

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

User's Guide

Publication Number 33250-90002 (*order as 33250-90100 manual set*)
Edition 2, March 2003

© Copyright Agilent Technologies, Inc. 2000, 2003

For Safety information, Warranties, and Regulatory information, see the pages following the Index.

Agilent 33250A
80 MHz Function / Arbitrary Waveform Generator

Agilent 33250A at a Glance

The Agilent Technologies 33250A is a high-performance 80 MHz synthesized function generator with built-in arbitrary waveform and pulse capabilities. Its combination of bench-top and system features makes this function generator a versatile solution for your testing requirements now and in the future.

Convenient bench-top features

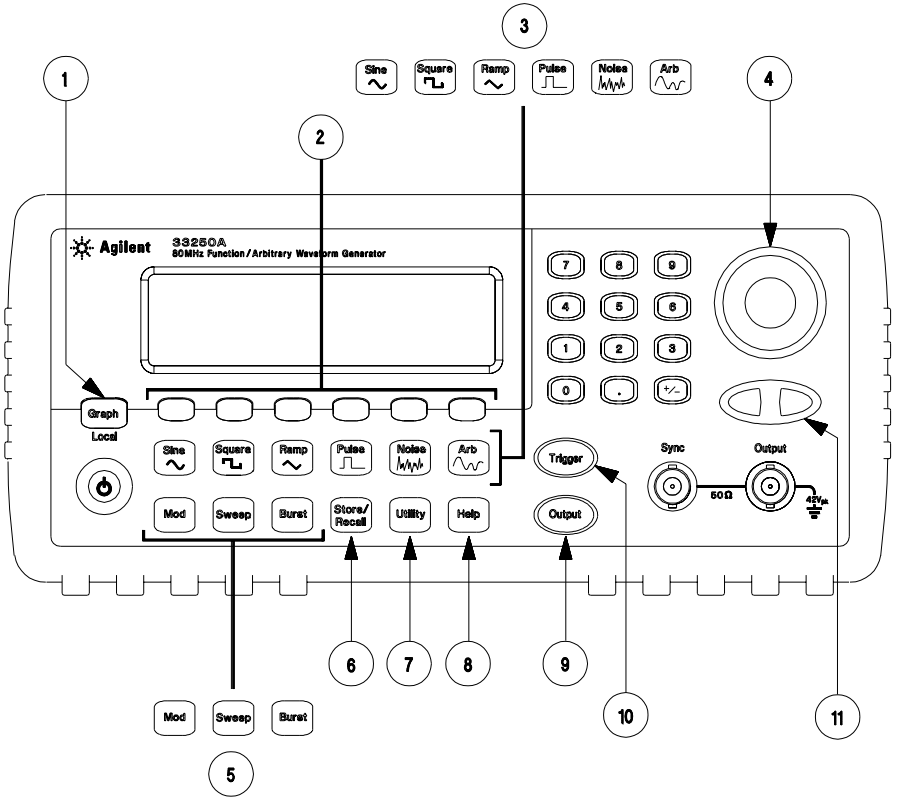
- 10 standard waveforms
- Built-in 12-bit 200 MSa/s arbitrary waveform capability
- Precise pulse waveform capabilities with adjustable edge time
- LCD color display provides numeric and graphical views
- Easy-to-use knob and numeric keypad
- Instrument state storage with user-defined names
- Portable, ruggedized case with non-skid feet

Flexible system features

- Four downloadable 64K-point arbitrary waveform memories
- GPIB (IEEE-488) interface and RS-232 interface are standard
- SCPI (Standard Commands for Programmable Instruments) compatibility

Note: *Unless otherwise indicated, this manual applies to all Serial Numbers.*

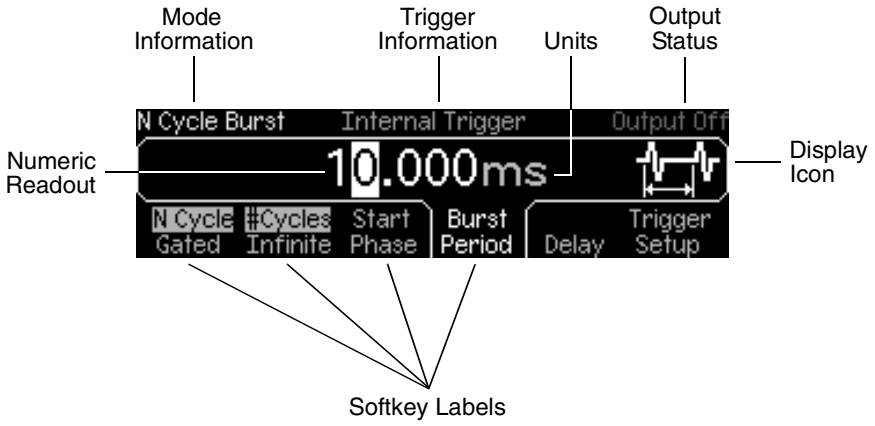
The Front Panel at a Glance



1 Graph Mode/Local Key
2 Menu Operation Softkeys
3 Waveform Selection Keys
4 Knob
5 Modulation/Sweep/Burst Menus
6 State Storage Menu
7 Utility Menu
8 Instrument Help Topic Menu
9 Output Enable/Disable Key
10 Manual Trigger Key (*used for Sweep and Burst only*)
11 Cursor Keys

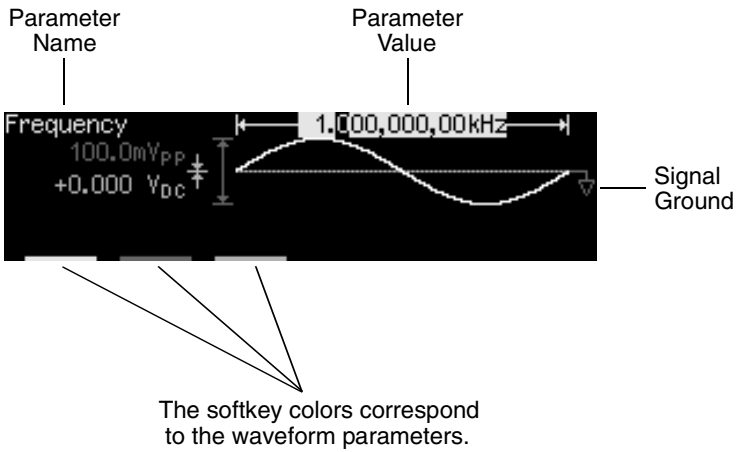
Note: *To get context-sensitive help on any front-panel key or menu softkey, press and hold down that key.*

Menu Mode



Graph Mode

To enter the Graph Mode, press the Graph *key.*



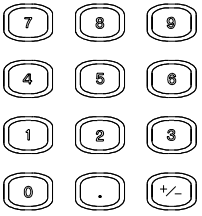
Front-Panel Number Entry

You can enter numbers from the front-panel using one of two methods.

Use the knob and arrow keys to modify the displayed number.

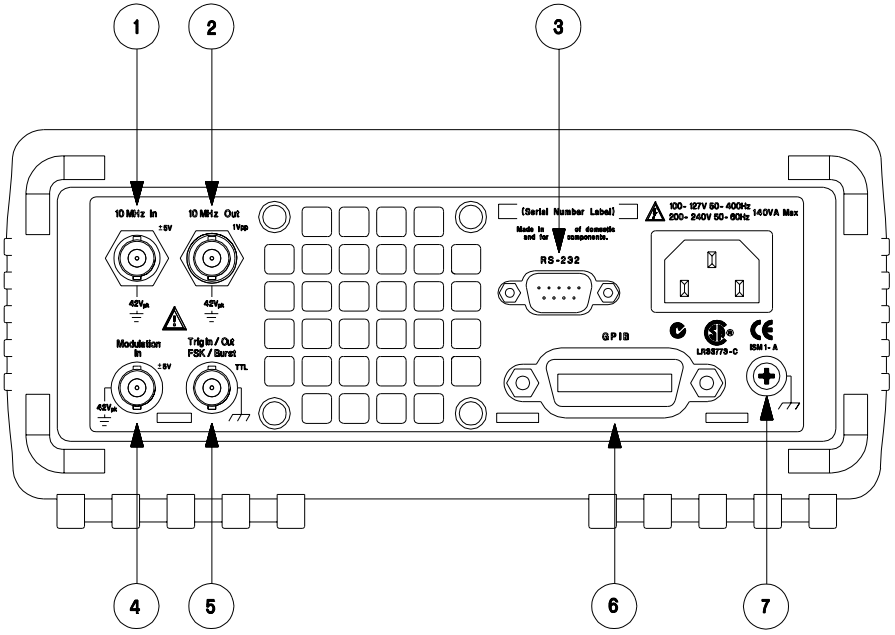


Use the numeric keypad and menu softkeys to select the units.





The Rear Panel at a Glance



1 External 10 MHz Reference Input Terminal
2 Internal 10 MHz Reference Output Terminal
3 RS-232 Interface Connector
4 External Modulation Input Terminal
5 Input: External Trig/FSK/Burst Gate
Output: Trigger Output
6 GPIB Interface Connector
7 Chassis Ground

Use the Utility menu to:

- Select the GPIB or RS-232 interface (see chapter 2).
- Select the GPIB address (see chapter 2).
- Set the RS-232 baud rate, parity, and handshake mode (see chapter 2).

WARNING

For protection from electrical shock, the power cord ground must not be defeated. If only a two-contact electrical outlet is available, connect the instrument's chassis ground screw (see above) to a good earth ground.

In This Book

Quick Start Chapter 1 prepares the function generator for use and helps you get familiar with a few of its front-panel features.

Front-Panel Menu Operation Chapter 2 introduces you to the front-panel menu and describes some of the function generator's menu features.

Features and Functions Chapter 3 gives a detailed description of the function generator's capabilities and operation. You will find this chapter useful whether you are operating the function generator from the front panel or over the remote interface.

Remote Interface Reference Chapter 4 contains reference information to help you program the function generator over the remote interface.

Error Messages Chapter 5 lists the error messages that may appear as you are working with the function generator. Each listing contains enough information to help you diagnose and solve the problem.

Application Programs Chapter 6 contains several remote interface application programs to help you develop programs for your application.

Tutorial Chapter 7 discusses the fundamentals of signal generation and modulation techniques.

Specifications Chapter 8 lists the function generator's specifications.



If you have questions relating to the operation of the Agilent 33250A, call **1-800-452-4844** in the United States, or contact your nearest Agilent Technologies Office.

If your 33250A fails within three years of purchase, Agilent will either repair or replace it free of charge. Call **1-877-447-7278** in the United States (and ask for "Agilent Express") or contact your local Agilent Technologies Office.

Contents

Chapter 1 Quick Start

To Prepare the Function Generator for Use 15
To Adjust the Carrying Handle 16
To Set the Output Frequency 17
To Set the Output Amplitude 18
To Set a DC Offset Voltage 20
To Set the Duty Cycle 21
To Configure a Pulse Waveform 22
To View a Waveform Graph 23
To Output a Stored Arbitrary Waveform 24
To Use the Built-In Help System 25
To Rack Mount the Function Generator 27

Chapter 2 Front-Panel Menu Operation

Front-Panel Menu Reference 31
To Select the Output Termination 33
To Reset the Function Generator 33
To Output a Modulated Waveform 34
To Output an FSK Waveform 36
To Output a Frequency Sweep 38
To Output a Burst Waveform 40
To Trigger a Sweep or Burst 42
To Store the Instrument State 43
To Configure the Remote Interface 44

Chapter 3 Features and Functions

Output Configuration 49
Pulse Waveforms 64
Amplitude Modulation (AM) 67
Frequency Modulation (FM) 72
Frequency-Shift Keying (FSK) Modulation 78
Frequency Sweep 82
Burst Mode 89
Triggering 98
Arbitrary Waveforms 103
System-Related Operations 109
Remote Interface Configuration 118
Calibration Overview 123
Factory Default Settings 127

Chapter 4 Remote Interface Reference

SCPI Command Summary 131
Simplified Programming Overview 142
Using the APPLy Command 144
Output Configuration Commands 153
Pulse Configuration Commands 166
Amplitude Modulation (AM) Commands 169
Frequency Modulation (FM) Commands 172
Frequency-Shift Keying (FSK) Commands 176
Frequency Sweep Commands 179
Burst Mode Commands 187
Triggering Commands 195
Arbitrary Waveform Commands 198
State Storage Commands 209
System-Related Commands 213
Interface Configuration Commands 218
RS-232 Interface Configuration 219
Phase-Lock Commands 223
The SCPI Status System 225
Status Reporting Commands 235
Calibration Commands 239
An Introduction to the SCPI Language 241
Using Device Clear 246

Chapter 5 Error Messages

Command Errors 249
Execution Errors 252
Query Errors 267
Instrument Errors 268
Self-Test Errors 269
Calibration Errors 271
Arbitrary Waveform Errors 272

Chapter 6 Application Programs

Introduction 276
Example: BASIC for Windows 278
Example: Microsoft Visual Basic for Windows 282
Example: Microsoft Visual C++ for Windows 287

Chapter 7 Tutorial

Direct Digital Synthesis 295
Creating Arbitrary Waveforms 298
Square Waveform Generation 300
Pulse Waveform Generation 300
Signal Imperfections 302
Output Amplitude Control 304
Ground Loops 305
Attributes of AC Signals 307
Modulation 309
Frequency Sweep 312
Burst 315

Chapter 8 Specifications

Frequency Characteristics 320
Sinewave Spectral Purity 320
Signal Characteristics 320
Output Characteristics 321
Modulation Characteristics 321
Burst 321
Sweep 321
System Characteristics 322
Trigger Characteristics 322
Clock Reference 322
Sync Output 322
General Specifications 323
Product Dimensions 324

Index 325

Quick Start

1

Quick Start

One of the first things you will want to do with your function generator is to become acquainted with the front panel. We have written the exercises in this chapter to prepare the instrument for use and help you get familiar with some of its front-panel operations. This chapter is divided into the following sections:

- To Prepare the Function Generator for Use, *on page 15*
- To Adjust the Carrying Handle, *on page 16*
- To Set the Output Frequency, *on page 17*
- To Set the Output Amplitude, *on page 18*
- To Set a DC Offset Voltage, *on page 20*
- To Set the Duty Cycle, *on page 21*
- To Configure a Pulse Waveform, *on page 22*
- To View a Waveform Graph, *on page 23*
- To Output a Stored Arbitrary Waveform, *on page 24*
- To Use the Built-In Help System, *on page 25*
- To Rack Mount the Function Generator, *on page 27*

To Prepare the Function Generator for Use

1 Check the list of supplied items.

Verify that you have received the following items with your instrument. If anything is missing, please contact your nearest Agilent Sales Office.

- ❑ One power cord.
- ❑ This *User's Guide*.
- ❑ One *Service Guide*.
- ❑ One folded *Quick Reference Guide*.
- ❑ Certificate of Calibration.
- ❑ Connectivity software on CD-ROM.
- ❑ One RS-232 cable.


Power Switch

2 Connect the power cord and turn on the function generator.

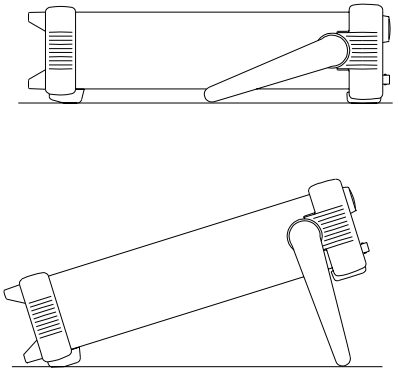
Several power-on information messages are displayed after the function generator performs its power-on self-test. The GPIB address is displayed. The function generator powers up in the *sine wave* function at 1 kHz with an amplitude of 100 mV peak-to-peak (into a 50Ω termination). At power-on, the *Output* connector is disabled. To enable the *Output* connector, press the Output key.

If the function generator *does not* turn on, verify that the power cord is firmly connected to the power receptacle on the rear panel (the power-line voltage is automatically sensed at power-on). You should also make sure that the function generator is connected to a power source that is energized. Then, verify that the function generator is turned on.

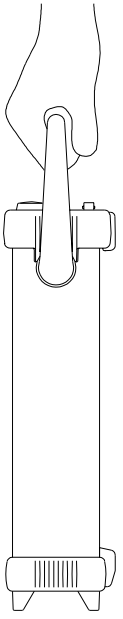
If you need further assistance, refer to the Agilent 33250A *Service Guide* for instructions on returning the function generator to Agilent for service.

To Adjust the Carrying Handle

To adjust the position, grasp the handle by the sides and *pull outward*. Then, rotate the handle to the desired position.



Bench-top viewing positions



Carrying position

To Set the Output Frequency

At power-on, the function generator outputs a sine wave at 1 kHz with an amplitude of 100 mV peak-to-peak (into a 50Ω termination).
The following steps show you how to change the frequency to 1.2 MHz.

1 Press the "Freq" softkey.

The displayed frequency is either the power-on value or the frequency previously selected. When you change functions, the same frequency is used if the present value is valid for the new function. To set the waveform *period* instead, press the **Freq** softkey again to toggle to the **Period** softkey (the current selection is highlighted).



2 Enter the magnitude of the desired frequency.

Using the numeric keypad, enter the value "1.2".



3 Select the desired units.

Press the softkey that corresponds to the desired units. When you select the units, the function generator outputs a waveform with the displayed frequency (if the output is enabled). For this example, press **MHz**.



Note: *You can also enter the desired value using the knob and arrow keys.*

To Set the Output Amplitude

At power-on, the function generator outputs a sine wave with an amplitude of 100 mV peak-to-peak (into a 50Ω termination).
The following steps show you how to change the amplitude to 50 mVrms.

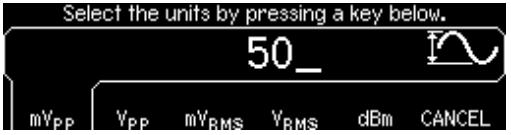
1 Press the "Ampl" softkey.

The displayed amplitude is either the power-on value or the amplitude previously selected. When you change functions, the same amplitude is used if the present value is valid for the new function. To set the amplitude using a *high level* and *low level*, press the **Ampl** softkey again to toggle to the **HiLevel** and **LoLevel** softkeys (the current selection is highlighted).



2 Enter the magnitude of the desired amplitude.

Using the numeric keypad, enter the value "50".



3 Select the desired units.

Press the softkey that corresponds to the desired units. When you select the units, the function generator outputs the waveform with the displayed amplitude (if the output is enabled). For this example, press **mV$_{RMS}$**.



Note: *You can also enter the desired value using the knob and arrow keys.*

You can easily convert the displayed amplitude from one unit to another. For example, the following steps show you how to convert the amplitude from Vrms to Vpp.

4 Enter the numeric entry mode.

Press the [+/-] key to enter the numeric entry mode.



5 Select the new units.

Press the softkey that corresponds to the desired units. The displayed value is converted to the new units. For this example, press the **Vpp** softkey to convert 50 mVrms to its equivalent in volts peak-to-peak.



To change the displayed amplitude by *decades*, press the right-arrow key to move the cursor to the units on the right side of the display. Then, rotate the knob to increase or decrease the displayed amplitude by decades.



To Set a DC Offset Voltage

At power-on, the function generator outputs a sine wave with a dc offset of 0 volts (into a 50Ω termination). *The following steps show you how to change the offset to –1.5 mVdc.*

1 Press the "Offset" softkey.

The displayed offset voltage is either the power-on value or the offset previously selected. When you change functions, the same offset is used if the present value is valid for the new function.



2 Enter the magnitude of the desired offset.

Using the numeric keypad, enter the value "–1.5".



3 Select the desired units.

Press the softkey that corresponds to the desired units. When you select the units, the function generator outputs the waveform with the displayed offset (if the output is enabled). For this example, press **mV_{DC}**.



Note: *You can also enter the desired value using the knob and arrow keys.*

Note: *To select dc volts from the front panel, press* Utility *and then select the* **DC On** *softkey. Press the* **Offset** *softkey to enter the desired voltage level.*

To Set the Duty Cycle

Applies only to square waves. At power-on, the duty cycle for square waves is 50%. You can adjust the duty cycle from 20% to 80% for output frequencies up to 25 MHz. *The following steps show you how to change the duty cycle to 30%.*

1 Select the square wave function.

Press the Square key and then set the desired output frequency to any value less than 25 MHz.

2 Press the "Duty Cycle" softkey.

The displayed duty cycle is either the power-on value or the percentage previously selected. The duty cycle represents the amount of time per cycle that the square wave is at a *high* level (note the icon on the right side of the display).



3 Enter the desired duty cycle.

Using the numeric keypad or the knob, select a duty cycle value of "30". The function generator adjusts the duty cycle immediately and outputs a square wave with the specified value (if the output is enabled).



To Configure a Pulse Waveform

You can configure the function generator to output a pulse waveform with variable pulse width and edge time. *The following steps show you how to configure a 500 ms pulse waveform with a pulse width of 10 ms and edge times of 50 µs.*

1 Select the pulse function.

Press the Pulse key to select the pulse function and output a pulse waveform with the default parameters.

2 Set the pulse period.

Press the **Period** softkey and then set the pulse period to 500 ms.



3 Set the pulse width.

Press the **Pulse Width** softkey and then set the pulse width to 10 ms. The pulse width represents the time from the 50% threshold of the rising edge to the 50% threshold of the next falling edge (note the display icon).



4 Set the edge time for both edges.

Press the **Edge Time** softkey and then set the edge time for *both* the rising and falling edges to 50 µs. The edge time represents the time from the 10% threshold to the 90% threshold of each edge (note the display icon).

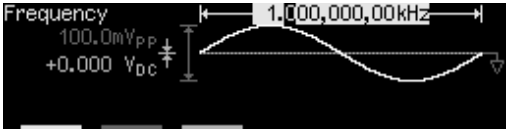


To View a Waveform Graph

In the *Graph Mode*, you can view a graphical representation of the current waveform parameters. Each softkey parameter is shown in a different color corresponding to the lines above the softkeys at the bottom of the display. Note that the softkeys are listed in the same order as in the normal display mode.

1 Enable the Graph Mode.

Press the Graph key to enable the Graph Mode. The name of the parameter currently selected is shown in the upper-left corner of the display and the numeric value is highlighted.



2 Select the desired parameter.

To select a specific parameter, note the colored bars above the softkeys at the bottom of the display and select the corresponding color. For example, to select amplitude, press the softkey below the magenta-colored bar.

- As in the normal display mode, you can edit numbers using either the numeric keypad or the knob and arrow keys.
- Parameters which normally toggle when you press a key a second time (e.g., **Freq**/**Period**) also toggle in the Graph Mode.
- To exit the Graph Mode, press Graph again.

The Graph key also serves as a Local key to restore front-panel control after remote interface operations.

To Output a Stored Arbitrary Waveform

There are five built-in arbitrary waveforms stored in non-volatile memory. *The following steps show you how to output the built-in "exponential fall" waveform from the front panel.*

> For information on creating a custom arbitrary waveform, refer to "To Create and Store an Arbitrary Waveform" on page 103.

1 Select the arbitrary waveform function.

When you press the Arb key to select the arbitrary waveform function, a temporary message is displayed indicating which waveform is currently selected (the default is "exponential rise").

2 Select the active waveform.

Press the **Select Wform** softkey and then press the **Built-In** softkey to select from the five built-in waveforms. Then press the **Exp Fall** softkey. The waveform is output using the present settings for frequency, amplitude, and offset unless you change them.



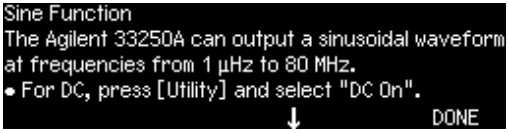
The selected waveform is now assigned to the Arb key. Whenever you press this key, the selected arbitrary waveform is output. To quickly determine which arbitrary waveform is currently selected, press Arb.

To Use the Built-In Help System

The built-in help system is designed to provide context-sensitive assistance on any front-panel key or menu softkey. A list of help topics is also available to assist you with several front-panel operations.

1 View the help information for a function key.

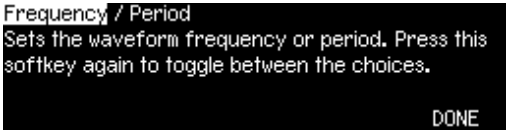
Press and hold down the Sine key. If the message contains more information than will fit on the display, press the ↓ softkey or turn the knob clockwise to view the remaining information.



Press **DONE** to exit the help menu.

2 View the help information for a menu softkey.

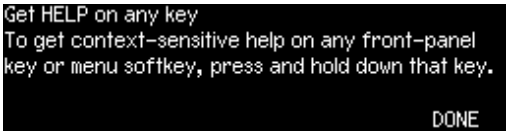
Press and hold down the **Freq** softkey. If the message contains more information than will fit on the display, press the ↓ softkey or rotate the knob clockwise to view the remaining information.



Press **DONE** to exit the help menu.

3 View the list of help topics.

Press the Help key to view the list of available help topics. To scroll through the list, press the ↑ or ↓ softkey or rotate the knob. Select the third topic "*Get HELP on any key*" and then press **SELECT**.

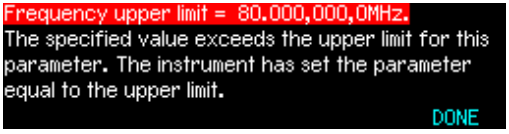


Press **DONE** to exit the help menu.

4 View the help information for displayed messages.

Whenever a limit is exceeded or any other invalid configuration is found, the function generator will display a message. For example, if you enter a value that exceeds the frequency limit for the selected function, a message will be displayed. The built-in help system provides additional information on the most recent message to be displayed.

Press the Help key, select the first topic "*View the last message displayed*", and then press **SELECT**.



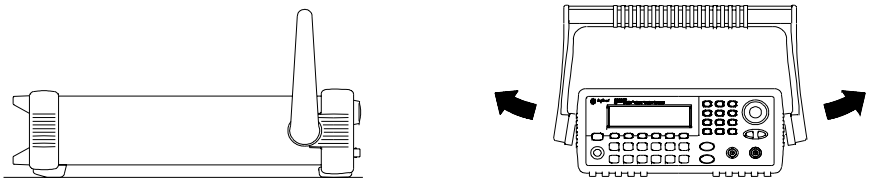
Local Language Help: The built-in help system in available in multiple languages. All messages, context-sensitive help, and help topics appear in the selected language. The menu softkey labels and status line messages are not translated.

To select the local language, press the Utility key, press the **System** softkey, and then press the **Help In** softkey. Select the desired language.

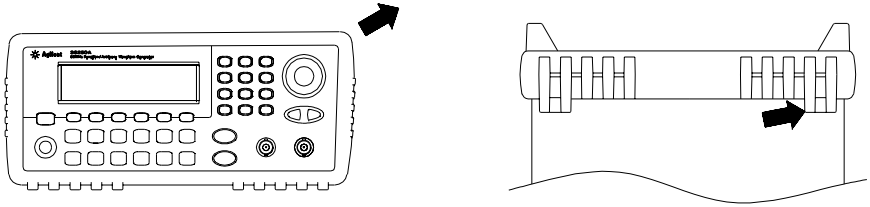
To Rack Mount the Function Generator

You can mount the Agilent 33250A in a standard 19-inch rack cabinet using one of two optional kits available. Instructions and mounting hardware are included with each rack-mounting kit. Any Agilent *System II* instrument of the same size can be rack-mounted beside the Agilent 33250A.

Note: *Remove the carrying handle, and the front and rear rubber bumpers, before rack-mounting the instrument.*



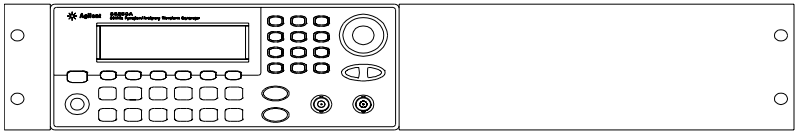
To remove the handle, rotate it to vertical and pull the ends outward.



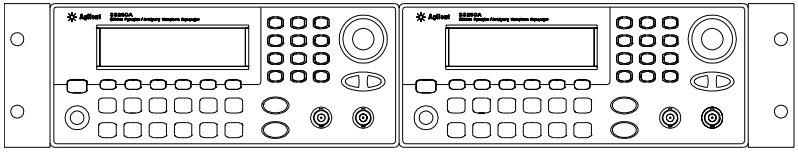
Front

Rear (bottom view)

To remove the rubber bumper, stretch a corner and then slide it off.



To rack mount a single instrument, order adapter kit 5063-9240.



To rack mount two instruments side-by-side, order lock-link kit 5061-9694 and flange kit 5063-9212. Be sure to use the support rails in the rack cabinet.

In order to prevent overheating, do not block the flow of air into or out of the instrument. Be sure to allow enough clearance at the rear, sides, and bottom of the instrument to permit adequate internal air flow.

Front-Panel Menu Operation

Front-Panel Menu Operation

2

This chapter introduces you to the front-panel keys and menu operation. This chapter does not give a detailed description of every front-panel key or menu operation. It does, however, give you an overview of the front-panel menus and many front-panel operations. See chapter 3 “Features and Functions,” starting on page 47, for a complete discussion of the function generator’s capabilities and operation.

- Front-Panel Menu Reference, *on page 31*
- To Select the Output Termination, *on page 33*
- To Reset the Function Generator, *on page 33*
- To Output a Modulated Waveform, *on page 34*
- To Output an FSK Waveform, *on page 36*
- To Output a Frequency Sweep, *on page 38*
- To Output a Burst Waveform, *on page 40*
- To Trigger a Sweep or Burst, *on page 42*
- To Store the Instrument State, *on page 43*
- To Configure the Remote Interface, *on page 44*

Front-Panel Menu Reference

This section gives an overview of the front-panel menus. The remainder of this chapter contains examples of using the front-panel menus.

Mod **Configure the modulation parameters for AM, FM, and FSK.**

- Select the modulation type.
- Select an internal or external modulation source.
- Specify the AM modulation depth, modulating frequency, and modulation shape.
- Specify the FM frequency deviation, modulating frequency, and modulation shape.
- Specify the FSK “hop” frequency and FSK rate.

Sweep **Configure the parameters for frequency sweep.**

- Select linear or logarithmic sweeping.
- Select the start/stop frequencies or center/span frequencies.
- Select the time in seconds required to complete a sweep.
- Specify a marker frequency.
- Specify an internal or external trigger source for the sweep.
- Specify the slope (rising or falling edge) for an external trigger source.
- Specify the slope (rising or falling edge) of the “Trig Out” signal.

Burst **Configure the parameters for burst.**

- Select the triggered (N Cycle) or externally-gated burst mode.
- Select the number of cycles per burst (1 to 1,000,000, or Infinite).
- Select the starting phase angle of the burst (-360° to +360°).
- Specify the time from the start of one burst to the start of the next burst.
- Specify a delay between the trigger and the start of the burst.
- Specify an internal or external trigger source for the burst.
- Specify the slope (rising or falling edge) for an external trigger source.
- Specify the slope (rising or falling edge) of the “Trig Out” signal.

2

Store/Recall **Store and recall instrument states.**

- Store up to four instrument states in non-volatile memory.
- Assign a custom name to each storage location.
- Recall stored instrument states.
- Restore all instrument settings to their factory default values.
- Select the instrument's power-on configuration (last or factory default).

Utility **Configure system-related parameters.**

- Generate a dc-only voltage level.
- Enable/disable the Sync signal which is output from the "Sync" connector.
- Select the output termination (1Ω to 10 kΩ, or Infinite).
- Enable/disable amplitude autoranging.
- Select the waveform polarity (normal or inverted).
- Select the GPIB address.
- Configure the RS-232 interface (baud rate, parity, and handshake mode).
- Select how periods and commas are used in numbers displayed on the front panel.
- Select the local language for front-panel messages and help text.
- Enable/disable the tone heard when an error is generated.
- Enable/disable the display bulb-saver mode.
- Adjust the contrast setting of the front-panel display.
- Perform an instrument self-test.
- Secure/unsecure the instrument for calibration and perform manual calibrations.
- Query the instrument's firmware revision codes.

Help **View the list of Help topics.**

- View the last message displayed.
- View the remote command error queue.
- Get HELP on any key.
- How to generate a dc-only voltage level.
- How to generate a modulated waveform.
- How to create an arbitrary waveform.
- How to reset the instrument to its default state.
- How to view a waveform in the Graph Mode.
- How to synchronize multiple instruments.
- How to obtain Agilent Technical Support.

To Select the Output Termination

The Agilent 33250A has a fixed series output impedance of 50 ohms to the front-panel *Output* connector. If the actual load impedance is different than the value specified, the displayed amplitude and offset levels will be incorrect. The load impedance setting is simply provided as a convenience to ensure that the displayed voltage matches the expected load.

1 Press Utility.

2 Navigate the menu to set the output termination.

Press the **Output Setup** softkey and then select the **Load** softkey.



3 Select the desired output termination.

Use the knob or numeric keypad to select the desired load impedance or press the **Load** softkey again to choose "High Z".

To Reset the Function Generator

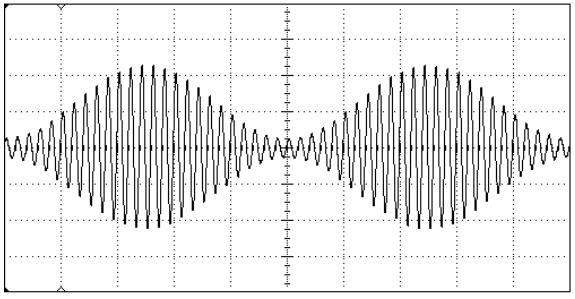
To reset the instrument to its factory default state, press Store/Recall and then select the **Set to Defaults** softkey. Press **YES** to confirm the operation.

For a complete listing of the instrument's power-on and reset conditions, see "Factory Default Settings" on page 127.

2

To Output a Modulated Waveform

A modulated waveform consists of a *carrier* and a *modulating waveform*. In AM (amplitude modulation), the amplitude of the carrier is varied by the amplitude of the modulating waveform. For this example, you will output an AM waveform with 80% modulation depth. The carrier will be a 5 kHz sine wave and the modulating waveform will be a 200 Hz sine wave.



1 Select the function, frequency, and amplitude of the carrier.

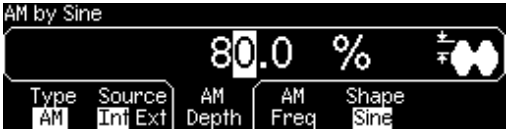
Press Sine and then press the **Freq**, **Ampl**, and **Offset** softkeys to configure the carrier waveform. *For this example, select a 5 kHz sine wave with an amplitude of 5 Vpp.*

2 Select AM.

Press Mod and then select "AM" using the **Type** softkey. Notice that a status message "AM by Sine" is shown in the upper-left corner of the display.

3 Set the modulation depth.

Press the **AM Depth** softkey and then set the value to 80% using the numeric keypad or the knob and arrow keys.



4 Set the modulating frequency.

Press the **AM Freq** softkey and then set the value to 200 Hz using the numeric keypad or the knob and arrow keys.



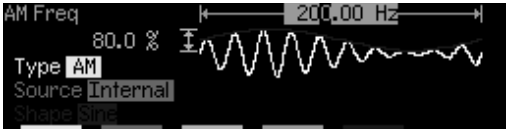
5 Select the modulating waveform shape.

Press the **Shape** softkey to select the shape of the modulating waveform. For this example, select a sine wave.

At this point, the function generator outputs an AM waveform with the specified modulation parameters (if the output is enabled).

6 View the waveform.

Press Graph to view the waveform parameters.

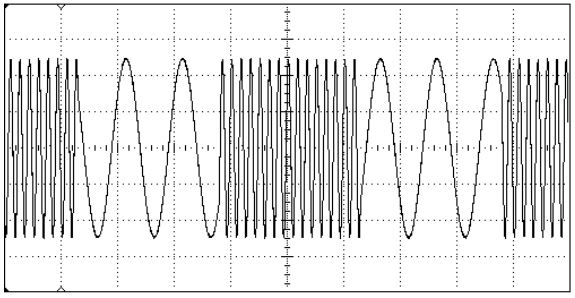


To turn off the Graph Mode, press Graph again.

2

To Output an FSK Waveform

You can configure the function generator to "shift" its output frequency between two preset values using FSK modulation. The rate at which the output shifts between the two frequencies (called the "carrier frequency" and the "hop frequency") is determined by the internal rate generator or the signal level on the rear-panel *Trig In* connector. *For this example, you will set the "carrier" frequency to 3 kHz and the "hop" frequency to 500 Hz, with an FSK rate of 100 Hz.*



1 Select the function, frequency, and amplitude of the carrier.

Press Sine and then press the **Freq**, **Ampl**, and **Offset** softkeys to configure the carrier waveform. *For this example, select a 3 kHz sine wave with an amplitude of 5 Vpp.*

2 Select FSK.

Press Mod and then select "FSK" using the **Type** softkey. Notice that a status message "FSK" is shown in the upper-left corner of the display.

3 Set the "hop" frequency.

Press the **Hop Freq** softkey and then set the value to 500 Hz using the numeric keypad or the knob and arrow keys.



4 Set the FSK "shift" rate.

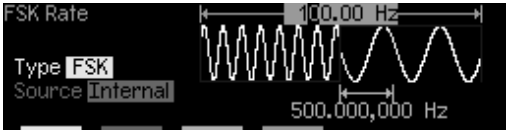
Press the **FSK Rate** softkey and then set the value to 100 Hz using the numeric keypad or the knob and arrow keys.



At this point, the function generator outputs an FSK waveform (if the output is enabled).

5 View the waveform.

Press Graph to view the waveform parameters.

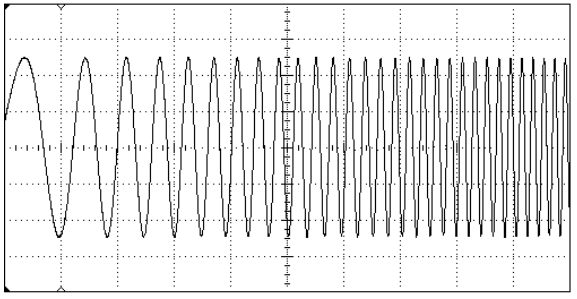


To turn off the Graph Mode, press Graph again.

2

To Output a Frequency Sweep

In the frequency sweep mode, the function generator "steps" from the *start frequency* to the *stop frequency* at a *sweep rate* which you specify. You can sweep up or down in frequency, and with either linear or logarithmic spacing. *For this example, you will output a swept sine wave from 50 Hz to 5 kHz. You will not change the other parameters from their default settings: internal sweep trigger, linear spacing, and 1 second sweep time.*



1 Select the function and amplitude for the sweep.

For sweeps, you can select sine, square, ramp, or arbitrary waveforms (pulse, noise, and dc are not allowed). *For this example, select a sine wave with an amplitude of 5 Vpp.*

2 Select the sweep mode.

Press Sweep and then verify that the linear sweep mode is currently selected. Notice that a status message "Linear Sweep" is shown in the upper-left corner of the display.

3 Set the start frequency.

Press the **Start** softkey and then set the value to 50 Hz using the numeric keypad or the knob and arrow keys.



4 Set the stop frequency.

Press the **Stop** softkey and then set the value to 5 kHz using the numeric keypad or the knob and arrow keys.

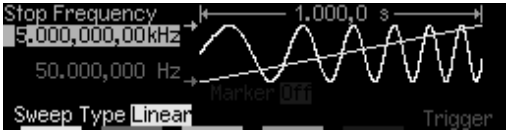


At this point, the function generator outputs a continuous sweep from 50 Hz to 5 kHz (if the output is enabled).

Note: If desired, you can set the frequency boundaries of the sweep using a *center frequency* and *frequency span*. These parameters are similar to the start frequency and stop frequency and are included to give you added flexibility. To achieve the same results, set the center frequency to 2.525 kHz and the frequency span to 4.950 kHz.

5 View the waveform.

Press Graph to view the waveform parameters.



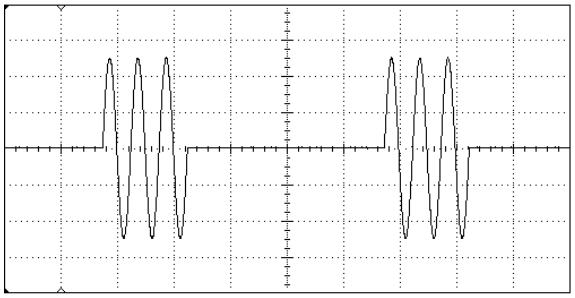
To turn off the Graph Mode, press Graph again.

> You can generate a single frequency sweep by pressing the Trigger key. For more information, see “To Trigger a Sweep or Burst” on page 42.

2

To Output a Burst Waveform

You can configure the function generator to output a waveform with a specified number of cycles, called a *burst*. You can output the burst at a rate determined by the internal rate generator or the signal level on the rear-panel *Trig In* connector. *For this example, you will output a three-cycle sine wave with a 20 ms burst period. You will not change the other parameters from their default settings: internal burst source and 0 degree starting phase.*



1 Select the function and amplitude for the burst.

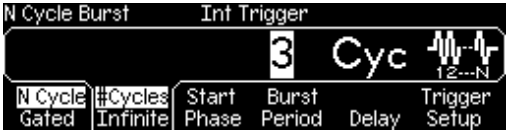
For burst waveforms, you can select sine, square, ramp, pulse, or arbitrary waveforms (noise is allowed only in the "gated" burst mode and dc is not allowed). *For this example, select a sine wave with an amplitude of 5 Vpp.*

2 Select the burst mode.

Press Burst and then verify that the "N Cycle" (internally-triggered) mode is currently selected. Notice that a status message "N Cycle Burst" is shown in the upper-left corner of the display.

3 Set the burst count.

Press the **#Cycles** softkey and then set the count to "3" using the numeric keypad or knob.



4 Set the burst period.

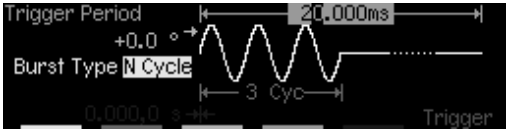
Press the **Burst Period** softkey and then set the period to 20 ms using the numeric keypad or the knob and arrow keys. The burst period sets the time from the start of one burst to the start of the next burst (note the display icon).



At this point, the function generator outputs a continuous three-cycle burst (if the output is enabled).

5 View the waveform.

Press Graph to view the waveform parameters.



To turn off the Graph Mode, press Graph again.

> You can generate a single burst (with the specified count) by pressing the Trigger key. For more information, see "To Trigger a Sweep or Burst" on page 42.
>
> You can also use an external gate signal to either turn the output signal "on" or "off" based on the external signal applied to the rear-panel *Trig In* connector. For more information, see "Burst Mode" on page 89.

2

To Trigger a Sweep or Burst

You can issue triggers from the front panel for sweeps and bursts using a *manual* trigger or an *internal* trigger.

- *Internal* or "automatic" triggering is enabled with the default settings of the function generator. In this mode, the function generator outputs continuously when the sweep or burst mode is selected.
- *Manual* triggering initiates one sweep or outputs one burst each time you press the Trigger key from the front-panel. Continue pressing this key to re-trigger the function generator.
- The Trigger key is disabled when in remote (the remote icon turns on while in remote) and when a function other than sweep or burst is currently selected (or when the output is disabled). The Trigger key flashes off momentarily when using a manual trigger.

To Store the Instrument State

You can store the instrument state in one of four non-volatile storage locations. A fifth storage location automatically holds the power-down configuration of the instrument. When power is restored, the instrument can automatically return to its state before power-down.

1 Select the desired storage location.

Press Store/Recall and then select the **Store State** softkey.



2 Select a custom name for the selected location.

If desired, you can assign a custom name to each of the four locations.



- The name can contain up to 12 characters. The first character must be a letter but the remaining characters can be letters, numbers, or the underscore character ("_").
- To add additional characters, press the right-arrow key until the cursor is to the right of the existing name and then turn the knob.
- To delete all characters to the right of the cursor position, press +/-.
- To use numbers in the name, you can enter them directly from the numeric keypad. Use the decimal point from the numeric keypad to add the underscore character ("_") to the name.

3 Store the instrument state.

Press the **STORE STATE** softkey. The instrument stores the selected function, frequency, amplitude, dc offset, duty cycle, symmetry, as well as any modulation parameters in use. The instrument *does not* store volatile waveforms created in the arbitrary waveform function.

2

To Configure the Remote Interface

The instrument is shipped with both a GPIB (IEEE-488) interface and an RS-232 interface. Only one interface can be enabled at a time. The GPIB interface is selected when the instrument is shipped from the factory.

GPIB Configuration

1 Select the GPIB interface.

Press (Utility) and then select the **GPIB** softkey from the "I/O" menu.



2 Select the GPIB address.

Press the **GPIB Address** softkey and enter the desired address using the numeric keypad or knob. The factory setting is "10".

The GPIB address is shown on the front-panel display at power-on.

3 Exit the menu.

Press the **DONE** softkey.

RS-232 Configuration

1 Select the RS-232 interface.

Press (Utility) and then select the **RS-232** softkey from the "I/O" menu.



2 Set the baud rate.

Press the **Baud Rate** softkey and select one of the following:
300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600 (*factory setting*), or 115200 baud.

3 Select the parity and number of data bits.

Press the **Parity/# Bits** softkey and select one of the following:
None (8 data bits, *factory setting*), Even (7 data bits), or Odd (7 data bits).
When you set the parity, you are also setting the number of data bits.

4 Select the handshake mode.

Press the **Handshake** softkey and select one of the following:
None, DTR/DSR (*factory setting*), Modem, RTS/CTS, or XON/XOFF.

5 Exit the menu.

Press the **DONE** softkey.

Features and Functions

Features and Functions

3

You will find that this chapter makes it easy to look up all the details about a particular feature of the function generator. Whether you are operating the function generator from the front panel or over the remote interface, this chapter will be useful. This chapter is divided into the following sections:

- Output Configuration, *on page 49*
- Pulse Waveforms, *on page 64*
- Amplitude Modulation (AM), *on page 67*
- Frequency Modulation (FM), *on page 72*
- Frequency-Shift Keying (FSK) Modulation, *on page 78*
- Frequency Sweep, *on page 82*
- Burst Mode, *on page 89*
- Triggering, *on page 98*
- Arbitrary Waveforms, *on page 103*
- System-Related Operations, *on page 109*
- Remote Interface Configuration, *on page 118*
- Calibration Overview, *on page 123*
- Factory Default Settings, *on page 127*

Some knowledge of the front-panel menus will be helpful before you read this chapter. If you have not already read chapter 2, "Front-Panel Menu Operation," starting on page 29, you may want to read it now. Chapter 4, "Remote Interface Reference," starting on page 129, lists the syntax for the SCPI commands available to program the function generator.

Throughout this manual, the following conventions are used for SCPI command syntax for remote interface programming:

- Square brackets (**[ ]**) indicate optional keywords or parameters.
- Braces (**{ }**) enclose parameters within a command string.
- Triangle brackets (**< >**) enclose parameters for which you must substitute a value.
- A vertical bar (**|**) separates multiple parameter choices.

Output Configuration

This section contains information to help you configure the function generator for outputting waveforms. You may never have to change some of the parameters discussed here, but they are provided to give you the flexibility you might need.

Output Function

The function generator can output five standard waveforms including sine, square, ramp, pulse, and noise. You can also select one of five built-in arbitrary waveforms or create your own custom waveforms. You can internally modulate any of the standard waveforms (except pulse and noise) and also arbitrary waveforms using AM, FM, or FSK. Linear or logarithmic frequency sweeping is available for any of the standard waveforms (except pulse and noise) and also arbitrary waveforms. You can generate a burst waveform using any of the standard waveforms and also arbitrary waveforms. *The default function is sine wave*.

- The table below shows which output functions are allowed with modulation, sweep, and burst. Each " • " indicates a valid combination. If you change to a function that is not allowed with modulation, sweep, or burst, then the modulation or mode is turned off.

	Sine	Square	Ramp	Pulse	Noise	DC	Arb
AM, FM Carrier	•	•	•				•
FSK Carrier	•	•	•				•
Sweep Mode	•	•	•				•
Burst Mode	•	•	•	•	•[1]		•

[1] *Allowed in the External Gated burst mode only*.

- *Function Limitations*: If you change to a function whose maximum frequency is less than that of the current function, the frequency is adjusted to the maximum value for the new function. For example, if you are currently outputting an 80 MHz sine wave and then change to the ramp function, the function generator will automatically adjust the output frequency to 1 MHz (the upper limit for ramps).

- *Amplitude Limitations*: If you change to a function whose maximum amplitude is less than that of the current function, the amplitude is automatically adjusted to the maximum value for the new function. This may occur when the output units are *Vrms* or *dBm* due to the differences in crest factor for the various output functions.

  For example, if you output a 5 Vrms square wave (into 50 ohms) and then change to the sine wave function, the function generator will automatically adjust the output amplitude to 3.536 Vrms (the upper limit for sine in Vrms).

- *Front-Panel Operation*: To select a function, press any key in the top row of function keys. Press (Arb) to output the arbitrary waveform currently selected. To view the other arbitrary waveform choices, press the **Select Wform** softkey.

  To select *dc volts* from the front panel, press (Utility) and then select the **DC On** softkey. Press the **Offset** softkey to enter the desired offset voltage level.

- *Remote Interface Operation*:

  ```
  FUNCtion:SHAPe {SINusoid|SQUare|RAMP|PULSe|NOISe|DC|USER}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Output Frequency

As shown below, the output frequency range depends on the function currently selected. *The default frequency is 1 kHz for all functions*.

Function	Minimum Frequency	Maximum Frequency
Sine	1 μHz	80 MHz
Square	1 μHz	80 MHz
Ramp	1 μHz	1 MHz
Pulse	500 μHz	50 MHz
Noise, DC	Not Applicable	Not Applicable
Arbs	1 μHz	25 MHz

- *Function Limitations*: If you change to a function whose maximum frequency is less than that of the current function, the frequency is adjusted to the maximum value for the new function. For example, if you are currently outputting an 80 MHz sine wave and then change to the ramp function, the function generator will automatically adjust the output frequency to 1 MHz (the upper limit for ramps).
- *Burst Limitation*: For internally-triggered bursts, the minimum frequency is 2 mHz. For sine and square waveforms, frequencies above 25 MHz are allowed only with an "infinite" burst count.
- *Duty Cycle Limitations*: For square waveforms, the function generator may not be able to use the full range of duty cycle values at higher frequencies as shown below.

  20% to 80% (*frequency* $\leq$ 25 MHz)
  40% to 60% (25 MHz < *frequency* $\leq$ 50 MHz)
  50% (*frequency* > 50 MHz)

  If you change to a frequency that cannot produce the current duty cycle, the duty cycle is automatically adjusted to the maximum value for the new frequency. For example, if you currently have the duty cycle set to 70% and then change the frequency to 60 MHz, the function generator will automatically adjust the duty cycle to 50% (the upper limit for this frequency).

- *Front-Panel Operation*: To set the output frequency, press the **Freq** softkey for the selected function. Then use the knob or numeric keypad to enter the desired frequency. To set the waveform period instead, press the **Freq** softkey again to toggle to the **Period** softkey.
- *Remote Interface Operation*:

  ```
  FREQuency {<frequency>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Output Amplitude

The default amplitude is 100 mVpp (into 50 ohms) for all functions.

- *Offset Voltage Limitations*: The relationship between output amplitude and offset voltage is shown below. V*max* is the maximum *peak voltage* for the selected output termination (5 volts for a 50Ω load or 10 volts for a high-impedance load).

  $$Vpp \leq 2 \times (Vmax - |Voffset|)$$

- *Limits Due to Output Termination*: If you change the output termination setting, the displayed output amplitude will be adjusted (and no error will be generated). For example, if you set the amplitude to 10 Vpp and then change the output termination from 50 ohms to "high impedance", the amplitude displayed on the function generator's front-panel will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half. *For more information, see "Output Termination" on page 57.*
- *Limits Due to Units Selection*: In some cases, the amplitude limits are determined by the output units selected. This may occur when the units are *Vrms* or *dBm* due to the differences in crest factor for the various output functions. For example, if you output a 5 Vrms square wave (into 50 ohms) and then change to the sine wave function, the function generator will automatically adjust the output amplitude to 3.536 Vrms (the upper limit for sine waves in Vrms).

- You can set the output amplitude in Vpp, Vrms, or dBm. *For more information, see "Output Units" on page 56.*
- You cannot specify the output amplitude in dBm if the output termination is currently set to "high impedance". The units are automatically converted to Vpp. *For more information, see "Output Units" on page 56.*
- *Arbitrary Waveform Limitations*: For arbitrary waveforms, the maximum amplitude will be limited if the waveform data points do not span the full range of the output DAC (Digital- to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is limited to 6.087 Vpp (into 50 ohms).
- While changing amplitude, you may notice a momentary disruption in the output waveform at certain voltages due to switching of the output attenuators. The amplitude is controlled, however, so the output voltage will never exceed the current setting while switching ranges. To prevent this disruption in the output, you can disable the voltage autoranging feature as described on page 60.
- You can also set the amplitude (with an associated offset voltage) by specifying a high level and low level. For example, if you set the high level to +2 volts and the low level to -3 volts, the resulting amplitude is 5 Vpp (with an offset voltage of -500 mV).
- For dc volts, the output level is actually controlled by setting the offset voltage. You can set the dc level to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit. *See "DC Offset Voltage" on the following page for more information.*

  To select *dc volts* from the front panel, press (Utility) and then select the **DC On** softkey. Press the **Offset** softkey to set the desired offset voltage level.
- *Front-Panel Operation*: To set the output amplitude, press the **Ampl** softkey for the selected function. Then use the knob or numeric keypad to enter the desired amplitude. To set the amplitude using a high level and low level, press the **Ampl** softkey again to toggle to the **HiLevel** and **LoLevel** softkeys.

- *Remote Interface Operation*:

  ```
  VOLTage {<amplitude>|MINimum|MAXimum}
  ```

  Or, you can set the amplitude by specifying a high level and low level using the following commands.

  ```
  VOLTage:HIGH {<voltage>|MINimum|MAXimum}
  VOLTage:LOW {<voltage>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

DC Offset Voltage

The default offset is 0 volts for all functions.

- *Limits Due to Amplitude*: The relationship between offset voltage and output amplitude is shown below. V*max* is the maximum *peak voltage* for the selected output termination (5 volts for a 50Ω load or 10 volts for a high-impedance load).

  $$|\mathrm{V}offset| \leq \mathrm{V}max - \frac{\mathrm{V}pp}{2}$$

  If the specified offset voltage is not valid, the function generator will automatically adjust it to the maximum dc voltage allowed with the amplitude specified.

- *Limits Due to Output Termination*: The offset limits are determined by the current output termination setting. For example, if you set the offset to 100 mVdc and then change the output termination from 50 ohms to "high impedance", the offset voltage displayed on the function generator's front-panel will *double* to 200 mVdc (and no error will be generated). If you change from "high impedance" to 50 ohms, the displayed offset will drop in half. *See "Output Termination" on page 57 for more information.*

- *Arbitrary Waveform Limitations*: For arbitrary waveforms, the maximum offset and amplitude will be limited if the waveform data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum offset is limited to 4.95 volts (into 50 ohms).
- You can also set the offset by specifying a high level and low level. For example, if you set the high level to +2 volts and the low level to -3 volts, the resulting amplitude is 5 Vpp (with an offset voltage of -500 mV).
- For dc volts, the output level is actually controlled by setting the offset voltage. You can set the dc level to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit.

  To select *dc volts* from the front panel, press Utility and then select the **DC On** softkey. Press the **Offset** softkey to set the desired offset voltage level.
- *Front-Panel Operation*: To set the dc offset, press the **Offset** softkey for the selected function. Then use the knob or numeric keypad to enter the desired offset. To set the offset using a high level and low level, press the **Offset** softkey again to toggle to the **HiLevel** and **LoLevel** softkeys.
- *Remote Interface Operation*:

  ```
  VOLTage:OFFSet {<offset>|MINimum|MAXimum}
  ```

  Or, you can set the offset by specifying a high level and low level using the following commands.

  ```
  VOLTage:HIGH {<voltage>|MINimum|MAXimum}
  VOLTage:LOW {<voltage>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Output Units

Applies to output amplitude only. At power-on, the units for output amplitude are volts peak-to-peak.

- Output units: **Vpp**, Vrms, or dBm. *The default is Vpp.*
- The unit setting is stored in *volatile* memory; the units are set to "Vpp" when power has been off or after a remote interface reset.
- The function generator uses the current units selection for both front panel and remote interface operations. For example, if you select "VRMS" from the remote interface, the units are displayed as "VRMS" on the front panel.
- The output units for amplitude cannot be set to dBm if the output termination is currently set to "high impedance". The units are automatically converted to Vpp.
- *Front-Panel Operation*: Use the numeric keypad to enter the desired magnitude and then press the appropriate softkey to select the units. You can also convert from one unit to another from the front panel. For example, to convert 2 Vpp to its equivalent value in Vrms, press (+/-) and then press the **V_{RMS}** softkey. The converted value is 707.1 mVrms for a sine wave.
- *Remote Interface Operation*:

```
VOLTage:UNIT {VPP|VRMS|DBM}
```

Output Termination

Applies to output amplitude and offset voltage only. The Agilent 33250A has a fixed series output impedance of 50 ohms to the front-panel *Output* connector. If the actual load impedance is different than the value specified, the displayed amplitude and offset levels will be incorrect.

- Output termination: 1Ω to 10 kΩ, or Infinite. The default is 50Ω.
- The output termination setting is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset. When shipped from the factory, 50Ω is selected.
- If you specify a 50-ohm termination but are actually terminating into an open circuit, the actual output will be *twice* the value specified. For example, if you set the offset to 100 mVdc (and specify a 50-ohm load) but are terminating the output into an open circuit, the actual offset will be 200 mVdc.
- If you change the output termination setting, the displayed output amplitude and offset levels are automatically adjusted (no error will be generated). For example, if you set the amplitude to 10 Vpp and then change the output termination from 50 ohms to "high impedance", the amplitude displayed on the function generator's front-panel will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half.
- You cannot specify the output amplitude in dBm if the output termination is currently set to "high impedance". The units are automatically converted to Vpp.
- *Front-Panel Operation*: Press Utility and select the **Output Setup** softkey. Then use the knob or numeric keypad to select the desired load impedance or press the **Load** softkey again to choose "High Z".
- *Remote Interface Operation*:

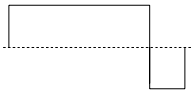
```
OUTPut:LOAD {<ohms>|INFinity|MINimum|MAXimum}
```

Duty Cycle

Applies to square waves only. Duty cycle represents the amount of time per cycle that the square wave is at a *high level* (assuming that the waveform polarity is not reversed).



20% Duty Cycle



80% Duty Cycle

- Duty Cycle: 20% to 80% (*frequency* ≤ 25 MHz)
  40% to 60% (25 MHz < *frequency* ≤ 50 MHz)
  50% (*frequency* > 50 MHz)
- The duty cycle is stored in *volatile* memory; the duty cycle is set to 50% when power has been off or after a remote interface reset.
- The duty cycle setting is remembered when you change from square wave to another function. When you return to the square function, the previous duty cycle is used.
- *Limits Due to Frequency*: If the square wave function is selected and you change to a frequency that cannot produce the current duty cycle, the duty cycle is automatically adjusted to the maximum value for the new frequency. For example, if you currently have the duty cycle set to 70% and then change the frequency to 60 MHz, the function generator will automatically adjust the duty cycle to 50% (the upper limit for this frequency).
- If you select a square waveform as the *modulating* waveform for AM or FM, the duty cycle setting *does not* apply. The function generator always uses a square waveform with a 50% duty cycle.
- *Front-Panel Operation*: After selecting the square wave function, press the **Duty Cycle** softkey. Then use the knob or numeric keypad to enter the desired duty cycle.
- *Remote Interface Operation*:

  ```
  FUNCtion:SQUare:DCYCle {<percent>|MINimum|MAXimum}
  ```

  The `APPLy` command automatically sets the duty cycle to 50%.

Symmetry

Applies to ramp waves only. Symmetry represents the amount of time per cycle that the ramp wave is *rising* (assuming that the waveform polarity is not reversed).



0% Symmetry **100% Symmetry**

- The symmetry is stored in *volatile* memory; the symmetry is set to 100% when power has been off or after a remote interface reset.
- The symmetry setting is remembered when you change from ramp wave to another function. When you return to the ramp function, the previous symmetry is used.
- If you select a ramp waveform as the *modulating* waveform for AM or FM, the symmetry setting *does not* apply.
- *Front-Panel Operation*: After selecting the ramp function, press the **Symmetry** softkey. Then use the knob or numeric keypad to enter the desired symmetry.
- *Remote Interface Operation*:

  ```
  FUNCtion:RAMP:SYMMetry {<percent>|MINimum|MAXimum}
  ```

  The `APPLy` command automatically sets the symmetry to 100%.

Voltage Autoranging

In the default mode, autoranging is enabled and the function generator automatically selects the optimal settings for the output amplifier and attenuators. With autoranging disabled, the function generator uses the current amplifier and attenuator settings.

- One advantage of disabling autoranging is to eliminate momentary disruptions caused by switching of the attenuators while changing amplitude. However, the amplitude and offset accuracy and resolution (as well as waveform fidelity) may be adversely affected when reducing the amplitude below the expected range change.
- *Front-Panel Operation*: Press Utility and select the **Output Setup** softkey. Then press the **Range** softkey again to toggle between the "Auto" and "Hold" selections.
- *Remote Interface Operation*:

  ```
  VOLTage:RANGe:AUTO {OFF|ON|ONCE}
  ```

  The `APPLy` command overrides the voltage autorange setting and automatically enables autoranging.

Output Control

You can disable or enable the front-panel *Output* connector. At power-on, the output is disabled. When enabled, the Output key is illuminated.

- If an excessive external voltage is applied to the front-panel *Output* connector, an error message will be displayed and the output will be disabled. To re-enable the output, remove the overload from the *Output* connector and press Output to enable the output.
- *Front-Panel Operation*: Press Output to enable or disable the output.
- *Remote Interface Operation*:

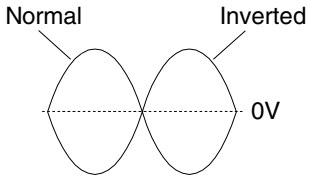
  ```
  OUTPut {OFF|ON}
  ```

  The `APPLy` command overrides the current setting and automatically enables the *Output* connector.

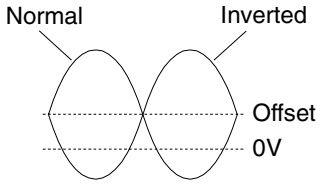
Waveform Polarity

In the *normal* mode (default), the waveform goes positive during the first part of the cycle. In the *inverted* mode, the waveform goes negative during the first part of the cycle.

- As shown in the examples below, the waveform is inverted *relative to* the offset voltage. Any offset voltage present will remain unchanged when the waveform is inverted.



No Offset Voltage



With Offset Voltage

- When a waveform is inverted, the Sync signal associated with the waveform *is not* inverted.
- *Front-Panel Operation*: Press Utility and select the **Output Setup** softkey. Then press the **Normal** softkey again to toggle between the "Normal" and "Invert" selections.
- *Remote Interface Operation*:

```
OUTPut:POLarity {NORMal|INVerted}
```

Sync Output Signal

A sync output is provided on the front-panel *Sync* connector. All of the standard output functions (except dc and noise) have an associated Sync signal. For certain applications where you may not want to output the Sync signal, you can disable the *Sync* connector.

- By default, the Sync signal is routed to the *Sync* connector (enabled). When the Sync signal is disabled, the output level on the *Sync* connector is at a logic "low" level.
- When a waveform is inverted (see "Waveform Polarity" on the previous page), the Sync signal associated with the waveform *is not* inverted.
- The Sync signal setting is overridden by the setting of the Marker frequency used with the sweep mode (*see page 86*). Therefore, when the marker frequency is enabled (and the sweep mode is also enabled), the Sync signal setting is ignored.
- For *sine*, *ramp*, and *pulse* waveforms, the Sync signal is a square waveform with a 50% duty cycle. The Sync signal is a TTL "high" when the waveform's output is positive, relative to zero volts (or the dc offset value). The Sync signal is a TTL "low" when the output is negative, relative to zero volts (or the dc offset value).
- For *square* waveforms, the Sync signal is a square waveform with the same duty cycle as the main output. The Sync signal is a TTL "high" when the waveform's output is positive, relative to zero volts (or the dc offset value). The Sync signal is a TTL "low" when the output is negative, relative to zero volts (or the dc offset value).
- For *arbitrary* waveforms, the Sync signal is a square waveform with a 50% duty cycle. The Sync signal is a TTL "high" when the first downloaded waveform point is output.
- For internally-modulated *AM* and *FM*, the Sync signal is referenced to the modulating waveform (not the carrier) and is a square waveform with a 50% duty cycle. The Sync signal is a TTL "high" during the first half of the modulating waveform.
- For externally-modulated *AM* and *FM*, the Sync signal is referenced to the carrier waveform (not the modulating waveform) and is a square waveform with a 50% duty cycle.

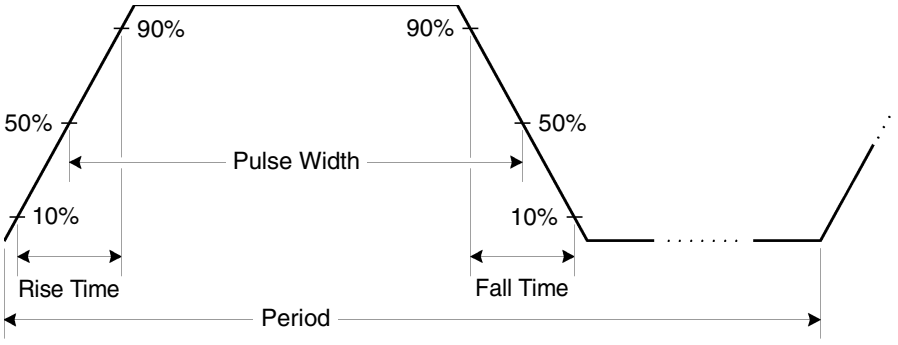
- For *FSK*, the Sync signal is referenced to the "hop" frequency and is a square waveform with a 50% duty cycle. The Sync signal is a TTL "high" on the transition to the "hop" frequency.
- For *frequency sweeps* with *Marker Off*, the Sync signal is a square waveform with a 50% duty cycle. The Sync signal is a TTL "high" at the beginning of the sweep and goes "low" at the midpoint of the sweep. The frequency of the sync waveform equals the specified sweep time.
- For *frequency sweeps* with *Marker On*, the Sync signal is a TTL "high" at the beginning of the sweep and goes "low" at the marker frequency.
- For a *triggered burst*, the Sync signal is a TTL "high" when the burst begins. The Sync signal is a TTL "low" at the end of the specified number of cycles (may not be the zero-crossing point if the waveform has an associated start phase). For an *infinite count burst*, the Sync signal is the same as for a continuous waveform.
- For an *externally-gated burst*, the Sync signal follows the external gate signal. However, note that the signal will not go to a TTL "low" until the end of the last cycle (may not be the zero-crossing point if the waveform has an associated start phase).
- *Front-Panel Operation*: Press Utility and select the **Sync** softkey again to toggle between "off" and "on".
- *Remote Interface Operation*:

  `OUTPut:SYNC {OFF|ON}` *Setting is stored in non-volatile memory*

3

Pulse Waveforms

A shown below, a pulse waveform consists of a *period*, a *pulse width*, a *rising edge*, and a *falling edge*.



Pulse Period

- Pulse period: 20 ns to 2000 seconds. *The default is 1 ms*.
- The specified period must be greater than the sum of the *pulse width* and the *edge time* as shown below. The function generator will adjust the pulse width and edge time as needed to accommodate the specified period.

$$\text{Period} \geq \text{Pulse Width} + (1.6 \times \text{Edge Time})$$

- *Function Limitations*: If you change to a function whose minimum period is greater than that for a pulse waveform, the period is adjusted to the minimum value allowed for the new function. For example, if you output a pulse waveform with a period of 50 ns and then change to the ramp function, the function generator will automatically adjust the period to 1 µs (the lower limit for ramps).

- *Front-Panel Operation*: After selecting the pulse function, press the **Freq** softkey again to toggle to the **Period** softkey. Then use the knob or numeric keypad to enter the desired pulse period.
- *Remote Interface Operation*:

```
PULSe:PERiod {<seconds>|MINimum|MAXimum}
```


Pulse Width

The pulse width represents the time from the 50% threshold of the rising edge of the pulse to the 50% threshold of the next falling edge.

- Pulse width: 8 ns to 2000 seconds (see restrictions below). *The default pulse width is* 100 µs.
- The specified pulse width must be less than the difference between the *period* and the *edge time* as shown below. The function generator will automatically adjust the pulse width as needed to accommodate the specified period.

$$\text{Pulse Width} \leq \text{Period} - (1.6 \times \text{Edge Time})$$

- The pulse width must also be greater than the total time of one edge as shown below.

$$\text{Pulse Width} \geq 1.6 \times \text{Edge Time}$$

- *Front-Panel Operation*: After selecting the pulse function, press the **Pulse Width** softkey. Then use the knob or numeric keypad to enter the desired pulse width.
- *Remote Interface Operation*:

```
PULSe:WIDTh {<seconds>|MINimum|MAXimum}
```

Edge Time

The edge time represents the time from the 10% threshold to the 90% threshold of both the rising and falling edges.

- Edge time: 5 ns to 1 ms (see restrictions below). *The default edge time is 5 ns*.
- The specified edge time must fit within the specified pulse width as shown below. The function generator will adjust the edge time as needed to accommodate the specified pulse width.

  $$\text{Edge Time} \leq 0.625 \times \text{Pulse Width}$$

- *Front-Panel Operation*: After selecting the pulse function, press the **Edge Time** softkey. Then use the knob or numeric keypad to enter the desired edge time.
- *Remote Interface Operation*:

```
PULSe:TRANsition {<seconds>|MINimum|MAXimum}
```

Amplitude Modulation (AM)

A modulated waveform consists of a *carrier waveform* and a *modulating waveform*. In AM, the amplitude of the carrier is varied by the instantaneous voltage of the modulating waveform. The function generator will accept an internal or external modulation source.

For more information on the fundamentals of Amplitude Modulation, refer to chapter 7, "Tutorial".

To Select AM Modulation

- The function generator will allow only one modulation mode to be enabled at a time. For example, you cannot enable AM and FM at the same time. When you enable AM, the previous modulation mode is turned off.
- The function generator will not allow AM to be enabled at the same time that sweep or burst is enabled. When you enable AM, the sweep or burst mode is turned off.
- *Front-Panel Operation*: You must enable AM *before* setting up any of the other modulation parameters. Press Mod and then select "AM" using the **Type** softkey. The AM waveform is output using the present settings for the carrier frequency, modulating frequency, output amplitude, and offset voltage.
- *Remote Interface Operation*: To avoid multiple waveform changes, enable AM *after* you have set up the other modulation parameters.

```
AM:STATe {OFF|ON}
```

Carrier Waveform Shape

- AM carrier shape: **Sine**, Square, Ramp, or Arbitrary waveform. *The default is Sine.* You *cannot* use pulse, noise, or dc as the carrier waveform.
- *Front-Panel Operation*: Press any of the front-panel function keys except Pulse or Noise. For arbitrary waveforms, press Arb and then choose the **Select Wform** softkey to select the active waveform.
- *Remote Interface Operation*:

```
FUNCtion:SHAPe {SINusoid|SQUare|RAMP|USER}
```

  You can also use the APPLy command to select the function, frequency, amplitude, and offset with a single command.

Carrier Frequency

The maximum carrier frequency depends on the function selected as shown below. *The default is 1 kHz for all functions*.

Function	Minimum Frequency	Maximum Frequency
Sine	1 µHz	80 MHz
Square	1 µHz	80 MHz
Ramp	1 µHz	1 MHz
Arbs	1 µHz	25 MHz

- *Front-Panel Operation*: To set the carrier frequency, press the **Freq** softkey for the selected function. Then use the knob or numeric keypad to enter the desired frequency.
- *Remote Interface Operation*:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

  You can also use the APPLy command to select the function, frequency, amplitude, and offset with a single command.

Modulating Waveform Shape

The function generator will accept an internal or external modulation source for AM.

- Modulating waveform shape (*internal source*): **Sine**, Square, Ramp, Negative Ramp, Triangle, Noise, or Arb waveform. *The default is Sine.*
  - Square has 50% duty cycle. 
  - Ramp has 100% symmetry. 
  - Triangle has 50% symmetry. 
  - Negative ramp has 0% symmetry. 
- You can use noise as the modulating waveshape, but you *cannot* use noise, pulse, or dc as the carrier waveform.
- If you select an arbitrary waveform as the *modulating* waveshape, the waveform is automatically limited to 8K points. Extra waveform points are removed using decimation.
- *Front-Panel Operation*: After enabling AM, press the **Shape** softkey.
- *Remote Interface Operation*:

```
AM:INTernal:FUNCtion {SINusoid|SQUare|RAMP|NRAMp|
   TRIangle|NOISe|USER}
```

Modulating Waveform Frequency

The function generator will accept an internal or external modulation source for AM.

- Modulating frequency (*internal source*): 2 mHz to 20 kHz. *The default is 100 Hz*.
- *Front-Panel Operation*: After enabling AM, press the **AM Freq** softkey.
- *Remote Interface Operation*:

```
AM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
```

Modulation Depth

The *modulation depth* is expressed as a percentage and represents the extent of the amplitude variation. At 0% depth, the output amplitude is half of the selected value. At 100% depth, the output amplitude equals the selected value.

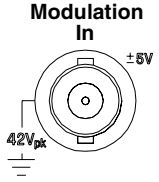
- Modulation depth: 0% to 120%. *The default is 100%*.
- Note that even at greater than 100% depth, the function generator will not exceed ±5V peak on the output (into a 50Ω load).
- If you select the *External* modulating source, the carrier waveform is modulated with an external waveform. The modulation depth is controlled by the ±5V signal level present on the rear-panel *Modulation In* connector. For example, if you have set the modulation depth to 100%, then when the modulating signal is at +5 volts, the output will be at the *maximum* amplitude. When the modulating signal is at -5 volts, then the output will be at the *minimum* amplitude.
- *Front-Panel Operation*: After enabling AM, press the **AM Depth** softkey. Then use the knob or numeric keypad to enter the depth.
- *Remote Interface Operation*:

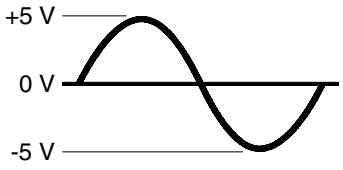
```
AM:DEPTh {<depth in percent>|MINimum|MAXimum}
```

Modulating Source

The function generator will accept an internal or external modulation source for AM.

- Modulating source: **Internal** or External. *The default is Internal*.
- If you select the *External* source, the carrier waveform is modulated with an external waveform. The modulation depth is controlled by the ±5V signal level present on the rear-panel *Modulation In* connector. For example, if you have set the modulation depth to 100%, then when the modulating signal is at +5 volts, the output will be at the *maximum* amplitude. When the modulating signal is at -5 volts, then the output will be at the *minimum* amplitude.





- *Front-Panel Operation*: After enabling AM, press the **Source** softkey.
- *Remote Interface Operation*:

```
AM:SOURce {INTernal|EXTernal}
```

Frequency Modulation (FM)

A modulated waveform consists of a *carrier waveform* and a *modulating waveform*. In FM, the frequency of the carrier is varied by the instantaneous voltage of the modulating waveform.

For more information on the fundamentals of Frequency Modulation, refer to chapter 7, "Tutorial".

To Select FM Modulation

- The function generator will allow only one modulation mode to be enabled at a time. For example, you cannot enable FM and AM at the same time. When you enable FM, the previous modulation mode is turned off.
- The function generator will not allow FM to be enabled at the same time that sweep or burst is enabled. When you enable FM, the sweep or burst mode is turned off.
- *Front-Panel Operation*: You must enable FM *before* setting up any of the other modulation parameters. Press (Mod) and then select "FM" using the **Type** softkey. The FM waveform is output using the present settings for the carrier frequency, modulating frequency, output amplitude, and offset voltage.
- *Remote Interface Operation*: To avoid multiple waveform changes, enable FM *after* you have set up the other modulation parameters.

```
FM:STATe {OFF|ON}
```

Carrier Waveform Shape

- FM carrier shape: **Sine**, Square, Ramp, or Arbitrary waveform. *The default is Sine.* You *cannot* use pulse, noise, or dc as the carrier waveform.
- *Front-Panel Operation*: Press any of the front-panel function keys except Pulse or Noise . For arbitrary waveforms, press Arb and then choose the **Select Wform** softkey to select the active waveform.
- *Remote Interface Operation*:

```
FUNCtion:SHAPe {SINusoid|SQUare|RAMP|USER}
```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Carrier Frequency

The maximum carrier frequency depends on the function selected as shown below. *The default is 1 kHz for all functions*.

Function	Minimum Frequency	Maximum Frequency
Sine	5 Hz	80 MHz
Square	5 Hz	80 MHz
Ramp	5 Hz	1 MHz
Arbs	5 Hz	25 MHz

- The *carrier frequency* must always be greater than or equal to the frequency deviation. If you attempt to set the deviation to a value greater than the carrier frequency (with FM enabled), the function generator will automatically adjust the deviation to the maximum value allowed with the present carrier frequency.
- The sum of the *carrier frequency* and deviation must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (80.1 MHz for sine and square, 1.1 MHz for ramp, and 25.1 MHz for arbitrary waveforms). If you attempt to set the deviation to a value that is not valid, the function generator will automatically adjust it to the maximum value allowed with the present carrier frequency.
- *Front-Panel Operation*: To set the carrier frequency, press the **Freq** softkey for the selected function. Then use the knob or numeric keypad to enter the desired frequency.
- *Remote Interface Operation*:

  ```
  FREQuency {<frequency>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Modulating Waveform Shape

The function generator will accept an internal or external modulation source for FM.

- Modulating waveform shape (*internal source*): **Sine**, Square, Ramp, Negative Ramp, Triangle, Noise, or Arb waveform. *The default is Sine.*
  - Square has 50% duty cycle. 
  - Ramp has 100% symmetry. 
  - Triangle has 50% symmetry. 
  - Negative ramp has 0% symmetry. 
- You can use noise as the modulating waveshape, but you *cannot* use noise, pulse, or dc as the carrier waveform.
- If you select an arbitrary waveform as the *modulating* waveshape, the waveform is automatically limited to 8K points. Extra waveform points are removed using decimation.
- *Front-Panel Operation*: After enabling FM, press the **Shape** softkey.
- *Remote Interface Operation*:

```
FM:INTernal:FUNCtion {SINusoid|SQUare|RAMP|NRAMp|
   TRIangle|NOISe|USER}
```

Modulating Waveform Frequency

The function generator will accept an internal or external modulation source for FM.

- Modulating frequency (*internal source*): 2 mHz to 20 kHz. *The default is 10 Hz*.
- *Front-Panel Operation*: After enabling FM, press the **FM Freq** softkey.
- *Remote Interface Operation*:

```
FM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}
```

Peak Frequency Deviation

The *peak frequency deviation* represents the variation in frequency of the modulating waveform from the carrier frequency.

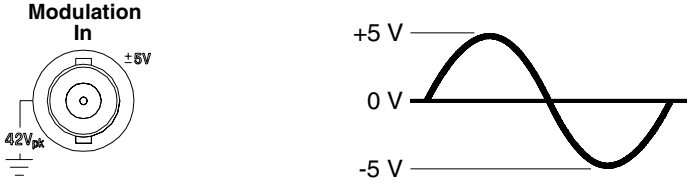
- Peak frequency deviation: 5 Hz to 40.05 MHz (limited to 550 kHz for ramps and 12.55 MHz for arbitrary waveforms). *The default is 100 Hz*.
- The *carrier frequency* must always be greater than or equal to the deviation. If you attempt to set the deviation to a value greater than the carrier frequency (with FM enabled), the function generator will limit the deviation to the maximum value allowed with the present carrier frequency.
- The sum of the *carrier frequency* and deviation must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (80.1 MHz for sine and square, 1.1 MHz for ramp, and 25.1 MHz for arbitrary waveforms). If you attempt to set the deviation to a value that is not valid, the function generator will limit it to the maximum value allowed with the present carrier frequency.
- *Front-Panel Operation*: After enabling FM, press the **Freq Dev** softkey. Then use the knob or numeric keypad to enter the desired deviation.
- *Remote Interface Operation*:

```
FM:DEViation {<peak deviation in Hz>|MINimum|MAXimum}
```

Modulating Source

The function generator will accept an internal or external modulation source for FM.

- Modulating source: **Internal** or External. *The default is Internal*.
- If you select the *External* source, the carrier waveform is modulated with an external waveform. The frequency deviation is controlled by the ±5V signal level present on the rear-panel *Modulation In* connector. For example, if you have set the deviation to 100 kHz, then a +5V signal level corresponds to a 100 kHz increase in frequency. Lower external signal levels produce less deviation and negative signal levels reduce the frequency below the carrier frequency.



- *Front-Panel Operation*: After enabling FM, press the **Source** softkey.
- *Remote Interface Operation*:

```
FM:SOURce {INTernal|EXTernal}
```

3

Frequency-Shift Keying (FSK) Modulation

You can configure the function generator to "shift" its output frequency between two preset values using FSK modulation. The rate at which the output shifts between the two frequencies (called the "carrier frequency" and the "hop frequency") is determined by the internal rate generator or the signal level on the rear-panel *Trig In* connector.

For more information on the fundamentals of FSK Modulation, refer to chapter 7, "Tutorial".

To Select FSK Modulation

- The function generator will allow only one modulation mode to be enabled at a time. For example, you cannot enable FSK and AM at the same time. When you enable FSK, the previous modulation mode is turned off.
- The function generator will not allow FSK to be enabled at the same time that sweep or burst is enabled. When you enable FSK, the sweep or burst mode is turned off.
- *Front-Panel Operation*: You must enable FSK *before* setting up any of the other modulation parameters. Press Mod and then select "FSK" using the **Type** softkey. The FSK waveform is output using the present settings for the carrier frequency, output amplitude, and offset voltage.
- *Remote Interface Operation*: To avoid multiple waveform changes, enable FSK *after* you have set up the other modulation parameters.

```
FSKey:STATe {OFF|ON}
```

Carrier Waveform Shape

- FSK carrier shape: **Sine**, Square, Ramp, or Arbitrary waveform. *The default is Sine.* You *cannot* use pulse, noise, or dc as the carrier waveform.
- *Front-Panel Operation*: Press any of the front-panel function keys except Pulse or Noise . For arbitrary waveforms, press Arb and then choose the **Select Wform** softkey to select the active waveform.
- *Remote Interface Operation*:

```
FUNCtion:SHAPe {SINusoid|SQUare|RAMP|USER}
```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

FSK Carrier Frequency

The maximum carrier frequency depends on the function selected as shown below. *The default is 1 kHz for all functions*.

Function	Minimum Frequency	Maximum Frequency
Sine	1 µHz	80 MHz
Square	1 µHz	80 MHz
Ramp	1 µHz	1 MHz
Arbs	1 µHz	25 MHz

- When the *External* source is selected, the output frequency is determined by the signal level on the rear-panel *Trig In* connector. When a logic *low* level is present, the *carrier* frequency is output. When a logic *high* level is present, the *hop* frequency is output.
- *Front-Panel Operation*: To set the carrier frequency, press the **Freq** softkey for the selected function. Then use the knob or numeric keypad to enter the desired frequency.
- *Remote Interface Operation*:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

FSK "Hop" Frequency

The maximum alternate (or "hop") frequency depends on the function selected as shown below. *The default is 100 Hz for all functions*.

Function	Minimum Frequency	Maximum Frequency
Sine	1 µHz	80 MHz
Square	1 µHz	80 MHz
Ramp	1 µHz	1 MHz
Arbs	1 µHz	25 MHz

- The internal modulating waveform is a *square wave* with a 50% duty cycle.
- When the *External* source is selected, the output frequency is determined by the signal level on the rear-panel *Trig In* connector. When a logic *low* level is present, the *carrier* frequency is output. When a logic *high* level is present, the *hop* frequency is output.
- *Front-Panel Operation*: To set the "hop" frequency, press the **Hop Freq** softkey. Then use the knob or numeric keypad to enter the desired frequency.
- *Remote Interface Operation*:

```
FSKey:FREQuency {<frequency>|MINimum|MAXimum}
```

FSK Rate

The *FSK rate* is the rate at which the output frequency "shifts" between the carrier frequency and the hop frequency when you select the internal FSK source.

- FSK rate (*internal source*): 2 mHz to 100 kHz. *The default is 10 Hz*.
- The FSK rate is ignored when the external FSK source is selected.
- *Front-Panel Operation*: To set the FSK rate, press the **FSK Rate** softkey. Then use the knob or numeric keypad to enter the desired rate.
- *Remote Interface Operation*:

```
FSKey:INTernal:RATE {<rate in Hz>|MINimum|MAXimum}
```

FSK Source

- FSK source: **Internal** or External. *The default is Internal*.
- When the *Internal* source is selected, the rate at which the output frequency "shifts" between the carrier frequency and hop frequency is determined by the *FSK rate* specified.
- When the *External* source is selected, the output frequency is determined by the signal level on the rear-panel *Trig In* connector. When a logic *low* level is present, the *carrier* frequency is output. When a logic *high* level is present, the *hop* frequency is output.
- The maximum external FSK rate is 1 MHz.
- Note that the connector used for externally-controlled FSK waveforms (*Trig In*) is not the same connector that is used for externally-modulated AM and FM waveforms (*Modulation In*). When used for FSK, the *Trig In* connector *does not* have adjustable edge polarity.
- *Front-Panel Operation*: After enabling FSK, press the **Source** softkey.
- *Remote Interface Operation*:

```
FSKey:SOURce {INTernal|EXTernal}
```

3

Frequency Sweep

In the frequency sweep mode, the function generator "steps" from the *start frequency* to the *stop frequency* at a *sweep rate* which you specify. You can sweep up or down in frequency, and with either linear or logarithmic spacing. You can also configure the function generator to output a single sweep (one pass from start frequency to stop frequency) by applying an external or manual trigger. The function generator can produce a frequency sweep for sine, square, ramp, or arbitrary waveforms (pulse, noise, and dc are not allowed).

For more information on the fundamentals of a sweep, refer to chapter 7, "Tutorial".

To Select Sweep

- The function generator will not allow the sweep mode to be enabled at the same time that burst or any modulation mode is enabled. When you enable sweep, the burst or modulation mode is turned off.
- *Front-Panel Operation*: You must enable sweep *before* setting up any of the other sweep parameters. Press Sweep to output a sweep using the present settings for frequency, output amplitude, and offset.
- *Remote Interface Operation*: To avoid multiple waveform changes, enable the sweep mode *after* you have set up the other parameters.

```
SWEep:STATe {OFF|ON}
```

Start Frequency and Stop Frequency

The *start frequency* and *stop frequency* set the upper and lower frequency bounds for the sweep. The function generator begins at the start frequency, sweeps to the stop frequency, and then resets back to the start frequency.

- Start and Stop frequencies: 1 µHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). The sweep is phase continuous over the full frequency range. *The default start frequency is 100 Hz. The default stop frequency is 1 kHz*.
- To sweep ***up*** in frequency, set the start frequency **<** stop frequency. To sweep ***down*** in frequency, set the start frequency **>** stop frequency.
- For sweeps with *Marker Off*, the Sync signal is a square waveform with a 50% duty cycle. The Sync signal is a TTL "high" at the beginning of the sweep and goes "low" at the midpoint of the sweep. The frequency of the sync waveform is equal to the specified sweep time. The signal is output from the front-panel *Sync* connector.
- For sweeps with *Marker On*, the Sync signal is a TTL "high" at the beginning of the sweep and goes "low" at the marker frequency. The signal is output from the front-panel *Sync* connector.
- *Front-Panel Operation*: After enabling sweeps, press the **Start** or **Stop** softkey. Then use the knob or numeric keypad to enter the desired frequency.
- *Remote Interface Operation*:

```
FREQuency:STARt {<frequency>|MINimum|MAXimum}
FREQuency:STOP {<frequency>|MINimum|MAXimum}
```

Center Frequency and Frequency Span

If desired, you can set the frequency boundaries of the sweep using a *center frequency* and *frequency span*. These parameters are similar to the start frequency and stop frequency (*see the previous page*) and are included to give you added flexibility.

- Center frequency: 1 μHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). *The default is 550 Hz*.
- Frequency span: 0 Hz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). *The default is 900 Hz*.
- To sweep ***up*** in frequency, set a *positive* frequency span.
  To sweep ***down*** in frequency, set a *negative* frequency span.
- For sweeps with *Marker Off*, the Sync signal is a square waveform with a 50% duty cycle. The Sync signal is a TTL "high" at the beginning of the sweep and goes "low" at the midpoint of the sweep. The frequency of the sync waveform is equal to the specified sweep time. The signal is output from the front-panel *Sync* connector.
- For sweeps with *Marker On*, the Sync signal is a TTL "high" at the beginning of the sweep and goes "low" at the marker frequency. The signal is output from the front-panel *Sync* connector.
- *Front-Panel Operation*: After enabling sweeps, press the **Start** or **Stop** softkey again to toggle to the **Center** or **Span** softkey. Then use the knob or numeric keypad to enter the desired values.
- *Remote Interface Operation*:

```
FREQuency:CENTer {<frequency>|MINimum|MAXimum}
FREQuency:SPAN {<frequency>|MINimum|MAXimum}
```

Sweep Mode

You can sweep with either linear or logarithmic spacing. For a *linear* sweep, the function generator varies the output frequency in a linear fashion during the sweep. For a *logarithmic* sweep, the function generator varies the output frequency in a logarithmic fashion.

- Sweep mode: Linear or Logarithmic. *The default is Linear*.
- *Front-Panel Operation*: After enabling sweeps, press the **Linear** softkey again to toggle between the linear or log mode.
- *Remote Interface Operation*:

```
SWEep:SPACing {LINear|LOGarithmic}
```


Sweep Time

The *sweep time* specifies the number of seconds required to sweep from the start frequency to the stop frequency. The number of discrete frequency points in the sweep is automatically calculated by the function generator and is based on the sweep time you select.

- Sweep time: 1 ms to 500 seconds. *The default is 1 second*.
- *Front-Panel Operation*: After enabling sweeps, press the **Sweep Time** softkey. Then use the knob or numeric keypad to enter the desired sweep time.
- *Remote Interface Operation*:

```
SWEep:TIME {<seconds>|MINimum|MAXimum}
```

Marker Frequency

If desired, you can set the frequency at which the signal on the front-panel *Sync* connector goes to a logic low during the sweep. The Sync signal always goes from low to high at the beginning of the sweep.

- Marker frequency: 1 µHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). *The default is 500 Hz*.
- When the sweep mode is enabled, the marker frequency *must* be between the specified start frequency and stop frequency. If you attempt to set the marker frequency to a frequency not in this range, the function generator will automatically set the marker frequency equal to the start frequency or stop frequency (whichever is closer).
- The Sync signal setting is overridden by the setting of the Marker frequency used with the sweep mode (*see page 62*). Therefore, when the marker frequency is enabled (and the sweep mode is also enabled), the Sync signal setting is ignored.
- *Front-Panel Operation*: After enabling sweeps, press the **Marker** softkey. Then use the knob or numeric keypad to enter the desired marker frequency.
- *Remote Interface Operation*:

```
MARKer:FREQuency {<frequency>|MINimum|MAXimum}
```

Sweep Trigger Source

In the sweep mode, the function generator outputs a single sweep when a trigger signal is received. After one sweep from the start frequency to the stop frequency, the function generator waits for the next trigger while outputting the start frequency.

- Sweep trigger source: **Internal**, External, or Manual. *The default is Internal*.
- When the *Internal* (immediate) source is selected, the function generator outputs a continuous sweep at a rate determined by the *sweep time* specified.
- When the *External* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Trig In* connector. The function generator initiates one sweep each time *Trig In* receives a TTL pulse with the specified polarity.
- The trigger period must be greater than or equal to the specified sweep time ***plus*** 1 ms.
- When the *Manual* or *External* source is selected, the function generator outputs one sweep each time the front-panel (Trigger) key is pressed.
- *Front-Panel Operation*: Press the **Trigger Setup** softkey and then select the desired source by pressing the **Source** softkey.

  To specify whether the function generator triggers on the rising or falling edge of the *Trig In* connector, press the **Trigger Setup** softkey. Then select the desired edge by pressing the **Slope** softkey.
- *Remote Interface Operation*:

  ```
  TRIGger:SOURce {IMMediate|EXTernal|BUS}
  ```

  Use the following command to specify whether the function generator triggers on the rising or falling edge of the signal on the *Trig In* connector.

  ```
  TRIGger:SLOPe {POSitive|NEGative}
  ```

  See "Triggering" on page 98 for more information.

Trigger Out Signal

A "trigger out" signal is provided on the rear-panel *Trig Out* connector (*used with sweep and burst* only). When enabled, a TTL-compatible square waveform with either a rising edge (default) or falling edge is output from the *Trig Out* connector at the beginning of the sweep.

- When the *Internal* (immediate) trigger source is selected, the function generator outputs a square waveform with a 50% duty cycle from the *Trig Out* connector at the beginning of the sweep. The frequency of the waveform is equal to the specified *sweep time*.
- When the *External* trigger source is selected, the function generator automatically disables the "trigger out" signal. The *Trig Out* connector cannot be used for both operations at the same time (an externally-triggered waveform uses the same connector to trigger the sweep).
- When the *Manual* trigger source is selected, the function generator outputs a pulse (>1 µs pulse width) from the *Trig Out* connector at the beginning of each sweep or burst.
- *Front-Panel Operation*: After enabling sweeps, press the **Trigger Setup** softkey. Then select the desired edge by pressing the **Trig Out** softkey.
- *Remote Interface Operation*:

```
OUTPut:TRIGger:SLOPe {POSitive|NEGative}
OUTPut:TRIGger {OFF|ON}
```

Burst Mode

You can configure the function generator to output a waveform with a specified number of cycles, called a *burst*. The function generator can produce a burst using sine, square, ramp, pulse, or arbitrary waveforms (noise is allowed only in the gated burst mode and dc is not allowed).

For more information on the fundamentals of the burst mode, refer to chapter 7, "Tutorial".

To Select Burst

- The function generator will not allow burst to be enabled at the same time that sweep or any modulation mode is enabled. When you enable burst, the sweep or modulation mode is turned off.
- *Front-Panel Operation*: You must enable burst *before* setting up any of the other burst parameters. Press Burst to output a burst using the present settings for frequency, output amplitude, and offset voltage.
- *Remote Interface Operation*: To avoid multiple waveform changes, enable the burst mode *after* you have set up the other parameters.

```
BURSt:STATe {OFF|ON}
```

Burst Type

You can use burst in one of two modes as described below. The function generator enables one burst mode at a time based on the *trigger source* and *burst source* that you select (see the table below).

- *Triggered Burst Mode*: In this mode (default), the function generator outputs a waveform with a specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles have been output, the function generator stops and waits for the next trigger. You can configure the function generator to use an internal trigger to initiate the burst or you can provide an external trigger by pressing the front-panel Trigger key, applying a trigger signal to the rear-panel *Trig In* connector, or sending a software trigger command from the remote interface.
- *External Gated Burst Mode*: In this mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Trig In* connector. When the gate signal is *true*, the function generator outputs a continuous waveform. When the gate signal goes *false*, the current waveform cycle is completed and then the function generator stops while remaining at the voltage level corresponding to the starting burst phase of the selected waveform. For a noise waveform, the output stops immediately when the gate signal goes false.

	Burst Mode (BURS:MODE)	**Burst Count** (BURS:NCYC)	**Burst Period** (BURS:INT:PER)	**Burst Phase** (BURS:PHAS)	**Trigger Source** (TRIG:SOUR)
Triggered Burst Mode: Internal Trigger	TRIGgered	Available	Available	Available	IMMediate
Triggered Burst Mode: External Trigger	TRIGgered	Available	Not Used	Available	EXTernal, BUS
Gated Burst Mode: External Trigger	GATed	Not Used	Not Used	Available	Not Used

- When the *gated* mode is selected, the burst count, burst period, and trigger source are ignored (these parameters are used for the triggered burst mode only). If a manual trigger is received, it is ignored and no error will be generated.
- When the *gated* mode is selected, you can also select the polarity of the signal on the rear-panel *Trig In* connector.
- *Front-Panel Operation*: After enabling bursts, press the **N Cycle** (triggered) or **Gated** softkey.

  To select the polarity of the external gate signal on the *Trig In* connector, press the **Polarity** softkey. The default polarity is POS (true-high logic).
- *Remote Interface Operation*:

```
BURSt:MODE {TRIGgered|GATed}
```

  Use the following command to select the polarity of the external gate signal on the *Trig In* connector. The default is NORM (true-high logic).

```
BURSt:GATE:POLarity {NORMal|INVerted}
```

Waveform Frequency

The *waveform frequency* defines the repetition rate of the burst waveform in the triggered and external gated modes. In the *triggered mode*, the number of cycles specified by the *burst count* is output at the waveform frequency. In the *external gated mode*, the waveform frequency is output when the external gate signal is true.

Keep in mind that the waveform frequency is different than the "burst period" which specifies the interval between bursts (triggered mode only).

- Waveform frequency: 2 mHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). *The default waveform frequency is 1 kHz*. You can select a sine, square, ramp, pulse, or arbitrary waveform (noise is allowed only in the gated burst mode and dc is not allowed).
- For sine and square waveforms, frequencies above 25 MHz are allowed only with an "infinite" burst count.
- *Front-Panel Operation*: To set the waveform frequency, press the **Freq** softkey for the selected function. Then use the knob or numeric keypad to enter the desired frequency.
- *Remote Interface Operation*:

  ```
  FREQuency {<frequency>|MINimum|MAXimum}
  ```

  You can also use the `APPLy` command to select the function, frequency, amplitude, and offset with a single command.

Burst Count

The burst count defines the number of cycles to be output per burst. *Used in the triggered burst mode only (internal or external source)*.

- Burst count: 1 to 1,000,000 cycles, in 1 cycle increments. You can also select an infinite burst count. *The default is 1 cycle*.
- When the *Internal* trigger source is selected, the specified number of cycles is output continuously at a rate determined by the *burst period* setting. The burst period defines the interval between bursts.
- When the *Internal* trigger source is selected, the burst count must be less than the product of the burst period and the waveform frequency as shown below.

  Burst Count < Burst Period X Waveform Frequency

- The function generator will automatically increase the burst period up to its maximum value to accommodate the specified burst count (but the waveform frequency *will not* be changed).
- When the *gated* burst mode is selected, the burst count is ignored. However, if you change the burst count from the remote interface while in the gated mode, the function generator remembers the new count and will use it when the triggered mode is selected.
- *Front-Panel Operation*: To set the burst count, press the **#Cycles** softkey and then use the knob or numeric keypad to enter the count. To select an infinite count burst instead, press the **#Cycles** softkey again to toggle to the **Infinite** softkey (press Trigger to stop the waveform).
- *Remote Interface Operation*:

  `BURSt:NCYCles {<# cycles>|INFinity|MINimum|MAXimum}`

Burst Period

The *burst period* defines time from the start of one burst to the start of the next burst. *Used in the internal triggered burst mode only.*

Keep in mind that burst period is different than the "waveform frequency" which specifies the frequency of the bursted signal.

- Burst period: 1 µs to 500 seconds. *The default is 10 ms*.
- The burst period setting is used only when *Internal* triggering is enabled. The burst period is ignored when manual or external triggering is enabled (or when the *gated* burst mode is selected).
- It *is not* possible to specify a burst period which is too short for the function generator to output with the specified burst count and frequency (*see below*). If the burst period is too short, the function generator will automatically adjust it as needed to continuously re-trigger the burst.

$$\text{Burst Period} > \frac{\text{Burst Count}}{\text{Waveform Frequency}} + 200\ \text{ns}$$

- *Front-Panel Operation*: To set the burst period, press the **Burst Period** softkey and then use the knob or numeric keypad to enter the period.
- *Remote Interface Operation*:

```
BURSt:INTernal:PERiod {<seconds>|MINimum|MAXimum}
```

Burst Phase

The *burst phase* defines the starting phase of the burst.

- Burst phase: -360 degrees to +360 degrees. *The default is 0 degrees.*
- From the remote interface, you can set the starting phase in degrees or radians using the `UNIT:ANGL` command (see page 192).
- From the front panel, the starting phase is always displayed in degrees (radians are not available). If you set the starting phase in radians from the remote interface and then return to front-panel operation, you will see that the function generator converts the phase to degrees.
- For sine, square, and ramp waveforms, 0 degrees is the point at which the waveform crosses zero volts (or the dc offset value) in a positive-going direction. For arbitrary waveforms, 0 degrees is the first waveform point downloaded to memory. The burst phase has no effect on pulse or noise waveforms.
- The burst phase is also used in the *gated* burst mode. When the gate signal goes *false*, the current waveform cycle is completed and then the function generator stops. The output will remain at the voltage level corresponding to the starting burst phase.
- *Front-Panel Operation*: To set the burst phase, press the **Start Phase** softkey and then use the knob or numeric keypad to enter the desired phase in degrees.
- *Remote Interface Operation*:

```
BURSt:PHASe {<angle>|MINimum|MAXimum}
```

Burst Trigger Source

In the *triggered* burst mode, the function generator outputs a burst with the specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles have been output, the function generator stops and waits for the next trigger. *At power-on, the internally-triggered burst mode is enabled.*

- Burst trigger source: **Internal**, External, or Manual. *The default is Internal*.
- When the *Internal* (immediate) source is selected, the frequency at which the burst is generated is determined by the *burst period*.
- When the *External* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Trig In* connector. The function generator outputs the specified number of cycles each time *Trig In* receives a TTL pulse with the specified polarity. External trigger signals that occur during a burst are ignored.
- When the *Manual* source is selected, the function generator outputs one burst each time the front-panel (Trigger) key is pressed.
- When the *External* or *Manual* trigger source is selected, the *burst count* and *burst phase* remain in effect but the *burst period* is ignored.
- You can insert a time delay between the receipt of the trigger and the start of the burst waveform (used in *triggered* burst mode only).
- *Front-Panel Operation*: Press the **Trigger Setup** softkey and then select the desired source by pressing the **Source** softkey.

  To insert a trigger delay, press the **Delay** softkey (used in the *triggered* burst mode only).

  To specify whether the function generator triggers on the rising or falling edge of the signal on the *Trig In* connector, press the **Trigger Setup** softkey. Then select the desired edge by pressing the **Slope** softkey.

- *Remote Interface Operation*:

  ```
  TRIGger:SOURce {IMMediate|EXTernal|BUS}
  ```

  Use the following command to insert a trigger delay.

  ```
  TRIGger:DELay {<seconds>|MINimum|MAXimum}
  ```

  Use the following command to specify whether the function generator triggers on the rising or falling edge of the *Trig In* connector.

  ```
  TRIGger:SLOPe {POSitive|NEGative}
  ```

  See "Triggering" on page 98 for more information on triggering.

Trigger Out Signal

A "trigger out" signal is provided on the rear-panel *Trig Out* connector (*used with burst and sweep only*). When enabled, a TTL-compatible square waveform with either a rising edge (default) or falling edge is output from the *Trig Out* connector at the beginning of the burst.

- When the *Internal* (immediate) trigger source is selected the function generator outputs a square waveform with a 50% duty cycle from the *Trig Out* connector at the beginning of the burst. The frequency of the waveform is equal to the specified *burst period*.
- When the *External* trigger source is selected, the function generator automatically disables the "trigger out" signal. The *Trig Out* connector cannot be used for both operations at the same time (an externally-triggered waveform uses the same connector to trigger the burst).
- When the *Manual* trigger source is selected, the function generator outputs a pulse (>1 µs pulse width) from the *Trig Out* connector at the beginning of each burst.
- *Front-Panel Operation*: After enabling bursts, press the **Trigger Setup** softkey. Then select the desired edge by pressing the **Trig Out** softkey.
- *Remote Interface Operation*:

  ```
  OUTPut:TRIGger:SLOPe {POSitive|NEGative}
  OUTPut:TRIGger {OFF|ON}
  ```

Triggering

Applies to sweep and burst only. You can issue triggers for sweeps or bursts using *internal* triggering, *external* triggering, or *manual* triggering.

- Internal or "automatic" triggering is enabled when you turn on the function generator. In this mode, the function generator outputs continuously when the sweep or burst mode is selected.
- External triggering uses the rear-panel *Trig In* connector to control the sweep or burst. The function generator initiates one sweep or outputs one burst each time *Trig In* receives a TTL pulse. You can select whether the function generator triggers on the rising or falling edge of the external trigger signal.
- Manual triggering initiates one sweep or outputs one burst each time you press (Trigger) from the front-panel. Continue pressing this key to re-trigger the function generator.
- The (Trigger) key is disabled when in remote and when a function other than burst or sweep is currently selected.

Trigger Source Choices

Applies to sweep and burst only. You must specify the source from which the function generator will accept a trigger.

- Sweep trigger source: **Internal**, External, or Manual. *The default is Internal*.
- The function generator will accept a manual trigger, a hardware trigger from the rear-panel *Trig In* connector, or continuously output sweeps or bursts using an internal trigger. *At power-on, internal trigger is selected.*

- The trigger source setting is stored in *volatile* memory; the source is set to internal trigger (front panel) or immediate (remote interface) when power has been off or after a remote interface reset.
- *Front-Panel Operation*: After enabling sweeps or bursts, press the **Trigger Setup** softkey. Then select the desired source by pressing the **Source** softkey.
- *Remote Interface Operation*:

```
TRIGger:SOURce {IMMediate|EXTernal|BUS}
```

  The `APPLy` command automatically sets the source to *Immediate.*

Internal Triggering In the internal trigger mode, the function generator continuously outputs the sweep or burst (as specified by the *sweep time* or *burst period*). This is the power-on trigger source for both front-panel and remote interface use.

- *Front-Panel Operation*: Press the **Trigger Setup** softkey and then select the **Source Int** softkey.
- *Remote Interface Operation*:

```
TRIGger:SOURce IMMediate
```

Manual Triggering In the manual trigger mode (front panel only), you can manually trigger the function generator by pressing the front-panel (Trigger) key. The function generator initiates one sweep or outputs one burst for each time you press the key. The (Trigger) key is illuminated while the function generator is waiting for a manual trigger (the key is disabled when in remote).

External Triggering In the external trigger mode, the function generator will accept a hardware trigger applied to the rear-panel *Trig In* connector. The function generator initiates one sweep or outputs one burst each time *Trig In* receives a TTL pulse with the specified edge.

See also "Trigger Input Signal," on the following page.

- *Front-Panel Operation*: The external trigger mode is like the manual trigger mode except that you apply the trigger to the *Trig In* connector. To select the external source, press the **Trigger Setup** softkey and then select the **Source Ext** softkey.

  To specify whether the function generator triggers on the rising or falling edge, press the **Trigger Setup** softkey and then select the desired edge by pressing the **Slope** softkey.

- *Remote Interface Operation*:

  ```
  TRIGger:SOURce EXTernal
  ```

  Use the following command to specify whether the function generator triggers on the rising or falling edge.

  ```
  TRIGger:SLOPe {POSitive|NEGative}
  ```

Software (Bus) Triggering The bus trigger mode is available only from the remote interface. This mode is similar to the manual trigger mode from the front panel, but you trigger the function generator by sending a bus trigger command. The function generator initiates one sweep or outputs one burst each time a bus trigger command is received.

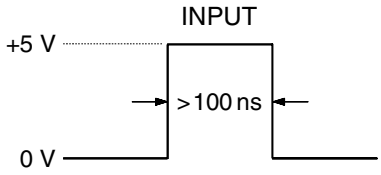
- To select the bus trigger source, send the following command.

  ```
  TRIGger:SOURce BUS
  ```

- To trigger the function generator from the remote interface (GPIB or RS-232) when the *Bus* source selected, send the `TRIG` or `*TRG` (trigger) command. The front-panel (Trigger) key is illuminated when the function generator is waiting for a bus trigger.

Trigger Input Signal





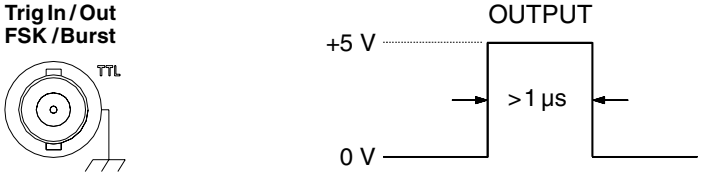
Rising edge shown.

This rear-panel connector is used in the following modes:

- *Triggered Sweep Mode*: To select the external source, press the **Trigger Setup** softkey and then select the **Source Ext** softkey or execute the `TRIG:SOUR EXT` command from the remote interface (sweep must be enabled). When the rising or falling edge (you specify which edge) of a TTL pulse is received on the *Trig In* connector, the function generator outputs a single sweep.
- *Externally-Modulated FSK Mode*: To enable the externally-modulated mode, press the **Source** softkey from the front-panel or execute the `FSK:SOUR EXT` command from the remote interface (FSK must be enabled). When a logic *low* level is present, the *carrier* frequency is output. When a logic *high* level is present, the *hop* frequency is output. The maximum external FSK rate is 1 MHz.
- *Triggered Burst Mode*: To select the external source, press the **Trigger Setup** softkey and then select the **Source Ext** softkey or execute the `TRIG:SOUR EXT` command from the remote interface (burst must be enabled). The function generator outputs a waveform with a specified number of cycles (*burst count*) each time a trigger is received from the specified trigger source.
- *External Gated Burst Mode*: To enable the gated mode, press the **Gated** softkey or execute the `BURS:MODE GAT` command from the remote interface (burst must be enabled). When the external gate signal is *true*, the function generator outputs a continuous waveform. When the external gate signal goes *false*, the current waveform cycle is completed and then the function generator stops while remaining at the voltage level corresponding to the starting burst phase. For noise, the output stops immediately when the gate signal goes false.

Trigger Output Signal

A "trigger out" signal is provided on the rear-panel *Trig Out* connector (*used with sweep and burst only*). When enabled, a TTL-compatible square waveform with either a rising edge (default) or falling edge is output from the rear-panel *Trig Out* connector at the beginning of the sweep or burst.



Rising edge shown.

- When the *Internal* (immediate) trigger source is selected, the function generator outputs a square waveform with a 50% duty cycle from the *Trig Out* connector at the beginning of the sweep or burst. The period of the waveform is equal to the specified *sweep time* or *burst period*.
- When the *External* trigger source is selected, the function generator automatically disables the "trigger out" signal. The *Trig Out* connector cannot be used for both operations at the same time (an externally-triggered waveform uses the same connector to trigger the sweep or burst).
- When the *Bus* (software) trigger source is selected, the function generator outputs a pulse (>1 µs pulse width) from the *Trig Out* connector at the beginning of each sweep or burst.
- *Front-Panel Operation*: After enabling sweeps or burst, press the **Trigger Setup** softkey. Then select the desired edge by pressing the **Trig Out** softkey.
- *Remote Interface Operation*:

```
OUTPut:TRIGger:SLOPe {POSitive|NEGative}
OUTPut:TRIGger {OFF|ON}
```

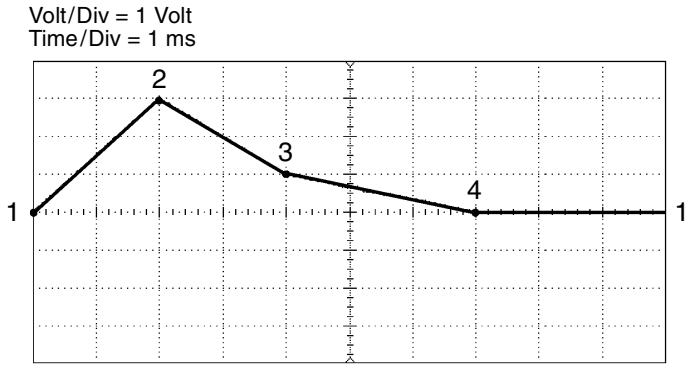
Arbitrary Waveforms

There are five built-in arbitrary waveforms stored in non-volatile memory. You can also store up to four user-defined waveforms in non-volatile memory in addition to one in volatile memory. Each waveform can contain between 1 (a dc voltage) and 65,536 (64K) data points.

Refer to chapter 7, "Tutorial", for more information on the internal operation of downloading and outputting an arbitrary waveform.

To Create and Store an Arbitrary Waveform

This section gives an example which shows you how to create and store an arbitrary waveform from the front panel. To download an arbitrary waveform from the remote interface, see "Arbitrary Waveform Commands" starting on page 198. For this example, you will create and store the ramp waveform shown below using four waveform points.



1 Select the arbitrary waveform function.

When you press Arb to select the arbitrary function, a temporary message is displayed indicating which waveform is currently selected.

2 Start the arbitrary waveform editor.

Press the **Create New** softkey to start the waveform editor. While in the waveform editor, you define the waveform by specifying time and voltage values for each point in the waveform. When creating a new waveform, the previous waveform in volatile memory is overwritten.

3 Set the waveform period.

Press the **Cycle Period** softkey to set the *time* boundaries for the waveform. The time value of the last point that can be defined in the waveform must be *less* than the specified cycle period.

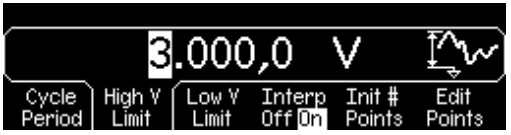
For this example, set the period of the waveform to 10 ms.



4 Set the waveform voltage limits.

Press the **High V Limit** and **Low V Limit** softkeys to set the upper and lower voltage levels that can be reached while editing the waveform. The upper limit *must* be greater than the lower limit. By default, Point #1 is set equal to the upper limit and Point #2 is set equal to the lower limit.

For this example, set the upper limit to 3.0 V and the lower limit to 0 V.



5 Select the interpolation method.

Press the **Interp** softkey to enable or disable linear interpolation between waveform points (this feature is available from the front panel only). With interpolation enabled (default), the waveform editor makes a straight-line connection between points. With interpolation disabled, the waveform editor maintains a constant voltage level between points and creates a “step-like” waveform.

For this example, turn on linear interpolation.

6 Set the initial number of waveform points.

You can create an arbitrary waveform with up to 65,536 (64K) points. The waveform editor initially builds a waveform with two points and automatically connects the last point of the waveform to the voltage level of the first point to create a continuous waveform. Press the **Init # Points** softkey to specify the initial number of waveform points (you can add or remove points later if necessary).

For this example, set the initial number of points to "4".

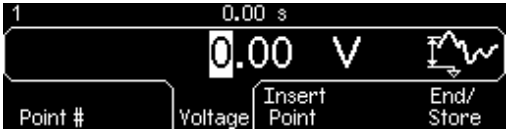
7 Start the point-by-point editing process.

Press the **Edit Points** softkey to accept the initial waveform settings and begin point-by-point editing. The status line at the top of the display window shows the point number in *yellow*, the time value of the current point in *green*, and the voltage value of the current point in *magenta*.

8 Define the first waveform point.

Press the **Voltage** softkey to set the voltage level for Point #1 (this point is fixed at a time of 0 seconds). By default, Point #1 is set equal to the upper limit.

For this example, set the voltage level of Point #1 to 0 V.

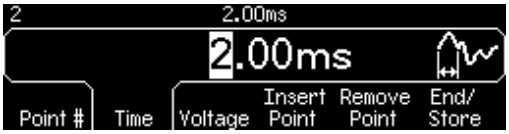


The waveform editor performs all amplitude calculations using Vpp, rather than Vrms or dBm.

9 Define the next waveform point.

Press the **Point #** softkey and then turn the knob to move to Point #2. Press the **Time** softkey to set the time for the current point (this softkey is not available for Point #1). Press the **Voltage** softkey to set the voltage level for the current point.

For this example, set the time to 2 ms and the voltage level to 3.0 V.



10 Define the remaining waveform points.

Using the **Time** and **Voltage** softkeys, define the remaining waveform points using the values shown in the table below.

Point	Time Value	Voltage Value
1	0 s	0 V
2	2 ms	3 V
3	4 ms	1 V
4	7 ms	0 V

- The time value of the last point that can be defined in the waveform must be *less* than the specified cycle period.
- The waveform editor automatically connects the last waveform point to the voltage level of first point to create a continuous waveform.
- To insert additional points after the current waveform point, press the **Insert Point** softkey. The new point is inserted midway between the current point and the next defined point.
- To remove the current waveform point, press the **Remove Point** softkey. The remaining points are joined using the interpolation method currently selected. You cannot remove Point #1 since the waveform must have a defined initial value.

11 Store the arbitrary waveform in memory.

Press the **End / Store** softkey to store the new waveform in memory. Then press the **DONE** softkey to store the waveform in *volatile* memory ***or*** press the **Store in Non-Vol** softkey to store the waveform in one of four *non-volatile* memory locations.

You can assign a custom name to the four non-volatile memory locations.

- The custom name can contain up to 12 characters. The first character must be a letter but the remaining characters can be letters, numbers, or the underscore character (" _ ").
- To add additional characters, press the right-arrow key until the cursor is to the right of the existing name and then turn the knob.
- To delete all characters to the right of the cursor, press the [+/-] key.

For this example, assign the name "RAMP_NEW" to memory location 1 and then press the **STORE ARB** *softkey to store the waveform.*



The waveform is now stored in non-volatile memory and is currently being output from the function generator. The name that you used to store the waveform should now appear in the list of stored waveforms (under the **Stored Wform** softkey).

Additional Information on Arbitrary Waveforms

- As a shortcut to determine which arbitrary waveform is selected, press Arb. A temporary message is displayed on the front panel.
- In addition to creating a new arbitrary waveform from the front panel, you can also edit any existing user-defined waveforms. You can edit waveforms that were created either from the front panel or from the remote interface. However, you cannot edit any of the five built-in arbitrary waveforms.
- Press the **Edit Wform** softkey to edit any of the arbitrary waveforms stored in non-volatile memory or the waveform currently stored in volatile memory. While editing an existing waveform, please note the following interactions.
  - If you *increase* the cycle period, some points could potentially match existing points. The waveform editor will preserve the earliest points and remove all duplicates.
  - If you *decrease* the cycle period, the waveform editor will remove all points that were previously defined beyond the new period.
  - If you *increase* the voltage limits, there will be no change in the voltage levels of any existing points but there may be some loss of vertical resolution.
  - If you *decrease* the voltage limits, some of the existing points could potentially exceed the new limits. The waveform editor will reduce the voltage levels of such points to equal the new limits.
- If you select an arbitrary waveform as the *modulating* waveshape for AM or FM, the waveform is automatically limited to 8K points. Extra waveform points are removed using decimation.

System-Related Operations

This section gives information on topics such as instrument state storage, power-down recall, error conditions, self test, and front-panel display control. This information is not directly related to waveform generation but is an important part of operating the function generator.

Instrument State Storage

The function generator has five storage locations in non-volatile memory to store instrument states. The locations are numbered 0 through 4. The function generator automatically uses location "0" to hold the state of the instrument at power down. You can also assign a user-defined name to each of the locations (1 through 4) for use from the front panel.

- You can store the instrument state in any of the five storage locations. However, you can only recall a state from a location that contains a previously-stored state.
- From the *remote interface* only, you can use storage location "0" to store a fifth instrument state (you cannot store to this location from the front panel). However, note that location "0" is automatically overwritten when power is cycled (the instrument state previously stored will be overwritten).
- The state storage feature "remembers" the selected function (including arbitrary waveforms), frequency, amplitude, dc offset, duty cycle, symmetry, as well as any modulation parameters in use.
- When shipped from the factory, storage locations "1" through "4" are empty (location "0" contains the power-down state).
- When power is turned off, the function generator automatically stores its state in storage location "0". You can configure the function generator to automatically recall the power-down state when power is restored. However, when shipped from the factory, the function generator is configured to automatically recall the factory default state at power-on.

- You can assign a custom name to each of the storage locations (however, you cannot name location “0” from the front panel). You can name a location from the front panel or over the remote interface, but you can only recall a state by name from the front panel. From the remote interface, you can only recall a stored state using a number (0 through 4).
- The name can contain up to 12 characters. The first character *must* be a letter (A-Z), but the remaining characters can be letters, numbers (0-9), or the underscore character (“ _ ”). Blank spaces are not allowed. An error will be generated if you specify a name with more than 12 characters.
- The function generator *will not* prevent you from assigning the *same* custom name to different storage locations. For example, you can assign the same name to locations “1” and “2”.
- If you delete an arbitrary waveform from non-volatile memory after storing the instrument state, the waveform data is lost and the function generator *will not* output the waveform when the state is recalled. The built-in “exponential rise” waveform is output in place of the deleted waveform.
- The front-panel display state (*see “Display Control” on page 115*) is saved when you store the instrument state. When you recall the state, the front-panel display will return to the previous state.
- An instrument reset *does not* affect the configurations stored in memory. Once a state is stored, it remains until it is overwritten or specifically deleted.

- *Front-Panel Operation*: Press Store/Recall and then select the **Store State** or **Recall State** softkey. To delete a stored state, select the **Delete State** softkey (also removes the custom name for this memory location).

  To configure the function generator to recall the factory default state at power-on, press Store/Recall and then select the **Pwr-On Default** softkey. To configure the function generator to recall the power-down state when power is restored, press Store/Recall and select the **Pwr-On Last** softkey.

  You can assign a custom name to each of the four storage locations.

  - The custom name can contain up to 12 characters. The first character must be a letter but the remaining characters can be letters, numbers, or the underscore character (" _ ").
  - To add additional characters, press the right-arrow key until the cursor is to the right of the existing name and then turn the knob.
  - To delete all characters to the right of the cursor, press the +/- key.

- *Remote Interface Operation*:

  ```
  *SAV {0|1|2|3|4}
  *RCL {0|1|2|3|4}
  ```
  State 0 is the instrument state at power down.
  States 1, 2, 3, and 4 are user-defined states.

  To assign a custom name to a stored state to be recalled from the front panel, send the following command. From the remote interface, you can only recall a stored state using a number (0 through 4).

  ```
  MEM:STATE:NAME 1,TEST_WFORM_1
  ```

  To configure the function generator to automatically recall the power-down state when power is restored, send the following command.

  ```
  MEMory:STATe:RECall:AUTO ON
  ```

Error Conditions

A record of up to 20 command syntax or hardware errors can be stored in the function generator's error queue. *See chapter 5 for a complete listing of the errors.*

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. Errors are cleared as you read them. The function generator beeps once each time an error is generated (unless you have disabled the beeper).
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with "*Queue overflow*". No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the function generator responds with "*No error*".
- The error queue is cleared by the `*CLS` (clear status) command or when the power is cycled. The errors are also cleared when you read the error queue. The error queue *is not* cleared by an instrument reset (`*RST` command).
- *Front-Panel Operation*: Press Help and then select the topic entitled "*View the remote command error queue*" (topic number 2). Then press the **SELECT** softkey to view the errors in the error queue. As shown below, the first error in the list (i.e., the error at the top of the list) is the first error that was generated.



- *Remote Interface Operation*:

  `SYSTem:ERRor?` *Reads one error from the error queue*

  Errors have the following format (the error string may contain up to 255 characters).

  ```
  -113,"Undefined header"
  ```

Beeper Control

Normally, the function generator will emit a tone when an error is generated from the front-panel or over the remote interface. You may want to disable the front-panel beeper for certain applications.

- The beeper state is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset. When shipped from the factory, the beeper is enabled.
- Turning off the beeper *does not* disable the key click generated when you press a front-panel key or turn the knob.
- *Front-Panel Operation*: Press Utility and then select the **Beep** softkey from the "System" menu.
- *Remote Interface Operation*:

```
SYSTem:BEEPer                    Issue a single beep immediately
SYSTem:BEEPer:STATe {OFF|ON}     Disable/enable beeper
```

Display Bulb-Saver

The front-panel display bulb will normally turn off and the screen will go blank after *one hour* of inactivity. For certain applications, you may want to disable the bulb-saver feature. *This feature is available from the front panel only.*

- The bulb-saver setting is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset. When shipped from the factory, the bulb-saver mode is enabled.
- *Front-Panel Operation*: Press Utility and then select the **Scrn Svr** (screen saver) softkey from the "System" menu.

Display Contrast

To optimize the readability of the front-panel display, you can adjust the contrast setting. *This feature is available from the front panel only.*

- Display contrast: 0 to 100. *The default is 50*.
- The contrast setting is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- *Front-Panel Operation*: Press Utility and then select the **Display Contr** softkey from the "System" menu.

Self-Test

- A *power-on* self-test occurs automatically when you turn on the function generator. This limited test assures you that the function generator is operational.
- A *complete* self-test runs a series of tests and takes approximately 15 seconds to execute. If all tests pass, you can have high confidence that the function generator is fully operational.
- If the complete self-test is successful, "Self-Test Passed" is displayed on the front panel. If the self-test fails, "Self-Test Failed" is displayed and an error number is shown. See the Agilent 33250A *Service Guide* for instructions on returning the instrument to Agilent for service.
- *Front-Panel Operation*: Press Utility and then select the **Self Test** softkey from the "Test/Cal" menu.
- *Remote Interface Operation*:

  ```
  *TST?
  ```

  Returns "0" if the self-test passes or "1" if it fails. If the self-test fails, an error message is also generated with additional information on why the test failed.

Display Control

For security reasons, or to speed up the rate at which the function generator can execute commands from the remote interface, you may want to turn off the front-panel display. From the remote interface, you can also display a 12-character message on the front panel.

- You can only disable the front-panel display by sending a command from the remote interface (you cannot disable the front panel while in local operation).
- When disabled, the front-panel display is blanked (however, the bulb used to backlight the display remains enabled). All keys except Local are locked out when the display is disabled.
- Sending a message to the front-panel display from the remote interface overrides the display state. This means that you can display a message even if the display is currently disabled (remote interface errors are always displayed even if the display is disabled).
- The display is automatically enabled when power is cycled, after an instrument reset (`*RST` command), or when you return to local (front panel) operation. Press the Local key or execute the IEEE-488 GTL (*Go To Local*) command from the remote interface to return to the local state.
- The display state is saved when you store the instrument state using `*SAV` command. When you recall the instrument state using `*RCL` command, the front-panel display will return to the previous state.
- You can display a text message on the front panel by sending a command from the remote interface. You can use upper- or lower-case letters (A-Z), numbers (0-9), and any other character on a standard computer keyboard. Depending on the number of characters that you specify, the function generator will choose one of two font sizes to display the message. You can display approximately 12 characters in a large font and approximately 40 characters in a small font.

- *Remote Interface Operation*: The following command turns off the front-panel display.

```
DISP OFF
```

  The following command displays a message on the front panel and turns on the display if currently disabled.

```
DISP:TEXT 'Test in Progress...'
```

  To clear the message displayed on the front panel (without changing the display state), send the following command.

```
DISP:TEXT CLEAR
```

Number Format

The function generator can show numbers on the front-panel display with periods or commas for the decimal point and digits separator. *This feature is available from the front panel only.*



Decimal Point: Period
Digits Separator: Comma



Decimal Point: Comma
Digits Separator: None

- The number format is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset. When shipped from the factory, a period is used for the decimal point and commas are used for the digits separator (e.g., 1.000,000,00 kHz).
- *Front-Panel Operation*: Press (Utility) and then select the **Number Format** softkey from the "System" menu.

Firmware Revision Query

You can query the function generator to determine which revision of firmware is currently installed. The revision code contains five numbers in the form "**m.mm-l.ll-f.ff-gg-p**".

m.mm	=	Main firmware revision number
l.ll	=	Loader firmware revision number
f.ff	=	I/O processor firmware revision number
gg	=	Gate array revision number
p	=	Printed circuit board revision number

- *Front-Panel Operation*: Press Utility and then select the **Cal Info** softkey from the "Test/Cal" menu. The revision code is listed as one of the messages on the front-panel display.
- *Remote Interface Operation*: Use the following command to read the function generator's firmware revision numbers (be sure to dimension a string variable with at least 50 characters).

  ```
  *IDN?
  ```

  This command returns a string in the form:

  Agilent Technologies,33250A,0,**m.mm-l.ll-f.ff-gg-p**

SCPI Language Version Query

The function generator complies with the rules and conventions of the present version of SCPI (*Standard Commands for Programmable Instruments*). You can determine the SCPI version with which the instrument is in compliance by sending a query from the remote interface.

You cannot query the SCPI version from the front panel.

- *Remote Interface Operation*:

  ```
  SYSTem:VERSion?
  ```

  Returns a string in the form "YYYY.V", where "YYYY" represents the year of the version, and "V" represents a version number for that year (e.g., 1997.0).

Remote Interface Configuration

This section gives information on configuring the function generator for remote interface communication. For information on configuring the instrument from the front panel, see "To Configure the Remote Interface" starting on page 44. For information on the SCPI commands available to program the function generator over the remote interface, see chapter 4, "Remote Interface Reference" starting on page 129.

GPIB Address

Each device on the GPIB (IEEE-488) interface must have a unique address. You can set the function generator's address to any value between 0 and 30. The address is set to "**10**" when the function generator is shipped from the factory. The GPIB address is displayed at power-on.

You can set the GPIB address from the front panel only.

- The address is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- Your computer's GPIB interface card has its own address. Be sure to avoid using the computer's address for any instrument on the interface bus.
- *Front-Panel Operation*: Press Utility and then select the **GPIB Address** softkey from the "I/O" menu.

  See also "To Configure the Remote Interface," on page 44.

Remote Interface Selection

The instrument is shipped with both a GPIB (IEEE-488) interface and an RS-232 interface. Only one interface can be enabled at a time. When shipped from the factory, the GPIB interface is selected.

- The interface selection is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- If you select the GPIB interface, you must select a unique address for the instrument. The GPIB address is displayed on the front panel when you turn on the function generator.
- If you select the RS-232 interface, you must also set the baud rate, parity, and handshake mode for the function generator. The interface selection is displayed on the front panel when you turn on the function generator.
- *Front-Panel Operation*: Press Utility and then select the **GPIB** or **RS-232** softkey from the "I/O" menu.

  See also "To Configure the Remote Interface," on page 44.

- *Remote Interface Operation*:

```
SYSTem:INTerface {GPIB|RS232}
```

Refer to "RS-232 Interface Configuration" on page 219 for information on connecting the 33250A to a computer over the RS-232 interface.

Baud Rate Selection (RS-232)

You can select one of several baud rates for RS-232 operation.
When shipped from the factory, the baud rate is set to 57,600 baud.

You can set the baud rate from the front panel only.

- Select one of the following: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, **57600** (*factory setting*), or 115200 baud.
- The baud rate selection is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- *Front-Panel Operation*: Press Utility and then select the **Baud Rate** softkey from the "I/O" menu.

Parity Selection (RS-232)

You can select the parity for RS-232 operation. When shipped from the factory, the instrument is configured for no parity with 8 data bits.

You can set the parity from the front panel only.

- Select one of the following: **None** (8 data bits), Even (7 data bits), or Odd (7 data bits). When you set the parity, you are also setting the number of data bits.
- The parity selection is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- *Front-Panel Operation*: Press Utility and then select the **Parity/# Bits** softkey from the "I/O" menu.

Handshake Selection (RS-232)

You can select one of several handshake (or "flow control") methods to coordinate the transfer of data between the function generator and your computer or modem. The method that you select will be determined by the handshake mode used by your computer or modem.

You can select the handshake mode from the front panel only.

- Select one of the following: None, **DTR/DSR** (*factory setting*), Modem, RTS/CTS, or XON/XOFF.
- *None*: In this mode, data is sent and received over the interface without any flow control used. When using this method, use a slower baud rate (< 9600 baud) and avoid sending more than 128 characters without stopping or reading a response.
- *DTR/DSR*: In this mode, the function generator monitors the state of the DSR (data set ready) line on the RS-232 connector. When the line goes true, the function generator sends data over the interface. When the line goes false, the function generator stops sending information (typically within six characters). The function generator sets the DTR line false when the input buffer is almost full (about 100 characters) and releases the line when space is available again.
- *Modem*: This mode uses the DTR/DSR and RTS/CTS lines to control the flow of data between the function generator and a modem. When the RS-232 interface is selected, the function generator sets the DTR line true. The DSR line is set true when the modem is on-line. The function generator sets the RTS line true when it is ready to receive data. The modem sets the CTS line true when it is ready to accept data. The function generator sets the RTS line false when the input buffer is almost full (about 100 characters) and releases the line when space is available again.
- *RTS/CTS*: This mode operates the same as the DTR/DSR mode but uses the RTS (request to send) and CTS (clear to send) lines on the RS-232 connector instead. When the CTS line goes true, the function generator sends data over the interface. When the line goes false, the function generator stops sending information (typically within six characters). The function generator sets the RTS line false when the input buffer is almost full (about 100 characters) and releases the line when space is available again.

- *XON/XOFF*: This mode uses special characters embedded in the data stream to control the flow. If the function generator is addressed to send data, it continues sending data until the "XOFF" character (13H) is received. When the "XON" character (11H) is received, the function generator resumes sending data.
- The handshake selection is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- To download binary data for arbitrary waveforms over the RS-232 interface, you can use any handshake mode except XON/XOFF. Be sure to select "Parity None" (8 data bits). You must also insert a pause of approximately 1 ms between sending the header and sending the binary block.
- *Front-Panel Operation*: Press Utility and then select the **Handshake** softkey from the "I/O" menu.

Calibration Overview

This section gives a brief introduction to the calibration features of the function generator. For a more detailed discussion of the calibration procedures, see chapter 4 in the Agilent 33250A *Service Guide*.

Calibration Security

This feature allows you to enter a security code to prevent accidental or unauthorized calibrations of the function generator. When you first receive your function generator, it is secured. Before you can perform a calibration, you must unsecure the function generator by entering the correct security code.

> If you forget your security code, you can disable the security feature by adding a jumper inside the instrument. See the Agilent 33250A Service Guide for more information.

- The security code is set to "AT33250A" when the function generator is shipped from the factory. The security code is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- The security code may contain up to 12 alphanumeric characters. The first character must be a letter, but the remaining characters can be letters, numbers, or an underscore (" _ "). You do not have to use all 12 characters but the first character must always be a letter.

To Unsecure for Calibration You can unsecure the function generator either from the front panel or over the remote interface. The function generator is secured when shipped from the factory and the security code is set to "AT33250A".

- Once you enter a security code, that code must be used for both front-panel and remote operation. For example, if you secure the function generator from the front panel, you must use that same code to unsecure it from the remote interface.
- *Front-Panel Operation*: Press Utility and then select the **Secure Off** softkey from the "Test/Cal" menu.
- *Remote Interface Operation*: To unsecure the function generator, send the following command with the correct security code.

```
CAL:SECURE:STATE OFF,AT33250A
```

To Secure Against Calibration You can secure the function generator either from the front panel or over the remote interface. The function generator is secured when shipped from the factory and the security code is set to "AT33250A".

- Once you enter a security code, that code must be used for both front-panel and remote operation. For example, if you secure the function generator from the front panel, you must use that same code to unsecure it from the remote interface. .
- *Front-Panel Operation*: Press Utility and then select the **Secure On** softkey from the "Test/Cal" menu.
- *Remote Interface Operation*: To secure the function generator, send the following command with the correct security code.

```
CAL:SECURE:STATE ON,AT33250A
```

To Change the Security Code To change the security code, you must first unsecure the function generator, and then enter a new code. Make sure you have read the security code rules described on page 123 before attempting to change the security code.

- *Front-Panel Operation*: To change the security code, unsecure the function generator using the old security code. Then press Utility and select the **Secure Code** softkey from the "Test/Cal" menu. Changing the code from the front panel also changes the security code as seen from the remote interface.
- *Remote Interface Operation*: To change the security code, you must first unsecure the function generator using the old security code. Then enter the new code as shown below.

```
CAL:SECURE:STATE OFF, AT33250A    Unsecure with old code
CAL:SECURE:CODE SN123456789       Enter new code
```

Calibration Count

You can query the function generator to determine how many calibrations have been performed. Note that your function generator was calibrated before it left the factory. When you receive your function generator, be sure to read the count to determine its initial value.

- The calibration count is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- The calibration count increments up to a maximum of 65,535 after which it rolls over to "0". Since the value increments by one for each calibration point, a complete calibration may increase the value by many counts.
- *Front-Panel Operation*: Press Utility and then select the **Cal Info** softkey from the "Test/Cal" menu. The calibration count is listed as one of the messages on the display.
- *Remote Interface Operation*:

```
CALibration:COUNt?
```

Calibration Message

The function generator allows you to store one message in calibration memory in the mainframe. For example, you can store such information as the date when the last calibration was performed, the date when the next calibration is due, the function generator's serial number, or the name and phone number of the person to contact for a new calibration.

- You can record a calibration message *only* from the remote interface and *only* when the function generator is unsecured. You can read the message from either the front-panel or over the remote interface. You can *read* the calibration message whether the function generator is secured or unsecured.
- The calibration message may contain up to 40 characters (additional characters are truncated).
- Storing a calibration message will overwrite any message previously stored in memory.
- The calibration message is stored in *non-volatile* memory and *does not* change when power has been off or after a remote interface reset.
- *Front-Panel Operation*: Press Utility and then select the **Cal Info** softkey from the "Test/Cal" menu. The calibration message is listed as one of the messages on the display.
- *Remote Interface Operation*: To store the calibration message, send the following command.

```
CAL:STR 'Cal Due: 01 June 2001'
```

Factory Default Settings

Output Configuration	**Factory Setting**
Function	Sine wave
Frequency	1 kHz
Amplitude / Offset	100 mVpp / 0.000 Vdc
Output Units	Vpp
Output Termination	50Ω
Autorange	On
Modulation (AM, FM, FSK)	**Factory Setting**
Carrier Waveform	1 kHz Sine wave
Modulating Waveform	100 Hz Sine wave
AM Depth	100%
FM Deviation	100 Hz
FSK "Hop" Frequency	100 Hz
FSK Rate	10 Hz
Modulation State	Off
Sweep	**Factory Setting**
Start / Stop Frequency	100 Hz / 1 kHz
Sweep Time	1 Second
Sweep Mode	Linear
Sweep State	Off
Burst	**Factory Setting**
Burst Frequency	1 kHz
Burst Count	1 Cycle
Burst Period	10 ms
Burst Starting Phase	0°
Burst State	Off
System-Related Operations	**Factory Setting**
• Power-Down Recall	• Disabled
Display Mode	On
Error Queue	Errors are Cleared
Stored States, Stored Arbs	No Change
Output State	Off
Triggering Operations	**Factory Setting**
Trigger Source	Internal (Immediate)
Remote Interface Configuration	**Factory Setting**
• GPIB Address	• 10
• Interface	• GPIB (IEEE-488)
• Baud Rate	• 57,600 Baud
• Parity	• None (8 data bits)
• Handshake	• DTR / DSR
Calibration	**Factory Setting**
Calibration State	Secured

Parameters marked with a bullet (•) are stored in *non-volatile* memory.

Note: *The power-on state will be different if you have enabled the power-down recall mode.*

See "Instrument State Storage" on page 109.

For your convenience, this table is duplicated on the rear cover of this manual and on the Quick Reference Card.

Remote Interface Reference

Remote Interface Reference

- SCPI Command Summary, *on page 131*
- Simplified Programming Overview, *on page 142*
- Using the APPLy Command, *on page 144*
- Output Configuration Commands, *on page 153*
- Pulse Configuration Commands, *on page 166*
- Amplitude Modulation (AM) Commands, *on page 169*
- Frequency Modulation (FM) Commands, *on page 172*
- Frequency-Shift Keying (FSK) Commands, *on page 176*
- Frequency Sweep Commands, *on page 179*
- Burst Mode Commands, *on page 187*
- Triggering Commands, *on page 195*
- Arbitrary Waveform Commands, *on page 198*
- State Storage Commands, *on page 209*
- System-Related Commands, *on page 213*
- Interface Configuration Commands, *on page 218*
- RS-232 Interface Configuration, *on page 219*
- Phase-Lock Commands, *on page 223*
- The SCPI Status System, *on page 225*
- Status Reporting Commands, *on page 235*
- Calibration Commands, *on page 239*
- An Introduction to the SCPI Language, *on page 241*
- Using Device Clear, *on page 246*

4

If you are a first-time user of the SCPI language, please refer to these sections to become familiar with the language before attempting to program the function generator.

SCPI Command Summary

Throughout this manual, the following conventions are used for SCPI command syntax for remote interface programming:

- Square brackets (**[ ]**) indicate optional keywords or parameters.
- Braces (**{ }**) enclose parameters within a command string.
- Triangle brackets (**< >**) enclose parameters for which you must substitute a value.
- A vertical bar (**|**) separates multiple parameter choices.

The APPLy Commands

(*see page 144 for more information*)

```
APPLy
  :SINusoid [<frequency> [,<amplitude> [,<offset>] ]]
  :SQUare [<frequency> [,<amplitude> [,<offset>] ]]
  :RAMP [<frequency> [,<amplitude> [,<offset>] ]]
  :PULSe [<frequency> [,<amplitude> [,<offset>] ]]
  :NOISe [<frequency>|DEF>[1] [,<amplitude> [,<offset>] ]]
  :DC [<frequency>|DEF>[1] [,<amplitude>|DEF>[1] [,<offset>] ]]
  :USER [<frequency> [,<amplitude> [,<offset>] ]]
APPLy?
```

[1] *This parameter has no effect for this command but you MUST specify a value or "DEFault".*

Output Configuration Commands

(*see page 153 for more information*)

```
FUNCtion {SINusoid|SQUare|RAMP|PULSe|NOISe|DC|USER}
FUNCtion?

FREQuency {<frequency>|MINimum|MAXimum}
FREQuency? [MINimum|MAXimum]

VOLTage {<amplitude>|MINimum|MAXimum}
VOLTage? [MINimum|MAXimum]

VOLTage:OFFSet {<offset>|MINimum|MAXimum}
VOLTage:OFFSet? [MINimum|MAXimum]

VOLTage
  :HIGH {<voltage>|MINimum|MAXimum}
  :HIGH? [MINimum|MAXimum]
  :LOW {<voltage>|MINimum|MAXimum}
  :LOW? [MINimum|MAXimum]

VOLTage:RANGe:AUTO {OFF|ON|ONCE}
VOLTage:RANGe:AUTO?

VOLTage:UNIT {VPP|VRMS|DBM}
VOLTage:UNIT?

FUNCtion:SQUare:DCYCle {<percent>|MINimum|MAXimum}
FUNCtion:SQUare:DCYCle? [MINimum|MAXimum]

FUNCtion:RAMP:SYMMetry {<percent>|MINimum|MAXimum}
FUNCtion:RAMP:SYMMetry? [MINimum|MAXimum]

OUTPut {OFF|ON}
OUTPut?

OUTPut:LOAD {<ohms>|INFinity|MINimum|MAXimum}
OUTPut:LOAD? [MINimum|MAXimum]

OUTPut:POLarity {NORMal|INVerted}
OUTPut:POLarity?

OUTPut:SYNC {OFF|ON}
OUTPut:SYNC?
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Pulse Configuration Commands

(*see page 166 for more information*)

```
PULSe:PERiod {<seconds>|MINimum|MAXimum}
PULSe:PERiod? [MINimum|MAXimum]

PULSe
  :WIDTh {<seconds>|MINimum|MAXimum}          50% to 50% Thresholds
  :WIDTh? [MINimum|MAXimum]
  :TRANsition {<seconds>|MINimum|MAXimum}     10% to 90% Thresholds
  :TRANsition? [MINimum|MAXimum]
```

Modulation Commands

(*see page 169 for more information*)

AM Commands

```
AM:INTernal
  :FUNCtion {SINusoid|SQUare|RAMP|NRAMp|TRIangle|NOISe|USER}
  :FUNCtion?

AM:INTernal
  :FREQuency {<frequency>|MINimum|MAXimum}
  :FREQuency? [MINimum|MAXimum]

AM:DEPTh {<depth in percent>|MINimum|MAXimum}
AM:DEPTh? [MINimum|MAXimum]

AM:SOURce {INTernal|EXTernal}
AM:SOURce?

AM:STATe {OFF|ON}
AM:STATe?
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

FM Commands

```
FM:INTernal
  :FUNCtion {SINusoid|SQUare|RAMP|NRAMp|TRIangle|NOISe|USER}
  :FUNCtion?
```

```
FM:INTernal
  :FREQuency {<frequency>|MINimum|MAXimum}
  :FREQuency? [MINimum|MAXimum]
```

```
FM:DEViation {<peak deviation in Hz>|MINimum|MAXimum}
FM:DEViation? [MINimum|MAXimum]
```

```
FM:SOURce {INTernal|EXTernal}
FM:SOURce?
```

```
FM:STATe {OFF|ON}
FM:STATe?
```

FSK Commands

```
FSKey:FREQuency {<frequency>|MINimum|MAXimum}
FSKey:FREQuency? [MINimum|MAXimum]
```

```
FSKey:INTernal:RATE {<rate in Hz>|MINimum|MAXimum}
FSKey:INTernal:RATE? [MINimum|MAXimum]
```

```
FSKey:SOURce {INTernal|EXTernal}
FSKey:SOURce?
```

```
FSKey:STATe {OFF|ON}
FSKey:STATe?
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Sweep Commands

(*see page 181 for more information*)

```
FREQuency
  :STARt {<frequency>|MINimum|MAXimum}
  :STARt? [MINimum|MAXimum]
  :STOP {<frequency>|MINimum|MAXimum}
  :STOP? [MINimum|MAXimum]

FREQuency
  :CENTer {<frequency>|MINimum|MAXimum}
  :CENTer? [MINimum|MAXimum]
  :SPAN {<frequency>|MINimum|MAXimum}
  :SPAN? [MINimum|MAXimum]

SWEep
  :SPACing {LINear|LOGarithmic}
  :SPACing?
  :TIME {<seconds>|MINimum|MAXimum}
  :TIME? [MINimum|MAXimum]

SWEep:STATe {OFF|ON}
SWEep:STATe?

TRIGger:SOURce {IMMediate|EXTernal|BUS}
TRIGger:SOURce?

TRIGger:SLOPe {POSitive|NEGative}        "Trig In" Connector
TRIGger:SLOPe?

OUTPut
  :TRIGger:SLOPe {POSitive|NEGative}     "Trig Out" Connector
  :TRIGger:SLOPe?
  :TRIGger {OFF|ON}
  :TRIGger?

MARKer:FREQuency {<frequency>|MINimum|MAXimum}
MARKER:FREQuency? [MINimum|MAXimum]

MARKer {OFF|ON}
MARKer?
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Burst Commands

(*see page 187 for more information*)

```
BURSt:MODE {TRIGgered|GATed}
BURSt:MODE?

BURSt:NCYCles {<# cycles>|INFinity|MINimum|MAXimum}
BURSt:NCYCles? [MINimum|MAXimum]

BURSt:INTernal:PERiod {<seconds>|MINimum|MAXimum}
BURSt:INTernal:PERiod? [MINimum|MAXimum]

BURSt:PHASe {<angle>|MINimum|MAXimum}
BURSt:PHASe? [MINimum|MAXimum]

BURSt:STATe {OFF|ON}
BURSt:STATe?

UNIT:ANGLe {DEGree|RADian}
UNIT:ANGLe?

TRIGger:SOURce {IMMediate|EXTernal|BUS}      Triggered Burst
TRIGger:SOURce?

TRIGger:DELay {<seconds>|MINimum|MAXimum}
TRIGger:DELay? [MINimum|MAXimum]

TRIGger:SLOPe {POSitive|NEGative}            "Trig In" Connector
TRIGger:SLOPe?

BURSt:GATE:POLarity {NORMal|INVerted}        External Gated Burst
BURSt:GATE:POLarity?

OUTPut                                       "Trig Out" Connector
  :TRIGger:SLOPe {POSitive|NEGative}
  :TRIGger:SLOPe?
  :TRIGger {OFF|ON}
  :TRIGger?
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Arbitrary Waveform Commands

(*see page 198 for more information)*

```
DATA VOLATILE, <value>, <value>, . . .

DATA:DAC VOLATILE, {<binary block>|<value>, <value>, . . . }

FORMat:BORDer {NORMal|SWAPped}          Specify Byte Order
FORMat:BORDer?

DATA:COPY <destination arb name> [,VOLATILE]

FUNCtion:USER {<arb name>[1]|VOLATILE}
FUNCtion:USER?

FUNCtion USER
FUNCtion?

DATA
  :CATalog?
  :NVOLatile:CATalog?
  :NVOLatile:FREE?

DATA:DELete <arb name>
DATA:DELete:ALL

DATA
  :ATTRibute:AVERage? [<arb name>[1]]
  :ATTRibute:CFACtor? [<arb name>[1]]
  :ATTRibute:POINts? [<arb name>[1]]
  :ATTRibute:PTPeak? [<arb name>[1]]
```

[1] *The names of the built-in arb waveforms are*: EXP_RISE, EXP_FALL, NEG_RAMP, SINC, and CARDIAC.

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Triggering Commands

(*see page 195 for more information*)

These commands are used for Sweep and Burst only.

```
TRIGger:SOURce {IMMediate|EXTernal|BUS}
TRIGger:SOURce?
```

```
TRIGger
```

```
*TRG
```

```
TRIGger:DELay {<seconds>|MINimum|MAXimum}
TRIGger:DELay? [MINimum|MAXimum]
```
Triggered Burst Mode

```
TRIGger:SLOPe {POSitive|NEGative}
TRIGger:SLOPe?
```
"Trig In" Connector

```
BURSt:GATE:POLarity {NORMal|INVerted}
BURSt:GATE:POLarity?
```
External Gated Burst

```
OUTPut
  :TRIGger:SLOPe {POSitive|NEGative}
  :TRIGger:SLOPe?
  :TRIGger {OFF|ON}
  :TRIGger?
```
"Trig Out" Connector

State Storage Commands

(*see page 209 for more information*)

```
*SAV {0|1|2|3|4}
*RCL {0|1|2|3|4}
```
State 0 is the instrument state at power down.
States 1 through 4 are user-defined states.

```
MEMory:STATe
  :NAME {0|1|2|3|4} [,<name>]
  :NAME? {0|1|2|3|4}
  :DELete {0|1|2|3|4}
  :RECall:AUTO {OFF|ON}
  :RECall:AUTO?
  :VALid? {0|1|2|3|4}
```

```
MEMory:NSTates?
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

System-Related Commands

(*see page 213 for more information)*

```
SYSTem:ERRor?
*IDN?
DISPlay {OFF|ON}
DISPlay?
DISPlay
  :TEXT <quoted string>
  :TEXT?
  :TEXT:CLEar
*RST
*TST?
SYSTem:VERSion?
SYSTem
  :BEEPer
  :BEEPer:STATe {OFF|ON}
  :BEEPer:STATe?
*LRN?
*OPC
*OPC?
*WAI
```

Interface Configuration Commands

(*see page 218 for more information*)

```
SYSTem:INTerface {GPIB|RS232}
SYSTem:LOCal
SYSTem:RWLock
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Phase-Lock Commands

(*see page 223 for more information)*

```
PHASe {<angle>|MINimum|MAXimum}
PHASe? [MINimum|MAXimum]

PHASe:REFerence

PHASe:UNLock:ERRor:STATe {OFF|ON}
PHASe:UNLock:ERRor:STATe?

UNIT:ANGLe {DEGree|RADian}
UNIT:ANGLe?
```

Status Reporting Commands

(*see page 235 for more information)*

```
*STB?

*SRE <enable value>
*SRE?

STATus
  :QUEStionable:CONDition?
  :QUEStionable[:EVENt]?
  :QUEStionable:ENABle <enable value>
  :QUEStionable:ENABle?

*ESR?

*ESE <enable value>
*ESE?

*CLS

STATus:PRESet

*PSC {0|1}
*PSC?

*OPC
```

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Calibration Commands

(*see page 239 for more information)*

```
CALibration?

CALibration
  :SECure:STATe {OFF|ON},<code>
  :SECure:STATe?
  :SECure:CODE <new code>
  :SETup <0|1|2|3| . . . |115>
  :SETup?
  :VALue <value>
  :VALue?
  :COUNt?
  :STRing <quoted string>
  :STRing?
```

IEEE 488.2 Common Commands

```
*CLS

*ESR?
*ESE <enable value>
*ESE?

*IDN?

*LRN?

*OPC

*OPC?

*PSC {0|1}
*PSC?

*RST

*SAV {0|1|2|3|4}
*RCL {0|1|2|3|4}

*STB?
*SRE <enable value>
*SRE?

*TRG

*TST?
```

State 0 is the instrument state at power down. States 1 through 4 are user-defined states.

Parameters shown in **bold** *are selected following a* `*RST` *(reset) command.*

Simplified Programming Overview

This section gives an overview of the basic techniques used to program the function generator over the remote interface. This section is only an overview and does not give all of the details you will need to write your own application programs. Refer to the remainder of this chapter and also the application examples in chapter 6 for more details. You may also want to refer to the reference manual that came with your programming application for further details on instrument control.

Using the APPLy Command

The APPLy command provides the most straightforward method to program the function generator over the remote interface. For example, the following command string sent from your computer will output a 3 Vpp sine wave at 5 kHz with a -2.5 volt offset.

```
APPL:SIN 5.0E+3, 3.0, -2.5
```

Using the Low-Level Commands

Although the APPLy command provides the most straightforward method to program the function generator, the low-level commands give you more flexibility to change individual parameters. For example, the following command strings sent from your computer will output a 3 Vpp sine wave at 5 kHz with a -2.5 volt offset.

```
FUNC SIN                 Select sine wave function
FREQ 5000                Set frequency to 5 kHz
VOLT 3.0                 Set amplitude to 3 Vpp
VOLT:OFFS -2.5           Set offset to -2.5 Vdc
```

Reading a Query Response

Only the query commands (commands that end with "**?**") will instruct the function generator to send a response message. Queries return internal instrument settings. For example, the following command string sent from your computer will read the function generator's error queue and retrieve the response from the most recent error.

```
dimension statement      Dimension string array (255 elements)
SYST:ERR?                Read error queue
enter statement          Enter error string response
```

Selecting a Trigger Source

When *Sweep* or *Burst* is enabled, the function generator will accept an immediate internal trigger, a hardware trigger from the rear-panel *Trig In* connector, a manual trigger from the front-panel (Trigger) key, or a software (bus) trigger. By default, the internal trigger source is selected. If you want to use an external or a software trigger source, you must first select that source. For example, the following command strings sent from your computer will output a 3-cycle burst each time the rear-panel *Trig In* connector receives the rising edge of a TTL pulse.

```
BURS:NCYC 3              Set burst count to 3 cycles
TRIG:SLOP POS            Set polarity to rising edge
TRIG:SOUR EXT            Select external trigger source
BURS:STAT ON             Enable burst mode
```

Using the APPLy Command

See also "Output Configuration" starting on page 49 in chapter 3.

The APPLy command provides the most straightforward method to program the function generator over the remote interface. You can select the function, frequency, amplitude, and offset all in one command as shown in the syntax statement below.

```
APPLy:<function> [<frequency> [,<amplitude> [,<offset>] ]]
```

For example, the following command string sent from your computer will output a 3 Vpp sine wave at 5 kHz with a -2.5 volt offset.

```
APPL:SIN 5 KHZ, 3.0 VPP, -2.5 V
```

The APPLy command performs the following operations:

- Sets the trigger source to *Immediate* (equivalent to sending the TRIG:SOUR IMM command).
- Turns off any modulation, sweep, or burst mode currently enabled and places the instrument in the continuous waveform mode.
- Turns on the *Output* connector (OUTP ON command) but *does not* change the output termination setting (OUTP:LOAD command).
- Overrides the voltage autorange setting and automatically enables autoranging (VOLT:RANG:AUTO command).
- For square waveforms, overrides the current duty cycle setting and automatically selects 50% (FUNC:SQU:DCYC command).
- For ramp waveforms, overrides the current symmetry setting and automatically selects 100% (FUNC:RAMP:SYMM command).

The syntax statements for the APPLy command are shown on page 149.

Output Frequency

- For the *frequency* parameter of the `APPLy` command, the output frequency range depends on the function specified. You can substitute "MINimum", "MAXimum", or "DEFault" in place of a specific value for the *frequency* parameter. MIN selects the lowest frequency allowed for the function specified and MAX selects the highest frequency allowed. *The default frequency is 1 kHz for all functions*.

Function	Minimum Frequency	Maximum Frequency
Sine	1 µHz	80 MHz
Square	1 µHz	80 MHz
Ramp	1 µHz	1 MHz
Pulse	500 µHz	50 MHz
Noise, DC	Not Applicable	Not Applicable
Arbs	1 µHz	25 MHz

- *Limits Due to Function*: The frequency limits are determined by the function specified in the `APPLy` command. For example, if you are currently outputting an 80 MHz sine wave and then change to the ramp function using the `APPLy` command, the function generator will automatically adjust the output frequency to 1 MHz (the upper limit for ramps). *From the remote interface, a "Data out of range" error will be generated and the frequency will be adjusted as described.*

Output Amplitude

- For the *amplitude* parameter of the APPLy command, the output amplitude range depends on the function specified and the output termination. You can substitute "MINimum", "MAXimum", or "DEFault" in place of a specific value for the *amplitude* parameter. MIN selects the smallest amplitude (1 mVpp into 50 ohms). MAX selects the largest amplitude for the function specified (at most 10 Vpp into 50 ohms depending on function and offset voltage). *The default amplitude is 100 mVpp (into 50 ohms) for all functions.*

- *Limits Due to Output Termination*: The output amplitude limits are determined by the current output termination setting (the APPLy command does not change the termination setting). For example, if you set the amplitude to 10 Vpp and then change the output termination from 50 ohms to "high impedance", the amplitude displayed on the function generator's front-panel will *double* to 20 Vpp (and no error will be generated). If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half. *See the* OUTP:LOAD *command on page 163 for more information.*

- You can set the output amplitude in Vpp, Vrms, or dBm by specifying the units as part of the APPLy command as shown below.

  ```
  APPL:SIN 5.0E+3, 3.0 VRMS, -2.5
  ```

  Or, you can use the VOLT:UNIT command (see page 165) to specify the output units for all commands that follow. Unless you specify the units as part of the APPLy command, the VOLT:UNIT command takes precedence. For example, if you select "Vrms" using the VOLT:UNIT command and *do not* include units with the APPLy command, the value specified for the *amplitude* parameter in the APPLy command will be in "Vrms".

- You cannot specify the output amplitude in dBm if the output termination is currently set to "high impedance". The units are automatically converted to Vpp. *See the* `VOLT:UNIT` *command on page 165 for more information.*
- *Limits Due to Units Selection*: In some cases, the amplitude limits are determined by the output units selected. This may occur when the units are *Vrms* or *dBm* due to the differences in crest factor for the various output functions. For example, if you output a 5 Vrms square wave (into 50 ohms) and then change to the sine wave function, the function generator will automatically adjust the output amplitude to 3.536 Vrms (the upper limit for sine waves in Vrms). *From the remote interface, a "Data out of range" error will be generated and the amplitude will be adjusted as described.*
- *Arbitrary Waveform Limitations*: For arbitrary waveforms, the maximum amplitude will be limited if the waveform data points do not span the full range of the output DAC (Digital- to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is limited to 6.087 Vpp (into 50 ohms).
- While changing amplitude, you may notice a momentary disruption in the output waveform at certain voltages due to switching of the output attenuators. The amplitude is controlled, however, so the output voltage will never exceed the current setting while switching ranges. To prevent this disruption in the output, you can disable the voltage autoranging feature using the `VOLT:RANG:AUTO` command (*see page 160 for more information*). The `APPLy` command automatically enables autoranging.

DC Offset Voltage

- For the *offset* parameter of the `APPLy` command, you can substitute "MINimum", "MAXimum", or "DEFault" in place of a specific value for the parameter. MIN selects the most negative dc offset voltage for the function and amplitude specified. MAX selects the largest dc offset for the function and amplitude specified. *The default offset is 0 volts for all functions.*

- *Limits Due to Amplitude*: The relationship between offset voltage and output amplitude is shown below. V*max* is the maximum *peak voltage* for the selected output termination (5 volts for a 50Ω load or 10 volts for a high-impedance load).

  $$|\,V_{offset}\,| \;\leq\; V_{max} \;-\; \frac{V_{pp}}{2}$$

  If the specified offset voltage is not valid, the function generator will automatically adjust it to the maximum dc voltage allowed with the amplitude specified. *From the remote interface, a "Data out of range" error will be generated and the offset will be adjusted as described.*

- *Limits Due to Output Termination*: The offset limits are determined by the current output termination setting (the `APPLy` command does not change the termination setting). For example, if you set the offset to 100 mVdc and then change the output termination from 50 ohms to "high impedance", the offset voltage displayed on the function generator's front-panel will *double* to 200 mVdc (and no error will be generated). If you change from "high impedance" to 50 ohms, the displayed offset will drop in half. *See the* `OUTP:LOAD` *command on page 163 for more information.*

- *Arbitrary Waveform Limitations*: For arbitrary waveforms, the maximum offset and amplitude will be limited if the waveform data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum offset is limited to 4.95 volts (into 50 ohms). The "0" DAC value is still used as the offset reference even if the waveform data points do not span the full range of the output DAC.

APPLy Command Syntax

- Because of the use of optional parameters in the APPLy commands (enclosed in square brackets), you must specify *frequency* to use the *amplitude* parameter, and you must specify both *frequency* and *amplitude* to use the *offset* parameter. For example, the following command string is valid (*frequency* and *amplitude* are specified but *offset* is omitted and therefore uses a default value).

  ```
  APPL:SIN 5.0E+3, 3.0
  ```

  However, you cannot specify an *amplitude* or *offset* without specifying a *frequency*.

- You can substitute "MINimum", "MAXimum", or "DEFault" in place of specific values for the *frequency*, *amplitude*, and *offset* parameters. For example, the following statement outputs a 3 Vpp sine wave at 80 MHz (the maximum frequency for sine) with a -2.5 volt offset.

  ```
  APPL:SIN MAX, 3.0, -2.5
  ```

- The APPLy command performs the following operations:
  - Sets the trigger source to *Immediate* (equivalent to sending the TRIG:SOUR IMM command).
  - Turns off any modulation, sweep, or burst mode currently enabled and places the instrument in the continuous waveform mode.
  - Turns on the *Output* connector (OUTP ON command) but *does not* change the output termination setting (OUTP:LOAD command).
  - Overrides the voltage autorange setting and automatically enables autoranging (VOLT:RANG:AUTO command).
  - For square waveforms, overrides the current duty cycle setting and automatically selects 50% (FUNC:SQU:DCYC command).
  - For ramp waveforms, overrides the current symmetry setting and automatically selects 100% (FUNC:RAMP:SYMM command).

APPLy:SINusoid [<*frequency*> [,<*amplitude*> [,<*offset*>]]]

Output a sine wave with the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed.

APPLy:SQUare [<*frequency*> [,<*amplitude*> [,<*offset*>]]]

Output a square wave with the specified frequency, amplitude, and dc offset. This command overrides the current duty cycle setting and automatically selects 50%. The waveform is output as soon as the command is executed.

APPLy:RAMP [<*frequency*> [,<*amplitude*> [,<*offset*>]]]

Output a ramp wave with the specified frequency, amplitude, and dc offset. This command overrides the current symmetry setting and automatically selects 100%. The waveform is output as soon as the command is executed.

APPLy:PULSe [<*frequency*> [,<*amplitude*> [,<*offset*>]]]

Output a pulse wave with the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed.

- This command preserves the current pulse width setting (`PULS:WIDT` command) and edge time setting (`PULS:TRAN` command). However, based on the specified frequency, the function generator will adjust the pulse width or edge time to comply with the frequency restrictions for pulse waveforms. *See page 166 for more information on setting the pulse width and edge time.*
- In most applications, the repetition rate for pulse waveforms is specified using a waveform *period* rather than frequency. The `APPLy` command will make approximations in some cases to generate a pulse with the specified frequency. Therefore, the recommended method for setting the repetition rate for pulse waveforms is using the `PULS:PER` command (see page 166).

APPLy:NOISe [<*frequency*|**DEFault**> [**,**<*amplitude*> [**,**<*offset*>]]]

Output Gaussian noise with the specified amplitude and dc offset. The waveform is output as soon as the command is executed.

- The *frequency* parameter has no effect for this command but you must specify a value or "DEFault" (the noise function has a 50 MHz bandwidth). If you specify a frequency, it has no effect on the noise output, but the value is remembered when you change to a different function. The following statement shows the use of the APPLy command for noise.

```
APPL:NOIS DEF, 5.0, 2.0
```

APPLy:DC [<*frequency*|**DEFault**> [**,**<*amplitude*>|**DEFault**> [**,**<*offset*>]]]

Output a dc voltage with the level specified by the *offset* parameter. You can set the dc voltage to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit. The dc voltage is output as soon as the command is executed.

- The *frequency* and *amplitude* parameters have no effect for this command but you must specify a value or "DEFault". If you specify a frequency and amplitude, they have no effect on the dc output, but the values are remembered when you change to a different function. The following statement shows the use of the APPLy command for a dc output.

```
APPL:DC DEF, DEF, -2.5
```

APPLy:USER [<*frequency*> [**,**<*amplitude*> [**,**<*offset*>]]]

Output the arbitrary waveform currently selected by the FUNC:USER command. The waveform is output using the specified frequency, amplitude, and dc offset. The waveform is output as soon as the command is executed. *See page 198 for more information on downloading arbitrary waveforms to memory.*

APPLy?

Query the function generator's current configuration and return a quoted string. The purpose of this command is to allow you to append this query response to an `APPL:` command in your programming application and use the result to place the function generator in the specified state. The function, frequency, amplitude, and offset are returned as shown in the sample string below (the quotation marks are returned as part of the string).

"SIN +5.0000000000000E+03,+3.0000000000000E+00,-2.5000000000000E+00"

Output Configuration Commands

See also "Output Configuration" starting on page 49 in chapter 3.

This section describes the low-level commands used to program the function generator. Although the APPLy command provides the most straightforward method to program the function generator, the low-level commands give you more flexibility to change individual parameters.

FUNCtion {**SINusoid**|**SQUare**|**RAMP**|**PULSe**|**NOISe**|**DC**|**USER**}
FUNCtion?

Select the output function. The selected waveform is output using the previously selected frequency, amplitude, and offset voltage settings. The default is SIN. The FUNC? query returns "SIN", "SQU", "RAMP", "PULS", "NOIS", "DC", or "USER".

- If you select "USER", the function generator outputs the arbitrary waveform currently selected by the FUNC:USER command.
- The table below shows which output functions are allowed with modulation, sweep, and burst. Each " • " indicates a valid combination. If you change to a function that is not allowed with modulation, sweep, or burst, the modulation or mode is turned off.

	Sine	Square	Ramp	Pulse	Noise	DC	User
AM, FM Carrier	•	•	•				•
FSK Carrier	•	•	•				•
Sweep Mode	•	•	•				•
Burst Mode	•	•	•	•	•[1]		•

[1] *Allowed in the External Gated burst mode only.*

- *Function Limitations*: If you change to a function whose maximum frequency is less than that of the current function, the frequency is adjusted to the maximum value for the new function. For example, if you are currently outputting an 80 MHz sine wave and then change to the ramp function, the function generator will automatically adjust the output frequency to 1 MHz (the upper limit for ramps). *From the remote interface, a "Settings conflict" error will be generated and the frequency will be adjusted as described.*
- *Amplitude Limitations*: If you change to a function whose maximum amplitude is less than that of the current function, the amplitude is automatically adjusted to the maximum value for the new function. This may occur when the output units are *Vrms* or *dBm* due to the differences in crest factor for the various output functions.

  For example, if you output a 5 Vrms square wave (into 50 ohms) and then change to the sine wave function, the function generator will automatically adjust the output amplitude to 3.536 Vrms (the upper limit for sine in Vrms). *From the remote interface, a "Settings conflict" error will be generated and the amplitude will be adjusted as described.*

FREQuency {<*frequency*>|**MINimum**|**MAXimum**}
FREQuency? [**MINimum**|**MAXimum**]

Set the output frequency. MIN selects the lowest frequency allowed for the selected function and MAX selects the highest frequency allowed. The default is 1 kHz for all functions. The FREQ? query returns the frequency setting in hertz for the function currently selected.

Function	Minimum Frequency	Maximum Frequency
Sine	1 μHz	80 MHz
Square	1 μHz	80 MHz
Ramp	1 μHz	1 MHz
Pulse	500 μHz	50 MHz
Noise, DC	Not Applicable	Not Applicable
Arbs	1 μHz	25 MHz

- *Function Limitations*: If you change to a function whose maximum frequency is less than that of the current function, the frequency is adjusted to the maximum value for the new function. For example, if you are currently outputting an 80 MHz sine wave and then change to the ramp function, the function generator will automatically adjust the output frequency to 1 MHz (the upper limit for ramps). *From the remote interface, a "Settings conflict" error will be generated and the frequency will be adjusted as described.*

- *Duty Cycle Limitations*: For square waveforms, the function generator may not be able to use the full range of duty cycle values at higher frequencies as shown below.

  20% to 80% (*frequency* ≤ 25 MHz)
  40% to 60% (25 MHz < *frequency* ≤ 50 MHz)
  50% (*frequency* > 50 MHz)

  If you change to a frequency that cannot produce the current duty cycle, the duty cycle is automatically adjusted to the maximum value for the new frequency. For example, if you currently have the duty cycle set to 70% and then change the frequency to 60 MHz, the function generator will automatically adjust the duty cycle to 50% (the upper limit for this frequency). *From the remote interface, a "Settings conflict" error will be generated and the duty cycle will be adjusted as described.*

VOLTage {<*amplitude*>|**MINimum**|**MAXimum**}
VOLTage? [**MINimum**|**MAXimum**]

Set the output amplitude. The default amplitude is 100 mVpp (into 50Ω) for all functions. MIN selects the smallest amplitude (1 mVpp into 50Ω). MAX selects the largest amplitude for the selected function (at most 10 Vpp into 50Ω depending on the selected function and offset voltage). The VOLT? query returns the output amplitude for the function currently selected. Values are always returned in the units set by the most recent VOLT:UNIT command.

- *Offset Voltage Limitations*: The relationship between output amplitude and offset voltage is shown below. V*max* is the maximum *peak voltage* for the selected output termination (5 volts for a 50Ω load or 10 volts for a high-impedance load).

  $$Vpp \leq 2 \times (Vmax - |Voffset|)$$

- *Limits Due to Output Termination*: If you change the output termination setting, the displayed output amplitude will be automatically adjusted (and no error will be generated). For example, if you set the amplitude to 10 Vpp and then change the output termination from 50 ohms to "high impedance", the amplitude displayed on the function generator's front-panel will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half. *For more information, see the* OUTP:LOAD *command on page 163*.

- You can set the output amplitude in Vpp, Vrms, or dBm by specifying the units as part of the VOLT command as shown below.

  ```
  VOLT 3.0 VRMS
  ```

  Or, you can use the VOLT:UNIT command (see page 165) to specify the output units for all commands that follow.

- You cannot specify the output amplitude in dBm if the output termination is currently set to "high impedance". The units are automatically converted to Vpp. *See the* VOLT:UNIT *command on page 165 for more information.*

- *Limits Due to Units Selection*: In some cases, the amplitude limits are determined by the output units selected. This may occur when the units are *Vrms* or *dBm* due to the differences in crest factor for the various output functions. For example, if you output a 5 Vrms square wave (into 50 ohms) and then change to the sine wave function, the function generator will automatically adjust the output amplitude to 3.536 Vrms (the upper limit for sine waves in Vrms). *From the remote interface, a "Settings conflict" error will be generated and the amplitude will be adjusted as described.*
- *Arbitrary Waveform Limitations*: For arbitrary waveforms, the maximum amplitude will be limited if the waveform data points do not span the full range of the output DAC (Digital- to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is limited to 6.087 Vpp (into 50 ohms).
- While changing amplitude, you may notice a momentary disruption in the output waveform at certain voltages due to switching of the output attenuators. The amplitude is controlled, however, so the output voltage will never exceed the current setting while switching ranges. To prevent this disruption in the output, you can disable the voltage autoranging feature using the `VOLT:RANG:AUTO` command (*see page 160 for more information*).
- You can also set the amplitude (with an associated offset voltage) by specifying a high level and low level. For example, if you set the high level to +2 volts and the low level to -3 volts, the resulting amplitude is 5 Vpp (with an associated offset voltage of -500 mV). *See the* `VOLT:HIGH` *and* `VOLT:LOW` *commands on page 159 for more information.*
- To output a *dc voltage level*, select the dc voltage function using the `FUNC DC` command and then set the offset voltage level using the `VOLT:OFFS` command. You can set the dc level to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit.

VOLTage:OFFSet {<*offset*>|**MINimum**|**MAXimum**}
VOLTage:OFFSet? [**MINimum**|**MAXimum**]

Set the dc offset voltage. The default offset is 0 volts for all functions. MIN selects the most negative dc offset voltage for the selected function and amplitude. MAX selects the largest dc offset for the selected function and amplitude. The :OFFS? query returns the offset voltage for the function currently selected.

- *Limits Due to Amplitude*: The relationship between offset voltage and output amplitude is shown below. V*max* is the maximum *peak voltage* for the selected output termination (5 volts for a 50Ω load or 10 volts for a high-impedance load).

$$|V_{offset}| \leq V_{max} - \frac{V_{pp}}{2}$$

  If the specified offset voltage is not valid, the function generator will automatically adjust it to the maximum dc voltage allowed with the amplitude specified. *From the remote interface, a "Data out of range" error will be generated and the offset will be adjusted as described.*

- *Limits Due to Output Termination*: The offset limits are determined by the current output termination setting. For example, if you set the offset to 100 mVdc and then change the output termination from 50 ohms to "high impedance", the offset voltage displayed on the function generator's front-panel will *double* to 200 mVdc (and no error will be generated). If you change from "high impedance" to 50 ohms, the displayed offset will drop in half. *See the* OUTP:LOAD *command on page 163 for more information.*

- *Arbitrary Waveform Limitations*: For arbitrary waveforms, the maximum offset and amplitude will be limited if the waveform data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum offset is limited to 4.95 volts (into 50 ohms).

- You can also set the offset by specifying a high level and low level. For example, if you set the high level to +2 volts and the low level to -3 volts, the resulting amplitude is 5 Vpp (with an associated offset voltage of -500 mV). *See the* `VOLT:HIGH` *and* `VOLT:LOW` *commands below for more information.*
- To output a *dc voltage level*, select the dc voltage function using the `FUNC DC` command and then set the offset voltage level using the `VOLT:OFFS` command. You can set the dc level to any value between ±5 Vdc into 50 ohms or ±10 Vdc into an open circuit.

VOLTage
:HIGH {<*voltage*>|**MINimum**|**MAXimum**}
:HIGH? [**MINimum**|**MAXimum**]
:LOW {<*voltage*>|**MINimum**|**MAXimum**}
:LOW? [**MINimum**|**MAXimum**]

Set the high or low voltage levels. For all functions, the default *high level* is +50 mV and the default *low level* is -50 mV. MIN selects the most negative voltage level for the selected function and MAX selects the largest voltage level. The `:HIGH?` and `:LOW?` queries return the high and low levels, respectively.

- *Limits Due to Amplitude*: You can set the voltage levels to a positive or negative value with the restrictions shown below. V*pp* is the maximum peak-to-peak amplitude for the selected output termination (10 Vpp for a 50-ohm load or 20 Vpp for a high-impedance load).

$$V_{high} - V_{low} \leq V_{pp}\,(\max) \quad \textbf{and} \quad V_{high}, V_{low} \leq \frac{V_{pp}\,(\max)}{2}$$

  If the specified level is not valid, the function generator will automatically adjust it to the maximum voltage allowed. *From the remote interface, a "Data out of range" error will be generated and the level will be adjusted as described.*
- You can set the levels to a positive or negative value, but note that the high level *must* always be greater than the low level. If you specify a low level that is greater than the high level, the function generator will automatically set the low level to be 1 mV less than the high level. *From the remote interface, a "Data out of range" error will be generated and the level will be adjusted as described.*

- Note that when you set the high and low levels, you are also setting the amplitude of the waveform. For example, if you set the high level to +2 volts and the low level to -3 volts, the resulting amplitude is 5 Vpp (with an offset voltage of -500 mV).
- *Limits Due to Output Termination*: If you change the output termination setting, the displayed voltage levels will be automatically adjusted (and no error will be generated). For example, if you set the high level to +100 mVdc and then change the output termination from 50 ohms to "high impedance", the voltage displayed on the function generator's front-panel will *double* to +200 mVdc. If you change from "high impedance" to 50 ohms, the displayed voltage will drop in half. *See the* `OUTP:LOAD` *command on page 163 for more information.*
- To invert the waveform relative to the offset voltage, you can use the `OUTP:POL` command. *See page 164 for more information.*

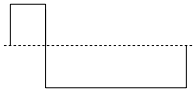
`VOLTage:RANGe:AUTO` {**`OFF`**|**`ON`**|**`ONCE`**}
`VOLTage:RANGe:AUTO?`

Disable or enable voltage autoranging for all functions. In the default mode, autoranging is enabled ("ON") and the function generator automatically selects the optimal settings for the output amplifier and attenuators. With autoranging disabled ("OFF"), the function generator uses the current amplifier and attenuator settings. The `:AUTO?` query returns "0" (OFF) or "1" (ON).

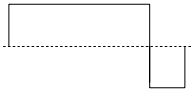
- The `APPLy` command overrides the voltage autorange setting and automatically enables autoranging ("ON").
- The advantage of disabling autoranging is to eliminate momentary disruptions caused by switching of the attenuators while changing amplitude. However, the amplitude and offset accuracy and resolution (as well as waveform fidelity) may be adversely affected when reducing the amplitude below the expected range change.
- The "ONCE" parameter has the same effect as turning autoranging "ON", and then "OFF". This parameter allows you to make a one-time change to the amplifier/attenuator setting before returning to the `VOLT:RANG:AUTO OFF` setting.

FUNCtion:SQUare:DCYCle {<*percent*>|**MINimum**|**MAXimum**}
FUNCtion:SQUare:DCYCle? [**MINimum**|**MAXimum**]

Set the duty cycle percentage for *square waves*. Duty cycle represents the amount of time per cycle that the square wave is at a *high level* (assuming that the waveform polarity is not inverted). The default is 50%. MIN selects the minimum duty cycle for the selected frequency and MAX selects the maximum duty cycle (see restrictions below). The :DCYC? query returns the current duty cycle setting in percent.



20% Duty Cycle

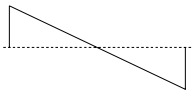


80% Duty Cycle

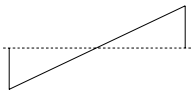
- Duty Cycle: 20% to 80% (*frequency* ≤ 25 MHz)
  40% to 60% (25 MHz < *frequency* ≤ 50 MHz)
  50% (*frequency* > 50 MHz)
- For square waveforms, the APPLy command overrides the current duty cycle setting and automatically selects 50%.
- The duty cycle setting is remembered when you change from square wave to another function. When you return to the square wave function, the previous duty cycle is used.
- *Limits Due to Frequency*: If the square wave function is selected and you change to a frequency that cannot produce the current duty cycle, the duty cycle is automatically adjusted to the maximum value for the new frequency. For example, if you currently have the duty cycle set to 70% and then change the frequency to 60 MHz, the function generator will automatically adjust the duty cycle to 50% (the upper limit for this frequency). *From the remote interface, a "Settings conflict" error will be generated and the duty cycle will be adjusted as described.*
- If you select a square waveform as the *modulating* waveform for AM or FM, the duty cycle setting *does not* apply. The function generator always uses a square waveform with a 50% duty cycle.

FUNCtion:RAMP:SYMMetry {<*percent*>|**MINimum**|**MAXimum**}
FUNCtion:RAMP:SYMMetry? [**MINimum**|**MAXimum**]

Set the symmetry percentage for *ramp waves*. Symmetry represents the amount of time per cycle that the ramp wave is *rising* (assuming that the waveform polarity is not inverted). You can set the symmetry to any value from 0% to 100%. The default is 100%. MIN = 0%. MAX = 100%. The :SYMM? query returns the current symmetry setting in percent.



0% Symmetry



100% Symmetry

- For ramp waveforms, the APPLy command overrides the current symmetry setting and automatically selects 100%.
- The symmetry setting is remembered when you change from ramp wave to another function. When you return to the ramp wave function, the previous symmetry is used.
- If you select a ramp waveform as the *modulating* waveform for AM or FM, the symmetry setting *does not* apply.

OUTPut {**OFF**|**ON**}
OUTPut?

Disable or enable the front-panel *Output* connector. The default is "OFF". When the output is enabled, the (Output) key is illuminated on the function generator's front panel. The OUTP? query returns "0" (OFF) or "1" (ON).

- The APPLy command overrides the current OUTP command setting and automatically enables the *Output* connector ("ON").
- If an excessive external voltage is applied to the front-panel *Output* connector, an error message will be displayed and the output will be disabled. To re-enable the output, remove the overload from the *Output* connector and send the OUTP ON command.

OUTPut:LOAD {<*ohms*>|**INFinity**|**MINimum**|**MAXimum**}
OUTPut:LOAD? [**MINimum**|**MAXimum**]

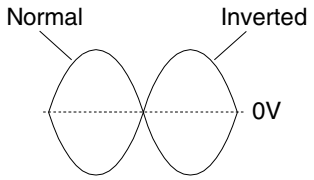
Select the desired output termination (i.e., the impedance of the load attached to the output of the Agilent 33250A). The specified value is used for amplitude, offset, and high/low level settings. You can set the load to any value from 1Ω to 10 kΩ. MIN selects 1Ω. MAX selects 10 kΩ. INF sets the output termination to "high impedance" (>10 kΩ). The default is 50Ω. The :LOAD? query returns the current load setting in ohms or "9.9E+37" (for "high impedance").

- The Agilent 33250A has a fixed series output impedance of 50 ohms to the front-panel *Output* connector. If the actual load impedance is different than the value specified, the displayed amplitude, offset, and high/low levels will be incorrect.
- If you change the output termination setting, the displayed output amplitude, offset, and high/low levels are automatically adjusted (and no error will be generated). For example, if you set the amplitude to 10 Vpp and then change the output termination from 50 ohms to "high impedance", the amplitude displayed on the function generator's front-panel will *double* to 20 Vpp. If you change from "high impedance" to 50 ohms, the displayed amplitude will drop in half.
- You cannot specify the output amplitude in dBm if the output termination is currently set to "high impedance". The units are automatically converted to Vpp. *See the* VOLT:UNIT *command on page 165 for more information.*

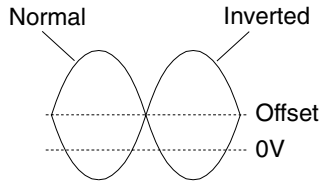
OUTPut:POLarity {**NORMal**|**INVerted**}
OUTPut:POLarity?

Invert the waveform relative to the offset voltage. In the *normal* mode (default), the waveform goes positive during the first part of the cycle. In the *inverted* mode, the waveform goes negative during the first part of the cycle. The :POL? query returns "NORM" or "INV".

- As shown in the examples below, the waveform is inverted *relative to* the offset voltage. Any offset voltage present will remain unchanged when the waveform is inverted.



No Offset Voltage



With Offset Voltage

- When a waveform is inverted, the Sync signal associated with the waveform *is not* inverted.

OUTPut:SYNC {**OFF**|**ON**}
OUTPut:SYNC?

Disable or enable the front-panel *Sync* connector. At lower amplitudes, you can reduce output distortion by disabling the Sync signal. The default setting is "ON". The :SYNC? query returns "0" (OFF) or "1" (ON).

- For more details on the Sync signal for each waveform function, see "Sync Output Signal" on page 62.
- When the Sync signal is disabled, the output level on the *Sync* connector is a logic "low" level.
- When a waveform is inverted (OUTP:POL command), the Sync signal associated with the waveform *is not* inverted.
- The OUTP:SYNC command is overridden by the setting of the MARK command used with the sweep mode (see page 186). Therefore, when the marker frequency is enabled (and the sweep mode is also enabled), the OUTP:SYNC command is ignored.

VOLTage:UNIT {**VPP**|**VRMS**|**DBM**}
VOLTage:UNIT?

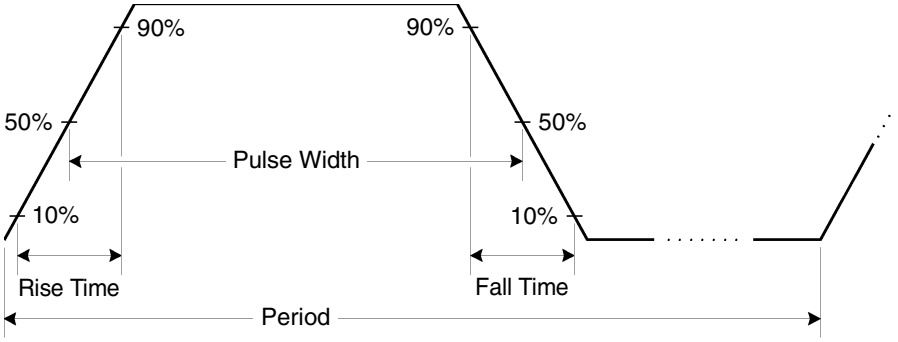
Select the units for output amplitude (does not affect offset voltage or high/low levels). The default is VPP. The :UNIT? query returns "VPP", "VRMS", or "DBM".

- The function generator uses the current units selection for both front panel and remote interface operations. For example, if you select "VRMS" from the remote interface using the VOLT:UNIT command, the units are displayed as "VRMS" on the front panel.
- The VOLT? query command (see page 156) returns the output amplitude in the units set by the most recent VOLT:UNIT command.
- The output units for amplitude cannot be set to dBm if the output termination is currently set to "high impedance". The units are automatically converted to Vpp. *See the* OUTP:LOAD *command on page 163 for more information.*
- Unless you specify the units as part of the APPLy or VOLT command, the VOLT:UNIT command takes precedence. For example, if you select "Vrms" using the VOLT:UNIT command and *do not* include units with the APPLy or VOLT command, the value specified for the *amplitude* parameter in the APPLy command will be in "Vrms".

Pulse Configuration Commands

See also "Pulse Waveforms" starting on page 64 in chapter 3.

This section describes the low-level commands used to program the function generator to output a pulse waveform. To select the pulse function, use the `FUNC PULS` command (see page 153). Refer to the diagram below for the command descriptions that follow.



`PULSe:PERiod` {<*seconds*>|**`MINimum`**|**`MAXimum`**}
`PULSe:PERiod?` [**`MINimum`**|**`MAXimum`**]

Set the period for pulses. Select a period from 20 ns to 2000 seconds. The default is 1 ms. MIN = 20 ns. MAX = 2000 seconds. The `:PER?` query returns the period of the pulse waveform in seconds.

- The specified period must be greater than the sum of the *pulse width* and the *edge time* as shown below. The function generator will adjust the pulse width and edge time as needed to accommodate the specified period. *From the remote interface, a "Data out of range" error will be generated and the period will be adjusted as described.*

$$\text{Period} \geq \text{Pulse Width} + (1.6 \times \text{Edge Time})$$

- This command affects the period (and frequency) for *all* waveform functions (not just pulse). For example, if you select a period using the `PULS:PER` command and then change the function to sine wave, the specified period will be used for the new function.
- *Function Limitations*: If you change to a function whose minimum period is greater than that for a pulse waveform, the period is adjusted to the minimum value allowed for the new function. For example, if you output a pulse waveform with a period of 50 ns and then change to the ramp function, the function generator will automatically adjust the period to 1 µs (the lower limit for ramps). *From the remote interface, a "Settings conflict" error will be generated and the period will be adjusted as described.*

PULSe:WIDTh {<*seconds*>|**MINimum**|**MAXimum**}
PULSe:WIDTh? [**MINimum**|**MAXimum**]

Set the pulse width in seconds. The pulse width represents the time from the 50% threshold of the rising edge of the pulse to the 50% threshold of the next falling edge. You can vary the pulse width from 8 ns to 2000 seconds (*see restrictions below*). The default pulse width is 100 µs. MIN = 8 ns. MAX = 2000 seconds. The `:WIDT?` query returns the pulse width in seconds.

- The minimum pulse width is affected by the period.

  For period > 20 seconds, minimum pulse width = 1 µs
  For period > 200 seconds, minimum pulse width = 10 µs

- The specified pulse width must be less than the difference between the *period* and the *edge time* as shown below. The function generator will adjust the pulse width as needed to accommodate the specified period. *From the remote interface, a "Settings conflict" error will be generated and the pulse width will be adjusted as described.*

  $$\text{Pulse Width} \leq \text{Period} - (1.6 \times \text{Edge Time})$$

- The pulse width must also be greater than the total time of one edge as shown below.

  $$\text{Pulse Width} \geq 1.6 \times \text{Edge Time}$$

PULSe:TRANsition {<*seconds*>|**MINimum**|**MAXimum**}
PULSe:TRANsition? [**MINimum**|**MAXimum**]

Set the edge time in seconds for *both* the rising and falling edges. The edge time represents the time from the 10% threshold to the 90% threshold of each edge. You can vary the edge time from 5 ns to 1 ms (*see restrictions below*). The default edge time is 5 ns. MIN = 5 ns. MAX = 1 ms. The :TRAN? query returns the edge time in seconds.

- The specified edge time must fit within the specified pulse width as shown below. The function generator will adjust the edge time as needed to accommodate the specified pulse width. *From the remote interface, a "Data out of range" error will be generated and the edge time will be adjusted as described.*

  Edge Time $\leq$ 0.625 X Pulse Width

Amplitude Modulation (AM) Commands

See also "Amplitude Modulation" starting on page 67 in chapter 3.

AM Overview

The following is an overview of the steps required to generate an AM waveform. The commands used for AM are listed on the next page.

1 Configure the carrier waveform.

Use the `APPLy` command or the equivalent `FUNC`, `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the function, frequency, amplitude, and offset of the carrier waveform. You can select a sine, square, ramp, or arbitrary waveform for the carrier (pulse, noise, and dc are not allowed).

2 Select the modulation source.

The function generator will accept an internal or external modulation source. Select the modulation source using the `AM:SOUR` command. *For an external source, you can skip steps 3 and 4 below.*

3 Select the shape of the modulating waveform.

You can modulate the carrier with a sine, square, ramp, noise, or arbitrary waveform (pulse and dc are not allowed). Use the `AM:INT:FUNC` command to select the modulating waveshape.

4 Set the modulating frequency.

Set the modulating frequency to any value from 2 mHz to 20 kHz using the `AM:INT:FREQ` command.

5 Set the modulation depth.

Set the modulation depth (also called "percent modulation") to any value from 0% to 120% using the `AM:DEPT` command.

6 Enable AM modulation.

After you have set up the other modulation parameters, use the `AM:STAT ON` command to enable AM.

AM Commands

> Use the APPLy command or the equivalent FUNC, FREQ, VOLT, and VOLT:OFFS commands to configure the carrier waveform.

AM:SOURce {**INTernal**|**EXTernal**}
AM:SOURce?

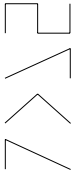
Select the source of the modulating signal. The function generator will accept an internal or external modulation source. The default is INT. The :SOUR? query returns "INT" or "EXT".

- If you select the *External* source, the carrier waveform is modulated with an external waveform. The modulation depth is controlled by the ±5V signal level present on the rear-panel *Modulation In* connector. For example, if you have set the modulation depth to 100% using the AM:DEPT command, then when the modulating signal is at +5 volts, the output will be at the *maximum* amplitude. When the modulating signal is at -5 volts, then the output will be at the *minimum* amplitude.

AM:INTernal
:FUNCtion {**SINusoid**|**SQUare**|**RAMP**|**NRAMp**|**TRIangle**|**NOISe**|**USER**}
:FUNCtion?

Select the shape of the *modulating* waveform. Used only when the *Internal* modulation source is selected (AM:SOUR INT command). You can use noise as the modulating waveshape, but you *cannot* use noise, pulse, or dc as the carrier waveform. The default is SIN. The :FUNC? query returns "SIN", "SQU", "RAMP", "NRAM", "TRI", "NOIS", or "USER".

- Select "SQU" for a square waveform with a *50% duty cycle*.
- Select "RAMP" for a ramp waveform with *100% symmetry*.
- Select "TRI" for a ramp waveform with *50% symmetry*.
- Select "NRAM" (negative ramp) for a ramp waveform with *0% symmetry*.



- If you select an arbitrary waveform as the *modulating* waveshape ("USER"), the waveform is automatically limited to 8K points. Extra waveform points are removed using decimation.

AM:INTernal:FREQuency {<*frequency*>|**MINimum**|**MAXimum**}
AM:INTernal:FREQuency? [**MINimum**|**MAXimum**]

Set the frequency of the *modulating* waveform. Used only when the *Internal* modulation source is selected (`AM:SOUR INT` command). Select from 2 mHz to 20 kHz. The default is 100 Hz. MIN = 2 mHz. MAX = 20 kHz. The `:FREQ?` query returns the internal modulating frequency in hertz.

AM:DEPTh {<*depth in percent*>|**MINimum**|**MAXimum**}
AM:DEPTh? [**MINimum**|**MAXimum**]

Set the internal modulation depth (or "percent modulation") in percent. Select from 0% to 120%. The default is 100%. MIN = 0%. MAX = 120%. The `:DEPT?` query returns the modulation depth in percent.

- Note that even at greater than 100% depth, the function generator will not exceed ±5V peak on the output (into a 50Ω load).
- If you select the *External* modulating source (`AM:SOUR EXT` command), the carrier waveform is modulated with an external waveform. The modulation depth is controlled by the ±5V signal level present on the rear-panel *Modulation In* connector. For example, if you have set the modulation depth to 100% using the `AM:DEPT` command, then when the modulating signal is at +5 volts, the output will be at the *maximum* amplitude. When the modulating signal is at -5 volts, then the output will be at the *minimum* amplitude.

AM:STATe {**OFF**|**ON**}
AM:STATe?

Disable or enable AM. To avoid multiple waveform changes, you can enable AM *after* you have set up the other modulation parameters. The default is OFF. The `:STAT?` query returns "0" (OFF) or "1" (ON).

- The function generator will allow only one modulation mode to be enabled at a time. For example, you cannot enable AM and FM at the same time. When you enable AM, the previous modulation mode is turned off.
- The function generator will not allow AM to be enabled at the same time that sweep or burst is enabled. When you enable AM, the sweep or burst mode is turned off.

Frequency Modulation (FM) Commands

See also "Frequency Modulation" starting on page 72 in chapter 3.

FM Overview

The following is an overview of the steps required to generate an FM waveform. The commands used for FM are listed on the next page.

1 Configure the carrier waveform.

Use the `APPLy` command or the equivalent `FUNC`, `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the function, frequency, amplitude, and offset of the carrier waveform. You can select a sine, square, ramp, or arbitrary waveform for the carrier (pulse, noise, and dc are not allowed).

2 Select the modulation source.

The function generator will accept an internal or external modulation source. Select the modulation source using the `FM:SOUR` command. *For an external source, you can skip steps 3 and 4 below.*

3 Select the shape of the modulating waveform.

You can modulate the carrier with a sine, square, ramp, noise, or arbitrary waveform (pulse and dc are not allowed). Use the `FM:INT:FUNC` command to select the modulating waveshape.

4 Set the modulating frequency.

Set the modulating frequency to any value from 2 mHz to 20 kHz using the `FM:INT:FREQ` command.

5 Set the peak frequency deviation.

Set the frequency deviation to any value from 5 Hz to 40.05 MHz (limited to 550 kHz for ramps and 12.55 MHz for arbitrary waveforms) using the `FM:DEV` command.

6 Enable FM modulation.

After you have set up the other modulation parameters, use the `FM:STAT ON` command to enable FM.

FM Commands

Use the APPLy command or the equivalent FUNC, FREQ, VOLT, and VOLT:OFFS commands to configure the carrier waveform.

FM:SOURce {**INTernal**|**EXTernal**}
FM:SOURce?

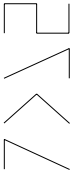
Select the source of the modulating signal. The function generator will accept an internal or external modulation source. The default is INT. The :SOUR? query returns "INT" or "EXT".

- If you select the *External* source, the carrier waveform is modulated with an external waveform. The frequency deviation is controlled by the ±5V signal level present on the rear-panel *Modulation In* connector. For example, if you have set the deviation to 100 kHz using the FM:DEV command, then a +5V signal level corresponds to a 100 kHz increase in frequency. Lower external signal levels produce less deviation and negative signal levels reduce the frequency below the carrier frequency.

FM:INTernal
:FUNCtion {**SINusoid**|**SQUare**|**RAMP**|**NRAMp**|**TRIangle**|**NOISe**|**USER**}
:FUNCtion?

Select the shape of the *modulating* waveform. Used only when the *Internal* modulation source is selected (FM:SOUR INT command). You can use noise as the modulating waveshape, but you *cannot* use noise, pulse, or dc as the carrier waveform. The default is SIN. The :FUNC? query returns "SIN", "SQU", "RAMP", "NRAM", "TRI", "NOIS", or "USER".

- Select "SQU" for a square waveform with a *50% duty cycle*.
- Select "RAMP" for a ramp waveform with *100% symmetry*.
- Select "TRI" for a ramp waveform with *50% symmetry*.
- Select "NRAM" (negative ramp) for a ramp waveform with *0% symmetry*.



- If you select an arbitrary waveform as the *modulating* waveshape ("USER"), the waveform is automatically limited to 8K points. Extra waveform points are removed using decimation.

FM:INTernal:FREQuency {<*frequency*>|**MINimum**|**MAXimum**}
FM:INTernal:FREQuency? [**MINimum**|**MAXimum**]

Set the frequency of the *modulating* waveform. Used only when the *Internal* modulation source is selected (`FM:SOUR INT` command). Select from 2 mHz to 20 kHz. The default is 10 Hz. MIN = 2 mHz. MAX = 20 kHz. The `:FREQ?` query returns the internal modulating frequency in hertz.

FM:DEViation {<*peak deviation in Hz*>|**MINimum**|**MAXimum**}
FM:DEViation? [**MINimum**|**MAXimum**]

Set the peak frequency deviation in hertz. This value represents the variation in frequency of the *modulating* waveform from the carrier frequency. Select any value from 5 Hz to 40.05 MHz (limited to 550 kHz for ramps and 12.55 MHz for arbitrary waveforms). The default is 100 Hz. MIN = 5 Hz. MAX = based on the frequency of the carrier waveform as shown below. The `:DEV?` query returns the deviation in hertz.

$$\text{Max. Deviation} = \frac{\text{Carrier}}{2} \qquad \textit{For Carrier} < 40\ \textit{MHz}$$

$$\text{Max. Deviation} = \frac{\text{Max. Frequency} - \text{Carrier}}{2} \qquad \textit{For Carrier} > 40\ \textit{MHz}$$

- The *carrier frequency* must always be greater than or equal to the deviation. If you attempt to set the deviation to a value greater than the carrier frequency (with FM enabled), the function generator will automatically adjust the deviation to the maximum value allowed with the present carrier frequency. *From the remote interface, a "Data out of range" error will be generated and the deviation will be adjusted as described.*
- The sum of the *carrier frequency* and deviation must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (80.1 MHz for sine and square, 1.1 MHz for ramp, and 25.1 MHz for arbitrary waveforms). If you attempt to set the deviation to a value that is not valid, the function generator will automatically adjust it to the maximum value allowed with the present carrier frequency. *From the remote interface, a "Data out of range" error will be generated and the deviation will be adjusted as described.*

- If the deviation causes the carrier waveform to exceed a frequency boundary for the current duty cycle (square waveform only), the function generator will automatically adjust the duty cycle to the maximum value allowed with the present carrier frequency. *From the remote interface, a "Settings conflict" error will be generated and the duty cycle will be adjusted as described.*
- If you select the *External* modulating source (`FM:SOUR EXT` command), the deviation is controlled by the ±5V signal level present on the rear-panel *Modulation In* connector. For example, if you have set the frequency deviation to 100 kHz, then a +5V signal level corresponds to a 100 kHz increase in frequency. Lower external signal levels produce less deviation and negative signal levels reduce the frequency below the carrier frequency.

`FM:STATe` {**`OFF`**|**`ON`**}
`FM:STATe?`

Disable or enable FM. To avoid multiple waveform changes, you can enable FM *after* you have set up the other modulation parameters. The default is OFF. The `:STAT?` query returns "0" (OFF) or "1" (ON).

- The function generator will allow only one modulation mode to be enabled at a time. For example, you cannot enable FM and AM at the same time. When you enable FM, the previous modulation mode is turned off.
- The function generator will not allow FM to be enabled at the same time that sweep or burst is enabled. When you enable FM, the sweep or burst mode is turned off.

Frequency-Shift Keying (FSK) Commands

See also "FSK Modulation" starting on page 78 in chapter 3.

FSK Overview

The following is an overview of the steps required to generate an FSK modulated waveform. The commands used for FSK are listed on the next page.

1 Configure the carrier waveform.

Use the `APPLy` command or the equivalent `FUNC`, `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the function, frequency, amplitude, and offset of the carrier waveform. You can select a sine, square, ramp, or arbitrary waveform for the carrier (pulse, noise, and dc are not allowed).

2 Select the FSK source.

The function generator will accept an internal or external FSK source. Select the FSK source using the `FSK:SOUR` command.

3 Select the FSK "hop" frequency.

Set the alternate (or "hop") frequency to any value from 1 μHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms) using the `FSK:FREQ` command.

4 Set the FSK rate.

Set the FSK rate to any value from 2 mHz to 100 kHz using the `FSK:INT:RATE` command (internal FSK source only). The FSK rate sets the pace at which the output frequency "shifts" between the carrier frequency and the hop frequency.

5 Enable FSK modulation.

After you have set up the other FSK parameters, use the `FSK:STAT ON` command to enable FSK modulation.

FSK Commands

> Use the APPLy command or the equivalent FUNC, FREQ, VOLT, and VOLT:OFFS commands to configure the carrier waveform.

`FSKey:SOURce` {**`INTernal`**|**`EXTernal`**}
`FSKey:SOURce?`

Select an internal or external FSK source. The default is INT. The `:SOUR?` query returns "INT" or "EXT".

- When the *Internal* source is selected, the rate at which the output frequency "shifts" between the carrier frequency and hop frequency is determined by the *FSK rate* specified (`FSK:INT:RATE` command).
- When the *External* source is selected, the output frequency is determined by the signal level on the rear-panel *Trig In* connector. When a logic *low* level is present, the *carrier* frequency is output. When a logic *high* level is present, the *hop* frequency is output.
- The maximum external FSK rate is 1 MHz.
- Note that the connector used for externally-controlled FSK waveforms (*Trig In*) is not the same connector that is used for externally-modulated AM and FM waveforms (*Modulation In*). When used for FSK, the *Trig In* connector *does not* have adjustable edge polarity and is not affected by the `TRIG:SLOP` command.

FSKey:FREQuency {<*frequency*>|**MINimum**|**MAXimum**}
FSKey:FREQuency? [**MINimum**|**MAXimum**]

Set the FSK alternate (or "hop") frequency. Select from 1 μHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). The default is 100 Hz. MIN = 1 μHz. MAX = 80 MHz. The `:FREQ?` query returns the "hop" frequency in hertz.

- The modulating waveform is a *square wave* with a 50% duty cycle.

FSKey:INTernal:RATE {<*rate in Hz*>|**MINimum**|**MAXimum**}
FSKey:INTernal:RATE? [**MINimum**|**MAXimum**]

Set the rate at which the output frequency "shifts" between the carrier and hop frequency. Select from 2 mHz to 100 kHz. The default is 10 Hz. MIN = 2 mHz. MAX = 100 kHz. The `:RATE?` query returns the FSK rate in hertz.

- The FSK rate is only used when the *Internal* source is selected (`FSK:SOUR INT` command) and is ignored when the external source is selected (`FSK:SOUR EXT` command).

FSKey:STATe {**OFF**|**ON**}
FSKey:STATe?

Disable or enable FSK modulation. To avoid multiple waveform changes, you can enable FSK *after* you have set up the other modulation parameters. The default is OFF. The `:STAT?` query returns "0" (OFF) or "1" (ON).

- The function generator will allow only one modulation mode to be enabled at a time. For example, you cannot enable FSK and AM at the same time. When you enable FSK, the previous modulation mode is turned off.
- The function generator will not allow FSK to be enabled at the same time that sweep or burst is enabled. When you enable FSK, the sweep or burst mode is turned off.

Frequency Sweep Commands

See also "Frequency Sweep" starting on page 82 in chapter 3.

Sweep Overview

The following is an overview of the steps required to generate a sweep. The commands used for sweep are listed on page 181.

1 Select the waveform shape, amplitude, and offset.

Use the `APPLy` command or the equivalent `FUNC`, `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the function, frequency, amplitude, and offset. You can select a sine, square, ramp, or arbitrary waveform (pulse, noise, and dc are not allowed).

2 Select the frequency boundaries of the sweep.

You can set the frequency boundaries using one of two methods.

a *Start Frequency*/*Stop Frequency*: Use the `FREQ:STAR` command to set the start frequency and the `FREQ:STOP` command to set the stop frequency.

To sweep ***up*** in frequency, set the start frequency **<** stop frequency.
To sweep ***down*** in frequency, set the start frequency **>** stop frequency.

b *Center Frequency*/*Frequency Span*: Use the `FREQ:CENT` command to set the center frequency and the `FREQ:SPAN` command to set the frequency span.

To sweep ***up*** in frequency, set a *positive* span.
To sweep ***down*** in frequency, set a *negative* span.

3 Select the sweep mode.

Use the `SWE:SPAC` command to select linear or logarithmic spacing for the sweep.

4 Set the sweep time.

Use the `SWE:TIME` command to set the number of seconds required to sweep from the start frequency to the stop frequency.

5 Select the sweep trigger source.

Use the `TRIG:SOUR` command to select the source from which the sweep will be triggered.

6 Set the marker frequency. (*Optional*)

If desired, you can set the frequency at which the signal on the front-panel *Sync* connector goes to a logic low during the sweep. Use the `MARK:FREQ` command to set the marker frequency to any value between the start frequency and stop frequency. Use the `MARK ON` command to enable the frequency marker.

7 Enable the sweep mode.

After you have set up the other sweep parameters, use the `SWE:STAT ON` command to enable the sweep mode.

Sweep Commands

FREQuency:STARt {<*frequency*>|**MINimum**|**MAXimum**}
FREQuency:STARt? [**MINimum**|**MAXimum**]

Set the start frequency (used in conjunction with the *stop frequency*). Select from 1 μHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). The default is 100 Hz. MIN = 1 μHz. MAX = 80 MHz. The `:STAR?` query returns the start frequency in hertz.

- To sweep ***up*** in frequency, set the start frequency **<** stop frequency. To sweep ***down*** in frequency, set the start frequency **>** stop frequency.

FREQuency:STOP {<*frequency*>|**MINimum**|**MAXimum**}
FREQuency:STOP? [**MINimum**|**MAXimum**]

Set the stop frequency (used in conjunction with the *start frequency*). Select from 1 μHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). The default is 1 kHz. MIN = 1 μHz. MAX = 80 MHz. The `:STOP?` query returns the stop frequency in hertz.

FREQuency:CENTer {<*frequency*>|**MINimum**|**MAXimum**}
FREQuency:CENTer? [**MINimum**|**MAXimum**]

Set the center frequency (used in conjunction with the *frequency span*). Select from 1 µHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). The default is 550 Hz. MIN = 1 µHz. MAX = based on the frequency span and maximum frequency for the selected function as shown below. The :CENT? query returns the center frequency in hertz.

$$\text{Center Frequency (max)} = \text{Max. Frequency} - \frac{\text{Span}}{2}$$

- The following equation shows the relationship between the center frequency and the start/stop frequency.

$$\text{Center Frequency} = \frac{|\ \text{Stop Frequency} - \text{Start Frequency}\ |}{2}$$

FREQuency:SPAN {<*frequency*>|**MINimum**|**MAXimum**}
FREQuency:SPAN? [**MINimum**|**MAXimum**]

Set the frequency span (used in conjunction with the *center frequency*). Select from 0 Hz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). The default is 900 Hz. MIN = 0 Hz. MAX = based on the center frequency and maximum frequency for the selected function. The :SPAN? query returns the span in hertz (can be a positive or negative value).

$$\text{Frequency Span (max)} = 2 \times (\text{Max. Frequency} - \text{Center Frequency})$$

- To sweep ***up*** in frequency, set a *positive* frequency span.
  To sweep ***down*** in frequency, set a *negative* frequency span.
- The following equation shows the relationship between the span and the start/stop frequency.

$$\text{Frequency Span} = \text{Stop Frequency} - \text{Start Frequency}$$

SWEep:SPACing {**LINear**|**LOGarithmic**}
SWEep:SPACing?

Select linear or logarithmic spacing for the sweep. The default is Linear. The :SPAC? query returns "LIN" or "LOG".

- For a *linear* sweep, the function generator varies the output frequency in a linear fashion during the sweep.
- For a *logarithmic* sweep, the function generator varies the output frequency in a logarithmic fashion during the sweep.

SWEep:TIME {<*seconds*>|**MINimum**|**MAXimum**}
SWEep:TIME? [**MINimum**|**MAXimum**]

Set the number of seconds required to sweep from the start frequency to the stop frequency. Select from 1 ms to 500 seconds. The default is 1 second. MIN = 1 ms. MAX = 500 seconds. The :TIME? query returns the sweep time in seconds.

- The number of discrete frequency points in the sweep is automatically calculated by the function generator and is based on the sweep time that you select.

SWEep:STATe {**OFF**|**ON**}
SWEep:STATe?

Disable or enable the sweep mode. To avoid multiple waveform changes, you can enable the sweep mode *after* you have set up the other sweep parameters. The default is OFF. The :STAT? query returns "0" (OFF) or "1" (ON).

- The function generator will not allow the sweep mode to be enabled at the same time that burst or any modulation mode is enabled. When you enable sweep, the burst or modulation mode is turned off.

TRIGger:SOURce {**IMMediate**|**EXTernal**|**BUS**}
TRIGger:SOURce?

Select the source from which the function generator will accept a trigger. The function generator will accept an immediate internal trigger, a hardware trigger from the rear-panel *Trig In* connector, or a software (bus) trigger. The default is IMM. The `:SOUR?` query returns "IMM", "EXT", or "BUS"

- When the *Immediate* (internal) source is selected, the function generator outputs a continuous sweep at a rate determined by the sweep time specified (`SWE:TIME` command).
- When the *External* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Trig In* connector. The function generator initiates one sweep each time *Trig In* receives a TTL pulse with the edge polarity specified by the `TRIG:SLOP` command (see page 185). Note that the trigger period must be greater than or equal to the specified sweep time ***plus*** 1 ms.
- When the *Bus* (software) source is selected, the function generator initiates one sweep each time a bus trigger command is received. To trigger the function generator from the remote interface (either GPIB or RS-232), send the `TRIG` or `*TRG` (trigger) command. The front-panel (Trigger) key is illuminated when the function generator is waiting for a bus trigger.
- The `APPLy` command automatically sets the trigger source to *Immediate* (equivalent to `TRIG:SOUR IMM` command).
- To ensure synchronization when the *Bus* source is selected, send the `*WAI` (wait) command. When the `*WAI` command is executed, the function generator waits for all pending operations to complete before executing any additional commands. For example, the following command string guarantees that the first trigger is accepted and the operation is executed before the second trigger is recognized.

```
TRIG:SOUR BUS;*TRG;*WAI;*TRG;*WAI
```

- You can use the `*OPC?` (operation complete query) command or the `*OPC` (operation complete) command to signal when the sweep is complete. The `*OPC?` command returns "1" to the output buffer when the sweep is complete. The `*OPC` command sets the "Operation Complete" bit (bit 0) in the Standard Event register when the sweep is complete.

TRIGger:SLOPe {**POSitive**|**NEGative**}
TRIGger:SLOPe?

Select whether the function generator uses the rising edge or falling edge of the trigger signal on the rear-panel *Trig In* connector for an externally-triggered sweep. The default is POS (rising edge). The :SLOP? query returns "POS" or "NEG".

OUTPut:TRIGger:SLOPe {**POSitive**|**NEGative**}
OUTPut:TRIGger:SLOPe?

Select a rising or falling edge for the "trigger out" signal. When enabled using the OUTP:TRIG command (see below), a TTL-compatible square waveform with the specified edge is output from the rear-panel *Trig Out* connector at the beginning of the sweep. Select "POS" to output a pulse with a rising edge or select "NEG" to output a pulse with a falling edge. The default is POS. The :SLOP? query returns "POS" or "NEG".

- When the *Immediate* (internal) trigger source is selected (TRIG:SOUR IMM command), the function generator outputs a square waveform with a 50% duty cycle from the *Trig Out* connector at the beginning of the sweep. The period of the waveform is equal to the specified sweep time (SWE:TIME command).
- When the *External* trigger source is selected (TRIG:SOUR EXT command), the function generator automatically disables the "trigger out" signal. The rear-panel *Trig Out* connector cannot be used for both operations at the same time (an externally-triggered sweep uses the same connector to trigger the sweep).
- When the *Bus* (software) trigger source is selected (TRIG:SOUR BUS command), the function generator outputs a pulse (>1 µs pulse width) from the *Trig Out* connector at the beginning of each sweep.

OUTPut:TRIGger {**OFF**|**ON**}
OUTPut:TRIGger?

Disable or enable the "trigger out" signal. When enabled, a TTL-compatible square waveform with the specified edge (OUTP:TRIG:SLOP command) is output from the rear-panel *Trig Out* connector at the beginning of the sweep. The default is OFF. The :TRIG? query returns "0" (OFF) or "1" (ON).

MARKer:FREQuency {<*frequency*>|**MINimum**|**MAXimum**}
MARKer:FREQuency? [**MINimum**|**MAXimum**]

Set the marker frequency. This is the frequency at which the signal on the front-panel *Sync* connector goes to a logic low during the sweep. The Sync signal always goes from low to high at the beginning of the sweep. Select from 1 μHz to 80 MHz (limited to 1 MHz for ramps and 25 MHz for arbitrary waveforms). The default is 500 Hz. MIN = 1 μHz. MAX = the start frequency or stop frequency (whichever is higher). The `:FREQ?` query returns the marker frequency in hertz.

- When sweep is enabled, the marker frequency *must* be between the specified start frequency and stop frequency. If you attempt to set the marker frequency to a frequency not in this range, the function generator will automatically set the marker frequency equal to the start frequency or stop frequency (whichever is closer). *From the remote interface, a "Settings conflict" error will be generated and the marker frequency will be adjusted as described.*

MARKer {**OFF**|**ON**}
MARKer?

Disable or enable the frequency marker. When the frequency marker is disabled, the signal output from the *Sync* connector is the normal Sync signal for the carrier waveform (see "Sync Output Signal" on page 62). The default is OFF. The `MARK?` query returns "0" (OFF) or "1" (ON).

- The `OUTP:SYNC` command is overridden by the setting of the `MARK` command. Therefore, when the marker frequency is enabled (and the sweep mode is also enabled), the `OUTP:SYNC` command is ignored

Burst Mode Commands

See also "Burst Mode" starting on page 89 in chapter 3.

Burst Mode Overview

The following is an overview of the steps required to generate a burst. You can use burst in one of two modes as described below. The function generator enables one burst mode at a time based on the *trigger source* and *burst source* that you select (see the table below).

- *Triggered Burst Mode*: In this mode (default), the function generator outputs a waveform with a specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles have been output, the function generator stops and waits for the next trigger. You can configure the function generator to use an internal trigger to initiate the burst or you can provide an external trigger by pressing the front-panel (Trigger) key, by applying a trigger signal to the rear-panel *Trig In* connector, or by sending a software trigger command from the remote interface.
- *External Gated Burst Mode*: In this mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Trig In* connector. When the gate signal is *true*, the function generator outputs a continuous waveform. When the gate signal goes *false*, the current waveform cycle is completed and then the function generator stops while remaining at the voltage level corresponding to the starting burst phase of the selected waveform.

	Burst Mode (BURS:MODE)	**Burst Count** (BURS:NCYC)	**Burst Period** (BURS:INT:PER)	**Burst Phase** (BURS:PHAS)	**Trigger Source** (TRIG:SOUR)
Triggered Burst Mode: Internal Trigger	TRIGgered	Available	Available	Available	IMMediate
Triggered Burst Mode: External Trigger	TRIGgered	Available	Not Used	Available	EXTernal, BUS
Gated Burst Mode: External Trigger	GATed	Not Used	Not Used	Available	Not Used

1 Configure the burst waveform.

Use the APPLy command or the equivalent FUNC, FREQ, VOLT, and VOLT:OFFS commands to select the function, frequency, amplitude, and offset of the waveform. You can select a sine, square, ramp, pulse, or arbitrary waveform (noise is allowed only in the gated burst mode and dc is not allowed). For internally-triggered bursts, the minimum frequency is 2 mHz. For sine and square waveforms, frequencies above 25 MHz are allowed only with an "infinite" burst count.

2 Select the "triggered" or "gated" burst mode.

Select the *triggered* burst mode or external *gated* burst mode using the BURS:MODE command.

3 Set the burst count.

Set the burst count (the number of cycles per burst) to any value between 1 and 1,000,000 cycles (or infinite) using the BURS:NCYC command. *Used in the triggered burst mode only.*

4 Set the burst period.

Set the burst period (the interval at which internally-triggered bursts are generated) to any value from 1 µs to 500 seconds using the BURS:INT:PER command. *Used only in the triggered burst mode with an internal trigger source.*

5 Set the burst starting phase.

Set the starting phase of the burst to any value from -360 degrees to +360 degrees using the BURS:PHAS command.

6 Select the trigger source.

Select the trigger source using the TRIG:SOUR command. *Used in the triggered burst mode only.*

7 Enable the burst mode.

After you have set up the other burst parameters, use the BURS:STAT ON command to enable the burst mode.

Burst Mode Commands

Use the APPLy command or the equivalent FUNC, FREQ, VOLT, and VOLT:OFFS commands to configure the waveform. For internally-triggered bursts, the minimum frequency is 2 mHz. For sine and square waveforms, frequencies above 25 MHz are allowed only with an "infinite" burst count.

BURSt:MODE {**TRIGgered**|**GATed**}
BURSt:MODE?

Select the burst mode. In the *triggered* mode, the function generator outputs a waveform with a specified number of cycles (*burst count*) each time a trigger is received from the specified trigger source (`TRIG:SOUR` command). In the *gated* mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Trig In* connector. The default is TRIG. The `:MODE?` query returns "TRIG" or "GAT".

- When the *gated* mode is selected, the waveform generator either runs or stops based on the logic level of the gate signal applied to the rear-panel *Trig In* connector. You can select the polarity of the *Trig In* connector using the `BURS:GATE:POL` command (see page 194). When the gate signal is *true*, the function generator outputs a continuous waveform. When the gate signal goes *false*, the current waveform cycle is completed and then the function generator stops while remaining at the voltage level corresponding to the starting burst phase of the selected waveform. For a noise waveform, the output stops immediately when the gate signal goes false.
- When the *gated* mode is selected, the burst count, burst period, and trigger source are ignored (these parameters are used for the triggered burst mode only). If a manual trigger is received (`TRIG` command), it is ignored and no error will be generated.

BURSt:NCYCles {<# *cycles*>|**INFinity**|**MINimum**|**MAXimum**}
BURSt:NCYCles? [**MINimum**|**MAXimum**]

Set the number of cycles to be output per burst (*triggered burst mode only*). Select from 1 cycle to 1,000,000 cycles, in 1 cycle increments (*see the restrictions below*). The default is 1 cycle. MIN = 1 cycle. MAX = based on the burst period and frequency as shown below. Select INF to generate a continuous burst waveform. The `:NCYC?` query returns the burst count from 1 to 1,000,000 or "9.9E+37" (for an infinite count).

- If the *Immediate* trigger source is selected (`TRIG:SOUR IMM` command), the burst count must be less than the product of the burst period and the waveform frequency as shown below.

  Burst Count < Burst Period X Waveform Frequency

- The function generator will automatically increase the burst period up to its maximum value to accommodate the specified burst count (but the waveform frequency *will not* be changed). *From the remote interface, a "Settings conflict" error will be generated and the burst period will be adjusted as described.*
- For sine and square waveforms, frequencies above 25 MHz are allowed only with an "infinite" burst count.
- When the *gated* burst mode is selected, the burst count is ignored. However, if you change the burst count while in the gated mode, the function generator remembers the new count and will use it when the triggered mode is selected.

BURSt:INTernal:PERiod {<*seconds*>|**MINimum**|**MAXimum**}
BURSt:INTernal:PERiod? [**MINimum**|**MAXimum**]

Set the burst period for internally-triggered bursts. The burst period defines time from the start of one burst to the start of the next burst. Select from 1 µs to 500 seconds. The default is 10 ms. MIN = 1 µs. MAX = based on the burst count and waveform frequency as shown below. The :PER? query returns the burst period in seconds.

- The burst period setting is used only when *Immediate* triggering is enabled (TRIG:SOUR IMM command). The burst period is ignored when manual or external triggering is enabled (or when the *gated* burst mode is selected).
- It *is not* possible to specify a burst period which is too short for the function generator to output with the specified burst count and frequency (see below). If the burst period is too short, the function generator will automatically adjust it as needed to continuously re-trigger the burst. *From the remote interface, a "Data out of range" error will be generated and the burst period will be adjusted as described.*

$$\text{Burst Period} > \frac{\text{Burst Count}}{\text{Waveform Frequency}} + 200\ \text{ns}$$

BURSt:PHASe {<*angle*>|**MINimum**|**MAXimum**}
BURSt:PHASe? [**MINimum**|**MAXimum**]

Set the starting phase for the burst in degrees or radians as specified by the previous UNIT:ANGL command. Select from -360 degrees to +360 degrees or -2π to +2π radians. The default is 0 degrees (0 radians). MIN = -360 degrees (-2π radians). MAX = +360 degrees (+2π). The :PHAS? query returns the starting phase in degrees or radians.

- For sine, square, and ramp waveforms, 0 degrees is the point at which the waveform crosses zero volts (or the dc offset value) in a positive-going direction. For arbitrary waveforms, 0 degrees is the first waveform point downloaded to memory. This command has no effect on pulse or noise waveforms.
- The burst phase is also used in the *gated* burst mode. When the gate signal goes *false*, the current waveform cycle is completed and then the function generator stops. The output will remain at the voltage level corresponding to the starting burst phase.

BURSt:STATe {**OFF**|**ON**}
BURSt:STATe?

Disable or enable the burst mode. To avoid multiple waveform changes, you can enable the burst mode *after* you have set up the other burst parameters. The default is OFF. The :STAT? query returns "0" (OFF) or "1" (ON).

- The function generator will not allow the burst mode to be enabled at the same time that sweep or any modulation mode is enabled. When you enable burst, the sweep or modulation mode is turned off.

UNIT:ANGLe {**DEGree**|**RADian**}
UNIT:ANGLe?

Select degrees or radians to set the starting phase for the burst with the BURS:PHAS command (remote interface only). The default is DEG. The :ANGL? query returns "DEG" or "RAD".

- From the front panel, the starting phase is always displayed in degrees (radians are not available). If you set the starting phase in radians from the remote interface and then return to front-panel operation, you will see that the function generator converts the phase to degrees.

TRIGger:SOURce {**IMMediate**|**EXTernal**|**BUS**}
TRIGger:SOURce?

Select the trigger source for the *triggered* burst mode only. In the triggered burst mode, the function generator outputs a waveform with the specified number of cycles (*burst count*) each time a trigger is received. After the specified number of cycles have been output, the function generator stops and waits for the next trigger. The default is IMM. The :SOUR? query returns "IMM", "EXT", or "BUS"

- When the *Immediate* (internal) source is selected, the frequency at which the burst is generated is determined by the *burst period* (BURS:INT:PER command).
- When the *External* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Trig In* connector. The function generator outputs the specified number of cycles each time *Trig In* receives a TTL pulse with the edge polarity specified by the TRIG:SLOP command (see page 193). External trigger signals that occur during a burst are ignored.

- When the *Bus* (software) source is selected, the function generator outputs one burst each time a bus trigger command is received. To trigger the function generator from the remote interface (either GPIB or RS-232), send the `TRIG` or `*TRG` (trigger) command. The front-panel (Trigger) key is illuminated when the function generator is waiting for a bus trigger.
- When the *External* or *Bus* trigger source is selected, the *burst count* and *burst phase* remain in effect but the *burst period* is ignored.
- The `APPLy` command automatically sets the trigger source to *Immediate* (equivalent to `TRIG:SOUR IMM` command).
- To ensure synchronization when the *Bus* source is selected, send the `*WAI` (wait) command. When the `*WAI` command is executed, the function generator waits for all pending operations to complete before executing any additional commands. For example, the following command string guarantees that the first trigger is accepted and the operation is executed before the second trigger is recognized.

```
TRIG:SOUR BUS;*TRG;*WAI;*TRG;*WAI
```

- You can use the `*OPC?` (operation complete query) command or the `*OPC` (operation complete) command to signal when the burst is complete. The `*OPC?` command returns "1" to the output buffer when the burst is complete. The `*OPC` command sets the "Operation Complete" bit (bit 0) in the Standard Event register when the burst is complete.

TRIGger:DELay {<*seconds*>|**MINimum**|**MAXimum**}
TRIGger:DELay? [**MINimum**|**MAXimum**]

Insert a time delay between the receipt of the trigger and the start of the burst waveform (used in *triggered* burst mode only). Select from 0 seconds to 85 seconds. The default delay is 0. MIN = 0 seconds. MAX = 85 seconds. The `:DEL?` query returns the trigger delay in seconds.

TRIGger:SLOPe {**POSitive**|**NEGative**}
TRIGger:SLOPe?

Select whether the function generator uses the rising edge or falling edge of the trigger signal on the rear-panel *Trig In* connector for an externally-triggered burst. The default is POS (rising edge). The `:SLOP?` query returns "POS" or "NEG".

BURSt:GATE:POLarity {**NORMal**|**INVerted**}
BURSt:GATE:POLarity?

Select whether the function generator uses true-high or true-low logic levels on the rear-panel *Trig In* connector for an externally-gated burst. The default is NORM (true-high logic). The `:POL?` query returns "NORM" or "INV".

OUTPut:TRIGger:SLOPe {**POSitive**|**NEGative**}
OUTPut:TRIGger:SLOPe?

Select a rising or falling edge for the "trigger out" signal. When enabled using the `OUTP:TRIG` command (see below), a TTL-compatible square waveform with the specified edge is output from the rear-panel *Trig Out* connector at the beginning of the burst. Select "POS" to output a pulse with a rising edge or select "NEG" to output a pulse with a falling edge. The default is POS. The `:SLOP?` query returns "POS" or "NEG".

- When the *Immediate* (internal) trigger source is selected (`TRIG:SOUR IMM` command), the function generator outputs a square waveform with a 50% duty cycle from the *Trig Out* connector at the beginning of the burst. The frequency of the waveform is equal to the specified burst period (`BURS:INT:PER` command).
- When the *External* trigger source is selected (`TRIG:SOUR EXT` command) ***or*** when the *Gated* mode is selected (`BURS:MODE GAT` command), the function generator automatically disables the "trigger out" signal. The rear-panel *Trig Out* connector cannot be used for both operations at the same time (an externally-triggered waveform uses the same connector to trigger the burst).
- When the *Bus* (software) trigger source is selected (`TRIG:SOUR BUS` command), the function generator outputs a pulse (>1 µs pulse width) from the *Trig Out* connector at the beginning of each burst.

OUTPut:TRIGger {**OFF**|**ON**}
OUTPut:TRIGger?

Disable or enable the "trigger out" signal (used with burst and sweep only). When enabled, a TTL-compatible square waveform with the specified edge (`OUTP:TRIG:SLOP` command) is output from the rear-panel *Trig Out* connector at the beginning of the burst. The default is OFF. The `:TRIG?` query returns "0" (OFF) or "1" (ON).

Triggering Commands

Applies to Sweep and Burst only. See also "Triggering" starting on page 98 in chapter 3.

TRIGger:SOURce {**IMMediate**|**EXTernal**|**BUS**}
TRIGger:SOURce?

Select the source from which the function generator will accept a trigger. The function generator will accept an immediate internal trigger, a hardware trigger from the rear-panel *Trig In* connector, or a software (bus) trigger. The default is IMM. The `:SOUR?` query returns "IMM", "EXT", or "BUS"

- When the *Immediate* (internal) source is selected, the function generator outputs continuously when the sweep mode or burst mode is enabled.
- When the *External* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Trig In* connector. The function generator initiates one sweep or outputs one burst each time *Trig In* receives a TTL pulse with the edge polarity specified by the `TRIG:SLOP` command (see page 196).
- When the *Bus* (software) source is selected, the function generator initiates one sweep or outputs one burst each time a bus trigger command is received. To trigger the function generator from the remote interface (GPIB or RS-232) when the *Bus* source selected, send the `TRIG` or `*TRG` (trigger) command. The front-panel Trigger key is illuminated when the function generator is waiting for a bus trigger.
- The `APPLy` command automatically sets the trigger source to *Immediate* (equivalent to `TRIG:SOUR IMM` command).

- To ensure synchronization when the *Bus* source is selected, send the `*WAI` (wait) command. When the `*WAI` command is executed, the function generator waits for all pending operations to complete before executing any additional commands. For example, the following command string guarantees that the first trigger is accepted and the operation is executed before the second trigger is recognized.

```
TRIG:SOUR BUS;*TRG;*WAI;*TRG;*WAI
```

- You can use the `*OPC?` (operation complete query) command or the `*OPC` (operation complete) command to signal when the sweep or burst is complete. The `*OPC?` command returns "1" to the output buffer when the sweep or burst is complete. The `*OPC` command sets the "Operation Complete" bit (bit 0) in the Standard Event register when the sweep or burst is complete.

`TRIGger`

Trigger a sweep or burst from the remote interface. This command can be used with any of the available trigger sources (`TRIG:SOUR` command). For example, you can use the `TRIG` command to issue an immediate trigger while waiting for an external trigger.

`*TRG`

Trigger a sweep or burst from the remote interface *only* if the bus (software) trigger source is currently selected (`TRIG:SOUR BUS` command).

`TRIGger:DELay` {<*seconds*>|**`MINimum`**|**`MAXimum`**}
`TRIGger:DELay?` [**`MINimum`**|**`MAXimum`**]

Insert a time delay between the receipt of the trigger and the start of the burst waveform (used in *triggered* burst mode only). Select from 0 seconds to 85 seconds. The default delay is 0. MIN = 0 seconds. MAX = 85 seconds. The `:DEL?` query returns the trigger delay in seconds.

`TRIGger:SLOPe` {**`POSitive`**|**`NEGative`**}
`TRIGger:SLOPe?`

Select whether the function generator uses the rising edge or falling edge of the trigger signal on the rear-panel *Trig In* connector. The default is POS (rising edge). The `:SLOP?` query returns "POS" or "NEG".

BURSt:GATE:POLarity {**NORMal**|**INVerted**}
BURSt:GATE:POLarity?

Select whether the function generator uses true-high or true-low logic levels on the rear-panel *Trig In* connector for an externally-gated burst. The default is NORM (true-high logic). The :POL? query returns "NORM" or "INV".

OUTPut:TRIGger:SLOPe {**POSitive**|**NEGative**}
OUTPut:TRIGger:SLOPe?

Select a rising or falling edge for the "trigger out" signal. When enabled using the OUTP:TRIG command (see below), a TTL-compatible square waveform with the specified edge is output from the rear-panel *Trig Out* connector at the beginning of the sweep or burst. Select "POS" to output a pulse with a rising edge or select "NEG" to output a pulse with a falling edge. The default is POS. The :SLOP? query returns "POS" or "NEG".

- When the *Immediate* (internal) trigger source is selected (TRIG:SOUR IMM command), the function generator outputs a square waveform with a 50% duty cycle from the *Trig Out* connector at the beginning of the sweep or burst. The period of the waveform is equal to the specified sweep time (SWE:TIME command) or burst period (BURS:INT:PER command).
- When the *External* trigger source is selected (TRIG:SOUR EXT command), the function generator automatically disables the "trigger out" signal. The rear-panel *Trig Out* connector cannot be used for both operations at the same time (an externally-triggered waveform uses the same connector to trigger the sweep or burst).
- When the *Bus* (software) trigger source is selected (TRIG:SOUR BUS command), the function generator outputs a pulse (>1 µs pulse width) from the *Trig Out* connector at the beginning of each sweep or burst.

OUTPut:TRIGger {**OFF**|**ON**}
OUTPut:TRIGger?

Disable or enable the "trigger out" signal (used for sweep and burst only). When enabled, a TTL-compatible square waveform with the specified edge (OUTP:TRIG:SLOP command) is output from the rear-panel *Trig Out* connector at the beginning of the sweep or burst. The default is OFF. The :TRIG? query returns "0" (OFF) or "1" (ON).

Arbitrary Waveform Commands

See also "Arbitrary Waveforms" starting on page 103 in chapter 3.

Arbitrary Waveform Overview

The following is an overview of the steps required to download and output an arbitrary waveform over the remote interface. The commands used for arbitrary waveforms are listed on page 200. *Refer to chapter 7, "Tutorial", for more information on the internal operation of downloading and outputting an arbitrary waveform.*

> Chapter 6, "Application Programs", contains example programs which show the use of arbitrary waveforms. You may find it useful to refer to the programs after reading the following section in this chapter.

1 Select the waveform frequency, amplitude, and offset.

Use the `APPLy` command or the equivalent `FREQ`, `VOLT`, and `VOLT:OFFS` commands to select the frequency, amplitude, and offset of the waveform.

2 Download the waveform points into volatile memory.

You can download from 1 point (a dc signal) to 65,536 (64K) points per waveform. You can download the points as floating-point values, binary integer values, or decimal integer values. Use the `DATA` command to download floating-point values from -1.0 to +1.0. Use the `DATA:DAC` command to download binary integer or decimal integer values from -2047 to +2047.

To ensure that binary data is downloaded properly, you must select the order in which the bytes are downloaded using the `FORM:BORD` *command.*

3 Copy the arbitrary waveform to non-volatile memory.

You can output the arbitrary waveform directly from volatile memory or you can copy the waveform to non-volatile memory using the `DATA:COPY` command.

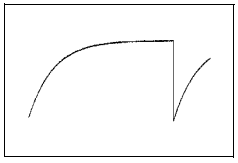
4 Select the arbitrary waveform to output.

You can select one of the five built-in arbitrary waveforms, one of four user-defined waveforms, or the waveform currently downloaded to volatile memory. Use the `FUNC:USER` command to select the waveform.

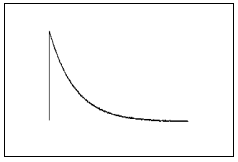
5 Output the selected arbitrary waveform.

Use the `FUNC USER` command to output the waveform previously selected with the `FUNC:USER` command.

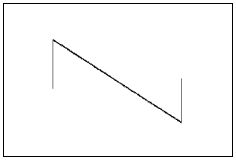
The five built-in arbitrary waveforms are shown below.



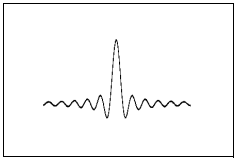
Exponential Rise



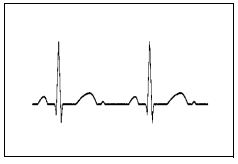
Exponential Fall



Negative Ramp



Sinc



Cardiac

Arbitrary Waveform Commands

DATA VOLATILE, <*value*>**,** <*value*>**, . . .**

Download *floating-point* values from -1 to +1 into volatile memory. You can download from 1 to 65,536 (64K) points per waveform. The function generator takes the specified number of points and expands them to fill waveform memory. If you download *less than* 16,384 (16K) points, a waveform with 16,384 points is automatically generated. If you download *more than* 16,384 points, a 65,536-point waveform is generated.

- The values -1 and +1 correspond to the *peak* values of the waveform (if the offset is 0 volts). For example, if you set the amplitude to 10 Vpp (0V offset), "+1" corresponds to +5V and "-1" corresponds to -5V.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of values between ±1 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- Downloading floating-point values (using `DATA VOLATILE`) is slower than downloading binary values (using `DATA:DAC VOLATILE`) but is more convenient when using trigonometric functions which return values from -1 to +1.
- The `DATA` command overwrites the previous waveform in volatile memory (and no error will be generated). Use the `DATA:COPY` command to copy the waveform to *non-volatile* memory.
- Up to four user-defined waveforms can be stored in non-volatile memory. Use the `DATA:DEL` command to delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory (as well as the five built-in waveforms).
- After downloading the waveform data to memory, use the `FUNC:USER` command to choose the active waveform and the `FUNC USER` command to output it.
- The following statement shows how to use the `DATA` command to download seven points to volatile memory.

```
DATA VOLATILE, 1, .67, .33, 0, -.33, -.67, -1
```

DATA:DAC VOLATILE, {<*binary block*>|<*value*>**,** <*value*>**, . . .** }

Download *binary* or *decimal* integer values from -2047 to +2047 into volatile memory. You can download from 1 to 65,536 (64K) points per waveform in IEEE-488.2 binary block format or as a list of values. The range of values corresponds to the values available using internal 12-bit DAC (Digital-to-Analog Converter) codes. The function generator takes the specified number of points and expands them to fill waveform memory. If you download *less than* 16,384 (16K) points, a waveform with 16,384 points is automatically generated. If you download *more than* 16,384 points, a 65,536-point waveform is generated.

- The values -2047 and +2047 correspond to the peak values of the waveform (if the offset is 0 volts). For example, if you set the output amplitude to 10 Vpp, "+2047" corresponds to +5V and "-2047" corresponds to -5V.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC. For example, the built-in "Sinc" waveform does not use the full range of values between ±2047 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- The `DATA:DAC` command overwrites the previous waveform in volatile memory (and no error will be generated). Use the `DATA:COPY` command to copy the waveform to *non-volatile* memory.
- Up to four user-defined waveforms can be stored in non-volatile memory. Use the `DATA:DEL` command to delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory (as well as the five built-in waveforms).
- After downloading the waveform data to memory, use the `FUNC:USER` command to choose the active waveform and the `FUNC USER` command to output it.
- To download binary data over the RS-232 interface, you can use any handshake mode except XON/XOFF and be sure to select "Parity None" (8 data bits). You must also insert a pause of approximately 1 ms between sending the header and sending the binary block.
  See page 219 for more information on configuring the RS-232 interface.

- The following statement shows how to use the `DATA:DAC` command to download seven integer points using the binary block format (*see also "Using the IEEE-488.2 Binary Block Format" below*).

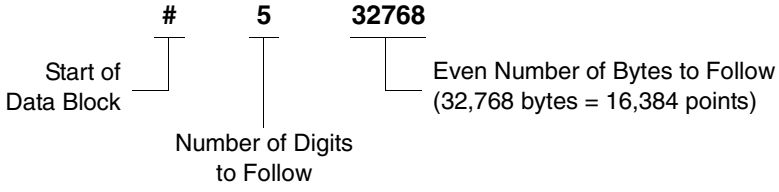
```
DATA:DAC VOLATILE, #214 <Binary Data>
```

- The following statement shows how to use the `DATA:DAC` command to download five integer points in decimal format.

```
DATA:DAC VOLATILE, 2047, 1024, 0, -1024, -2047
```


Using the IEEE-488.2 Binary Block Format

In the binary block format, a *block header* precedes the waveform data. The block header has the following format:



The function generator represents binary data as 16-bit integers, which are sent as two bytes. Therefore, **the total number of bytes is always twice the number of data points in the waveform** (and must always be an **even number**). For example, 32,768 bytes are required to download a waveform with 16,384 points.

Use the FORM:BORD command to select the byte order for binary transfers in block mode. If you specify FORM:BORD NORM (default), the most-significant byte (MSB) of each data point is assumed first. If you specify FORM:BORD SWAP, the least-significant byte (LSB) of each data point is assumed first. Most computers use the "swapped" byte order.

FORMat:BORDer {**NORMal**|**SWAPped**}
FORMat:BORDer?

Used for binary block transfers only. Select the byte order for binary transfers in the block mode using the DATA:DAC command. The default is NORM. The :BORD? query returns "NORM" or "SWAP".

- In *NORM* byte order (default), the most-significant byte (MSB) of each data point is assumed first.
- In *SWAP* byte order, the least-significant byte (LSB) of each data point is assumed first. Most computers use the "swapped" byte order.
- The function generator represents binary data as signed 16-bit integers, which are sent as two bytes. Therefore, each waveform data point requires 16 bits, which must be transferred as two bytes on the function generator's 8-bit interface.

DATA:COPY <*destination arb name*> [**,VOLATILE**]

Copy the waveform from volatile memory to the specified name in non-volatile memory. The source for the copy is always "volatile". You cannot copy ***from*** any other source and you cannot copy ***to*** "volatile".

- The arb name may contain up to 12 characters. The first character *must* be a letter (A-Z), but the remaining characters can be numbers (0-9) or the underscore character (" _ "). Blank spaces are not allowed. If you specify a name with more than 12 characters, a "Program mnemonic too long" error is generated.
- The VOLATILE parameter is optional and can be omitted. Note that the keyword "VOLATILE" *does not* have a short form.
- The following built-in waveform names are reserved and cannot be used with the DATA:COPY command: "EXP_RISE", "EXP_FALL", "NEG_RAMP", "SINC", and "CARDIAC". If you specify one of the built-in waveforms, a "Cannot overwrite a built-in waveform" error is generated.
- The function generator does not distinguish between upper- and lower-case letters. Therefore, **ARB_1** and **arb_1** are the same name. All characters are converted to upper case.

- If you copy to a waveform name that already exists, the previous waveform is overwritten (and no error will be generated). However, you cannot overwrite any of the five built-in waveforms.
- Up to four user-defined waveforms can be stored in non-volatile memory. If memory is full and you try to copy a new waveform to non-volatile memory, a "Not enough memory" error is generated. Use the `DATA:DEL` command to delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory. Use the `DATA:CAT?` command to list all waveforms currently stored in volatile and non-volatile memory.
- The following statement shows how to use the `DATA:COPY` command to copy the VOLATILE waveform into named storage "ARB_1".

```
DATA:COPY ARB_1, VOLATILE
```

FUNCtion:USER {<*arb name*>|**VOLATILE**}
FUNCtion:USER?

Select one of the five built-in arbitrary waveforms, one of four user-defined waveforms, or the waveform currently downloaded to volatile memory. The `:USER?` query returns "EXP_RISE", "EXP_FALL", "NEG_RAMP", "SINC", "CARDIAC", "VOLATILE", or the name of any user-defined waveforms in non-volatile memory.

- Note that this command *does not* output the selected arbitrary waveform. Use the `FUNC USER` command (see the following page) to output the selected waveform.
- The names of the five built-in arbitrary waveforms are: "EXP_RISE", "EXP_FALL", "NEG_RAMP", "SINC", and "CARDIAC".
- To select the waveform currently stored in volatile memory, specify the VOLATILE parameter. The keyword "VOLATILE" *does not* have a short form.
- If you select a waveform name that is not currently downloaded, a "Specified arb waveform does not exist" error is generated.

- The function generator does not distinguish between upper- and lower-case letters. Therefore, **ARB_1** and **arb_1** are the same name. All characters are converted to upper case.
- Use the DATA:CAT? command to list the names of the five built-in waveforms (non-volatile), "VOLATILE" if a waveform is currently downloaded to volatile memory, and the names of any user-defined waveforms (non-volatile).

FUNCtion USER
FUNCtion?

Select the arbitrary waveform function and output the current arbitrary waveform. When executed, this command outputs the arbitrary waveform currently selected by the FUNC:USER command (see the previous page). The selected waveform is output using the current frequency, amplitude, and offset voltage settings. The FUNC? query returns "SIN", "SQU", "RAMP", "PULS", "NOIS", "DC", or "USER".

- Use the APPLy command or the equivalent FREQ, VOLT, and VOLT:OFFS commands to select the frequency, amplitude, and offset of the waveform.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "SINC" waveform does not use the full range of binary values between ±1 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- If you select an arbitrary waveform as the *modulating* waveshape ("USER"), the waveform is automatically limited to 8K points. Extra waveform points are removed using decimation.

DATA:CATalog?

List the names of *all* waveforms currently available for selection. Returns the names of the five built-in waveforms (non-volatile memory), "VOLATILE" if a waveform is currently downloaded to volatile memory, and all user-defined waveforms downloaded to non-volatile memory.

- A series of quoted strings separated with commas is returned as shown in the example below.

```
"VOLATILE","EXP_RISE","EXP_FALL","NEG_RAMP",
"SINC","CARDIAC","TEST1_ARB","TEST2_ARB"
```

- Use the DATA:DEL command to delete the waveform in volatile memory or any of the user-defined waveforms in non-volatile memory.

DATA:NVOLatile:CATalog?

List the names of all user-defined arbitrary waveforms downloaded to *non-volatile* memory. Returns the names of up to four waveforms.

- A series of quoted strings separated with commas is returned as shown in the example below. If no user-defined waveforms are currently downloaded, the command returns a null string (" ").

```
"TEST1_ARB","TEST2_ARB","TEST3_ARB","TEST4_ARB"
```

- Use the DATA:DEL command to delete any of the user-defined waveforms in non-volatile memory.

DATA:NVOLatile:FREE?

Query the number of non-volatile memory slots available to store user-defined waveforms. Returns the number of memory slots available to store user-defined waveforms. Returns "0" (memory is full), "1", "2", "3", or "4".

DATA:DELete *<arb name>*

Delete the specified arbitrary waveform from memory. You can delete the waveform in volatile memory or any of the four user-defined waveforms in non-volatile memory.

- You cannot delete the arbitrary waveform that is currently being output. If you attempt to delete this waveform, a "Not able to delete the currently selected active arb waveform" error is generated.
- You cannot delete any of the five built-in arbitrary waveforms. If you attempt to delete one of these waveforms, a "Not able to delete a built-in arb waveform" error is generated.
- Use the DATA:DEL:ALL command to delete the waveform in volatile memory and all user-defined non-volatile waveforms *all at once*. If one of the waveforms is currently being output, a "Not able to delete the currently selected active arb waveform" error is generated.

DATA:DELete:ALL

Delete all user-defined arbitrary waveforms from memory. This command deletes the waveform in volatile memory and all user-defined waveforms in non-volatile memory. The five built-in waveforms in non-volatile memory *are not* deleted.

- The colon before the ALL parameter is required (DATA:DELete**:**ALL). If you insert a space instead of a colon, the function generator will attempt to delete an arbitrary waveform with the name "ALL". If no such waveform is stored in memory, a "Specified arb waveform does not exist" error is generated.
- Use the DATA:DEL *<arb name>* command to delete stored waveforms *one at a time*.
- You cannot delete the arbitrary waveform that is currently being output. If you attempt to delete this waveform, a "Not able to delete the currently selected active arb waveform" error is generated.
- You cannot delete any of the five built-in arbitrary waveforms. If you attempt to delete one of these waveforms, a "Not able to delete a built-in arb waveform" error is generated.

DATA:ATTRibute:AVERage? [<*arb name*>]

Query the *arithmetic average* of all data points for the specified arbitrary waveform (-1 ≤ *average* ≤ +1). The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.

DATA:ATTRibute:CFACtor? [<*arb name*>]

Query the *crest factor* of all data points for the specified arbitrary waveform. Crest factor is the ratio of the peak value to the RMS value of the waveform. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.

DATA:ATTRibute:POINts? [<*arb name*>]

Query the *number of points* in the specified arbitrary waveform. Returns a value from 1 to 65,536 points. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.

DATA:ATTRibute:PTPeak? [<*arb name*>]

Query the *peak-to-peak* value of all data points for the specified arbitrary waveform. The default *arb name* is the arbitrary waveform currently active (selected with FUNC:USER command).

- This command returns a value from "0" to "+1.0", with "+1.0" indicating full amplitude available.
- The maximum amplitude will be limited if the data points do not span the full range of the output DAC (Digital-to-Analog Converter). For example, the built-in "Sinc" waveform does not use the full range of binary values between ±1 and therefore its maximum amplitude is 6.087 Vpp (into 50 ohms).
- If you query a waveform that is not currently stored in memory, a "Specified arb waveform does not exist" error is generated.

State Storage Commands

The function generator has five storage locations in non-volatile memory to store instrument states. The locations are numbered 0 through 4. The function generator automatically uses location "0" to hold the state of the instrument at power down. You can also assign a user-defined name to each of the locations (1 through 4) for use from the front panel.

***SAV** {**0**|**1**|**2**|**3**|**4**}

Store (save) the current instrument state in the specified non-volatile storage location. Any state previously stored in the same location will be overwritten (and no error will be generated).

- You can store the instrument state in any of the five storage locations. However, you can only recall a state from a location that contains a previously-stored state.
- From the *remote interface* only, you can use storage location "0" to store a fifth instrument state (you cannot store to this location from the front panel). However, note that location "0" is automatically overwritten when power is cycled (the instrument state previously stored will be overwritten).
- The state storage feature "remembers" the selected function (including arbitrary waveforms), frequency, amplitude, dc offset, duty cycle, symmetry, as well as any modulation parameters in use.
- If you delete an arbitrary waveform from non-volatile memory after storing the instrument state, the waveform data is lost and the function generator *will not* output the waveform when the state is recalled. The built-in "exponential rise" waveform is output in place of the deleted waveform.

- When power is turned off, the function generator automatically stores its state in storage location "0". You can configure the function generator to automatically recall the power-down state when power is restored. *See the* `MEM:STAT:REC:AUTO` *command on page 212 for more information.*
- The front-panel display state (`DISP` command) is saved when you store the instrument state. When you recall the instrument state, the front-panel display will return to the previous state.
- An instrument reset (`*RST` command) *does not* affect the configurations stored in memory. Once a state is stored, it remains until it is overwritten or specifically deleted.

`*RCL` {**`0`**|**`1`**|**`2`**|**`3`**|**`4`**}

Recall the instrument state stored in the specified non-volatile storage location. You cannot recall the instrument state from a storage location that is empty.

- When shipped from the factory, storage locations "1" through "4" are empty (location "0" has the power-on state).
- From the *remote interface* only, you can use location "0" to store a fifth instrument state (you cannot store to this location from the front panel). However, keep in mind that location "0" is automatically overwritten when power is cycled (the instrument state previously stored will be overwritten).

MEMory:STATe:NAME {**0**|**1**|**2**|**3**|**4**} [**,**<*name*>]
MEMory:STATe:NAME? {**0**|**1**|**2**|**3**|**4**}

Assign a custom name to the specified storage location. You can name a location from the front panel or over the remote interface, but you can only recall a state by name from the front panel (the `*RCL` command requires a numeric parameter). The `:NAME?` query returns a quoted string containing the name currently assigned to the specified storage location. If you have not assigned a user-defined name to the specified location, the default name is returned ("AUTO_RECALL", "STATE_1", "STATE_2", "STATE_3", or "STATE_4").

- The name can contain up to 12 characters. The first character *must* be a letter (A-Z), but the remaining characters can be letters, numbers (0-9), or the underscore character (" _ "). Blank spaces are not allowed. An error is generated if you specify a name with more than 12 characters. An example is shown below.

```
MEM:STATE:NAME 1,TEST_WFORM_1
```

- From the front panel, you *cannot* assign a custom name to storage location "0".
- If you do not specify a name (note that the *name* parameter is optional), the default name is assigned to that state. This provides a way to clear a name (however, the stored state *is not* deleted).
- The function generator *will not* prevent you from assigning the *same* name to different storage locations. For example, you can assign the same name to locations "1" and "2".

MEMory:STATe:DELete {**0**|**1**|**2**|**3**|**4**}

Delete the contents of the specified storage location. If you have assigned a user-defined name to a location (`MEM:STAT:NAME` command), this command also removes the name that you assigned and restores the default name ("AUTO_RECALL", "STATE_1", "STATE_2", etc.). Note that you cannot recall the instrument state from a storage location that is empty. An error is generated if you attempt to recall a deleted state.

MEMory:STATe:RECall:AUTO {**OFF**|**ON**}
MEMory:STATe:RECall:AUTO?

Disable or enable the automatic recall of the power-down state from storage location "0" when power is turned on. Select "ON" to automatically recall the power-down state when power is turned on. Select "OFF" (default) to issue a reset (`*RST` command) when power is turned on (state "0" is not automatically recalled). The `:AUTO?` query returns "0" (OFF) or "1" (ON).

MEMory:STATe:VALid? {**0**|**1**|**2**|**3**|**4**}

Query the specified storage location to determine if a valid state is currently stored in that location. You can use this command before sending the `*RCL` command to determine if a state has been previously stored in this location. Returns "0" if no state has been stored or if it has been deleted. Returns "1" if a valid state is stored in the specified location.

MEMory:NSTates?

Query the total number of memory locations available for state storage. Always returns "5" (memory location "0" is included).

System-Related Commands

See also "System-Related Operations" starting on page 109 in chapter 3.

SYSTem:ERRor?

Read and clear one error from the function generator's *error queue*. A record of up to 20 command syntax or hardware errors can be stored in the error queue. *See chapter 5 for a complete listing of the error messages.*

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. Errors are cleared as you read them. The function generator beeps once each time an error is generated (unless disabled using the `SYST:BEEP:STAT` command).
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with "*Queue overflow*". No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the function generator responds with "*No error*".
- The error queue is cleared by the `*CLS` (clear status) command or when the power is cycled. The errors are also cleared when you read the queue. The error queue *is not* cleared by a reset (`*RST` command).
- Errors have the following format (the error string may contain up to 255 characters).

```
-113,"Undefined header"
```

***IDN?**

Read the function generator's identification string which contains four fields separated by commas. The first field is the manufacturer's name, the second field is the model number, the third field is not used (always "0"), and the fourth field is a revision code which contains five numbers separated by dashes.

- The command returns a string with the following format (be sure to dimension a string variable with at least 50 characters).

  Agilent Technologies,33250A,0,**m.mm-l.ll-f.ff-gg-p**

  m.mm = Main firmware revision number
  l.ll = Loader firmware revision number
  f.ff = I/O processor firmware revision number
  gg = Gate array revision number
  p = Printed circuit board revision number

DISPlay {**OFF**|**ON**}
DISPlay?

Disable or enable the function generator front-panel display. When it is disabled, the front-panel display is blanked (however, the bulb used to backlight the display remains enabled). The DISP? query returns "0" (OFF) or "1" (ON).

- With the front-panel display disabled, there will be some improvement in command execution speed from the remote interface.
- Sending a message to the front-panel display from the remote interface (DISP:TEXT command) overrides the display state. This means that you can display a message even if the display is disabled (remote interface errors are always displayed even if the display is disabled).
- The display is automatically enabled when power is cycled, after an instrument reset (*RST command), or when you return to local (front panel) operation. Press the Local key or execute the IEEE-488 GTL (*Go To Local*) command from the remote interface to return to the local state.
- The display state is saved when you store the instrument state using *SAV command. When you recall the instrument state using *RCL command, the front-panel display will return to the previous state.

DISPlay:TEXT *<quoted string>*
DISPlay:TEXT?

Display a text message on the function generator's front-panel display. Sending a text message to the display overrides the display state as set by the DISP command. The :TEXT? query reads the message sent to the front-panel display and returns a quoted string.

- You can use upper- or lower-case letters (A-Z), numbers (0-9), and any other character on a standard computer keyboard. Depending on the number of characters you specify in the string, the function generator will choose one of two font sizes to display the message. You can display approximately 12 characters in a large font and approximately 40 characters in a small font. An example is shown below.

```
DISP:TEXT 'Test in Progress...'
```

- While a message is displayed, information relating to the output waveform, such as frequency and amplitude, *is not* sent to the front-panel display.

DISPlay:TEXT:CLEar

Clear the text message currently shown on the function generator's front-panel display.

- If the display is currently enabled (DISP ON command), the DISP:TEXT:CLEAR command will return the normal front-panel display mode.
- If the display is currently disabled (DISP OFF command), the DISP:TEXT:CLEAR command will clear the message but the display will remain disabled. To enable the display, send the DISP ON command, press the Local key, send the GTL (*Go To Local*) command for GPIB, or send the SYST:LOCAL command for RS-232.

***RST**

Reset the function generator to its factory default state (*see "Factory Default Settings" on page 127*) independent of the `MEM:STAT:REC:AUTO` command setting. This command will abort a sweep or burst in progress and will re-enable the front-panel display if it was previously disabled (`DISP OFF` command).

***TST?**

Perform a complete self-test of the function generator. Returns "+0" (PASS) or "+1" (FAIL). If the test fails, one or more error messages will be generated to provide additional information on the failure. Use the `SYST:ERR?` command to read the error queue (see page 213).

SYSTem:VERSion?

Query the function generator to determine the present SCPI version. Returns a string in the form "YYYY.V", where "YYYY" represents the year of the version, and "V" represents a version number for that year (e.g., 1997.0).

SYSTem:BEEPer

Issue a single beep immediately.

SYSTem:BEEPer:STATe {**OFF**|**ON**}
SYSTem:BEEPer:STATe?

Disable or enable the tone heard when an error is generated from the front-panel or over the remote interface. The current selection is stored in *non-volatile* memory. The `:STAT?` query returns "0" (OFF) or "1" (ON).

***LRN?**

Query the function generator and return a string of SCPI commands containing the current settings (learn string). You can then send the string back to the instrument to restore this state at a later time. For proper operation, do not modify the returned string before sending it to the function generator. The returned string will contain about 1,500 characters. To avoid possible instrument errors, execute the `*RST` command first, *before* sending the learn string to the instrument.

***OPC**

Set the "Operation Complete" bit (bit 0) in the Standard Event register after the previous commands have been executed. Other commands may be executed before the bit is set. This command is used in the triggered sweep or triggered burst modes to provide a way to poll or interrupt the computer when the sweep or burst is complete.

***OPC?**

Return "1" to the output buffer after the previous commands have been executed. Other commands cannot be executed until this command completes. *Used in the triggered sweep and triggered burst modes only*.

***WAI**

Wait for all pending operations to complete before executing any additional commands over the interface. *Used in the triggered sweep and triggered burst modes only*.

Interface Configuration Commands

See also "Remote Interface Configuration" on page 118 in chapter 3.

SYSTem:INTerface {**GPIB**|**RS232**}

Select the remote interface. Only one interface can be enabled at a time. The GPIB interface is selected when the function generator is shipped from the factory. The command *does not* have a query form.

SYSTem:LOCal

Place the function generator in the *local* mode for RS-232 operation. All keys on the front panel are fully functional while in the local mode.

SYSTem:RWLock

Place the function generator in the *remote* mode for RS-232 operation and disable all front-panel keys.

<**Break**>

Clear the operation in progress over the RS-232 interface and discard any pending output data. This is equivalent to the IEEE-488 Device Clear operation over the GPIB interface.

RS-232 Interface Configuration

See also "Remote Interface Configuration" on page 118 in chapter 3.

This section contains information to help you use the function generator over the RS-232 interface. The programming commands for RS-232 operation are listed on the previous page.

RS-232 Configuration Overview

Configure the RS-232 interface using the parameters shown below. Use the front-panel *Utility–I/O* menu to select the baud rate, parity, number of data bits, and handshake mode (see page 45 for more information on using the front-panel menus).

Baud Rate:	300, 600, 1200, 2400, 4800, 9600, 19200, 38400, **57600** (*factory setting*), 115200
Parity / Data Bits:	**None / 8 data bits** (*factory setting*) Even / 7 data bits Odd / 7 data bits
Handshake Mode:	None (no handshake mode) **DTR / DSR** (*factory setting*) Modem RTS / CTS XON / XOFF
Number of Start Bits:	**1 bit** (*fixed*)
Number of Stop Bits:	**1 bit** (*fixed*)

To download binary data for arbitrary waveforms over the RS-232 interface, you can use any handshake mode except XON/XOFF and be sure to select **Parity None** (8 data bits). You must also insert a pause of approximately 1 ms between sending the header and sending the binary block.

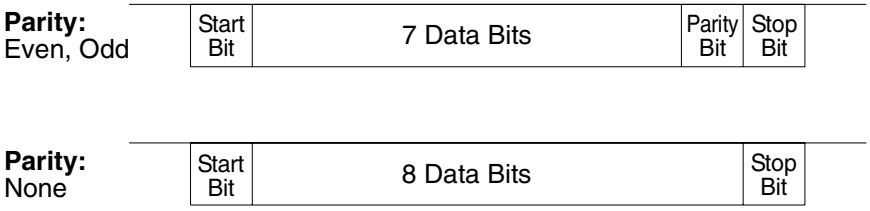
RS-232 Handshake Methods

You can select one of several handshake (or "flow control") methods to coordinate the transfer of data between the function generator and your computer or modem. The default handshake is DTR/DSR.

- *None*: In this mode, data is sent and received over the interface without any flow control used. When using this method, use a slower baud rate (< 9600 baud) and avoid sending more than 128 characters without stopping or reading a response.
- *DTR/DSR*: In this mode, the function generator monitors the state of the DSR (data set ready) line on the RS-232 connector. When the line goes true, the function generator sends data over the interface. When the line goes false, the function generator stops sending information (typically within six characters). The function generator sets the DTR line false when the input buffer is almost full (about 100 characters) and releases the line when space is available again.
- *Modem*: This mode uses the DTR/DSR and RTS/CTS lines to control the flow of data between the function generator and a modem. When the RS-232 interface is selected, the function generator sets the DTR line true. The DSR line is set true when the modem is on-line. The function generator sets the RTS line true when it is ready to receive data. The modem sets the CTS line true when it is ready to accept data. The function generator sets the RTS line false when the input buffer is almost full (about 100 characters) and releases the line when space is available again.
- *RTS/CTS*: This mode operates the same as the DTR/DSR mode but uses the RTS (request to send) and CTS (clear to send) lines on the RS-232 connector instead. When the CTS line goes true, the function generator sends data over the interface. When the line goes false, the function generator stops sending information (typically within six characters). The function generator sets the RTS line false when the input buffer is almost full (about 100 characters) and releases the line when space is available again.
- *XON/XOFF*: This mode uses special characters embedded in the data stream to control the flow. If the function generator is addressed to send data, it continues sending data until the "XOFF" character (13H) is received. When the "XON" character (11H) is received, the function generator resumes sending data.

RS-232 Data Frame Format

A character *frame* consists of all the transmitted bits that make up a single character. The frame is defined as the bits from the *start bit* to the *stop bit*, inclusively. Within the frame, you can select the baud rate, number of data bits, and parity type. The function generator uses the following frame formats for seven and eight data bits.



Connection to a Computer

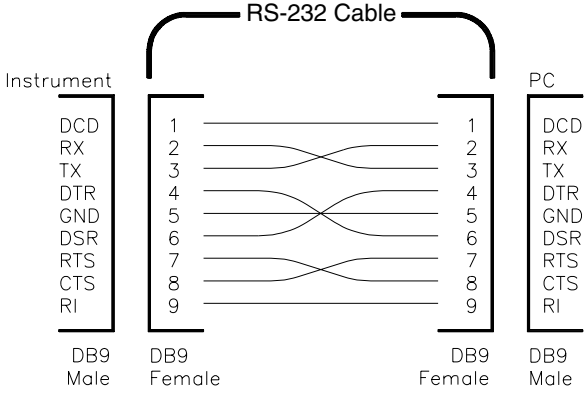
To connect the function generator to a computer, you must have the proper interface cable. Most computers are DTE (*Data Terminal Equipment*) devices. Since the function generator is also a DTE device, you must use a DTE-to-DTE interface cable. These cables are also called *null-modem*, *modem-eliminator*, or *crossover cables*.

The interface cable must also have the proper connector on each end and the internal wiring must be correct. Connectors typically have 9 pins (DB-9 connector) or 25 pins (DB-25 connector) with a “male” or “female” pin configuration. A male connector has pins inside the connector shell and a female connector has holes inside the connector shell.

If you cannot find the correct cable for your configuration, you may have to use a *wiring adapter*. If you are using a DTE-to-DTE cable, make sure the adapter is a “straight-through” type. Typical adapters include gender changers, null-modem adapters, and DB-9 to DB-25 adapters.

If your computer has a 9-pin serial port with a male connector, use the cable included with the function generator. If you need an additional cable, order the F1047-80002 cable which is part of the *Agilent 34398A Cable Kit*. This cable has a 9-pin female connector on each end.

The cable pin diagram is shown below (this is the pinout for the cable shipped with the 33250A). For proper operation, you must use an RS-232 cable with the same pinout as shown below.



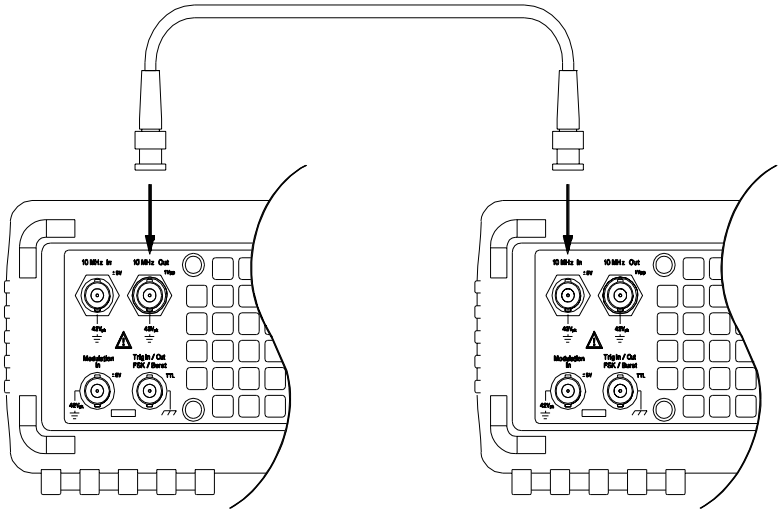
RS-232 Troubleshooting

Here are a few things to check if you are having problems communicating over the RS-232 interface. If you need additional help, refer to the documentation that came with your computer.

- Verify that the function generator's RS-232 interface is selected (GPIB is the default interface). Then, verify that the function generator and your computer are configured for the same baud rate, parity, and number of data bits. Make sure that your computer is set up for *1 start bit* and *1 stop bit* (these values are fixed on the 33250A).
- Verify that you have connected the correct interface cable and adapters. Even if the cable has the proper connectors for your system, the internal wiring may not be correct. You can use the *34398A Cable Kit* to connect the function generator to most computers.
- Verify that you have connected the interface cable to the correct serial port on your computer (COM1, COM2, etc.).

Phase-Lock Commands

The rear-panel *10 MHz In* and *10 MHz Out* connectors allow synchronization between multiple Agilent 33250As (see connection diagram below) or to an external 10 MHz clock signal. You can also control the phase offset from the front panel or over the remote interface.



PHASe {<*angle*>|**MINimum**|**MAXimum**}
PHASe? [**MINimum**|**MAXimum**]

Adjust the phase offset of the output waveform in degrees or radians as specified by the previous UNIT:ANGL command (not available for pulse and noise). Select from -360 degrees to +360 degrees or -2π to +2π radians. The default is 0 degrees (0 radians). MIN = -360 degrees (-2π radians). MAX = +360 degrees (+2π radians). The PHAS? query returns the phase offset in degrees or radians.

- The specified phase adjustment causes a "bump" or "hop" in the output waveform in order to change the phase relationship to the external signal to which it is currently locked.
- This phase adjustment for phase-lock applications is independent of the burst phase as set by the BURS:PHAS command (see page 191).

UNIT:ANGLe {**DEGree**|**RADian**}
UNIT:ANGLe?

Select degrees or radians to set the phase offset value using the PHAS command (remote interface only). The default is DEG. The :ANGL? query returns "DEG" or "RAD".

- From the front panel, the phase offset is always displayed in degrees (radians are not available). If you set the phase offset in radians from the remote interface and then return to front-panel operation, you will see that the function generator converts the phase offset to degrees.

PHASe:REFerence

Immediately set the zero-phase reference point without changing the output of the function generator. This command *does not* change the phase offset as set with the PHAS command – it only changes the phase reference. This command *does not* have a query form.

PHASe:UNLock:ERRor:STATe {**OFF**|**ON**}
PHASe:UNLock:ERRor:STATe?

Disable or enable the function generator from generating an error if the phase-lock is ever lost. The default is OFF. If the phase-lock is lost and the error is enabled, a "Reference phase-locked loop is unlocked" error is generated. The unlock error setting is stored in *non-volatile* memory. The :STAT? query returns "0" (OFF) or "1" (ON).

The SCPI Status System

This section describes the structure of the SCPI status system used by the function generator. The status system records various conditions and states of the instrument in several register groups as shown on the following page. Each of the register groups is made up of several low-level registers called Condition registers, Event registers, and Enable registers which control the action of specific bits within the register group.

What is a Condition Register?

A *condition register* continuously monitors the state of the instrument. The bits in the condition register are updated in real time and the bits are not latched or buffered. This is a read-only register and bits are not cleared when you read the register. A query of a condition register returns a decimal value which corresponds to the binary-weighted sum of all bits set in that register.

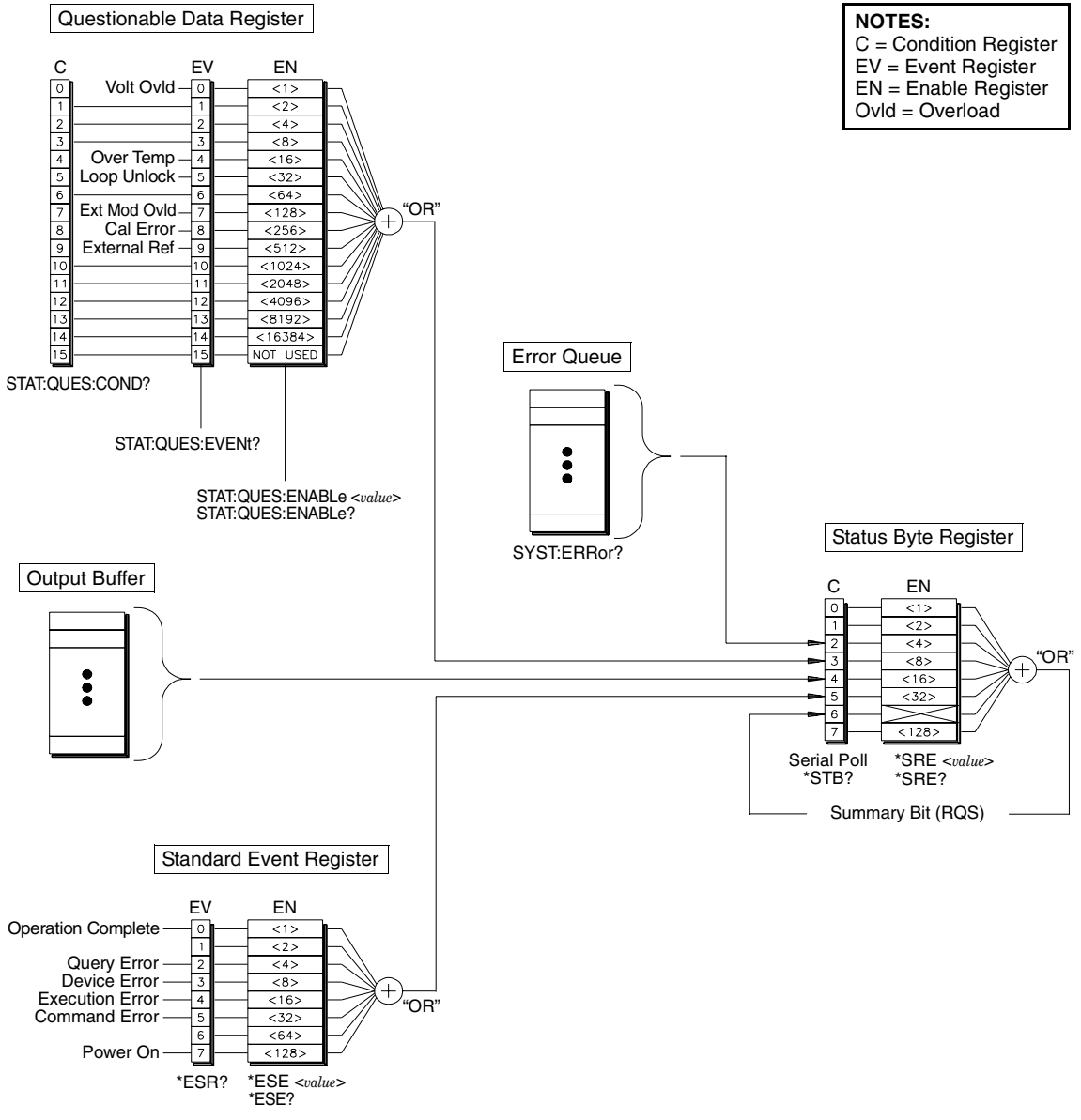
What is an Event Register?

An *event register* latches the various events from changes in the condition register. There is no buffering in this register; while an event bit is set, subsequent events corresponding to that bit are ignored. This is a read-only register. Once a bit is set, it remains set until cleared by query command (such as `STAT:QUES:EVEN?`) or a `*CLS` (clear status) command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in that register.

What is an Enable Register?

An *enable register* defines which bits in the event register will be reported to the Status Byte register group. You can write to or read from an enable register. A `*CLS` (clear status) command *will not* clear the enable register but it does clear all bits in the event register. A `STAT:PRES` command clears all bits in the enable register. To enable bits in the enable register to be reported to the Status Byte register, you must write a decimal value which corresponds to the binary-weighted sum of the corresponding bits.

Agilent 33250A Status System



The Status Byte Register

The Status Byte summary register reports conditions from the other status registers. Data that is waiting in the function generator's output buffer is immediately reported on the "Message Available" bit (bit 4). Clearing an event register from one of the other register groups will clear the corresponding bits in the Status Byte condition register. Reading all messages from the output buffer, including any pending queries, will clear the "Message Available" bit. To set the enable register mask and generate an SRQ (service request), you must write a decimal value to the register using the `*SRE` command.

Bit Definitions – Status Byte Register

Bit Number	Decimal Value	Definition
0 *Not Used*	1	*Not Used. Returns "0".*
1 *Not Used*	2	*Not Used. Returns "0".*
2 Error Queue	**4**	One or more errors are stored in the Error Queue.
3 Questionable Data	**8**	One or more bits are set in the Questionable Data Register (bits must be enabled).
4 Message Available	**16**	Data is available in the instrument's output buffer.
5 Standard Event	**32**	One or more bits are set in the Standard Event Register (bits must be enabled).
6 Master Summary	**64**	One or more bits are set in the Status Byte Register (bits must be enabled).
7 *Not Used*	128	*Not Used. Returns "0".*

The Status Byte *condition register* is cleared when:

- You execute the `*CLS` (clear status) command.
- You read the event register from one of the other register groups (only the corresponding bits are cleared in the condition register).

The Status Byte *enable register* is cleared when:

- You execute the `*SRE 0` command.
- You turn the power on and have previously configured the function generator to clear the enable register using the `*PSC 1` command. Note that the enable register *will not* be cleared at power-on if you have configured the function generator using the `*PSC 0` command.

Using Service Request (SRQ) and Serial Poll

You must configure your computer to respond to the IEEE-488 service request (SRQ) interrupt to use this capability. Use the Status Byte enable register (`*SRE` command) to select which condition bits will assert the IEEE-488 SRQ line. If bit 6 (RQS) transitions from a "0" to a "1", then an IEEE-488 service request message is sent to your computer. The computer may then poll the instruments on the interface bus to identify which one is asserting the service request line (that is, the instrument with bit 6 set in its Serial Poll response).

> When a Serial Poll is issued, bit 6 (RQS) is cleared in the Serial Poll response (no other bits are affected) and the service request line is cleared. The "Master Summary Bit" in the *STB? response is not cleared.

To obtain the Serial Poll response, send an IEEE-488 Serial Poll message. The instrument sends a one-byte binary response. Performing a Serial Poll is handled automatically by the IEEE-488 bus interface hardware.

> Unlike ASCII commands, and some other GPIB commands, a Serial Poll is executed immediately and does not involve the instrument's main processor. Therefore, the status indicated by a Serial Poll may not necessarily indicate the effect of the most recent command. Use the *OPC? command to guarantee that commands previously sent to the instrument have completed before executing a Serial Poll.

Using *STB? to Read the Status Byte

The `*STB?` command is similar to a Serial Poll but it is processed like any other ASCII instrument command. The `*STB?` command returns the same result as a Serial Poll but bit 6 *is not* cleared as long as the enabled conditions remain.

The `*STB?` command is not handled automatically by the IEEE-488 bus interface hardware and will be executed only after previous commands have completed. You cannot clear an SRQ using the `*STB?` command.

Using the Message Available Bit (MAV)

You can use the Status Byte "Message Available" bit (bit 4) to determine when data is available to read into your computer. The instrument subsequently clears bit 4 only after all messages have been read from the output buffer.

To Interrupt Your Computer Using SRQ

1. Send a Device Clear message to return the function generator to a responsive state and clear it's output buffer (e.g., `CLEAR 710`).
2. Clear the event registers and error queue using the `*CLS` command.
3. Set up the enable register masks. Use the `*ESE` command to configure the Standard Event enable register and the `*SRE` command to configure the Status Byte enable register.
4. Send the `*OPC?` command and read the result to ensure synchronization.
5. Enable your computer's IEEE-488 SRQ interrupt.

To Determine When a Command Sequence is Completed

1. Send a Device Clear message to return the function generator to a responsive state and clear it's output buffer (e.g., `CLEAR 710`).
2. Clear the event registers and error queue using the `*CLS` command.
3. Enable the "Operation Complete" bit (bit 0) in the Standard Event register by executing the `*ESE 1` command.
4. Send the `*OPC?` command and read the result to ensure synchronization.
5. Execute your command string to program the desired configuration, and then execute the `*OPC` command as the last command. When the command sequence is completed, the "Operation Complete" bit (bit 0) is set in the Standard Event register.
6. Use a Serial Poll to check to see when bit 5 (routed from the Standard Event register) is set in the Status Byte condition register. You could also configure the function generator for an SRQ interrupt by sending `*SRE 32` (Status Byte enable register, bit 5).

The Questionable Data Register

The Questionable Data register group provides information about the quality or integrity of the function generator. Any or all of these conditions can be reported to the Questionable Data summary bit through the *enable register*. To set the enable register mask, you must write a decimal value to the register using the `STAT:QUES:ENABle` command.

Bit Definitions – Questionable Data Register

Bit Number	Decimal Value	Definition
0 Voltage Overload	**1**	Voltage overload on OUTPUT connector. The Output has been disabled.
1 *Not Used*	2	*Not Used. Returns "0".*
2 *Not Used*	4	*Not Used. Returns "0".*
3 *Not Used*	8	*Not Used. Returns "0".*
4 Over Temperature	**16**	Internal temperature is too high and a power shutdown is likely.
5 Loop Unlocked	**32**	Function generator has lost phase lock. Frequency accuracy will be affected.
6 *Not Used*	64	*Not Used. Returns "0".*
7 External Mod Overload	**128**	Voltage overload on MOD IN connector.
8 Calibration Error	**256**	Error occurred during cal, or cal memory lost, or calibration is unsecured.
9 External Reference	**512**	External timebase is being used.
10 *Not Used*	1024	*Not Used. Returns "0".*
11 *Not Used*	2048	*Not Used. Returns "0".*
12 *Not Used*	4096	*Not Used. Returns "0".*
13 *Not Used*	8192	*Not Used. Returns "0".*
14 *Not Used*	16384	*Not Used. Returns "0".*
15 *Not Used*	32768	*Not Used. Returns "0".*

The Questionable Data *event register* is cleared when:

- You execute the `*CLS` (clear status) command.
- You query the event register using the `STAT:QUES:EVEN?` command.

The Questionable Data *enable register* is cleared when:

- You turn on the power (the `*PSC` command does not apply).
- You execute the `STAT:PRES` command.
- You execute the `STAT:QUES:ENAB 0` command.

The Standard Event Register

The Standard Event register group reports the following types of events: power-on detected, command syntax errors, command execution errors, self-test or calibration errors, query errors, or the `*OPC` command has been executed. Any or all of these conditions can be reported to the Standard Event summary bit through the *enable register*. To set the enable register mask, you must write a decimal value to the register using the `*ESE` command.

Bit Definitions – Standard Event Register

Bit Number	Decimal Value	Definition
0 Operation Complete	**1**	All commands prior to and including *OPC have completed and the overlapped command (e.g., *TRG for burst) has completed.
1 *Not Used*	2	*Not Used. Returns "0".*
2 Query Error	**4**	The instrument tried 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 and output buffers are full.
3 Device Error	**8**	A self-test, cal, or other device-specific error has occurred (see chapter 5).
4 Execution Error	**16**	An execution error has occurred (see chapter 5).
5 Command Error	**32**	A command syntax error has occurred (see chapter 5).
6 *Not Used*	64	*Not Used. Returns "0".*
7 Power On	**128**	Power has been cycled on since the last time the event register was read or cleared.

The Standard *event register* is cleared when:

- You execute the `*CLS` command.
- You query the event register using the `*ESR?` command.

The Standard Event *enable register* is cleared when:

- You execute the `*ESE 0` command.
- You turn the power on and have previously configured the function generator to clear the enable register using the `*PSC 1` command. Note that the enable register *will not* be cleared at power-on if you have configured the function generator using the `*PSC 0` command.

Status Reporting Commands

An application program is included in chapter 6 which shows the use of the Status System Registers. See page 275 for more information.

Status Byte Register Commands

See the table on page 227 for the register bit definitions.

***STB?**

Query the summary (condition) register in this register group. This command is similar to a Serial Poll but it is processed like any other instrument command. This command returns the same result as a Serial Poll but the "Master Summary" bit (bit 6) *is not* cleared by the `*STB?` command.

***SRE** *<enable value>*
***SRE?**

Enable bits in the Status Byte to generate a Service Request. To enable specific bits, you must write a decimal value which corresponds to the binary-weighted sum of the bits in the register. The selected bits are summarized in the "Master Summary" bit (bit 6) of the Status Byte Register. If any of the selected bits change from "0" to "1", a Service Request signal is generated. The `*SRE?` query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the `*SRE` command.

- A `*CLS` (clear status) *does not* clear the enable register but it does clear all bits in the event register.
- A `STATus:PRESet` *does not* clear the bits in the Status Byte enable register.
- A `*PSC 0` preserves the contents of the enable register through power cycles.

Questionable Data Register Commands

See the table on page 231 for the register bit definitions.

STATus:QUEStionable:CONDition?

Query the condition register in this group. This is a read-only register and bits are not cleared when you read the register. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

STATus:QUEStionable[:EVENt]?

Query the event register in this register group. This is a read-only register. Once a bit is set, it remains set until cleared by this command or `*CLS` (clear status) command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

STATus:QUEStionable:ENABle *<enable value>*
STATus:QUEStionable:ENABle?

Enable bits in the enable register in this register group. The selected bits are then reported to the Status Byte. A `*CLS` (clear status) *will not* clear the enable register but it does clear all bits in the event register. The `STATus:PRESet` command clears all bits in the enable register. To enable bits in the enable register, you must write a decimal value which corresponds to the binary-weighted sum of the bits you wish to enable in the register.

The `:ENAB?` query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the `STAT:QUES:ENAB` command.

Standard Event Register Commands

See the table on page 233 for the register bit definitions.

***ESR?**

Query the Standard Event Status Register. Once a bit is set, it remains set until cleared by a `*CLS` (clear status) command or queried by this command. A query of this register returns a decimal value which corresponds to the binary-weighted sum of all bits set in the register.

***ESE** *<enable value>*
***ESE?**

Enable bits in the Standard Event Status Register to be reported in the Status Byte. The selected bits are summarized in the "Standard Event" bit (bit 5) of the Status Byte Register. The `*ESE?` query returns a decimal value which corresponds to the binary-weighted sum of all bits enabled by the `*ESE` command.

- A `*CLS` (clear status) *does not* clear the enable register but it does clear all bits in the event register.
- A `STATus:PRESet` *does not* clear the bits in the Status Byte enable register.
- A `*PSC 0` preserves the contents of the enable register through power cycles.

Miscellaneous Status Register Commands

***CLS**

Clear the event register in all register groups. This command also clears the error queue and cancels a `*OPC` operation.

STATus:PRESet

Clear all bits in the Questionable Data enable register and the Standard Operation enable register.

***PSC** {**0**|**1**}
***PSC?**

Power-On Status Clear. Clear the Standard Event enable register and Status Byte condition register at power on (`*PSC 1`). When `*PSC 0` is in effect, these two registers *are not* cleared at power on. The default is `*PSC 1`. The `*PSC?` query returns the power-on status clear setting. Returns "0" (do not clear at power on) or "1" (clear at power on).

***OPC**

Set the "Operation Complete" bit (bit 0) in the Standard Event register after the previous commands have completed. When used with a bus-triggered sweep or burst, you may have the opportunity to execute commands *after* the `*OPC` command and *before* the "Operation Complete" bit is set in the register.

Calibration Commands

For an overview of the calibration features of the function generator, refer to "Calibration Overview" in chapter 3 starting on page 123. For a detailed discussion of the function generator's calibration procedures, refer to chapter 4 in the Agilent 33250A *Service Guide*.

CALibration:SECure:STATe {**OFF**|**ON**},<*code*>
CALibration:SECure:STATe?

Unsecure or secure the instrument for calibration. The calibration code may contain up to 12 characters. The :STAT? query returns "0" (OFF) or "1" (ON).

CALibration:SETup <**0**|**1**|**2**|**3**| **. . .** |**115**>
CALibration:SETup?

Configure the function generator's internal state for each of the calibration steps to be performed. The :SET? query reads the calibration setup number and returns a value from "0" to "115".

CALibration:VALue <*value*>
CALibration:VALue?

Specify the value of the known calibration signal as outlined in the calibration procedures in the Agilent 33250A *Service Guide*. Use the CAL:SET command to configure the function generator's internal state for each of the calibration steps to be performed. The :VAL? query returns a number in the form "+1.0000000000000E+01".

CALibration?

Perform a calibration of the instrument using the specified calibration value (CAL:VAL command). Before you can calibrate the function generator, you must unsecure it by entering the correct security code. Returns "0" (PASS) or "1" (FAIL).

CALibration:SECure:CODE *<new code>*

Enter a new security code. To change the security code, you must first unsecure the function generator using the old security code, and then enter a new code. The security code is stored in *non-volatile* memory.

- The calibration code may contain up to 12 characters. The first character *must* be a letter (A-Z), but the remaining characters can be letters, numbers (0-9), or the underscore character (" _ "). You do not have to use all 12 characters, but the first character *must* always be a letter.

CALibration:COUNt?

Query the instrument to determine the number of times it has been calibrated. Note that your instrument was calibrated before it left the factory. When you receive your instrument from the factory, be sure to read the count to determine its initial value.

- The calibration count is stored in *non-volatile* memory. The count increments up to a maximum of "65,535" after which it rolls over to "0". Since the value increments by one for each calibration point, a complete calibration may increase the value by many counts.

CALibration:STRing *<quoted string>*
CALibration:STRing?

Store a message in *non-volatile* calibration memory. Storing a message will overwrite any message previously stored in memory. The `:STR?` query reads the calibration message and returns a quoted string.

- The calibration message may contain up to 40 characters (additional characters are truncated). An example is shown below.

```
CAL:STR 'Cal Due: 01 June 2001'
```

- You can record a calibration message *only* from the remote interface and *only* when the instrument is unsecured. You can read the message either from the front panel or over the remote interface. You can *read* the calibration message whether the instrument is secured or unsecured.

An Introduction to the SCPI Language

SCPI (*Standard Commands for Programmable Instruments*) is an ASCII-based instrument command language designed for test and measurement instruments. Refer to "Simplified Programming Overview" starting on page 142, for an introduction to the basic techniques used to program the function generator over the remote interface.

SCPI commands are based on a hierarchical structure, also known as a *tree system*. In this system, associated commands are grouped together under a common node or root, thus forming *subsystems*. A portion of the SOURce subsystem is shown below to illustrate the tree system.

```
SOURce:
   FREQuency
      :STARt {<frequency>|MINimum|MAXimum}
      :STARt? [MINimum|MAXimum]

   FREQuency
      :STOP {<frequency>|MINimum|MAXimum}
      :STOP? [MINimum|MAXimum]

   SWEep
      :SPACing {LINear|LOGarithmic}
      :SPACing?

   SWEep
      :TIME {<seconds>|MINimum|MAXimum}
      :TIME? [MINimum|MAXimum]

   SWEep
      :STATe {OFF|ON}
      :STATe?
```

SOURce is the root keyword of the command, FREQuency and SWEep are second-level keywords, and STARt and STOP are third-level keywords. A colon (**:**) separates a command keyword from a lower-level keyword.

Command Format Used in This Manual

The format used to show commands in this manual is illustrated below:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

The command syntax shows most commands (and some parameters) as a mixture of upper- and lower-case letters. The upper-case letters indicate the abbreviated spelling for the command. For shorter program lines, you can send the abbreviated form. For better program readability, you can send the long form.

For example, in the above syntax statement, `FREQ` and `FREQUENCY` are both acceptable forms. You can use upper- or lower-case letters. Therefore, `FREQUENCY`, `freq`, and `Freq` are all acceptable. Other forms, such as `FRE` and `FREQUEN`, will generate an error.

- *Braces* (**{ }**) enclose the parameter choices for a given command string. The braces *are not* sent with the command string.
- A *vertical bar* (**|**) separates multiple parameter choices for a given command string.
- *Triangle brackets* (**< >**) indicate that you must specify a value for the enclosed parameter. For example, the above syntax statement shows the *frequency* parameter enclosed in triangle brackets. The brackets *are not* sent with the command string. You must specify a value for the parameter (such as `"FREQ 5000"`).
- Some parameters are enclosed in *square brackets* (**[ ]**). The square brackets indicate that the parameter is optional and can be omitted. The brackets *are not* sent with the command string. If you do not specify a value for an optional parameter, the function generator chooses a default value.

Command Separators

A *colon* (**:**) is used to separate a command keyword from a lower-level keyword. You must insert a *blank space* to separate a parameter from a command keyword. If a command requires more than one parameter, you must separate adjacent parameters using a *comma* as shown below:

```
"APPL:SIN 5 KHZ, 3.0 VPP, -2.5 V"
```

A *semicolon* (**;**) is used to separate commands within the same subsystem, and can also minimize typing. For example, sending the following command string:

```
"FREQ:START 10; STOP 1000"
```

... is the same as sending the following two commands:

```
"FREQ:START 10"
"FREQ:STOP 1000"
```

Use a colon *and* a semicolon to link commands from *different* subsystems. For example, in the following command string, an error is generated if you do not use both the colon *and* semicolon:

```
"SWE:STAT ON;:TRIG:SOUR EXT"
```

Using the *MIN* and *MAX* Parameters

You can substitute “MINimum” or “MAXimum” in place of a parameter for many commands. For example, consider the following command:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

Instead of selecting a specific frequency, you can substitute MIN to set the frequency to its minimum value or MAX to set the frequency to its maximum value.

Querying Parameter Settings

You can query the current value of most parameters by adding a *question mark* ("**?**") to the command. For example, the following command sets the output frequency to 5 kHz:

```
"FREQ 5000"
```

You can then query the frequency value by executing:

```
"FREQ?"
```

You can also query the minimum or maximum frequency allowed with the present function as follows:

```
"FREQ? MIN"
"FREQ? MAX"
```


SCPI Command Terminators

A command string sent to the function generator *must* terminate with a <*new line*> character. The IEEE-488 *EOI* (End-Or-Identify) message is interpreted as a <*new line*> character and can be used to terminate a command string in place of a <*new line*> character. A <*carriage return*> followed by a <*new line*> is also accepted. Command string termination will *always* reset the current SCPI command path to the root level.

IEEE-488.2 Common Commands

The IEEE-488.2 standard defines a set of *common commands* that perform functions such as reset, self-test, and status operations. Common commands always begin with an asterisk (*****), are three characters in length, and may include one or more parameters. The command keyword is separated from the first parameter by a *blank space*. Use a *semicolon* (**;**) to separate multiple commands as shown below:

```
"*RST; *CLS; *ESE 32; *OPC?"
```

SCPI Parameter Types

The SCPI language defines several different data formats to be used in program messages and response messages.

Numeric Parameters Commands that require numeric parameters will accept all commonly used decimal representations of numbers including optional signs, decimal points, and scientific notation. Special values for numeric parameters like `MINimum`, `MAXimum`, and `DEFault` are also accepted. You can also send engineering unit suffixes with numeric parameters (e.g., `Mhz` or `Khz`). If only specific numeric values are accepted, the function generator will automatically round the input numeric parameters. The following command uses a numeric parameter:

```
FREQuency {<frequency>|MINimum|MAXimum}
```

Discrete Parameters Discrete parameters are used to program settings that have a limited number of values (like `BUS`, `IMMediate`, `EXTernal`). They have a short form and a long form just like command keywords. You can mix upper- and lower-case letters. Query responses will always return the short form in all upper-case letters. The following command uses discrete parameters:

```
SWEep:SPACing {LINear|LOGarithmic}
```

Boolean Parameters Boolean parameters represent a single binary condition that is either true or false. For a false condition, the function generator will accept “OFF” or “0”. For a true condition, the function generator will accept “ON” or “1”. When you query a boolean setting, the instrument will *always* return “0” or “1”. The following command uses a boolean parameter:

```
AM:STATe {OFF|ON}
```

String Parameters String parameters can contain virtually any set of ASCII characters. A string *must* begin and end with matching quotes; either with a single quote or a double quote. You can include the quote delimiter as part of the string by typing it twice without any characters in between. The following command uses a string parameter:

```
DISPlay:TEXT <quoted string>
```

Using Device Clear

Device Clear is an IEEE-488 low-level bus message that you can use to return the function generator to a responsive state. Different programming languages and IEEE-488 interface cards provide access to this capability through their own unique commands. The status registers, the error queue, and all configuration states are left unchanged when a Device Clear message is received. Device Clear performs the following actions.

- The function generator's input and output buffers are cleared.
- The function generator is prepared to accept a new command string.
- For RS-232 operation, sending the <*Break*> character will perform the equivalent operations of the IEEE-488 Device Clear message.
- An overlapped command, if any, will be terminated with no "Operation Complete" indication (applies to the `*TRG` command). Any sweep or burst in progress will be aborted immediately.

Error Messages

- Errors are retrieved in first-in-first-out (FIFO) order. The first error returned is the first error that was stored. Errors are cleared as you read them. The function generator beeps once each time an error is generated (unless you have disabled the beeper).
- If more than 20 errors have occurred, the last error stored in the queue (the most recent error) is replaced with "*Queue overflow*". No additional errors are stored until you remove errors from the queue. If no errors have occurred when you read the error queue, the function generator responds with "*No error*".
- The error queue is cleared by the `*CLS` (clear status) command or when the power is cycled. The errors are also cleared when you read the error queue. The error queue *is not* cleared by an instrument reset (`*RST` command).
- *Front-Panel Operation*: Press Help and then select the topic entitled "*View the remote command error queue*" (topic number 2). Then press the **SELECT** softkey to view the errors in the error queue. As shown below, the first error in the list (i.e., the error at the top of the list) is the first error that was generated.



- *Remote Interface Operation*:

  `SYSTem:ERRor?` *Reads one error from the error queue*

  Errors have the following format (the error string may contain up to 255 characters).

  ```
  -113,"Undefined header"
  ```

Command Errors

-101 **Invalid character**
An invalid character was found in the command string. You may have used an invalid character such as #, $, or % in the command header or within a parameter. *Example*: `TRIG:SOUR BUS#`

-102 **Syntax error**
Invalid syntax was found in the command string. You may have inserted a blank space before or after a colon in the command header, or before a comma. *Example*: `APPL:SIN ,1`

-103 **Invalid separator**
An invalid separator was found in the command string. You may have used a comma instead of a colon, semicolon, or blank space – or you may have used a blank space instead of a comma. *Examples*: `TRIG:SOUR,BUS` *or* `APPL:SIN 1 1000`

-105 **GET not allowed**
A Group Execute Trigger (`GET`) is not allowed within a command string.

-108 **Parameter not allowed**
More parameters were received than were expected for the command. You may have entered an extra parameter or added a parameter to a command that does not require a parameter. *Example*: `APPL? 10`

-109 **Missing parameter**
Fewer parameters were received than were expected for this command. You have omitted one or more parameters that are required for this command. *Example*: `OUTP:LOAD`

-112 **Program mnemonic too long**
A command header was received which contained more than the maximum 12 characters allowed. This error is also reported when a character-type parameter is too long. *Example*: `OUTP:SYNCHRONIZATION ON`

-113 **Undefined header**
A command was received that is not valid for this instrument. You may have misspelled the command or it may not be a valid command. If you are using the short form of the command, remember that it may contain up to four letters. *Example*: `TRIGG:SOUR BUS`

-123 **Exponent too large**
A numeric parameter was found whose exponent was larger than 32,000. *Example*: `BURS:NCYCL 1E34000`

-124 **Too many digits**
A numeric parameter was found whose mantissa contained more than 255 digits, excluding leading zeros.

-128 **Numeric data not allowed**
A numeric parameter was received but a character string was expected. *Example*: `DISP:TEXT 123`

-131 **Invalid suffix**
A suffix was incorrectly specified for a numeric parameter. You may have misspelled the suffix. *Example*: `SWE:TIME 0.5 SECS`

-138 **Suffix not allowed**
A suffix is not supported for this command. *Example*: `BURS:NCYC 12 CYC`

-148 **Character data not allowed**
A discrete parameter was received but a character string or numeric parameter was expected. Check the list of parameters to verify that you have used a valid parameter type. *Example*: `DISP:TEXT ON`

-151 **Invalid string data**
An invalid character string was received. Check to see if you have enclosed the character string in quotation marks and verify that the string contains valid ASCII characters.
Example: `DISP:TEXT 'TESTING` (the ending quote is missing)

-158 **String data not allowed**
A character string was received but is not allowed for this command. Check the list of parameters to verify that you have used a valid parameter type. *Example*: `BURS:NCYC 'TEN'`

-161 **Invalid block data**
Applies only to the `DATA:DAC VOLATILE` *command*.
For a definite-length block, the number of bytes of data sent does not match the number of bytes that you specified in the block header. For an indefinite-length block, an EOI (End-or-Identify) was received without an accompanying <*new line*> character.

-168 **Block data not allowed**
Data was sent to the function generator in definite-length block format but this command does not accept this format. Check to see if you have sent the correct data type with the command. *Example*: `BURS:NCYC #10`

-170 to -178 **Expression errors**
The function generator does not accept mathematical expressions.

Execution Errors

-211 **Trigger ignored**
A Group Execute Trigger (`GET`) or `*TRG` was received but the trigger was ignored. Make sure that you have selected the proper trigger source and verify that the sweep or burst mode is enabled.

-223 **Too much data**
An arbitrary waveform was specified that contains more than 65,536 waveform points. Verify the number of points in the `DATA VOLATILE` or `DATA:DAC VOLATILE` command.

-221 **Settings conflict;**
turned off infinite burst to allow immediate trigger source
An infinite count burst is allowed only when the *external* or *bus* (software) trigger source is selected. The burst count has been set to the maximum value (1,000,000 cycles).

-221 **Settings conflict;**
infinite burst changed trigger source to BUS
An infinite count burst is allowed only when the *external* or *bus* (software) trigger source is selected. Sending the `BURS:NCYC INF` command has automatically changed the trigger source from *immediate* to *bus*.

-221 **Settings conflict;**
burst period increased to fit entire burst
The number of cycles specified by the `BURS:NCYC` command takes priority over the burst period (as long as the burst period is not at its maximum value). The function generator has increased the burst period to accommodate the specified burst count or waveform frequency.

-221 **Settings conflict;**
burst count reduced
Since the burst period is currently at its maximum, the function generator has reduced the burst count to allow the specified waveform frequency.

-221 **Settings conflict;**
trigger delay reduced to fit entire burst
The function generator has reduced the trigger delay in order to maintain the current burst period and burst count. The trigger delay is the time between the receipt of the trigger and the start of the burst waveform.

-221 **Settings conflict;**
triggered burst not available for noise
You cannot use the noise function in the *triggered* burst mode. Noise is allowed only in the *gated* burst mode.

-221 **Settings conflict;**
amplitude units changed to Vpp due to high-Z load
The output units (`VOLT:UNIT` command) cannot be set to dBm if the output termination is currently set to "high impedance" (`OUTP:LOAD` command). The function generator has converted the units to Vpp.

-221 **Settings conflict;**
trigger output disabled by trigger external
When the *external* trigger source is selected (`TRIG:SOUR EXT` command), the function generator automatically disables the "trigger out" signal. The rear-panel *Trig* connector cannot be used for both operations at the same time (an externally-triggered waveform uses the same connector to trigger the sweep or burst).

-221 **Settings conflict;**
trigger output connector used by FSK
If you have enabled FSK and have selected the *external* trigger source (`FSK:SOUR EXT` command), the "trigger out" signal cannot be enabled (`OUTP:TRIG ON` command). The rear-panel *Trig* connector cannot be used for both operations at the same time.

-221 **Settings conflict;**
trigger output connector used by burst gate
If you have selected the gated burst mode (`BURS:MODE GAT` command) with burst enabled, the "trigger out" signal cannot be enabled (`OUTP:TRIG ON` command). The rear-panel *Trig* connector cannot be used for both operations at the same time.

-221 **Settings conflict;**
trigger output connector used by trigger external
When the *external* trigger source is selected (`TRIG:SOUR EXT` command), the function generator automatically disables the "trigger out" signal. The rear-panel *Trig* connector cannot be used for both operations at the same time.

-221 **Settings conflict;**
frequency reduced for user function
For arbitrary waveforms, the output frequency is limited to 25 MHz. When you change to the arbitrary waveform function (`APPL:USER` or `FUNC:USER` command) from a function that allows a higher frequency, the function generator will automatically adjust the frequency to 25 MHz.

-221 **Settings conflict;**
frequency reduced for pulse function
For pulse waveforms, the output frequency is limited to 50 MHz. When you change to the pulse function (`APPL:PULS` or `FUNC:PULS` command) from a function that allows a higher frequency, the function generator will automatically adjust the frequency to 50 MHz.

-221 **Settings conflict;**
frequency reduced for ramp function
For ramp waveforms, the output frequency is limited to 1 MHz. When you change to the ramp function (`APPL:RAMP` or `FUNC:RAMP` command) from a function that allows a higher frequency, the function generator will automatically adjust the frequency to 1 MHz.

-221 **Settings conflict;**
frequency made compatible with burst mode
For an internally-triggered burst, the output frequency is limited to a minimum of 2 mHz. The function generator has adjusted the frequency to be compatible with the current settings.

-221 **Settings conflict;**
frequency made compatible with FM
When FM is enabled, the output frequency of the carrier waveform is limited to a minimum of 5 Hz. The function generator has adjusted the frequency to be compatible with the current settings.

-221 **Settings conflict;**
burst turned off by selection of other mode or modulation
The function generator will allow only one modulation, sweep, or burst mode to be enabled at the same time. When you enable a modulation, sweep, or burst mode, all other modes are turned off.

-221 **Settings conflict;**
FSK turned off by selection of other mode or modulation
The function generator will allow only one modulation, sweep, or burst mode to be enabled at the same time. When you enable a modulation, sweep, or burst mode, all other modes are turned off.

-221 **Settings conflict;**
FM turned off by selection of other mode or modulation
The function generator will allow only one modulation, sweep, or burst mode to be enabled at the same time. When you enable a modulation, sweep, or burst mode, all other modes are turned off.

-221 **Settings conflict;**
AM turned off by selection of other mode or modulation
The function generator will allow only one modulation, sweep, or burst mode to be enabled at the same time. When you enable a modulation, sweep, or burst mode, all other modes are turned off.

-221 **Settings conflict;**
sweep turned off by selection of other mode or modulation
The function generator will allow only one modulation, sweep, or burst mode to be enabled at the same time. When you enable a modulation, sweep, or burst mode, all other modes are turned off.

-221 **Settings conflict;**
not able to modulate this function
The function generator cannot generate a modulated waveform using the pulse, noise, or dc voltage functions.

-221 **Settings conflict;**
not able to sweep this function
The function generator cannot generate a sweep using the pulse, noise, or dc voltage functions.

-221 **Settings conflict;**
not able to burst this function
The function generator cannot generate a burst using the dc voltage function.

-221 **Settings conflict;**
not able to modulate noise, modulation turned off
The function generator cannot generate a modulated waveform using the noise function. The selected modulation mode has been turned off.

-221 **Settings conflict;**
not able to sweep pulse, sweep turned off
The function generator cannot generate a sweep using the pulse function. The sweep mode has been turned off.

-221 **Settings conflict;**
not able to modulate dc, modulation turned off
The function generator cannot generate a modulated waveform using the dc voltage function. The selected modulation mode has been turned off.

-221 **Settings conflict;**
not able to sweep dc, modulation turned off
The function generator cannot generate a sweep using the dc voltage function. The sweep mode has been turned off.

-221 **Settings conflict;**
not able to burst dc, burst turned off
The function generator cannot generate a burst using the dc voltage function. The burst mode has been turned off.

-221 **Settings conflict;**
not able to sweep noise, sweep turned off
The function generator cannot generate a sweep using the noise function. The sweep mode has been turned off.

-221 **Settings conflict;**
pulse width decreased due to period
For a pulse waveform, the function generator will automatically adjust the waveform parameters in the following order as needed to generate a valid pulse: (1) edge time, (2) pulse width, and then (3) period.

In this case, the function generator has decreased the pulse width to accommodate the specified period (the edge time is already at its minimum setting).

-221 **Settings conflict;**
edge time decreased due to period
For a pulse waveform, the function generator will automatically adjust the waveform parameters in the following order as needed to generate a valid pulse: (1) edge time, (2) pulse width, and then (3) period.

In this case, the function generator has decreased the edge time to accommodate the specified period and preserve the pulse width setting.

-221 **Settings conflict;**
edge time decreased due to pulse width
For a pulse waveform, the function generator will automatically adjust the waveform parameters in the following order as needed to generate a valid pulse: (1) edge time, (2) pulse width, and then (3) period.

In this case, the function generator has decreased the edge time to accommodate the specified pulse width.

Pulse Width $\geq$ 1.6 X Edge Time

-221 **Settings conflict;**
amplitude changed due to function
In some cases, the amplitude limits are determined by the output units currently selected. This may occur when the units are *Vrms* or *dBm* due to the differences in crest factor for the various output functions.
For example, if you output a 5 Vrms square wave (into 50 ohms) and then change to the sine wave function, the function generator will automatically adjust the output amplitude to 3.536 Vrms (upper limit for sine waves in Vrms).

-221 **Settings conflict;**
offset changed on exit from dc function
In the dc voltage function, the voltage level is controlled by adjusting the offset voltage (the current amplitude is ignored). When you select a different function, the function generator will adjust the offset voltage as needed to be compatible with the current amplitude setting.

-221 **Settings conflict;**
FM deviation cannot exceed carrier
The carrier frequency must always be greater than or equal to the frequency deviation. If you attempt to set the deviation to a value greater than the carrier frequency (with FM enabled), the function generator will automatically adjust the deviation to the maximum value allowed with the present carrier frequency.

-221 **Settings conflict;**
FM deviation exceeds max frequency
The sum of the carrier frequency and deviation must be less than or equal to the maximum frequency for the selected function ***plus 100 kHz*** (80.1 MHz for sine and square, 1.1 MHz for ramp, and 25.1 MHz for arbitrary waveforms). If you attempt to set the deviation to a value that is not valid, the function generator will automatically adjust it to the maximum value allowed with the present carrier frequency.

-221 **Settings conflict;**
frequency forced duty cycle change
If the square wave function is selected and you change to a frequency that cannot produce the current duty cycle, the duty cycle is automatically adjusted to the maximum value for the new frequency. For example, if you currently have the duty cycle set to 70% and then change the frequency to 60 MHz, the function generator will automatically adjust the duty cycle to 50% (upper limit for this frequency).

Duty Cycle: 20% to 80% ($frequency \leq 25$ MHz)
40% to 60% (25 MHz $< frequency \leq 50$ MHz)
50% ($frequency > 50$ MHz)

-221 **Settings conflict;**
selected arb is missing, changing selection to default
If you delete an arbitrary waveform from non-volatile memory after storing the instrument state, the waveform data is lost and the function generator *will not* output the waveform when the state is recalled. The built-in “exponential rise” waveform is output in place of the deleted waveform.

-221 **Settings conflict;**
offset changed due to amplitude
The relationship between offset voltage and output amplitude is shown below. Vmax is the maximum peak voltage for the selected output termination (5 volts for a 50Ω load or 10 volts for a high-impedance load). If the specified offset voltage is not valid, the function generator will automatically adjust it to the maximum dc voltage allowed with the amplitude specified.

$$|\text{Voffset}| \leq \text{Vmax} - \frac{\text{V}pp}{2}$$

-221 **Settings conflict;**
amplitude changed due to offset
The relationship between output amplitude and offset voltage is shown below. Vmax is the maximum peak voltage for the selected output termination (5 volts for a 50Ω load or 10 volts for a high-impedance load). If the specified amplitude not valid, the function generator will automatically adjust it to the maximum value allowed with the offset voltage specified.

$$\text{Vpp} \leq 2 \times (\text{Vmax} - |\text{Voffset}|)$$

-221 **Settings conflict;**
low level changed due to high level
You can set the levels to a positive or negative value, but note that the high level *must* always be greater than the low level. If you specify a high level that is less than the low level, the function generator will automatically set the low level to be 1 mV less than the high level.

-221 **Settings conflict;**
high level changed due to low level
You can set the levels to a positive or negative value, but note that the high level *must* always be greater than the low level. If you specify a low level that is greater than the high level, the function generator will automatically set the high level to be 1 mV greater than the low level.

-222 **Data out of range;**
value clipped to upper limit
The specified parameter is outside of the function generator's capability. The function generator has adjusted the parameter to the maximum value allowed. *Example*: `PHAS 1000`

-222 **Data out of range;**
value clipped to lower limit
The specified parameter is outside of the function generator's capability. The function generator has adjusted the parameter to the minimum value allowed. *Example*: `PHAS -1000`

-222 **Data out of range;**
pulse edge time limited by period
The specified edge time must fit within the specified period. The function generator will adjust the edge time as needed to accommodate the specified period.

-222 **Data out of range;**
pulse width limited by period; value clipped to ...
The specified pulse width must be less than the difference between the *period* and the *edge time* as shown below. The function generator will adjust the pulse width as needed to accommodate the specified period.

$$\text{Pulse Width} \leq \text{Period} - (1.6 \times \text{Edge Time})$$

-222 **Data out of range;**
pulse edge time limited by width; value clipped to ...
The specified edge time must fit within the specified pulse width as shown below. The function generator will adjust the edge time as needed to accommodate the specified pulse width.

$$\text{Edge Time} \leq 0.625 \times \text{Pulse Width}$$

-222 **Data out of range;**
period; value clipped to ...
This generic message indicates that the waveform period has been limited to an upper or lower boundary.

-222 **Data out of range;**
frequency; value clipped to ...
This generic message indicates that the waveform frequency has been limited to an upper or lower boundary.

-222 **Data out of range;**
user frequency; value clipped to upper limit
This generic message indicates that the waveform frequency has been limited to an upper boundary due to the selection of the arbitrary waveform function (`APPL:USER` or `FUNC:USER` command).

-222 **Data out of range;**
ramp frequency; value clipped to upper limit
This generic message indicates that the waveform frequency has been limited to an upper boundary due to the selection of the ramp waveform function (`APPL:RAMP` or `FUNC:RAMP` command).

-222 **Data out of range;**
pulse frequency; value clipped to upper limit
This generic message indicates that the waveform frequency has been limited to an upper boundary due to the selection of the pulse waveform function (`APPL:PULS` or `FUNC:PULS` command).

-222 **Data out of range;**
burst period; value clipped to ...
This generic message indicates that the burst period has been limited to an upper or lower boundary.

-222 **Data out of range;**
burst count; value clipped to ...
This generic message indicates that the burst count has been limited to an upper or lower boundary.

-222 **Data out of range;**
burst period limited by length of burst; value clipped to upper limit
It *is not* possible to specify a burst period which is too short for the function generator to output with the specified burst count and frequency (*see below*). If the burst period is too short, the function generator will automatically adjust it as needed to continuously re-trigger the burst.

$$\text{Burst Period} > \frac{\text{Burst Count}}{\text{Waveform Frequency}} + 200\ \text{ns}$$

-222 **Data out of range;**
burst count limited by length of burst; value clipped to lower limit
If the *Immediate* trigger source is selected (`TRIG:SOUR IMM` command), the burst count must be less than the product of the burst period and the waveform frequency as shown below.

$$\text{Burst Count} < \text{Burst Period} \times \text{Waveform Frequency}$$

-222 **Data out of range;**
amplitude; value clipped to ...
This generic message indicates that the waveform amplitude has been limited to an upper or lower boundary.

-222 **Data out of range;**
offset; value clipped to ...
This generic message indicates that the offset voltage has been limited to an upper or lower boundary.

-222 **Data out of range;**
frequency in burst mode; value clipped to ...
This generic message indicates that the frequency has been limited to an upper or lower boundary as dictated by the burst period.

-222 **Data out of range;**
frequency in FM; value clipped to ...
This generic message indicates that the carrier frequency has been limited to a lower boundary as determined by the `FM:DEV` command. The carrier frequency must always be greater than or equal to the frequency deviation.

-222 **Data out of range;**
marker confined to sweep span; value clipped to ...
This generic message indicates that the specified marker frequency is outside the range of the start frequency and stop frequency. The marker frequency *must* be between the specified start frequency and stop frequency. If you attempt to set the marker frequency to a frequency not in this range, the function generator will automatically set the marker frequency equal to the start frequency or stop frequency (whichever is closer). This error is generated only when both the sweep mode and the marker frequency are enabled.

-222 **Data out of range;**
pulse width; value clipped to ...
This generic message indicates that the desired pulse width is limited to an upper or lower boundary which is usually dictated by the period of the pulse waveform.

-222 **Data out of range;**
pulse edge time; value clipped to ...
This generic message indicates that the desired edge time is limited to an upper or lower boundary which is usually dictated by the pulse width and/or period of the pulse waveform.

-222 **Data out of range;**
FM deviation; value clipped to ...
This generic message indicates that the desired FM deviation is limited to a lower or upper boundary as set by the current function's frequency.

-222 **Data out of range;**
trigger delay; value clipped to upper limit
The trigger delay is limited to a maximum of 85 seconds. The trigger delay sets the time between the receipt of the trigger and the start of the burst waveform (used in *triggered* burst mode only).

-222 **Data out of range;**
trigger delay limited by length of burst; value clipped to upper limit
The sum of the specified trigger delay and the time required to complete the burst waveform must be less than the period of the burst.

-222 **Data out of range;**
duty cycle; value clipped to ...
The duty cycle is limited to values between 20% and 80% when the frequency is less than 25 MHz.

Duty Cycle: 20% to 80% (*frequency* ≤ 25 MHz)
40% to 60% (25 MHz < *frequency* ≤ 50 MHz)
50% (*frequency* > 50 MHz)

-222 **Data out of range;**
duty cycle limited by frequency; value clipped to upper limit
The duty cycle is limited to 50% when the frequency is greater than 50 MHz.

Duty Cycle: 20% to 80% (*frequency* ≤ 25 MHz)
40% to 60% (25 MHz < *frequency* ≤ 50 MHz)
50% (*frequency* > 50 MHz)

-313 **Calibration memory lost;**
memory corruption detected
The non-volatile memory used to store the function generator's calibration constants has detected a checksum error. This error may be the result of a device failure or extreme conditions such as lightning or strong magnetic fields.

-314 **Save/recall memory lost;**
memory corruption detected
The non-volatile memory used to store instrument states has detected a checksum error. This error may be the result of a device failure or extreme conditions such as lightning or strong magnetic fields.

-315 **Configuration memory lost;**
memory corruption detected
The non-volatile memory used to store the function generator's configuration settings (i.e., remote interface settings) has detected a checksum error. This error may be the result of a device failure or extreme conditions such as lightning or strong magnetic fields.

-350 **Queue overflow**
The error queue is full because more than 20 errors have occurred. No additional errors are stored until you remove errors from the queue. The error queue is cleared by the `*CLS` (clear status) command or when the power is cycled. The errors are also cleared when you read the queue. The error queue *is not* cleared by an instrument reset (`*RST` command).

-361 **Parity error in program message**
This error is most likely to occur when the function generator's parity setting (RS-232 interface) does not match the computer's parity setting. This error may also occur in the presence of noise on the RS-232 cable.

-362 **Framing error in program message**
This error is most likely to occur when the number of stop bits for the computer (RS-232 interface) does not match the function generator's setting (fixed at 1 stop bit).

-363 **Input buffer overrun**
Too many characters have been sent to the function generator while configured to use the RS-232 interface. This error typically occurs when you have selected no handshaking of data between the computer and the function generator. To avoid this error, select one of the handshake modes available for the 33250A (see "Remote Interface Configuration" on page 118 for more information).

Query Errors

-410 **Query INTERRUPTED**
A command was received but the output buffer contained data from a previous command (the previous data is lost).

-420 **Query UNTERMINATED**
The function generator was addressed to talk (i.e., to send data over the interface) but a command has not been received which sends data to the output buffer. For example, you may have executed an `APPLy` command (which does not generate data) and then attempted an "Enter" statement to read data from the interface.

-430 **Query DEADLOCKED**
A command was received which generates too much data to fit in the output buffer and the input buffer is also full. Command execution continues but all data is lost.

-440 **Query UNTERMINATED after indefinite response**
The `*IDN?` command must be the last query command within a command string. *Example*: `*IDN?;:SYST:VERS?`

Instrument Errors

501 to 504

501: Cross-isolation UART framing error
502: Cross-isolation UART overrun error
503: Cross-isolation UART parity error
504: Cross-isolation UART noise error
These errors indicate either an internal hardware failure or a defect in the firmware controlling interactions with the GPIB and RS-232 logic circuits. The isolation between chassis ground circuits and floating circuits is controlled by an optical isolation barrier and a serial link.

514

Not able to execute command while GPIB selected
The `SYST:LOCAL` and `SYST:RWLOCK` commands are valid only when the RS-232 interface is selected.

522

I/O processor output buffer overflow
This error indicates either an internal hardware failure or a defect in the firmware controlling interactions with the GPIB and RS-232 logic circuits.

523

I/O processor received unknown code
This error indicates either an internal hardware failure or a defect in the firmware controlling interactions with the GPIB and RS-232 logic circuits.

580

Reference phase-locked loop is unlocked
The `PHAS:UNL:ERR:STAT` has been enabled (“on”) and the internal phase-locked loop that controls the frequency is currently unlocked. This error is most likely to occur when the external reference is out of lock range.

590

I/O processor had unexpected reset
This error indicates either an internal hardware failure or a defect in the firmware controlling interactions with the GPIB and RS-232 logic circuits.

Self-Test Errors

The following errors indicate failures that may occur during a self-test. Refer to the Agilent 33250A *Service Guide* for more information.

601 **Self-test failed; system logic**
This error indicates that the main CPU (U202) cannot communicate with the main logic FPGA (U302).

602 **Self-test failed; dsp**
This error indicates that the main CPU (U202) cannot communicate with the DSP (U506).

603 **Self-test failed; waveform logic**
This error indicates that the main CPU (U202) cannot communicate with the waveform logic FPGA (U1201).

604 **Self-test failed; even waveform memory bank**
This error indicates that either the "even" waveform memory (U1304) or the waveform logic FPGA circuitry (U1301, U1302, U1306) has failed.

605 **Self-test failed; odd waveform memory bank**
This error indicates that either the "odd" waveform memory (U1305) or the waveform logic FPGA circuitry (U1302, U1303, U1307) has failed.

606 **Self-test failed; cross-isolation interface**
This error indicates that the I/O processor (U105) has either timed out or failed its self-test.

607 to 614
607: Self-test failed; ground
608: Self-test failed; +16V supply
609: Self-test failed; +12V supply
610: Self-test failed; +5V supply
611: Self-test failed; +3.3V supply
612: Self-test failed; -2.1V supply
613: Self-test failed; -5.2V supply
614: Self-test failed; -16V supply
These errors indicate that the internal ADC has detected a power supply with a voltage measurement outside of the expected range.

615 **Self-test failed; primary phase locked loop**
This error indicates that the primary PLL (U901, U903) has failed to lock.

616 **Self-test failed; secondary phase locked loop at 200 MHz**
This error indicates that the secondary PLL (U904-U907), which is used for the pulse function, has failed to lock at 200 MHz.

617 **Self-test failed; secondary phase locked loop at 100 MHz**
This error indicates that the secondary PLL (U904-U907), which is used for the pulse function, has failed to lock at 100 MHz.

618 to 625

618: Self-test failed; display contrast DAC
619: Self-test failed; leading edge DAC
620: Self-test failed; trailing edge DAC
621: Self-test failed; square-wave threshold DAC
622: Self-test failed; time base calibration DAC
623: Self-test failed; dc offset DAC
624: Self-test failed; null DAC
625: Self-test failed; amplitude DAC
These errors indicate a malfunctioning system DAC (U701-U705), DAC MUX logic circuit, or DAC MUX (U706-U708, U603) channels. These self-tests use the internal ADC to verify that the system DAC's are operating properly. For each DAC, the readings are taken at 25%, 50%, and 75% of full scale.

626 to 630

626: Self-test failed; analog-digital path select relay
627: Self-test failed; -10 dB attenuator path
628: Self-test failed; -20 dB attenuator path
629: Self-test failed; +20 dB amplifier path
These errors indicate that the specified relay is not being properly switched or the attenuator/amplifier is not providing the expected attenuation or gain. These self-tests use the internal ADC to verify that the output path relays, output amplifier hybrid (+20 dB), and output attenuators are operating properly.

Calibration Errors

The following errors indicate failures that may occur during a calibration. For more information on the calibration procedures, see chapter 4 in the Agilent 33250A *Service Guide*.

701 **Calibration error; security defeated by hardware jumper**
The function generator's calibration security feature has been disabled due to the installation of a hardware jumper on the internal circuit board.

702 **Calibration error; calibration memory is secured**
A calibration cannot be performed when calibration memory is secured. To unsecure the instrument, use the `CAL:SEC:STAT ON` command with the correct security code.

703 **Calibration error; secure code provided was invalid**
The security code specified with the `CAL:SEC:STAT ON` command is invalid.

705 **Calibration error; calibration aborted**
The function generator will not accept configuration commands (such as `APPL:SIN`) while performing a calibration.

706 **Calibration error; provided value is out of range**
The calibration value specified with the `CAL:VAL` command is out of range.

707 **Calibration error; signal input is out of range**
The internal analog-to-digital converter (ADC) has determined that the signal applied to the rear-panel *Modulation In* connector is out of range.

850 **Calibration error; setup is invalid**
You have specified an invalid calibration setup number with the `CAL:SET` command. Refer to the Agilent 33250A *Service Guide* for more information on the calibration procedures.

851 **Calibration error; setup is out of order**
Certain calibration setups must be performed in a specific sequence in order to be valid. Refer to the Agilent 33250A *Service Guide* for more information on the calibration procedures.

Arbitrary Waveform Errors

The following errors indicate failures that may occur during arbitrary waveform operation. *Refer to "Arbitrary Waveform Commands" on page 198 for more information*.

770 **Nonvolatile arb waveform memory corruption detected**
The non-volatile memory used to store arbitrary waveforms has detected a checksum error. The arbitrary waveform cannot be retrieved from memory.

781 **Not enough memory to store new arb waveform; use DATA:DELETE**
The four non-volatile memory locations already contain arbitrary waveforms. To store another waveform, you must first delete one of the stored waveforms using the `DATA:DELete` command.

781 **Not enough memory to store new arb waveform; bad sectors**
Due to a hardware error, there are no more memory locations available to store arbitrary waveforms. This error is likely the result of a failed Flash memory device.

782 **Cannot overwrite a built-in waveform**
The following built-in waveform names are reserved and cannot be used with the `DATA:COPY` command: "EXP_RISE", "EXP_FALL", "NEG_RAMP", "SINC", and "CARDIAC".

784 **Name of source arb waveform for copy must be VOLATILE**
When using the `DATA:COPY` command, you cannot copy ***from*** any source other than "VOLATILE".

785 **Specified arb waveform does not exist**
The `DATA:COPY` command copies the waveform from volatile memory to the specified name in non-volatile memory. Before executing the `DATA:COPY` command, you must download the waveform using the `DATA VOLATILE` or `DATA:DAC VOLATILE` command.

786 **Not able to delete a built-in arb waveform**
You cannot delete any of the five built-in waveforms: “EXP_RISE”, “EXP_FALL”, “NEG_RAMP”, “SINC”, and “CARDIAC”.

787 **Not able to delete the currently selected active arb waveform**
You cannot delete the arbitrary waveform that is currently being output (`FUNC:USER` command).

788 **Cannot copy to VOLATILE arb waveform**
The `DATA:COPY` command copies the waveform from volatile memory to the specified name in non-volatile memory. The source for the copy is always “VOLATILE”. You cannot copy ***from*** any other source and you cannot copy ***to*** “VOLATILE”.

800 **Block length must be even**
The function generator represents binary data as 16-bit integers, which are sent as two bytes (`DATA:DAC VOLATILE` command).

810 **State has not been stored**
The storage location that you specified in the `*RCL` command was not used in a previous `*SAV` command. You cannot recall the instrument state from a storage location that is empty.

Application Programs

Application Programs

This chapter contains several remote interface example programs to help you develop programs for your own application. Chapter 4, "Remote Interface Reference" starting on page 129, lists the syntax for the SCPI (*Standard Commands for Programmable Instruments*) commands available to program the function generator.

Introduction

Three example programs are included in this chapter and each example illustrates the same functionality in a different programming language. The example programs are written in BASIC for Windows®, Microsoft® Visual Basic for Windows®, and Microsoft® Visual C++ for Windows®. Each example program has embedded comments to help you understand the operation of the program. Each example covers the following topics related to programming the function generator.

- Using the short and long form of SCPI commands.
- Configuring an AM waveform.
- Configuring an FM waveform.
- Configuring a linear frequency sweep.
- Configuring a pulse waveform with varying edge times.
- Configuring a triggered burst waveform.
- Downloading ASCII and binary data for an arbitrary waveform.
- Using the 33250A status registers.



The example programs are also included on the CD-ROM shipped with the Agilent 33250A (see the "examples" directory). The examples are placed in their own sub-directory based on programming language. The "Basic" directory contains one ASCII file and you can use the BASIC command *GET "filename"* to retrieve the example program. The other two example sub-directories contain all of the project files necessary for use with Microsoft Visual Basic or Visual C++.

The installation procedure on the CD-ROM gives you the option to install the ActiveX™ components for instrument control. These components are required for Visual Basic and Visual C++. All of the required hardware-level drivers, such as the SICL (*Standard Instrument Control Language*) libraries or the NI-488.2 libraries, should have been loaded when you installed your GPIB interface card. If you are using the RS-232 interface, the SICL or NI-488.2 libraries also give you access to the RS-232 ports on your PC and make those drivers available for all three languages.

The instrument is shipped with both a GPIB interface and an RS-232 interface. Only one interface can be enabled at a time. The GPIB interface is selected when the instrument is shipped from the factory and the address is set to "10". The default settings for RS-232 are 57.6K bps, 8 bits with no parity, and DTR/DSR handshake. The programming examples in this chapter assume these default settings.

To change the GPIB address or to select the RS-232 interface instead, press the Utility key and make your selection from the "I/O" menu.



Example: BASIC for Windows

```
10    ! """"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""!
20    ! Copyright (c) 2000 Agilent Technologies.  All Rights Reserved.                                    !
30    !                                                                                                   !
40    ! Agilent Technologies provides programming samples for illustration                                !
50    ! purposes only.  This sample program assumes that you are familiar                                !
60    ! with the programming language being demonstrated and the tools used                              !
70    ! to create and debug procedures.  Agilent support engineers can help                              !
80    ! answer questions relating to the functionality of the software                                    !
90    ! components provided by Agilent, but they will not modify these samples                           !
100   ! to provide added functionality or construct procedures to meet your                              !
110   ! specific needs.                                                                                  !
120   ! You have a royalty-free right to use, modify, reproduce, and distribute                          !
130   ! this sample program (and/or any modified version) in any way you find                            !
140   ! useful, provided that you agree that Agilent has no warranty,                                    !
150   ! obligations, or liability for any sample programs.                                               !
160   ! """"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""!
170   ! Agilent 33250A 80MHz Function/Arb Waveform Generator Examples
180   !
190   ! Examples include Modulation, Pulse, Sweeping, and Burst.
200   ! Examples illustrate various uses of short/long form SCPI.
210   ! Examples also illustrate enabling/disabling output BNCs.
220   ! To view results on Scope, set to:
230   !    Channel 1: Output BNC, 50ohms, 50us/div, 200mV/div
240   !    Channel 2: Sync BNC, 50us/div, 500mV/div, trigger on Channel 2
250   !
260   ! BASIC for Windows examples for GPIB/RS-232
270   ! 3-30-00
280   !
290   ! GPIB Configuration
300   !
310   ASSIGN @Fgen TO 710                        ! GPIB ASCII data/commands
320   ASSIGN @Bin TO 710;FORMAT OFF              ! GPIB Binary data
330   !
340   ! RS-232 Configuration: uncomment these lines - comment out GPIB lines
350   !
360 ! ASSIGN @Fgen TO 9                           ! RS-232 ASCII data/commands
370 ! ASSIGN @Bin TO 9;FORMAT OFF                 ! RS-232 Binary data
380 ! CONTROL 9,3;57600                          ! 57.6k Baud
390 ! CONTROL 9,4;3                              ! 8 bits no parity; 2 stop bits
400 ! CONTROL 9,5;1                              ! DTR on OUTPUT
410 ! CONTROL 9,12;16                            ! DSR on ENTER
420 ! CONTROL 9,100;0                           ! Turn OFF XON/XOFF
430   COM /Instrument/@Fgen,@Bin                 ! Global instrument addresses
440   !
450   ! Return 33250A to turn-ON conditions
460   !
470   OUTPUT @Fgen;"*RST"                        ! Default state of instrument
480   OUTPUT @Fgen;"*CLS"                        ! Clear errors and status
490   !
```

Continued...

```
500   PRINT "AM Modulation - press CONTinue"
510   !
520   OUTPUT @Fgen;"OUTPut:LOAD INFinity"       ! Configure for Hi Z load
530   OUTPUT @Fgen;"APPLy:SINusoid 1e6,1,0"     ! 1MHz Sine, 1Vpp, 0Vdc Offset
540   OUTPUT @Fgen;"AM:INTernal:FUNCtion RAMP"  ! Modulating signal: Ramp
550   OUTPUT @Fgen;"AM:INTernal:FREQuency 10e3" ! Modulating frequency: 10kHz
560   OUTPUT @Fgen;"AM:DEPTh 80"                ! Modulating depth: 80%
570   OUTPUT @Fgen;"AM:STATe ON"                ! Turn ON AM modulation
580   Check_errors                              ! Routine checks for errors
590   PAUSE
600   OUTPUT @Fgen;"am:stat off"                ! Turn OFF AM modulation
610   !
620   PRINT "FM Modulation - press CONTinue"
630   !
640   OUTPUT @Fgen;"outp:load 50"               ! Configure for 50 ohm load
650   OUTPUT @Fgen;"appl:sin 20e3,1,0"          ! 20kHz Sine, 1Vpp, 0Vdc Offset
660   OUTPUT @Fgen;"fm:dev 20e3"                ! FM deviation: 20kHz
670   OUTPUT @Fgen;"fm:int:freq 1000"           ! FM Modulating Freq: 1kHz
680   OUTPUT @Fgen;"fm:stat on"                 ! Turn ON FM modulation
690   Check_errors                              ! Routine checks for errors
700   PAUSE
710   OUTPUT @Fgen;"fm:stat off"                ! Turn OFF FM modulation
720   !
730   PRINT "Linear Sweep - press CONTinue"
740   !
750   OUTPUT @Fgen;"sweep:time 1"               ! 1 second sweep time
760   OUTPUT @Fgen;"freq:start 100"             ! Start frequency: 100Hz
770   OUTPUT @Fgen;"freq:stop 20000"            ! Stop frequency: 20kHz
780   OUTPUT @Fgen;"sweep:stat on"              ! Turn ON sweeping
790   Check_errors                              ! Routine checks for errors
800   PAUSE
810   OUTPUT @Fgen;"sweep:stat off"             ! Turn OFF sweeping
820   !
830   PRINT "Pulse Waveform with variable Edge Times - press CONTinue"
840   !
850   OUTPUT @Fgen;"output:state off"           ! Disable Output BNC
860   OUTPUT @Fgen;"volt:low 0;:volt:high 0.75" ! Low = 0V, High = 0.75V
870   OUTPUT @Fgen;"pulse:period 1e-3"          ! 1ms intervals
880   OUTPUT @Fgen;"pulse:width  100e-6"        ! 100us pulse width
890   OUTPUT @Fgen;"pulse:tran 10e-6"           ! Edge time 10us
900   OUTPUT @Fgen;"func pulse"                 ! Select Function Pulse
910   OUTPUT @Fgen;"output:state on"            ! Enable Output BNC
920   FOR I=1 TO 10                             ! Vary edge by 1usec steps
930     OUTPUT @Fgen;"puls:tran ";1.0E-5+I*1.E-6
940     WAIT .3
950   NEXT I
960   Check_errors                              ! Routine checks for errors
970   PAUSE
980   !
```

Continued...

```
990   PRINT "Triggered Burst - press CONTinue"
1000  !
1010  OUTPUT @Fgen;"output:state off"           ! Turn OFF Output BNC
1020  OUTPUT @Fgen;"output:sync off"            ! Disable Sync BNC
1030  OUTPUT @Fgen;"func square"                ! Select square wave
1040  OUTPUT @Fgen;"frequency 20e3"             ! 20kHz
1050  OUTPUT @Fgen;"volt 1;:volt:offset 0"      ! 1Vpp and 0V offset
1060  OUTPUT @Fgen;"func:square:dcycle 20"      ! 20% duty cycle
1070  OUTPUT @Fgen;"trig:sour bus"              ! Bus triggering
1080  OUTPUT @Fgen;"burst:ncycles 3"            ! Burst of 3 cycles per trigger
1090  OUTPUT @Fgen;"burst:state on"             ! Enable Burst
1100  OUTPUT @Fgen;"output:state on"            ! Turn ON Output BNC
1110  OUTPUT @Fgen;"output:sync on"             ! Enable Sync BNC
1120  Check_errors                              ! Routine checks for errors
1130  FOR I=1 TO 20
1140    OUTPUT @Fgen;"*trg"                     ! Send BUS trigger
1150    WAIT .1                                 ! Wait 100msec
1160  NEXT I
1170  PAUSE
1180  !
1190  PRINT "Download 20 point Arbitrary waveform using ASCII - press CONTinue"
1200  !
1210  REAL Arb_20(1:20)                         ! Valid range: -1.0 to +1.0
1220  DATA -1,1,-1,-1,1,1,-1,-1,-1,1,1,1,-1,-1,-1,-1,1,1,1,1,1
1230  READ Arb_20(*)                            ! Read constants into array
1240  OUTPUT @Fgen;"data volatile,";Arb_20(*)   ! Download 20 point waveform
1250  OUTPUT @Fgen;"func:user volatile"         ! Select downloaded waveform
1260  OUTPUT @Fgen;"apply:user 10e3,1,0"        ! Output waveform: 10kHz, 1Vpp
1270  Check_errors                              ! Routine checks for errors
1280  PAUSE
1290  !
1300  PRINT "Download 6 point Arbitrary waveform using Binary - press CONTinue"
1310  !
1320  INTEGER Arb_6(1:6)                        ! Valid range: -2047 to +2047
1330  DATA 2047,-2047,2047,2047,-2047,-2047
1340  READ Arb_6(*)                             ! Read constants into array
1350  OUTPUT @Fgen;"data:dac volatile,#212";    ! Send command; suppress CR/LF
1360  ! Note that the WAIT commands are not needed for GPIB - only for RS-232
1370  WAIT .1                                   ! Time to switch to binary mode
1380  OUTPUT @Bin;Arb_6(*);                     ! 12 bytes - no terminator
1390  WAIT .1                                   ! Time to switch to ASCII mode
1400  OUTPUT @Fgen;";:apply:user 5000,1,0"      ! Terminator + APPLy
1410  Check_errors                              ! Routine checks for errors
1420  PAUSE
1430  !
```

Continued...

```
1440  PRINT "Using the Status Registers"
1450  !
1460  OUTPUT @Fgen;"appl:sin 10e3,1,0"          ! 10kHz Sine wave; 1Vpp
1470  OUTPUT @Fgen;"trig:sour bus"               ! Bus Trigger in Burst
1480  OUTPUT @Fgen;"burst:ncycles 50000"         ! 50000 cycles x 0.1ms = 5s
1490  OUTPUT @Fgen;"burst:stat on"               ! Turn ON burst mode
1500  OUTPUT @Fgen;"*ese 1"                      ! Operation complete enabled
1510  OUTPUT @Fgen;"*sre 32"                     ! Operation complete sets SRQ
1520  Check_errors                               ! Routine checks for errors
1530  OUTPUT @Fgen;"*trg;*opc"                   ! Trigger burst
1540                                             ! *OPC signals end of *TRG
1550  !
1560  ! Now wait for Operation Complete to signal Burst complete
1570  !
1580  WHILE 1
1590    OUTPUT @Fgen;"*stb?"                     ! Request Status Byte
1600    ENTER @Fgen;Stb                          ! Read Status Byte
1610    IF (BIT(Stb,6)) THEN                     ! Test for Master Summary Bit
1620      PRINT "Done"
1630      STOP
1640    END IF
1650  END WHILE
1660  END
1670  !
1680  ! Subprogram to check for instrument errors.
1690  !
1700  SUB Check_errors
1710    DIM Description$[100]
1720    INTEGER Err_num
1730    COM /Instrument/@Fgen,@Bin
1740    !
1750    ! Query the error queue until a "0, No Error" is found
1760    !
1770    WHILE 1
1780      OUTPUT @Fgen;"SYSTem:ERRor?"           ! Request Error message
1790      ENTER @Fgen;Err_num,Description$       ! Error number,Description
1800      IF NOT Err_num THEN SUBEXIT            ! If error = 0 then exit
1810      PRINT Err_num,Description$             ! Print Error,Description
1820    END WHILE
1830  SUBEND
```

Example: Microsoft Visual Basic for Windows

```
Option Explicit
Dim m_Count As Integer                      ' Used to sequence messages
Private IOUtils As New AgtUtilsObject

' """"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""'
' Copyright (c) 2000 Agilent Technologies.  All Rights Reserved.           '
'                                                                         '
' Agilent Technologies provides programming samples for illustration      '
' purposes only.  This sample program assumes that you are familiar       '
' with the programming language being demonstrated and the tools used     '
' to create and debug procedures.  Agilent support engineers can help     '
' answer questions relating to the functionality of the software          '
' components provided by Agilent, but they will not modify these samples  '
' to provide added functionality or construct procedures to meet your     '
'specificneeds.                                                          '
' You have a royalty-free right to use, modify, reproduce, and distribute '
' this sample program (and/or any modified version) in any way you find   '
' useful, provided that you agree that Agilent has no warranty,           '
' obligations, or liability for any sample programs.                      '
' """"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""'
'
'  Agilent 33250A 80 MHz Function/Arbitrary Waveform Generator Examples
'
'  Examples include Modulation, Pulse, Sweeping, Burst, and Status checking.
'  Examples illustrate various uses of short/long form SCPI.
'  Examples also illustrate enabling/disabling output BNCs.
'  To view results on Scope, set to:
'     Channel 1: Output BNC, 50ohms, 50us/div, 200mV/div
'     Channel 2: Sync BNC, 50us/div, 500mV/div, trigger on Channel 2
'
'  Microsoft Visual Basic 6.0 Programming Examples
'  3-30-00
'
Private Sub cmdStart_Click()

    Dim i As Integer                        ' Used as general purpose counter

    cmdStart.Enabled = False                ' Disable Start button
'
' Return the 33250A to turn-ON conditions
'
    Arb.IO.Output "*RST"                    ' Default state of instrument
    Arb.IO.Output "*CLS"                    ' Clear errors and status
```

Continued...

```
'
'  AM Modulation
'
    Arb.Output "OUTPut:LOAD INFinity"       ' Configure for Hi Z load
    Arb.Output "APPLy:SINusoid 1e6,1,0"     ' 1MHz Sine, 1Vpp, 0Vdc offset
    Arb.Output "AM:INTernal:FUNCtion RAMP"  ' Modulating signal: Ramp
    Arb.Output "AM:INTernal:FREQuency 10e3" ' Modulating frequency: 10kHz
    Arb.Output "AM:DEPTh 80"                ' Modulating depth: 80%
    Arb.Output "AM:STATe ON"                ' Turn ON AM modulation
    Check_Errors                            ' Routine checks for errors
    MsgBox "AM Modulation", vbOKOnly, "33250A Example"
    Arb.Output "AM:STATe OFF"               ' Turn OFF AM modulation

'
'   FM Modulation
'
    Arb.Output "outp:load 50"               ' Configure for 50 ohm load
    Arb.Output "appl:sin 20e3,1,0"          ' 20kHz Sine, 1Vpp, 0Vdc Offset
    Arb.Output "fm:dev 20e3"                ' FM deviation: 20kHz
    Arb.Output "fm:int:freq 1000"           ' FM Modulating Freq: 1kHz
    Arb.Output "fm:stat on"                 ' Turn ON FM modulation
    Check_Errors                            ' Routine checks for errors
    MsgBox "FM Modulation", vbOKOnly, "33250A Example"
    Arb.Output "fm:stat off"                ' Turn OFF FM modulation
'
'   Linear Sweep
'
    Arb.Output "sweep:time 1"               ' 1 second sweep time
    Arb.Output "freq:start 100"             ' Start frequency: 100Hz
    Arb.Output "freq:stop 20000"            ' Stop frequency: 20kHz
    Arb.Output "sweep:stat on"              ' Turn ON sweeping
    Check_Errors                            ' Routine checks for errors
    MsgBox "Linear Sweep", vbOKOnly, "33250A Example"
    Arb.Output "sweep:stat off"             ' Turn OFF sweeping
'
'   Pulse Waveform with variable Edge Times
'
    Arb.Output "output:state off"           ' Disable Output BNC
    Arb.Output "volt:low 0;:volt:high 0.75" ' Low = 0V, High = 0.75V
    Arb.Output "pulse:period 1e-3"          ' 1ms intervals
    Arb.Output "pulse:width 100e-6"         ' 100us pulse width
    Arb.Output "pulse:tran 10e-6"           ' Edge time 10us
    Arb.Output "func pulse"                 ' Select Function Pulse
    Arb.Output "output:state on"            ' Enable Output BNC
    For i = 1 To 20                         ' Vary edge by 1usec steps
        Arb.Output "puls:tran " & (0.00001 + i * 0.000001)
        Sleep 300                           ' Wait 300msec
    Next i
    Check_Errors                            ' Routine checks for errors
    MsgBox "Pulse Waveform with variable Edge Times", vbOKOnly, "33250A Example"
```

Continued...

```
'
'   Triggered Burst
'
    Arb.Output "output:state off"           ' Turn OFF Output BNC
    Arb.Output "output:sync off"            ' Disable Sync BNC
    Arb.Output "func square"                ' Select Function square
    Arb.Output "frequency 20e3"             ' 20kHz
    Arb.Output "volt 1;:volt:offset 0"      ' 1Vpp and 0V offset
    Arb.Output "func:square:dcycle 20"      ' 20% duty cycle
    Arb.Output "trig:sour bus"              ' Bus triggering
    Arb.Output "burst:ncycles 3"            ' Burst of 3 cycles per trigger
    Arb.Output "burst:state on"             ' Enable Burst
    Arb.Output "output:state on"            ' Turn ON Output BNC
    Arb.Output "output:sync on"             ' Enable Sync BNC
    Check_Errors                            ' Routine checks for errors
    For i = 1 To 20
        Arb.Output "*trg"                   ' Send BUS trigger
        Sleep 100                           ' Wait 100msec
    Next i
    MsgBox "Triggered Burst", vbOKOnly, "33250A Example"
'
'   Download a 20 point Arbitrary waveform using ASCII.
'
    Dim Arb_20(0 To 19) As Double           ' Allocate array of 20 reals
    Fill_array Arb_20                       ' Call routine to fill array
    With Arb.IO.Write
        .Command "data volatile,", False    ' Place command into buffer
        .Argument(0) = Arb_20               ' Place comma separated data into buffer
        .Send                               ' Send command + data
    End With
    Arb.Output "func:user volatile"         ' Select downloaded waveform
    Arb.Output "apply:user 10e3,1,0"        ' Output waveform: 10kHz, 1Vpp
    Check_Errors                            ' Routine checks for errors
    MsgBox "Download a 20 point Arb waveform using ASCII.", vbOKOnly, "33250A Example"
'
'   Download a 6 point Arbitrary waveform using Binary.
'   This example for GPIB only
'
    Dim Arb_6()                             ' Create array
    Dim Length As Long                      ' Used to find total length of array
    Dim Command() As Byte                   ' Used to store total command sequence

    Arb_6 = Array(2047, -2047, 2047, 2047, -2047, -2047)
    Length = IOUtils.CreateIEEEBlock(Arb_6, IIOUtils_Short, IIOUtils_BigEndian,
"data:dac volatile,", Command)
    Arb.IO.WriteBytes Length, Command       ' Download command and bytes
    Arb.Output "apply:user 5000,1,0"        ' Output waveform: 5kHz, 1Vpp
    Check_Errors
    MsgBox "Download a 6 point Arb waveform using Binary.", vbOKOnly, "33250A Example"
```

Continued...

```
'
'   Using the Status Registers
'
    Arb.Output "apply:sin 10e3,1,0"          ' 10kHz Sine wave; 1Vpp
    Arb.Output "trig:sour bus"               ' Bus Trigger in Burst
    Arb.Output "burst:ncycles 50000"         ' 50000 cycles x 0.1ms = 5s
    Arb.Output "burst:stat on"               ' Turn ON burst mode
    Arb.Output "*ese 1"                      ' Operation complete enabled
    Arb.Output "*sre 32"                     ' Operation complete sets SRQ
    Check_Errors                             ' Routine checks for errors
    Arb.Output "*trg;*opc"                   ' Trigger burst
                                             ' *OPC signals end of *TRG
    Dim Stats As Integer                     ' Variable to store status
    Dim Done As Boolean                      ' Controls While loop
    Done = False

    While Not Done
        Arb.Output "*stb?"                   ' Request status byte
        Arb.Enter Stats                      ' Read status byte
        If Stats And 64 Then                 ' Test Master Summary bit
          Done = True
        End If
    Wend

    MsgBox "Done", vbOKOnly, "33250A "
    cmdStart.Enabled = True
End Sub

Private Sub Form_Load()
    Dim IdStr As String
    m_Count = 1

    Arb.Output "*IDN?"                       ' Query instrument information
    Arb.Enter IdStr                          ' Read result into IdStr
    Caption = IdStr                          ' Make that data the message on box
End Sub

Sub Check_Errors()
    Dim ErrVal(0 To 1)

    With Arb
        .Output "syst:err?"                  ' Query any errors data
        .Enter ErrVal                        ' Read: Errnum,"Error String"
        While ErrVal(0) <> 0                 ' End if find: 0,"No Error"
            lstErrors.AddItem ErrVal(0) & "," & ErrVal(1) ' Display errors
            lstErrors.Refresh                ' Update the box
            .Output "SYST:ERR?"              ' Request error message
            .Enter ErrVal                    ' Read error message
        Wend
    End With
End Sub
```

Continued...

```
Sub WaitForOPC()
    Dim Stats As Byte

    With Arb
        Stats = .IO.Query("*STB?")          ' Read Status Byte

        Do While (Stats And 64) = 0         ' Test for Master Summary Bit
            Sleep 100                       ' Pause for 100msec
            Stats = .IO.Query("*STB?")      ' Read Status Byte
        Loop
    End With
End Sub

Sub Fill_array(ByRef data_array() As Double)

'
' Routine can be used to fill array passed from Main Program.  Fills entire array
' with sequence of +/- 1.0
'
    data_array(0) = -1#
    data_array(1) = 1#
    data_array(2) = -1#
    data_array(3) = -1#
    data_array(4) = 1#
    data_array(5) = 1#
    data_array(6) = -1#
    data_array(7) = -1#
    data_array(8) = -1#
    data_array(9) = 1#
    data_array(10) = 1#
    data_array(11) = 1#
    data_array(12) = -1#
    data_array(13) = -1#
    data_array(14) = -1#
    data_array(15) = -1#
    data_array(16) = 1#
    data_array(17) = 1#
    data_array(18) = 1#
    data_array(19) = 1#

End Sub
```

Example: Microsoft Visual C++ for Windows

```
//'""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""'
//' Copyright (c) 2000 Agilent Technologies.  All Rights Reserved.           '
//'                                                                          '
//' Agilent Technologies provides programming samples for illustration       '
//' purposes only.  This sample program assumes that you are familiar        '
//' with the programming language being demonstrated and the tools used      '
//' to create and debug procedures.  Agilent support engineers can help      '
//' answer questions relating to the functionality of the software           '
//' components provided by Agilent, but they will not modify these samples   '
//' to provide added functionality or construct procedures to meet your      '
//' specific needs.                                                          '
//' You have a royalty-free right to use, modify, reproduce, and distribute  '
//' this sample program (and/or any modified version) in any way you find    '
//' useful, provided that you agree that Agilent has no warranty,            '
//' obligations, or liability for any sample programs.                       '
//'""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""'
//
// Agilent 33250A 80 MHz Function/Arb Waveform Generator Examples
//
// Examples include Modulation, Pulse, Sweeping, Burst, and Status Checking.
// Examples illustrate various uses of short/long form SCPI.
// Examples also illustrate enabling/disabling output BNCs.
// To view results on Scope, set to:
//    Channel 1: Output BNC, 50ohms, 50us/div, 200mV/div
//    Channel 2: Sync BNC, 50us/div, 500mV/div, trigger on Channel 2
//
// Microsoft Visual C++ 6.0 for GPIB/RS-232
// 3-30-00
//
#include <stdio.h>
#include <comdef.h>

//
// Import the IOUtils
// (your directory is dependent on where BenchLlink XL was installed)
//
#pragma warning(disable:4192)              // Suppresses warning from import
#import "C:\siclnt\servers\AgtIOUtils.dll"
using namespace AgilentIOUtilsLib;
```

Continued...

6

```
void Check_Errors(IIO *pIOObj)
{
   _variant_t ErrNum, ErrStr;

   while (1)
   {
      ErrNum = "";                          // Initialize variants
      ErrStr = "";

      pIOObj->Output(":SYST:ERR?");
      pIOObj->Enter(&ErrNum, "#,K");        // Read number; don't flush input buffer
      pIOObj->Enter(&ErrStr, "K");          // Read the string
      ErrNum.ChangeType(VT_I4);

      if ((long) ErrNum == 0)               // Checking for: 0,"No Error"
      {
         break;
      }
      else
      {
         printf ("\nERROR %d:  %S\n", (long) ErrNum, ErrStr.bstrVal);
      }
   }
}

void Pause()
{
   // Routine to permit stopping of execution of program

   printf ("Press Enter to continue...");
   fflush(stdout);
   fgetc(stdin);                            // Wait for LF
   printf("\n");
}

void Fill_array(double data[20])
{
   // Routine can be used to fill array passed from Main Program.  Fills entire
   // array with sequence of +/- 1.0

   data[0] = -1.0;
   data[1] = 1.0;
   data[2] = -1.0;
   data[3] = -1.0;
   data[4] = 1.0;
   data[5] = 1.0;
   data[6] = -1.0;
   data[7] = -1.0;
   data[8] = -1.0;
   data[9] = 1.0;
   data[10] = 1.0;
   data[11] = 1.0;
   data[12] = -1.0;
   data[13] = -1.0;
   data[14] = -1.0;
   data[15] = -1.0;
   data[16] = 1.0;
   data[17] = 1.0;
   data[18] = 1.0;
   data[19] = 1.0;
```

Continued...

```
}
int main(int argc, char* argv[])
{
   CoInitialize(NULL);
   int i;                                    // General purpose counter
   char cmds[ 100 ];                         // Used to store command string

   try
   {
      IIOManagerPtr IOMgr;
      IIOPtr IOObj;
      IIOUtilsPtr IOUtils;

      IOUtils.CreateInstance(__uuidof(AgtUtilsObject));
      IOMgr.CreateInstance(__uuidof(AgtIOManager));

      //
      // RS-232 Configuration: uncomment line - comment out GPIB line
      // IOObj = IOMgr->ConnectToInstrument(L"COM1::Baud=57600,Handshake=DTR_DSR");

      //
      // GPIB Configuration
      IOObj = IOMgr->ConnectToInstrument(L"GPIB0::10");

      //
      // Return 33250A to turn-on conditions
      //
      IOObj->Output("*RST");                          // Default state of instrument
      IOObj->Output("*CLS");                          // Clear errors and status
//
// AM Modulation
//
      printf ("AM Modulation\n");
      IOObj->Output("OUTPut:LOAD INFinity");          // Configure for Hi Z load
      IOObj->Output("APPLy:SINusoid 1e6,1,0");        // 1MHz Sine, 1Vpp, 0Vdc offset
      IOObj->Output("AM:INTernal:FUNCtion RAMP");     // Modulating signal: Ramp
      IOObj->Output("AM:INTernal:FREQuency 10e3");    // Modulating frequency: 10kHz
      IOObj->Output("AM:DEPTh 80");                   // Modulation depth: 80%
      IOObj->Output("AM:STATe ON");                   // Turn ON AM modulation
      Check_Errors(IOObj);                            // Routine check for errors
      Pause();
      IOObj->Output("am:stat off");                   // Turn OFF AM modulation
//
// FM Modulation
//
      printf ("FM Modulation\n");
      IOObj->Output("outp:load 50");                  // Configure for 50 ohm load
      IOObj->Output("appl:sin 20e3,1,0");             // 20kHz Sine, 1Vpp, 0Vdc Offset
      IOObj->Output("fm:dev 20e3");                   // FM deviation: 20kHz
      IOObj->Output("fm:int:freq 1000");              // FM Modulation Freq: 1kHz
      IOObj->Output("fm:stat on");                    // Turn ON FM modulation
      Check_Errors(IOObj);                            // Routine check for errors
      Pause();
      IOObj->Output("fm:stat off");                   // Turn OFF FM modulation
```

Continued...

```
//
// Linear Sweep
//
      printf ("Linear Sweep\n");
      IOObj->Output("sweep:time 1");                // 1 second sweep time
      IOObj->Output("freq:start 100");              // Start frequency 100Hz
      IOObj->Output("freq:stop 20000");             // Stop frequency 20kHz
      IOObj->Output("sweep:stat on");               // Turn ON sweeping
      Check_Errors(IOObj);                          // Routine check for errors
      Pause();
      IOObj->Output("sweep:stat off");              // Turn OFF sweeping
//
// Pulse period with variable Edge Times
//
      printf ("Pulse Waveform with variable Edge Times\n");
      IOObj->Output("output:state off");            // Disable Output BNC
      IOObj->Output("volt:low 0;:volt:high 0.75");  // Low = 0V, High = 0.75V
      IOObj->Output("pulse:period 1e-3");           // 1ms intervals
      IOObj->Output("pulse:width 100e-6");          // 100us pulse width
      IOObj->Output("pulse:tran 10e-6");            // Edge time 10us
      IOObj->Output("func pulse");                  // Select Function Pulse
      IOObj->Output("output:state on");             // Enable Output BNC
      for ( i = 0; i < 10; i++ ) {                  // Vary edge by 1usec steps
         sprintf(cmds, "puls:tran %f\n",0.00001+0.000001*float(i));
         IOObj->Output(cmds);
         SleepEx(300, 0);                           // Wait 300msec
      }
      Check_Errors(IOObj);                          // Routine check for errors
      Pause();
//
// Triggered Burst
//
      printf ("Triggered Burst\n");
      IOObj->Output("output:state off");            // Turn OFF Output BNC
      IOObj->Output("output:sync off");             // Disable Sync BNC
      IOObj->Output("func square");                 // Select square wave
      IOObj->Output("frequency 20e3");              // 20kHz
      IOObj->Output("volt 1;:volt:offset 0");       // 1Vpp and 0V offset
      IOObj->Output("func:square:dcycle 20");       // 20% duty cycle
      IOObj->Output("trig:sour bus");               // Bus triggering
      IOObj->Output("burst:ncycles 3");             // Burst of 3 cycles
      IOObj->Output("burst:state on");              // Enable Burst
      IOObj->Output("output:state on");             // Turn On Output BNC
      IOObj->Output("output:sync on");              // Enable Sync BNC
      Check_Errors(IOObj);                          // Routine check for errors

      for (int i = 1; i <= 20; i++)
      {
         IOObj->Output("*trg");                     // Send BUS trigger
         SleepEx(100, 0);                           // Wait 100msec
      }
      Pause();
```

Continued...

```
//
// Download a 20 point Arbitrary waveform using ASCII.
//
      printf ("Download a 20 point Arbitrary waveform using ASCII\n");
      // Download 20 point waveform
      {
         double Real_array[20];
         Fill_array(Real_array);

         IWritePtr pWrite = IOObj->Write();
         pWrite->Command ("data volatile, ", VARIANT_FALSE); // Command into buffer

         for (int i = 0; i < 20; i++)
            pWrite->PutArgument(i, Real_array[i]);   // Comma separated list to buffer

         pWrite->Send ();                            // Send buffer to the instrument
      }
      IOObj->Output("func:user volatile");           // Select downloaded waveform
      IOObj->Output("apply:user 10e3,1,0");          // Output waveform: 10kHz, 1Vpp
      Check_Errors(IOObj);                           // Routine check for errors
      Pause();
//
// Download a 6 point arbitrary waveform using Binary.
//
      printf ("Download a 6 point Arbitrary waveform using Binary\n");
      long Len;
      _variant_t DataArray = "2047,-2047,2047,2047,-2047,-2047";
      SAFEARRAY *pBlock;

      // Create SCPI command with Binary block appended on end
      Len = IOUtils->CreateIEEEBlock(DataArray, IIOUtils_Short, IIOUtils_BigEndian,
            _variant_t("data:dac volatile, "), &pBlock);
      IOObj->WriteBytes(Len, &pBlock);               // Send command and data
      SleepEx(100, 0);                               // Wait 100msec for interface
                                                     // (for RS-232 only)
      IOObj->Output("apply:user 5000,1,0");          // Output waveform: 5kHz, 1Vpp
      Check_Errors(IOObj);                           // Routine check for errors
      Pause();
```

Continued...

```
//
// Using the Status Registers
//
      printf ("Using the Status Registers\n");
      IOObj->Output("apply:sin 10e3,1,0");          // 10kHz Sine wave; 1Vpp
      IOObj->Output("trig:sour bus");               // Bus Trigger in Burst
      IOObj->Output("burst:ncycles 50000");         // 50000 cycles x 0.1 = 5s
      IOObj->Output("burst:stat on");               // Turn ON burst mode
      IOObj->Output("*ese 1");                      // Operation complete enabled
      IOObj->Output("*sre 32");                     // Operation complete sets SRQ
      Check_Errors(IOObj);                          // Routine check for errors
      IOObj->Output("*trg;*opc");                   // *OPC signals end of *TRG

      _variant_t Stb;
      Stb.vt = VT_I2;                               // Force Enter() to convert to Short

      while (1)
      {
         IOObj->Output("*stb?");                    // Request Status Byte
         IOObj->Enter(&Stb, "K");                   // Read Status Byte
         if ((short) Stb & 0x40)                    // Test for Master Summary Bit
         {
            break;
         }
      }

      printf ("End of Program\n");
   }
   catch (_com_error &e)
   {
      _bstr_t dsp = e.Description();
      _bstr_t emsg = e.ErrorMessage();
      fprintf (stderr, "COM Exception occurred during
processing!\nDescription::%s\nMessage::%s\n",
               (char *) dsp, (char *) emsg);
   }

   CoUninitialize();
   return 0;
}
```

Tutorial

Tutorial

In order to achieve the best performance from the Agilent 33250A, it may be helpful for you to gain a better understanding of the internal operations of the instrument. This chapter describes basic signal-generation concepts and gives specific details on the internal operations of the function generator.

- Direct Digital Synthesis, *on page 295*
- Creating Arbitrary Waveforms, *on page 298*
- Square Waveform Generation, *on page 300*
- Pulse Waveform Generation, *on page 300*
- Signal Imperfections, *on page 302*
- Output Amplitude Control, *on page 304*
- Ground Loops, *on page 305*
- Attributes of AC Signals, *on page 307*
- Modulation, *on page 309*
- Frequency Sweep, *on page 312*
- Burst, *on page 315*

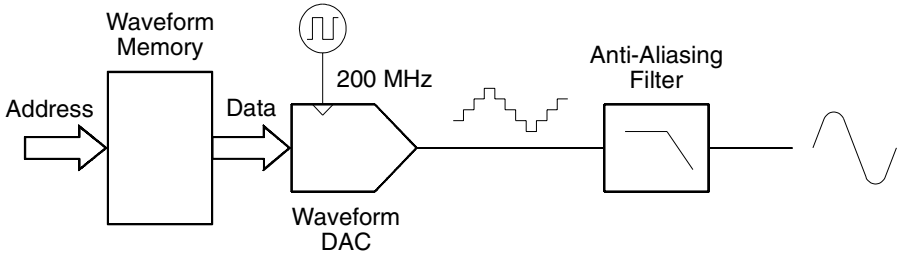
You can use an arbitrary waveform generator in a variety of applications where it might be otherwise difficult or impossible to generate complex output waveforms. With an arbitrary waveform generator, signal imperfections such as rise time, ringing, glitches, noise, and random timing variations can be easily simulated in a controlled manner.

Physics, chemistry, biomedicine, electronics, mechanics, and other fields can benefit from the versatility of an arbitrary waveform generator. Wherever things vibrate, pump, pulse, bubble, burst, or change with time in any way, there are possible applications – limited only by your ability to specify the waveform data.

Direct Digital Synthesis

Digital signal processing methods are used in many everyday applications. Whether it is a digital audio compact disc player, an electronic synthesized piano, or a voice-synthesized telephone message system, complex waveforms can be easily created or reproduced using digital signal-generation methods.

The 33250A uses a signal-generation technique called *Direct Digital Synthesis* (DDS) for all waveform functions except pulse. As shown below, a stream of digital data representing the desired waveform is sequentially read from waveform memory and is applied to the input of a digital-to-analog converter (DAC). The DAC is clocked at the function generator's sampling frequency of 200 MHz and outputs a series of voltage steps approximating the desired waveform. A low-pass "anti-aliasing" filter then smooths the voltage steps to create the final waveform.

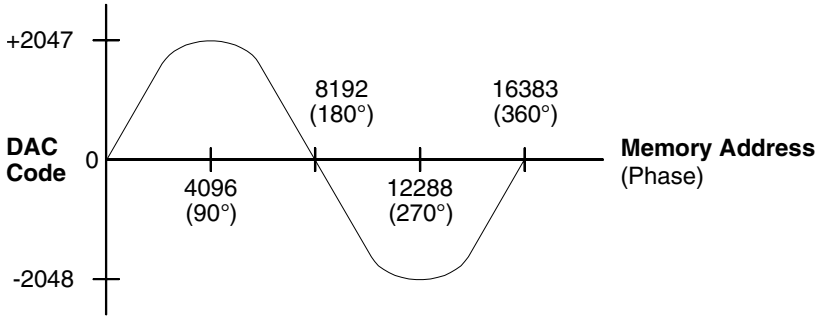


Direct Digital Synthesis Circuitry

The 33250A uses two anti-aliasing filters. A ninth-order elliptical filter is used for continuous sine waves because of its nearly flat passband and sharp cutoff above 80 MHz. Since elliptical filters exhibit severe ringing for waveforms other than continuous sine waves, a seventh-order linear-phase filter is used for all other waveforms functions.

For standard waveforms, and arbitrary waveforms that are defined with fewer than 16,384 (16K) points, the function generator uses waveform memory that is 16K words deep. For arbitrary waveforms that are defined with more than 16K points, the function generator uses waveform memory that is 65,536 (64K) words deep.

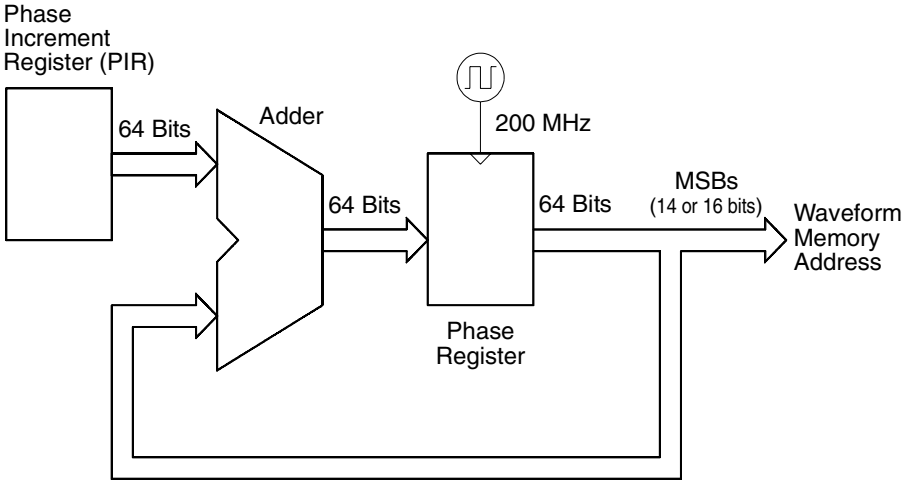
The 33250A represents amplitude values by 4,096 discrete voltage levels (or 12-bit vertical resolution). The specified waveform data is divided into samples such that one waveform cycle exactly fills waveform memory (see the illustration below for a sine wave). If you create an arbitrary waveform that does not contain exactly 16K or 64K points, the waveform is automatically "stretched" by repeating points or by interpolating between existing points as needed to fill waveform memory. Since all of waveform memory is filled with one waveform cycle, each memory location corresponds to a phase angle of $2\pi/16{,}384$ radians or $2\pi/65{,}536$ radians.



Sine Wave Representation in Waveform Memory

Direct digital synthesis (DDS) generators use a *phase accumulation* technique to control waveform memory addressing. Instead of using a counter to generate sequential memory addresses, an "adder" is used (see the following page). On each clock cycle, the constant loaded into the phase increment register (PIR) is added to the present result in the phase accumulator. The most-significant bits of the phase accumulator output are used to address waveform memory. By changing the PIR constant, the number of clock cycles required to step through the entire waveform memory changes, thus changing the output frequency. When a new PIR constant is loaded into the register, the waveform output frequency changes phase continuously following the next clock cycle.

The PIR determines how fast the phase value changes with time and ultimately controls the frequency being synthesized. More bits in the phase accumulator result in finer frequency resolution. Since the PIR affects only the rate of change of the phase value (and not the phase itself), changes in waveform frequency are phase-continuous.



Phase Accumulator Circuitry

The 33250A uses a 64-bit phase accumulator which yields 2^{-64} x 200 MHz or 10.8 picohertz frequency resolution internally. Note that only the 14 or 16 most-significant bits of the Phase Register are used to address waveform memory. Therefore, when synthesizing low frequencies (less than 12.21 kHz), the address will not change during every clock cycle. However, at higher frequencies (greater than 12.2 kHz), the address will change by more than one location during each clock cycle and some points will be skipped. If too many points are skipped, a phenomenon known as “aliasing” will occur and the waveform output will become somewhat distorted.

The *Nyquist Sampling Theorem* states that in order to prevent aliasing, the highest frequency component of the desired output waveform must be **less than half** of the sampling frequency (100 MHz for the 33250A).

Creating Arbitrary Waveforms

For most applications, it is not necessary to create an arbitrary waveform with a specific number of points since the function generator will repeat points (or interpolate) as necessary to fill waveform memory. For example, if you specify 100 points, each waveform point will be repeated an average of 16,384/100 or 163.84 times. For the 33250A, you do not have to change the length of the waveform to change its output frequency. All you have to do is create a waveform of any length and then adjust the function generator's output frequency. However, in order to get the best results (and minimize voltage quantization errors), it is recommended that you use the full range of the waveform DAC (use all 4,096 levels).

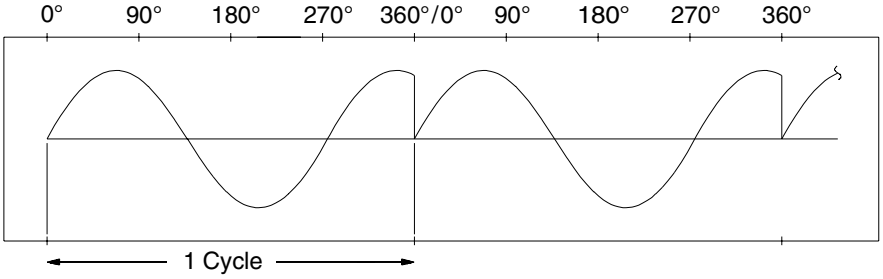
When entering waveform points from the function generator's front panel, you are not required to enter the points at evenly-spaced intervals in time. You can always add additional points as needed where the waveform is more complex. From the front panel only, you can also use linear interpolation to smooth the transition between waveform points. These features make it possible to create useful arbitrary waveforms using a relatively small number of points.

With the 33250A, you can output an arbitrary waveform to an upper frequency limit of 25 MHz. However, note that the *useful* upper limit is usually less due to the function generator's bandwidth limitation and aliasing. Waveform components above the function generator's -3 dB bandwidth will be attenuated.

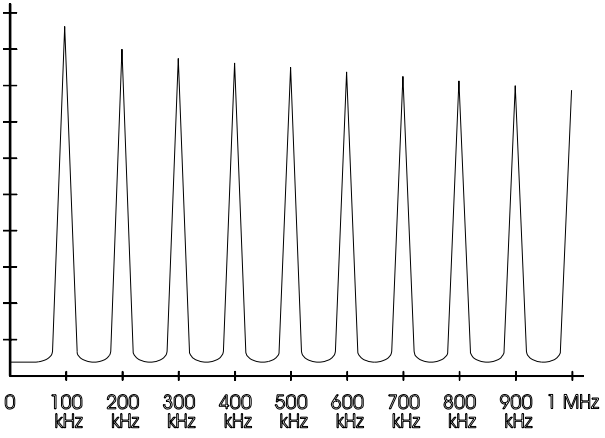
For example, consider an arbitrary waveform consisting of 10 cycles of a sine waveform. When you set the output frequency to 5 MHz, the actual output frequency will be 50 MHz and the amplitude will be attenuated by 3 dB. As you increase the frequency above 5 MHz, more attenuation will occur. At approximately 8 MHz, waveform distortion due to aliasing will become evident. Some aliasing will be present in most arbitrary waveforms, but whether or not it will be troublesome depends on your specific application.

When creating arbitrary waveforms, the function generator will always attempt to replicate the finite-length time record to produce a periodic version of the data in waveform memory. However, as shown on the following page, it is possible that the shape and phase of a signal may be such that a discontinuity is introduced at the end point. When the waveshape is repeated for all time, this end-point discontinuity will introduce *leakage errors* in the frequency domain because many spectral terms are required to describe the discontinuity.

Leakage error is caused when the waveform record does not include an integral number of cycles of the fundamental frequency. Power from the fundamental frequency, and its harmonics, is transferred to the spectral components of the rectangular sampling function. You can reduce leakage errors by adjusting the window length to include an integer number of cycles or by including more cycles within the window to reduce the residual end-point discontinuity size. Some signals are composed of discrete, non-harmonically related frequencies. Since these signals are non-repetitive, all frequency components cannot be harmonically related to the window length. You should be careful in these situations to minimize end-point discontinuities and spectral leakage.



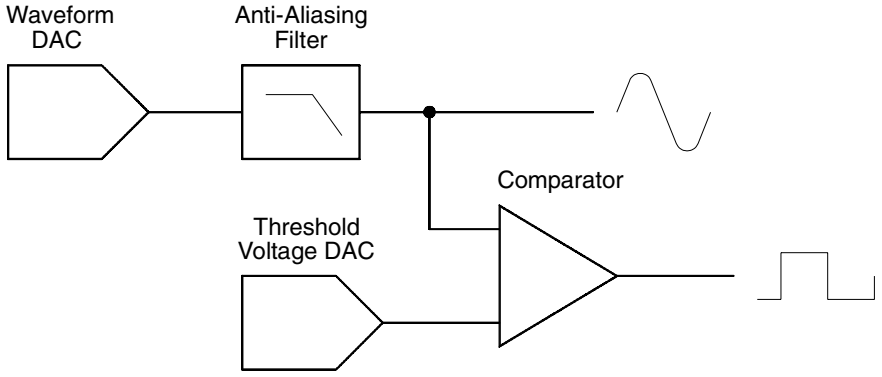
Arbitrary Waveform with Discontinuity



Spectrum of Above Waveform at 100 kHz

Square Waveform Generation

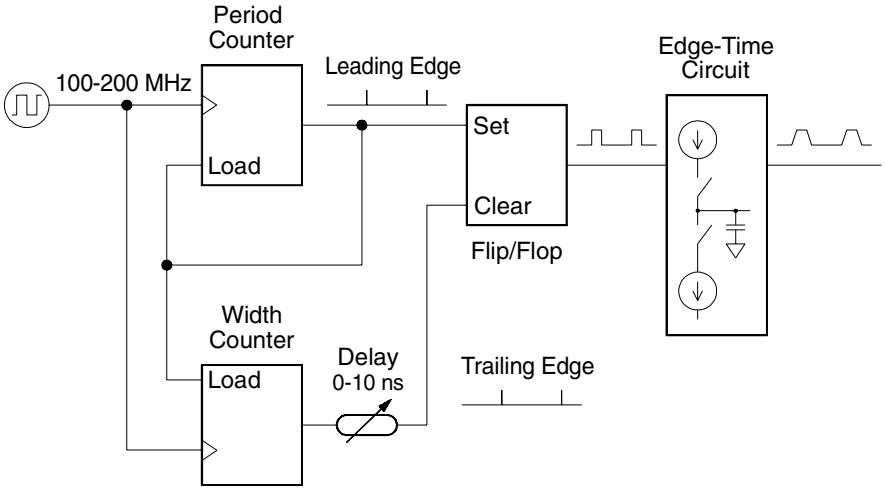
To eliminate distortion due to aliasing at higher frequencies, the 33250A uses a different waveform generation technique to create square waves. For frequencies above 2 MHz, square waveforms are created by routing a DDS-generated sine wave into a comparator. The digital output from the comparator is then used as the basis for the square wave output. The duty cycle of the waveform can be varied by changing the comparator's threshold. For frequencies below 2 MHz, different waveshapes are loaded into waveform memory to minimize jitter.



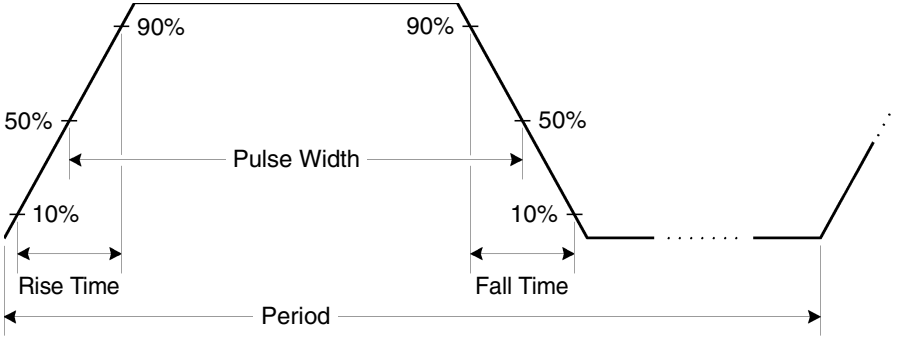
Square Waveform Generation Circuitry

Pulse Waveform Generation

To eliminate distortion due to aliasing at higher frequencies, the 33250A also uses a different waveform generation technique to create pulse waves. For pulse waveform generation, clock cycles are counted to derive both the period and pulse width. To achieve fine period resolution, the clock frequency is varied from 100 MHz to 200 MHz by a phase-locked loop (see the following page). To achieve fine period width resolution, an analog delay (0 to 10 ns) is applied to the trailing edge. The rising and falling edge times are controlled by a circuit that varies the charging currents in a capacitor. Period, pulse width, and edge time are controlled independently, within certain limits.



Pulse Waveform Generation Circuitry



Pulse Waveform Parameters

Signal Imperfections

For sine waveforms, signal imperfections are easiest to describe and observe in the frequency domain using a spectrum analyzer.
Any component of the output signal which has a different frequency than the fundamental (or "carrier") is considered to be spurious.
The signal imperfections can be categorized as *harmonic*, *non-harmonic*, or *phase noise* and are specified in "decibels relative to the carrier level" or "dBc".

Harmonic Imperfections Harmonic components always appear at multiples of the fundamental frequency and are created by non-linearities in the waveform DAC and other elements of the signal path. The 33250A uses a 100 MHz low-pass filter to attenuate very-high-frequency harmonics. At lower frequencies and low amplitudes, another possible source of harmonic distortion is due to the current flowing through the cable connected to the function generator's *Sync* output connector.
This current can cause a small square-wave voltage drop across the resistance of the cable's shield and some of this voltage can be imposed on the main signal. If this is a concern for your application, you should remove the cable or disable the *Sync* output connector. If your application requires that you use the *Sync* output connector, you can minimize the effect by terminating the cable in a high impedance load (rather than into a 50Ω load).

Non-Harmonic Imperfections The biggest source of non-harmonic spurious components (called "spurs") is the waveform DAC. Nonlinearity in the DAC leads to harmonics that are aliased, or "folded back", into the passband of the function generator. These spurs are most significant when there is a simple fractional relationship between the signal frequency and the function generator's sampling frequency (200 MHz). For example, at 75 MHz, the DAC produces harmonics at 150 MHz and 225 MHz. These harmonics, which are 50 MHz and 25 MHz from the function generator's 200 MHz sampling frequency, will appear as spurs at 50 MHz and 25 MHz.

Another source of non-harmonic spurs is the coupling of unrelated signal sources (such as the microprocessor clock) into the output signal.
These spurs usually have a constant amplitude (≤ -75 dBm or 112 µVpp) regardless of the signal's amplitude and are most troublesome at signal amplitudes below 100 mVpp. To obtain low amplitudes with minimum spurious content, keep the function generator's output level relatively high and use an external attenuator if possible.

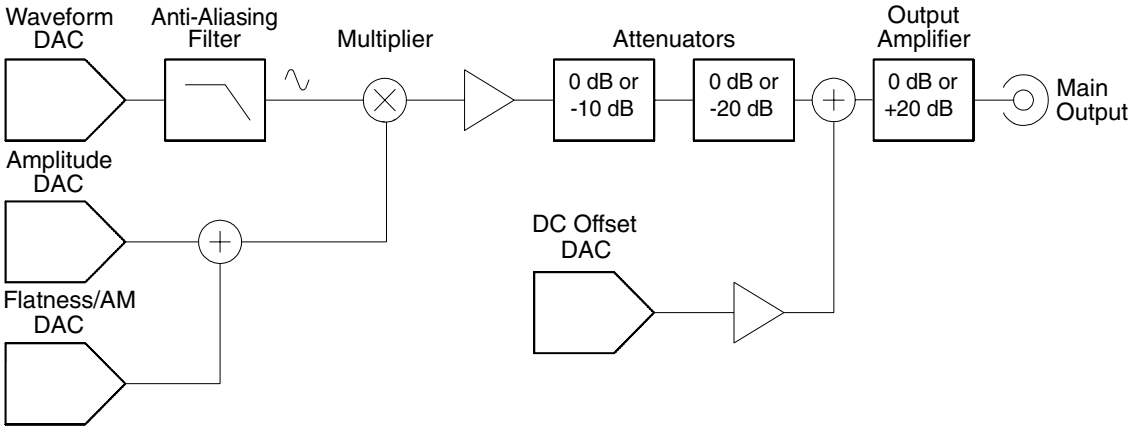
Phase Noise Phase noise results from small, instantaneous changes in the output frequency ("jitter"). It is seen as an elevation of the apparent noise floor near the fundamental frequency and increases at 6 dBc/octave with the carrier frequency. The 33250A's phase noise specification represents the sum of all noise components in a 30 kHz band centered on the fundamental frequency. This "integrated phase noise" is related to jitter by the following equation.

$$\text{Jitter in Seconds (rms)} = \frac{1}{2\pi \times \text{Frequency}} \times 10^{(\text{Phase Noise in dBc}/20)}$$

Quantization Errors Finite DAC resolution (12 bits) leads to voltage quantization errors. Assuming the errors are uniformly distributed over a range of ±0.5 least-significant bit (LSB), the equivalent noise level is -74 dBc for a sine wave that uses the full DAC range (4,096 levels). Similarly, finite-length waveform memory leads to phase quantization errors. Treating these errors as low-level phase modulation and assuming a uniform distribution over a range of ±0.5 LSB, the equivalent noise level is -76 dBc for a sine wave that is 16K samples long. All of the 33250's standard waveforms use the entire DAC range and are 16K samples in length. Any arbitrary waveforms that use less than the entire DAC range, or that are specified with fewer than 16,384 points, will exhibit proportionally higher relative quantization errors.

Output Amplitude Control

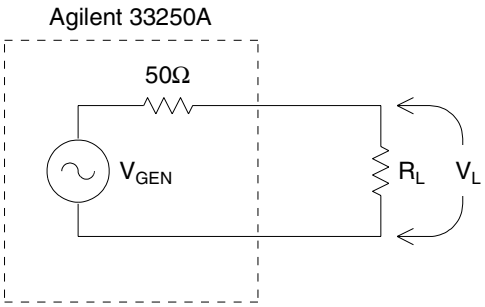
The 33250A uses an analog multiplier to control the signal amplitude over a 10 dB range. As shown below, one of the multiplier's inputs is routed from the output of an anti-aliasing filter. The other input is routed from a dc control voltage which is the sum of the outputs from two DACs. One of the DACs is set to a nominal voltage corresponding to the desired output amplitude. The second DAC provides a voltage to correct for frequency response variations in the function generator. The calibration procedures for the 33250A provide all of the information necessary to calculate the proper DAC values (see the Agilent 33250A *Service Guide*). Two attenuators (-10 dB and -20 dB) and an amplifier (+20 dB) are used in various combinations to control the output amplitude in 10 dB steps over a wide range of amplitude values (1 mVpp to 10 Vpp).



Note that the dc offset is summed with the ac signal *after* the attenuator stages but *before* the output amplifier. This allows relatively small ac signals to be offset by relatively large dc voltages. For example, you can offset a 100 mVpp signal by almost 5 Vdc (into a 50Ω load).

When changing ranges, the 33250A always switches attenuators such that the output voltage never exceeds the current amplitude setting. However, momentary disruptions or "glitches" caused by switching can cause problems in some applications. For this reason, the 33250A incorporates a *range hold* feature to "freeze" the attenuator and amplifier switches in their current states. However, the amplitude and offset accuracy and resolution (as well as waveform fidelity) may be adversely affected when reducing the amplitude below the expected range change.

As shown below, the 33250A has a fixed series output impedance of 50Ω, forming a voltage divider with the load resistance.

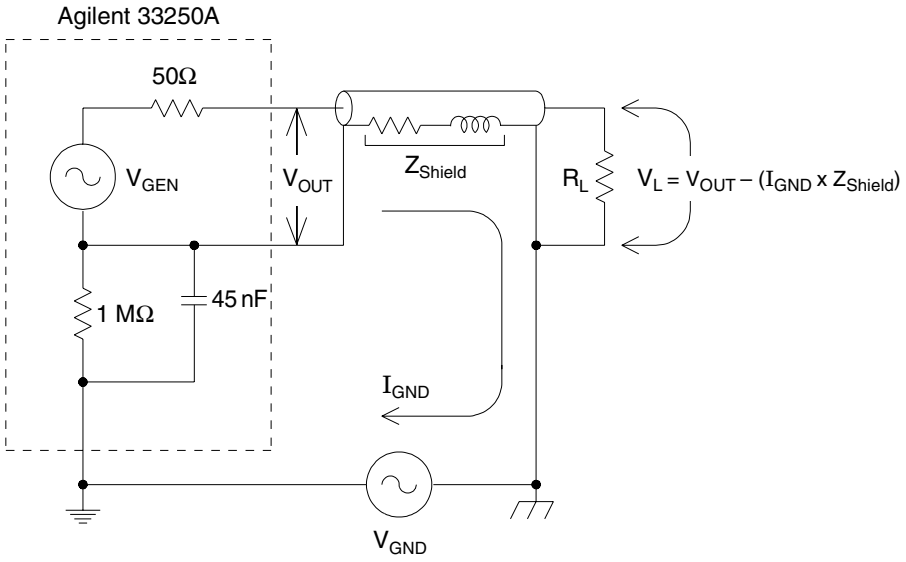


As a convenience, you can specify the load impedance as seen by the function generator and thereby display the correct load voltage. If the actual load impedance is different than the value specified, the displayed amplitude, offset, and high/low levels will be incorrect. Variations in source resistance are measured and taken into account during an instrument calibration. Therefore, the accuracy of the load voltage depends primarily on the accuracy of the load resistance as shown below.

$$\Delta V_L(\%) \cong \frac{50}{R_L + 50} \times \Delta R_L(\%)$$

Ground Loops

Except for its remote interface connectors and trigger connector, the 33250A is isolated from chassis (earth) ground. This isolation helps to eliminate ground loops in your system and also allows you to reference the output signal to voltages other than ground. The illustration on the following page shows the function generator connected to a load through a coaxial cable. Any difference in ground potentials (V_{GND}) will tend to make current I_{GND} flow in the shield of the cable, thus causing a voltage drop due to the shield's impedance (Z_{Shield}). The resulting voltage drop ($I_{GND} \times Z_{Shield}$) appears as an error in the load voltage. However, since the instrument is isolated, there is a high series impedance (typically 1 MΩ in parallel 45 nF) in the path to oppose the flow of I_{GND} and thereby minimize this effect.



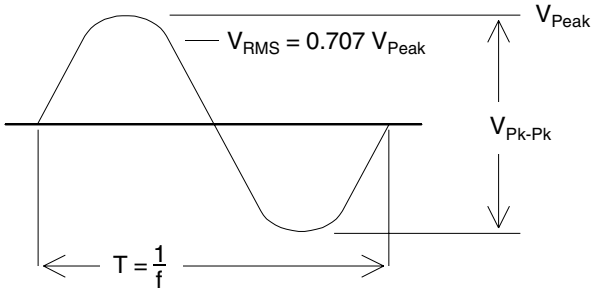
Ground Loop Effects

At frequencies above a few kilohertz, a coaxial cable's shield becomes inductive, rather than resistive, and the cable acts as a transformer. When this happens, it tends to force the shield and center-conductor currents to be equal but opposite. For any voltage drop in the shield due to I_{GND}, there is a similar drop in the center conductor. This is known as the *balun effect* and it reduces ground loops at higher frequencies. Note that lower shield resistance causes the balun effect to become more of a factor at lower frequencies. Therefore, coaxial cables with two or three braided shields are much better than those with single braids.

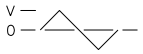
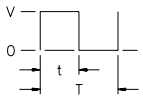
To reduce errors due to ground loops, connect the function generator to the load using a high-quality coaxial cable and ground it at the load through the cable's shield. If possible, make sure the function generator and the load are connected to the same electrical outlet to minimize further differences in ground potential.

Attributes of AC Signals

The most common ac signal is a sine wave. In fact, any periodic signal can be represented as the sum of different sine waves. The magnitude of a sine wave is usually specified by its peak, peak-to-peak, or root-mean-square (RMS) value. All of these measures assume that the waveform has zero offset voltage.



The *peak voltage* of a waveform is the maximum absolute value of all points in the waveform. The *peak-to-peak voltage* is the difference between the maximum and minimum. The *RMS voltage* is found by summing the squares of the voltages of every point in the waveform, dividing the sum by the number of points, and then taking the square root of that quotient. The RMS value of a waveform also represents the one-cycle average power in the signal: Power = V_{RMS}^2 / R_L. *Crest factor* is the ratio of a signal's peak value to its RMS value and will differ according to waveshape. The table below shows several common waveforms with their respective crest factors and RMS values.

Waveform Shape	Crest Factor (C.F.)	AC RMS	AC+DC RMS
	1.414	$\frac{V}{1.414}$	$\frac{V}{1.414}$
	1.732	$\frac{V}{1.732}$	$\frac{V}{1.732}$
	$\sqrt{\frac{T}{t}}$	$\frac{V}{C.F.} \times \sqrt{1-\left(\frac{1}{C.F.}\right)^2}$	$\frac{V}{C.F.}$

You may occasionally see ac levels specified in "decibels relative to 1 milliwatt" (dBm). Since dBm represents a power level, you will need to know the signal's RMS voltage and the load resistance in order to make the calculation.

$$dBm = 10 \times \log_{10}(P/0.001) \qquad \textit{where } P = V_{RMS}^2 / R_L$$

For a sine wave into a 50Ω load, the following table relates dBm to voltage.

dBm	RMS Voltage	Peak-to-Peak Voltage
+23.98 dBm	3.54 Vrms	10.00 Vpp
+13.01 dBm	1.00 Vrms	2.828 Vpp
+10.00 dBm	707 mVrms	2.000 Vpp
+6.99 dBm	500 mVrms	1.414 Vpp
0.00 dBm	224 mVrms	632 mVpp
-6.99 dBm	100 mVrms	283 mVpp
-10.00 dBm	70.7 mVrms	200 mVpp
-36.02 dBm	3.54 mVrms	10.0 mVpp

For 75Ω or 600Ω loads, use the following conversions.

$$dBm\ (75\Omega) = dBm\ (50\Omega) - 1.76$$
$$dBm\ (600\Omega) = dBm\ (50\Omega) - 10.79$$

Modulation

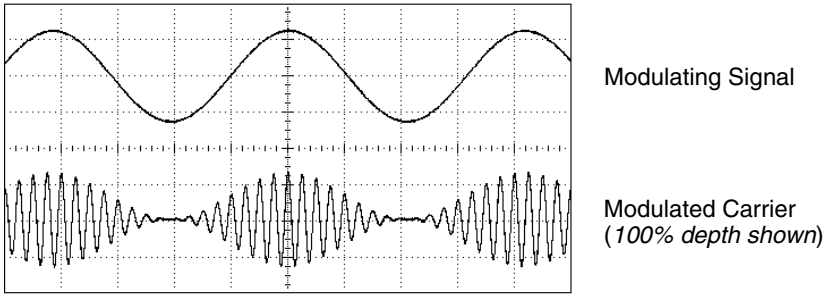
Modulation is the process of modifying a high-frequency signal (called the *carrier signal*) with low-frequency information (called the *modulating signal*). The carrier and modulating signals can have any waveshape, but the carrier is usually a sine waveform.

The two most common types of modulation are *amplitude modulation* (AM) and *frequency modulation* (FM). These two forms of modulation modify the carrier's amplitude or frequency, respectively, according to the instantaneous value of the modulating signal. A third type of modulation is *frequency-shift keying* (FSK), where the output frequency "shifts" between two frequencies depending on the state of a digital modulating signal.

The function generator will accept an *internal* or *external* modulation source. If you select the *internal* source, the modulated waveform is generated by a DDS process running in an internal digital signal processor (DSP). If you select the *external* source, the modulated waveform is controlled by the signal level present on the function generator's rear-panel *Modulation In* connector. The external signal is sampled and digitized by an analog-to-digital converter (ADC) and is then routed to the DSP. With either modulation source, the result is a stream of digital samples representing the modulating waveform.

Note that for FSK, the output frequency is determined by the signal level present on the rear-panel *Trig In* connector.

Amplitude Modulation (AM) For AM, the DSP routes modulation samples to a digital-to-analog converter (DAC) which then controls the output amplitude via an analog multiplier. The DAC and multiplier are the same as those used to set the function generator's output level (*see "Output Amplitude Control" on page 304*). This form of AM is called "double sideband transmitted carrier" and is the type of modulation used by most AM radio stations.



Amplitude Modulation

The amount of amplitude modulation is called the *modulation depth* which refers to the portion of the amplitude range that will be used by the modulation. For example, a depth setting of 80% causes the amplitude to vary from 10% to 90% of the amplitude setting (90% – 10% = 80%) with either an internal or a full-scale (±5V) external modulating signal.

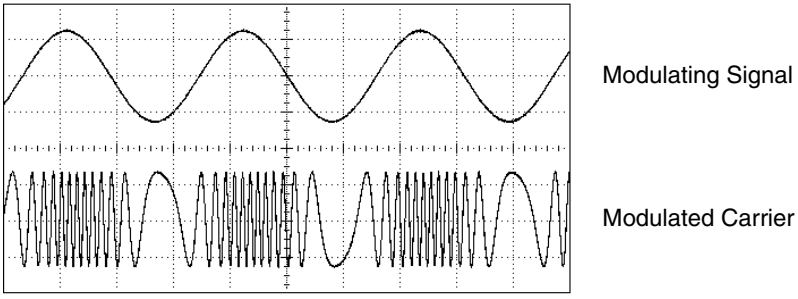
Frequency Modulation (FM) For FM, the DSP uses modulation samples to modify the output frequency of the instrument by changing the content of the PIR (*see "Direct Digital Synthesis" on page 295*). Note that since the rear-panel *Modulation In* connector is dc-coupled, you can use the 33250A to emulate a voltage-controlled oscillator (VCO).

The variation in frequency of the modulating waveform from the carrier frequency is called the *frequency deviation*. Waveforms with frequency deviations less than 1% of the modulating signal's bandwidth are referred to as *narrowband FM*. Waveforms with larger deviations are referred to as *wideband FM*. The bandwidth of the modulated signal can be approximated by the following equations.

BW ≅ 2 x (Modulating Signal Bandwidth) *For narrowband FM*

BW ≅ 2 x (Deviation + Modulating Signal Bandwidth) *For wideband FM*

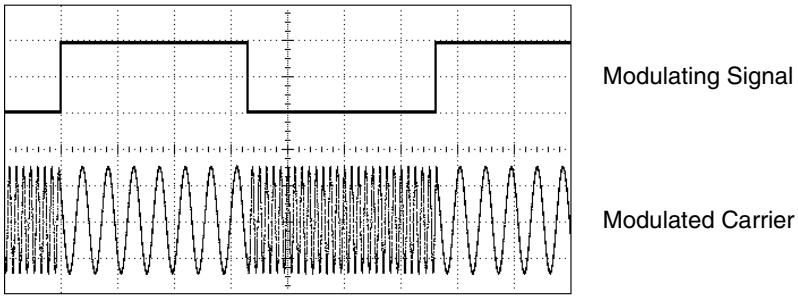
In the United States, commercial FM stations usually have a modulation bandwidth of 15 kHz and deviation of 75 kHz, making them "wideband". Therefore, the modulated bandwidth is: 2 x (75 kHz + 15 kHz) = 180 kHz. Channel spacing is 200 kHz.



Frequency Modulation

Frequency-Shift Keying (FSK) FSK is similar to FM except the frequency alternates between two preset values. The rate at which the output shifts between the two frequencies (called the "carrier frequency" and the "hop frequency") is determined by the internal rate generator or the signal level on the rear-panel *Trig In* connector. Frequency changes are instantaneous and phase-continuous.

The internal modulating signal is a square waveform with 50% duty cycle. You can set the internal FSK rate from 2 mHz to 100 kHz.



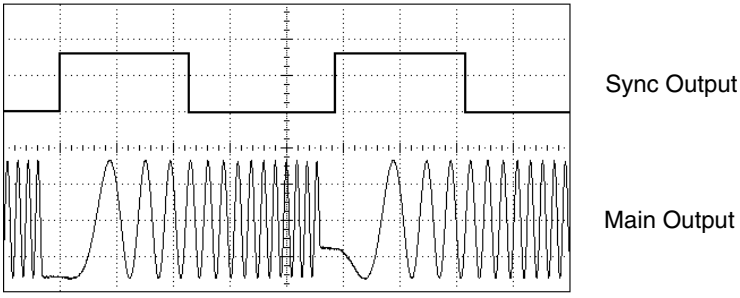
Frequency-Shift Keying

Frequency Sweep

Frequency sweeping is similar to FM but no modulating waveform is used. Instead, the internal DSP sets the output frequency based on either a linear or logarithmic function. In a *linear* sweep, the output frequency changes in a constant "hertz per second" manner. In a *logarithmic* sweep, the output frequency changes in a constant "octaves per second" or "decades per second" manner. Logarithmic sweeps are useful for covering wide frequency ranges where resolution at low frequencies would potentially be lost in a linear sweep.

You can generate a sweep using an *internal* trigger source or an *external* hardware trigger source. When the *internal* source is selected, the function generator outputs a continuous sweep at a rate determined by the *sweep time* specified. When the *external* source is selected, the function generator will accept a hardware trigger applied to the rear-panel *Trig In* connector. The function generator initiates one sweep each time *Trig In* receives a TTL pulse.

A sweep consists of a finite number of small frequency steps. Since each step takes the same amount of time, longer sweep times result in smaller steps and therefore better resolution. The number of discrete frequency points in the sweep is automatically calculated by the function generator and is based on the *sweep time* you select.

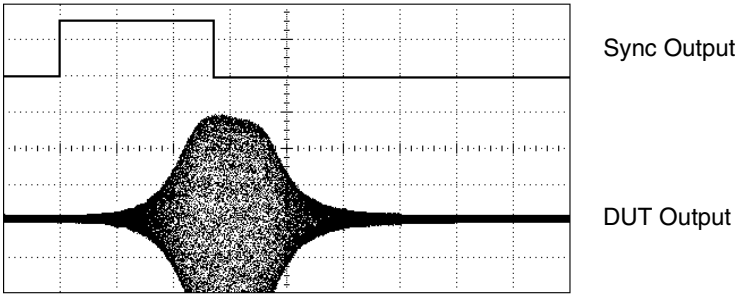


Frequency Sweep

For triggered sweeps, the trigger source can be an external signal, the Trigger key, or a command received from the remote interface. The input for external trigger signals is the rear-panel *Trig In* connector. This connector accepts TTL-compatible levels and is referenced to chassis ground (not floating ground). When not used as an input, the *Trig In* connector can be configured as an output to enable the 33250A to trigger other instruments at the same time as its internal trigger occurs.

Sync and Marker Signals The output from the front-panel *Sync* connector goes "high" at the beginning of each sweep. If you have disabled the Marker function, the Sync signal goes "low" at the midpoint of the sweep. However, if you have enabled the Marker function, the Sync signal goes "low" when the output frequency reaches the specified marker frequency. The marker frequency *must* be between the specified start frequency and stop frequency.

You can use the Marker function to identify a notable frequency in the response of a device under test (DUT) – for example, you may want to identify a resonance. To do this, connect the *Sync* output to one channel of your oscilloscope and connect the DUT output to another channel. Then, trigger the oscilloscope with the rising edge of the Sync signal to position the start frequency on the left side of the screen. Adjust the marker frequency until the falling edge of the Sync signal lines up with the interesting feature in the device's response. You can then read the frequency from the front-panel display of the 33250A.

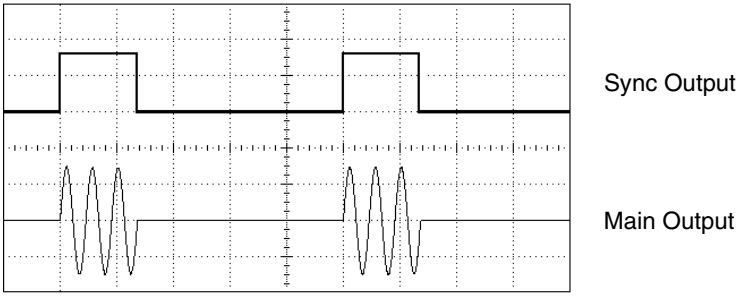


Sweep with Marker at DUT Resonance

Burst

You can configure the function generator to output a waveform with a specified number of cycles, called a *burst*. You can use burst in one of two modes: *N-Cycle Burst* (also called "triggered burst") or *Gated Burst*.

N-Cycle Burst An N-Cycle burst consists of a specific number of waveform cycles (1 to 1,000,000) and is always initiated by a trigger event. You can also set the burst count to "Infinite" which results in a continuous waveform once the function generator is triggered.



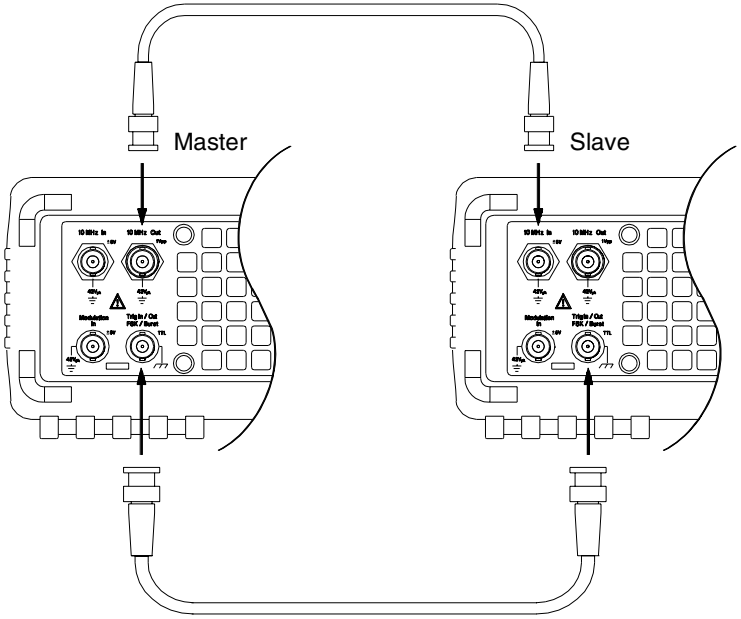
Three-Cycle Burst Waveform

For bursts, the trigger source can be an external signal, an internal timer, the Trigger key, or a command received from the remote interface. The input for external trigger signals is the rear-panel *Trig In* connector. This connector accepts TTL-compatible levels and is referenced to chassis ground (not floating ground). When not used as an input, the *Trig In* connector can be configured as an output to enable the 33250A to trigger other instruments at the same time as its internal trigger occurs.

The effect of a trigger can be delayed up to 85 seconds (in increments of 100 picoseconds) to synchronize the start of the burst with other events. You can also insert a *trigger delay* to compensate for the cable delays and response times of other instruments in your system.

An N-Cycle burst always begins and ends at the same point in the waveform, called the *start phase*. A starting phase of 0° corresponds to the beginning of the waveform record and 360° corresponds to the end of the waveform record.

As an example, suppose that your application requires two 5 MHz sine waveforms that are exactly 90° out of phase from one another. You can use two 33250A's as described below. First, designate one function generator as the "master" and the other as the "slave". As shown below, connect the master's *10 MHz Out* connector to the slave's *10 MHz In* connector using a high-quality coaxial cable. This configuration will ensure that both instruments will generate exactly the same frequency and that there will not be any long-term phase shift between the two instruments. Next, connect the two *Trig In*/*Out* connectors together to allow the master to trigger the slave.



After making the connections shown above, follow the steps below to configure the two instruments.

1 Configure both instruments to output a 5 MHz sine waveform.

2 On both instruments, enable the N-Cycle burst mode, set the burst count to three cycles, and set the starting phase to 0 degrees.

3 On the "master", select the *Internal* trigger source and enable the "trigger out" signal with a *rising edge* from the *Trig Out* connector.

4 On the "slave", select the *External* trigger source and enable triggering on the *rising edge* of the trigger signal.

5 Using an oscilloscope, verify that both instruments are now generating a three-cycle burst waveform. Then, adjust the *trigger delay* parameter of one instrument to bring the two bursts into alignment with each other. The two instruments are now synchronized and will remain synchronized until you adjust the trigger delay parameters.

6 Set the starting phase of one instrument to 90°. Then, adjust the burst count on each instrument as needed for your application. If continuous burst waveforms are required, select an "Infinite" burst count on both instruments and enable manual triggering on the "master".

In this example, the trigger delay parameters become, in effect, system calibration constants. Once established, they keep the two instruments aligned in time even if their frequency or starting phase are changed. Each time the master triggers the slave, the two instruments are re-synchronized. If the power is cycled, you can realign the instruments by restoring the previous trigger delays. Note that different delay values may be required if a different pair of instruments is used or if different waveform functions are selected.

Gated Burst In the Gated burst mode, the output waveform is either "on" or "off" based on the level of the external signal applied to the rear-panel *Trig In* connector. When the gate signal is *true*, the function generator outputs a continuous waveform. When the gate signal goes *false*, the current waveform cycle is completed and then the function generator stops while remaining at the voltage level corresponding to the starting burst phase of the selected waveform. For a noise waveform, the output stops immediately when the gate signal goes false.

Specifications

WAVEFORMS

Standard Waveforms: Sine, Square, Ramp, Pulse, Noise, Sin(x)/x, Exponential Rise, Exponential Fall, Negative Ramp, Cardiac, DC Volts

Arbitrary Waveforms

Waveform Length:	1 to 64K points
Amplitude Resolution:	12 bits (including sign)
Repetition Rate:	1 µHz to 25 MHz
Sample Rate:	200 MSa/s
Filter Bandwidth:	50 MHz
Non-Volatile Memory:	Four 64K waveforms [1]

FREQUENCY CHARACTERISTICS

Sine:	1 µHz to 80 MHz
Square:	1 µHz to 80 MHz
Ramp:	1 µHz to 1 MHz
Pulse:	500 µHz to 50 MHz
Noise (Gaussian)**:**	50 MHz bandwidth
Arb:	1 µHz to 25 MHz
Resolution:	1 µHz; except pulse, 5 digits
Accuracy (1 year)**:**	2 ppm, 18°C to 28°C 3 ppm, 0°C to 55°C

SINEWAVE SPECTRAL PURITY

Harmonic Distortion

	< 3 Vpp [2]	> 3 Vpp
DC to 1 MHz:	-60 dBc	-55 dBc
1 MHz to 5 MHz:	-57 dBc	-45 dBc
5 MHz to 80 MHz:	-37 dBc	-30 dBc

Total Harmonic Distortion

DC to 20 kHz:	< 0.2% + 0.1 mVrms

Spurious (non-harmonic) [3]

DC to 1 MHz:	-60 dBc
1 MHz to 20 MHz:	-50 dBc
20 MHz to 80 MHz:	-50 dBc + 6 dBc/octave

Phase Noise (30 kHz band)

10 MHz	< -65 dBc (typical)
80 MHz	< -47 dBc (typical)

SIGNAL CHARACTERISTICS

Square Wave

Rise / Fall Time:	< 8 ns [4]
Overshoot:	< 5%
Asymmetry:	1% of period + 1 ns
Jitter (rms)	
< 2 MHz:	0.01% + 525 ps
≥ 2 MHz:	0.1% + 75 ps
Duty Cycle	
≤ 25 MHz:	20.0% to 80.0%
25 MHz to 50 MHz:	40.0% to 60.0%
50 MHz to 80 MHz:	50.0% (fixed)

Pulse

Period:	20.00 ns to 2000.0 s
Pulse Width:	8.0 ns to 1999.9 s
Variable Edge Time:	5.00 ns to 1.00 ms
Overshoot:	< 5%
Jitter (rms):	100 ppm + 50 ps

Ramp

Linearity:	< 0.1% of peak output
Symmetry:	0.0% to 100.0%

Arb

Minimum Edge Time:	< 10 ns
Linearity:	< 0.1% of peak output
Settling Time:	< 50 ns to 0.5% of final value
Jitter (rms):	30 ppm + 2.5 ns

---

[1] A total of four waveforms can be stored.

[2] Harmonic distortion at low amplitudes is limited by a -70 dBm floor.

[3] Spurious noise at low amplitudes is limited by a -75 dBm floor.

[4] Edge time decreased at higher frequency.

OUTPUT CHARACTERISTICS [1]

Amplitude (into 50Ω)**:** 10 mVpp to 10 Vpp [2]
- Accuracy (at 1 kHz, >10 mVpp, Autorange On): ± 1% of setting ±1 mVpp
- Flatness (sinewave relative to 1 kHz, Autorange On)
  - < 10 MHz: ± 1% (0.1 dB) [3]
  - 10 MHz to 50 MHz: ± 2% (0.2 dB)
  - 50 MHz to 80 MHz ± 5% (0.4 dB)
- Units: Vpp, Vrms, dBm, High Level, Low Level
- Resolution: 0.1 mV or 4 digits

Offset (into 50Ω): ± 5 Vpk ac + dc
- Accuracy: 1% of setting + 2 mV + 0.5% of amplitude

Waveform Output
- Impedance: 50Ω typical (fixed) >10 MΩ (output disabled)
- Isolation: 42 Vpk max. to Earth
- Protection: Short-circuit protected; [4] Overload relay automatically disables main output

MODULATION CHARACTERISTICS

AM Modulation
- Carrier Waveforms: Sine, Square, Ramp, Arb
- Modulating Waveforms: Sine, Square, Ramp, Noise, Arb
- Modulating Frequency: 2 mHz to 20 kHz
- Depth: 0.0% to 120.0%
- Source: Internal / External

FM Modulation
- Carrier Waveforms: Sine, Square, Ramp, Arb
- Modulating Waveforms: Sine, Square, Ramp, Noise, Arb
- Modulating Frequency: 2 mHz to 20 kHz
- Peak Deviation: DC to 80 MHz
- Source: Internal / External

FSK
- Carrier Waveforms: Sine, Square, Ramp, Arb
- Modulating Waveforms: 50% duty cycle square
- Internal Rate: 2 mHz to 100 kHz
- Frequency Range: 1 µHz to 80 MHz
- Source: Internal / External

External Modulation Input
- Voltage Range: ± 5V full scale
- Input Impedance: 10 kΩ
- Frequency: DC to 20 kHz

BURST

- Waveforms: Sine, Square, Ramp, Pulse, Noise, Arb
- Frequency: 1 µHz to 80 MHz [5]
- Burst Count: 1 to 1,000,000 cycles, or Infinite
- Start / Stop Phase: -360.0° to +360.0°
- Internal Period: 1 ms to 500 s
- Gate Source: External Trigger
- Trigger Source: Single, External, or Internal Rate
- Trigger Delay
  - N-Cycle, Infinite: 0.0 ns to 85.000 s

SWEEP

- Waveforms: Sine, Square, Ramp, Arb
- Type: Linear or Logarithmic
- Direction: Up or Down
- Start F / Stop F: 100 µHz to 80 MHz
- Sweep Time: 1 ms to 500 s
- Trigger: Single, External, or Internal
- Marker: Falling edge of Sync signal (programmable)

---

[1] Add 1/10th of output amplitude and offset specification per °C for operation outside of 18 °C to 28 °C range (1-year specification).

[2] 20 mVpp to 20 Vpp into open-circuit load.

[3] dB rounded to 1 digit. Instrument adheres to "%" specification.

[4] Short-circuit protected to ground at all times.

[5] Sine and square waveforms above 25 MHz are allowed only with an "Infinite" burst count.

SYSTEM CHARACTERISTICS

Configuration TImes (typical) [1]

Function Change	
Standard: [2]	102 ms
Pulse:	660 ms
Built-In Arb: [2]	240 ms
Frequency Change:	24 ms
Amplitude Change:	50 ms
Offset Change:	50 ms
Select User Arb:	< 400 ms for < 16K points
Modulation Change:	< 200 ms

Arb Download Times GPIB / RS-232 (115 Kbps) [3]

Arb Length	Binary	ASCII Integer	ASCII Real
64K points	23 sec	92 sec	154 sec
16K points	6 sec	23 sec	39 sec
8K points	3 sec	12 sec	20 sec
4K points	1.5 sec	6 sec	10 sec
2K points	0.75 sec	3 sec	5 sec

TRIGGER CHARACTERISTICS

Trigger Input

Input Level:	TTL-compatible
Slope:	Rising or falling (selectable)
Pulse Width:	> 100 ns
Input Impedance:	10 kΩ, DC coupled
Latency	
Sweep:	< 10 µs (typical)
Burst:	< 100 ns (typical)
Jitter (rms)	
Sweep:	2.5 µs
Burst:	1 ns; except pulse, 300 ps

Trigger Output

Level:	TTL-compatible into 50Ω
Pulse Width:	> 450 ns
Maximum Rate:	1 MHz
Fanout:	≤ 4 Agilent 33250As

CLOCK REFERENCE

Phase Offset

Range:	-360° to +360°
Resolution:	0.001°

External Reference Input

Lock Range:	10 MHz ± 35 kHz
Level:	100 mVpp to 5 Vpp
Impedance:	1 kΩ nominal, ac coupled
Lock Time:	< 2 s

Internal Reference Output

Frequency:	10 MHz
Level:	632 mVpp (0 dBm), nominal
Impedance:	50Ω nominal, ac coupled

SYNC OUTPUT

Level:	TTL-compatible into > 1 kΩ
Impedance:	50Ω nominal

---

[1] Time to change parameter and output new signal.

[2] Modulation or sweep off.

[3] Times for 5-digit integer and 12-digit real numbers.

GENERAL SPECIFICATIONS

Power Supply:	100-240 V (±10%) for 50-60 Hz operation, 100-127 V (±10%) for 50-400 Hz operation. IEC 60664 CAT II
Power Consumption:	140 VA
Operating Environment:	0 °C to 55 °C 80% R.H. to 40 °C
Operating Altitude:	3000 meters
Pollution Degree:	Indoor or Sheltered Use, IEC 60664 Degree 2
Storage Temperature:	-30 °C to 70 °C
Stored States:	Four (4) named user configurations
Power-On State:	Default or Last
Interface:	IEEE-488 and RS-232 standard
Language:	SCPI-1997, IEEE-488.2
Dimensions (WxHxD) Bench Top: Rack Mount:	 254 x 104 x 374 mm 213 x 89 x 348 mm
Weight:	4.6 kg

Safety Designed to:	EN61010-1, CSA1010.1, UL-3111-1
EMC Tested to: [1]	IEC-61326-1 IEC-61000-4-3 criteria B IEC-61000-4-6 criteria B
Acoustic Noise:	40 dBA
Warm-Up Time:	1 hour
Calibration Interval:	1 year
Warranty:	3 years standard
Accessories Included:	User's Guide, Service Guide, Quick Reference Guide, Test Data, Connectivity Software, RS-232 Cable, Power Cord

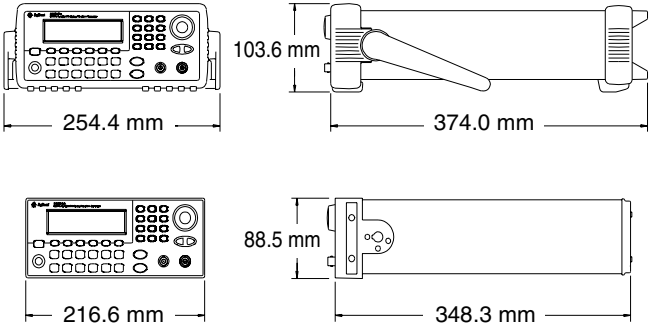
[1] Radiated and conducted immunity testing: When the product is tested at 3 V/m according to IEC/EN 61000-4-3:1995 or tested at 3 Vrms according to IEC/EN 61000-4-6:1996, the product may not meet criteria A, but does meet criteria B.

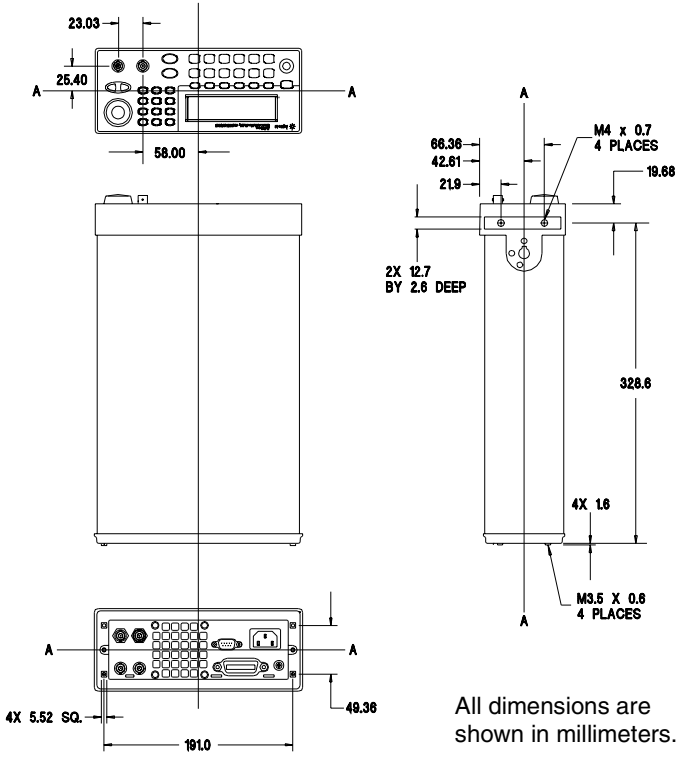
This ISM device complies with Canadian ICES-001.

Cet appareil ISM est conforme à la norme NMB-001 du Canada.

N10149

PRODUCT DIMENSIONS





Index

*If you have questions relating to the operation of the Agilent 33250A, call **1-800-452-4844** in the United States, or contact your nearest Agilent Technologies Office.*

10 MHz In connector, 223
10 MHz Out connector, 223
*CLS command, 238
*ESE command, 237
*IDN? command, 214
*LRN? command, 216
*OPC command, 217, 238
*OPC? command, 184, 193, 217
*PSC command, 238
*RCL command, 210
*RST command, 216
*SAV command, 209
*SRE command, 235
*STB? command, 229, 235
*TRG command, 184, 193, 196
*TST? command, 216
*WAI command, 184, 193, 217

A

ac connector, 6
ActiveX drivers, 277
address, GPIB, 118
Agilent Express, 7
air flow, 28
aliasing, 297
AM (amplitude modulation)
  carrier waveform, 68
  depth, 70, 310
  example in BASIC, 279
  example in Visual Basic, 283
  example in Visual C++, 289
  front-panel operation, 34
  modulation depth, 70, 171, 310
  modulation source, 71, 170
  modulation waveform, 69, 171
  overview, 169
  remote operation, 169
  tutorial description, 309
AM:DEPT command, 171
AM:INT:FREQ command, 171
AM:INT:FUNC command, 170
AM:SOURce command, 170
AM:STATe command, 171
amplitude, 18, 146
  arb waveform limits, 53, 147
  dBm limits, 156
  high/low level, 53, 157
  load limit, 52, 146, 156
  offset limits, 52, 156
  range hold, 60
  termination (load), 52, 146, 156
  tutorial description, 304
  unit limit, 52
  units selection, 56
amplitude modulation (AM)
  carrier waveform, 68
  depth, 70, 310
  example in BASIC, 279
  example in Visual Basic, 283
  example in Visual C++, 289
  front-panel operation, 34
  modulation depth, 70, 171, 310
  modulation source, 71, 170
  modulation waveform, 69, 171
  overview, 169
  remote operation, 169
  tutorial description, 309
amplitude units, converting, 19
angle, burst phase, 191, 192
anti-aliasing filters, 295
application examples
  BASIC for Windows, 278
  Visual Basic, 282
  Visual C++, 287
application programs, 275
APPLy:DC command, 151
APPLy:NOISe command, 151
APPLy:PULSe command, 150
APPLy:RAMP command, 150
APPLy:SIN command, 150
APPLy:SQUare command, 150
APPLy:USER command, 151
APPLy? command, 152
arbitrary waveforms
  amplitude limits, 147
  as modulating waveshape, 108
  binary values, 201
  built-in waveforms, 24, 199
  calculating crest factor, 208
  creating from front panel, 103
  deleting from memory, 207
  download times, 322
  error messages, 272
  example in BASIC, 280
  example in Visual Basic, 284
  example in Visual C++, 291
  front-panel operation, 24
  front-panel rules, 106
  handshake modes, 219
  naming, 107, 203
  overview, 198
  point interpolation, 104
  rules, 108
  tutorial description, 298
  using binary values, 201
  using floating-point values, 200
  using integer values, 201
attenuator settings, 160
auto power-down recall, 109, 212
autorange, 60, 160, 304

B

balun effect, 306
BASIC examples, 278
baud rate (RS-232), 45, 120
beeper enable/disable, 113, 216
binary block format, 202
binary download of arbs, 201
bit definitions
  questionable data register, 231
  standard event register, 233
  status byte register, 227
blank display, 113
block format (binary), 202

BNC connectors
10 MHz In, 223
10 MHz Out, 223
Modulation In, 71, 77, 81
Output, 60, 162
Sync Output, 62, 186
Trig In, 87, 101, 184
Trig Out, 88, 102, 185
break, 218
brightness, display, 114
built-in arb waveforms, 199, 204
built-in help system, 25
bulb-saver mode, 113
bumpers, removing, 27
burst, 89, 187
burst count, 93, 190
burst period, 94, 191
burst phase, 95, 191
burst type, 90, 189
example in BASIC, 280
example in Visual Basic, 284
example in Visual C++, 290
external gated mode, 90, 189
external trigger, 101, 192
front-panel operation, 40
gate polarity, 194, 197
gated burst, 90, 187, 317
n-cycle burst, 90, 189, 315
overview, 187
starting phase, 95, 191
trigger delay, 193, 196, 315
trigger out signal, 97, 102
trigger source, 96
triggered mode, 90, 187
waveform frequency, 92
BURSt
:GATE:POL command, 194, 197
:INT:PERiod command, 191
:MODE command, 189
:NCYCles command, 190
:PHASe command, 191
:STATe command, 192
burst phase
degrees vs. radians, 192
bus, interface configuration, 118
bus (software) trigger, 100, 195
byte order, binary transfers, 203

C

cable pinout (RS-232), 222
cable (RS-232), 15
calibration
certificate, 15
commands, 239
error messages, 271
message, 126
read count, 125, 240
securing, 239
security code, 123
setups, 239
text message, 240
unsecuring, 239
CALibration
:COUNt? command, 240
:SECure:CODE command, 240
:STATe command, 239
:SETup command, 239
:STRing command, 240
:VALue command, 239
CALibration? command, 239
cardiac waveform, 199
carriage return, 244
carrying handle
changing positions, 16
removing, 27
catalog, arb waveforms, 206
CD-ROM, 15, 276
center frequency, sweep, 182
chassis ground, 6
comma separator, 116
command
errors, 112, 213, 247
libraries, 277
parameter types, 245
reference, 129
summary, 131
terminators, 244
command examples
BASIC for Windows, 278
Visual Basic, 282
Visual C++, 287
communication problems, 222
connectors (BNC), 223
10 MHz In, 223
10 MHz Out, 223
Modulation In, 71, 77, 81
Output, 60, 162
Sync Output, 62, 186
Trig In, 87, 101, 184
Trig Out, 88, 102, 185
context-sensitive help, 25
contrast, display, 114
count (burst), 93, 190
crest factor, 208, 307
custom naming
arbitrary waveforms, 107, 203
stored states, 43, 110, 211
cycle count (burst), 93, 190

D

data bits (RS-232), 120
data frame (RS-232), 220, 221
DATA
:ATTR:CFACtor? command, 208
:CATalog? command, 206
:COPY command, 203
:DAC VOLATILE command, 201
:DELete:ALL command, 207
:DELete command, 207
:NVOL:CAT? command, 206
:FREE? command, 206
DATA VOLATILE command, 200

dBc, 302
dBm, 56, 165, 308
dc offset
  amplitude limits, 54, 148, 158
  arb limits, 55, 148, 158
  front-panel selection, 20
  load limits, 54, 148, 158
dc voltage, 20, 157
DDS (direct digital synthesis), 295
decimal point, 116
degrees, 191
delay, trigger, 193, 196
deleting arb waveforms, 207
deleting stored states, 211
depth (AM), 171, 310
deviation (FM), 76, 174, 311
device clear, 218, 246
digital signal processor (DSP), 309
digits separator, 116
dimensions (33250A), 322
direct digital synthesis (DDS), 295
display, 214
  brightness, 114
  bulb saver mode, 113
  contrast, 114
  displaying message, 115, 215
  graph mode, 23
  enable/disable, 115, 214
  number format, 116
  overview, 4
DISPlay
  :TEXT:CLEar command, 215
  :TEXT command, 215
download times, arbs, 322
DSP (digital signal processor), 309
DTR/DSR (RS-232), 121, 220
duty cycle
  definition, 58, 161
  frequency limits, 58, 155, 161
  front-panel selection, 21
  modulation limits, 58, 161
  remote selection, 161

E

edge time, 22, 66, 168
EOI (end-or-identify), 244
errors, 112, 213, 247
  arb waveform, 272
  calibration, 271
  "data out of range", 261
  execution, 252
  instrument, 268
  number allowed in queue, 213
  phase unlock, 224
  query errors, 267
  self-test errors, 269
  "settings conflict", 253
even parity, 120
example programs
  BASIC for Windows, 278
  Visual Basic, 282
  Visual C++, 287
exchange instrument, 7
exponential fall waveform, 199
exponential rise waveform, 199
external gated burst, 90, 187
external reference, 223
external modulation source
  AM, 71, 170
  FM, 77, 173
  FSK, 81, 177
external trigger, 100, 101, 195

F

filters, anti-aliasing, 295
firmware revision, 117
floating-point download, arbs, 200
flow control (RS-232), 121, 220
FM (frequency modulation)
  carrier waveform, 73
  deviation, 76, 174, 311
  example in BASIC, 279
  example in Visual Basic, 283
  example in Visual C++, 289
  frequency deviation, 76, 311
  modulating source, 77, 173
  modulating waveform, 75, 173
  overview, 172
  remote operation, 172
  tutorial description, 309
FM:DEV command, 174
FM:INT:FREQ command, 174
FM:INT:FUNC command, 173
FM:SOURce command, 173
FM:STATe command, 175
FORM:BORD command, 203
frame format (RS-232), 220, 221
FREQ:CENTer command, 182
FREQ:SPAN command, 182
FREQ:STARt command, 181
FREQ:STOP command, 181
FREQuency command, 155
frequency
  burst limitation, 51, 189
  duty cycle limits, 51, 155
  front-panel selection, 17
  function limits, 51, 145, 155
frequency deviation, 76, 174, 311

frequency modulation (FM)
  carrier waveform, 73
  deviation, 76, 174, 311
  example in BASIC, 279
  example in Visual Basic, 283
  example in Visual C++, 289
  frequency deviation, 76, 311
  modulating source, 77, 173
  modulating waveform, 75, 173
  overview, 172
  remote operation, 172
  tutorial description, 309
frequency span, sweep, 182
frequency sweep
  center frequency, 84, 182
  example in BASIC, 279
  example in Visual Basic, 283
  example in Visual C++, 290
  external trigger source, 101
  frequency span, 84, 182
  front-panel operation, 38
  linear vs. log, 85, 183, 312
  marker frequency, 86, 186
  overview, 179
  start frequency, 83, 181
  stop frequency, 83, 181
  sweep time, 85, 183
  Sync signal, 83
  trigger out signal, 88, 102
  trigger source, 87
frequency-shift keying
  see FSK
front panel
  connectors, 3
  creating arb waveforms, 103
  display overview, 4
  enable/disable, 115, 214
  graph mode, 23
  menu operation, 29
  menu quick reference, 31
  number entry, 5
  number format, 116
  overview, 3
FSK (frequency-shift keying)
  carrier waveform, 79
  front-panel operation, 36
  FSK rate, 37, 80
  "hop" frequency, 80, 178
  modulating source, 81, 177
  modulating waveform, 79
  overview, 176
  remote operation, 176
  tutorial description, 309
`FSK:FREQ` command, 178
`FSK:INT:RATE` command, 178
`FSK:SOURce` command, 177
`FSK:STATe` command, 178
function
  allowed with mod, 49, 153
  amplitude limits, 50, 156
  frequency limits, 50, 155
  pulse period limits, 167
`FUNC` command, 153
`FUNC USER` command, 205
`FUNC:RAMP:SYMM` command, 162
`FUNC:SQU:DCYCle` command, 161
`FUNC:USER` command, 204

G

gate polarity (burst), 194, 197
gated burst, 90, 187, 317
gaussian noise, 151, 153
GPIB interface (IEEE-488)
  address, 118
  connector, 6
  default address, 44
  front-panel configuration, 44
  interface selection, 119, 218
  setting address, 44
graph mode, 23
graphical display, 214
  brightness, 114
  bulb saver mode, 113
  contrast, 114
  displaying message, 115, 215
  graph mode, 23
  enable/disable, 115, 214
  number format, 116
  overview, 4
ground loops, 305

H

handle
  changing positions, 16
  removing, 27
handshake (RS-232), 45, 121, 220
  arb limits, 201, 219
harmonic distortion, 320
header, binary block, 202
heart (cardiac) waveform, 199
help system, 25
  language selection, 26
high impedance load, 33, 57
high level/low level, 53, 159
high Z load, 33, 57
"hop" frequency (FSK), 36, 80, 178

I

ID string (`*IDN?` command), 214
IEEE-488 interface (GPIB)
  address, 118
  connector, 6
  default address, 44
  front-panel configuration, 44
  interface selection, 119, 218
  setting address, 44
IEEE-488 block format, 202
IEEE-488 service request, 229
immediate trigger, 184, 192, 195
impedance, load, 33, 57, 305
imperfections, signal, 302
instrument calibration
  certificate, 15
  commands, 239
  error messages, 271
  message, 126
  read count, 125, 240
  securing, 239
  security code, 123
  setups, 239
  text message, 240
  unsecuring, 239
instrument
  dimensions, 322
  drivers, 277
  errors, 112, 213
  ID string (`*IDN?`), 214
  overview, 2
  reset, 33, 216
  self test, 114
  self-test, 216
  specifications, 319
  weight, 323
instrument exchange, 7
instrument state storage, 109, 209
  default names, 211
  deleting from memory, 211
  naming states, 43, 110
  power-down recall, 109, 212
  storing state, 43, 109, 209
integer download, arbs, 201
interface (bus) trigger, 100
interface configuration, 118
interface errors, 112, 213, 247
  arb waveform, 272
  calibration, 271
  "data out of range", 261
  execution, 252
  instrument, 268
  number allowed in queue, 213
  phase unlock, 224
  query errors, 267
  self-test errors, 269
  "settings conflict", 253
internal trigger, 99, 195
internal modulation source
  AM, 71, 170
  FM, 77, 173
  FSK, 81, 177
interpolation (arb points), 104
invert waveform polarity, 61, 164

L

language,
  SCPI overview, 241
  help system, 26
leakage errors, 298
learn string (`*LRN?` command), 216
linear interpolation, 104
linear sweep, 183
load termination, 33, 57, 305
local language (help system), 26
local operation (RS-232), 218
logarithmic sweep, 183
low level/high level, 53, 159

M

manual trigger, 99
marker frequency, 86, 186, 314
`MARKer:FREQ` command, 186
MAV (message available), 230
`MEM:NSTates?` command, 212
`MEM:STATe:DEL` command, 211
`MEM:NAME` command, 211
`MEM:REC:AUTO` command, 212
`MEM:VALid?` command, 212
menu quick reference, 31
menu operation, 29
message
  calibration, 126, 240
  front-panel display, 115, 215
message available (MAV), 230
Microsoft Visual Basic
  programming examples, 282
Microsoft Visual C++
  programming examples, 287
modem
  handshake (RS-232), 121, 220
modulation
  AM, 67, 169
  example in BASIC, 279
  example in Visual Basic, 283
  example in Visual C++, 289
  FM, 72, 172
  FSK, 78, 176
  overview, 172
  remote operation, 172
  tutorial description, 309
modulation depth (AM), 171, 310
Modulation In connector, 71, 77
modulation source
  AM, 71, 170
  FM, 77, 173
  FSK, 81, 177

N

naming
  arbitrary waveforms, 107, 203
  stored states, 43, 110, 211
narrowband FM, 311
n-cycle burst, 315
negative ramp waveform, 199
negative slope, 185, 193, 196
new line, 244
NI-488.2 libraries, 277
no parity (RS-232), 120
noise, 151, 153
number entry, 5
number keypad, 5
number of cycles, burst, 190
number of bits (RS-232), 120
number of errors allowed, 213
numeric entry, 5
nyquist sampling theorem, 297

O

odd parity, 120
offset
  amplitude limits, 54, 148, 158
  arb limits, 55, 148, 158
  front-panel selection, 20
  load limits, 54, 148, 158
operation complete, 217
Output connector
  enable/disable, 60, 162
  polarity, 61
output amplitude, 18, 146
  arb waveform limits, 53, 147
  dBm limits, 156
  high/low level, 53, 157
  load limit, 52, 146, 156
  offset limits, 52, 156
  range hold, 60
  termination (load), 52, 146, 156
  tutorial description, 304
  unit limit, 52
  units selection, 56
output frequency
  burst limits, 51, 189
  duty cycle limits, 51, 155
  front-panel selection, 17
  function limits, 51, 145, 155
output function
  allowed with mod, 49, 153
  amplitude limits, 50, 156
  frequency limits, 50, 155
  pulse period limits, 167
output impedance, 33, 57, 305
output overload, 162
output period
  front-panel selection, 17
output units, 56
  dBm limitation, 165
output waveform polarity, 61
`OUTPut`
  `:POLarity` command, 164
  `:SYNC` command, 164
  `:TRIG:SLOPe` command, 197
  `:TRIGger` command, 197
`OUTPut` command, 162
overheating, 28
overload, output, 60, 162
overview
  display, 4
  front panel, 3
  front-panel menus, 31
  number entry, 5
  product, 2
  programming, 142
  rear panel, 6

P

parameter types, 245
parity (RS-232), 45, 120
password, calibration, 123
peak deviation (FM), 76, 174
peak voltage, 307
peak-to-peak voltage, 307
percent mod (AM), 171, 310
period
  burst mode, 94
  front-panel selection, 17
  pulse waveform, 64, 166
phase (burst), 95, 191
phase errors, 303
phase lock, 223
  error on unlock, 224
  phase offset, 223
  rear-panel connections, 223
phase noise, 303, 320
phase quantization errors, 303
`PHASe` command, 223
`PHASe`
  `:REF` command, 224
  `:UNL:ERR:STATe` command, 224
point interpolation (arb), 104
polarity
  trigger, 185, 193, 196
  waveform, 61, 164
positive slope, 185, 193, 196
power connector, 6
power switch, 15
power-down recall, 109, 212
product
  dimensions, 322
  drivers, 277
  errors, 112, 213
  ID string (`*IDN?`), 214
  overview, 2
  reset, 33, 216
  self test, 114
  self-test, 216
  specifications, 319
  weight, 323
program examples
  BASIC for Windows, 278
  Visual Basic, 282
  Visual C++, 287
programming commands, 129
programming overview, 142

pulse
edge time, 66, 168
example in BASIC, 279
example in Visual Basic, 283
example in Visual C++, 290
front-panel configuration, 22
function limits, 167
parameter definitions, 166
period, 64, 166
pulse width, 22, 65, 167
tutorial description, 300
PULSe
:PERiod command, 166
:TRANsition command, 168
:WIDTh command, 167

Q

quantization errors, 303
questionable data register
bit definitions, 231
commands, 236
operation, 231
queue, error, 112, 213
quick reference, commands, 131
quick start, 13

R

rack mounting kits, 28
radians, 191
ramp symmetry, 59, 162
range hold, 60, 160, 304
rear panel
connectors, 6
overview, 6
recalling stored states, 210
reference (external), 223
registers (status), 225, 226
remote (bus) trigger, 100
remote errors, 112, 213, 247
arb waveform, 272
calibration, 271
"data out of range", 261
execution, 252
instrument, 268
number allowed in queue, 213
phase unlock, 224
query errors, 267
self-test errors, 269
"settings conflict", 253
remote interface
command reference, 129
command summary, 131
configuration, 118
selection, 119, 218
remote (bus) trigger, 196
reset, 33, 216
resistance, load, 33, 305
revision, firmware, 117
RMS voltage, 307
RS-232 (serial) interface
arb limits, 201, 219
baud rate selection, 45, 120
cable kits, 15, 222
cable pinout, 222
configuring interface, 219
connector, 6
data frame format, 220, 221
front-panel configuration, 45
handshake selection, 45, 121
interface selection, 119, 218
parity selection, 45, 120
return to local mode, 218
troubleshooting, 222
RTS/CTS (RS-232), 121, 220
rubber bumpers, removing, 27

S

sample programs
BASIC for Windows, 278
Visual Basic, 282
Visual C++, 287
saved states, 109, 209
default names, 211
deleting from memory, 211
naming states, 43, 110
power-down recall, 109, 212
storing state, 43, 109, 209
SCPI language
command reference, 129
command summary, 131
command terminators, 244
language overview, 241
parameter types, 245
status system, 225
version, 117, 216
SCPI status registers, 225
condition register, 225
enable register, 225
event register, 225
example in BASIC, 281
example in Visual Basic, 285
example in Visual C++, 292
questionable data register, 231
register diagram, 226
standard event register, 233
status byte register, 227
screen saver mode, 113
screen, front panel, 4
brightness, 114
bulb saver mode, 113
contrast, 114
displaying message, 115, 215
graph mode, 23
enable/disable, 115, 214
number format, 116
overview, 4

security, calibration, 123
self test, 114, 216
  error messages, 269
serial interface (RS-232)
  arb limits, 201, 219
  baud rate selection, 45, 120
  cable kits, 15, 222
  cable pinout, 222
  configuring interface, 219
  connector, 6
  data frame format, 220, 221
  front-panel configuration, 45
  handshake selection, 45, 121
  interface selection, 119, 218
  parity selection, 45, 120
  return to local mode, 218
  troubleshooting, 222
serial poll, 229
service request (SRQ), 229
shipping contents, 15
shock hazard, 6
SICL command libraries, 277
signal imperfections, 302
sinc (sin(x)/x) waveform, 199
sinewave spectral purity, 320
slope (trigger), 185, 193, 196
  burst, 96
  sweep, 87
  trigger input, 185
  trigger output, 185
softkey labels, 4
software (bus) trigger, 100, 195
software revision, 117
sound (beeper), 113
specifications, 319
spurs, 302
square wave
  duty cycle, 21, 58, 161
  tutorial description, 300
SRQ (service request), 229
standard event register
  bit definitions, 233
  commands, 237
  operation, 233
start frequency (sweep), 181
starting phase (burst), 95, 191
state storage, 109, 209
  default names, 211
  deleting from memory, 211
  naming states, 43, 110
  power-down recall, 109, 212
  storing state, 43, 109, 209
status byte register
  bit definitions, 227
  commands, 235
  operation, 227
status registers, 225
  condition register, 225
  enable register, 225
  event register, 225
  example in BASIC, 281
  example in Visual Basic, 285
  example in Visual C++, 292
  questionable data register, 231
  register diagram, 226
  standard event register, 233
  status byte register, 227
STATus
  :PRESet command, 238
  :QUES:COND? command, 236
  :ENABle command, 236
  :QUES? command, 236
stop frequency (sweep), 181
state storage, 109, 209
  default names, 211
  deleting from memory, 211
  storing state, 43, 109, 209
  naming states, 43, 110
  power-down recall, 109, 212
strings, error, 247
support, technical, 7
swapped byte order, 203
sweep
  center frequency, 84, 182
  example in BASIC, 279
  example in Visual Basic, 283
  example in Visual C++, 290
  external trigger source, 101
  frequency span, 84, 182
  front-panel operation, 38
  linear vs. log, 85, 183, 312
  marker frequency, 86, 186
  overview, 179
  start frequency, 83, 181
  stop frequency, 83, 181
  sweep time, 85, 183
  Sync signal, 83
  trigger out signal, 88, 102
  trigger source, 87
SWEep:SPACing command, 183
SWEep:STATe command, 183
SWEep:TIME command, 183
symmetry definition, 59, 162
Sync connector, 62, 186, 314
  enable/disable, 63, 164
  for all waveform functions, 62
syntax, SCPI commands, 131
SYSTem
  :BEEP:STATe command, 216
  :BEEP command, 216
  :ERRor? command, 213
  :INTerface command, 218
  :LOCal command, 218
  :RWLock command, 218
  :VERSion? command, 216
system error, 112, 213

T

technical specifications, 319
technical support, 7
temperature overload, 28
terminals (BNC)
  10 MHz In, 223
  10 MHz Out, 223
  Modulation In, 71, 77, 81
  Output, 60, 162
  Sync Output, 62, 186
  Trig In, 87, 101, 184
  Trig Out, 88, 102, 185
termination (load), 33, 57, 305
terminators, command, 244
test, 114, 216
text message
  calibration, 126, 240
  front-panel display, 115, 215
time delay (trigger), 193, 196
time (sweep), 183
tone, enable/disable, 216
transition time (pulse), 168
translated help system, 26
Trig In connector, 87, 101, 184
Trig Out connector, 88, 102, 185
trigger
  burst, 96
  bus (software), 192, 195
  delay, 193, 196
  external, 100, 184, 192, 195
  front-panel operation, 42
  immediate, 99, 184, 192, 195
  manual source, 99
  slope, 185, 193, 196
  software (bus), 100, 184
  sources, 98, 184, 192, 195
  sweep, 87
trigger out signal, 185, 194, 197
Trig In connector, 101
Trig Out connector, 102
`TRIGger`
  `:DELay` command, 193, 196
  `:SLOPe` command, 185, 193, 196
  `:SOURce` command, 184, 192, 195
`TRIGger` command, 196
troubleshooting (RS-232), 222
tutorial, 293
TXCO timebase, 223

U

`UNIT:ANGLe` command, 224
units
  amplitude, 56
  as part of command, 165
  burst phase, 192
  converting voltage, 19
  voltage, 146

V

version, SCPI, 117, 216
Visual Basic examples, 282
Visual C++ examples, 287
voltage autorange, 60, 160, 304
voltage output, 18, 146
  arb waveform limits, 53, 147
  dBm limits, 156
  high/low level, 53, 157
  load limit, 52, 146, 156
  offset limits, 52, 156
  range hold, 60
  termination (load), 52, 146, 156
  tutorial description, 304
  unit limit, 52
  units selection, 56
voltage offset
  amplitude limits, 54, 148, 158
  arb limits, 55, 148, 158
  front-panel selection, 20
  load limits, 54, 148, 158
voltage overload, 60
voltage units, 56, 146, 165
  converting, 19
`VOLTage`
  `:HIGH` command, 159
  `:LOW` command, 159
  `:OFFSet` command, 158
  `:RANGe:AUTO` command, 160
  `:UNIT` command, 165
`VOLTage` command, 156
vpp, 56, 165
vrms, 56, 165

W

warranty, *inside front cover*
waveform imperfections, 302
waveform invert, 61
waveform output
  connector, 60
  enable/disable, 60, 162
  polarity, 61
waveform tutorial, 293
weight (33250A), 322, 323
wideband FM, 311
width, pulse definition, 167

X

XON/XOFF (RS-232), 121, 220

Z

zero-phase reference, 224

© Copyright Agilent Technologies, Inc. 2000, 2003

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Agilent Technologies as governed by the United States and international copyright laws.

Manual Part Number

33250-90002, March 2003
(*order as 33250-90100 manual set*)

Edition

Edition 2, March 2003
Edition 1, April 2000

Printed in Malaysia

Agilent Technologies, Inc.
815 14th Street S.W.
Loveland, Colorado 80537 U.S.A.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products. For assistance, contact your nearest Agilent Technologies Sales and Service Office. Further information is available on the Agilent web site at *www.agilent.com/find/assist*.

Trademark Information

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation. All other brand and product names are trademarks or registered trademarks of their respective companies.

Certification

Agilent Technologies certifies that this product met its published specifications at the time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement will control.

Technologies Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Restricted Rights Legend

If software is for use in the performance of a U.S. Government prime contract or subcontract, Software is delivered and licensed as "Commercial computer software" as defined in DFAR 252.227-7014 (June 1995), or as a "commercial item" as defined in FAR 2.101(a) or as "Restricted computer software" as defined in FAR 52.227-19 (June 1987) or any equivalent agency regulation or contract clause. Use, duplication or disclosure of Software is subject to Agilent Technologies' standard commercial license terms, and non-DOD Departments and Agencies of the U.S. Government will receive no greater than Restricted Rights as defined in FAR 52.227-19(c)(1-2) (June 1987). U.S. Government users will receive no greater than Limited Rights as defined in FAR 52.227-14 (June 1987) or DFAR 252.227-7015 (b)(2) (November 1995), as applicable in any technical data.

Safety Notices

Do not defeat power cord safety ground feature. Plug in to a grounded outlet.

Do not use product in any manner not specified by the manufacturer.

Do not install substitute parts or perform any unauthorized modification to the product. Return the product to an Agilent Technologies Sales and Service Office for service and repair to ensure that safety features are maintained.

WARNING

A WARNING notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

CAUTION

A CAUTION notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

Symbols


Earth ground


Chassis ground


Risk of electric shock

WARNING

Main power disconnect: Unplug product from wall outlet and remove power cord before servicing. Only qualified, service-trained personnel should remove the cover from the instrument.

For continued protection against fire, replace the line fuse only with a fuse of the specified type and rating.

Agilent Technologies

DECLARATION OF CONFORMITY

According to ISO/IEC Guide 22 and CEN/CENELEC EN 45014

CE

Manufacturer's Name: Agilent Technologies, Inc.

Manufacturer's Address: 815 14th Street SW
Loveland, Colorado 80537
U.S.A.

Declares, that the product

Product Name: 80 MHz Function / Arbitrary Waveform Generator

Model Number: 33250A

Product Options: This declaration covers all options of the above product.

Conforms with the following European Directives:

The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (including 93/68/EEC) and carries the CE Marking accordingly.

Conforms with the following product standards:

EMC	Standard	Limit
	IEC 61326-1:1997+A1:1998 / EN 61326-1:1997+A1:1998 [1]	
	CISPR 11:1990 / EN 55011:1991	*Group 1 Class A*
	IEC 61000-4-2:1995+A1:1998 / EN 61000-4-2:1995	*4kV CD, 8kV AD*
	IEC 61000-4-3:1995 / EN 61000-4-3:1995	*3 V/m, 80-1000 MHz*
	IEC 61000-4-4:1995 / EN 61000-4-4:1995	*0.5kV signal lines, 1kV power lines*
	IEC 61000-4-5:1995 / EN 61000-4-5:1995	*0.5 kV line-line, 1 kV line-ground*
	IEC 61000-4-6:1996 / EN 61000-4-6:1996	*3V, 0.15-80 MHz I cycle, 100%*
	IEC 61000-4-11:1994 / EN 61000-4-11:1994	*Dips: 30% 10ms; 60% 100ms Interrupt > 95%@5000ms*
	Canada: ICES-001:1998 [2]	
	Australia/New Zealand: AS/NZS 2064.1	

[1] *The product was tested in a typical configuration with Agilent Technologies test systems.*

[2] *This ISM device complies with Canadian ICES-001-1998.*
Cet appareil ISM est conforme à la norme NMB-001-1998 du Canada.

Safety *IEC 61010-1:1990+A1:1992+A2:1995 / EN 61010-1:1993+A2:1995*
Canada: CSA C22.2 No. 1010.1:1992
UL 3111-1: 1994

March 12, 2001

Date

Ray Corson
Product Regulations Program Manager

For further information, please contact your local Agilent Technologies sales office, agent or distributor.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP