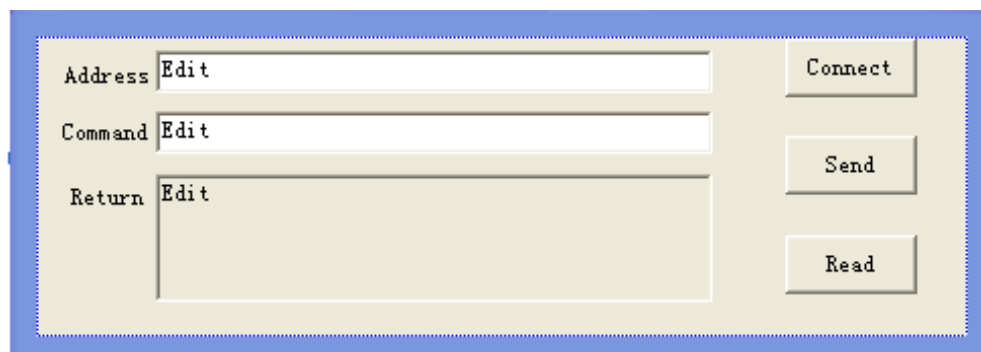
### NOTE

The two paths added here are related to the installation path of NI-VISA on your PC. By default, NI-VISA is installed under C:\Program Files\IVI Foundation\VISA.





4. Add the Text, Edit, and Button controls. The layout is as shown in the figure below.



5. Click **View > ClassWizard** and add the control variables in the "Member Variables" tab in the pop-up interface.

- Instrument address: CString m\_strInstrAddr
- Command: CString m\_strCommand
- Returned value: CString m\_strResult

6. Encapsulate the read and write operations of VISA.

a. Encapsulate the write operation of VISA for easier operation.

```
bool CDSG5000_DEMO_VCDlg::InstrWrite(CString strAddr, CString
strContent) //write function
{
    ViSession defaultRM, instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendBuf = NULL;
```

```

char * SendAddr = NULL;
bool bWriteOK = false;
CString str;

//Change the address's data style from CString to char*
SendAddr = strAddr.GetBuffer(strAddr.GetLength());
strcpy(SendAddr, strAddr);
strAddr.ReleaseBuffer();

//Change the command's data style from CString to char*
SendBuf = strContent.GetBuffer(strContent.GetLength());
strcpy(SendBuf, strContent);
strContent.ReleaseBuffer();

//open the VISA instrument
status = viOpenDefaultRM(&defaultRM);
if (status < VI_SUCCESS)
{
    AfxMessageBox("No VISA instrument was opened !");
    return false;
}

status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL,
&instr);
//write command to the instrument
status = viWrite(instr, (unsigned char *)SendBuf,
strlen(SendBuf), &retCount);

//close the instrument
status = viClose(instr);
status = viClose(defaultRM);

return bWriteOK;
}

```

**b. Encapsulate the read operation of VISA for easier operation.**

```

bool CDSG5000_DEMO_VCDlg::InstrRead(CString strAddr, CString
*pstrResult) //Read from the instrument
{
    ViSession defaultRM, instr;
    ViStatus status;
    ViUInt32 retCount;
    char * SendAddr = NULL;
    unsigned char RecBuf[MAX_REC_SIZE];
    bool bReadOK = false;
    CString str;

    //Change the address's data style from CString to char*
    SendAddr = strAddr.GetBuffer(strAddr.GetLength());
    strcpy(SendAddr, strAddr);
    strAddr.ReleaseBuffer();

    memset(RecBuf, 0, MAX_REC_SIZE);

    //open the VISA instrument
    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        //Error Initializing VISA...exiting
        AfxMessageBox("No VISA instrument was opened !");
        return false;
    }

    //open the instrument
    status = viOpen(defaultRM, SendAddr, VI_NULL, VI_NULL,

```

```

&instr);

    //read from the instrument
    status = viRead(instr, RecBuf, MAX_REC_SIZE, &retCount);

    //close the instrument
    status = viClose(instr);
    status = viClose(defaultRM);

    (*pstrResult).Format("%s", RecBuf);

    return bReadOK;
}

```

## 7. Add the control message response codes.

### a. Connect to the instrument

```

void CDSG5000_DEMO_VCDlg::OnConnect()
{
    //TODO: Add your control notification handler code here
    ViStatus status;
    ViSession defaultRM;
    ViString expr = "?*";
    ViPFindList findList = new unsigned long;
    ViPUInt32 retcnt = new unsigned long;
    ViChar instrDesc[1000];
    CString strSrc = "";
    CString strInstr = "";
    unsigned long i = 0;
    bool bFindDSG = false;

    status = viOpenDefaultRM(&defaultRM);
    if (status < VI_SUCCESS)
    {
        //Error Initializing VISA...exiting
        MessageBox("No VISA instrument was opened ! ");
        return ;
    }

    memset(instrDesc,0,1000);

    //Find resource
    status = viFindRsrc(defaultRM,expr,findList, retcnt,
instrDesc);

    for (i = 0;i < (*retcnt);i++)
    {
        //Get instrument name
        strSrc.Format("%s",instrDesc);
        InstrWrite(strSrc,"*IDN?");
        ::Sleep(200);
        InstrRead(strSrc,&strInstr);

        //If the instrument(resource) belongs to the DSG
series then jump out from the loop
        trInstr.MakeUpper();
        if (strInstr.Find("DSG") >= 0)
        {
            bFindDSG = true;
            m_strInstrAddr = strSrc;
            break;
        }

        //Find next instrument
    }
}

```

```

        status = viFindNext(*findList,instrDesc);
    }

    if (bFindDSG == false)
    {
        MessageBox("Didn't find any DSG!");
    }
    UpdateData(false);
}

```

### b. Write Operation

```

void CDSG5000_DEMO_VCDlg::OnSend()
{
    //TODO: Add your control notification handler code here
    UpdateData(true);
    if (m_strInstrAddr.IsEmpty())
    {
        MessageBox("Please connect to the instrument first!");
    }
    InstrWrite(m_strInstrAddr,m_strCommand);
    m_strResult.Empty();
    UpdateData(false);
}

```

### c. Read Operation

```

void CDSG5000_DEMO_VCDlg::OnRead()
{
    //TODO: Add your control notification handler code here
    UpdateData(true);
    InstrRead(m_strInstrAddr,&m_strResult);
    UpdateData(false);
}

```

## 8. Run the results.

- a. Click **Connect** to find and connect the microwave signal generator. If the instrument is successfully connected, the corresponding USB VISA descriptor will be displayed in the address bar.
- b. Input command in the "Command" edit box; for example, :SWE:DIR?.
- c. Click **Send** to send the command.
- d. Click **Read** to read the return value.

The execution result is as shown in the figure below.

The screenshot shows the application window with the following fields and buttons:

- Address:** USB0::0x1AB1::0x0993::DSG SD242600001::INSTR
- Command:** :SWE:DIR?
- Return:** FWD
- Buttons:** Connect, Send, Read

## 6 Appendix

### 6.1 Appendix A: Factory Setting

| Parameter                         | Factory Setting |
|-----------------------------------|-----------------|
| <b>FREQ</b>                       |                 |
| Frequency                         | 20 GHz          |
| Frequency Offset                  | 0 Hz            |
| Phase Offset                      | 0 deg           |
| <b>LEVEL</b>                      |                 |
| Level                             | -30 dBm         |
| Level Limit                       | 25 dBm          |
| Level Offset                      | 0 dBm           |
| Level Unit                        | dBm             |
| <b>SWEEP</b>                      |                 |
| Sweep Function                    | Off             |
| Sweep Type                        | Step            |
| Sweep Mode                        | Continuous      |
| Start Frequency                   | 1 GHz           |
| Stop Frequency                    | 2 GHz           |
| Start Level                       | -10 dBm         |
| Stop Level                        | -20 dBm         |
| Sweep Points                      | 11              |
| Dwell Time                        | 500 ms          |
| Sweep Space                       | Lin             |
| Sweep Shape                       | Triangle        |
| Trigger Type for the Sweep Period | Auto Trigger    |
| Point Trigger Type                | Auto Trigger    |
| Slope of the External Trigger     | Pos             |
| Sweep Direction                   | Fwd             |
| <b>AM</b>                         |                 |
| Switch                            | OFF             |
| Modulation Source                 | Internal        |
| Modulation Depth                  | 50%             |
| Modulation Frequency              | 10 kHz          |
| Modulating Waveform               | Sine            |
| Equal Level Output                | OFF             |

| Parameter                     | Factory Setting |
|-------------------------------|-----------------|
| <b>FM</b>                     |                 |
| Switch                        | OFF             |
| Modulation Source             | Internal        |
| Frequency Deviation           | 10 kHz          |
| Modulation Frequency          | 10 kHz          |
| Modulating Waveform           | Sine            |
| <b>PM</b>                     |                 |
| Switch                        | OFF             |
| Modulation Source             | Internal        |
| Phase Deviation               | 5 rad           |
| Modulation Frequency          | 10 kHz          |
| Modulating Waveform           | Sine            |
| <b>Pulse</b>                  |                 |
| Switch                        | OFF             |
| Modulation Source             | Internal        |
| Pulse Type                    | Single          |
| Pulse Period                  | 1 ms            |
| Pulse Width                   | 500 us          |
| Trigger Type                  | Auto            |
| Pulse Output                  | OFF             |
| Trig Delay                    | 200 ns          |
| Pulse Polarity                | Normal          |
| Gated Polarity                | Normal          |
| Slope of the External Trigger | Pos             |
| <b>Output Control</b>         |                 |
| RF Switch                     | OFF             |
| MOD Switch                    | OFF             |
| <b>System</b>                 |                 |
| Language                      | English         |
| Preset Type                   | Factory         |
| DHCP                          | ON              |
| Auto IP                       | ON              |
| Manual IP                     | OFF             |
| Screen Lock                   | OFF             |
| Power Status                  | Default         |

## 6.2 Appendix B: Warranty

---

RIGOL TECHNOLOGIES CO., LTD. (hereinafter referred to as RIGOL) warrants that the product mainframe and product accessories will be free from defects in materials and workmanship within the warranty period.

If a product proves defective within the respective period, RIGOL guarantees free replacement or repair of any defective products within a reasonable period of time. To get repair service, please contact with your nearest RIGOL sales or service office.

There is no other warranty, expressed or implied, except such as is expressly set forth herein or other applicable warranty card. There is no implied warranty of merchantability or fitness for a particular purpose. Under no circumstances shall RIGOL be liable for any consequential, indirect, ensuing, or special damages for any breach of warranty in any case.

#### HEADQUARTER

**RIGOL** TECHNOLOGIES CO., LTD.  
No.8 Keling Road, New District, Suzhou,  
JiangSu, P.R.China  
Tel: +86-400620002  
Email: info@rigol.com

#### EUROPE

**RIGOL**TECHNOLOGIES EU GmbH  
Carl-Benz-Str.11  
82205 Gilching  
Germany  
Tel: +49(0)8105-27292-0  
Email: info-europe@rigol.com

#### NORTH AMERICA

**RIGOL**TECHNOLOGIES, USA INC.  
10220 SW Nimbus Ave.  
Suite K-7  
Portland, OR 97223  
Tel: +1-877-4-**RIGOL**-1  
Fax: +1-877-4-**RIGOL**-1  
Email: info@rigol.com

#### JAPAN

**RIGOL**TECHNOLOGIES JAPAN, LLC  
501, LATORRETTA, 2-37-1,  
Numabukuro,  
Nakano-Ku, Tokyo, Japan  
Tel: +81-3-6262-8932  
Fax: +81-3-6262-8933  
Email: info-japan@rigol.com

---

**RIGOL**® is the trademark of **RIGOL**TECHNOLOGIES CO., LTD. Product information in this document subject to update without notice. For the latest information about **RIGOL**'s products, applications and services, please contact local **RIGOL** channel partners or access **RIGOL** official website: [www.rigol.com](http://www.rigol.com)