RIGOL

DG800 Pro/DG900 Pro Series

Function/Arbitrary Waveform Generator

Programming Guide

Dec. 2023

Guaranty and Declaration

Copyright

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

Website: *http://www.rigol.com*

Section	Description	Page

List of Tables

1 Document Overview

This manual introduces how to program and control DG800 Pro and DG900 Pro series function/arbitrary waveform generator via the remote interface by using the SCPI (Standard Commands for Programmable Instruments) commands. DG800 Pro and DG900 Pro series can communicate with PC through USB or LAN interface.

TIP

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

Publication Number

PGB16100-1110

Software Version

00.01.00

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. **Key**

   The front panel key is denoted by the menu key icon. For example, Default indicates the "Default" key.

2. **Menu**

   The menu item is denoted by the format of "Menu Name (Bold) + Character Shading" in the manual. For example, **Setup**.

3. **Operation Procedures**

   The next step of the operation is denoted by ">" in the manual. For example, > **Utility** indicates first clicking or tapping and then clicking or tapping **Utility**.

Content Conventions in this Manual

DG800 Pro series function/arbitrary waveform generator includes the following models. Unless otherwise specified, this manual takes the two-channel model DG852 Pro as an example to illustrate the functions and operation methods of DG800 Pro series.

Model	No. of Channels	Sample Rate	Max. Output Frequency
DG821 Pro	1	625 MSa/s	25 MHz

Model	No. of Channels	Sample Rate	Max. Output Frequency
DG822 Pro	2	625 MSa/s	25 MHz
DG852 Pro	2	625 MSa/s	50 MHz

DG900 Pro series function/arbitrary waveform generator includes the following models. Unless otherwise specified, this manual takes DG922 Pro as an example to illustrate the functions and operation methods of DG900 Pro series.

Model	No. of Channels	Sample Rate	Max. Output Frequency
DG902 Pro	2	1.25 GSa/s	70 MHz
DG912 Pro	2	1.25 GSa/s	150 MHz
DG922 Pro	2	1.25 GSa/s	200 MHz

2 Programming Overview

2.1 SCPI Introduction

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 question 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,

:COUNter:TYPE <*type*>

:COUNter:TYPE?

COUNter is the root keyword of the command, **TYPE** is the second-level keyword. The command line starts with a colon, and different levels of keywords are also separated by colons. <*type*> indicates a settable parameter. The command ending with a quotation mark indicates querying a function. The command keywords **:COUNter:TYPE** and the parameter <*type*> are separated by a space.

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

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

Symbol Description

The following symbols are not sent with the commands.

1. **Braces { }**

   The contents in the braces can contain one or multiple parameters. These parameters can be omitted or used for several times. Parameters are usually separated by the vertical bar "|". When using the command, you must select one of the parameters.

2. **Vertical Bar |**

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

3. **Square Brackets [ ]**

   The contents in the square brackets can be omitted.

4. **Angle Brackets < >**

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

Parameter Type

1. **Bool**

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

   :SYSTem:BEEPer:STATe <*bool*>

   :SYSTem:BEEPer:STATe?

   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:PSTatus <*sat*>

   :SYSTem:PSTatus?

   Wherein,

   - <*sat*> can be set to DEFault|OPEN.
   - The query returns an abbreviated form: DEF or OPEN.

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,

[**:SOURce**[<*n*>]]**:SWEep:STEP** {<*step*>|<*lim*>}

Wherein, <*step*> can be set to an integer ranging from 2 to 1024. The query returns an integer ranging from 2 to 1024.

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[<*n*>]]:AM:INTernal:FREQuency {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:AM:INTernal:FREQuency? [<*lim_query*>]

Wherein, <*frequency*> can be set to any real number ranging from 2E-3 (2 mHz) to 1E+6 (1 MHz). The query returns a real number in scientific notation.

5. ASCII String

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

:SYSTem:COMMunicate:LAN:GATEway <*string*>

Wherein, <*string*> can be set to 192.168.1.1.

Use the MINimum, MAXimum, and DEFault parameters

For many commands, you can substitute "MINimum" or "MAXimum" in place of a parameter. In some cases, you can also substitute "DEFault". For example,

[:SOURce[<*n*>]]:FREQuency:STARt {<*frequency*>|<*lim_set*>}

Wherein, <*lim_set*> can be set to MINimum|MAXimum|DEFault.

Instead of selecting a specific value for <*frequency*>, you can substitute MINimum to set the frequency to its minimum value and MAXimum to set the frequency to its maximum value. You can also use DEFault to set the frequency to its default value.

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:LANGuage?

can be abbreviated as

:SYST:LANG?

2.2 Remote Control

This instrument can be connected to the PC via the USB and LAN interfaces to set up communication and realize remote control through the PC. The remote control can be realized by using SCPI (Standard Commands for Programmable Instruments) commands.

PC Software

Users usually need to use the PC software to send commands to control the instrument remotely. When the instrument is connected to the PC via the USB or LAN interface, the PC software can search for instrument resources and enable command interaction.

Web Control

When the instrument is connected to the PC via the LAN interface, you can use Web Control to send SCPI commands from the PC to the instrument. Operation Procedures:

1. Obtain the instrument's IP address and input it in the browser address bar to log in to the Web Control page.
2. After you enter the Web Control interface, click the "SCPI Panel Control" button to enter the SCPI Command interface.
3. Input the specified SCPI command and then click **Send & Read** to send the command. The operation process and the returned value will be displayed in the current interface.

3 Command System

This chapter introduces the syntax, functions, parameters, and usage of each DG800 Pro/DG900 Pro command.

For the parameter setting command (frequency, amplitude, offset, high level, low level, time), the instrument can recognize the unit. If omitted, it is interpreted as the default unit. For the default units of parameters, see the table below.

Table 3.1 Unit Description

Parameter	Available Units	Default Unit
Frequency	MHZ, KHZ, HZ, UHZ	HZ
Amplitude	VPP, MVPP, VRMS, MVRMS, DBM	VPP, VRMS, or DBM (related to the current amplitude unit)
Offset/High Level/Low Level	V, MV	V
Time	KS, S, MS, US, NS	S

NOTE

Since the SCPI parser is case-insensitive, there is some confusion over the letter "M" (or "m"). For your convenience, this instrument interprets "mV" (or "MV") as millivolts and "ms" (or "Ms") as milliseconds. However, it interprets "MHZ" (or "mhz") as megahertz.

3.1 :ABORt

Syntax

`:ABORt`

Description

Stops any operation that is triggered.

Parameter

None.

Remarks

None.

Return Format

None.

Examples

```
:ABORt /*Stops any operation that is triggered.*/
```

3.2 :COUNter Commands

:COUNter commands are used to enable/disable the counter function, and set the counter.

3.2.1 :COUNter:AVERage:ALL?

Syntax

:COUNter:AVERage:ALL?

Description

Queries the statistical results for counter measurements.

Parameter

None.

Remarks

The counter can provide the statistical function for the current measurement parameters. Use *:DISPlay:COUNter* to set and query the current measurement parameters.

Return Format

The query returns a string which consists of 6 parts (representing the current value, maximum value, minimum value, average value, standard deviation, and statistical count respectively) separated by commas. For example, the query might return +1.000004E+06,+1.000005E+06,+1.000003E+06,+1.000004E+06,+0.000000E+00,35, indicating that the current value is 1.000,004 MHz, the maximum value is 1.000,005 MHz, the minimum value is 1.000,003 MHz, the average value is 1.000,004 MHz, the standard deviation is 0 Hz, and the count is 35.

When the counter statistical function is disabled, send this command and the query returns 0,0,0,0,0,0.

Examples

```
:COUNter:AVERage:ALL? /*Queries the statistical results for counter
measurements. The query might return +1.000004E+06,+1.000005E
+06,+1.000003E+06,+1.000004E+06,+0.000000E+00,35.*/
```

3.2.2 :COUNter:AVERage:CLEar

Syntax

:COUNter:AVERage:CLEar

Description

Clears the statistical results for counter measurements.

Parameter

None.

Remarks

- This command is valid only when the statistical function (*:COUNter:AVERage[:STATe]*) is enabled.
- When the statistical function is disabled, the statistical results are automatically cleared.

Return Format

None.

Examples

```
:COUNter:AVERage:CLEar /*Clears the statistical results for counter measurements.*/
```

3.2.3 :COUNter:AVERage[:STATe]

Syntax

:COUNter:AVERage[:STATe] <*state*>

:COUNter:AVERage[:STATe]?

Description

Sets or queries the on/off status of the statistical function for the frequency counter.

Parameter

Name	Type	Range	Default
<state>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

None.

Return Format

The query returns 0 or 1.

Examples

```
:COUNter:AVERage:STATe ON /*Enables the statistical function for
the frequency counter.*/
:COUNter:AVERage:STATe? /*Queries the on/off status of the
statistical function for the frequency counter. The query returns
1.*/
```

3.2.4 :COUNter:COUPling

Syntax

:COUNter:COUPling <*coupling*>

:COUNter:COUPling?

Description

Sets or queries the coupling type of input signals for counter measurements.

Parameter

Name	Type	Range	Default
<coupling>	Discrete	{AC|DC}	AC

Remarks

- **AC:** AC coupling. It is available only when the input impedance (*:COUNter:IMPedance*) is set to 1 MΩ.
- **DC:** DC coupling.

Return Format

The query returns AC or DC.

Examples

```
:COUNter:COUPling DC /*Sets the coupling type to DC for counter
measurements.*/
:COUNter:COUPling? /*Queries the coupling type for counter
measurements. The query returns DC.*/
```

3.2.5 :COUNter[:FREQuency]:RANGe

Syntax

:COUNter[:FREQuency]:RANGe <*range*>

:COUNter[:FREQuency]:RANGe?

Description

Sets or queries the frequency range for counter measurements.

Parameter

Name	Type	Range	Default
<range>	Discrete	{250|500|1000}	250

Remarks

The following shows the frequency range represented by <range>.

- **250:** 0 to 250 MHz
- **500:** 250 MHz to 500 MHz
- **1000:** 500 MHz to 1 GHz (for DG900 Pro)

TIP

When the counter input impedance is set to 1 MΩ (*:COUNter:IMPedance*), the frequency range is fixed to 0 Hz to 250 MHz.

Return Format

The query returns 250, 500, or 1000.

Examples

```
:COUNter:FREQuency:RANGe 500 /*Sets the frequency range to 250 MHz
to 500 MHz for counter measurements.*/
:COUNter:FREQuency:RANGe? /*Queries the frequency range for counter
measurements. The query returns 500.*/
```

3.2.6 :COUNter:GATetime

Syntax

:COUNter:GATetime {<*time*>|<*lim*>}

:COUNter:GATetime?

Description

Sets or queries the gate time for counter measurements.

Parameter

Name	Type	Range	Default
<time>	Real	1 ms to 10,000 s	1 ms
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

When this command is used to specify the gate time, the auto gate function (*:COUNter:GATetime:AUTO[:STATe]*) is automatically disabled.

Return Format

The query returns the gate time in scientific notation. For example, the query might return +1.00000000000000E-02.

Examples

```
:COUNter:GATetime 0.01 /*Sets the gate time to 10 ms.*/
:COUNter:GATetime? /*Queries the gate time. The query returns
+1.00000000000000E-02.*/
```

3.2.7 :COUNter:GATetime:AUTO[:STATe]

Syntax

:COUNter:GATetime:AUTO[:STATe] <*bool*>

:COUNter:GATetime:AUTO[:STATe]?

Description

Sets or queries the on/off status of the auto gate function for counter measurements.

Parameter

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

Remarks

When it is enabled, the instrument will select an appropriate gate time automatically according to the characteristics of the signal to be measured.

Return Format

The query returns 0 or 1.

Examples

```
:COUNter:GATetime:AUTO:STATe ON /*Sets the gate time to auto for
counter measurements.*/
:COUNter:GATetime:AUTO:STATe? /*Queries whether the auto gate
function is enabled. The query returns 1.*/
```

3.2.8 :COUNter:HF

Syntax

:COUNter:HF <*bool*>

:COUNter:HF?

Description

Sets or queries the on/off status of the HF Suppression function for counter measurements.

Parameter

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

Remarks

- In low-frequency signal measurements, you can enable this function to filter the high-frequency noises in a way to improve the measurement accuracy.
- When the counter input impedance is set to 50 Ω (*:COUNter:IMPedance*), the HF Suppression function is not supported.

Return Format

The query returns 0 or 1.

Examples

```
:COUNter:HF ON /*Enables the HF Suppression function for counter
measurements.*/
:COUNter:HF? /*Queries the on/off status of the HF Suppression
function for counter measurements. The query returns 1.*/
```

3.2.9 :COUNter:IMPedance

Syntax

`:COUNter:IMPedance` <*impedance*>

`:COUNter:IMPedance?`

Description

Sets or queries the input impedance for the frequency counter.

Parameter

Name	Type	Range	Default
<impedance>	Discrete	{OMEG|FIFTy}	FIFTy

Remarks

- **FIFTy:** sets the input impedance to 50 Ω.
- **OMEG:** sets the input impedance to 1 MΩ.

Return Format

The query returns FIFT or OMEG.

Examples

```
:COUNter:IMPedance OMEG /*Sets the input impedance to 1 MΩ for the
frequency counter.*/
:COUNter:IMPedance? /*Queries the input impedance for the frequency
counter. The query returns OMEG.*/
```

3.2.10 :COUNter:MEASure?

Syntax

:COUNter:MEASure?

Description

Queries the measurement results for the frequency counter.

Parameter

None.

Remarks

When the counter is in the "Run" or "Single" state, you can send this command to query the measurement value; when the counter is in "Stop" state, you can send this command to query the measurement value of the last measurement executed.

Return Format

The query returns a string which consists of 5 parts (representing the frequency, period, duty cycle, positive pulse width, and negative pulse width respectively) separated by commas. Each part is expressed in scientific notation. For example, the query might return +1.000000000000000E +03,+1.000000000000001E-03,+5.760000000000000E +01,+5.760000000000000E-04,+4.240000000000000E-04, indicating that the measurement results are 1 kHz frequency, 1 ms period, 57.6% duty cycle, 576 μs positive pulse width, and 424 μs negative pulse width.

When the counter function is disabled, send this command and the query returns 0,0,0,0,0.

Examples

```
:COUNter:MEASure? /*Queries the measurement results for the
frequency counter. The query might return +1.000000000000000E
+03,+1.000000000000001E-03,+5.760000000000000E
+01,+5.760000000000000E-04,+4.240000000000000E-04.*/
```

3.2.11 :COUNter:RUN:STATe

Syntax

:COUNter:RUN:STATe <*state*>

:COUNter:RUN:STATe?

Description

Sets or queries the operating status of the frequency counter.

Parameter

Name	Type	Range	Default
<state>	Discrete	{RUN|STOP}	RUN

Remarks

- **RUN:** sets the counter operating status to "Run". In this mode, the instrument measures the input signal continuously according to the current configuration. When the counter is enabled, the default setting is "Run".
- **STOP:** sets the counter operating status to "Stop". In this mode, the instrument stops the measurements after finishing the current measurement.

Return Format

The query returns RUN or STOP.

Examples

```
:COUNter:RUN:STATe RUN /*Sets the operating status of the frequency
counter to Run.*/
:COUNter:RUN:STATe? /*Queries the operating status of the frequency
counter. The query returns RUN.*/
```

3.2.12 :COUNter:SINGle

Syntax

`:COUNter:SINGle`

Description

Sets single measurement for the frequency counter.

Parameter

None.

Remarks

Send this command and then the instrument enters the "single" measurement state. That is, the instrument executes a single measurement and then enters the "Stop" state. This command is available only when the frequency counter (*:COUNter[:STATe]*) is enabled.

Return Format

None.

Examples

```
:COUNter:SINGle /*Executes a single measurement.*/
```

3.2.13 :COUNter[:STATe]

Syntax

:COUNter[:STATe] <*bool*>

:COUNter[:STATe]?

Description

Sets or queries the on/off status of the frequency counter.

Parameter

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

Remarks

- **0/OFF:** disables the frequency counter.
- **1/ON:** enables the frequency counter.

Return Format

The query returns 0 or 1.

Examples

```
:COUNter:STATe ON /*Enables the frequency counter.*/
:COUNter:STATe? /*Queries the on/off status of the frequency
counter. The query returns 1.*/
```

3.3 :DISPlay Commands

:DISPlay commands are used to set or query the status of the current channel and display, and select the method to specify the voltage range, frequency sweep range, and pulse duration.

3.3.1 :DISPlay:BRIGhtness

Syntax

:DISPlay:BRIGhtness {<*brightness*>|<*lim*>}

:DISPlay:BRIGhtness? [<*lim*>]

Description

Sets or queries the brightness of the display.

Parameter

Name	Type	Range	Default
<brightness>	Integer	1% to 100%	50%
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

None.

Return Format

The query returns an integer. For example, the query might return 80, indicating that the brightness is 80%.

Examples

```
:DISPlay:BRIGhtness 80 /*Sets the brightness of the display to
80%.*/
:DISPlay:BRIGhtness? /*Queries the brightness of the display. The
query returns 80.*/
```

3.3.2 :DISPlay:COUNter

Syntax

:DISPlay:COUNter <*type*>

:DISPlay:COUNter?

Description

Sets or queries the currently displayed measurement parameter for counter measurements.

Parameter

Name	Type	Range	Default
<type>	Discrete	{FREQuency|PERiod|DUTY| PWIDth|NWIDth}	FREQuency

Remarks

Available parameters include frequency (FREQuency), period (PERiod), duty cycle (DUTY), positive pulse width (PWIDth), and negative pulse width (NWIDth).

Return Format

The query returns FREQ, PER, DUTY, PWID, or NWID.

Examples

```
:DISPlay:COUNter PERiod /*Sets the currently displayed measurement
parameter to period for counter measurements.*/
:DISPlay:COUNter? /*Queries the currently displayed measurement
parameter for counter measurements. The query returns PER.*/
```

3.3.3 :DISPlay:FOCus

Syntax

:DISPlay:FOCus <*chan*>

:DISPlay:FOCus?

Description

Sets or queries the current channel.

Parameter

Name	Type	Range	Default
<chan>	Discrete	{CH1|CH2}	CH1

Remarks

As DG821 Pro has one channel, it only supports CH1 by default (unless the two-channel upgrade option is installed).

Return Format

The query returns CH1 or CH2.

Examples

```
:DISPlay:FOCus CH1 /*Sets the current channel to CH1.*/
:DISPlay:FOCus? /*Queries the current channel. The query returns
CH1.*/
```

3.3.4 :DISPlay[:STATe]

Syntax

:DISPlay[:STATe] <*bool*>

:DISPlay[:STATe]?

Description

Enables or disables the front-panel screen. Queries the current state of the front-panel screen.

Parameter

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

Remarks

- When it is disabled, the front-panel screen is black but the UI progress will not be blocked.
- When you press the front-panel Utility to return to local operation, the screen is automatically enabled.

Return Format

The query returns 1 or 0.

Examples

```
:DISPlay:STATe OFF /*Disables the front-panel screen.*/
:DISPlay:STATe? /*Queries the state of the front-panel screen. The
query returns 0.*/
```

3.3.5 :DISPlay:TEXT

Syntax

:DISPlay:TEXT <*string*>

:DISPlay:TEXT?

Description

Sets or queries the text message displayed on the front-panel screen.

Parameter

Name	Type	Range	Default
<string>	ASCII string	A string with a maximum of 40 characters enclosed by double quotation marks.	-

Remarks

When a text message is displayed on the front-panel screen, the instrument will enter the notifier mode. You can only press the front-panel Utility key to return to local mode or clear the text message (*:DISPlay:TEXT:CLEar*) to restore the screen to its normal display.

Return Format

The query returns a character string, for example, "Test in progress".

Examples

```
:DISPlay:TEXT "Test in progress" /*Sets the text message to "Test
in progress".*/
:DISPlay:TEXT? /*Queries the text message displayed on the front-
panel screen. The query returns "Test in progress".*/
```

3.3.6 :DISPlay:TEXT:CLEar

Syntax

`:DISPlay:TEXT:CLEar`

Description

Clears the text message displayed on the front-panel screen.

Parameter

None.

Remarks

None.

Return Format

None.

Examples

```
:DISPlay:TEXT:CLEar /*Clears the text message displayed on the
front-panel screen.*/
```

3.3.7 :DISPlay:UNIT:PULSe

Syntax

`:DISPlay:UNIT:PULSe` <*type*>

`:DISPlay:UNIT:PULSe?`

Description

Sets or queries the method to specify the pulse width.

Parameter

Name	Type	Range	Default
<type>	Discrete	{WIDTh|DUTY}	WIDTh

Remarks

- **WIDTh:** pulse width, to specify the pulse width in seconds (s).
- **DUTY:** duty cycle, to specify the pulse width in percentage (%).

Return Format

The query returns WIDT or DUTY

Examples

```
:DISPlay:UNIT:PULSe DUTY /*Sets the method to specify the pulse
width to duty cycle.*/
:DISPlay:UNIT:PULSe? /*Queries the method to specify the pulse
width. The query returns DUTY.*/
```

3.3.8 :DISPlay:UNIT:RATE

Syntax

:DISPlay:UNIT:RATE <*unit*>

:DISPlay:UNIT:RATE?

Description

Sets or queries the rate unit for Sine, Square, Ramp, Pulse, Arb, and Harmonic.

Parameter

Name	Type	Range	Default
<unit>	Discrete	{FREQuency|PERiod}	FREQuency

Remarks

- **FREQuency:** frequency in Hz.
- **PERiod:** period in s.

Return Format

The query returns FREQ or PER.

Examples

```
:DISPlay:UNIT:RATE PERiod /*Sets the rate unit to s for Sine,
Square, Ramp, Pulse, Arb, and Harmonic.*/
:DISPlay:UNIT:RATE? /*Queries the rate unit for Sine, Square, Ramp,
Pulse, Arb, and Harmonic. The query returns PER.*/
```

3.3.9 :DISPlay:UNIT:SWEep

Syntax

:DISPlay:UNIT:SWEep <*type*>

:DISPlay:UNIT:SWEep?

Description

Sets or queries the method to specify the frequency sweep range.

Parameter

Name	Type	Range	Default
<type>	Discrete	{STARtstop|CENTerspan}	STARtstop

Remarks

- **STARtstop:** uses the start value (*[:SOURce[<n>]]:FREQuency:STARt*) and stop value (*[:SOURce[<n>]]:FREQuency:STOP*) to specify the frequency sweep range.
- **CENTerspan:** uses the middle value (*[:SOURce[<n>]]:FREQuency:CENTer*) and span (*[:SOURce[<n>]]:FREQuency:SPAN*) to specify the frequency sweep range.

Return Format

The query returns STAR or CENT.

Examples

```
:DISPlay:UNIT:SWEep CENTerspan /*Uses the middle value and span to
specify the frequency sweep range.*/
:DISPlay:UNIT:SWEep? /*Queries the method to specify the frequency
sweep range. The query returns CENT.*/
```

3.3.10 :DISPlay:UNIT:VOLTage

Syntax

`:DISPlay:UNIT:VOLTage` <*type*>

`:DISPlay:UNIT:VOLTage?`

Description

Sets or queries the method to specify the voltage range.

Parameter

Name	Type	Range	Default
<type>	Discrete	{AMPLitudeoff|HIGHlow}	AMPLitudeoff

Remarks

- **AMPLitudeoff:** uses the amplitude (*[:SOURce[<n>]]:VOLTage*) and offset (*[:SOURce[<n>]]:VOLTage:OFFSet*) to specify the voltage range.
- **HIGHlow:** uses the high level (*[:SOURce[<n>]]:VOLTage:HIGH*) and low level (*[:SOURce[<n>]]:VOLTage:LOW*) to specify the voltage range.

Return Format

The query returns AMPL or HIGH.

Examples

```
:DISPlay:UNIT:VOLTage HIGHlow /*Uses the high level and low level
to specify the voltage range.*/
:DISPlay:UNIT:VOLTage? /*Queries the method to specify the voltage
range. The query returns HIGH.*/
```

3.4 :HCOPy Commands

:HCOPy commands are used to set or query the image format and execute the screenshot operation.

3.4.1 :HCOPy:SDUMp:DATA?

Syntax

:HCOPy:SDUMp:DATA?

Description

Captures the current screen.

Parameter

None.

Remarks

Use *:HCOPy:SDUMp:DATA:FORMat* to set and query the returned image format (BMP/PNG).

Return Format

The query returns the data stream of the screen image.

Examples

```
None.
```

3.4.2 :HCOPy:SDUMp:DATA:FORMat

Syntax

:HCOPy:SDUMp:DATA:FORMat <*type*>

:HCOPy:SDUMp:DATA:FORMat?

Description

Sets or queries the format of the screen image.

Parameter

Name	Type	Range	Default
<type>	Discrete	{BMP|PNG}	-

Remarks

None.

Return Format

The query returns BMP or PNG.

Examples

```
:HCOPy:SDUMp:DATA:FORMat PNG /*Sets the format of the image to
PNG.*/
:HCOPy:SDUMp:DATA:FORMat? /*Queries the format of the screen image.
The query returns PNG.*/
```

3.5 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.

The standard event status register (SESR) and status byte register (SBR) record the event of a certain type happened during the use of the instrument. IEEE488.2 defines to record one specific type of event for each bit in the status register.

Table 3.21 Table of the Bit Definition of Standard Event Status Register

Bit No.	Bit Name	Decimal Value	Description
0	Operation Complete (OPC)	1	"Operation complete" indicates that all pending operations were completed following the execution of the command.
1	Not Used	2	-
2	Query Error (QYE)	4	The instrument tries to read the output buffer but it was empty. Or, a new command line was received before a previous query has been read. Or, both the input buffer and output buffer are full.
3	Device-Specific Error (DDE)	8	Indicates that an error has occurred that is neither a Command Error, a Query Error, nor an Execution Error. A Device-Specific

Bit No.	Bit Name	Decimal Value	Description
			Error is any executed device operation that did not properly complete due to some condition, such as self-check error, calibration error, or other device-specific errors.
4	Execution Error (E)	16	An execution error occurred.
5	Command Error (CME)	32	A command error (command syntax error) has occurred.
6	Not Used	64	-
7	Power On (PON)	128	Indicates that an off-to-on transition has occurred in the device's power supply since last reading or the event register was cleared.

Table 3.22 Table of the Bit Definition of Status Byte Register

Bit No.	Bit Name	Decimal Value	Description
0	Not Used	1	-
1	Not Used	2	-
2	Error Queue	4	1 or multiple errors in the error queue
3	Questionable Data Summary	8	Sets 1 or multiple bits (must be the enabled bit) in the questionable data register.
4	Message Available (MAV)	16	Indicates the available data in the output buffer.
5	Standard Event Summary	32	Sets 1 or multiple bits (must be the enabled bit) in the standard event register.
6	Master Summary Status (MSS)	64	Sets 1 or multiple bits (must be the enabled bit) in the Status Byte Register and generate the service request.

Bit No.	Bit Name	Decimal Value	Description
7	Operation Status Register	128	Sets 1 or multiple bits (must be the enabled bit) in the Operation Status Register.

3.5.1 *CLS

Syntax

***CLS**

Description

Clears all the event registers, and also clears the error queue.

Parameter

None.

Remarks

None.

Return Format

None.

Example

```
None.
```

3.5.2 *ESE

Syntax

***ESE** <*maskargument*>

***ESE?**

Description

Sets or queries the enable register of the standard event register set.

Parameter

Name	Type	Range	Default
<maskargument>	Integer	0 to 255	0

Remarks

For the definitions of the bits in the standard event register, refer to *Table 3.21 Table of the Bit Definition of Standard Event Status Register* . The value of <maskargument>

is the sum of the decimal values of all bits set in the standard event register. For example, to enable Bit 2 (4 in decimal), Bit 3 (8 in decimal), and Bit 7 (128 in decimal), set the <maskargument> to 140 (4+8+128).

Return Format

The query returns an integer. The integer equals to the decimal-weighted sum of all the bits set in the register.

Example

```
*ESE 16 /*Enables Bit 4 (16 in decimal) in the register.*/
*ESE? /*The query returns the enable value of the register 16.*/
```

3.5.3 *ESR?

Syntax

***ESR?**

Description

Queries and clears the event register of the standard event status register.

Parameter

None.

Remarks

Bit 1 and Bit 6 in the standard event status register (*Table 3.21 Table of the Bit Definition of Standard Event Status Register*) are not used and are always treated as 0; therefore, the range of the returned value is a decimal number corresponding to a binary number X0XXXX0X (X is 1 or 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

```
None.
```

3.5.4 *IDN?

Syntax

***IDN?**

Description

Queries the ID string of the instrument.

Parameter

None.

Remarks

None.

Return Format

The query returns RIGOL TECHNOLOGIES,<model>,<serial number>,<software version>.

- **<model>:** indicates the model number of the instrument.
- **<serial number>:** indicates the serial number of the instrument.
- **<software version>:** indicates the software version of the instrument.

Example

```
None.
```

3.5.5 *OPC

Syntax

***OPC**

***OPC?**

Description

Sets the OPC (bit 0, "Operation Complete") of the Standard Event register to 1 after all commands are executed.

Queries whether all the previous commands are executed. The query returns 1 to the output buffer after the command is executed.

Parameter

None.

Remarks

- Operation complete means that all the previous commands including the *OPC command have been executed.
- When setting the instrument configuration through programming (by executing the command string), using this command as the last command can determine when the command queue is executed (when the command queue is executed, the bit0 (OPC, "operation complete" bit) in the event register of the Standard Event register will be set).
- Sending the *OPC? command and reading the result can ensure synchronization.

Return Format

Queries whether all the previous commands are executed. The query returns 1 when all commands are executed.

Examples

```
None.
```

3.5.6 *OPT?

Syntax

***OPT?**

Description

Queries the options installed in your instrument

Parameter

None.

Remarks

None.

Return Format

The query returns the options installed, separated by ",". If an option has been installed, the query returns the option name; if it is not installed, the query returns NONE.

Examples

```
*OPT? /*Queries the options installed. The query returns NONE.*/
```

3.5.7 *PSC

Syntax

***PSC** <*bool*>

***PSC?**

Description

Enables or disables the function of clearing the enable registers of the Status Byte and Standard Event registers at power-on.

Queries the on/off state of the function of clearing the enable registers of the Status Byte and Standard Event registers at power-on.

Parameter

Name	Type	Range	Default
<bool>	Bool	{0|1}	0

Remarks

- *PSC 1 denotes clearing the enable registers of the Status Byte and Standard Event registers at power-on; *PSC 0 denotes that the enable registers of the Status Byte and Standard Event registers will not be affected at power-on.
- You can also send **SRE* (*SRE 0) or (*ESE 0) to clear the enable registers of the Status Byte and Standard Event registers respectively.

Return Format

The query returns 0 or 1.

Examples

```
*PSC 1 /*Enables the function of clearing the enable registers of
the Status Byte and Standard Event registers at power-on.*/
*PSC? /*Queries the on/off state of the function of clearing the
registers at power-on. The query returns 1.*/
```

3.5.8 *RCL

Syntax

***RCL** <*value*>

Description

Recalls a previously stored instrument state from the specified non-volatile memory location.

Parameter

Name	Type	Range	Default
<value>	Discrete	{0|1|2|3|4|5}	0

Remarks

None.

Return Format

None.

Examples

```
*RCL 1 /*Recalls the instrument state stored in memory location 1.*/
```

3.5.9 *RST

Syntax

***RST**

Description

Resets the instrument to its factory default state.

Parameter

None.

Remarks

None.

Return Format

None.

Examples

```
None.
```

3.5.10 *SAV

Syntax

***SAV** <*value*>

Description

Stores the current instrument state to a specified location in non-volatile memory.

Parameter

Name	Type	Range	Default
<value>	Discrete	{0|1|2|3|4|5}	0

Remarks

If a file has already been stored in the specified location, this command will overwrite the original file directly without any prompt message.

Return Format

None.

Examples

```
*SAV 1 /*Stores the current instrument state to memory location 1.*/
```

3.5.11 *SRE

Syntax

***SRE** <*maskargument*>

***SRE?**

Description

Sets or queries the enable register of the status byte register set.

Parameter

Name	Type	Range	Default
<maskargument>	Integer	0 to 255	0

Remarks

For the definitions of the bits in the status byte register, refer to *Table 3.22 Table of the Bit Definition of Status Byte Register*. The value of <maskargument> is the sum of the decimal values of all bits set in the status byte register. For example, to enable Bit 2 (4 in decimal), Bit 3 (8 in decimal), and Bit 7 (128 in decimal), set the <maskargument> to 140 (4+8+128).

Return Format

The query returns an integer. The integer equals to the decimal-weighted sum of all the bits set in the register.

Example

```
*SRE 16 /*Enables Bit 4 (16 in decimal) in the register.*/
*SRE? /*The query returns the enable value of the register 16.*/
```

3.5.12 *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

None.

Remarks

Bit 0 and Bit 1 in the status byte register (*Table 3.22 Table of the Bit Definition of Status Byte Register*) are not used and are always treated as 0; therefore, the range of

the returned value is a decimal number corresponding to a binary number XXXXXX00 (X is 1 or 0).

Return Format

The query returns an integer. The integer equals to the decimal-weighted sum of all the bits set in the register.

Example

```
None.
```

3.5.13 *TRG

Syntax

`*TRG`

Description

Generates a trigger event.

Parameter

None.

Remarks

- This command is valid only when the sweep or burst function is enabled and the trigger source is set to manual.
- When the sweep or burst function is enabled and the trigger source is set to manual, you can also send *:TRIGger<n>[:IMMediate]* to trigger a sweep or burst.

Return Format

None.

Examples

```
*TRG /*Generates a trigger event.*/
```

3.5.14 *TST?

Syntax

`*TST?`

Description

Executes the self-test and returns the results.

Parameter

None.

Remarks

None.

Return Format

The query returns a decimal integer. The query returns a 0 if the self-test passes.

Examples

```
None.
```

3.5.15 *WAI

Syntax

`*WAI`

Description

Waits for the operation to complete. This command is used to make sure that the previous command is complete before the next command is issued.

Parameter

None.

Remarks

For example, you want to use **`:SOURce1:SWEep:SPACing LINear`** and then use **`:TRIGger1:IMMediate`**. To make sure that **`:SOURce1:SWEep:SPACing LINear`** can be completed before executing the next command, you can insert **`*WAI`** between the two commands.

Return Format

None.

Examples

```
None.
```

3.6 :INITiate Commands

:INITiate commands are used to set or query the "wait-for-trigger" state of the instrument.

NOTE

The states of the trigger system include "wait-for-trigger", "action-in-progress", and "idle".

- Wait-for-trigger: when the system is in "wait-for-trigger" state, it can detect the trigger event. Once a trigger event is generated, the system enters the "action-in-progress" state.

- Action-in-progress: once a channel is triggered, the system leaves the "wait-for-trigger" state and enters the"action-in-progress" state (e.g. burst-in-progress or sweep-in-progress). After the output is finished, the system returns to the "idle" state or "wait-for-trigger" state (set by *:INITiate[<n>]:CONTinuous* or *:INITiate[<n>]:CONTinuous:ALL*).
- Idle: the system ignores the trigger signal in the "idle" state. You can use *:INITiate[<n>][:IMMediate]* or *:INITiate[<n>][:IMMediate]:ALL* to change the trigger state from "idle" to "wait-for-trigger" for a specified channel or both channels.

3.6.1 :INITiate[<n>]:CONTinuous

Syntax

:INITiate[<*n*>]**:CONTinuous** <*bool*>

:INITiate[<*n*>]**:CONTinuous?**

Description

Sets or queries whether the trigger system always returns to the "wait-for-trigger" state for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	-
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- 1|ON: returns to the "wait-for-trigger" state; 0|OFF: remains in the "idle" state, ignoring triggers until *:INITiate[<n>][:IMMediate]* or *:INITiate[<n>][:IMMediate]:ALL* is issued.
- With 1|ON, the trigger count setting (*:TRIGger<n>:COUNt*) will be invalid. If you need to count the triggers, you must select 0|OFF.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:INITiate1:CONTinuous ON /*Sets CH1 to automatically return to
"wait-for-trigger".*/
:INITiate1:CONTinuous? /*Queries whether CH1 automatically returns
to "wait-for-trigger". The query returns 1.*/
```

3.6.2 :INITiate[<n>]:CONTinuous:ALL

Syntax

`:INITiate[<n>]:CONTinuous:ALL <bool>`

Description

Sets or queries whether the trigger system always returns to the "wait-for-trigger" state for both channels.

Parameter

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

Remarks

- 1|ON: returns to the "wait-for-trigger" state; 0|OFF: remains in the "idle" state, ignoring triggers until *:INITiate[<n>][:IMMediate]* or *:INITiate[<n>][:IMMediate]:ALL* is issued.
- With 1|ON, the trigger count setting (*:TRIGger<n>:COUNt*) will be invalid. If you need to count the triggers, you must select 0|OFF.

Return Format

None.

Examples

```
None.
```

3.6.3 :INITiate[<n>][:IMMediate]

Syntax

`:INITiate[<n>][:IMMediate]`

Description

Changes the status of the trigger system from "idle" to "wait-for-trigger" for the specified channel.

Parameter

None.

Remarks

- Use *:ABORt* to return the instrument to idle.

- If the specified channel has *:INITiate[<n>]:CONTinuous* set to ON, *:INITiate[<n>][:IMMediate]* and *:INITiate[<n>][:IMMediate]:ALL* have no effect on the trigger system.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
None.
```

3.6.4 :INITiate[<n>][:IMMediate]:ALL

Syntax

:INITiate[<*n*>][**:IMMediate**]**:ALL**

Description

Changes the status of the trigger system from "idle" to "wait-for-trigger" for both channels.

Parameter

None.

Remarks

- Use *:ABORt* to return the instrument to idle.
- If the specified channel has *:INITiate[<n>]:CONTinuous* set to ON, *:INITiate[<n>][:IMMediate]* and *:INITiate[<n>][:IMMediate]:ALL* have no effect on the trigger system.

Return Format

None.

Examples

```
None.
```

3.7 :LXI Commands

:LXI commands are used to set the on/off status of mDNS, restart the LAN, and restore the network settings to default values.

3.7.1 :LXI:MDNS:STATe

Syntax

:LXI:MDNS:STATe <*bool*>

:LXI:MDNS:STATe?

Description

Sets or queries the on/off status of the multicast Domain Name System (mDNS).

Parameter

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

Remarks

You can enable this system to provide the function of DNS server for service discovery in a small network without a DNS server.

Return Format

The query returns 0 or 1.

Examples

```
:LXI:MDNS:STATe ON /*Enables the mDNS.*/
:LXI:MDNS:STATe? /*Queries whether the mDNS is enabled. The query
returns 1.*/
```

3.7.2 :LXI:MDNS:SNAMe:DESired

Syntax

:LXI:MDNS:SNAMe:DESired <*name*>

:LXI:MDNS:SNAMe:DESired?

Description

Sets or queries the service name for mDNS.

Parameter

Name	Type	Range	Default
<name>	ASCII string	Refer to *Remarks*	-

Remarks

<name> specifies the mDNS service name, which can contain English characters and numbers. The total length is no more than 99 characters.

Return Format

The query returns an ASCII string.

Examples

```
:LXI:MDNS:SNAMe:DESired RIGOL /*Sets the mDNS service name to
RIGOL.*/
:LXI:MDNS:SNAMe:DESired? /*Queries the service name for mDNS. The
query returns RIGOL.*/
```

3.7.3 :LXI:RESet

Syntax

`:LXI:RESet`

Description

Sets the network settings to default values.

Parameter

None.

Remarks

It may take a few seconds for the LAN interface to restart after you send this command.

Return Format

None.

Examples

```
:LXI:RESet /*Sets the network settings to default values.*/
```

3.7.4 :LXI:RESTart

Syntax

`:LXI:RESTart`

Description

Restarts the LAN interface.

Parameter

None.

Remarks

It may take a few seconds for the LAN interface to restart after you send this command.

Return Format

None.

Examples

```
:LXI:RESTart /*Restarts the LAN interface.*/
```

3.8 :MEMory Commands

:MEMory commands are used to set and query the state file stored in the internal non-volatile memory of the instrument.

3.8.1 :MEMory:NSTates?

Syntax

`:MEMory:NSTates?`

Description

Queries the number of memory locations for storing states.

Parameter

None.

Remarks

None.

Return Format

The query returns 6.

Examples

```
:MEMory:NSTates? /*Queries the number of memory locations for the
state files in the non-volatile memory of the instrument. The query
returns 6.*/
```

3.8.2 :MEMory:STATe:CATalog?

Syntax

`:MEMory:STATe:CATalog?`

Description

Queries the names of the state files in the internal non-volatile memory of the instrument.

Parameter

None.

Remarks

This instrument provides 6 memory locations for state files in the internal non-volatile memory.

Return Format

The query returns a string which consists of six parts (representing the filenames of the files stored in locations 0 to 5 respectively), separated by commas. For example, the query might return
"AUTO_RECALL","STATE_1","STATE_2","STATE_3","STATE_4","STATE_5".

Examples

```
:MEMory:STATe:CATalog? /*Queries the names of locations 0 to 5. The
query returns
"AUTO_RECALL","STATE_1","STATE_2","STATE_3","STATE_4","STATE_5".*/
```

3.8.3 :MEMory:STATe:DELete

Syntax

:MEMory:STATe:DELete <*n*>

Description

Deletes the state files stored in the specified memory location.

Parameter

Name	Type	Range	Default
<n>	Discrete	{0|1|2|3|4|5}	-

Remarks

- This instrument provides 6 memory locations for state files in the internal non-volatile memory. 0 to 5 represents the state files stored in the specified locations respectively.
- This command is valid only when a state file has been stored in the specified memory location.

Return Format

None.

Examples

```
:MEMory:STATe:DELete 1 /*Deletes the state file stored in location
1.*/
```

3.8.4 :MEMory:STATe:NAME

Syntax

:MEMory:STATe:NAME <*n*>[,<*name*>]

:MEMory:STATe:NAME? <*n*>

Description

Sets or queries the filename of the state file stored in the internal non-volatile memory of the instrument.

Parameter

Name	Type	Range	Default
<n>	Discrete	{0|1|2|3|4|5}	-
<name>	ASCII string	Refer to *Remarks*	-

Remarks

- The parameter <name> is a string without quotation marks. It can contain up to 12 characters. The first character must be an English letter (A-Z). The other characters can be English letters, numbers (0-9), or underline (_). If the name is omitted, the factory default name will be used.
- This instrument provides 6 memory locations for state files in the internal non-volatile memory. 0 to 5 represents the state file stored in the specified locations respectively.

Return Format

The query returns a string with double quotation marks. For example, the query might return "state".

Examples

```
:MEMory:STATe:NAME 2,state /*Sets the filename of the state file
stored in location 2 of the internal non-volatile memory of the
instrument.*/
:MEMory:STATe:NAME? 2 /*Queries the filename of the state file
stored in location 2 of the internal non-volatile memory of the
instrument. The query returns "state".*/
```

3.8.5 :MEMory:STATe:RECall:AUTO

Syntax

:MEMory:STATe:RECall:AUTO <*bool*>

:MEMory:STATe:RECall:AUTO?

Description

Sets or queries whether to enable the automatic recall of the last power-on setting.

Parameter

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

Remarks

- **ON/1:** the instrument recalls the instrument state (at last power-off) in memory location 0 at the next power-on.
- **OFF/0:** the instrument uses the factory default values (except the parameters not affected by factory default settings) at the next power-on.

Return Format

The query returns 0 or 1.

Examples

```
:MEMory:STATe:RECall:AUTO ON /*Enables the automatic recall of the
last power-on setting in location 0 at the next power-on.*/
:MEMory:STATe:RECall:AUTO? /*Queries whether to enable the
automatic recall of the last power-on setting in location 0. The
query returns 1.*/
```

3.8.6 :MEMory:STATe:VALid?

Syntax

:MEMory:STATe:VALid? <*state*>

Description

Queries whether a state file is available in the specified memory location in the internal non-volatile memory of the instrument.

Parameter

Name	Type	Range	Default
<state>	Discrete	{0|1|2|3|4|5}	-

Remarks

Before using **SAV*, you can use this query command to avoid overwriting an existing file accidentally.

Return Format

The query returns 1 or 0 . 1 indicates that an existing state file has been stored in the specified location while 0 indicates that no state file is stored in the specified location.

Examples

```
:MEMory:STATe:VALid? 2 /*Queries whether an existing state file is
stored in the location 2 in the internal non-volatile memory of the
instrument. The query returns 1.*/
```

3.9 :MMEMory Commands

:MMEMory commands are used to set and query the related information of the internal and external memory. The internal memory ("INT:\") of the instrument is always present while the external memory ("USB:\") is available only when the rear-panel USB HOST interface detects the USB storage device.

Many **:MMEMory** commands refer to folders and files. These folders and files have specific structures, described below.

- Absolute paths begin with "/" or "\" and start at the driver identifier (INT or USB).
- Folder and file names cannot contain the following characters: \ / : * ? " < > |, and cannot start with ".".
- Relative paths begin with ".\" or "./" and level down from the current path.
- Going back to the parent path begins with "../" or "..\" and levels up from the current path.
- The combination of folder and file name cannot exceed 200 characters.

3.9.1 :MMEMory:CATalog[:ALL]?

Syntax

:MMEMory:CATalog[:ALL]? [<*folder*>]

Description

Queries all the files in the specified directory.

Parameter

Name	Type	Range	Default
<folder>	ASCII string	Valid directory	-

Remarks

The parameter <folder> is a valid directory in the internal or external memory. The query might return INT:\folder or USB:\Rigol. If it is omitted, this command queries the directory specified by *:MMEMory:CDIRectory*.

Return Format

The query returns a string in the format of space used,space available,"file name,file property,file size",.....; wherein, the units of the space used and space available are byte; the file name includes the file extension (if any); the file property is either STAT for STATe (*.sta) files, ASC for Arb files stored in ASCII for floating-point data (*.csv), FOLD for folders, SEQ for .seq files, ARB for Arb wave (*.arb) stored in floating-point data (-1.0-1.0), BARB for Arb wave stored as DAC code values (*.barb), GEL for upgrade files (*.GEL, *.gel), or null for all other file extensions; the file size is expressed in bytes. For example, the query might return 96256,1019770880,"command.exe,,375808", "MySetup.sta,STAT,8192", "MyWave.csv,ASC,11265", indicating that the space used is 96256 bytes and the space available is 1019770880 bytes; it contains one executable file, one state file (MySetup.sta), and one Arb file (MyWave.csv).

If no file exists in the target directory, the query only returns the space used and space available.

Examples

```
:MMEMory:CATalog:ALL? USB:\Mydata /*Queries all the files in the
folder named Mydata in the USB storage device. The query returns
96256,1019770880,"command.exe,,375808", "MySetup.sta,STAT,8192",
"MyWave.csv,ASC,11265".*/
```

3.9.2 :MMEMory:CATalog:DATA:ARBitrary?

Syntax

:MMEMory:CATalog:DATA:ARBitrary? [<*folder*>]

Description

Queries all the arbitrary waveform files (*.arb/*.barb/*.csv) and sequence files (*.seq) in the specified directory.

Parameter

Name	Type	Range	Default
<folder>	ASCII string	Valid directory	-

Remarks

The parameter <folder> is the valid directory in the internal or external memory. For example, it can be INT:\ or USB:\Rigol\. If it is omitted, this command queries the directory specified by *:MMEMory:CDIRectory*.

Return Format

The query returns a string in the format of space used,space available,"file name,file property,file size",.....; wherein, the units of the space used and space available are byte; the file name includes the file extension; the file property is either SEQ for .seq files, ARB for Arb wave (*.arb) stored in floating-point data (-1.0-1.0), BARB for Arb wave stored as DAC code values (*.barb), or ASC for Arb files stored in ASCII for floating-point data (*.csv); the file size is amount of space that the file uses. For example, the query might return 28672,4102361088,"Rigol1.seq,SEQ,1254","test.arb,ARB,587", indicating that the space used is 28672 bytes and the space available is 4102361088 bytes; it contains one sequence file (Rigol1.seq) and one Arb file; their sizes are 1254 bytes and 587 bytes.

If no Arb file or sequence file exists in the directory, the query only returns the space used and space available.

Examples

```
:MMEMory:CATalog:DATA:ARBitrary? INT:\folder /*Queries all the
arbitrary waveform files and sequence files in the folder named
"folder" under C disk. The query returns
28672,4102361088,"Rigol1.seq,SEQ,1254","test.arb,ARB,587".*/
```

3.9.3 :MMEMory:CATalog:STATe?

Syntax

:MMEMory:CATalog:STATe? [<*folder*>]

Description

Queries the state file (*.sta) under the specified directory.

Parameter

Name	Type	Range	Default
<folder>	ASCII string	Valid directory	-

Remarks

The parameter <folder> is a valid directory in the internal or external memory. For example, it can be INT:\ or USB:\Rigol. If it is omitted, this command queries the current directory.

Return Format

The query returns a string in the format of space used,space available,"file name,file property,file size",.....; wherein, the units of the space used and space available are byte; the file name includes the file extension; the file property is STAT; the file size is the amount of space that file uses. For example, the query might return 28672,4102361088,"MySetup.sta,STAT,8192", indicating that the space used is 28672

bytes and the space available is 4102361088 bytes; it contains one state file (MySetup.sta); the file size is 8192 bytes.

If no state file exists in the target directory, the query only returns the space used and space available.

Examples

```
:MMEMory:CATalog:STATe? /*Queries the state file under the current
directory. The query returns
28672,4102361088,"MySetup.sta,STAT,8192".*/
```

3.9.4 :MMEMory:CDIRectory

Syntax

:MMEMory:CDIRectory <*directory_name*>

:MMEMory:CDIRectory?

Description

Sets or queries the default directory for *:MMEMory Commands*.

Parameter

Name	Type	Range	Default
<directory_name>	ASCII string	Valid directory	-

Remarks

The parameter <directory_name> is the valid directory in the internal or external memory. For example, it can be INT:/folder or USB:/Rigol.

Return Format

The query returns a string. For example, the query might return INT:/folder, representing the folder named "folder" under local disk C.

Examples

```
:MMEMory:CDIRectory INT:/folder  /*Sets the default directory to
INT:/folder (the "folder" in internal memory).*/
:MMEMory:CDIRectory? /*Queries the default directory. The query
returns INT:/folder.*/
```

3.9.5 :MMEMory:COPY

Syntax

:MMEMory:COPY <*file_name*>,<*directory_name*>

Description

Copies a file to the specified directory (not the current directory).

Parameter

Name	Type	Range	Default
<file_name>	ASCII string	Valid filename	-
<directory_name>	ASCII string	Valid directory	-

Remarks

- The parameter <file_name> specifies the name of the file to be copied, for example, INT:\Arb.raf.
- The parameter <directory_name> specifies the target path, for example, INT:\TextFolder.
- This command overwrites the existing file of the same name under the target path.

Return Format

None.

Examples

```
:MMEMory:COPY INT:\Arb.raf,INT:\TextFolder /*Copies the Arb.raf
file under C disk to the directory named "TextFolder" under C
disk.*/
```

3.9.6 :MMEMory:COPY:SEQuence

Syntax

:MMEMory:COPY:SEQuence <*sequence*>,<*directoryname*>

Description

Copies a specified sequence file (.seq) to the specified directory.

Parameter

Name	Type	Range	Default
<sequence>	ASCII string	Valid sequence filename	-
<directoryname>	ASCII string	Valid directory	-

Remarks

- The parameter <sequence> is the filename of the sequence to be copied. The filename must contain the file extension, for example, INT:\SEQ.seq.
- The parameter <directoryname> specifies the target path, for example, INT:\TextFolder.

- This command overwrites the existing file of the same name under the target path.

Return Format

None.

Examples

```
:MMEMory:COPY:SEQuence INT:\Rigol\MySequence.seq,USB:\rigol /
*Copies the sequence file named MySequence.seq and its associated
files in the "Rigol" folder of C disk to the "rigol" folder of USB
storage device.*/
```

3.9.7 :MMEMory:DELete

Syntax

:MMEMory:DELete <*file_name*>

Description

Deletes a specific file under the specified directory.

Parameter

Name	Type	Range	Default
<file_name>	ASCII string	Valid filename	-

Remarks

- The parameter <file_name> is a valid directory in internal or external memory, for example, INT:\screenshot.png.
- To delete a folder, use *:MMEMory:RDIRectory*.

Return Format

None.

Examples

```
:MMEMory:DELete INT:\screenshot.png /*Deletes the image named
screenshot.png under C disk.*/
```

3.9.8 :MMEMory:LOAD:DATA

Syntax

:MMEMory:LOAD:DATA <*n*>,<*file_name*>

Description

Loads the sequence file (*.seq) or arbitrary waveform file (*.arb) from internal or external memory into the volatile memory of the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<file_name>	ASCII string	Valid filename	-

Remarks

- The parameter <n> specifies the channel (CH1 or CH2) to which the specified sequence or Arb file under the current directory is loaded.
- The parameter <file_name> specifies the name of the file under the specified directory, for example, INT:\SEQ.seq.

Return Format

None.

Examples

```
:MMEMory:LOAD:DATA 1,INT:\SEQ.seq /*Loads the sequence file named
SEQ.seq under the C disk into CH1.*/
```

3.9.9 :MMEMory:LOAD:STATe

Syntax

:MMEMory:LOAD:STATe <*file_name*>

Description

Loads the specified state file.

Parameter

Name	Type	Range	Default
<file_name>	ASCII string	Valid name of state file	-

Remarks

The parameter <file_name> specifies the name of the state file under the specified directory, for example, INT:\Mystate.sta.

Return Format

None.

Examples

```
:MMEMory:LOAD:STATe INT:\Mystate.sta /*Loads the instrument state
file named Mystate.sta from the root directory of the internal
memory.*/
```

3.9.10 :MMEMory:MDIRectory

Syntax

:MMEMory:MDIRectory <*dir_name*>

Description

Creates an empty folder with the specified name in the mass memory system.

Parameter

Name	Type	Range	Default
<dir_name>	ASCII string	Refer to *Remarks*	-

Remarks

- The parameter <dir_name> specifies the name of the folder to be created.
- If the specified directory has an existing folder of the same name, a prompt message will be displayed, indicating that errors occur on the remote command.

Return Format

None.

Examples

```
:MMEMory:MDIRectory TestFolder /*Creates a folder named
"TestFolder" in the internal mass memory.*/
```

3.9.11 :MMEMory:MOVE

Syntax

:MMEMory:MOVE <*file1*>,<*file2*>

Description

Moves file 1 under the current directory to the specified directory, or renames file 1 to file 2.

Parameter

Name	Type	Range	Default
<file1>	ASCII string	Valid filename	-
<file2>	ASCII string	Valid directory or filename	-

Remarks

- The parameter <file1> is the file under the current directory or the specified directory, for example, Rigol.sta.
- The parameter <file2> is the valid directory or filename in the internal or external memory. For example, it can be INT:\ or USB:\Rigol\.
- If you want to rename the file, <file1> and <file2> should share the same folder; if you want to move the file, then <file2> should be specified as another valid directory such as USB:\Rigol\.

Return Format

None.

Examples

```
:MMEMory:MOVE INT:\Rigol.sta,USB:\Rigol\ /*Moves the file named
Rigol.sta in internal memory to the folder named Rigol in the USB
storage device.*/
:MMEMory:MOVE USB:\Rigol1.sta,USB:\Rigol2.sta /*Names the file
named Rigol1.sta in the USB storage device to Rigol2.sta.*/
```

3.9.12 :MMEMory:RDIRectory

Syntax

`:MMEMory:RDIRectory` <*folder*>

Description

Deletes the specified directory (empty folder) in the mass memory system.

Parameter

Name	Type	Range	Default
<folder>	ASCII string	Folder name of the empty folder	-

Remarks

You can only delete empty folders, or an error message will be displayed.

Return Format

None.

Examples

```
:MMEMory:RDIRectory folder /*Deletes the empty folder named
"folder" in the internal mass memory.*/
```

3.9.13 :MMEMory:STORe:DATA

Syntax

:MMEMory:STORe:DATA <*n*>,<*file_name*>

Description

Stores a sequence file (*.seq) in volatile memory of the specified channel to the specified directory.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<file_name>	ASCII string	Valid directory	-

Remarks

None.

Return Format

None.

Examples

```
:MMEMory:STORe:DATA 1,INT:\Seq_1.seq /*Stores a sequence file
(*.seq) in volatile memory of CH1 to the internal memory with the
name of Seq_1.seq.*/
```

3.9.14 :MMEMory:STORe:STATe

Syntax

:MMEMory:STORe:STATe <*file_name*>

Description

Stores the current instrument state with the specified name in the specified directory.

Parameter

Name	Type	Range	Default
<file_name>	ASCII string	Valid directory	-

Remarks

None.

Return Format

None.

Examples

```
:MMEMory:STORe:STATe INT:\state.sta  /*Stores the instrument state
in the file named state.sta under C disk.*/
```

3.10 :OUTPut Commands

:OUTPut commands are used to set and query the channel outputs.

TIP

As DG821 Pro has only one channel, it only supports CH1 parameter settings (unless the two-channel upgrade option is installed).

3.10.1 :OUTPut[<n>]:IDLE

Syntax

:OUTPut[<*n*>]**:IDLE** {<*idle*>|<*position*>}

:OUTPut[<*n*>]**:IDLE?**

Description

Sets or queries the idle level position of the burst mode for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<idle>	Integer	0 to 65535	-
<position>	Discrete	{FPT|TOP|CENTer|BOTTom}	FPT

Remarks

- The parameter <idle> is used to self-define the idle level position and <position> allows you to set the position to the first point (FPT), the top (TOP), the center (CENTer), or the bottom (BOTTom) of the waveform.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns FPT, TOP, CENT, or BOTT; or returns the idle level position in an integer.

Examples

```
:OUTPut1:IDLE TOP /*Sets the idle level to the top point of the
waveform.*/
```

```
:OUTPut1:IDLE? /*Queries the idle level position of the burst
signal. The query returns TOP.*/
```

3.10.2 :OUTPut[<n>]:LOAD

Syntax

:OUTPut[<*n*>]**:LOAD** {<*ohms*>|<*lim_set*>}

:OUTPut[<*n*>]**:LOAD?** <*lim_query*>

Description

Sets or queries the output impedance for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<ohms>	Integer	1 Ω to 10 kΩ	50 Ω
<lim_set>	Discrete	{INFinity|MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- When [<n>] is omitted, it is interpreted as CH1.
- The parameter <ohms> indicates setting the output impedance to a specified value. The <lim_set> allows you to set the impedance to HighZ (INFinity), the minimum value (MINimum), the maximum value (MAXimum), and the default value (DEFault).
- The output impedance setting affects the output amplitude and DC offset. If the actual load impedance differs from the value specified, the voltage level displayed would not match the voltage level of the device under test. To ensure correct voltage level, the load impedance setting must match the actual load.

Return Format

The query returns the output impedance in scientific notation. For example, the query might return +1.000000000000000E+02, indicating that the output impedance is 100 Ω. If the output impedance of the output connector of the specified channel is set to HighZ (INFinity), the query returns 9.9E+37.

Examples

```
:OUTPut1:LOAD INFinity /*Sets the output impedance to HighZ for CH1
output connector.*/
```

```
:OUTPut1:LOAD? /*Queries the output impedance for CH1 output
connector. The query returns 9.9E+37.*/
```

3.10.3 :OUTPut[<n>]:POLarity

Syntax

:OUTPut[<*n*>]**:POLarity** <*polarity*>

:OUTPut[<*n*>]**:POLarity?**

Description

Sets or queries the output polarity for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<polarity>	Discrete	{NORMal|INVerted}	NORMal

Remarks

- The output polarity sets the signal on the output connector to normal (NORMal) or inverted (INVerted). In Normal mode, the instrument outputs a normal waveform; in Invert mode, the instrument inverts the waveform and then outputs the inverted waveform.
- The waveform is inverted relative to the offset voltage. The offset voltage remains unchanged when the waveform is inverted. The sync signal related to the waveform is not inverted.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns NORM or INV.

Examples

```
:OUTPut1:POLarity NORMal /*Sets the CH1 output polarity to Normal.*/
:OUTPut1:POLarity? /*Queries the CH1 output polarity. The query
returns NORM.*/
```

3.10.4 :OUTPut[<n>][:STATe]

Syntax

:OUTPut[<*n*>][**:STATe**] <*state*>

:OUTPut[<*n*>][**:STATe?**]

Description

Sets or queries the output on/off status for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<state>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- 0/OFF disables the channel output; 1/ON enables the channel output.
- The parameter [<n>] is used to specify the channel. If [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:OUTPut1:STATe 1 /*Enables the output of CH1.*/
:OUTPut1:STATe? /*Queries the output on/off status for CH1. The
query returns 1.*/
```

3.10.5 :OUTPut[<n>]:SYNC

Syntax

:OUTPut[<*n*>]**:SYNC** <*state*>

:OUTPut[<*n*>]**:SYNC?**

Description

Sets or queries the output state of the sync signal.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<state>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- Sine, Square, Ramp, Pulse, Arb (except DC), Sweep waveforms, Bursts, and modulated waveforms have associated sync signals. For the characteristics of sync signals for different signals, refer to the user manual of this product.

- You can use *:OUTPut[<n>]:SYNC:SOURce* to select the output terminal of the sync signal.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:OUTPut1:SYNC ON /*Enables the sync signal output.*/
:OUTPut1:SYNC? /*Queries the output state of the sync signal. The
query returns 1.*/
```

3.10.6 :OUTPut[<n>]:SYNC:MODE

Syntax

:OUTPut[<*n*>]**:SYNC:MODE** <*mode*>

:OUTPut[<*n*>]**:SYNC:MODE?**

Description

Sets or queries whether the frequency mark function is enabled for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<mode>	Discrete	{NORMal|MARKer}	NORMal

Remarks

- **NORMal:** disables the frequency mark function.
- **MARKer:** enables the frequency mark function.
- This function can be enabled only when the Sweep function (*[:SOURce[<n>]]:SWEep:STATe*) is enabled for the specified channel.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns NORM or MARK.

Examples

```
:OUTPut1:SYNC:MODE MARKer /*Enables the frequency mark function for
CH1.*/
```

```
:OUTPut1:SYNC:MODE? /*Queries whether the frequency mark function
is enabled for CH1. The query returns MARK.*/
```

3.10.7 :OUTPut[<n>]:SYNC:POLarity

Syntax

:OUTPut[<*n*>]**:SYNC:POLarity** <*polarity*>

:OUTPut[<*n*>]**:SYNC:POLarity?**

Description

Sets or queries the polarity of sync signal for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<polarity>	Discrete	{NORMal|INVerted}	NORMal

Remarks

- It sets the sync signal output polarity to rising edge (NORMal) output or falling edge (INVerted) output.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns NORM or INV.

Examples

```
:OUTPut1:SYNC:POLarity NORMal /*Sets the output polarity of CH1
sync signal to rising edge.*/
:OUTPut1:SYNC:POLarity? /*Queries the output polarity of CH1 sync
signal. The query returns NORM.*/
```

3.10.8 :OUTPut[<n>]:SYNC:SOURce

Syntax

:OUTPut[<*n*>]**:SYNC:SOURce** <*port*>

:OUTPut[<*n*>]**:SYNC:SOURce?**

Description

Sets or queries the sync signal output port for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<port>	Discrete	{BACK|FRONt}	BACK

Remarks

- **FRONt:** front port. When the output of CH1/CH2 sync signal is enabled, the sync signal is output from the front-panel **[CH2]**/**[CH1]** connector. In this case, the front port for sync signal output cannot be configured.
- **BACK:** rear port. When the sync signal output is enabled, it is output from the rear-panel **[AUX IN/OUT]** connector.

TIP

As DG821 Pro has only one channel, it only supports rear-port sync signal output (unless the two-channel upgrade option is installed).

Return Format

The query returns FRON or BACK.

Examples

```
:OUTPut1:SYNC:SOURce FRONt /*Sets the sync signal output port to
front port for CH1.*/
:OUTPut1:SYNC:SOURce? /*Queries the sync signal output port for
CH1. The query returns FRON.*/
```

3.10.9 :OUTPut[<n>]:TRIGger

Syntax

:OUTPut[<*n*>]**:TRIGger** <*bool*>

:OUTPut[<*n*>]**:TRIGger?**

Description

Sets or queries whether the trigger output is enabled for Sweep or Burst mode.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- After it is enabled, the instrument outputs a pulse with the specified edge (*:OUTPut[<n>]:TRIGger:SLOPe*) via the rear-panel **[AUX IN/OUT]** connector at the beginning of the burst or sweep.
- The trigger output is disabled when the burst trigger source is set to External or the burst type is set to Gated. The rear-panel **[AUX IN/OUT]** connector cannot be used for trigger input and trigger output at the same time.
- The trigger output is disabled when the sweep trigger source is set to External.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:OUTPut:TRIGger ON /*Enables the trigger output for CH1.*/
:OUTPut:TRIGger? /*Queries whether the trigger output is enabled
for CH1. The query returns 1.*/
```

3.10.10 :OUTPut[<n>]:TRIGger:SLOPe

Syntax

:OUTPut[<*n*>]**:TRIGger:SLOPe** <*type*>

:OUTPut[<*n*>]**:TRIGger:SLOPe?**

Description

Sets or queries the edge of the trigger output signal for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<type>	Discrete	{POSitive|NEGative}	POSitive

Remarks

- POSitive: outputs a leading edge pulse; NEGative: outputs a trailing edge pulse.
- After the trigger output is enabled (*:OUTPut[<n>]:TRIGger*), the instrument outputs a pulse with the specified edge from the rear-panel **[AUX IN/OUT]** connector at the beginning of the sweep or burst.
- When [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns POS or NEG.

Examples

```
:OUTPut1:TRIGger:SLOPe NEGative /*Sets the edge of the trigger
output signal to trailing edge for CH1.*/
:OUTPut1:TRIGger:SLOPe? /*Queries the edge of the trigger output
signal for CH1. The query returns NEG.*/
```

3.11 :SOURce Commands

:SOURce commands are used to set and query channel and waveform parameters including basic waveform, sequence, harmonic, modulation, sweep, burst, coupling, and channel track.

Table 3.60 Range of Continuous Waveform Frequency

Waveform	DG821 Pro	DG822 Pro	DG852 Pro
Sine	1 μHz to 25 MHz	1 μHz to 25 MHz	1 μHz to 50 MHz
Square	1 μHz to 20 MHz	1 μHz to 20 MHz	1 μHz to 40 MHz
Ramp	1 μHz to 1 MHz	1 μHz to 1 MHz	1 μHz to 1 MHz
Pulse	1 μHz to 10 MHz	1 μHz to 10 MHz	1 μHz to 25 MHz
Arb	1 μHz to 10 MHz	1 μHz to 10 MHz	1 μHz to 15 MHz
Harmonic	1 mHz to 10 MHz	1 mHz to 10 MHz	1 mHz to 25 MHz

Waveform	DG902 Pro	DG912 Pro	DG922 Pro
Sine	1 μHz to 70 MHz	1 μHz to 150 MHz	1 μHz to 200 MHz
Square	1 μHz to 60 MHz	1 μHz to 60 MHz	1 μHz to 60 MHz
Ramp	1 μHz to 3 MHz	1 μHz to 5 MHz	1 μHz to 5 MHz
Pulse	1 μHz to 50 MHz	1 μHz to 50 MHz	1 μHz to 50 MHz
Arb	1 μHz to 30 MHz	1 μHz to 50 MHz	1 μHz to 50 MHz
Harmonic	1 mHz to 35 MHz	1 mHz to 75 MHz	1 mHz to 100 MHz

Table 3.61 Range of Continuous Waveform Period

Waveform	DG821 Pro	DG822 Pro	DG852 Pro
Sine	40 ns to 1 Ms	40 ns to 1 Ms	20 ns to 1 Ms
Square	50 ns to 1 Ms	50 ns to 1 Ms	25 ns to 1 Ms
Ramp	1 μs to 1 Ms	1 μs to 1 Ms	1 μs to 1 Ms
Pulse	100 ns to 1 Ms	100 ns to 1 Ms	40 ns to 1 Ms
Arb	100 ns to 1 Ms	100 ns to 1 Ms	66.6 ns to 1 Ms
Harmonic	100 ns to 1 ks	100 ns to 1 ks	40 ns to 1 ks

Waveform	DG902 Pro	DG912 Pro	DG922 Pro
Sine	14.3 ns to 1 Ms	6.7 ns to 1 Ms	5 ns to 1 Ms
Square	16.6 ns to 1 Ms	16.6 ns to 1 Ms	16.6 ns to 1 Ms
Ramp	333.3 ns to 1 Ms	200 ns to 1 Ms	200 ns to 1 Ms
Pulse	20 ns to 1 Ms	20 ns to 1 Ms	20 ns to 1 Ms
Arb	33.3 ns to 1 Ms	20 ns to 1 Ms	20 ns to 1 Ms
Harmonic	28.6 ns to 1 ks	13.3 ns to 1 ks	10 ns to 1 ks

Table 3.62 Range of Amplitude

Frequency	HighZ	Load (50 Ω)
[1 μHz, 50 MHz]	2 mVpp to 20 Vpp	1 mVpp to 10 Vpp
(50 MHz, 100 MHz]	2 mVpp to 10 Vpp	1 mVpp to 5 Vpp
(100 MHz, 200 MHz]	2 mVpp to 4 Vpp	1 mVpp to 2 Vpp

TIP

The noise amplitude range is related to the impedance setting:

- HighZ: 2 mVpp to 4 Vpp.
- Load (50 Ω): 1 mVpp to 2 Vpp.

3.11.1 [:SOURce[<n>]]:AM

[:SOURce[<n>]]:AM commands are used to set and query the AM parameters such as the AM modulation source, modulating waveform, modulation depth, modulating waveform frequency, and the on/off status of the DSSC function and the AM modulation function.

3.11.1.1 [:SOURce[<n>]]:AM:DEPTh

Syntax

[:SOURce[<*n*>]]:AM[:DEPTh] {<*percent*>|<*lim*>}

[:SOURce[<*n*>]]:AM[:DEPTh]? [<*lim*>]

Description

Sets or queries the AM modulation depth for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<percent>	Real	0% to 120%	100%
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- Modulation depth is a percentage that represents the amplitude variation. At 0% depth, the amplitude is one-half of the carrier's amplitude setting. At 100% depth, the amplitude is identical to the carrier's amplitude setting. At greater than 100% depth, the modulation depth upper limit is limited by the peak on the output.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the AM modulation depth in scientific notation. For example, the query might return +5.000000000000000E+01, indicating that the modulation depth is 50%.

Examples

```
:SOURce1:AM:DEPTh 50 /*Sets the AM modulation depth to 50% for
CH1.*/
:SOURce1:AM:DEPTh? /*Queries the AM modulation depth for CH1. The
query returns +5.000000000000000E+01.*/
```

3.11.1.2 [:SOURce[<n>]]:AM:DSSC

Syntax

[:SOURce[<*n*>]]:AM:DSSC <*state*>

[:SOURce[<*n*>]]:AM:DSSC?

Description

Sets or queries the on/off status of the AM DSSC function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<state>	Discrete	{ON|1|OFF|0}	0|OFF

Remarks

When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:AM:DSSC ON /*Enables the AM DSSC function for CH1.*/
:SOURce1:AM:DSSC? /*Queries the on/off status of the AM DSSC
function for CH1. The query returns 1.*/
```

3.11.1.3 [:SOURce[<n>]]:AM:INTernal:FREQuency

Syntax

[:SOURce[<*n*>]]:AM:INTernal:FREQuency {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:AM:INTernal:FREQuency? [<*lim_query*>]

Description

Sets or queries the frequency of the AM modulating waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	2 mHz to 1 MHz	100 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-

Name	Type	Range	Default
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is only available for internal modulation source (*[:SOURce[<n>]]:AM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency in scientific notation. For example, the query might return +1.500000000000000E+02, indicating that the frequency is 150 Hz.

Examples

```
:SOURce1:AM:INTernal:FREQuency 150 /*Sets the frequency of the AM
modulating waveform to 150 Hz for CH1.*/
:SOURce1:AM:INTernal:FREQuency? /*Queries the frequency of the AM
modulating waveform for CH1. The query returns +1.500000000000000E
+02.*/
```

3.11.1.4 [:SOURce[<n>]]:AM:INTernal:FUNCtion

Syntax

[:SOURce[<*n*>]]:AM:INTernal:FUNCtion <*function*>

[:SOURce[<*n*>]]:AM:INTernal:FUNCtion?

Description

Sets or queries the AM modulating waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<function>	Discrete	{SINusoid|SQUare|TRIangle|RAMP|NRAMp|NOISe|ARB}	SINusoid

Remarks

- AM supports the following internal modulating waveforms:
  - **SINusoid:** Sine wave.
  - **SQUare:** Square with 50% duty cycle.
  - **TRIangle:** Triangle with 50% symmetry.

- **RAMP:** UpRamp with 100% symmetry.
- **NRAMp:** DnRamp with 0% symmetry.
- **NOISe:** white gaussian noise.
- **ARB:** arbitrary waveform.

- This command is only available for internal modulation source (*[:SOURce[<n>]]:AM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns SIN, SQU, TRI, RAMP, NRAM, NOIS, or ARB.

Examples

```
:SOURce1:AM:INTernal:FUNCtion SQUare /*Sets the AM modulating
waveform to Square for CH1.*/
:SOURce1:AM:INTernal:FUNCtion? /*Queries the AM modulating
waveform for CH1. The query returns SQU.*/
```

3.11.1.5 [:SOURce[<n>]]:AM:SOURce

Syntax

[:SOURce[<*n*>]]:AM:SOURce <*source*>

[:SOURce[<*n*>]]:AM:SOURce?

Description

Sets or queries the AM modulation source for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- **INTernal:** internal modulation source. When the internal modulation source is selected, you can send *[:SOURce[<n>]]:AM:INTernal:FUNCtion* to select the internal modulating waveforms.
- **EXTernal:** external modulation source. When the external modulation source is selected, the generator receives the external modulating signal from the rear-panel **[AUX IN/OUT]** connector.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT or EXT.

Examples

```
:SOURce1:AM:SOURce INTernal /*Sets the AM modulation source to
internal modulation source for CH1.*/
:SOURce1:AM:SOURce? /*Queries the AM modulation source for CH1.
The query returns INT.*/
```

3.11.1.6 [:SOURce[<n>]]:AM:STATe

Syntax

[:SOURce[<*n*>]]:AM:STATe <*bool*>

[:SOURce[<*n*>]]:AM:STATe?

Description

Sets or queries the on/off status of the AM modulation function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	OFF

Remarks

- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the modulation function is enabled.
- The harmonic, noise, DC, and pulse cannot be modulated for AM.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:AM:STATe ON /*Enables the AM modulation function for
CH1.*/
:SOURce1:AM:STATe? /*Queries the on/off status of the AM
modulation function for CH1. The query returns 1.*/
```

3.11.2 [:SOURce[<n>]]:APPLy

3.11.2.1 [:SOURce[<n>]]:APPLy?

Syntax

[:SOURce[<*n*>]]:APPLy?

Description

Queries the waveform parameters for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Remarks

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.
- The table below shows the channel waveform types and the corresponding returned waveform names.

Sine	Square	Ramp	Pulse	Noise	DC	Harmonic	Arb	Sequence
SIN	SQU	RAMP	PULS	NOIS	DC	HARM	ARB	SEQ

Return Format

The query returns a string enclosed by double quotation marks. The returned value consists of 5 parts separated by commas. The format is "waveform name,frequency,amplitude,offset,phase" (expressed in scientific notation and the default units are Hz, Vpp, Vdc and ° respectively; the absent item is replaced by DEF). For example, the query might return "SIN,+5.000000000000000E +03,+3.0000000000000E+00,-3.0000000000000E+00,+4.0000000000000E+00", indicating that the current waveform is Sine wave, the frequency is 5 kHz, the amplitude is 3 Vpp, the offset is -3 Vdc, and the start phase is 4°.

Examples

```
:SOURce1:APPLy? /*Queries the waveform type, frequency, amplitude,
offset, and phase for CH1. The query returns "SIN,
+5.000000000000000E+03,+3.0000000000000E+00,-3.0000000000000E
+00,+4.0000000000000E+00".*/
```

3.11.2.2 [:SOURce[<n>]]:APPLy:ARBitrary

Syntax

[:SOURce[<*n*>]]:APPLy:ARBitrary [{<*frequency*>|<*lim_set*>}[,{<*amplitude*>|<*lim_set*>}[,{<*offset*>|<*lim_set*>}[,{<*phase*>|<*lim_set*>}]]]]

Description

Sets the specified channel to output an arbitrary waveform with the specified frequency, amplitude, offset, and phase.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<lim_set>	Discrete	{DEFault|MINimum|MAXimum}	-
<frequency>	Real	Refer to *Remarks*	1 kHz
<amplitude>	Real	Refer to *Remarks*	5 Vpp
<offset>	Real	Refer to *Remarks*	0 Vdc
<phase>	Real	-360° to 360°	0°

Remarks

- The range of <frequency> is related to the instrument model (*Table 3.60 Range of Continuous Waveform Frequency*). For the range of <amplitude>, refer to *Table 3.62 Range of Amplitude*. The range of <offset> is limited by the current amplitude setting.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:APPLy:ARBitrary 100,1,2,1 /*Sets CH1 to output an
arbitrary waveform with 100 Hz frequency, 1 Vpp amplitude, 2 Vdc
offset, and 1° start phase.*/
```

3.11.2.3 [:SOURce[<n>]]:APPLy:DC

Syntax

[**:SOURce**[<*n*>]]**:APPLy:DC** [{<*frequency*>|<*lim_set*>}[,{<*amplitude*>|<*lim_set*>}[,{<*offset*>|<*lim_set*>}[,{<*phase*>|<*lim_set*>}]]]]

Description

Sets the specified channel to output a DC with a specified offset.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<lim_set>	Discrete	{DEFault|MINimum|MAXimum}	-
<frequency>	Real	Refer to *Remarks*	-
<amplitude>	Real	Refer to *Remarks*	-
<offset>	Real	Refer to *Remarks*	0 Vdc
<phase>	Real	Refer to *Remarks*	-

Remarks

- <frequency>, <phase>, and <amplitude> are not applicable to DC. However, a placeholder must be specified for them.
- The range of <offset> is limited by the "Impedance" setting (*:OUTPut[<n>]:LOAD*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:APPLy:DC 100,5,1,90 /*Sets CH1 to output a DC with 1 Vdc
offset.*/
```

3.11.2.4 [:SOURce[<n>]]:APPLy:NOISe

Syntax

[**:SOURce**[<*n*>]]**:APPLy:NOISe** [{<*frequency*>|<*lim_set*>}[,{<*amplitude*>|<*lim_set*>}[,{<*offset*>|<*lim_set*>}[,{<*phase*>|<*lim_set*>}]]]]

Description

Sets the specified channel to output noise with the specified amplitude and offset.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<lim_set>	Discrete	{DEFault|MINimum|MAXimum}	-

Name	Type	Range	Default
<frequency>	Real	Refer to *Remarks*	-
<amplitude>	Real	Refer to *Remarks*	5 Vpp
<offset>	Real	Refer to *Remarks*	0 Vdc
<phase>	Real	Refer to *Remarks*	-

Remarks

- <frequency> and <phase>are not applicable to noise. However, a placeholder must be specified for them.
- For the range of <amplitude>, refer to *Table 3.62 Range of Amplitude* . The range of <offset> is limited by the current amplitude setting.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:APPLy:NOISe 100,2,1,90 /*Sets CH1 to output noise with 2
Vpp amplitude and 1 Vdc offset.*/
```

3.11.2.5 [:SOURce[<n>]]:APPLy:PULSe

Syntax

[:SOURce[<*n*>]]:APPLy:PULSe [{<*frequency*>|<*lim_set*>}[,{<*amplitude*>|<*lim_set*>}[,{<*offset*>|<*lim_set*>}[,{<*phase*>|<*lim_set*>}]]]]

Description

Sets the specified channel to output a pulse with the specified frequency, amplitude, offset, and phase.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<lim_set>	Discrete	{DEFault|MINimum|MAXimum}	-
<frequency>	Real	Refer to *Remarks*	1 kHz
<amplitude>	Real	Refer to *Remarks*	5 Vpp

Name	Type	Range	Default
<offset>	Real	Refer to *Remarks*	0 Vdc
<phase>	Real	-360° to 360°	0°

Remarks

- The range of <frequency> is related to the instrument model (*Table 3.60 Range of Continuous Waveform Frequency*). For the range of <amplitude>, refer to *Table 3.62 Range of Amplitude*. The range of <offset> is limited by the current amplitude setting.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:APPLy:PULSe 100,3,2,1 /*Sets CH1 to output a pulse with
100 Hz frequency, 3 Vpp amplitude, 2 Vdc offset, and 1° start
phase.*/
```

3.11.2.6 [:SOURce[<n>]]:APPLy:RAMP

Syntax

[:**SOURce**[<*n*>]]:**APPLy**:**RAMP** [{<*frequency*>|<*lim_set*>}[,{<*amplitude*>|<*lim_set*>}[,{<*offset*>|<*lim_set*>}[,{<*phase*>|<*lim_set*>}]]]]

Description

Sets the specified channel to output a ramp with the specified frequency, amplitude, offset, and phase.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<lim_set>	Discrete	{DEFault|MINimum|MAXimum}	-
<frequency>	Real	Refer to *Remarks*	1 kHz
<amplitude>	Real	Refer to *Remarks*	5 Vpp
<offset>	Real	Refer to *Remarks*	0 Vdc
<phase>	Real	-360° to 360°	0°

Remarks

- The range of <frequency> is related to the instrument model (*Table 3.60 Range of Continuous Waveform Frequency*). For the range of <amplitude>, refer to *Table 3.62 Range of Amplitude*. The range of <offset> is limited by the current amplitude setting.
- Executing this command will overwrite the current symmetry setting (*[:SOURce[<n>]]:FUNCtion:RAMP:SYMMetry*) and set 50% symmetry for the ramp.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:APPLy:RAMP 100,1,2,3 /*Sets CH1 to output a ramp with 100
Hz frequency, 1 Vpp amplitude, 2 Vdc offset, and 3° start phase.*/
```

3.11.2.7 [:SOURce[<n>]]:APPLy:SINusoid

Syntax

[:SOURce[<*n*>]]:APPLy:SINusoid [{<*frequency*>|<*lim_set*>}[,{<*amplitude*>|<*lim_set*>}[,{<*offset*>|<*lim_set*>}[,{<*phase*>|<*lim_set*>}]]]]

Description

Sets the specified channel to output a sine wave with the specified frequency, amplitude, offset, and phase.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<lim_set>	Discrete	{DEFault|MINimum|MAXimum}	-
<frequency>	Real	Refer to *Remarks*	1 kHz
<amplitude>	Real	Refer to *Remarks*	5 Vpp
<offset>	Real	Refer to *Remarks*	0 Vdc
<phase>	Real	-360° to 360°	0°

Remarks

- The range of <frequency> is related to the instrument model (*Table 3.60 Range of Continuous Waveform Frequency*). For the range of <amplitude>, refer to *Table 3.62 Range of Amplitude*. The range of <offset> is limited by the current amplitude setting.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:APPLy:SINusoid 100,3,2,1 /*Sets CH1 to output a sine wave with 100 Hz frequency, 3 Vpp amplitude, 2 Vdc offset, and 1° start phase.*/
```

3.11.2.8 [:SOURce[<n>]]:APPLy:SQUare

Syntax

[:SOURce[<*n*>]]:APPLy:SQUare [{<*frequency*>|<*lim_set*>}[,{<*amplitude*>|<*lim_set*>}[,{<*offset*>|<*lim_set*>}[,{<*phase*>|<*lim_set*>}]]]]

Description

Sets the specified channel to output a square wave with the specified frequency, amplitude, offset, and phase.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<lim_set>	Discrete	{DEFault|MINimum|MAXimum}	-
<frequency>	Real	Refer to *Remarks*	1 kHz
<amplitude>	Real	Refer to *Remarks*	5 Vpp
<offset>	Real	Refer to *Remarks*	0 Vdc
<phase>	Real	-360° to 360°	0°

Remarks

- The range of <frequency> is related to the instrument model (*Table 3.60 Range of Continuous Waveform Frequency*). For the range of <amplitude>, refer to *Table 3.62 Range of Amplitude*. The range of <offset> is limited by the current amplitude setting.

- Executing this command will overwrite the current duty cycle setting (*[:SOURce[<n>]]:FUNCtion:SQUare:DCYCle*) and set 50% duty cycle for the square wave.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:APPLy:SQUare 100,5,0.5,3 /*Sets CH1 to output a square
wave with 100 Hz frequency, 5 Vpp amplitude, 0.5 Vdc offset, and
3° start phase.*/
```

3.11.3 [:SOURce[<n>]]:ASKey

[:SOURce[<n>]]:ASKey commands are used to set and query ASK parameters such as the modulation amplitude, modulation rate, modulation polarity, modulation source, and the on/off status of the ASK modulation function.

3.11.3.1 [:SOURce[<n>]]:ASKey:AMPLitude

Syntax

[**:SOURce**[<*n*>]]**:ASKey:AMPLitude** {<*amplitude*>|<*lim_set*>}

[**:SOURce**[<*n*>]]**:ASKey:AMPLitude?** [<*lim_query*>]

Description

Sets or queries the ASK modulation amplitude for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<amplitude>	Real	Refer to *Remarks*	2 Vpp
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- In ASK modulation, the instrument shift its output amplitude between two preset values (called the carrier amplitude and the modulation amplitude).
- The range of <amplitude> is consistent with the amplitude range of basic waveforms (*Table 3.62 Range of Amplitude*).

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the ASK modulation amplitude in scientific notation. For example, the query might return +1.000000000000000E+00, indicating that the modulation amplitude is 1 Vpp.

Examples

```
:SOURce1:ASKey:AMPLitude 1 /*Sets the ASK modulation amplitude to
1 Vpp for CH1.*/
:SOURce1:ASKey:AMPLitude? /*Queries the ASK modulation amplitude
for CH1. The query returns +1.000000000000000E+00.*/
```

3.11.3.2 [:SOURce[<n>]]:ASKey:INTernal:RATE

Syntax

[:SOURce[<*n*>]]:ASKey:INTernal:RATE {<*rate*>|<*lim*>}

[:SOURce[<*n*>]]:ASKey:INTernal:RATE? [<*lim*>]

Description

Sets or queries the ASK modulation rate for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<rate>	Real	2 mHz to 1 MHz	100 Hz
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is only available for internal modulation source (*[:SOURce[<n>]]:ASKey:SOURce*).
- ASK rate is the rate at which the output amplitude "shifts" between the carrier amplitude (*[:SOURce[<n>]]:VOLTage*) and modulation amplitude (*[:SOURce[<n>]]:ASKey:AMPLitude*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the ASK rate in scientific notation. For example, the query might return +1.500000000000000E+02, indicating that the ASK rate is 150 Hz.

Examples

```
:SOURce1:ASKey:INTernal:RATE 150 /*Sets the ASK rate to 150 Hz for
CH1.*/
:SOURce1:ASKey:INTernal:RATE? /*Queries the ASK rate for CH1. The
query returns +1.500000000000000E+02.*/
```

3.11.3.3 [:SOURce[<n>]]:ASKey:POLarity

Syntax

[:SOURce[<*n*>]]:ASKey:POLarity <*polarity*>

[:SOURce[<*n*>]]:ASKey:POLarity?

Description

Sets or queries the ASK modulation polarity for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<polarity>	Discrete	{POSitive|NEGative}	POSitive

Remarks

- The ASK modulation polarity can be set to the following:
  - **POSitive:** positive polarity.
  - **NEGative:** negative polarity.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns POS or NEG.

Examples

```
:SOURce1:ASKey:POLarity NEGative /*Sets the ASK modulation
polarity to negative for CH1.*/
:SOURce1:ASKey:POLarity? /*Queries the ASK modulation polarity for
CH1. The query returns NEG.*/
```

3.11.3.4 [:SOURce[<n>]]:ASKey:SOURce

Syntax

[:SOURce[<*n*>]]:ASKey:SOURce <*source*>

[:SOURce[<*n*>]]:ASKey:SOURce?

Description

Sets or queries the ASK modulation source for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- The instrument can receive modulating waveforms from internal or external modulation source.
  - **INTernal:** internal modulation source. The modulating waveform is a square waveform with 50% duty cycle.
  - **EXTernal:** when the external modulation source is selected, the generator receives the external modulating signal from the rear-panel **[AUX IN/OUT]** connector.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT or EXT.

Examples

```
:SOURce1:ASKey:SOURce INTernal /*Sets the ASK modulation source to
internal modulation source for CH1.*/
:SOURce1:ASKey:SOURce? /*Queries the ASK modulation source for
CH1. The query returns INT.*/
```

3.11.3.5 [:SOURce[<n>]]:ASKey:STATe

Syntax

[:SOURce[<*n*>]]:ASKey:STATe <*bool*>

[:SOURce[<*n*>]]:ASKey:STATe?

Description

Sets or queries the on/off status of the ASK modulation function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

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

Remarks

- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the modulation function is enabled.
- The harmonic, noise, DC, and pulse cannot be modulated for ASK.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:ASKey:STATe ON /*Enables the ASK modulation function for
CH1.*/
:SOURce1:ASKey:STATe? /*Queries the on/off status of the ASK
modulation function for CH1. The query returns 1.*/
```

3.11.4 [:SOURce[<n>]]:BURSt

[:SOURce[<n>]]:BURSt commands are used to set the burst parameters such as burst type, gate polarity, burst count, start phase, and internal trigger period.

3.11.4.1 [:SOURce[<n>]]:BURSt:GATE:POLarity

Syntax

[:SOURce[<*n*>]]:BURSt:GATE:POLarity <*polarity*>

[:SOURce[<*n*>]]:BURSt:GATE:POLarity?

Description

Sets or queries the gate polarity of the burst waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<polarity>	Discrete	{NORMal|INVerted}	NORMal

Remarks

- The gate polarity is only available for the gated burst mode (*[:SOURce[<n>]]:BURSt:MODE*). The generator controls the burst output based on the external signal levels (gated signals) at the rear-panel **[AUX IN/OUT]** connector of the corresponding channel.
- The gate polarity can be set to the following:
  - **NORMal:** positive polarity. When the external signal level is high (low), the gate signal is true (false).
  - **INVerted:** negative polarity. When the external signal level is low (high), the gate signal is true (false).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns NORM or INV.

Examples

```
:SOURce1:BURSt:GATE:POLarity NORMal /*Sets the gate polarity of
the burst waveform to positive for CH1.*/
:SOURce1:BURSt:GATE:POLarity? /*Queries the gate polarity of the
burst waveform for CH1. The query returns NORM.*/
```

3.11.4.2 [:SOURce[<n>]]:BURSt:INTernal:PERiod

Syntax

[:SOURce[<*n*>]]:BURSt:INTernal:PERiod {<*seconds*>|<*lim*>}

[:SOURce[<*n*>]]:BURSt:INTernal:PERiod? [<*lim*>]

Description

Sets or queries the burst period of internally-triggered N-Cycle bursts for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	4 µs to 8000 s	10 ms
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- Burst period is only available for internally triggered N-Cycle burst mode.

- For the relations among the burst period, basic waveform period, and burst count, the following formula should be satisfied:

  Burst Period ≥ ⌈(Burst Count x Waveform Period) ÷ 6.4 ns⌉ x 6.4 ns + 4 μs[1]

- If the burst period is too short, the generator will increase it automatically to ensure the output of the specified number of cycles.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

NOTE

[1]: ⌈x⌉ indicates that x is rounded up.

Return Format

The query returns the burst period in scientific notation. For example, the query might return +1.000000000000000E-01, indicating that the period is 0.1 s.

Examples

```
:SOURce1:BURSt:INTernal:PERiod 0.1 /*Sets the burst period of
internally-triggered N-Cycle bursts to 0.1 s for CH1.*/
:SOURce1:BURSt:INTernal:PERiod? /*Queries the burst period of
internally-triggered N-Cycle bursts for CH1. The query returns
+1.000000000000000E-01.*/
```

3.11.4.3 [:SOURce[<n>]]:BURSt:MODE

Syntax

[:SOURce[<*n*>]]:BURSt:MODE <*mode*>

[:SOURce[<*n*>]]:BURSt:MODE?

Description

Sets or queries the burst type for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<mode>	Discrete	{TRIGgered|GATed}	TRIGgered

Remarks

- When the output mode of the specified channel is set to Burst, you can use this command to set the burst type.
  - **TRIGgered:** N-Cycle burst. The generator outputs a waveform with the specified number of cycles (burst count) when it receives the trigger signal. You can use *[:SOURce[<n>]]:BURSt:NCYCles* to set the burst count.

- **GATed:** Gated burst. The generator controls the waveform output based on the external signal levels at the rear-panel **[AUX IN/OUT]** connector. You can use *[:SOURce[<n>]]:BURSt:GATE:POLarity* to select the signal's polarity.

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns TRIG or GAT.

Examples

```
:SOURce1:BURSt:MODE GATed /*Sets the burst type to Gated for CH1.*/
:SOURce1:BURSt:MODE? /*Queries the burst type for CH1. The query
returns GAT.*/
```

3.11.4.4 [:SOURce[<n>]]:BURSt:NCYCles

Syntax

[**:SOURce**[<*n*>]]**:BURSt:NCYCles** {<*cycles*>|<*lim_set*>}

[**:SOURce**[<*n*>]]**:BURSt:NCYCles?** [<*lim_query*>]

Description

Sets or queries the number of cycles (burst count) of the N-Cycle bursts for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<cycles>	Integer	1 to 1,000,000	1
<lim_set>	Discrete	{INFinity|MINimum|MAXimum}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- In N-Cycle Burst mode (*[:SOURce[<n>]]:BURSt:MODE*), the instrument outputs a waveform for a specified number of cycles (burst count) each time trigger signal is received. You can also set the burst count to "Infinity".
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the burst count in scientific notation. For example, the query might return +5.000000000000000E+02, indicating that the burst count is 500.

Examples

```
:SOURce1:BURSt:NCYCles 500 /*Sets the burst count of the N-Cycle
bursts to 500.*/
:SOURce1:BURSt:NCYCles? /*Queries the burst count of the N-Cycle
bursts for CH1. The query returns +5.000000000000000E+02.*/
```

3.11.4.5 [:SOURce[<n>]]:BURSt:PHASe

Syntax

[:SOURce[<*n*>]]:BURSt:PHASe {<*phase*>|<*lim*>}

[:SOURce[<*n*>]]:BURSt:PHASe? [<*lim*>]

Description

Sets or queries the start phase of the burst waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<phase>	Real	-360° to 360°	0°
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the start phase in scientific notation. For example, the query might return +1.000000000000000E+01, indicating that the start phase is 10°.

Examples

```
:SOURce1:BURSt:PHASe 10 /*Sets the start phase of the burst
waveform to 10° for CH1.*/
:SOURce1:BURSt:PHASe? /*Queries the start phase of the burst
waveform for CH1. The query returns +1.000000000000000E+01.*/
```

3.11.4.6 [:SOURce[<n>]]:BURSt:STATe

Syntax

[:SOURce[<*n*>]]:BURSt:STATe <*bool*>

[:SOURce[<*n*>]]:BURSt:STATe?

Description

Sets or queries the on/off status of the burst mode for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- After the burst mode is enabled, the modulation, sweep, or sequence mode (if enabled currently) will be disabled automatically.
- The Burst mode cannot be enabled when the basic waveform frequency is less than or equal to 125 μHz.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:BURSt:STATe ON /*Enables the burst mode for CH1.*/
:SOURce1:BURSt:STATe? /*Queries whether the burst mode is enabled
for CH1. The query returns 1.*/
```

3.11.5 [:SOURce[<n>]]:FM

[:SOURce[<n>]]:FM commands are used to set and query the FM parameters such as the frequency deviation, modulating waveform frequency, modulation source, modulating waveform, and the on/off status of the FM modulation.

3.11.5.1 [:SOURce[<n>]]:FM[:DEViation]

Syntax

[:SOURce[<*n*>]]:FM[:DEViation] {<*deviation*>|<*lim_set*>}

[:SOURce[<*n*>]]:FM[:DEViation]? [<*lim_query*>]

Description

Sets or queries the FM frequency deviation for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<deviation>	Real	Refer to *Remarks*	100 Hz

Name	Type	Range	Default
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Frequency deviation represents the peak variation in frequency of the modulated waveform from the carrier frequency. The minimum frequency deviation is 0 Hz and the maximum value is limited by the frequency setting value (*[:SOURce[<n>]]:FREQuency*) and the carrier frequency upper limit:
  - Frequency Deviation ≤ Current Carrier Frequency Value - 1 μHz.
  - Frequency Deviation ≤ Carrier Frequency Upper Limit - Carrier Frequency Value.
- When an external modulation source (*[:SOURce[<n>]]:FM:SOURce*) is selected, the frequency deviation is controlled by the signal from the real-panel **[AUX IN/OUT]** connector.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency deviation in scientific notation. For example, the query might return +1.000000000000000E+02, indicating that the frequency deviation is 100 Hz.

Examples

```
:SOURce1:FM:DEViation 100 /*Sets the FM frequency deviation to 100
Hz for CH1.*/
:SOURce1:FM:DEViation? /*Queries the FM frequency deviation for
CH1. The query returns +1.000000000000000E+02.*/
```

3.11.5.2 [:SOURce[<n>]]:FM:INTernal:FREQuency

Syntax

[**:SOURce**[<*n*>]]**:FM:INTernal:FREQuency** {<*frequency*>|<*lim_set*>}

[**:SOURce**[<*n*>]]**:FM:INTernal:FREQuency?** [<*lim_query*>]

Description

Sets or queries the FM modulation frequency for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Name	Type	Range	Default
<frequency>	Real	2 mHz to 1 MHz	100 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is only available for internal modulation source (*[:SOURce[<n>]]:FM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency in scientific notation. For example, the query might return +1.500000000000000E+02, indicating that the frequency is 150 Hz.

Examples

```
:SOURce1:FM:INTernal:FREQuency 150 /*Sets the frequency of the FM
modulating waveform to 150 Hz for CH1.*/
:SOURce1:FM:INTernal:FREQuency? /*Queries the frequency of the FM
modulating waveform for CH1. The query returns +1.500000000000000E
+02.*/
```

3.11.5.3 [:SOURce[<n>]]:FM:INTernal:FUNCtion

Syntax

[**:SOURce**[<*n*>]]**:FM:INTernal:FUNCtion** <*function*>

[**:SOURce**[<*n*>]]**:FM:INTernal:FUNCtion?**

Description

Sets or queries the FM modulating waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<function>	Discrete	{SINusoid|SQUare|TRIangle|RAMP|NRAMp|NOISe|ARB}	SINusoid

Remarks

- FM supports the following internal modulating waveforms:
  - **SINusoid:** Sine wave.

- **SQUare:** Square with 50% duty cycle.
- **TRIangle:** Triangle with 50% symmetry.
- **RAMP:** UpRamp with 100% symmetry.
- **NRAMp:** DnRamp with 0% symmetry.
- **NOISe:** white gaussian noise.
- **ARB:** arbitrary waveform.

- This command is only available for internal modulation source (*[:SOURce[<n>]]:FM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns SIN, SQU, TRI, RAMP, NRAM, NOIS, or ARB.

Examples

```
:SOURce1:FM:INTernal:FUNCtion SQUare /*Sets the FM modulating
waveform to Square for CH1.*/
:SOURce1:FM:INTernal:FUNCtion? /*Queries the FM modulating
waveform for CH1. The query returns SQU.*/
```

3.11.5.4 [:SOURce[<n>]]:FM:SOURce

Syntax

[:SOURce[<*n*>]]:FM:SOURce <*source*>

[:SOURce[<*n*>]]:FM:SOURce?

Description

Sets or queries the FM modulation source for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- The instrument can receive modulating waveforms from internal or external modulation source.

- **INTernal:** internal modulation source. When the internal modulation source is selected, you can use *[:SOURce[<n>]]:FM:INTernal:FUNCtion* to select the internal modulating waveforms.
- **EXTernal:** when the external modulation source is selected, the generator receives the external modulating signal from the rear-panel **[AUX IN/OUT]** connector.

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT or EXT.

Examples

```
:SOURce1:FM:SOURce INTernal /*Sets the FM modulation source to
internal modulation source for CH1.*/
:SOURce1:FM:SOURce? /*Queries the FM modulation source for CH1.
The query returns INT.*/
```

3.11.5.5 [:SOURce[<n>]]:FM:STATe

Syntax

[:SOURce[<*n*>]]:FM:STATe <*bool*>

[:SOURce[<*n*>]]:FM:STATe?

Description

Sets or queries the on/off status of the FM modulation function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the modulation function is enabled.
- The harmonic, noise, DC, and pulse cannot be modulated for FM.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 1 or 0.

Examples

```
:SOURce1:FM:STATe ON /*Enables the FM modulation function for
CH1.*/
:SOURce1:FM:STATe? /*Queries the on/off status of the FM
modulation function for CH1. The query returns 1.*/
```

3.11.6 [:SOURce[<n>]]:FREQuency

[:SOURce[<n>]]:FREQuency commands are used to set the instrument's output frequency and dual-channel coupling mode.

3.11.6.1 [:SOURce[<n>]]:FREQuency

Syntax

[:SOURce[<*n*>]]:**FREQuency** {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:**FREQuency?** [<*lim_query*>]

Description

Sets or queries the frequency of the continuous waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	Refer to *Remarks*	1 kHz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The available frequency range varies for different waveforms. For this, refer to *Table 3.60 Range of Continuous Waveform Frequency* .
- When the wave type of the specified channel is changed, the instrument still uses the frequency if the frequency is valid. Otherwise, the instrument automatically sets the frequency as the upper limit for the new wave type.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency in scientific notation. For example, the query might return +1.000000000000000E+06, indicating that the frequency is 1 MHz.

Examples

```
:SOURce1:FREQuency 1000000 /*Sets the frequency of the continuous
waveform to 1 MHz for CH1.*/
:SOURce1:FREQuency? /*Queries the frequency of the continuous
waveform for CH1. The query returns +1.000000000000000E+06.*/
```

3.11.6.2 [:SOURce[<n>]]:FREQuency:CENTer

Syntax

[:SOURce[<*n*>]]:FREQuency:CENTer {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:FREQuency:CENTer? [<*lim_query*>]

Description

Sets or queries the center frequency of the sweep mode for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	Refer to *Remarks*	550 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- You can set the sweep boundaries using the center frequency and frequency span (*[:SOURce[<n>]]:FREQuency:SPAN*). Different sweep waveforms have different center frequency and frequency span ranges. Also, the center frequency and frequency span affect each other.
- Center Frequency = (Start Frequency + Stop Frequency)/2. Frequency Span = Stop Frequency - Start Frequency.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the center frequency in scientific notation. For example, the query might return +5.000000000000000E+03, indicating that the center frequency is 5 kHz.

Examples

```
:SOURce1:FREQuency:CENTer 5000 /*Sets the center frequency of the
sweep mode to 5 kHz for CH1.*/
:SOURce1:FREQuency:CENTer? /*Queries the center frequency of the
sweep mode for CH1. The query returns +5.000000000000000E+03.*/
```

3.11.6.3 [:SOURce[<n>]]:FREQuency:COUPle:MODE

Syntax

[:SOURce[<*n*>]]:FREQuency:COUPle:MODE <*mode*>

[:SOURce[<*n*>]]:FREQuency:COUPle:MODE?

Description

Sets or queries the type of the frequency coupling mode for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<mode>	Discrete	{OFFSet|RATio}	RATio

Remarks

- The frequency coupling mode can be set to the following:
  - **OFFSet:** frequency deviation mode. You can use *[:SOURce[<n>]]:FREQuency:COUPle:OFFSet* to set the frequency deviation of the coupled channel.
  - **RATio:** frequency ratio mode. You can use *[:SOURce[<n>]]:FREQuency:COUPle:RATio* to set the frequency ratio of the coupled channel.
- When the coupling is enabled, if switching to ratio coupling from deviation coupling causes either channel to exceed the frequency upper/lower limit after calculation according to coupling rules, the generator will automatically adjust the parameters to avoid parameter overlimit. If the parameter overlimit is caused by switching to deviation coupling from ratio coupling, the frequency coupling will be automatically disabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns OFFS or RAT.

Examples

```
:SOURce1:FREQuency:COUPle:MODE OFFSet /*Sets the type of the
frequency coupling mode to frequency deviation for CH1.*/
:SOURce1:FREQuency:COUPle:MODE? /*Queries the type of the
frequency coupling mode for CH1. The query returns OFFS.*/
```

3.11.6.4 [:SOURce[<n>]]:FREQuency:COUPle:OFFSet

Syntax

[:SOURce[<*n*>]]:FREQuency:COUPle:OFFSet <*offset*>

[:SOURce[<*n*>]]:FREQuency:COUPle:OFFSet?

Description

Sets or queries the frequency deviation of the frequency coupling for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<offset>	Real	Refer to *Remarks*	0

Remarks

- When modifying the deviation, if base channel parameters exceed the limits of the coupled channel after coupling calculation, the generator will automatically adjust the coupling deviation to avoid parameter overlimit.
- The range of the frequency deviation is limited by the waveform type and instrument model.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency deviation in scientific notation. For example, the query might return +1.000000000000000E-02, indicating that the frequency deviation is 10 mHz.

Examples

```
:SOURce1:FREQuency:COUPle:OFFSet 0.01 /*Sets the frequency
deviation of the frequency coupling to 10 mHz for CH1.*/
:SOURce1:FREQuency:COUPle:OFFSet? /*Queries the frequency
deviation of the frequency coupling for CH1. The query returns
+1.000000000000000E-02.*/
```

3.11.6.5 [:SOURce[<n>]]:FREQuency:COUPle:RATio

Syntax

[:SOURce[<*n*>]]:FREQuency:COUPle:RATio <*ratio*>

[:SOURce[<*n*>]]:FREQuency:COUPle:RATio?

Description

Sets or queries the frequency ratio of the frequency coupling for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<ratio>	Real	0.001 to 1000	1

Remarks

- When modifying the ratio, if base channel parameters exceed the limits of the coupled channel after coupling calculation, the generator will automatically adjust the waveform frequency to avoid parameter overlimit.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency ratio in scientific notation. For example, the query might return 2.000000000000000E+00, indicating that the frequency ratio is 2.

Examples

```
:SOURce1:FREQuency:COUPle:RATio 2 /*Sets the frequency ratio of
the frequency coupling to 2 for CH1.*/
:SOURce1:FREQuency:COUPle:RATio? /*Queries the frequency ratio of
the frequency coupling for CH1. The query returns
2.000000000000000E+00.*/
```

3.11.6.6 [:SOURce[<n>]]:FREQuency:COUPle[:STATe]

Syntax

[:SOURce[<*n*>]]:FREQuency:COUPle[:STATe] <*bool*>

[:SOURce[<*n*>]]:FREQuency:COUPle[:STATe]?

Description

Sets or queries the on/off status of the frequency coupling function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- The frequency coupling function is not available for DG821 Pro (unless the two-channel upgrade option is installed).
- When the coupling mode is set to "ratio", if the frequency, amplitude, or phase of either channel exceeds the upper/lower limit after calculation according to the coupling rules due to modifying the coupling ratio, switching waveform, or changing waveform parameters, the generator will automatically adjust the waveform parameters to avoid parameter overlimit.
- When the coupling mode is set to "deviation", if the frequency, amplitude or phase of either channel exceeds the upper/lower limit after calculation according to the coupling rules due to modifying the coupling deviation, the generator will automatically adjust the coupling deviation to avoid parameter overlimit; if it is due to switching waveform or changing waveform parameters, the corresponding coupling function is disabled.
- The frequency coupling function is available only when the output modes of both channels are set to Continuous and the output waveform is set to Sine, Square, or Ramp.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:FREQuency:COUPle:STATe ON /*Enables the frequency
coupling function for CH1.*/
:SOURce1:FREQuency:COUPle:STATe? /*Queries the on/off status of
the frequency coupling function for CH1. The query returns 1.*/
```

3.11.6.7 [:SOURce[<n>]]:FREQuency:SPAN

Syntax

[:SOURce[<*n*>]]:FREQuency:SPAN {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:FREQuency:SPAN? [<*lim_query*>]

Description

Sets or queries the frequency span of the sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	Refer to *Remarks*	900 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- You can set the sweep boundaries using the center frequency (*[:SOURce[<n>]]:FREQuency:CENTer*) and frequency span. Different sweep waveforms have different center frequency and frequency span ranges. Also, the center frequency and frequency span affect each other. Define the maximum start/stop frequency as F_{max} and the minimum start/stop frequency as F_{min}. $F_m = (F_{max} - F_{min})/2$. The frequency span is affected by the center frequency: when Center Frequency ≤ F_m, the Frequency Span is ±2 x (Center Frequency-F_{min}); when Center Frequency > F_m, the Frequency Span is ±2 x (F_{max} - Center Frequency).
- Center Frequency = (Start Frequency + Stop Frequency)/2. Frequency Span = Stop Frequency - Start Frequency.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency span in scientific notation. For example, the query might return +8.000000000000000E+02, indicating that the frequency span is 800 Hz.

Examples

```
:SOURce1:FREQuency:SPAN 800 /*Sets the frequency span of the sweep
function to 800 Hz for CH1.*/
:SOURce1:FREQuency:SPAN? /*Queries the frequency span of the sweep
function for CH1. The query returns +8.000000000000000E+02.*/
```

3.11.6.8 [:SOURce[<n>]]:FREQuency:STARt

Syntax

[:SOURce[<*n*>]]:FREQuency:STARt {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:FREQuency:STARt? [<*lim_query*>]

Description

Sets or queries the start frequency of the sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	Refer to *Remarks*	100 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The start frequency and stop frequency (*[:SOURce[<n>]]:FREQuency:STOP*) set the sweep's upper and lower frequency bounds. The sweep begins at the start frequency, sweeps to the stop frequency, and then resets back to the start frequency. When the start frequency is less than the stop frequency, the generator sweeps up in frequency; when is start frequency is greater than the stop frequency, the generator sweeps down in frequency. When the start frequency is equal to the stop frequency, the generator sweeps in a fixed frequency.
- Center Frequency = (Start Frequency + Stop Frequency)/2. Frequency Span = Stop Frequency - Start Frequency.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the start frequency in scientific notation. For example, the query might return +1.000000000000000E+02, indicating that the start frequency is 100 Hz.

Examples

```
:SOURce1:FREQuency:STARt 100 /*Sets the start frequency of the
sweep function to 100 Hz for CH1.*/
:SOURce1:FREQuency:STARt? /*Queries the start frequency of the
sweep function for CH1. The query returns +1.000000000000000E+02.*/
```

3.11.6.9 [:SOURce[<n>]]:FREQuency:STOP

Syntax

[:SOURce[<*n*>]]:FREQuency:STOP {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:FREQuency:STOP? [<*lim_query*>]

Description

Sets or queries the stop frequency of the sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	Refer to *Remarks*	1 kHz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The start frequency (*[:SOURce[<n>]]:FREQuency:STARt*) and stop frequency set the sweep's upper and lower frequency bounds. The sweep begins at the start frequency, sweeps to the stop frequency, and then resets back to the start frequency. When the start frequency is less than the stop frequency, the generator sweeps up in frequency; when is start frequency is greater than the stop frequency, the generator sweeps down in frequency. When the start frequency is equal to the stop frequency, the generator sweeps in a fixed frequency.
- Center Frequency = (Start Frequency + Stop Frequency)/2. Frequency Span = Stop Frequency - Start Frequency.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the stop frequency in scientific notation. For example, the query might return +9.000000000000000E+02, indicating that the stop frequency is 900 Hz.

Examples

```
:SOURce1:FREQuency:STOP 900 /*Sets the stop frequency of the sweep function to 900 Hz for CH1.*/
:SOURce1:FREQuency:STOP? /*Queries the stop frequency of the sweep function for CH1. The query returns +9.000000000000000E+02.*/
```

3.11.7 [:SOURce[<n>]]:FSKey

[:SOURce[<n>]]:FSKey commands are used to set and query FSK parameters such as the FSK hopping frequency, modulation rate, modulation polarity, modulation source, and the on/off status of the FSK modulation function.

3.11.7.1 [:SOURce[<n>]]:FSKey:FREQuency

Syntax

[:SOURce[<*n*>]]:FSKey:FREQuency {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:FSKey:FREQuency? [<*lim_query*>]

Description

Sets or queries the FSK hop frequency for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	Refer to *Remarks*	10 kHz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- In FSK modulation, the generator "shift" its output frequency between the carrier frequency (*[:SOURce[<n>]]:FREQuency*) and hop frequency.
- The range of the hop frequency <frequency> is consistent with the range of the carrier frequency.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the hop frequency in scientific notation. For example, the query might return +5.000000000000000E+03, indicating that the frequency is 5 kHz.

Examples

```
:SOURce1:FSKey:FREQuency 5000 /*Sets the FSK hop frequency to 5
kHz for CH1.*/
:SOURce1:FSKey:FREQuency? /*Queries the FSK hop frequency for CH1.
The query returns +5.000000000000000E+03.*/
```

3.11.7.2 [:SOURce[<n>]]:FSKey:INTernal:RATE

Syntax

[:SOURce[<*n*>]]:FSKey:INTernal:RATE {<*rate*>|<*lim*>}

[:SOURce[<*n*>]]:FSKey:INTernal:RATE? [<*lim*>]

Description

Sets or queries the FSK modulation rate for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<rate>	Real	2 mHz to 1 MHz	100 Hz
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is only available for internal modulation source (*[:SOURce[<n>]]:FSKey:SOURce*). FSK rate is the rate at which the output frequency "shifts" between the carrier frequency (*[:SOURce[<n>]]:FREQuency*) and hop frequency (*[:SOURce[<n>]]:FSKey:FREQuency*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the FSK rate in scientific notation. For example, the query might return +1.500000000000000E+02, indicating that the FSK rate is 150 Hz.

Examples

```
:SOURce1:FSKey:INTernal:RATE 150 /*Sets the FSK rate to 150 Hz for
CH1.*/
:SOURce1:FSKey:INTernal:RATE? /*Queries the FSK rate for CH1. The
query returns +1.500000000000000E+02.*/
```

3.11.7.3 [:SOURce[<n>]]:FSKey:POLarity

Syntax

[:SOURce[<*n*>]]:FSKey:POLarity <*polarity*>

[:SOURce[<*n*>]]:FSKey:POLarity?

Description

Sets or queries the FSK modulation polarity for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<polarity>	Discrete	{POSitive|NEGative}	POSitive

Remarks

- The FSK modulation polarity can be set to the following:

- **POSitive:** positive polarity.
- **NEGative:** negative polarity.

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns POS or NEG.

Examples

```
:SOURce1:FSKey:POLarity NEGative /*Sets the FSK modulation
polarity to negative for CH1.*/
:SOURce1:FSKey:POLarity? /*Queries the FSK modulation polarity for
CH1. The query returns NEG.*/
```

3.11.7.4 [:SOURce[<n>]]:FSKey:SOURce

Syntax

[:SOURce[<*n*>]]:FSKey:SOURce <*source*>

[:SOURce[<*n*>]]:FSKey:SOURce?

Description

Sets or queries the FSK modulation source for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- The instrument can receive modulating waveforms from internal or external modulation source.
  - **INTernal:** internal modulation source. When the internal modulation source is selected, the modulating waveform is a square waveform with 50% duty cycle.
  - **EXTernal:** when the external modulation source is selected, the generator receives the external modulating signal from the rear-panel **[AUX IN/OUT]** connector.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT or EXT.

Examples

```
:SOURce1:FSKey:SOURce INTernal /*Sets the FSK modulation source to
internal modulation source for CH1.*/
:SOURce1:FSKey:SOURce? /*Queries the FSK modulation source for
CH1. The query returns INT.*/
```

3.11.7.5 [:SOURce[<n>]]:FSKey:STATe

Syntax

[:SOURce[<*n*>]]:FSKey:STATe <*bool*>

[:SOURce[<*n*>]]:FSKey:STATe?

Description

Sets or queries the on/off status of the FSK modulation function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the modulation function is enabled.
- The harmonic, noise, DC, and pulse cannot be modulated for FSK.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:FSKey:STATe ON /*Enables the FSK modulation function for
CH1.*/
:SOURce1:FSKey:STATe? /*Queries the on/off status of the FSK
modulation function for CH1. The query returns 1.*/
```

3.11.8 [:SOURce[<n>]]:FUNCtion

[:SOURce[<n>]]:FUNCtion commands are used to set and query some parameters of Arb, pulse, ramp, square, and sequence waveforms.

3.11.8.1 [:SOURce[<n>]]:FUNCtion

Syntax

[:SOURce[<*n*>]]:FUNCtion <*shape*>

[:SOURce[<*n*>]]:FUNCtion?

Description

Sets or queries the waveform type for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<shape>	Discrete	{SINusoid|SQUare|RAMP|PULSe|NOISe|ARB|HARMonic}	-

Remarks

- The parameter <shape> can be set to SINusoid, SQUare, RAMP, PULSe, NOISe, HARMonic, or ARB.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the waveform name, for example, SQU.

Examples

```
:SOURce1:FUNCtion SQUare /*Sets the waveform type to Square for
CH1.*/
:SOURce1:FUNCtion? /*Queries the waveform type for CH1. The query
returns SQU.*/
```

3.11.8.2 [:SOURce[<n>]]:FUNCtion:ARBitrary

Syntax

[:SOURce[<*n*>]]:FUNCtion:ARBitrary <*arb*>

[:SOURce[<*n*>]]:FUNCtion:ARBitrary?

Description

Sets or queries the arbitrary waveform type for the specified channel.

Parameter

Name	Type	Range	Default
<arb>	Discrete	Refer to *Remarks*	-

Remarks

- The parameter <arb> is used to set the Arb type. The range is {DC|KAISER|BLASEIWAV|SINC|NEGRAMP|ATTALT|AMPALT|STAIRDN|STAIRUP|STAIRUD|CPULSE|PPULSE|NPULSE|TRAPEZIA|ROUNDHALF|ABSSINE|ABSSINEHALF|SINETRA|SINEVER|EXPRISE|EXPFALL|TAN|COT|SQRT|X2DATA|GAUSS|HAVERSINE|LORENTZ|DIRICHLET|GAUSSPULSE|AIRY|CARDIAC|QUAKE|GAMMA|VOICE|TV|COMBIN|BANDLIMITED|STEPRESP|BUTTERWORTH|CHEBYSHEV1|CHEBYSHEV2|BOXCAR|BARLETT|TRIANG|BLACKMAN|HAMMING|HANNING|DUALTONE|ACOS|ACOSH|ACOTCON|ACOTPRO|ACOTHCON|ACOTHPRO|ACSCCON|ACSCPRO|ACSCHCON|ACSCHPRO|ASECCON|ASECPRO|ASECH|ASIN|ASINH|ATAN|ATANH|BESSELJ|BESSELY|CAUCHY|COSH|COSINT|COTHCON|COTHPRO|CSCCON|CSCPRO|CSCHCON|CSCHPRO|CUBIC|ERF|ERFC|ERFCINV|ERFINV|LAGUERRE|LAPLACE|LEGEND|LOG|LOGNORMAL|MAXWELL|RAYLEIGH|RECIPCON|RECIPPRO|SECCON|SECPRO|SECH|SINH|SININT|TANH|VERSIERA|WEIBULL|BARTHANN|BLACKMANH|BOHMANWIN|CHEBWIN|FLATTOPWIN|NUTTALLWIN|PARZENWIN|TAYLORWIN|TUKEYWIN|CWPULSE|LFPULSE|LFMPULSE|EOG|EEG|EMG|PULSILOGRAM|TENS1|TENS2|TENS3|SURGE|DAMPEDOSC|SWINGOSC|RADAR|THREEAM|THREEFM|THREEPM|THREEPWM|THREEPFM|RESSPEED|MCNOISE|PAHCUR|RIPPLE|ISO76372TP1|ISO76372TP2A|ISO76372TP2B|ISO76372TP3A|ISO76372TP3B|ISO76372TP4|ISO76372TP5A|ISO76372TP5B|ISO167502SP|ISO167502VR|SCR|IGNITION|NIMHDISCHARGE|GATEVIBR}.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the Arb type, for example, DC.

Examples

```
:SOURce1:FUNCtion:ARBitrary DC /*Sets the Arb type to DC for CH1.*/
:SOURce1:FUNCtion:ARBitrary? /*Queries the Arb type for CH1. The
query returns DC.*/
```

3.11.8.3 [:SOURce[<n>]]:FUNCtion:PULSe:DCYCle

Syntax

[:SOURce[<*n*>]]:FUNCtion:PULSe:DCYCle {<*percent*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:PULSe:DCYCle? [<*lim_query*>]

Description

Sets or queries the pulse duty cycle for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Name	Type	Range	Default
<percent>	Real	0.01% to 99.99%	50%
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The pulse duty cycle is defined as the percentage of the pulse width (*[:SOURce[<n>]]:FUNCtion:PULSe:WIDTh*) to the pulse period (*[:SOURce[<n>]]:FUNCtion:PULSe:PERiod*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the pulse duty cycle in scientific notation. For example, the query might return +4.500000000000000E+01, indicating that the pulse duty cycle is 45%.

Examples

```
:SOURce1:FUNCtion:PULSe:DCYCle 45 /*Sets the pulse duty cycle to
45% for CH1.*/
:SOURce1:FUNCtion:PULSe:DCYCle? /*Queries the pulse duty cycle for
CH1. The query returns +4.500000000000000E+01.*/
```

3.11.8.4 [:SOURce[<n>]]:FUNCtion:PULSe:PERiod

Syntax

[:**SOURce**[<*n*>]]:**FUNCtion:PULSe:PERiod** {<*seconds*>|<*lim_set*>}

[:**SOURce**[<*n*>]]:**FUNCtion:PULSe:PERiod?**[<*lim_query*>]

Description

Sets or queries the pulse period for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	Refer to *Remarks*	1 ms
<lim_set>	Discrete	{MAXimum|MINimum|DEFault}	-
<lim_query>	Discrete	{MAXimum|MINimum}	-

Remarks

- As the period and frequency (*[:SOURce[<n>]]:FREQuency*) specify the same parameter, the recently executed command will overwrite the previous command.
- For the pulse period setting range, refer to *Table 3.61 Range of Continuous Waveform Period*.
- When the wave type of the specified channel is changed, the instrument still uses the period if the period is valid. Otherwise, the instrument automatically sets the period as the lower limit for the new wave type.
- The instrument will adjust the edge time and pulse width automatically according to the period setting to accommodate the specified period: Period ≥ [Pulse Width + (Leading Edge Time + Trailing Edge Time)*0.625) + 5.25 ns].
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the pulse period in scientific notation. For example, the query might return +1.000000000000000E-01, indicating that the period is 0.1 s.

Examples

```
:SOURce1:FUNCtion:PULSe:PERiod 0.1 /*Sets the pulse period to 0.1
s for CH1.*/
:SOURce1:FUNCtion:PULSe:PERiod? /*Queries the pulse period for
CH1. The query returns +1.000000000000000E-01.*/
```

3.11.8.5 [:SOURce[<n>]]:FUNCtion:PULSe:TRANsition:LEADing

Syntax

[:SOURce[<*n*>]]:FUNCtion:PULSe:TRANsition:LEADing {<*seconds*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:PULSe:TRANsition:LEADing? [<*lim_query*>]

Description

Sets or queries the pulse edge time on the leading edge for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	3 ns to 1 s	3 ns
<lim_set>	Discrete	{MAXimum|MINimum|DEFault}	-
<lim_query>	Discrete	{MAXimum|MINimum}	-

Remarks

- Leading (rising) edge time is the time it takes for the pulse level to go from 10% to 90%.
- The range of the rise time is limited by the current waveform frequency and pulse width. When the set value exceeds the limits, the instrument will adjust the edge time automatically to accommodate the specified pulse width.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the pulse rise time in scientific notation. For example, the query might return +3.500000000000000E-08, indicating that the pulse rise time is 35 ns.

Examples

```
:SOURce1:FUNCtion:PULSe:TRANsition:LEADing 0.000000035 /*Sets the
pulse rise time to 35 ns for CH1.*/
:SOURce1:FUNCtion:PULSe:TRANsition:LEADing? /*Queries the pulse
rise time for CH1. The query returns +3.500000000000000E-08.*/
```

3.11.8.6 [:SOURce[<n>]]:FUNCtion:PULSe:TRANsition:TRAiling

Syntax

[:SOURce[<*n*>]]:FUNCtion:PULSe:TRANsition:TRAiling {<*seconds*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:PULSe:TRANsition:TRAiling? [<*lim_query*>]

Description

Sets or queries the pulse edge time on the trailing edge for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	3 ns to 1 s	3 ns
<lim_set>	Discrete	{MAXimum|MINimum|DEFault}	-
<lim_query>	Discrete	{MAXimum|MINimum}	-

Remarks

- The trailing (falling) edge time can be defined as the time it takes for the pulse level to go from 90% to 10%.
- The range of the trailing edge time is limited by the current waveform frequency, pulse width, and leading edge time. When the set value exceeds the limits, the instrument will adjust the edge time automatically.

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the pulse fall time in scientific notation. For example, the query might return +3.500000000000000E-08, indicating that the pulse fall time is 35 ns.

Examples

```
:SOURce1:FUNCtion:PULSe:TRANsition:TRAiling 0.000000035 /*Sets the pulse fall time to 35 ns for CH1.*/
:SOURce1:FUNCtion:PULSe:TRANsition:TRAiling? /*Queries the pulse fall time for CH1. The query returns +3.500000000000000E-08.*/
```

3.11.8.7 [:SOURce[<n>]]:FUNCtion:PULSe:WIDTh

Syntax

[:SOURce[<*n*>]]:FUNCtion:PULSe:WIDTh {<*seconds*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:PULSe:WIDTh? [<*lim_query*>]

Description

Sets or queries the pulse width for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	9 ns to 999.9 ks	500 µs
<lim_set>	Discrete	{MAXimum|MINimum|DEFault}	-
<lim_query>	Discrete	{MAXimum|MINimum}	-

Remarks

- Pulse width can be defined as the time from the 50% amplitude of a pulse's rising edge to the 50% amplitude of the next falling edge.
- The actual range of the pulse width (T_W) is limited by the pulse period (T): 0.01%*T ≤ T_W ≤ 99.99%*T.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the pulse width in scientific notation. For example, the query might return +1.000000000000000E-02, indicating that the pulse width is 10 ms (0.01 s).

Examples

```
:SOURce1:FUNCtion:PULSe:WIDTh 0.01 /*Sets the pulse width to 10 ms
(0.01 s) for CH1.*/
:SOURce1:FUNCtion:PULSe:WIDTh? /*Queries the pulse width for CH1.
The query returns +1.000000000000000E-02.*/
```

3.11.8.8 [:SOURce[<n>]]:FUNCtion:RAMP:SYMMetry

Syntax

[:SOURce[<*n*>]]:FUNCtion:RAMP:SYMMetry {<*symmetry*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:RAMP:SYMMetry? [<*lim_query*>]

Description

Sets or queries the ramp symmetry for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	0.1% to 99.9%	50%
<lim_set>	Discrete	{MAXimum|MINimum|DEFault}	-
<lim_query>	Discrete	{MAXimum|MINimum}	-

Remarks

- Symmetry is defined as the percentage of the amount of time Ramp wave is rising in the period.
- Sending *[:SOURce[<n>]]:APPLy:RAMP* overwrites the current symmetry setting and sets the symmetry to 50%.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the symmetry in scientific notation. For example, the query might return +5.500000000000000E+01, indicating that the ramp symmetry is 55%.

Examples

```
:SOURce1:FUNCtion:RAMP:SYMMetry 55 /*Sets the ramp symmetry to 55%
for CH1.*/
:SOURce1:FUNCtion:RAMP:SYMMetry? /*Queries the ramp symmetry for
CH1. The query returns +5.500000000000000E+01.*/
```

3.11.8.9 [:SOURce[<n>]]:FUNCtion:SEQuence:LIST:APPLy

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:APPLy

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:APPLy?

Description

Applies all waveform and loop settings to the sequence for the specified channel.

Queries whether the waveform and loop setting modifications are applied to the sequence for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Remarks

When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:FUNCtion:SEQuence:LIST:APPLy /*Applies all waveform and
loop settings to the sequence for CH1.*/
:SOURce1:FUNCtion:SEQuence:LIST:APPLy? /*Queries whether the
waveform and loop setting modifications are applied to the
sequence for CH1. The query returns 1.*/
```

3.11.8.10 [:SOURce[<n>]]:FUNCtion:SEQuence:LIST:CLEar

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:CLEar

Description

Clears all waveform and loop data of the sequence for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Remarks

When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:FUNCtion:SEQuence:LIST:CLEar /*Clears all waveform and
loop data of the sequence for CH1.*/
```

3.11.8.11 [:SOURce[<n>]]:FUNCtion:SEQuence:LIST:FILTer

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:FILTer <*filter*>

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:FILTer?

Description

Sets or queries the filter mode of the sequence waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<filter>	Discrete	{INSert|STEP|NORMal}	NORMal

Remarks

- The filter mode can be set to normal (NORMal), step (STEP), and interpolation (INSert).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns NORM, STEP, or INS.

Examples

```
:SOURce1:FUNCtion:SEQuence:LIST:FILTer STEP /*Sets the filter mode
of the sequence waveform to step for CH1.*/
:SOURce1:FUNCtion:SEQuence:LIST:FILTer? /*Queries the filter mode
of the sequence waveform for CH1. The query returns STEP.*/
```

3.11.8.12 [:SOURce[<n>]]:FUNCtion:SEQuence:LIST:LENGth

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:LENGth <*num*>,{<*value*>|<*lim*>}

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:LENGth? {<*num*>|<*all*>}

Description

Sets or queries the length of the specified step of the sequence waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<num>	Integer	1 to 64	-
<value>	Integer	Refer to *Remarks*	-
<lim>	Discrete	{MINimum|MAXimum}	-
<all>	Discrete	{ALL}	-

Remarks

- For DG800 Pro series, the range of <value> is from 32 pts to 2 Mpts (8 Mpts/CH optional) and the total points of valid steps (the loop is not 0) of the current sequence cannot exceed 2 Mpts (8 Mpts/CH optional); for DG900 Pro series, the range of <value> is from 32 pts to 16 Mpts (32 Mpts/CH optional) and the total points of valid steps (the loop is not 0) of the current sequence cannot exceed16 Mpts (32 Mpts/CH optional)
- You can use <all> to query the length of all steps.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

For a specified step, the query returns an integer; for all the steps (ALL), the query returns 64 integers (representing the lengths of 64 steps) separated by ",".

Examples

```
:SOURce1:FUNCtion:SEQuence:LIST:LENGth 1,100 /*Sets the length of
the step 1 of the sequence waveform to 100 for CH1.*/
:SOURce1:FUNCtion:SEQuence:LIST:LENGth? 1 /*Queries the length of
the step 1 of the sequence waveform for CH1. The query returns
100.*/
```

3.11.8.13 [:SOURce[<n>]]:FUNCtion:SEQuence:LIST:PERiod

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:PERiod <*num*>,{<*value*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:PERiod? {<*num*>|<lim_query>}

Description

Sets or queries the loop of the specified step of the sequence waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<num>	Integer	1 to 64	-
<value>	Integer	0 to 256	1
<lim_set>	Discrete	{MINimum|MAXimum}	-
<lim_query>	Discrete	{ALL}	-

Remarks

- MINimum|MAXimum specifies the minimum/maximum value. ALL indicates querying all the steps of the sequence waveform.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

For a specified step, the query returns an integer; for all the steps (ALL), the query returns 64 integers (representing the loops of 64 steps) separated by ",".

Examples

```
:SOURce1:FUNCtion:SEQuence:LIST:PERiod 1,10 /*Sets the loop of the
step 1 of sequence waveform to 10 for CH1.*/
:SOURce1:FUNCtion:SEQuence:LIST:PERiod? 1 /*Queries the loop of
the step 1 of sequence waveform for CH1. The query returns 10.*/
```

3.11.8.14 [:SOURce[<n>]]:FUNCtion:SEQuence:LIST:SRATe

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:SRATe {<*sample_rate*>|<*lim*>}

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:SRATe? [<*lim*>]

Description

Sets or queries the sample rate of the sequence waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Name	Type	Range	Default
<sample_rate>	Real	1 μSa/s to 312.5 MSa/s	1 MSa/s
<lim>	Discrete	{MAXimum|MINimum}	-

Remarks

When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the sample rate in scientific notation. For example, the query might return 3.00000000000000E+03, indicating the sample rate is 3 kSa/s.

Examples

```
:SOURce1:FUNCtion:SEQuence:LIST:SRATe 3000 /*Sets the sample rate
of the sequence waveform to 3 kSa/s for CH1.*/
:SOURce1:FUNCtion:SEQuence:LIST:SRATe? /*Queries the sample rate
of the sequence waveform for CH1. The query returns
3.00000000000000E+03.*/
```

3.11.8.15 [:SOURce[<n>]]:FUNCtion:SEQuence:LIST:WAVE

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:WAVE <*num*>,<*wavename*>

[:SOURce[<*n*>]]:FUNCtion:SEQuence:LIST:WAVE? {<*num*>|<*lim*>}

Description

Sets or queries the waveform of the specified step of the sequence waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<num>	Integer	1 to 64	-
<wavename>	Discrete	Refer to *Remarks*	-
<lim>	Discrete	{ALL}	-

Remarks

- The parameter <wavename> can be various basic waveforms and built-in waveforms. Its range is {SIN|SQU|RAMP|NOISe|KAISER|BLASEIWAV|SINC|NEGRAMP|ATTALT|AMPALT|STAIRDN|STAIRUP|STAIRUD|CPULSE|PPULSE|NPULSE|TRAPEZIA|ROUNDHALF|ABSSINE|ABSSINEHALF|SINETRA|SINEVER|EXPRISE|EXPFALL|TAN|COT|SQRT|X2DATA|GAUSS|HAVERSINE|LORENTZ|

DIRICHLET|GAUSSPULSE|AIRY|CARDIAC|QUAKE|GAMMA|VOICE|TV|COMBIN|BANDLIMITED|STEPRESP|BUTTERWORTH|CHEBYSHEV1|CHEBYSHEV2|BOXCAR|BARLETT|TRIANG|BLACKMAN|HAMMING|HANNING|DUALTONE|ACOS|ACOSH|ACOTCON|ACOTPRO|ACOTHCON|ACOTHPRO|ACSCCON|ACSCPRO|ACSCHCON|ACSCHPRO|ASECCON|ASECPRO|ASECH|ASIN|ASINH|ATAN|ATANH|BESSELJ|BESSELY|CAUCHY|COSH|COSINT|COTHCON|COTHPRO|CSCCON|CSCPRO|CSCHCON|CSCHPRO|CUBIC|ERF|ERFC|ERFCINV|ERFINV|LAGUERRE|LAPLACE|LEGEND|LOG|LOGNORMAL|MAXWELL|RAYLEIGH|RECIPCON|RECIPPRO|SECCON|SECPRO|SECH|SINH|SININT|TANH|VERSIERA|WEIBULL|BARTHANN|BLACKMANH|BOHMANWIN|CHEBWIN|FLATTOPWIN|NUTTALLWIN|PARZENWIN|TAYLORWIN|TUKEYWIN|CWPULSE|LFPULSE|LFMPULSE|EOG|EEG|EMG|PULSILOGRAM|TENS1|TENS2|TENS3|SURGE|DAMPEDOSC|SWINGOSC|RADAR|THREEAM|THREEFM|THREEPM|THREEPWM|THREEPFM|RESSPEED|MCNOISE|PAHCUR|RIPPLE|ISO76372TP1|ISO76372TP2A|ISO76372TP2B|ISO76372TP3A|ISO76372TP3B|ISO76372TP4|ISO76372TP5A|ISO76372TP5B|ISO167502SP|ISO167502VR|SCR|IGNITION|NIMHDISCHARGE|GATEVIBR}.

- ALL indicates querying the waveform types of all steps.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

For a specified step, the query returns the wave type (e.g. SQU); for all the steps (ALL), the query returns the wave types of 64 steps.

Examples

```
:SOURce1:FUNCtion:SEQuence:LIST:WAVE 1,SQU /*Sets the waveform of the step 1 of the sequence waveform to Square for CH1.*/
:SOURce1:FUNCtion:SEQuence:LIST:WAVE? 1 /*Queries the waveform of the step 1 of the sequence waveform for CH1. The query returns SQU.*/
```

3.11.8.16 [:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]

Syntax

[:SOURce[<*n*>]]:FUNCtion:SEQuence[:STATe] <*bool*>

[:SOURce[<*n*>]]:FUNCtion:SEQuence[:STATe]?

Description

Sets or queries the on/off status of the Sequence function.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Discrete	{ON|1|OFF|0}	0|OFF

Remarks

Enabling/disabling the Sequence function for one channel also enables/disables the Sequence function for the other channel.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:FUNCtion:SEQuence:STATe ON /*Enables the Sequence
function for CH1.*/
:SOURce1:FUNCtion:SEQuence:STATe? /*Queries whether the Sequence
function is enabled for CH1. The query returns 1.*/
```

3.11.8.17 [:SOURce[<n>]]:FUNCtion:SQUare:DCYCle

Syntax

[:SOURce[<*n*>]]:FUNCtion:SQUare:DCYCle {<*percent*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:SQUare:DCYCle? [<*lim_query*>]

Description

Sets or queries the duty cycle percentage of square wave for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<percent>	Real	0.01% to 99.99%	50%
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Duty cycle represents the amount of time per period that the square wave is at a high level.
- Sending *[:SOURce[<n>]]:APPLy:SQUare* overwrites the current duty cycle setting and sets the duty cycle to 50%.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the square duty cycle in scientific notation. For example, the query might return +4.500000000000000E+01, indicating that the square duty cycle is 45%.

Examples

```
:SOURce1:FUNCtion:SQUare:DCYCle 45 /*Sets the square duty cycle to
45% for CH1.*/
:SOURce1:FUNCtion:SQUare:DCYCle? /*Queries the square duty cycle
for CH1. The query returns +4.500000000000000E+01.*/
```

3.11.8.18 [:SOURce[<n>]]:FUNCtion:SQUare:PERiod

Syntax

[:SOURce[<*n*>]]:FUNCtion:SQUare:PERiod {<*seconds*>|<*lim_set*>}

[:SOURce[<*n*>]]:FUNCtion:SQUare:PERiod? [<*lim_query*>]

Description

Sets or queries the square period for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	Refer to *Remarks*	1 ms
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- When the wave type of the specified channel is changed, the instrument still uses the period if the period is valid. Otherwise, the instrument automatically sets the period as the lower limit for the new wave type.
- For the square period setting range, refer to *Table 3.61 Range of Continuous Waveform Period*.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the square period in scientific notation. For example, the query might return +1.000000000000000E+00, indicating that the square period is 1 s.

Examples

```
:SOURce1:FUNCtion:SQUare:PERiod 1 /*Sets the square period to 1 s
for CH1.*/
:SOURce1:FUNCtion:SQUare:PERiod? /*Queries the square period for
CH1. The query returns +1.000000000000000E+00.*/
```

3.11.9 [:SOURce[<n>]]:HARMonic

[:SOURce[<n>]]:HARMonic commands are used to set and query harmonic parameters.

3.11.9.1 [:SOURce[<n>]]:HARMonic:COMBine

Syntax

[:SOURce[<*n*>]]:HARMonic:COMBine <*user*>

[:SOURce[<*n*>]]:HARMonic:COMBine?

Description

Sets or queries the combine harmonic for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<user>	ASCII string	Refer to *Remarks*	X0000000000000000000

Remarks

- The parameter <user> ranges from X0000000000000000000 to X1111111111111111111. The 20-bit binary data represent the output states of the 20 orders of harmonics. The leftmost bit represents the fundamental waveform; it is a fixed X and cannot be modified. The remaining 19 bits correspond to the 2nd order of harmonic to the 20th order of harmonic from left to right. 1/0 indicates enabling/disabling the corresponding order harmonic output. For example, set the 20-bit binary data to X0010001000000000000, indicating the output of the fundamental waveform, the 4th order of harmonic, and 8th order of harmonic.
- The max. fundamental frequency (F_{fund}) is limited by the max. harmonic order (N) and harmonic frequency upper limit (F_{max}): $F_{fund} = (2 \times F_{max} \div N)$. Changing the max. harmonic order may modify the fundamental frequency. For the harmonic frequency upper limit (F_{max}) of different models, refer to *Table 3.60 Range of Continuous Waveform Frequency*.
- This setup command is valid only when the harmonic type (*[:SOURce[<n>]]:HARMonic[:TYPE]*) is set to Combine (COMBine).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns a string between X0000000000000000000 and X1111111111111111111. For example, the query might return X0010001000000000000.

Examples

```
:SOURce1:HARMonic:COMBine X0010001000000000000 /*Sets CH1 to
output the fundamental waveform, the 4th order of harmonic, and
8th order of harmonic.*/
:SOURce1:HARMonic:COMBine? /*Queries the harmonic output for CH1.
The query returns X0010001000000000000.*/
```

3.11.9.2 [:SOURce[<n>]]:HARMonic:COMBine:AMPLitude

Syntax

[:SOURce[<*n*>]]:HARMonic:COMBine:AMPLitude <*sn*>,{<*amplitude*>|<*lim_set*>}

[:SOURce[<*n*>]]:HARMonic:COMBine:AMPLitude? {<*sn*>|<*all*>}[,<*lim_query*>]

Description

Sets or queries the amplitude of the specified order of combine harmonic for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<sn>	Integer	2 to 20	2
<amplitude>	Real	2 mVpp (HighZ) to the amplitude upper limit of the specified channel	5 Vpp
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<all>	Discrete	{ALL}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The amplitude upper limit of the specified channel is limited by the impedance setting (*:OUTPut[<n>]:LOAD*) and frequency/period setting.
- You can use <all> to query the amplitudes of all orders of the harmonic.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

If querying the specified harmonic, the query returns the amplitude of the specified order of combine harmonic, for example, 1.000000E +00. If querying the amplitudes of all orders of the harmonic (ALL), the query returns all amplitudes separated by commas.

Examples

```
:SOURce1:HARMonic:COMBine:AMPLitude 5,1 /*Sets the amplitude of
the 5th order of harmonic to 1 Vpp for CH1.*/
:SOURce1:HARMonic:COMBine:AMPLitude? 5 /*Queries the amplitude of
the 5th order of harmonic for CH1. The query returns 1.000000E
+00.*/
```

3.11.9.3 [:SOURce[<n>]]:HARMonic:COMBine:PHASe

Syntax

[:SOURce[<*n*>]]:HARMonic:COMBine:PHASe <*sn*>,{<*phase*>|<*lim_set*>}

[:SOURce[<*n*>]]:HARMonic:COMBine:PHASe? {<*sn*>|<*all*>}[,<*lim_query*>]

Description

Sets or queries the phase of the specified order of combine harmonic for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<sn>	Integer	2 to 20	2
<phase>	Real	0° to 360°	0°
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<all>	Discrete	{ALL}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- You can use <all> to query the phases of all orders of the harmonic.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

If querying the specified harmonic component, the query returns the phase of the specified order of combine harmonic, for example, 10.000000. If querying the phases

of all orders of the harmonic (ALL), the query returns all phases separated by commas.

Examples

```
:SOURce1:HARMonic:COMBine:PHASe 5,10 /*Sets the phase of the 5th
order of harmonic to 10° for CH1.*/
:SOURce1:HARMonic:COMBine:PHASe? 5 /*Queries the phase of the 5th
order of harmonic for CH1. The query returns 10.000000.*/
```

3.11.9.4 [:SOURce[<n>]]:HARMonic:ORDer

Syntax

[:SOURce[<*n*>]]:HARMonic:ORDer {<*value*>|<*lim*>}

[:SOURce[<*n*>]]:HARMonic:ORDer? [<*lim*>]

Description

Sets or queries the order of the order harmonic for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<value>	Integer	2 to 20	2
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- The max. fundamental frequency (F_{fund}) is limited by the harmonic order (N) and max. harmonic frequency (F_{max}): F_{fund} = (2 x F_{max} ÷ N). Changing the harmonic order may modify the fundamental frequency. For the harmonic frequency upper limit (F_{max}) of different models, refer to *Table 3.60 Range of Continuous Waveform Frequency* .
- When the harmonic type (*[:SOURce[<n>]]:HARMonic[:TYPE]*) is set to order, you can use this command to set the harmonic component to be output.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns an integer. For example, the query might return 10, indicating that the harmonic order is 10.

Examples

```
:SOURce1:HARMonic:ORDer 10 /*Sets the order of the order harmonic
to 10 for CH1.*/
```

```
:SOURce1:HARMonic:ORDer? /*Queries the order of the order harmonic
for CH1. The query returns 10.*/
```

3.11.9.5 [:SOURce[<n>]]:HARMonic:ORDEr:AMPLitude

Syntax

[:SOURce[<*n*>]]:HARMonic:ORDEr:AMPLitude {<*amplitude*>|<*lim_set*>}

[:SOURce[<*n*>]]:HARMonic:ORDEr:AMPLitude? [<*lim_query*>]

Description

Sets or queries the amplitude of the specified order of order harmonic for the specified channel.

Parameter

Name	Type	Range	Default
<amplitude>	Real	2 mVpp (HighZ) to the amplitude upper limit of the specified channel	5 Vpp
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The amplitude upper limit of the specified channel is limited by the impedance setting (*:OUTPut[<n>]:LOAD*) and frequency/period setting.
- You can use *[:SOURce[<n>]]:HARMonic:ORDer* to set or query the order of the order harmonic for the specified channel.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the harmonic amplitude in scientific notation. For example, the query might return 1.000000E+00, indicating that the harmonic amplitude is 1 Vpp.

Examples

```
:SOURce1:HARMonic:ORDEr:AMPLitude 1 /*Sets the amplitude of the
current order of order harmonic to 1 Vpp for CH1.*/
:SOURce1:HARMonic:ORDEr:AMPLitude? /*Queries the amplitude of the
current order of order harmonic for CH1. The query returns
1.000000E+00.*/
```

3.11.9.6 [:SOURce[<n>]]:HARMonic:ORDEr:PHASe

Syntax

[:SOURce[<*n*>]]:HARMonic:ORDEr:PHASe {<*phase*>|<*lim_set*>}

[:SOURce[<*n*>]]:HARMonic:ORDEr:PHASe? [<*lim_query*>]

Description

Sets or queries the phase of the specified order of order harmonic for the specified channel.

Parameter

Name	Type	Range	Default
<phase>	Real	0° to 360°	0°
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- You can use *[:SOURce[<n>]]:HARMonic:ORDer* to set or query the order of the order harmonic for the specified channel.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns a real number. For example, the query might return 10.000000, indicating that the phase is 10°

Examples

```
:SOURce1:HARMonic:ORDEr:PHASe 10 /*Sets the phase of the current
order of order harmonic to 10° for CH1.*/
:SOURce1:HARMonic:ORDEr:PHASe? /*Queries the phase of the current
order of order harmonic for CH1. The query returns 10.000000.*/
```

3.11.9.7 [:SOURce[<n>]]:HARMonic[:TYPE]

Syntax

[:SOURce[<*n*>]]:HARMonic[:TYPE] <*type*>

[:SOURce[<*n*>]]:HARMonic[:TYPE]?

Description

Sets or queries the harmonic type for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<type>	Discrete	{ORDer|COMBine}	ORDer

Remarks

- The harmonic type can be set to the following:
  - **ORDer:** order harmonic which outputs the basic waveform and a single harmonic component (*[:SOURce[<n>]]:HARMonic:ORDer*).
  - **COMBine:** combine harmonic which outputs harmonics with multiple harmonic components. You can self-define the orders of harmonic (*[:SOURce[<n>]]:HARMonic:COMBine*). The maximum order is 20.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns ORD or COMB.

Examples

```
:SOURce1:HARMonic:TYPE ORDer /*Sets the harmonic type to order
harmonic for CH1.*/
:SOURce1:HARMonic:TYPE? /*Queries the harmonic type for CH1. The
query returns ORD.*/
```

3.11.10 [:SOURce[<n>]]:MARKer:FREQuency

Syntax

[:SOURce[<*n*>]]:MARKer:FREQuency {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:MARKer:FREQuency? [<*lim_query*>]

Description

Sets or queries the mark frequency for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	Refer to *Remarks*	550 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The parameter <frequency> is limited by the sweep "Start Frequency" (*[:SOURce[<n>]]:FREQuency:STARt*) and "Stop Frequency" (*[:SOURce[<n>]]:FREQuency:STOP*). It must lie between the start frequency and stop frequency.

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency in scientific notation. For example, the query might return +5.000000000000000E+02, indicating that the frequency is 500 Hz.

Examples

```
:SOURce1:MARKer:FREQuency 500 /*Sets the mark frequency to 500 Hz
for CH1.*/
:SOURce1:MARKer:FREQuency? /*Queries the mark frequency for CH1.
The query returns +5.000000000000000E+02.*/
```

3.11.11 [:SOURce[<n>]]:PARameter:COPY

Syntax

[:SOURce[<*n*>]]:PARameter:COPY <*ch*>

Description

Copies the states of the specified channel (source channel) to the target channel, or swaps the states of the two channels.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<ch>	Discrete	{CH1|CH2|SWAP}	-

Remarks

- <ch> can be set to CH1, CH2, or SWAP.
  - **CH1/CH2:** sets the target channel to CH1 or CH2. <n> specifies the source channel while <ch> specifies the target channel. They cannot be the same channel.
  - **SWAP:** swaps the states of CH1 and CH2.
- The channel copy function is not available for DG821 Pro (unless the two-channel upgrade option is installed).
- The channel on/off status is not included in the available copy states, for which you can refer to the User Guide.
- The Channel Copy function is disabled when channel track is enabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:PARameter:COPY SWAP /*Swaps the states of the two
channels.*/
```

3.11.12 [:SOURce[<n>]]:PHASe

[:SOURce[<n>]]:PHASe commands are used to set and query the start phase of the channel waveform (basic waveform and Arb), perform align phase, and set the phase coupling.

3.11.12.1 [:SOURce[<n>]]:PHASe

Syntax

[:SOURce[<*n*>]]:PHASe {<*phase*>|<*lim_set*>}

[:SOURce[<*n*>]]:PHASe? [<*lim_query*>]

Description

Sets or queries the start phase of the waveform (basic waveform and Arb) for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<phase>	Real	-360° to 360°	0°
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the start phase in scientific notation. For example, the query might return +5.000000000000000E+01, indicating that the start phase is 50°.

Examples

```
:SOURce1:PHASe 50 /*Sets the waveform start phase to 50° for CH1.*/
:SOURce1:PHASe? /*Queries the waveform start phase for CH1. The
query returns +5.000000000000000E+01.*/
```

3.11.12.2 [:SOURce[<n>]]:PHASe:COUPle:MODE

Syntax

[:SOURce[<*n*>]]:PHASe:COUPle:MODE <*mode*>

[:SOURce[<*n*>]]:PHASe:COUPle:MODE?

Description

Sets or queries the type of the phase coupling mode for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<mode>	Discrete	{OFFSet|RATio}	RATio

Remarks

- The phase coupling mode can be set to the following:
  - **OFFSet:** phase deviation mode. You can use *[:SOURce[<n>]]:PHASe:COUPle:OFFSet* to set the phase deviation of the coupled channel.
  - **RATio:** phase ratio mode. You can use *[:SOURce[<n>]]:PHASe:COUPle:RATio* to set the phase ratio of the coupled channel.
- When the coupling is enabled, if switching to ratio coupling from deviation coupling causes either channel to exceed the phase upper/lower limit after calculation according to coupling rules, the generator will automatically adjust the parameters to avoid parameter overlimit. If the parameter overlimit is caused by switching to deviation coupling from ratio coupling, the phase coupling will be automatically disabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns OFFS or RAT.

Examples

```
:SOURce1:PHASe:COUPle:MODE OFFSet /*Sets the type of the phase
coupling mode to phase deviation for CH1.*/
:SOURce1:PHASe:COUPle:MODE? /*Queries the type of the phase
coupling mode for CH1. The query returns OFFS.*/
```

3.11.12.3 [:SOURce[<n>]]:PHASe:COUPle:OFFSet

Syntax

[:SOURce[<*n*>]]:PHASe:COUPle:OFFSet <*offset*>

[:SOURce[<*n*>]]:PHASe:COUPle:OFFSet?

Description

Sets or queries the phase deviation of the phase coupling for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<offset>	Real	-360° to 360°	0°

Remarks

- When modifying the deviation, if base channel parameters exceed the limits of the coupled channel after coupling calculation, the generator will automatically adjust the coupling deviation to avoid parameter overlimit.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the phase deviation in scientific notation. For example, the query might return +1.000000000000000E+02, indicating that the phase deviation is 100°.

Examples

```
:SOURce1:PHASe:COUPle:OFFSet 100 /*Sets the phase deviation of the
phase coupling to 100° for CH1.*/
:SOURce1:PHASe:COUPle:OFFSet? /*Queries the phase deviation of the
phase coupling for CH1. The query returns +1.000000000000000E+02.*/
```

3.11.12.4 [:SOURce[<n>]]:PHASe:COUPle:RATio

Syntax

[:SOURce[<*n*>]]:PHASe:COUPle:RATio <*ratio*>

[:SOURce[<*n*>]]:PHASe:COUPle:RATio?

Description

Sets or queries the phase ratio of the phase coupling for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<ratio>	Real	0.01 to 100	1

Remarks

- When modifying the ratio, if base channel parameters exceed the limits of the coupled channel after coupling calculation, the generator will automatically adjust the waveform phase to avoid parameter overlimit.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the phase ratio in scientific notation. For example, the query might return +2.000000000000000E+00, indicating that the ratio is 2.

Examples

```
:SOURce1:PHASe:COUPle:RATio 2 /*Sets the phase ratio of the phase
coupling to 2 for CH1.*/
:SOURce1:PHASe:COUPle:RATio? /*Queries the phase ratio of the
phase coupling for CH1. The query returns +2.000000000000000E+00.*/
```

3.11.12.5 [:SOURce[<n>]]:PHASe:COUPle[:STATe]

Syntax

[:SOURce[<*n*>]]:PHASe:COUPle[:STATe] <*bool*>

[:SOURce[<*n*>]]:PHASe:COUPle[:STATe]?

Description

Sets or queries the on/off status of the phase coupling function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- The phase coupling function is not available for DG821 Pro (unless the two-channel upgrade option is installed).

- When the coupling mode is set to "ratio", if the frequency, amplitude, or phase of either channel exceeds the upper/lower limit after calculation according to the coupling rules due to modifying the coupling ratio, switching waveform, or changing waveform parameters, the generator will automatically adjust the waveform parameters to avoid parameter overlimit.
- When the coupling mode is set to "deviation", if the frequency, amplitude or phase of either channel exceeds the upper/lower limit after calculation according to the coupling rules due to modifying the coupling deviation, the generator will automatically adjust the coupling deviation to avoid parameter overlimit; if it is due to switching waveform or changing waveform parameters, the corresponding coupling function is disabled.
- The phase coupling function is available only when the output modes of both channels are set to Continuous and the output waveform is set to Sine, Square, or Ramp.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:PHASe:COUPle:STATe ON /*Enables the phase coupling
function for CH1.*/
:SOURce1:PHASe:COUPle:STATe? /*Queries the on/off status of the
phase coupling function for CH1. The query returns 1.*/
```

3.11.12.6 [:SOURce[<n>]]:PHASe:SYNChronize

Syntax

[:**SOURce**[<*n*>]]:**PHASe:SYNChronize**

Description

Performs an align phase operation.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Remarks

- The align phase function is only available for the Continuous mode as well as the internally triggered Burst and Sweep modes.
- The align phase function is not available for DG821 Pro (unless the two-channel upgrade option is installed).

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:PHASe:SYNChronize /*Performs an align phase operation.*/
```

3.11.13 [:SOURce[<n>]]:PM

[:SOURce[<n>]]:PM commands are used to set and query the PM parameters such as the phase deviation, modulating waveform frequency, modulation source, modulating waveform, and the on/off status of the PM modulation.

3.11.13.1 [:SOURce[<n>]]:PM:DEViation

Syntax

[:SOURce[<*n*>]]:PM:DEViation {<*deviation*>|<*lim_set*>}

[:SOURce[<*n*>]]:PM:DEViation? [<*lim_query*>]

Description

Sets or queries the PM phase deviation for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<deviation>	Real	0° to 360°	90°
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Phase deviation represents the peak variation in phase of the modulated waveform from the carrier waveform.
- When an external modulation source (*[:SOURce[<n>]]:PM:SOURce*) is selected, the phase deviation is controlled by the signal from the real-panel **[AUX IN/ OUT]** connector.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the PM phase deviation in scientific notation. For example, the query might return +5.000000000000000E+01, indicating that the phase deviation is 50°.

Examples

```
:SOURce1:PM:DEViation 50 /*Sets the PM phase deviation to 50° for
CH1.*/
:SOURce1:PM:DEViation? /*Queries the PM phase deviation for CH1.
The query returns +5.000000000000000E+01.*/
```

3.11.13.2 [:SOURce[<n>]]:PM:INTernal:FREQuency

Syntax

[:SOURce[<*n*>]]:PM:INTernal:FREQuency {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:PM:INTernal:FREQuency? [<*lim_query*>]

Description

Sets or queries the PM modulation frequency for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	2 mHz to 1 MHz	100 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is only available for internal modulation source (*[:SOURce[<n>]]:PM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency in scientific notation. For example, the query might return +1.500000000000000E+02, indicating that the frequency is 150 Hz.

Examples

```
:SOURce1:PM:INTernal:FREQuency 150 /*Sets the frequency of the PM
modulating waveform to 150 Hz for CH1.*/
:SOURce1:PM:INTernal:FREQuency? /*Queries the frequency of the PM
modulating waveform for CH1. The query returns +1.500000000000000E
+02.*/
```

3.11.13.3 [:SOURce[<n>]]:PM:INTernal:FUNCtion

Syntax

[:SOURce[<*n*>]]:PM:INTernal:FUNCtion <*function*>

[:SOURce[<*n*>]]:PM:INTernal:FUNCtion?

Description

Sets or queries the PM modulating waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<function>	Discrete	{SINusoid|SQUare|TRIangle|RAMP|NRAMp|NOISe|ARB}	SINusoid

Remarks

- PM supports the following internal modulating waveforms:
  - **SINusoid:** Sine wave.
  - **SQUare:** Square with 50% duty cycle.
  - **TRIangle:** Triangle with 50% symmetry.
  - **RAMP:** UpRamp with 100% symmetry.
  - **NRAMp:** DnRamp with 0% symmetry.
  - **NOISe:** white gaussian noise.
  - **ARB:** arbitrary waveform.
- This command is only available for internal modulation source (*[:SOURce[<n>]]:PM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns SIN, SQU, TRI, RAMP, NRAM, NOIS, or ARB.

Examples

```
:SOURce1:PM:INTernal:FUNCtion SQUare /*Sets the PM modulating
waveform to Square for CH1.*/
:SOURce1:PM:INTernal:FUNCtion? /*Queries the PM modulating
waveform for CH1. The query returns SQU.*/
```

3.11.13.4 [:SOURce[<n>]]:PM:SOURce

Syntax

[:SOURce[<*n*>]]:PM:SOURce <*source*>

[:SOURce[<*n*>]]:PM:SOURce?

Description

Sets or queries the PM modulation source for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- The instrument can receive modulating waveforms from internal or external modulation source.
  - **INTernal:** internal modulation source. When the internal modulation source is selected, you can use *[:SOURce[<n>]]:PM:INTernal:FUNCtion* to select the modulating waveforms.
  - **EXTernal:** external modulation source. When the external modulation source is selected, the generator receives the external modulating signal from the rear-panel **[AUX IN/OUT]** connector.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT or EXT.

Examples

```
:SOURce1:PM:SOURce INTernal/*Sets the PM modulation source to
internal modulation source for CH1.*/
:SOURce1:PM:SOURce? /*Queries the PM modulation source for CH1.
The query returns INT.*/
```

3.11.13.5 [:SOURce[<n>]]:PM:STATe

Syntax

[:SOURce[<*n*>]]:PM:STATe <*bool*>

[:SOURce[<*n*>]]:PM:STATe?

Description

Sets or queries the on/off status of the PM modulation function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the modulation function is enabled.
- The harmonic, noise, DC, and pulse cannot be modulated for PM.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:PM:STATe ON /*Enables the PM modulation function for
CH1.*/
:SOURce1:PM:STATe? /*Queries the on/off status of the PM
modulation function for CH1. The query returns 1.*/
```

3.11.14 [:SOURce[<n>]]:PSKey

[:SOURce[<n>]]:PSKey commands are used to set and query PSK parameters such as the PSK rate, modulation phase, modulation polarity, modulation source, and the on/off status of the PSK modulation function.

3.11.14.1 [:SOURce[<n>]]:PSKey:INTernal:RATE

Syntax

[:SOURce[<*n*>]]:PSKey:INTernal:RATE {<*rate*>|<*lim*>}

[:SOURce[<*n*>]]:PSKey:INTernal:RATE? [<*lim*>]

Description

Sets or queries the PSK modulation rate for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<rate>	Real	2 mHz to 1 MHz	100 Hz
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is only available for internal modulation source (*[:SOURce[<n>]]:PSKey:SOURce*). PSK rate is the rate at which the output phase "shifts" between the carrier phase (*[:SOURce[<n>]]:PHASe*) and modulation phase (*[:SOURce[<n>]]:PSKey:PHASe*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the PSK rate in scientific notation. For example, the query might return +1.500000000000000E+02, indicating that the PSK rate is 150 Hz.

Examples

```
:SOURce1:PSKey:INTernal:RATE 150 /*Sets the PSK rate to 150 Hz for
CH1.*/
:SOURce1:PSKey:INTernal:RATE? /*Queries the PSK rate for CH1. The
query returns +1.500000000000000E+02.*/
```

3.11.14.2 [:SOURce[<n>]]:PSKey:PHASe

Syntax

[:**SOURce**[<*n*>]]:**PSKey**:**PHASe** {<*phase*>|<*lim_set*>}

[:**SOURce**[<*n*>]]:**PSKey**:**PHASe?** [<*lim_query*>]

Description

Sets or queries the PSK modulation phase for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<phase>	Real	0° to 360°	180°
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- In PSK modulation, the generator "shift" its output phase between two preset phases, carrier phase (*[:SOURce[<n>]]:PHASe*) and modulation phase.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the PSK phase in scientific notation. For example, the query might return +9.000000000000000E+01, indicating that the PSK phase is 90°.

Examples

```
:SOURce1:PSKey:PHASe 90 /*Sets the PSK modulation phase to 90° for
CH1.*/
:SOURce1:PSKey:PHASe? /*Queries the PSK modulation phase for CH1.
The query returns +9.000000000000000E+01.*/
```

3.11.14.3 [:SOURce[<n>]]:PSKey:POLarity

Syntax

[:SOURce[<*n*>]]:PSKey:POLarity <*polarity*>

[:SOURce[<*n*>]]:PSKey:POLarity?

Description

Sets or queries the PSK modulation polarity for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<polarity>	Discrete	{POSitive|NEGative}	POSitive

Remarks

- The PSK modulation polarity can be set to the following:
  - **POSitive:** positive polarity.
  - **NEGative:** negative polarity.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns POS or NEG.

Examples

```
:SOURce1:PSKey:POLarity NEGative /*Sets the PSK modulation
polarity to negative for CH1.*/
:SOURce1:PSKey:POLarity? /*Queries the PSK modulation polarity for
CH1. The query returns NEG.*/
```

3.11.14.4 [:SOURce[<n>]]:PSKey:SOURce

Syntax

[:SOURce[<*n*>]]:PSKey:SOURce <*source*>

[:SOURce[<*n*>]]:PSKey:SOURce?

Description

Sets or queries the PSK modulation source for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- The instrument can receive modulating waveforms from internal or external modulation source.
  - **INTernal:** internal modulation source. When the internal modulation source is selected, the modulating waveform is a square waveform with 50% duty cycle.
  - **EXTernal:** external modulation source. When the external modulation source is selected, the generator receives the external modulating signal from the rear-panel **[AUX IN/OUT]** connector.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT or EXT.

Examples

```
:SOURce1:PSKey:SOURce INTernal /*Sets the PSK modulation source to
internal modulation source for CH1.*/
:SOURce1:PSKey:SOURce? /*Queries the PSK modulation source for
CH1. The query returns INT.*/
```

3.11.14.5 [:SOURce[<n>]]:PSKey:STATe

Syntax

[:SOURce[<*n*>]]:PSKey:STATe <*bool*>

[:SOURce[<*n*>]]:PSKey:STATe?

Description

Sets or queries the on/off status of the PSK modulation function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the modulation function is enabled.
- The harmonic, noise, DC, and pulse cannot be modulated for PSK.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:PSKey:STATe ON /*Enables the PSK modulation function for
CH1.*/
:SOURce1:PSKey:STATe? /*Queries the on/off status of the PSK
modulation function for CH1. The query returns 1.*/
```

3.11.15 [:SOURce[<n>]]:PWM

[:SOURce[<n>]]:PWM commands are used to set and query the PWM parameters such as the duty cycle deviation, width deviation, modulating waveform frequency, modulation source, modulating waveform, and the on/off status of the PWM modulation.

3.11.15.1 [:SOURce[<n>]]:PWM:DEViation

Syntax

[:SOURce[<*n*>]]:PWM:DEViation {<*deviation*>|<*lim_set*>}

[:SOURce[<*n*>]]:PWM:DEViation? [<*lim_query*>]

Description

Sets or queries the PWM width deviation for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<deviation>	Real	Refer to *Remarks*	10 µs
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Width deviation is limited by the pulse width, minimum pulse width, and edge time setting.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the PWM width deviation in scientific notation. For example, the query might return +1.000000000000000E-04, indicating that the width deviation is 100 µs (0.0001 s).

Examples

```
:SOURce1:PWM:DEViation 0.0001 /*Sets the PWM width deviation to
100 µs (0.0001 s) for CH1.*/
:SOURce1:PWM:DEViation? /*Queries the PWM width deviation for CH1.
The query returns +1.000000000000000E-04.*/
```

3.11.15.2 [:SOURce[<n>]]:PWM:DEViation:DCYCle

Syntax

[:SOURce[<*n*>]]:PWM:DEViation:DCYCle {<*percent*>|<*lim_set*>}

[:SOURce[<*n*>]]:PWM:DEViation:DCYCle? [<*lim_query*>]

Description

Sets or queries the PWM duty cycle deviation for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<percent>	Real	Refer to *Remarks*	1%
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Duty cycle deviation is limited by the duty cycle, minimum duty cycle, and edge time setting.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the PWM duty cycle deviation in scientific notation. For example, the query might return +1.500000000000000E+01, indicating that the PWM duty cycle deviation is 15%.

Examples

```
:SOURce1:PWM:DEViation:DCYCle 15 /*Sets the PWM duty cycle
deviation to 15% for CH1.*/
:SOURce1:PWM:DEViation:DCYCle? /*Queries the PWM duty cycle
deviation for CH1. The query returns +1.500000000000000E+01.*/
```

3.11.15.3 [:SOURce[<n>]]:PWM:INTernal:FREQuency

Syntax

[:SOURce[<*n*>]]:PWM:INTernal:FREQuency {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:PWM:INTernal:FREQuency? [<*lim_query*>]

Description

Sets or queries the modulation frequency of PWM modulation for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	2 mHz to 1 MHz	100 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-

Name	Type	Range	Default
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is available only when internal modulation source (*[:SOURce[<n>]]:PWM:SOURce*) is selected.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency in scientific notation. For example, the query might return +1.500000000000000E+05, indicating that the frequency is 150 kHz.

Examples

```
:SOURce1:PWM:INTernal:FREQuency 150000 /*Sets the frequency of the
PWM modulating waveform to 150 kHz for CH1.*/
:SOURce1:PWM:INTernal:FREQuency? /*Queries the frequency of the
PWM modulating waveform for CH1. The query returns
+1.500000000000000E+05.*/
```

3.11.15.4 [:SOURce[<n>]]:PWM:INTernal:FUNCtion

Syntax

[:SOURce[<*n*>]]:PWM:INTernal:FUNCtion <*function*>

[:SOURce[<*n*>]]:PWM:INTernal:FUNCtion?

Description

Sets or queries the PWM modulating waveform for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<function>	Discrete	{SINusoid|SQUare|TRIangle|RAMP|NRAMp|NOISe|ARB}	SINusoid

Remarks

- PWM supports the following internal modulating waveforms:
  - **SINusoid:** Sine wave.
  - **SQUare:** Square with 50% duty cycle.
  - **TRIangle:** Triangle with 50% symmetry.

- **RAMP:** UpRamp with 100% symmetry.
- **NRAMp:** DnRamp with 0% symmetry.
- **NOISe:** white gaussian noise.
- **ARB:** arbitrary waveform.

- This command is only available for internal modulation source (*[:SOURce[<n>]]:PWM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns SIN, SQU, RAMP, NRAM, TRI, NOIS, or ARB.

Examples

```
:SOURce1:PWM:INTernal:FUNCtion SQUare /*Sets the PWM modulating
waveform to Square for CH1.*/
:SOURce1:PWM:INTernal:FUNCtion? /*Queries the PWM modulating
waveform for CH1. The query returns SQU.*/
```

3.11.15.5 [:SOURce[<n>]]:PWM:SOURce

Syntax

[:SOURce[<*n*>]]:PWM:SOURce <*source*>

[:SOURce[<*n*>]]:PWM:SOURce?

Description

Sets or queries the PWM modulation source for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- The instrument can receive modulating waveforms from internal or external modulation.
  - **INTernal:** internal modulation source. When the internal modulation source is selected, you can use *[:SOURce[<n>]]:PWM:INTernal:FUNCtion* to select the modulating waveforms.

- **EXTernal:** external modulation source. When the external modulation source is selected, the generator receives the external modulating signal from the rear-panel **[AUX IN/OUT]** connector.

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT or EXT.

Examples

```
:SOURce1:PWM:SOURce EXTernal/*Sets the PWM modulation source to
external modulation source for CH1.*/
:SOURce1:PWM:SOURce? /*Queries the PWM modulation source for CH1.
The query returns EXT.*/
```

3.11.15.6 [:SOURce[<n>]]:PWM:STATe

Syntax

[:SOURce[<*n*>]]:PWM:STATe <*bool*>

[:SOURce[<*n*>]]:PWM:STATe?

Description

Sets or queries the on/off status of the PWM modulation function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- The carrier waveform of PWM can only be Pulse. Only when the current waveform is set to Pulse for the specified channel, can you enable the PWM function. Different settings of pulse parameters (e.g. frequency, amplitude, offset, pulse width, duty cycle) can affect the PWM modulated waveform.
- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the modulation function is enabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:PWM:STATe ON /*Enables the PWM modulation function for
CH1.*/
:SOURce1:PWM:STATe? /*Queries the on/off status of the PWM
modulation function for CH1. The query returns 1.*/
```

3.11.16 [:SOURce[<n>]]:SUM

[:SOURce[<n>]]:SUM commands are used to set and query the waveform summing parameters such as sum ratio, sum frequency, sum waveform, and the summing on/off status.

3.11.16.1 [:SOURce[<n>]]:SUM:AMPLitude

Syntax

[:SOURce[<*n*>]]:SUM:AMPLitude {<*amplitude*>|<*lim_set*>}

[:SOURce[<*n*>]]:SUM:AMPLitude? [<*lim_query*>]

Description

Sets or queries the sum ratio of the waveform summing function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<amplitude>	Real	0% to 100%	50%
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Sum ratio is the ratio of the amplitude of the waveform to be added relative to the amplitude of the basic waveform.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the sum ratio in scientific notation. For example, the query might return +5.000000000000000E+01, indicating that the sum ratio is 50%.

Examples

```
:SOURce1:SUM:AMPLitude 50 /*Sets the sum ratio of the waveform
summing function to 50% for CH1.*/
:SOURce1:SUM:AMPLitude? /*Queries the sum ratio of the waveform
summing function for CH1. The query returns +5.000000000000000E
+01.*/
```

3.11.16.2 [:SOURce[<n>]]:SUM:INTernal:FREQuency

Syntax

[:SOURce[<*n*>]]:SUM:INTernal:FREQuency {<*frequency*>|<*lim_set*>}

[:SOURce[<*n*>]]:SUM:INTernal:FREQuency? [<*lim_query*>]

Description

Sets or queries the sum frequency of the waveform summing function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<frequency>	Real	2 mHz to 1 MHz	100 Hz
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Sum frequency is the frequency of the waveform to be added to the current basic waveform.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the frequency in scientific notation. For example, the query might return +1.000000000000000E+02, indicating that the sum frequency is 100 Hz.

Examples

```
:SOURce1:SUM:INTernal:FREQuency 100 /*Sets the sum frequency of
the waveform summing function to 100 Hz for CH1.*/
:SOURce1:SUM:INTernal:FREQuency? /*Queries the sum frequency of
the waveform summing function for CH1. The query returns
+1.000000000000000E+02.*/
```

3.11.16.3 [:SOURce[<n>]]:SUM:INTernal:FUNCtion

Syntax

[:SOURce[<*n*>]]:SUM:INTernal:FUNCtion <*function*>

[:SOURce[<*n*>]]:SUM:INTernal:FUNCtion?

Description

Sets or queries the summing waveform of the waveform summing function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<function>	Discrete	{SINusoid|SQUare|RAMP|NOISe|ARB}	SINusoid

Remarks

- Summing waveform is the waveform to be added to the basic waveform. It can be set to Sine (SINusoid), Square (SQUare), Ramp (RAMP), Noise (NOISe), or Arb (ARB).
- This command is only available for internal summing source (*[:SOURce[<n>]]:SUM:SOURce*).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns SIN, SQU, RAMP, NOIS, or ARB.

Examples

```
:SOURce1:SUM:INTernal:FUNCtion SQUare /*Sets the summing waveform
of the waveform summing function to Square for CH1.*/
:SOURce1:SUM:INTernal:FUNCtion? /*Queries the summing waveform of
the waveform summing function for CH1. The query returns SQU.*/
```

3.11.16.4 [:SOURce[<n>]]:SUM:SOURce

Syntax

[:SOURce[<*n*>]]:SUM:SOURce <*source*>

[:SOURce[<*n*>]]:SUM:SOURce?

Description

Sets or queries the source of the waveform summing function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{INTernal|CH1|CH2}	INTernal

Remarks

- **INTernal:** sets the source to internal.
- **CH1:** sets the source to the waveform of CH1.
- **CH2:** sets the source to the waveform of CH2.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns INT, CH1, or CH2.

Examples

```
:SOURce1:SUM:SOURce INTernal /*Sets the source of the waveform
summing function to internal for CH1.*/
:SOURce1:SUM:SOURce? /*Queries the source of the waveform summing
function for CH1. The query returns INT.*/
```

3.11.16.5 [:SOURce[<n>]]:SUM:STATe

Syntax

[:SOURce[<*n*>]]:SUM:STATe <*bool*>

[:SOURce[<*n*>]]:SUM:STATe?

Description

Sets or queries the on/off status of the waveform summing function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{ON|1|OFF|0}	0|OFF

Remarks

- Waveform summing function is to sum the specified waveform on the basic waveform like Sine, Square, Ramp, and Arb (except DC) and then output the summed waveform.
- If the Sweep (*[:SOURce[<n>]]:SWEep:STATe*), Burst (*[:SOURce[<n>]]:BURSt:STATe*), or Sequence (*[:SOURce[<n>]]:FUNCtion:SEQuence[:STATe]*) function is currently enabled, it will be disabled automatically when the waveform summing function is enabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:SUM:STATe ON /*Enables the waveform summing function for
CH1.*/
:SOURce1:SUM:STATe? /*Queries the on/off status of the waveform
summing function for CH1. The query returns 1.*/
```

3.11.17 [:SOURce[<n>]]:SWEep

[:SOURce[<n>]]:SWEep are used to set and query sweep parameters including start hold time, stop hold time, return time, sweep type, number of steps, sweep time, and the on/off status of the sweep function.

3.11.17.1 [:SOURce[<n>]]:SWEep:HTIMe:STARt

Syntax

[:SOURce[<*n*>]]:SWEep:HTIMe:STARt {<*time*>|<*lim_set*>}

[:SOURce[<*n*>]]:SWEep:HTIMe:STARt? [<*lim_query*>]

Description

Sets or queries the start hold time of the sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<time>	Real	0 s to 3600 s	0 s
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-

Name	Type	Range	Default
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The maximum value of <time> is limited by the sweep type, trigger source, return time, sweep time, and stop hold time. T_P = Sweep Time + Start Hold Time + Return Time + Stop Hold Time.
  - **Linear Sweep:** internal trigger source, T_P + 1 ms ≤ 8,000 s; manual/ external trigger source, T_P ≤ 250,000 s.
  - **Log/Step Sweep:** T_P ≤ 500 s.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the start hold time in scientific notation. For example, the query might return +1.000000000000000E-03, indicating that the start hold time is 1 ms.

Examples

```
:SOURce1:SWEep:HTIMe:STARt 0.001 /*Sets the start hold time of the
sweep function to 1 ms for CH1.*/
:SOURce1:SWEep:HTIMe:STARt? /*Queries the start hold time of the
sweep function for CH1. The query returns +1.000000000000000E-03.*/
```

3.11.17.2 [:SOURce[<n>]]:SWEep:HTIMe[:STOP]

Syntax

[:SOURce[<*n*>]]:SWEep:HTIMe[:STOP] {<*time*>|<*lim_set*>}

[:SOURce[<*n*>]]:SWEep:HTIMe[:STOP]? [<*lim_query*>]

Description

Sets or queries the stop hold time of the sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<time>	Real	0 s to 3600 s	0 s
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The maximum value of <time> is limited by the sweep type, trigger source, return time, sweep time, and start hold time. T_P = Sweep Time + Start Hold Time + Return Time + Stop Hold Time.
  - **Linear Sweep:** internal trigger source, T_P + 1 ms ≤ 8,000 s; manual/ external trigger source, T_P ≤ 250,000 s.
  - **Log/Step Sweep:** T_P ≤ 500 s.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the stop hold time in scientific notation. For example, the query might return +1.000000000000000E-03, indicating that the stop hold time is 1 ms.

Examples

```
:SOURce1:SWEep:HTIMe:STOP 0.001 /*Sets the stop hold time of the
sweep function to 1 ms for CH1.*/
:SOURce1:SWEep:HTIMe:STOP? /*Queries the stop hold time of the
sweep function for CH1. The query returns +1.000000000000000E-03.*/
```

3.11.17.3 [:SOURce[<n>]]:SWEep:RTIMe

Syntax

[:SOURce[<*n*>]]:SWEep:RTIMe {<*time*>|<*lim_set*>}

[:SOURce[<*n*>]]:SWEep:RTIMe? [<*lim_query*>]

Description

Sets or queries the return time of the sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<time>	Real	0 s to 3600 s	0 s
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The maximum value of <time> is limited by the sweep type, start/stop hold time, trigger source, and sweep time. T_P = Sweep Time + Start Hold Time + Return Time + Stop Hold Time.

- **Linear Sweep:** internal trigger source, T_P + 1 ms ≤ 8,000 s; manual/ external trigger source, T_P ≤ 250,000 s.
- **Log/Step Sweep:** T_P ≤ 500 s.

• When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the return time in scientific notation. For example, the query might return +1.000000000000000E-03, indicating that the return time is 1 ms.

Examples

```
:SOURce1:SWEep:RTIMe 0.001 /*Sets the return time of the sweep
function to 1 ms for CH1.*/
:SOURce1:SWEep:RTIMe? /*Queries the return time of the sweep
function for CH1. The query returns +1.000000000000000E-03.*/
```

3.11.17.4 [:SOURce[<n>]]:SWEep:SPACing

Syntax

[:SOURce[<*n*>]]:SWEep:SPACing <*type*>

[:SOURce[<*n*>]]:SWEep:SPACing?

Description

Sets or queries the sweep type for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<type>	Discrete	{LINear|LOGarithmic|STEP}	LINear

Remarks

• When you enable the sweep function for the specified channel, you can use this command to set the sweep type. This instrument provides the following three sweep types.

- **LINear:** linear sweep. The instrument varies the output frequency linearly during the sweep, changing the output frequency by a constant number of Hz per second. It is characterized by "Start Frequency" (*[:SOURce[<n>]]:FREQuency:STARt*), "Stop Frequency" (*[:SOURce[<n>]]:FREQuency:STOP*), and "Sweep Time" (*[:SOURce[<n>]]:SWEep:TIME*).
- **LOGarithmic:** logarithmic sweep. The instrument varies the output frequency logarithmically during the sweep, changing the frequency by a

constant number of octaves. It is characterized by "Start Frequency", "Stop Frequency", and "Sweep Time".

- **STEP:** step sweep. The instrument "steps" through a list of frequencies during the sweep. The period that the output signal dwells on each frequency is determined by "Sweep Time" and "Step Number" (*[:SOURce[<n>]]:SWEep:STEP*).

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns LIN, LOG, or STEP.

Examples

```
:SOURce1:SWEep:SPACing LINear /*Sets the sweep type to linear
sweep for CH1.*/
:SOURce1:SWEep:SPACing? /*Queries the sweep type for CH1. The
query returns LIN.*/
```

3.11.17.5 [:SOURce[<n>]]:SWEep:STATe

Syntax

[:SOURce[<*n*>]]:SWEep:STATe <*bool*>

[:SOURce[<*n*>]]:SWEep:STATe?

Description

Sets or queries the on/off status of the Sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- If the Modulation, Sequence, or Burst function is currently enabled, it will be disabled automatically when the Sweep function is enabled.
- When the basic waveform (*[:SOURce[<n>]]:FUNCtion*) of the specified channel is set to harmonic, noise, DC, or pulse, the Sweep function cannot be enabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:SWEep:STATe ON /*Enables the Sweep function for CH1.*/
:SOURce1:SWEep:STATe? /*Queries the on/off status of the Sweep
function for CH1. The query returns 1.*/
```

3.11.17.6 [:SOURce[<n>]]:SWEep:STEP

Syntax

[:SOURce[<*n*>]]:SWEep:STEP {<*step*>|<*lim*>}

[:SOURce[<*n*>]]:SWEep:STEP? [<*lim*>]

Description

Sets or queries the step number of the step sweep function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<step>	Integer	2 to 1024	2
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- Step number specifies the number of steps to sweep from "Start Frequency" (*[:SOURce[<n>]]:FREQuency:STARt*) to "Stop Frequency" (*[:SOURce[<n>]]:FREQuency:STOP*). It is only available for step sweep.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the step number in scientific notation. For example, the query might return 5.000000000000000E+02, indicating that the step number is 500.

Examples

```
:SOURce1:SWEep:STEP 500 /*Sets the step number of the step sweep
function to 500 for CH1.*/
:SOURce1:SWEep:STEP? /*Queries the step number of the step sweep
function for CH1. The query returns 5.000000000000000E+02.*/
```

3.11.17.7 [:SOURce[<n>]]:SWEep:TIME

Syntax

[:SOURce[<*n*>]]:SWEep:TIME {<*time*>|<*lim_set*>}

[:SOURce[<*n*>]]:SWEep:TIME? [<*lim_query*>]

Description

Sets or queries the sweep time for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<time>	Real	1 ms to 250,000 s	1 s
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The maximum value of <time> is limited by the sweep type, start/stop hold time, trigger source, and return time. T_P = Sweep Time + Start Hold Time + Return Time + Stop Hold Time.
  - **Linear Sweep:** internal trigger source, T_P + 1 ms ≤ 8,000 s; manual/external trigger source, T_P ≤ 250,000 s.
  - **Log/Step Sweep:** T_P ≤ 500 s.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the sweep time in scientific notation. For example, the query might return +5.000000000000000E+00, indicating that the sweep time is 5 s.

Examples

```
:SOURce1:SWEep:TIME 5 /*Sets the sweep time to 5 s for CH1.*/
:SOURce1:SWEep:TIME? /*Queries the sweep time for CH1. The query
returns +5.000000000000000E+00.*/
```

3.11.18 [:SOURce[<n>]][:TRACe]:DATA:DAC16

Syntax

[:SOURce[<*n*>]][:TRACe]:DATA:DAC16 <*type*>,<*flag*>,<*data*>

Description

Downloads the Arb data to the volatile memory of the specified channel.

Parameter

Name	Type	Range	Default
<type>	Discrete	{CODE|VOLTage|BIN}	-
<flag>	Discrete	{HEADer|CONTinue|END}	-
<data>	ASCII string or IEEE 488.2 block	Refer to *Remarks*	-

Remarks

- <type> specifies the data type.
  - **CODE:** waveform point data. It is a decimal integer ranging from -32768 to +32767.
  - **VOLTage:** voltage data in floating point. The input data will be normalized.
  - **BIN:** binary code value, 2 bytes in a code value.
- <flag> specifies the data transmission status.
  - **HEADer** indicates discarding the previous data and using this data as the beginning of a waveform data package.
  - **CONTinue** indicates that there is data package following this data package.
  - **END** indicates that this is the last data package and the data transmission finishes. If you want to send only one data package, you must use "END".
- <Data> is the data to be downloaded. The binary code value is transmitted in binary bit stream; non-binary code value is transmitted in waveform point data or voltage data separated by comma. For example, the waveform point data can be 10,20,30,40,50,60,70,80,90,100,200,300,400,500,600,700,800,900,1000,1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,2100,2200,2300...
- It is recommended to send data (<data>) within 20 kbytes at a time. You can send data multiple times until the max. memory depth is reached. The max. memory depth varies for different series:
  - DG800 Pro: 2 Mpts (8 Mpts/CH optional).
  - DG900 Pro: 16 Mpts (32 Mpts/CH optional).
- The waveform table data is set under the first step of the sequence for the specified channel.
- It might take a while to download the long waveform table. It is recommended to use the `*OPC?` query command. Make sure that the data download is complete or data may be lost.

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

None.

Examples

```
:SOURce1:TRACe:DATA:DAC16
CODE,END,10,20,30,40,50,60,70,80,90,100,200,300,400,500,600,700,800,
900,1000,1100,1200,1300,1400,1500,1600,1700,1800,1900,2000,2100,2200
,2300 /*Downloads the waveform table data to the volatile memory of
CH1.*/
```

3.11.19 [:SOURce[<n>]]:TRACk

Syntax

[:SOURce[<*n*>]]:TRACk <*track*>

[:SOURce[<*n*>]]:TRACk?

Description

Sets or queries the state of the channel track function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<track>	Discrete	{ON|OFF|INVerted}	OFF

Remarks

- The channel track function has the following three states.
  - **ON:** enables the track function. If the track function is enabled for a specified channel, the instrument automatically copies the parameters and states (except channel output on/off state) to the other channel. When the parameters or states are modified, the modifications also apply to the other channel. In this way, the two channels can output identical signals (channel enabled).
  - **OFF:** disables the track function.
  - **INVerted:** the track function is enabled, but the output polarity of the target channel is opposite to that of the source channel.
- The track function is not available for DG821 Pro (unless the two-channel upgrade option is installed).

- The Track function is disabled when the Sequence mode, Counter, or front port sync output is enabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns ON, OFF, or INV.

Examples

```
:SOURce1:TRACk ON /*Enables the channel track function for CH1.*/
:SOURce1:TRACk? /*Queries the channel track function for CH1. The
query returns ON.*/
```

3.11.20 [:SOURce[<n>]]:VOLTage

[:SOURce[<n>]]:VOLTage commands are used to set and query the on/off status of amplitude coupling function, specify the waveform amplitude, waveform high/low level, waveform offset voltage, and amplitude unit for the specified channel.

3.11.20.1 [:SOURce[<n>]]:VOLTage

Syntax

[:SOURce[<*n*>]]:VOLTage {<*amplitude*>|<*lim_set*>}

[:SOURce[<*n*>]]:VOLTage? [<*lim_query*>]

Description

Sets or queries the output amplitude for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<amplitude>	Real	Refer to *Remarks*	5 Vpp
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The maximum value of <amplitude> is limited by the impedance (*:OUTPut[<n>]:LOAD*) and frequency (*[:SOURce[<n>]]:FREQuency*). For details, refer to *Table 3.62 Range of Amplitude* . Define the amplitude upper limit to Vpp_{max}: max. output amplitude ≤ (Vpp_{max} - 2*|V_{offset}|).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the amplitude in scientific notation. For example, the query might return +2.500000000000000E+00, indicating that the output amplitude is 2.5 Vpp.

Examples

```
:SOURce1:VOLTage 2.5 /*Sets the output amplitude to 2.5 Vpp for
CH1.*/
:SOURce1:VOLTage? /*Queries the output amplitude for CH1. The
query returns +2.500000000000000E+00.*/
```

3.11.20.2 [:SOURce[<n>]]:VOLTage:COUPle:MODE

Syntax

[:SOURce[<*n*>]]:VOLTage:COUPle:MODE <*mode*>

[:SOURce[<*n*>]]:VOLTage:COUPle:MODE?

Description

Sets or queries the type of the amplitude coupling mode for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<mode>	Discrete	{OFFSet|RATio}	RATio

Remarks

- The amplitude coupling mode can be set to the following:
  - **OFFSet:** amplitude deviation mode. You can use *[:SOURce[<n>]]:VOLTage:COUPle:OFFSet* to set the amplitude deviation of the coupled channel.
  - **RATio:** amplitude ratio mode. You can use *[:SOURce[<n>]]:VOLTage:COUPle:RATio* to set the amplitude ratio of the coupled channel.
- When the coupling is enabled, if switching to ratio coupling from deviation coupling causes either channel to exceed the amplitude upper/lower limit after calculation according to coupling rules, the generator will automatically adjust the parameters to avoid parameter overlimit. If the parameter overlimit is caused by switching to deviation coupling from ratio coupling, the amplitude coupling will be automatically disabled.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns OFFS or RAT.

Examples

```
:SOURce1:VOLTage:COUPle:MODE OFFSet /*Sets the type of the
amplitude coupling mode to amplitude deviation for CH1.*/
:SOURce1:VOLTage:COUPle:MODE /*Queries the type of the amplitude
coupling mode for CH1. The query returns OFFS.*/
```

3.11.20.3 [:SOURce[<n>]]:VOLTage:COUPle:OFFSet

Syntax

[:SOURce[<*n*>]]:VOLTage:COUPle:OFFSet <*voltage*>

[:SOURce[<*n*>]]:VOLTage:COUPle:OFFSet?

Description

Sets or queries the amplitude deviation of the amplitude coupling for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<voltage>	Real	-20 Vpp to +20 Vpp	0 Vpp

Remarks

- When modifying the deviation, if base channel parameters exceed the limits of the coupled channel after coupling calculation, the generator will automatically adjust the coupling deviation to avoid parameter overlimit.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the amplitude deviation in scientific notation. For example, the query might return +5.000000000000000E+00, indicating that the amplitude deviation is 5 Vpp.

Examples

```
:SOURce1:VOLTage:COUPle:OFFSet 5 /*Sets the amplitude deviation of
the amplitude coupling to 5 Vpp for CH1.*/
:SOURce1:VOLTage:COUPle:OFFSet? /*Queries the amplitude deviation
of the amplitude coupling for CH1. The query returns
+5.000000000000000E+00.*/
```

3.11.20.4 [:SOURce[<n>]]:VOLTage:COUPle:RATio

Syntax

[:SOURce[<*n*>]]:VOLTage:COUPle:RATio <*ratio*>

[:SOURce[<*n*>]]:VOLTage:COUPle:RATio?

Description

Sets or queries the amplitude ratio of the amplitude coupling for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<ratio>	Real	0.001 to 1000	1

Remarks

- When modifying the ratio, if base channel parameters exceed the limits of the coupled channel after coupling calculation, the generator will automatically adjust the waveform amplitude to avoid parameter overlimit.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the amplitude ratio in scientific notation. For example, the query might return +2.000000000000000E+00, indicating that the ratio is 2.

Examples

```
:SOURce1:VOLTage:COUPle:RATio 2 /*Sets the amplitude ratio of the
amplitude coupling to 2 for CH1.*/
:SOURce1:VOLTage:COUPle:RATio? /*Queries the amplitude ratio of
the amplitude coupling for CH1. The query returns
+2.000000000000000E+00.*/
```

3.11.20.5 [:SOURce[<n>]]:VOLTage:COUPle[:STATe]

Syntax

[:SOURce[<*n*>]]:VOLTage:COUPle[:STATe] <*bool*>

[:SOURce[<*n*>]]:VOLTage:COUPle[:STATe]?

Description

Sets or queries the on/off status of the amplitude coupling function for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<bool>	Bool	{0|1|OFF|ON}	0|OFF

Remarks

- The amplitude coupling function is not available for DG821 Pro (unless the two-channel upgrade option is installed).
- When the coupling mode is set to "ratio", if the frequency, amplitude, or phase of either channel exceeds the upper/lower limit after calculation according to the coupling rules due to modifying the coupling ratio, switching waveform, or changing waveform parameters, the generator will automatically adjust the waveform parameters to avoid parameter overlimit.
- When the coupling mode is set to "deviation", if the frequency, amplitude or phase of either channel exceeds the upper/lower limit after calculation according to the coupling rules due to modifying the coupling deviation, the generator will automatically adjust the coupling deviation to avoid parameter overlimit; if it is due to switching waveform or changing waveform parameters, the corresponding coupling function is disabled.
- The amplitude coupling function is available only when the output modes of both channels are set to Continuous and the output waveform is set to Sine, Square, Ramp, Noise, or Arb (except DC).
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns 0 or 1.

Examples

```
:SOURce1:VOLTage:COUPle:STATe ON /*Enables the amplitude coupling
function for CH1.*/
:SOURce1:VOLTage:COUPle:STATe? /*Queries the on/off status of the
amplitude coupling function. The query returns 1.*/
```

3.11.20.6 [:SOURce[<n>]]:VOLTage:HIGH

Syntax

[:SOURce[<*n*>]]:VOLTage:HIGH {<*voltage*>|<*lim_set*>}

[:SOURce[<*n*>]]:VOLTage:HIGH? [<*lim_query*>]

Description

Sets or queries the high level for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<voltage>	Real	Refer to *Remarks*	2.5 V
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Define the amplitude upper/lower limit to Vpp_{max}/Vpp_{min}. The range of the high level should conform to the following restrictions:

  |High Level| ≤ Vpp_{max}/2, and Vpp_{min} ≤ (High Level - Low Level) ≤ Vpp_{max}

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the high level in scientific notation. For example, the query might return +2.000000000000000E+00, indicating that the high level is 2 V.

Examples

```
:SOURce1:VOLTage:HIGH 2 /*Sets the high level to 2 V for CH1.*/
:SOURce1:VOLTage:HIGH? /*Queries the high level for CH1. The query
returns +2.000000000000000E+00.*/
```

3.11.20.7 [:SOURce[<n>]]:VOLTage:LOW

Syntax

[:SOURce[<*n*>]]:VOLTage:LOW {<*voltage*>|<*lim_set*>}

[:SOURce[<*n*>]]:VOLTage:LOW? [<*lim_query*>]

Description

Sets or queries the low level for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<voltage>	Real	Refer to *Remarks*	-2.5 V
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-

Name	Type	Range	Default
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Define the amplitude upper/lower limit (*Table 3.62 Range of Amplitude*) to Vpp_{max}/Vpp_{min}. The range of the low level should conform to the following restrictions:

  |Low Level| ≤ Vpp_{max}/2, and Vpp_{min} ≤ (High Level - Low Level) ≤ Vpp_{max}

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the low level in scientific notation. For example, the query might return -1.000000000000000E+00, indicating that the low level is -1 V.

Examples

```
:SOURce1:VOLTage:LOW -1 /*Sets the low level to -1 V for CH1.*/
:SOURce1:VOLTage:LOW? /*Queries the low level for CH1. The query
returns -1.000000000000000E+00.*/
```

3.11.20.8 [:SOURce[<n>]]:VOLTage:OFFSet

Syntax

[:SOURce[<*n*>]]:VOLTage:OFFSet {<*voltage*>|<*lim_set*>}

[:SOURce[<*n*>]]:VOLTage:OFFSet? [<*lim_query*>]

Description

Sets or queries the offset voltage for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<voltage>	Real	Refer to *Remarks*	0 Vdc
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- The parameter <voltage> is affected by the amplitude setting (*[:SOURce[<n>]]:VOLTage*). |Offset|*2 plus the current amplitude cannot exceed the upper limit of the amplitude (*Table 3.62 Range of Amplitude*).

- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns the offset voltage in scientific notation. For example, the query might return +5.000000000000000E-01, indicating that the offset value is 0.5 Vdc.

Examples

```
:SOURce1:VOLTage:OFFSet 0.5 /*Sets the offset voltage to 0.5 Vdc
for CH1.*/
:SOURce1:VOLTage:OFFSet? /*Queries the offset voltage for CH1. The
query returns +5.000000000000000E-01.*/
```

3.11.20.9 [:SOURce[<n>]]:VOLTage:UNIT

Syntax

[:SOURce[<*n*>]]:VOLTage:UNIT <*unit*>

[:SOURce[<*n*>]]:VOLTage:UNIT?

Description

Sets or queries the amplitude unit for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<unit>	Discrete	{VPP|VRMS|DBM}	VPP

Remarks

- VPP: indicates the signal peak-to-peak voltage (Vpp); VRMS: indicates the root-mean-square voltage (Vrms); DBM: expresses the absolute power level (dBm). dBm is not available when the output impedance is set to HighZ.
- When [:SOURce[<n>]] or [<n>] is omitted, it is interpreted as CH1.

Return Format

The query returns VPP, VRMS, or DBM.

Examples

```
:SOURce1:VOLTage:UNIT VPP /*Sets the amplitude unit to Vpp for
CH1.*/
:SOURce1:VOLTage:UNIT? /*Queries the amplitude unit for CH1. The
query returns VPP.*/
```

3.12 :SYSTem Commands

:SYSTem commands are used to set or query system parameters.

3.12.1 :SYSTem:BEEPer[:IMMediate]

Syntax

:SYSTem:BEEPer[**:IMMediate**]

Description

Issues a single beep immediately.

Parameter

None.

Remarks

This command is valid regardless of the beeper on/off state. This command issues a beep immediately even though the beeper is turned off.

Return Format

None.

Examples

```
:SYSTem:BEEPer:IMMediate /*Issues a single beep immediately.*/
```

3.12.2 :SYSTem:BEEPer:STATe

Syntax

:SYSTem:BEEPer:STATe <*state*>

:SYSTem:BEEPer:STATe?

Description

Sets or queries the on/off status of the beeper.

Parameter

Name	Type	Range	Default
<state>	Bool	{1|ON|0|OFF}	1|ON

Remarks

None.

Return Format

The query returns 1 or 0.

Examples

```
:SYSTem:BEEPer:STATe 1 /*Enables the beeper.*/
:SYSTem:BEEPer:STATe? /*Queries the on/off status of the beeper.
The query returns 1.*/
```

3.12.3 :SYSTem:DATE

Syntax

:SYSTem:DATE <*yyyy*>,<*mm*>,<*dd*>

:SYSTem:DATE?

Description

Sets or queries the system date.

Parameter

Name	Type	Range	Default
<yyyy>	Integer	1970 to 2037	-
<mm>	Integer	01 to 12	-
<dd>	Integer	01 to 31 (28, 29, or 30)	-

Remarks

None.

Return Format

The query returns the system date in character string. The year, month, and day are separated by "-".

Examples

```
:SYSTem:DATE 2022,05,01 /*Sets the system date to May 1, 2022.*/
:SYSTem:DATE? /*Queries the system date. The query returns
2022-05-01.*/
```

3.12.4 :SYSTem:COMMunicate

3.12.4.1 :SYSTem:COMMunicate:LAN:CONTrol?

Syntax

:SYSTem:COMMunicate:LAN:CONTrol?

Description

Reads the port number of the initial control connecting port for socket communication.

Parameter

None.

Remarks

None.

Return Format

The query returns the port number; if the interface does not support the socket, it returns 0.

Examples

```
:SYSTem:COMMunicate:LAN:CONTrol? /*Reads the port number of the initial control connecting port for socket communication. The query returns 5000.*/
```

3.12.4.2 :SYSTem:COMMunicate:LAN:DHCP[:STATe]

Syntax

:SYSTem:COMMunicate:LAN:DHCP[:STATe] <*bool*>

:SYSTem:COMMunicate:LAN:DHCP[:STATe]?

Description

Sets or queries the on/off status of the DHCP mode.

Parameter

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

Remarks

- In DHCP mode, it automatically assigns an IP address to the instrument from a DHCP server.
- When the three IP configuration types are all turned on, the priority of the parameter configuration from high to low is "DHCP", "Auto IP", and "Static IP".
- The three IP configuration types cannot be all turned off at the same time.
- After sending this command, execute *:SYSTem:COMMunicate:LAN:UPDate* to apply the current network setting for the new parameters to take effect.

Return Format

The query returns 0 or 1.

Examples

```
:SYSTem:COMMunicate:LAN:DHCP:STATe ON /*Enables the DHCP mode.*/
:SYSTem:COMMunicate:LAN:DHCP:STATe? /*Queries the on/off status of
the DHCP mode. The query returns 1.*/
```

3.12.4.3 :SYSTem:COMMunicate:LAN:DNS

Syntax

:SYSTem:COMMunicate:LAN:DNS <*dns*>

:SYSTem:COMMunicate:LAN:DNS?

Description

Sets or queries the domain name server (DNS) address.

Parameter

Name	Type	Range	Default
<dns>	ASCII string	Refer to *Remarks*	-

Remarks

- The format of <dns> is nnn.nnn.nnn.nnn; wherein, the first nnn ranges from 1 to 223 (excluding 127), and the other three range from 0 to 255.
- You are recommended to ask your network administrator for an address available.
- When Static IP mode is enabled with both DHCP and Auto IP disabled, the DNS address is used.
- After sending this command, execute *:SYSTem:COMMunicate:LAN:UPDate* to apply the current network setting for the new parameters to take effect.

Return Format

The query returns the DNS address, for example, 172.16.3.2.

Examples

```
:SYSTem:COMMunicate:LAN:DNS 172.16.3.2 /*Sets the DNS address to
172.16.3.2.*/
:SYSTem:COMMunicate:LAN:DNS? /*Queries the current DNS address.
The query returns 172.16.3.2.*/
```

3.12.4.4 :SYSTem:COMMunicate:LAN:DOMain?

Syntax

:SYSTem:COMMunicate:LAN:DOMain?

Description

Queries the domain name.

Parameter

None.

Remarks

None.

Return Format

The query returns a string.

Examples

```
:SYSTem:COMMunicate:LAN:DOMain? /*Queries the domain name. The
query returns YYYrigollan.*/
```

3.12.4.5 :SYSTem:COMMunicate:LAN:GATeway

Syntax

:SYSTem:COMMunicate:LAN:GATeway <*gateway*>

:SYSTem:COMMunicate:LAN:GATeway?

Description

Sets or queries the default gateway.

Parameter

Name	Type	Range	Default
<gateway>	ASCII string	Refer to *Remarks*	-

Remarks

- The format of <gateway> is nnn.nnn.nnn.nnn; wherein, the first nnn ranges from 1 to 223 (excluding 127), and the other three range from 0 to 255.
- You are recommended to ask your network administrator for a gateway available.
- When Static IP mode is enabled with both DHCP and Auto IP disabled, this default gateway is used.

- After sending this command, execute *:SYSTem:COMMunicate:LAN:UPDate* to apply the current network setting for the new parameters to take effect.

Return Format

The query returns a string, for example, 172.16.3.1.

Examples

```
:SYSTem:COMMunicate:LAN:GATeway 172.16.3.1 /*Sets the default
gateway to 172.16.3.1.*/
:SYSTem:COMMunicate:LAN:GATeway? /*Queries the current default
gateway. The query returns 172.16.3.1.*/
```

3.12.4.6 :SYSTem:COMMunicate:LAN:HOSTname

Syntax

:SYSTem:COMMunicate:LAN:HOSTname <*name*>

:SYSTem:COMMunicate:LAN:HOSTname?

Description

Sets or queries the hostname.

Parameter

Name	Type	Range	Default
<name>	ASCII string	Refer to *Remarks*	-

Remarks

The parameter <name> is the specified hostname with a maximum length of 15 characters. The name must start with the letter from A to Z. It can contain letters, numbers (0-9), and hyphens (-).

Return Format

The query returns a string.

Examples

```
:SYSTem:COMMunicate:LAN:HOSTname RIGOL123 /*Sets the hostname to
RIGOL123.*/
:SYSTem:COMMunicate:LAN:HOSTname? /*Queries the hostname. The
query returns RIGOL123.*/
```

3.12.4.7 :SYSTem:COMMunicate:LAN:IPADdress

Syntax

:SYSTem:COMMunicate:LAN:IPADdress <*ip*>

:SYSTem:COMMunicate:LAN:IPADdress?

Description

Sets or queries the IP address.

Parameter

Name	Type	Range	Default
<ip>	ASCII string	Refer to *Remarks*	-

Remarks

- The format of <ip> is nnn.nnn.nnn.nnn; wherein, the first nnn ranges from 1 to 223 (excluding 127), and the other three range from 0 to 255.
- You are recommended to ask your network administrator for an address available.
- When Static IP mode is enabled with both DHCP and Auto IP disabled, the IP address is used.
- After sending this command, execute *:SYSTem:COMMunicate:LAN:UPDate* to apply the current network setting for the new parameters to take effect.

Return Format

The query returns the IP address, for example, 172.16.3.128.

Examples

```
:SYSTem:COMMunicate:LAN:IPADdress 172.16.3.128 /*Sets the IP
address to 172.16.3.128.*/
:SYSTem:COMMunicate:LAN:IPADdress? /*Queries the current IP
address. The query returns 172.16.3.128.*/
```

3.12.4.8 :SYSTem:COMMunicate:LAN:MAC?

Syntax

`:SYSTem:COMMunicate:LAN:MAC?`

Description

Queries the MAC address.

Parameter

None.

Remarks

A MAC (Media Access Control) address, also referred to as a hardware address, is used to identify the network device address. For each instrument, the MAC address is unique. When assigning the IP address for the instrument, the system uses the MAC address to identify the instrument. The MAC address (48 bits, namely 6 bytes) is usually expressed in hexadecimal format, such as 00:2A:A0:AA:E0:56.

Return Format

The query returns the MAC address, for example, 00:2A:A0:AA:E0:56.

Examples

```
:SYSTem:COMMunicate:LAN:MAC? /*Queries the MAC address. The query
returns 00:2A:A0:AA:E0:56.*/
```

3.12.4.9 :SYSTem:COMMunicate:LAN:SMASk

Syntax

:SYSTem:COMMunicate:LAN:SMASk <*submask*>

:SYSTem:COMMunicate:LAN:SMASk?

Description

Sets or queries the subnet mask.

Parameter

Name	Type	Range	Default
<submask>	ASCII string	Refer to *Remarks*	-

Remarks

- The format of <submask> is nnn.nnn.nnn.nnn.; wherein, the range of "nnn" is from 0 to 255.
- You are recommended to ask your network administrator for a subnet mask available.
- When Static IP mode is enabled with both DHCP and Auto IP disabled, the subnet mask is used.
- After sending this command, execute *:SYSTem:COMMunicate:LAN:UPDate* to apply the current network setting for the new parameters to take effect.

Return Format

The query returns the subnet mask, such as 255.255.255.0.

Examples

```
:SYSTem:COMMunicate:LAN:SMASk 255.255.255.0 /*Sets the subnet mask
to 255.255.255.0.*/
:SYSTem:COMMunicate:LAN:SMASk? /*Queries the current subnet mask.
The query returns 255.255.255.0.*/
```

3.12.4.10 :SYSTem:COMMunicate:LAN:UPDate

Syntax

:SYSTem:COMMunicate:LAN:UPDate

Description

Stores all the modifications of the LAN setting to the non-volatile memory and restarts the LAN drive program using the updated setting.

Parameter

None.

Remarks

- You must send this command after modifying the DHCP, DNS, gateway, IP address, and subnet mask settings.
- Complete all the LAN settings before sending this command.

Return Format

None.

Examples

```
:SYSTem:COMMunicate:LAN:UPDate /*Stores all the modifications of
the LAN setting to the non-volatile memory and restarts the LAN
drive program using the updated setting.*/
```

3.12.4.11 :SYSTem:COMMunicate:USB:INFormation?

Syntax

`:SYSTem:COMMunicate:USB:INFormation?`

Description

Queries the instrument's USB information.

Parameter

None.

Remarks

None.

Return Format

The query returns a string, for example, USB0::0x1AB1::0x052::DG80000000001::INSTR.

Examples

```
:SYSTem:COMMunicate:USB:INFormation? /*Queries the instrument's
USB information. The query returns
USB0::0x1AB1::0x052::DG80000000001::INSTR.*/
```

3.12.5 :SYSTem:ERRor?

Syntax

:SYSTem:ERRor?

Description

Queries and clears an error message in the error queue.

Parameter

None.

Remarks

When you read the error queue, the error will be cleared. You can also clear the error queue by using **CLS* or turning on/off the instrument.

Return Format

The query returns a string consisting of two parts separated by commas. The first part is the number of the error message and the second part is the content of the error message; wherein, the content is a string enclosed in double quotation marks. For example, the query might return -109,"Missing parameter"; wherein, -109 is the number of the error message and Missing parameter (the content enclosed in the double quotation marks) is the content of the error message.

Examples

```
:SYSTem:ERRor? /*Queries and clears an error message in the error
queue. The query returns -109,"Missing parameter".*/
```

3.12.6 :SYSTem:KLOCk

Syntax

:SYSTem:KLOCk <*bool*>

:SYSTem:KLOCk?

Description

Sets or queries the status (locked or unlocked) of all front-panel keys.

Parameter

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

Remarks

- This instrument allows you to lock all knobs and keys on the front panel to avoid danger due to misoperation.

- <bool> is used to lock or unlock the keys. ON or 1 indicates locking all the keys; OFF or 0 indicates unlocking all the keys.

Return Format

The query returns 0 or 1.

Examples

```
:SYSTem:KLOCk ON /*Locks all front-panel keys.*/
:SYSTem:KLOCk?  /*Queries whether all front-panel keys are locked.
The query returns 1.*/
```

3.12.7 :SYSTem:LANGuage

Syntax

:SYSTem:LANGuage <*language*>

:SYSTem:LANGuage?

Description

Sets or queries the system language.

Parameter

Name	Type	Range	Default
<language>	Discrete	{SCHinese|ENGLish}	-

Remarks

The language settings are not affected by factory default settings (**RST*).

Return Format

The query returns SCH or ENGL.

Examples

```
:SYSTem:LANGuage ENGLish /*Sets the system language to English.*/
:SYSTem:LANGuage? /*Queries the system language. The query returns
ENGL.*/
```

3.12.8 :SYSTem:LICense:CATalog?

Syntax

:SYSTem:LICense:CATalog?

Description

Queries the options currently activated.

Parameter

None.

Remarks

None.

Return Format

The query returns the options currently activated such as "CHD,MEM". The query returns "" when there is no activated option.

Examples

```
:SYSTem:LICense:CATalog? /*Queries the option list. The query
returns "CHD,MEM".*/
```

3.12.9 :SYSTem:LICense:DELete

Syntax

:SYSTem:LICense:DELete <*name*>

Description

Deletes the installed option.

Parameter

Name	Type	Range	Default
<name>	Discrete	{CHD|MEM}	-

Remarks

- **CHD:** two-channel upgrade option (only for DG821 Pro).
- **MEM:** memory depth upgrade option. For DG800 Pro series, the default memory depth is 2 Mpts/CH which can be upgraded to 8 Mpts/CH with the MEM option installed. For DG900 Pro series, the default memory depth is 16 Mpts/CH which can be upgraded to 32 Mpts/CH with the MEM option installed.

Return Format

None.

Examples

```
:SYSTem:LICense:DELete MEM /*Deletes the installed memory depth
upgrade option.*/
```

3.12.10 :SYSTem:LICense:DELete:ALL

Syntax

:SYSTem:LICense:DELete:ALL

Description

Deletes all the installed options.

Parameter

None.

Remarks

None.

Return Format

None.

Examples

```
:SYSTem:LICense:DELete:ALL /*Deletes all the installed options.*/
```

3.12.11 :SYSTem:LICense:INSTall

Syntax

:SYSTem:LICense:INSTall "<*license*>"

Description

Installs the option using license.

Parameter

Name	Type	Range	Default
<license>	ASCII string	Valid license	-

Remarks

- To install the option, first purchase the required option to obtain the key, and then use the key to obtain the option license according to the following steps.
  - Log in to the RIGOL official website (*http://www.rigol.com*), click **SERVICE CENTRE** > **License Activation** to enter the software license registration interface.
  - In the software license registration interface, input the correct key, serial number (click or tap  > **Utility** > **About** to obtain the serial number of the instrument), and verification code. Then click **Generate** to obtain the

option license file download link. If you need to use the file, please download it to the USB storage device.

- The license is a fixed length of strings. Each instrument has a unique license.

Return Format

None.

Examples

```
None.
```

3.12.12 :SYSTem:LICense:INSTall:UDISk

Syntax

:SYSTem:LICense:INSTall:UDISk "<*path*>"

Description

Installs the option using license file.

Parameter

Name	Type	Range	Default
<path>	ASCII string	Valid directory	-

Remarks

- To install the option, first purchase the required option to obtain the key, and then use the key to obtain the option license according to the following steps.
  - Log in to the RIGOL official website (*http://www.rigol.com*), click **SERVICE CENTRE** > **License Activation** to enter the software license registration interface.
  - In the software license registration interface, input the correct key, serial number (click or tap  > **Utility** > **About** to obtain the serial number of the instrument), and verification code. Then click **Generate** to obtain the option license file download link. If you need to use the file, please download it to the USB storage device.
- "<path>" is a valid license file path which must contain file extension (.lic). For example, it can be set to "USB:/rigol/license.lic".

Return Format

None.

Examples

```
:SYSTem:LICense:INSTall:UDISk "USB:/rigol/license.lic" /*Installs
the option using the "license.lic" file in the "rigol" folder of
the USB storage device.*/
```

3.12.13 :SYSTem:PSTatus

Syntax

:SYSTem:PSTatus <*power*>

:SYSTem:PSTatus?

Description

Sets or queries the power status of the instrument.

Parameter

Name	Type	Range	Default
<power>	Discrete	{DEFault|OPEN}	OPEN

Remarks

- **DEFault:** After the instrument is connected to power, you need to press the front-panel power key to power on the instrument.
- **OPEN**: After the instrument is connected to power, it will be powered on immediately. It saves you from pressing the power key.

Return Format

The query returns DEF or OPEN.

Examples

```
:SYSTem:PSTatus DEFault /*Sets the power status to DEFault.*/
:SYSTem:PSTatus? /*Queries the power status of the instrument. The
query returns DEF.*/
```

3.12.14 :SYSTem:TIME

Syntax

:SYSTem:TIME <*hour*>,<*minute*>,<*second*>

:SYSTem:TIME?

Description

Sets or queries the system time.

Parameter

Name	Type	Range	Default
<hour>	Integer	0 to 23	-
<minute>	Integer	0 to 59	-
<second>	Integer	0 to 59	-

Remarks

There is a certain delay between the returned time value and the set time value due to the command response time and other factors.

Return Format

The query returns the system time in string.

Examples

```
:SYSTem:TIME 16,10,17 /*Sets the system time to 16:10:17.*/
:SYSTem:TIME? /*Queries the system time. The query returns
16:10:17.*/
```

3.12.15 :SYSTem:TOUCh

Syntax

:SYSTem:TOUCh <*bool*>

:SYSTem:TOUCh?

Description

Sets or queries whether the touch screen is enabled.

Parameter

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

Remarks

- 1/ON: enables the touch screen.
- 0/OFF: disables the touch screen.

Return Format

The query returns 0 or 1.

Examples

```
:SYSTem:TOUCh OFF /*Locks the touch screen.*/
:SYSTem:TOUCh? /*Queries whether the touch screen is enabled. The
query returns 0.*/
```

3.12.16 :SYSTem:VERSion?

Syntax

:SYSTem:VERSion?

Description

Queries the version number of the SCPI used by the system.

Parameter

None.

Remarks

None.

Return Format

The query returns a string (SCPI version) in the form of "YYYY.V", where "YYYY" represents the year of the version and "V" represents a version for that year (e.g. 1994.0).

Examples

```
:SYSTem:VERSion? /*Queries the version number of the SCPI used by
the system. It might return 1994.0.*/
```

3.12.17 :SYSTem:ROSCillator:SOURce

Syntax

:SYSTem:ROSCillator:SOURce <*source*>

:SYSTem:ROSCillator:SOURce?

Description

Sets or queries the system clock source.

Parameter

Name	Type	Range	Default
<source>	Discrete	{INTernal|EXTernal}	INTernal

Remarks

- This instrument provides an internal clock source (INTernal) and also accepts the external clock source (EXTernal) from the rear-panel **[10MHz In/Out]** connector.
- If an external source is selected, the system will detect whether a valid external clock signal is applied to the rear-panel **[10MHz In/Out]** connector. If no valid clock source is detected, a prompt message would be displayed and the clock source would be switched to "Internal".

Return Format

The query returns INT or EXT.

Examples

```
:SYSTem:ROSCillator:SOURce INTernal /*Sets the system clock source
to internal source.*/
:SYSTem:ROSCillator:SOURce? /*Queries the system clock source. The
query returns INT.*/
```

3.13 :TRIGger Commands

:TRIGger commands are used to set or query the trigger mode, trigger delay, trigger period, trigger edge, and trigger count.

3.13.1 :TRIGger<n>:COUNt

Syntax

:TRIGger<*n*>**:COUNt** {<*count*>|<*lim_set*>}

:TRIGger<*n*>**:COUNt?** [<*lim_query*>]

Description

Sets or queries the trigger count for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<count>	Integer	1 to 1000000	1
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

- Trigger count is used to set the number of cycles of the effective output signals after each trigger event. After the trigger count is complete, the trigger system enters the idle state.
- It is only available for remote mode.
- The trigger count is valid only when *:INITiate[<n>]:CONTinuous* is set to 0|OFF.

Return Format

The query returns an integer.

Examples

```
:TRIGger1:COUNt 100 /*Sets the trigger count to 100.*/
:TRIGger1:COUNt? /*Queries the trigger count. The query returns
100.*/
```

3.13.2 :TRIGger<n>:DELay

Syntax

:TRIGger<*n*>**:DELay** {<*seconds*>|<*lim_set*>}

:TRIGger<*n*>**:DELay?** [<*lim_query*>]

Description

Sets or queries the trigger delay for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<seconds>	Real	Refer to *Remarks*	0 s
<lim_set>	Discrete	{MINimum|MAXimum|DEFault}	-
<lim_query>	Discrete	{MINimum|MAXimum}	-

Remarks

Trigger delay is only available for N-Cycle burst mode. The range of <seconds> is related to the burst trigger source:

- Manual/external trigger: 0 s to 20 s.
- Internal trigger: 0 s to (T_{burst} - $\lceil T_{wave} \times N_{cycle} \div 6.4\ ns \rceil \times 6.4\ ns - 4\ \mu s$)[1], and no greater than 20 s.
  - T_{burst}: burst period

- T_{wave}: period of basic waveform (e.g. Sine and Square)
- N_{cycle}: burst count

NOTE

⌈x⌉ indicates that x is rounded up.

Return Format

The query returns the trigger delay in scientific notation. For example, the query might return +1.050000000000000E-01, indicating that the trigger delay is 105 ms.

Examples

```
:TRIGger1:DELay 0.105 /*Sets the trigger delay to 105 ms for CH1.*/
:TRIGger1:DELay? /*Queries the trigger delay for CH1. The query
returns +1.050000000000000E-01.*/
```

3.13.3 :TRIGger<n>[:IMMediate]

Syntax

:TRIGger<*n*>[**:IMMediate**]

Description

Generates a trigger event in the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1

Remarks

- This command is only available for the manual trigger (*:TRIGger<n>:SOURce*) of the Burst mode (*[:SOURce[<n>]]:BURSt:STATe*) and Sweep mode (*[:SOURce[<n>]]:SWEep:STATe*).
- If the output of the corresponding channel is not enabled, the trigger will be ignored.

Return Format

None.

Examples

```
:TRIGger1:IMMediate /*Generates a trigger event in CH1.*/
```

3.13.4 :TRIGger<n>:SLOPe

Syntax

:TRIGger<*n*>**:SLOPe** <*slope*>

:TRIGger<*n*>**:SLOPe?**

Description

Sets or queries the edge type of the external trigger signal for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<slope>	Discrete	{POSitive|NEGative}	POSitive

Remarks

- This command is only available for the external trigger (*:TRIGger<n>:SOURce*) of the Burst mode (*[:SOURce[<n>]]:BURSt:STATe*) and Sweep mode (*[:SOURce[<n>]]:SWEep:STATe*). When external trigger is selected, the instrument receives the trigger signal from the rear-panel **[AUX IN/OUT]** connector. A burst or sweep is generated each time a TTL pulse with the specified polarity is received.
- The edge of the input signal can be set to leading edge (POSitive) or trailing edge (NEGative).

Return Format

The query returns POS or NEG.

Examples

```
:TRIGger1:SLOPe NEGative /*Sets the edge type of the trigger input
signal to trailing edge for CH1.*/
:TRIGger1:SLOPe? /*Queries the edge type of the trigger input
signal for CH1. The query returns NEG.*/
```

3.13.5 :TRIGger<n>:SOURce

Syntax

:TRIGger<*n*>**:SOURce** <*source*>

:TRIGger<*n*>**:SOURce?**

Description

Sets or queries the trigger type for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<source>	Discrete	{IMMediate|EXTernal|BUS|TIMer}	IMMediate

Description

- This command is only available for the Burst mode or Sweep mode.
- **IMMediate:** immediate trigger. The frequency of N-Cycle bursts is determined by the "Burst Period" (*[:SOURce[<n>]]:BURSt:INTernal:PERiod*). The trigger period of the sweep waveforms is determined by the specified sweep time (*[:SOURce[<n>]]:SWEep:TIME*), return time (*[:SOURce[<n>]]:SWEep:RTIMe*), start hold time (*[:SOURce[<n>]]:SWEep:HTIMe:STARt*), and stop hold time (*[:SOURce[<n>]]:SWEep:HTIMe[:STOP]*).
- **EXTernal:** external trigger. The instrument receives the trigger signal from the rear-panel **[AUX IN/OUT]** connector. A burst/sweep output is generated each time a TTL pulse with the specified polarity is received. You can specify the edge type (*:TRIGger<n>:SLOPe*) for trigger input signal.
- **BUS:** manual/bus trigger. Sending **TRG* or *:TRIGger[<n>][:IMMediate]* initiates a burst or sweep output immediately in the corresponding channel. If the output of the corresponding channel is not enabled, the trigger will be ignored.
- **TIMer:** timer trigger. In Burst mode, the trigger event is determined by the timer (*:TRIGger<n>:TIMer*). The trigger occurs when the instrument is in the waiting-for-trigger state. It is only available for remote mode.

Return Format

The query returns IMM, EXT, TIM, or BUS.

Examples

```
:TRIGger2:SOURce EXTernal /*Sets the trigger source to external for
CH2.*/
:TRIGger2:SOURce? /*Queries the trigger source for CH2. The query
returns EXT.*/
```

3.13.6 :TRIGger<n>:TIMer

Syntax

:TRIGger<*n*>**:TIMer** {<*timer*>|<*lim*>}

:TRIGger<*n*>**:TIMer?** [<*lim*>]

Description

Sets or queries the trigger timer for the specified channel.

Parameter

Name	Type	Range	Default
<n>	Discrete	{1|2}	1
<timer>	Real	1 μs to 8000 s	-
<lim>	Discrete	{MINimum|MAXimum}	-

Remarks

- This command is available only when the trigger type is set to timer trigger (*:TRIGger<n>:SOURce*) for the specified channel.
- The timer starts only in waiting-for-trigger state. The trigger is generated only when the timer is complete.

Return Format

The query returns the trigger timer in scientific notation. For example, the query might return 1.000000E+00, indicating that the trigger timer is 1 s.

Examples

```
:TRIGger1:TIMer 1 /*Sets the trigger timer to 1 s for CH1.*/
:TRIGger1:TIMer? /*Queries the trigger timer for CH1. The query
returns 1.000000E+00.*/
```

4 Application Examples

This chapter provides application examples of the SCPI commands. You can use a series of SCPI commands to realize the main functions of the instrument.

NOTE

1. The ranges of some parameters may differ for different instrument models. Therefore, you need to adjust the parameter range for the specified model.
2. Before using the examples in this chapter, make the correct connections using USB or LAN. In addition, you have to install the PC software for sending commands on your PC.
3. The content enclosed in "/*" and "*/" after each command in the examples of this chapter is the annotation for easier understanding and is not a part of the command.

4.1 To Output Basic Waveforms

Requirements

Use the SCPI commands to realize the following functions:

Output a sine wave (500 Hz frequency, 2.5 Vpp amplitude, 1 Vdc offset, and 90° start phase) from the CH1 output connector on the front panel.

Method 1

```
*IDN? /*Queries the ID string of the signal generator to check
whether the remote communication is normal.*/
:SOURce1:APPLy:SINusoid 500,2.5,1,90 /*Sets the CH1 waveform to
sine wave with 500 Hz frequency, 2.5 Vpp amplitude, 1 Vdc offset,
and 90° start phase.*/
:OUTPut1:STATe ON /*Enables the output of CH1.*/
```

Method 2

```
*IDN? /*Queries the ID string of the signal generator to check
whether the remote communication is normal.*/
:SOURce1:FUNCtion SINusoid /*Sets the CH1 waveform to sine wave.*/
:SOURce1:FREQuency 500 /*Sets the frequency to 500 Hz for CH1.*/
:SOURce1:VOLTage 2.5 /*Sets the amplitude to 2.5 Vpp for CH1.*/
:SOURce1:VOLTage:OFFSet 1 /*Sets the offset to 1 Vdc for CH1.*/
:SOURce1:PHASe 90 /*Sets the start phase to 90° for CH1.*/
:OUTPut1:STATe ON /*Enables the output of CH1.*/
```

4.2 To Output Arbitrary Waveforms

Requirements

Use the SCPI commands to realize the following functions:

Output an arbitrary waveform (100 Hz frequency, 1 Vpp amplitude, 2 Vdc offset, and 3° start phase) from the CH1 output connector on the front panel.

Method

```
*IDN? /*Queries the ID string of the signal generator to check
whether the remote communication is normal.*/
:SOURce1:APPLy:ARBitrary 100,1,2,3 /*Sets the CH1 waveform to an
arbitrary wave with 100 Hz frequency, 1 Vpp amplitude, 2 Vdc
offset, and 3° start phase.*/
:OUTPut1 ON /*Enables the output of CH1.*/
```

4.3 To Output Harmonic

Requirements

Use the SCPI commands to realize the following functions:

Output harmonic waveform from the front-panel output connector of CH1; the fundamental waveform parameters are 1 kHz frequency, 5 Vpp amplitude, 0 Vdc offset, and 0° start phase; output 2nd order harmonic (2 Vpp amplitude, 30° phase) and 4th order harmonic (1 Vpp amplitude, 50° phase).

Method

```
*IDN? /*Queries the ID string of the signal generator to test
whether the remote communication works normally.*/
:SOURce1:FUNCtion HARMonic /*Enables the Harmonic function for
CH1.*/
:SOURce1:FREQuency 1000 /*Sets the fundamental frequency to 1 kHz.*/
:SOURce1:VOLTage 5 /*Sets the fundamental amplitude to 5 Vpp for
CH1.*/
:SOURce1:VOLTage:OFFSet 0 /*Sets the offset to 0 Vdc for CH1.*/
:SOURce1:PHASe 0 /*Sets the start phase to 0° for CH1.*/
:SOURce1:HARMonic:TYPE COMBine /*Sets the harmonic type to Combine
for CH1.*/
:SOURce1:HARMonic:COMBine X1010000000000000000 /*Sets the combine
harmonic to 2nd order harmonic and 4th order harmonic.*/
:SOURce1:HARMonic:COMBine:AMPLitude 2,2 /*Sets the amplitude to 2
Vpp for 2nd order harmonic.*/
:SOURce1:HARMonic:COMBine:PHASe 2,30 /*Sets the phase to 30° for
2nd order harmonic.*/
:SOURce1:HARMonic:COMBine:AMPLitude 4,1 /*Sets the amplitude to 1
Vpp for 4th order harmonic.*/
:SOURce1:HARMonic:COMBine:PHASe 4,50 /*Sets the phase to 50° for
4th order harmonic.*/
:OUTPut1 ON /*Enables the output of CH1.*/
```

4.4 To Output AM Modulated Waveform

Requirements

Use the SCPI commands to realize the following functions:

Output AM modulated waveform from the front-panel output connector of CH1; set the carrier waveform to Sine (1 kHz frequency, 5 Vpp amplitude, 0 Vdc offset voltage, and 0° start phase); select internal modulation source; set the modulating waveform to Sine, the modulation depth to 80%, and the modulation frequency to 200 Hz; enable the DSSC function.

Method

```
*IDN? /*Queries the ID string of the signal generator to test
whether the remote communication works normally.*/
:SOURce1:APPLy:SINusoid 1000,5,0,0 /*Sets the waveform of CH1 to
Sine with 1kHz frequency, 5 Vpp amplitude, 0 Vdc offset, and 0°
start phase.*/
:SOURce1:AM:STATe ON /*Enables the AM function for CH1.*/
:SOURce1:AM:SOURce INTernal /*Sets the AM modulation source to
internal modulation source for CH1.*/
:SOURce1:AM:INTernal:FUNCtion SINusoid /*Sets the AM modulation
waveform to Sine for CH1.*/
:SOURce1:AM:DEPTh 80 /*Sets the AM modulation depth to 80% for
CH1.*/
:SOURce1:AM:INTernal:FREQuency 200 /*Sets the AM modulation
frequency to 200 Hz for CH1.*/
:SOURce1:AM:DSSC ON /*Enables the AM DSSC function.*/
:OUTPut1 ON /*Enables the output of CH1.*/
```

4.5 To Output FSK Modulated Waveform

Requirements

Use the SCPI commands to realize the following functions:

Output FSK modulated waveform from the front-panel output connector of CH1; set the carrier waveform to Sine (1 kHz frequency, 5 Vpp amplitude, 0 Vdc offset voltage, and 0° start phase); select external modulation source; set the hop frequency to 2 kHz and the polarity to Positive.

Method

```
*IDN? /*Queries the ID string of the signal generator to test
whether the remote communication works normally.*/
:SOURce1:APPLy:SINusoid 1000,5,0,0 /*Sets the waveform of CH1 to
Sine with 1 kHz frequency, 5 Vpp amplitude, 0 Vdc offset, and 0°
start phase.*/
:SOURce1:FSKey:STATe ON /*Enables the FSK function for CH1.*/
:SOURce1:FSKey:SOURce EXTernal /*Sets the FSK modulation source to
external modulation source for CH1.*/
:SOURce1:FSKey:FREQuency 2000 /*Sets the FSK hop frequency to 2 kHz
for CH1.*/
:SOURce1:FSKey:POLarity POSitive /*Sets the FSK modulation polarity
to Positive for CH1.*/
:OUTPut1 ON /*Enables the output of CH1.*/
```

4.6 To Output a Frequency Sweep

Requirements

Use the SCPI commands to realize the following functions:

Output a sweep waveform from the front-panel output connector of CH1; set the sweep waveform to Sine (5 Vpp amplitude, 0 Vdc offset), the sweep type to Linear, the sweep time to 3 s, the return time to 0.1 s, the start frequency to 100 Hz, and the stop frequency to 1 kHz; enable the frequency mark function and set the mark frequency

to 500 Hz, the start hold time to 0.1 s, the stop hold to 0.1 s, the trigger source to manual, and the edge type of the trigger output signal to leading edge.

Method

```
*IDN? /*Queries the ID string of the signal generator to test
whether the remote communication works normally.*/
:SOURce1:FUNCtion SINusoid /*Sets the waveform to Sine for CH1.*/
:SOURce1:VOLTage 5 /*Sets the waveform amplitude to 5 Vpp for CH1.*/
:SOURce1:VOLTage:OFFSet 0 /*Sets the offset to 0 Vdc for CH1.*/
:SOURce1:SWEep:STATe ON /*Enables the Sweep function for CH1.*/
:SOURce1:SWEep:SPACing LINear /*Sets the sweep type to linear sweep
for CH1.*/
:SOURce1:SWEep:TIME 3 /*Sets the sweep time to 3 s for CH1.*/
:SOURce1:SWEep:RTIMe 0.1 /*Sets the return time of the Sweep
function to 0.1 s for CH1.*/
:SOURce1:FREQuency:STARt 100 /*Sets the start frequency of the
Sweep function to 100 Hz for CH1.*/
:SOURce1:FREQuency:STOP 1000 /*Sets the stop frequency of the Sweep
function to 1 kHz for CH1.*/
:OUTPut1:SYNC:MODE MARKer /*Enables the mark frequency function for
CH1.*/
:SOURce1:MARKer:FREQuency 500 /*Sets the mark frequency to 500 Hz
for CH1.*/
:SOURce1:SWEep:HTIMe:STARt 0.1 /*Sets the start hold time to 0.1 s
for CH1.*/
:SOURce1:SWEep:HTIMe:STOP 0.1 /*Sets the stop hold time to 0.1 s
for CH1.*/
:TRIGger1:SOURce BUS /*Sets the sweep trigger source to manual for
CH1.*/
:OUTPut1:TRIGger ON /*Enables the trigger output for CH1.*/
:OUTPut1:TRIGger:SLOPe POSitive /*Sets the edge of the trigger
output signal to leading edge for CH1.*/
:OUTPut1 ON /*Enables the output of CH1.*/
:TRIGger1:IMMediate /*Triggers a sweep immediately in CH1.*/
```

4.7 To Output a Burst Waveform

Requirements

Use the SCPI commands to realize the following functions:

Output a burst waveform from the CH1 output connector on the front panel; set the burst waveform to sine wave (1 kHz frequency, 5 Vpp amplitude, 0 Vdc offset, and 0° start phase), the burst type to N-Cycle burst with 10 burst count, the burst time to 0.1 s, the trigger source to internal trigger, the edge of the trigger output signal to trailing edge, and the trigger delay to 0.01 s.

Method

```
*IDN? /*Queries the ID string of the signal generator to check
whether the remote communication is normal.*/
:SOURce1:APPLy:SINusoid 1000,5,0,0 /*Sets the CH1 waveform to sine
wave with 1 kHz frequency, 5 Vpp amplitude, 0 Vdc offset, and *0°
start phase./
:SOURce1:BURSt:STATe ON /*Enables the burst function for CH1.*/
:SOURce1:BURSt:MODE TRIGgered /*Sets the burst type to N-Cycle for
CH1.*/
:TRIGger1:SOURce IMMediate /*Sets the trigger source to internal
for CH1.*/
```

```
:SOURce1:BURSt:NCYCles 10 /*Sets the burst count of N-Cycle bursts
to 10 for CH1.*/
:SOURce1:BURSt:INTernal:PERiod 0.1 /*Sets the burst period of
internally-triggered N-Cycle bursts to 0.1 s for CH1.*/
:OUTPut1:TRIGger ON /*Enables the trigger output of CH1*/
:OUTPut1:TRIGger:SLOPe NEGative /*Sets the edge of the trigger
output signal to trailing edge for CH1.*/
:TRIGger1:DELay 0.01 /*Sets the trigger delay of N-Cycle bursts to
0.01 s for CH1.*/
:OUTPut1 ON /*Enables the output of CH1.*/
```

4.8 To Use the Frequency Counter

Requirements

Use the SCPI commands to realize the following functions:

Enable the frequency counter function; set the instrument to select the proper gate time automatically according to the characteristic of the signal to be measured; turn on the statistical function; set the run mode to continuous; set the coupling mode to AC coupling; turn on the high-frequency suppression function; set the operating status to Run.

Method

```
*IDN? /*Queries the ID string of the signal generator to check
whether the remote communication is normal.*/
:COUNter:STATe ON /*Enables the frequency counter function.*/
:COUNter:IMPedance OMGE /*Sets the input impedance to HighZ.*/
:COUNter:GATetime:AUTO:STATe ON /*Sets the instrument to select the
proper gate time automatically according to the characteristic of
the signal to be measured.*/
:COUNter:AVERage:STATe ON /*Enables the statistical function of the
frequency counter.*/
:COUNter:COUPling AC /*Sets the coupling mode of the input signal
to AC coupling.*/
:COUNter:HF ON /*Enables the HF suppression function of the
frequency counter.*/
:COUNter:RUN:STATe RUN /*Sets the operating status of the frequency
counter to Run.*/
```

5 Programming Examples

This chapter illustrates how to control the instrument by programming in LabVIEW, Visual Basic, and Visual C++. These examples are programmed based on Virtual Instrument Software Architecture (VISA) library.

RIGOL official website (*http://www.rigol.com*) provides the programming examples based on LabVIEWk, Visual Basic, Visual C++, Matlab, and Python. You can go to the official website and click **SUPPORT** > **Programming Demo** to obtain the programming examples.

5.1 Programming Preparations

Before programming, you need to make the following preparations:

Install the NI-VISA library. In this manual, the default installation path is C:\Program Files\IVI Foundation\VISA.

The instrument communicates with the PC via the USB interface. Connect the rear-panel USB Device interface to the PC by using the USB cable. After the instrument is properly connected to the PC, power on the instrument to start it.

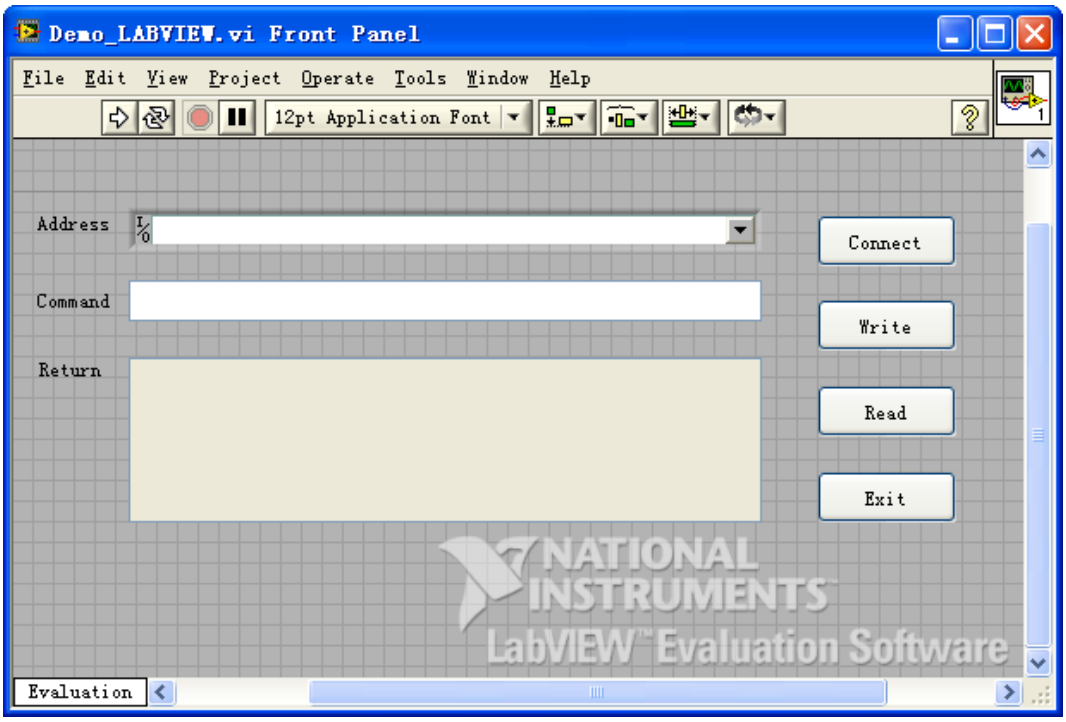
The following parts will make a detailed introduction about the programming examples in LabVIEW, Visual Basic, and Visual C++.

5.2 LabVIEW Programming Example

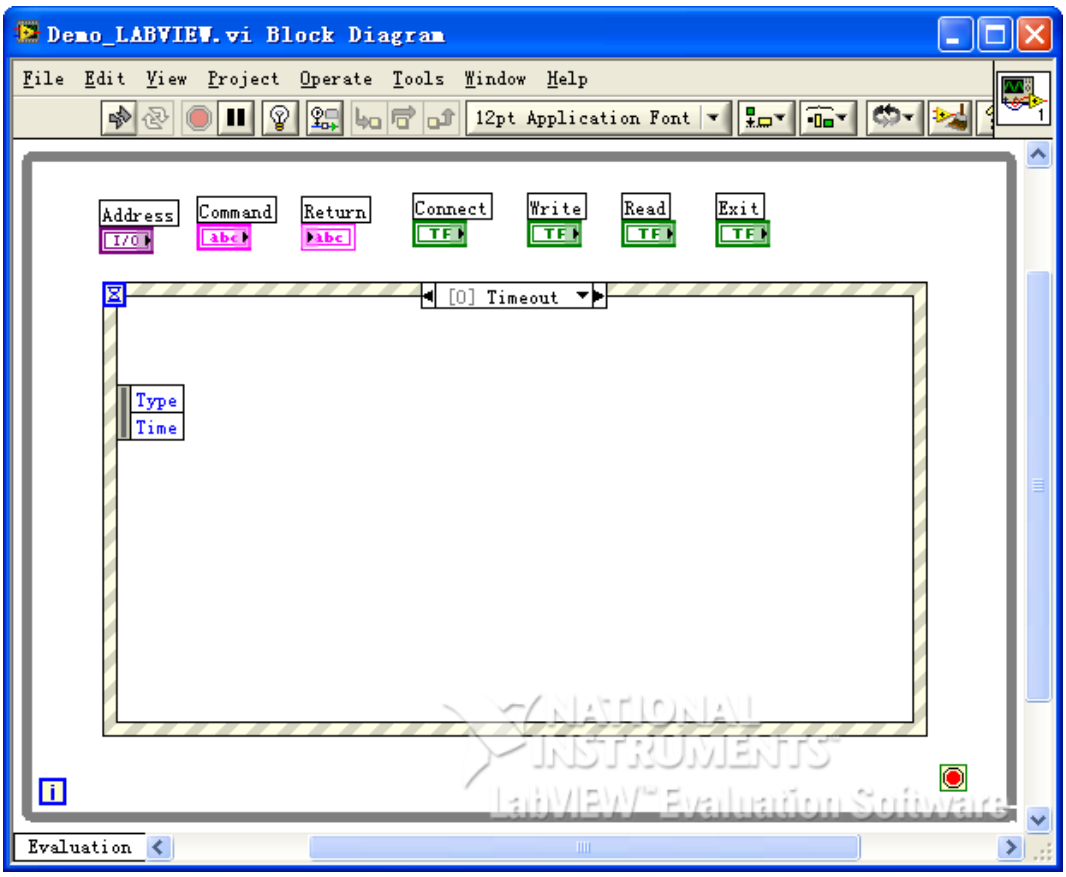
Program used in this example: LabVIEW 2009

Function realized in this example: search for the instrument address, connect the instrument, send command, and read the returned value.

1. Run LabVIEW, and then create a VI file named Demo_LABVIEW.
2. Add controls in the front panel interface, including the **Address**, **Command**, and **Return** field as well as the **Connect**, **Write**, **Read**, and **Exit** buttons.

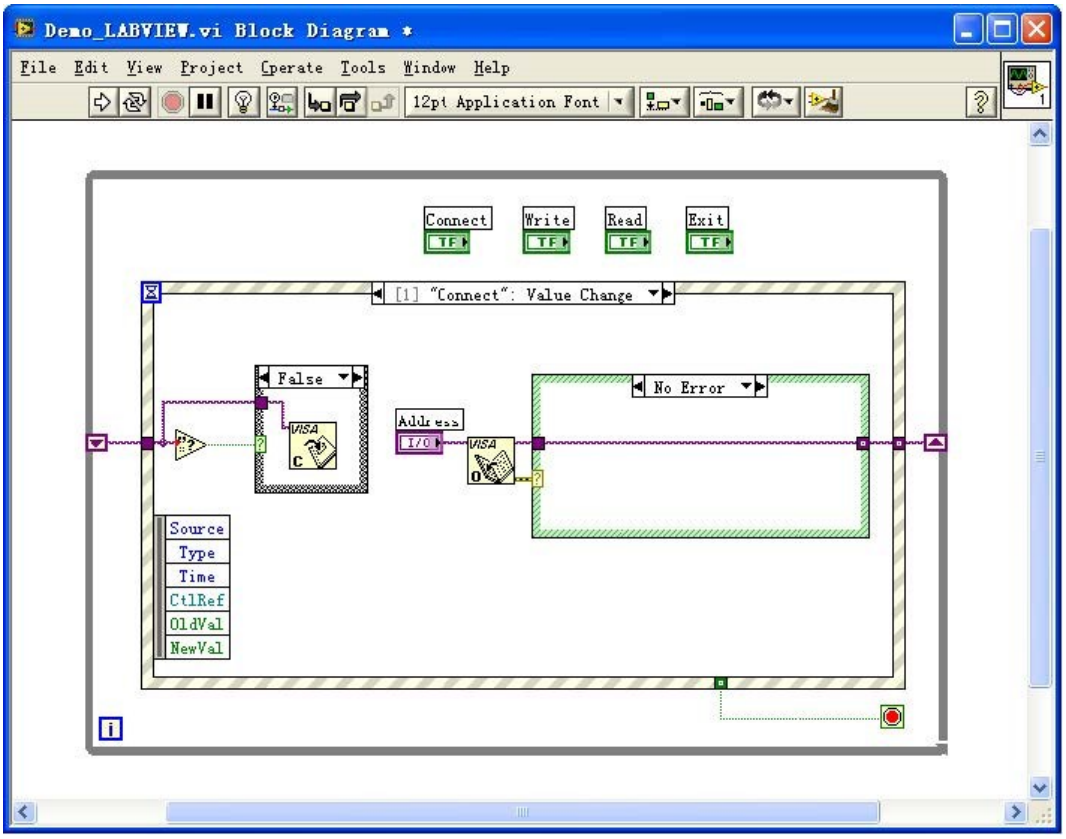


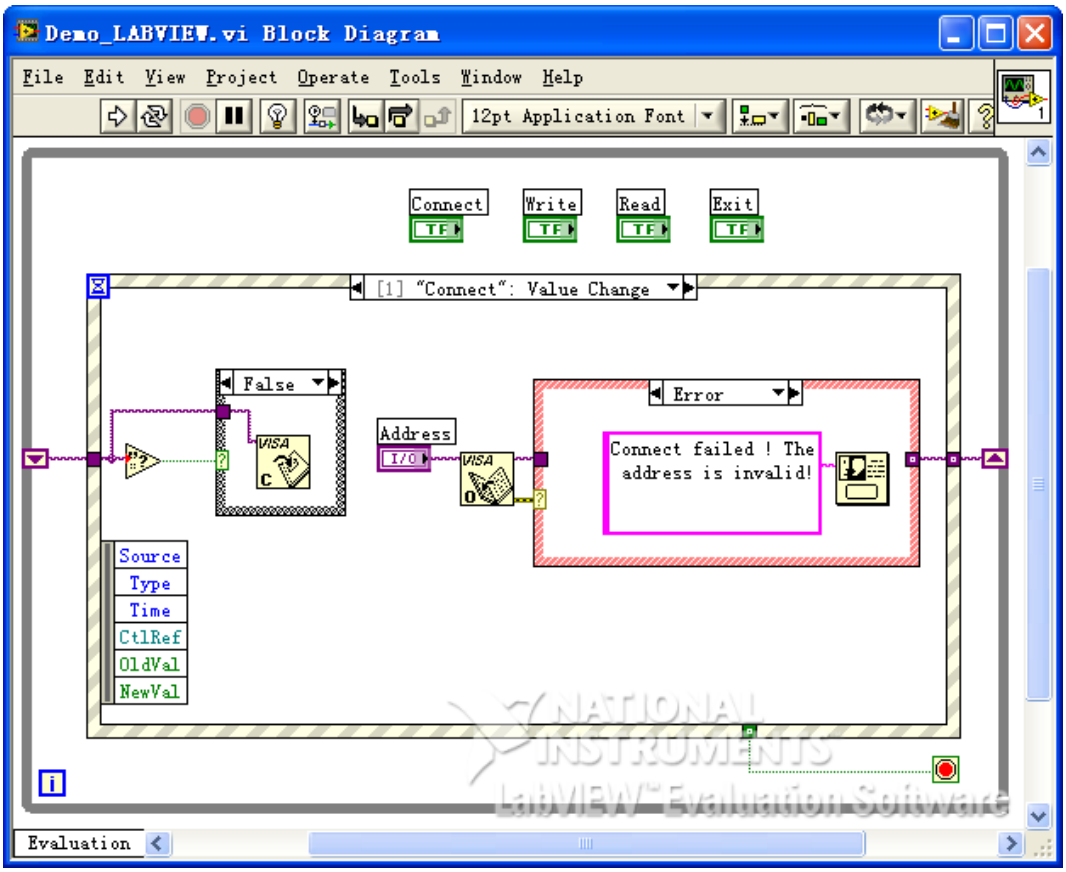
3. Click **Show Block Diagram** in the **Window** menu to create event structure.

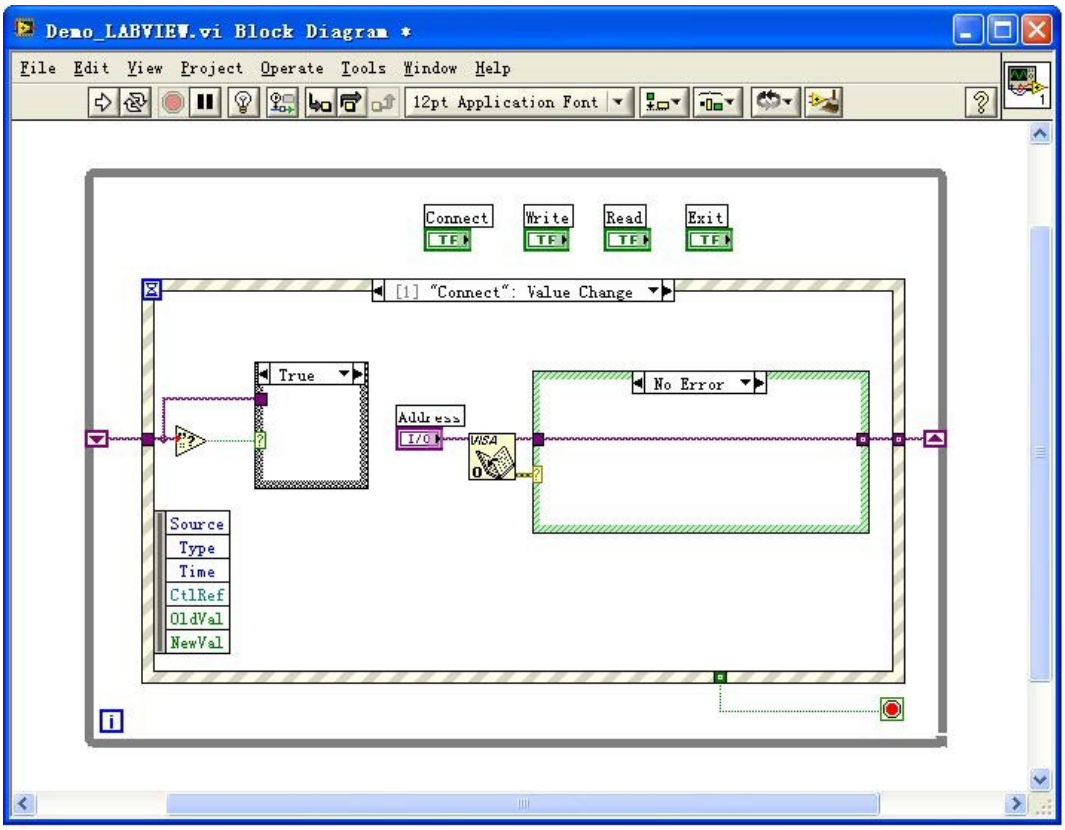


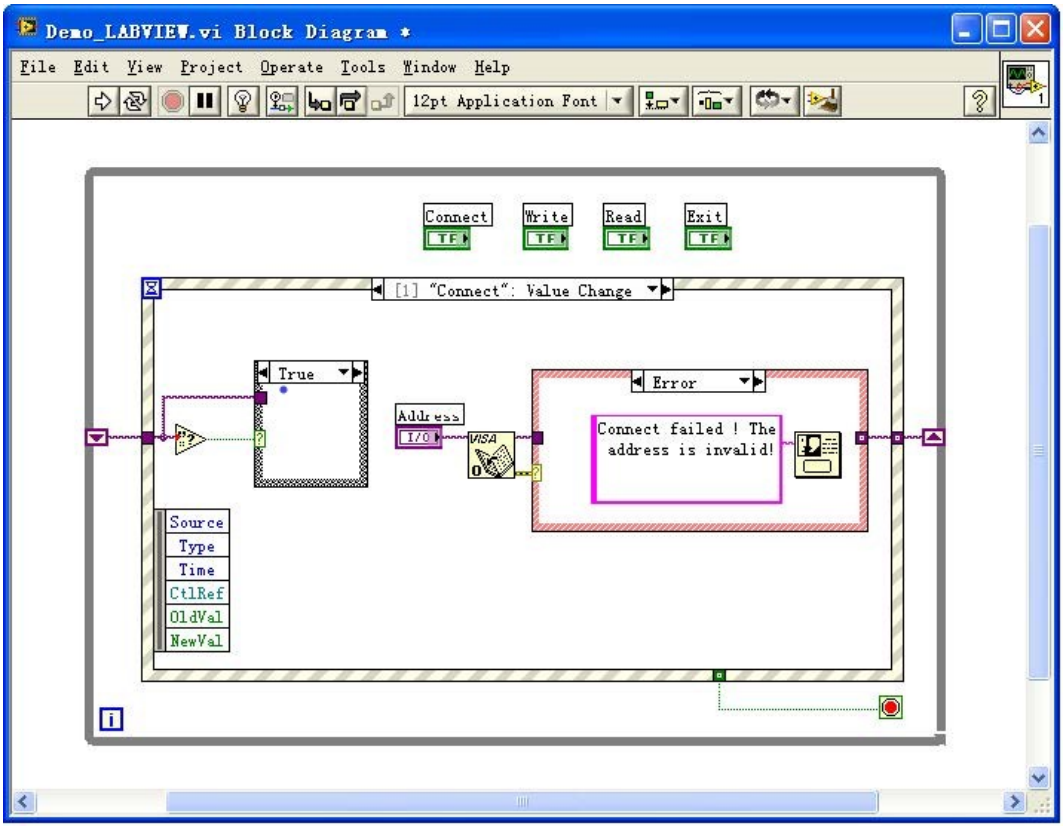
4. Add events, including connecting instrument, write operation, read operation, and exit.

a. Connect the instrument (including error processing):

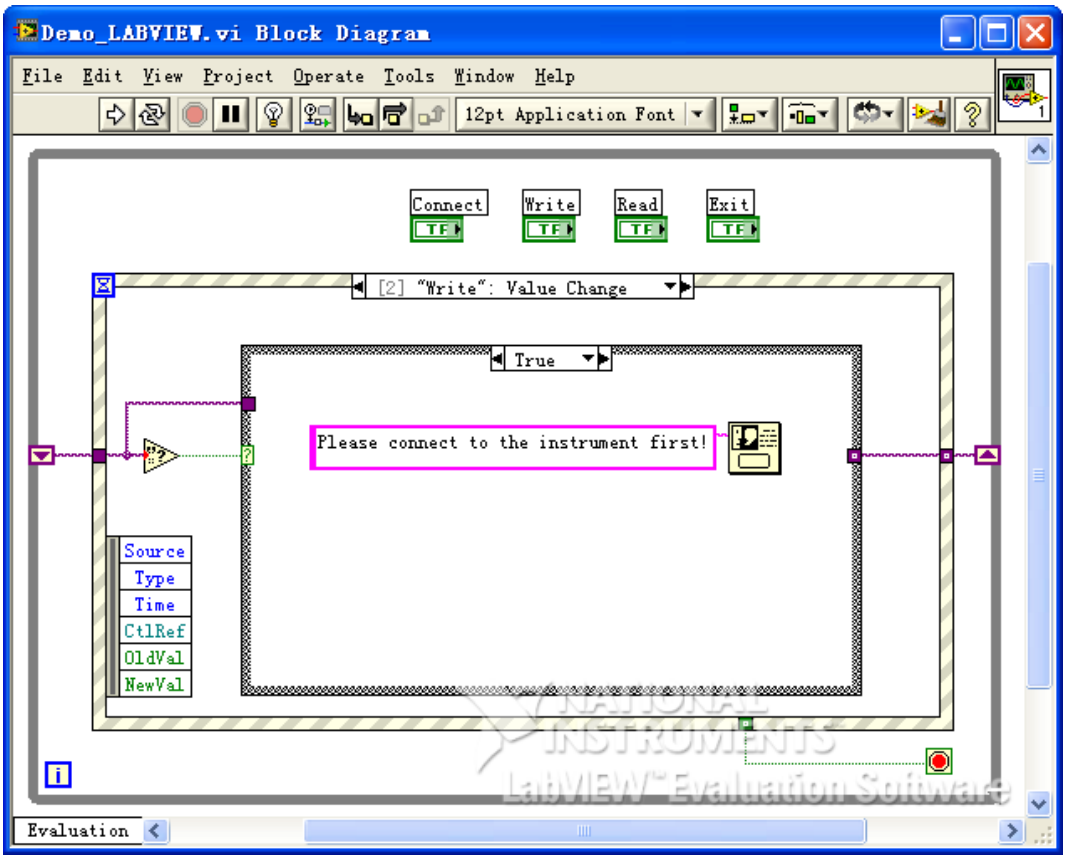


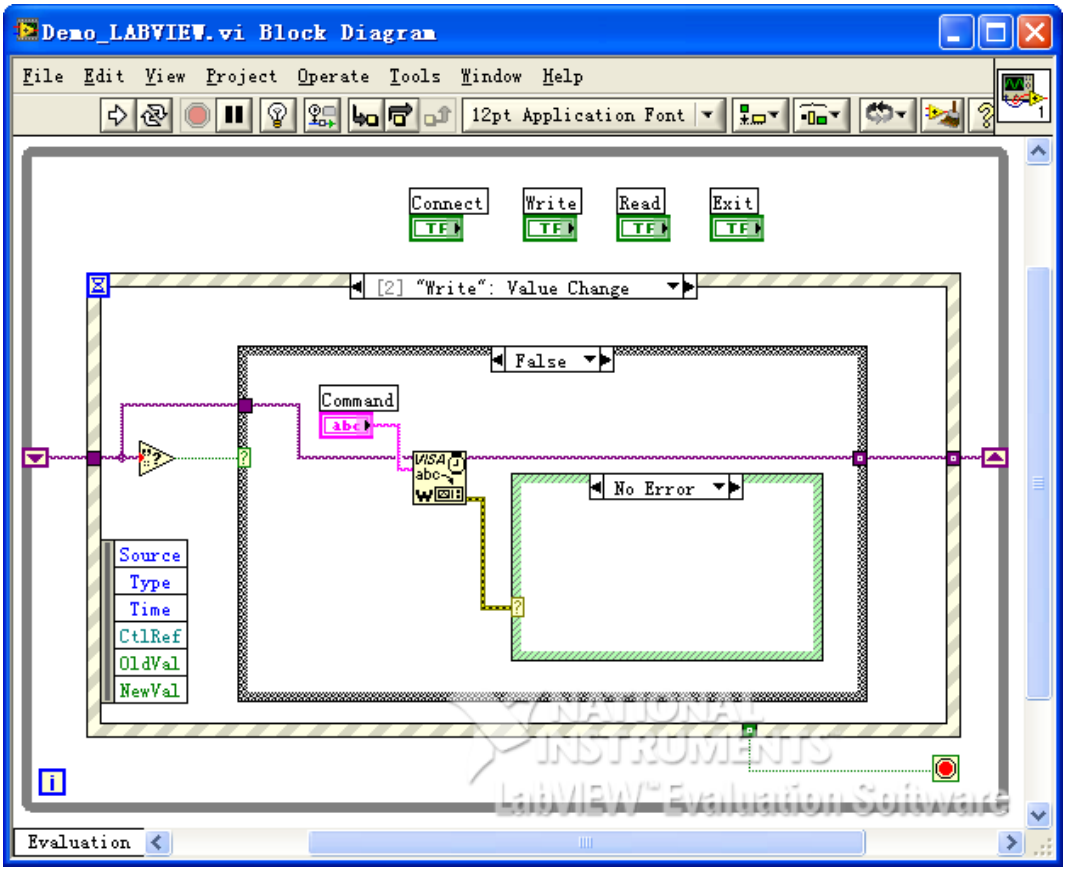




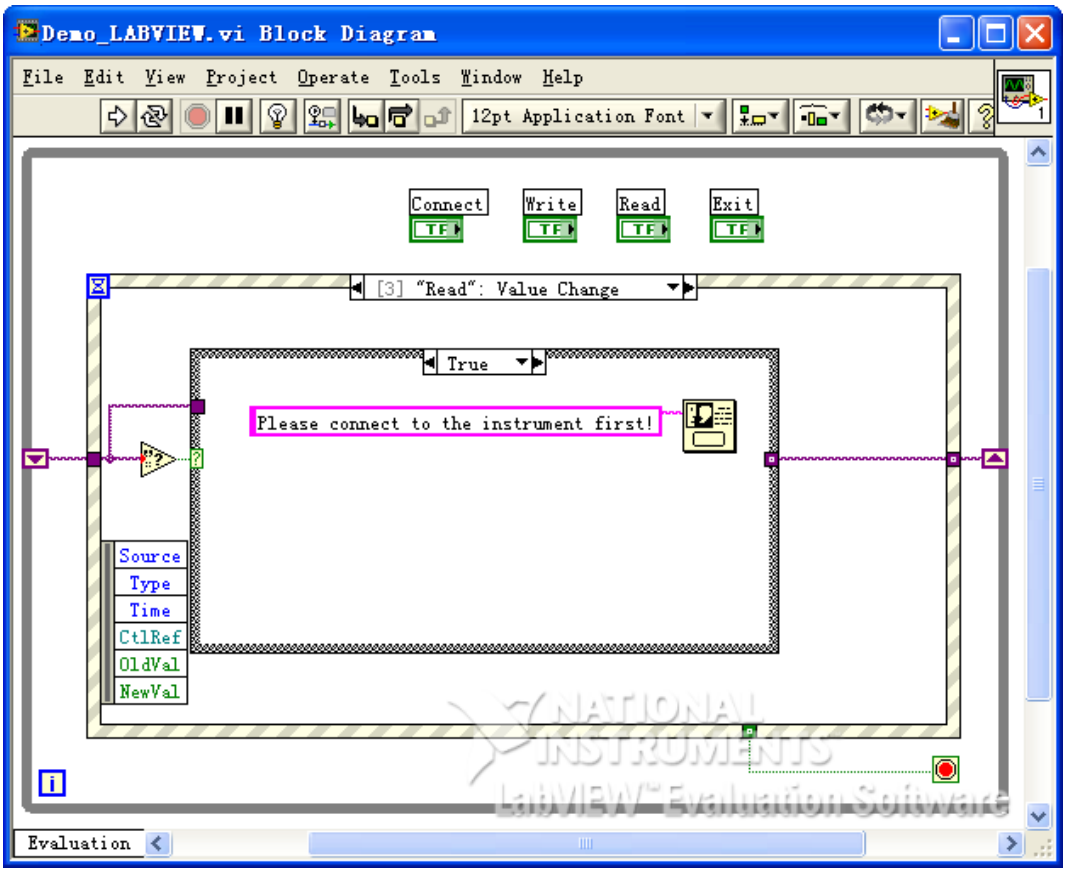


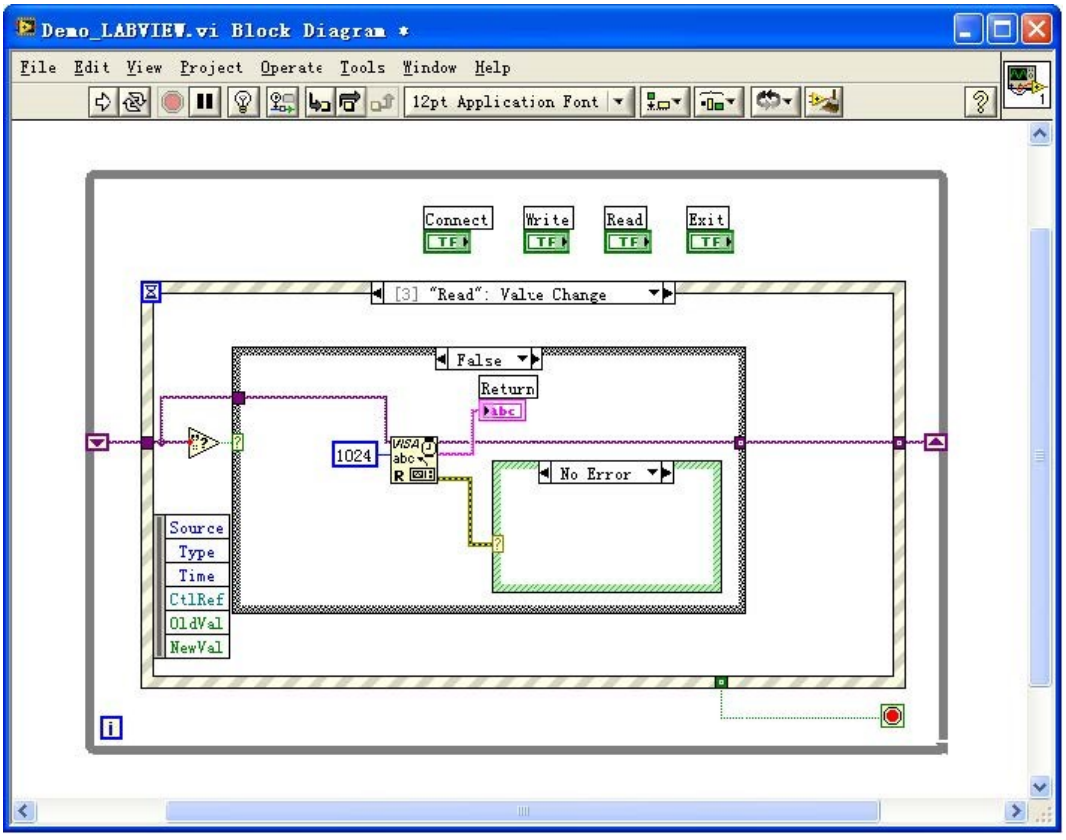
b. Write operation (including error judgment):



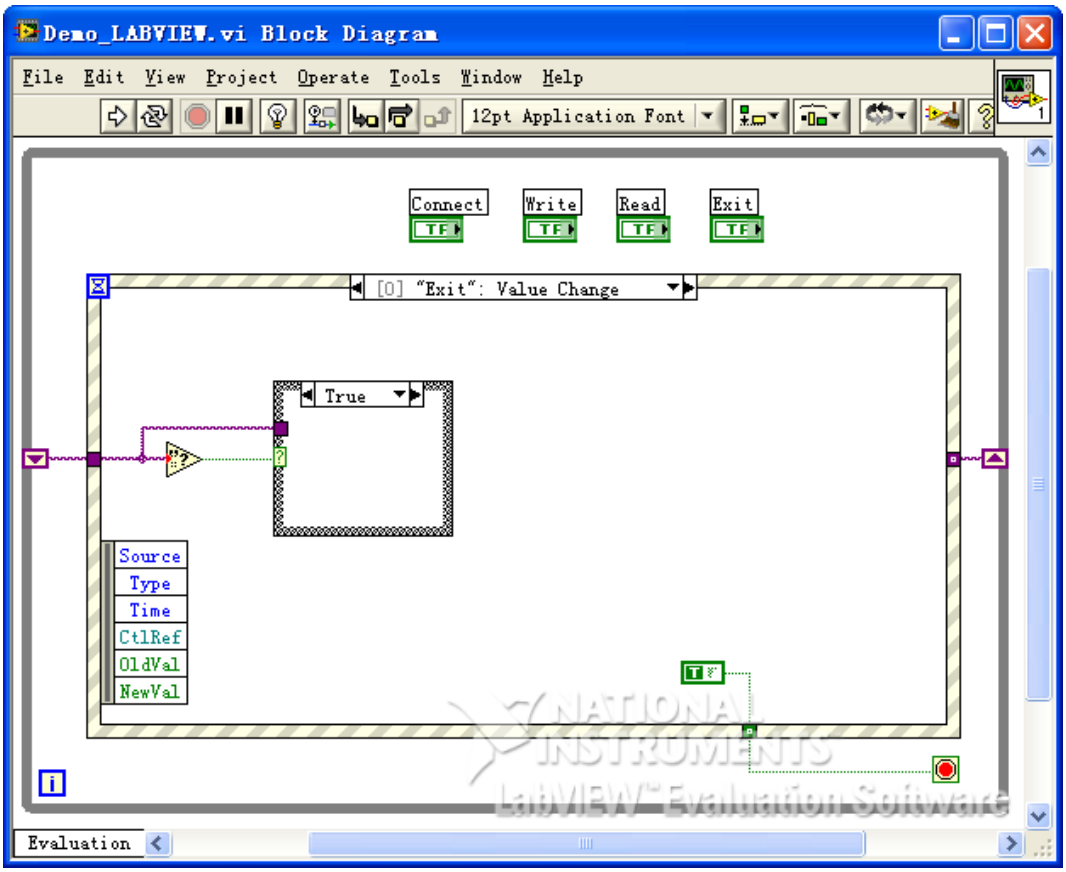


c. Read operation (including error processing):

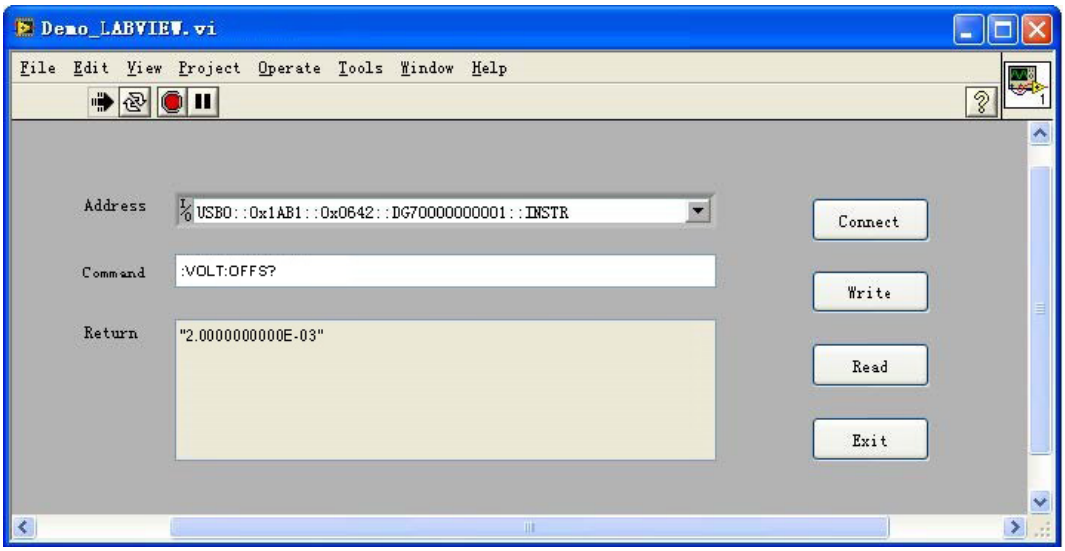




d. Exit:



5. Run the program and the interface as shown in the figure below is displayed. Click the **Address** drop-down button and select the VISA resource name; click **Connect** to connect the instrument; enter the command into the **Command** input field and click **Write** to write the command into the instrument. If the command is a query command, click **Read** and the returned value is displayed in the **Return** field.



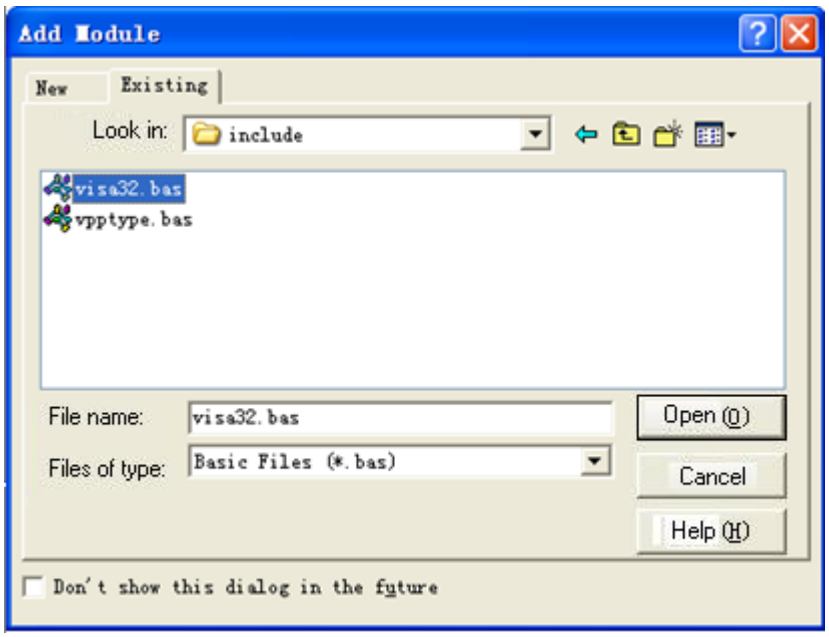
5.3 Visual Basic Programming Example

Program used in this example: Visual Basic 6.0

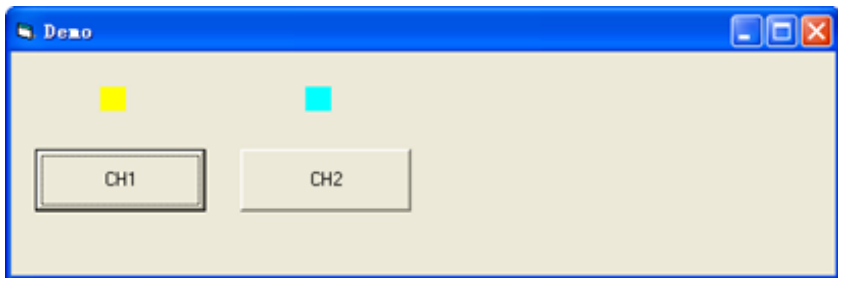
Function realized in this example: control the on/off state of either channel.

Enter the Visual Basic 6.0 programming environment, and perform the following procedures.

1. Build a standard application program project (Standard EXE), and name it "Demo".
2. Click **Project** > **Add Module** to open the Add Module dialog box. In the dialog box, click the Existing tab to search for the visa32.bas file in the include folder under the **NI-VISA** installation path and add the file.



3. In the Demo dialog box, add two buttons to represent CH1～CH2. Add two Labels (Label1(0) and Label1(1)) to represent the status of CH1～CH2 respectively (when the channel is enabled, it displays the color of the channel; when the channel is disabled, it displays gray). See the figure below.



4. Click **Project** > **Project1 Properties** to open the Project1 – Project Properties dialog box. In the dialog box, click on the General tab and select Form1 from the drop-down list under **Startup Object**.

5. Double-click CH1 to enter the programming environment. Add the following codes to control CH1-CH2. The codes of CH1 are as shown below; the codes of the other channel are similar.

```
Dim defrm As Long
Dim vi As Long
Dim strRes As String * 200
Dim list As Long
Dim nmatches As Long
Dim matches As String * 200 'Reserve the obtained device number
Dim s32Disp As Integer
' Obtain the usb resource of visa
Call viOpenDefaultRM(defrm)
Call viFindRsrc(defrm, "USB?*", list, nmatches, matches)
' Turn on the instrument
Call viOpen(defrm, matches, 0, 0, vi)
' Send a command to query the status of CH1
Call viVPrintf(vi, ":OUTP1:STAT?" + Chr$(10), 0)
' Obtain the status of CH1
Call viVScanf(vi, "%t", strRes)
s32Disp = CInt(strRes)
If (s32Disp = 1) Then
' Send the setting command
Call viVPrintf(vi, ":OUTP1:STAT 0" + Chr$(10), 0)
Label1(0).ForeColor = &H808080 'Gray
Else
Call viVPrintf(vi, ":OUTP1:STAT 1" + Chr$(10), 0)
Label1(0).ForeColor = &HFFFF& 'Yellow
End If
' Close the resource
Call viClose(vi)
Call viClose(defrm)
```

6. Save and run the project to obtain a single exe program for demo. When the instrument is correctly connected to the PC, you can control the on/off status of either channel.

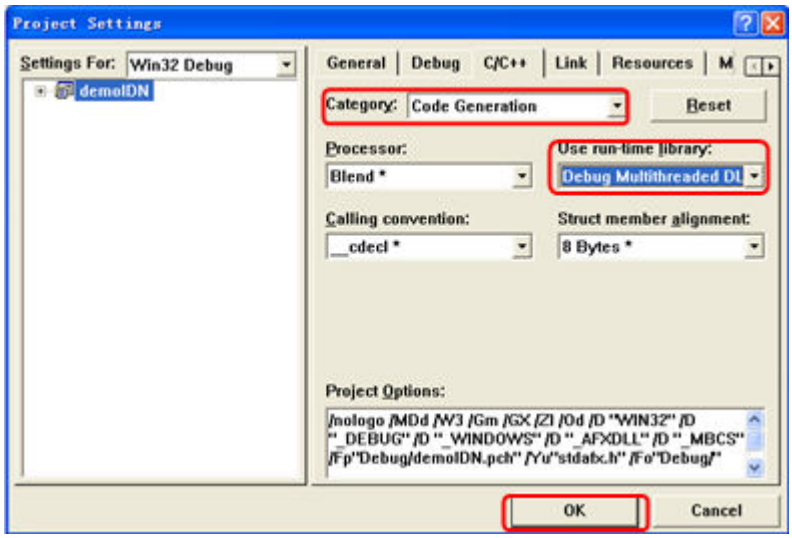
5.4 Visual C++ Programming Example

Program used in this example: Visual C++6.0

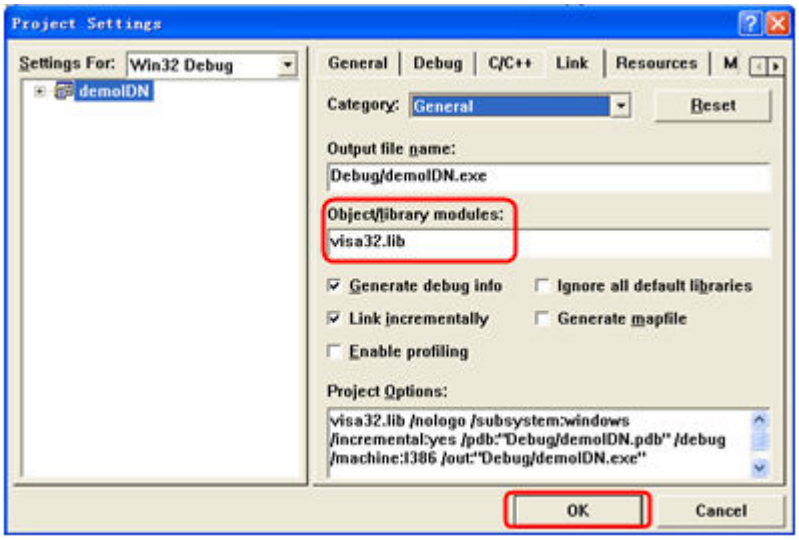
Function realized in this example:search for the instrument address, connect to the instrument, send commands, and read return values.

Enter the Visual C++6.0 programming environment, and perform the following procedures.

1. Create a MFC project based on a dialog box.
2. Click **Project** > **Settings** to open the **Project Setting** dialog box. In the dialog box, click the **C/C++** tab, select **Code Generation** from the drop-down list under **Category**. Choose **Debug Multithreaded DLL** from the drop-down list under **Use run-time library**. Click **OK** to close the dialog box.



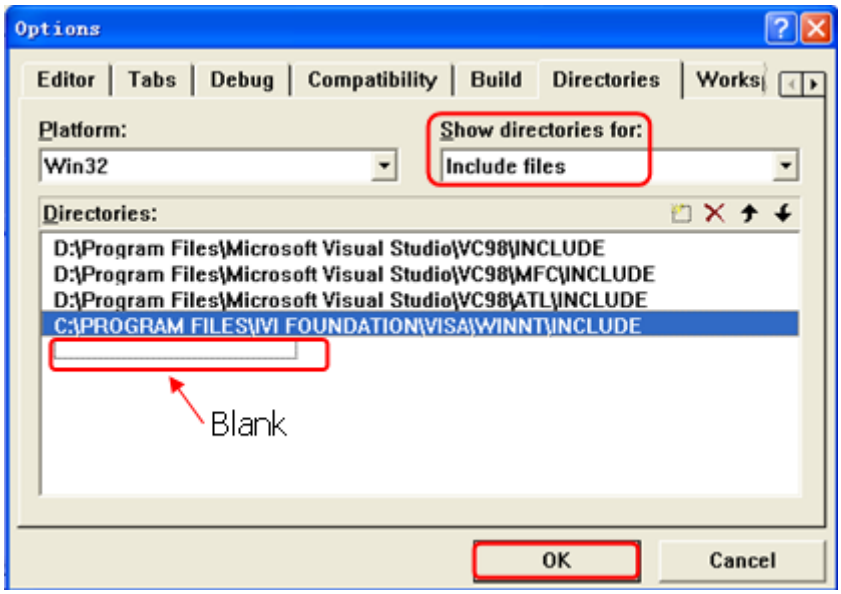
3. Click **Project** > **Settings** to open the **Project Setting** dialog box. In the dialog box, click the **Link** tab, add "visa32.lib" under **Object/library modules**, then click **OK** to close the dialog box.



4. Click **Tools** > **Options** to open the Options dialog box. Then click the **Directories** tab.

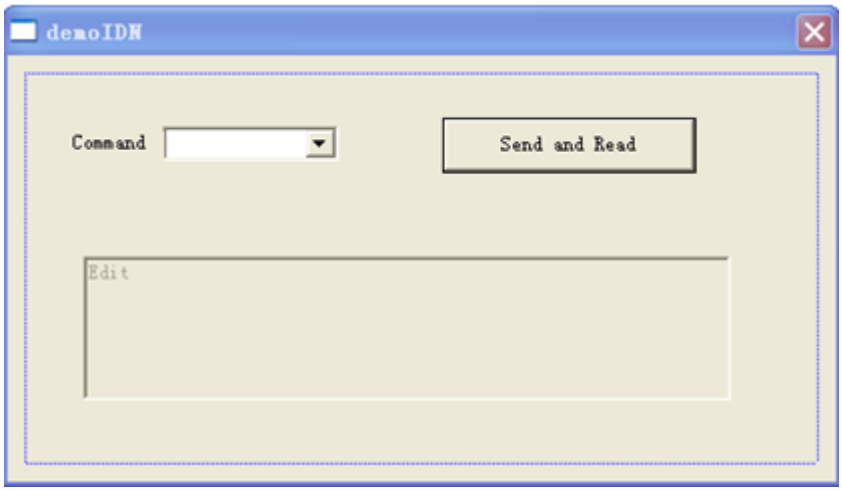
   Select **Include files** from the drop-down list under **Show directories for**. Double click the empty space under **Directories** to enter the specified path of Include files: C:\Program Files\IVI Foundation\VISA\WinNT\include. Click **OK** to close the dialog box.

   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: By now, VISA library has been added.

5. Add the **Text**, **Combo Box**, **Button**, and **Edit Box** controls. The layout interface for adding controls is as follows:



6. Modify the control attributes.

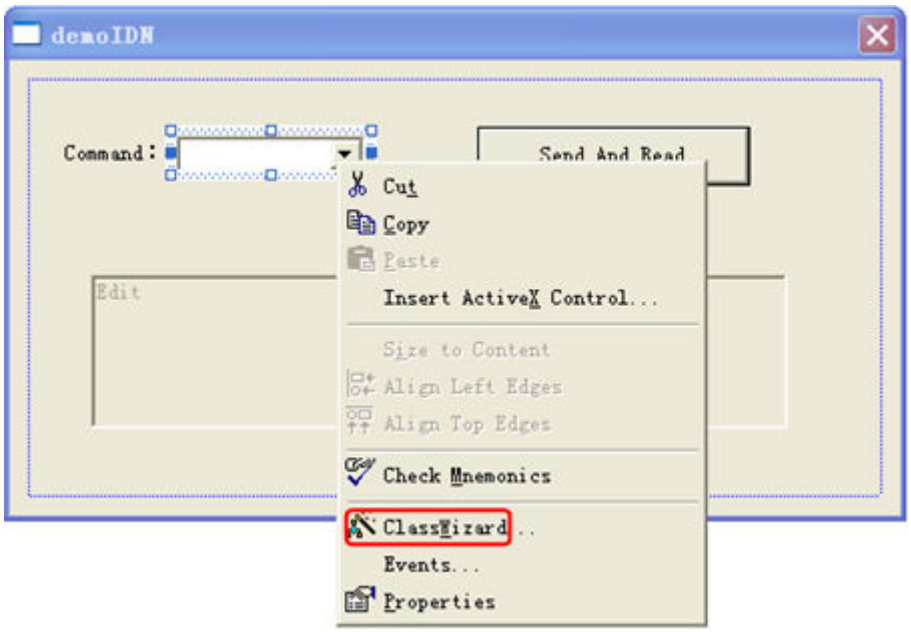
   a. Name **Text** as "Command".

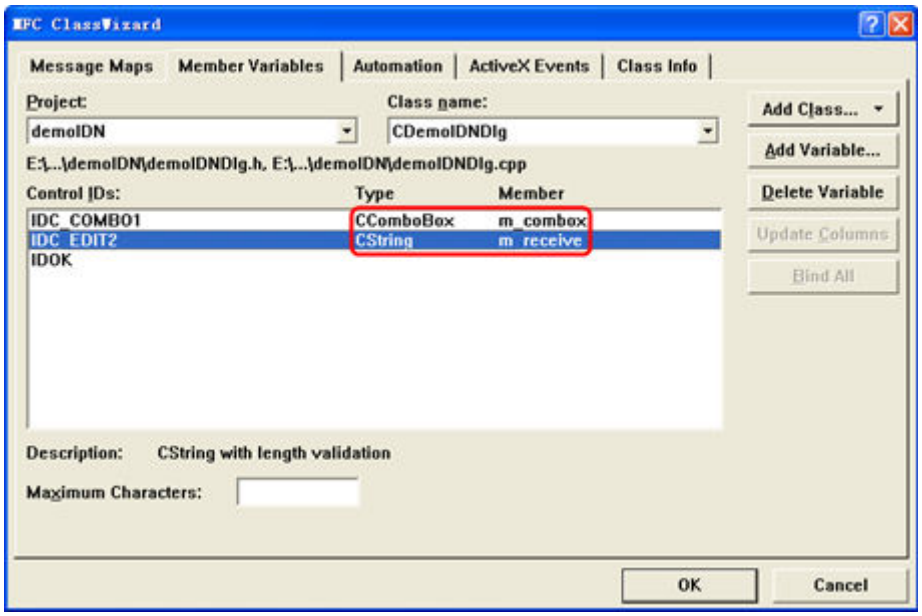
   b. Open the **Data** item in the **Combo Box** attribute and input the following command *IDN? manually.

   c. Open the **General** item in the **Edit Box** attribute and select **Disabled**.

   d. Name **Button** as **Send and Read**.

7. Add the variables m_combox and m_receive to the **Combo Box** and **Edit Box** controls respectively.





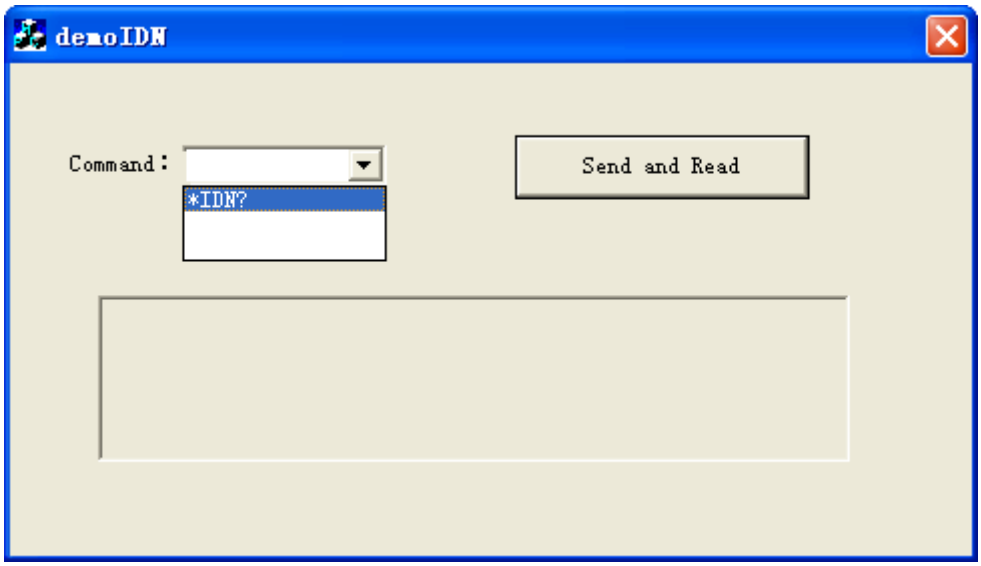
8. Add codes.

Double-click **Send and Read** to enter the programming environment. Declare the #include <visa.h> of the VISA library in the header file and then add the following codes:

```
ViSession defaultRM, vi;
char buf [256] = {0};
CString s,strTemp;
char* stringTemp;

ViChar buffer [VI_FIND_BUFLEN];
```

```
ViRsrc matches=buffer;
ViUInt32 nmatches;
ViFindList list;

viOpenDefaultRM (&defaultRM);
//Acquire the USB resource of VISA
viFindRsrc(defaultRM, "USB?*", &list,&nmatches, matches);
viOpen (defaultRM,matches,VI_NULL,VI_NULL,&vi);

//Send the command received
m_combox.GetLBText(m_combox.GetCurSel(),strTemp);
strTemp = strTemp + "\n";
stringTemp = (char *)(LPCTSTR)strTemp;
viPrintf (vi,stringTemp);

//Read the results
viScanf (vi, "%t\n", &buf);

//Display the results
UpdateData (TRUE);
m_receive = buf;
UpdateData (FALSE);
viClose (vi);
viClose (defaultRM);
```

9. Save, compile, and run the project to obtain a single exe file. When the instrument is correctly connected to the PC, enter a command (for example, *IDN?) and click **Send and Read** to execute the command. Then, the reading results will be returned.



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 JAPAN CO., LTD.
5F,3-45-6,Minamiotsuka,
Toshima-Ku,
Tokyo,170-0005,Japan
Tel: +81-3-6262-8932
Fax: +81-3-6262-8933
Email: info-japan@rigol.com