

# **Reference Manual**



## **AFG3000 Series Arbitrary/Function Generators 071-1639-00**

This document supports firmware version 1.0.0 and above.

**[www.tektronix.com](http://www.tektronix.com)**

Copyright © Tektronix, Inc. All rights reserved.

Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supercedes that in all previously published material. Specifications and price change privileges reserved.

Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

## WARRANTY 16

Tektronix warrants that the product will be free from defects in materials and workmanship for a period of three (3) years from the date of original purchase from an authorized Tektronix distributor. If the product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product. Batteries are excluded from this warranty. Parts, modules and replacement products used by Tektronix for warranty work may be new or reconditioned to like new performance. All replaced parts, modules and products become the property of Tektronix.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, shipping charges prepaid, and with a copy of customer proof of purchase. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; c) to repair any damage or malfunction caused by the use of non-Tektronix supplies; or d) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

**THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THE PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.**



# Table of Contents

<b>General Safety Summary</b> .....	<b>ix</b>
<b>Preface</b> .....	<b>xi</b>
Documentation .....	xi
Contacting Tektronix .....	xii

## Operating Basics

<b>Operating Basics</b> .....	<b>1-1</b>
Front Panel Controls .....	1-2
Display Area and Screen Interface .....	1-11
Waveform Parameters and Numeric Input .....	1-14
Rear Panel .....	1-19

## Reference

<b>Reference</b> .....	<b>2-1</b>
Menu System .....	2-1
Menu Structure .....	2-2
File Operations .....	2-28

## Syntax and Commands

<b>Syntax and Commands</b> .....	<b>3-1</b>
Command Syntax .....	3-1
Command Groups .....	3-9
Command Descriptions .....	3-15
ABORt (No Query Form) .....	3-15
AFGControl:CSCopy (No Query Form) .....	3-16
*CAL? .....	3-16
CALibration[:ALL] .....	3-17
*CLS (No Query Form) .....	3-17
DIAGnostic[:ALL] .....	3-18
DISPlay:CONTrast .....	3-19
DISPlay:SAVer[:STATe] .....	3-19
DISPlay[:WINDow]:TEXT[:DATA] .....	3-20
DISPlay[:WINDow]:TEXT:CLEAr (No Query Form) .....	3-20
*ESE .....	3-21
*ESR? .....	3-22
*IDN? .....	3-22
MEMory:STATe:VALId? .....	3-23
MEMory:STATe:DELeTe (No Query Form) .....	3-23
MEMory:STATe:LOCK .....	3-24
MEMory:STATe:RECall:AUTo .....	3-25
MMEMory:CATalog? .....	3-26
MMEMory:CDIRectory .....	3-27
MMEMory:DELeTe (No Query Form) .....	3-27
MMEMory:LOAD:STATe (No Query Form) .....	3-28

MMEMory:LOAD:TRACe (No Query Form) . . . . .	3-28
MMEMory:LOCK[:STATe] . . . . .	3-29
MMEMory:MDIRectory (No Query Form) . . . . .	3-29
MMEMory:STORE:STATe (No Query Form) . . . . .	3-30
MMEMory:STORE:TRACe (No Query Form) . . . . .	3-30
*OPC . . . . .	3-31
*OPT? . . . . .	3-31
OUTPut[1 2]:IMPedance. . . . .	3-32
OUTPut[1 2]:POLarity . . . . .	3-33
OUTPut[1 2]:STATe] . . . . .	3-33
OUTPut:TRIGger:MODE. . . . .	3-34
*PSC . . . . .	3-35
*RCL (No Query Form) . . . . .	3-35
*RST (No Query Form) . . . . .	3-36
*SAV (No Query Form) . . . . .	3-36
[SOURce[1 2]]:AM[:DEPTh] . . . . .	3-37
[SOURce[1 2]]:AM:INTErnal:FREQuency. . . . .	3-38
[SOURce[1 2]]:AM:INTErnal:FUNCTioN . . . . .	3-39
[SOURce[1 2]]:AM:INTErnal:FUNCTioN:EFILe . . . . .	3-40
[SOURce[1 2]]:AM:SOURce . . . . .	3-40
[SOURce[1 2]]:AM:STATe . . . . .	3-41
[SOURce[1 2]]:BURSt:MODE. . . . .	3-41
[SOURce[1 2]]:BURSt:NCYCles . . . . .	3-42
[SOURce[1 2]]:BURSt[:STATe] . . . . .	3-43
[SOURce[1 2]]:BURSt:TDELay. . . . .	3-44
[SOURce[1 2]]:COMBine:FEED . . . . .	3-45
[SOURce[1 2]]:FM[:DEViation]. . . . .	3-46
[SOURce[1 2]]:FM:INTErnal:FREQuency . . . . .	3-46
[SOURce[1 2]]:FM:INTErnal:FUNCTioN . . . . .	3-47
[SOURce[1 2]]:FM:INTErnal:FUNCTioN:EFILe. . . . .	3-48
[SOURce[1 2]]:FM:SOURce . . . . .	3-48
[SOURce[1 2]]:FM:STATe . . . . .	3-49
[SOURce[1 2]]:FREQuency:CENTer . . . . .	3-49
[SOURce[1 2]]:FREQuency:CONCurreNt[:STATe] . . . . .	3-50
[SOURce[1 2]]:FREQuency[:CW]:FIXed] . . . . .	3-51
[SOURce[1 2]]:FREQuency:MODE. . . . .	3-52
[SOURce[1 2]]:FREQuency:SPAN. . . . .	3-53
[SOURce[1 2]]:FREQuency:STARt . . . . .	3-54
[SOURce[1 2]]:FREQuency:STOP. . . . .	3-55
[SOURce[1 2]]:FSKey[:FREQuency] . . . . .	3-56
[SOURce[1 2]]:FSKey:INTErnal:RATE . . . . .	3-56
[SOURce[1 2]]:FSKey:SOURce . . . . .	3-57
[SOURce[1 2]]:FSKey:STATe. . . . .	3-57
[SOURce[1 2]]:FUNCTioN:EFILe . . . . .	3-58
[SOURce[1 2]]:FUNCTioN:RAMP:SYMMetry. . . . .	3-58
[SOURce[1 2]]:FUNCTioN[:SHAPE]. . . . .	3-59
[SOURce[1 2]]:PHASe[:ADJust] . . . . .	3-60
[SOURce[1 2]]:PHASe:INITiate (No Query Form) . . . . .	3-61
[SOURce[1 2]]:PM[:DEViation]. . . . .	3-61
[SOURce[1 2]]:PM:INTErnal:FREQuency . . . . .	3-62
[SOURce[1 2]]:PM:INTErnal:FUNCTioN . . . . .	3-63

[SOURce[1 2]]:PM:INTernal:FUNCTion:EFILe . . . . .	3-64
[SOURce[1 2]]:PM:SOURce . . . . .	3-64
[SOURce[1 2]]:PM:STATe . . . . .	3-65
SOURce<3 4>:POWer[:LEVe][:IMMediate][:AMPLitude] . . . . .	3-66
[SOURce[1 2]]:PULSe:DCYCLe . . . . .	3-67
[SOURce[1 2]]:PULSe:DELay . . . . .	3-68
[SOURce[1 2]]:PULSe:HOLD . . . . .	3-68
[SOURce[1 2]]:PULSe:PERiod . . . . .	3-69
[SOURce[1 2]]:PULSe:TRANsition[:LEADing] . . . . .	3-69
[SOURce[1 2]]:PULSe:TRANsition:TRAILing . . . . .	3-70
[SOURce[1 2]]:PULSe:WIDTh . . . . .	3-71
[SOURce[1 2]]:PWM:INTernal:FREQuency . . . . .	3-72
[SOURce[1 2]]:PWM:INTernal:FUNCTion . . . . .	3-73
[SOURce[1 2]]:PWM:INTernal:FUNCTion:EFILe . . . . .	3-74
[SOURce[1 2]]:PWM:SOURce . . . . .	3-74
[SOURce[1 2]]:PWM:STATe . . . . .	3-75
[SOURce[1 2]]:PWM[:DEViation]:DCYCLe . . . . .	3-76
[SOURce]:ROSCillator:SOURce . . . . .	3-77
[SOURce[1 2]]:SWEep:HTIME . . . . .	3-77
[SOURce[1 2]]:SWEep:MODE . . . . .	3-78
[SOURce[1 2]]:SWEep:RTIME . . . . .	3-79
[SOURce[1 2]]:SWEep:SPACing . . . . .	3-79
[SOURce[1 2]]:SWEep:TIME . . . . .	3-80
[SOURce[1 2]]:VOLTage:CONCurent[:STATe] . . . . .	3-81
[SOURce[1 2]]:VOLTage[:LEVe][:IMMediate]:HIGH . . . . .	3-82
[SOURce[1 2]]:VOLTage[:LEVe][:IMMediate]:LOW . . . . .	3-83
[SOURce[1 2]]:VOLTage[:LEVe][:IMMediate]:OFFSet . . . . .	3-84
[SOURce[1 2]]:VOLTage[:LEVe][:IMMediate][:AMPLitude] . . . . .	3-85
[SOURce[1 2]]:VOLTage:LIMit:HIGH . . . . .	3-86
[SOURce[1 2]]:VOLTage:LIMit:LOW . . . . .	3-87
[SOURce[1 2]]:VOLTage:UNIT . . . . .	3-88
*SRE . . . . .	3-89
STATus:OPERation:CONDition? . . . . .	3-90
STATus:OPERation:ENABLe . . . . .	3-90
STATus:OPERation[:EVENT]? . . . . .	3-91
STATus:PRESet (No Query Form) . . . . .	3-91
STATus:QUEStionable:CONDition? . . . . .	3-92
STATus:QUEStionable:ENABLe . . . . .	3-92
STATus:QUEStionable[:EVENT]? . . . . .	3-93
*STB? . . . . .	3-93
SYSTem:BEEPer[:IMMediate] (No Query Form) . . . . .	3-94
SYSTem:BEEPer:STATe . . . . .	3-94
SYSTem:ERRor[:NEXT]? . . . . .	3-95
SYSTem:KCLick[:STATe] . . . . .	3-95
SYSTem:KLOCK[:STATe] . . . . .	3-96
SYSTem:SECurity:IMMediate (No Query Form) . . . . .	3-96
SYSTem:ULANguage . . . . .	3-97
SYSTem:VERSion? . . . . .	3-97
TRACe DATA:CATalog? . . . . .	3-98
TRACe DATA:COPY (No Query Form) . . . . .	3-98
TRACe DATA[:DATA] . . . . .	3-99

TRACe DATA[:DATA]:LINE (No Query Form) . . . . .	3-100
TRACe DATA[:DATA]:VALue . . . . .	3-101
TRACe DATA:DEFine (No Query Form). . . . .	3-102
TRACe DATA:DELeTe[:NAME] (No Query Form) . . . . .	3-103
TRACe DATA:LOCK[:STATe] . . . . .	3-103
TRACe DATA:POINts. . . . .	3-104
*TRG (No Query Form) . . . . .	3-104
TRIGger[:SEQuence]:SLOPe . . . . .	3-105
TRIGger[:SEQuence]:SOURce . . . . .	3-105
TRIGger[:SEQuence]:TIMer . . . . .	3-106
TRIGger[:SEQuence][:IMMediate] (No Query Form). . . . .	3-106
*TST? . . . . .	3-107
*WAI (No Query Form) . . . . .	3-107

## Status and Events

<b>Status and Events . . . . .</b>	<b>4-1</b>
Status Reporting Structure . . . . .	4-1
Registers . . . . .	4-4
Queues . . . . .	4-10
Messages and Codes . . . . .	4-11

## Programming Examples

<b>Programming Examples . . . . .</b>	<b>5-1</b>
---------------------------------------	------------

## Appendices

<b>Accessories and Options . . . . .</b>	<b>A-1</b>
Options . . . . .	A-1
Accessories . . . . .	A-2
<b>General Care and Cleaning . . . . .</b>	<b>B-1</b>
General Care . . . . .	B-1
Cleaning . . . . .	B-1
<b>SCPI Conformance Information . . . . .</b>	<b>C-1</b>
<b>Default Setup . . . . .</b>	<b>D-1</b>

## Index

<b>Index . . . . .</b>	<b>Index-1</b>
------------------------	----------------



# List of Figures

Figure 1-1: Dual-channel model .....	1-1
Figure 1-2: Front panel controls .....	1-2
Figure 1-3: Waveform parameter and graph display .....	1-4
Figure 1-4: Graph comparison .....	1-4
Figure 1-5: Waveform parameter comparison (CH1 selected) .....	1-5
Figure 1-6: Waveform parameter comparison (CH2 selected) .....	1-5
Figure 1-7: Run Mode menu (Continuous) .....	1-8
Figure 1-8: Screen interface .....	1-11
Figure 1-9: Level meter .....	1-12
Figure 1-10: Graphical representation of button status .....	1-13
Figure 1-11: Default display .....	1-14
Figure 1-12: Frequency/Period/Phase Menu .....	1-14
Figure 1-13: Screen display with Frequency active (1) .....	1-15
Figure 1-14: Screen display with Frequency active (2) .....	1-15
Figure 1-15: Screen display with Frequency active (3) .....	1-15
Figure 1-16: Screen display with Frequency active (4) .....	1-16
Figure 1-17: Amplitude/Level Menu .....	1-16
Figure 1-18: PWM sample screen .....	1-17
Figure 1-19: PWM parameter menu (Freq is selected) .....	1-17
Figure 1-20: PWM parameter menu (Period is selected) .....	1-18
Figure 1-21: Rear panel connectors .....	1-19
 Figure 2-1: Sweep type .....	 2-9
Figure 2-2: Frequency sweep .....	2-10
Figure 2-3: Number of Points .....	2-16
Figure 2-4: Using the Cursor (Line edit) .....	2-18
Figure 2-5: Line Edit example .....	2-19
Figure 2-6: Operation – Edit by Data Point .....	2-19
Figure 2-7: Cut example .....	2-20
Figure 2-8: Paste at Beginning .....	2-21
Figure 2-9: Paste at End .....	2-22
Figure 2-10: Browse waveform files (USB memory) .....	2-28
Figure 2-11: Write As - Character entry box .....	2-30
Figure 2-12: Recall menu (Internal) .....	2-31
Figure 2-13: Save As - Character entry box .....	2-32
 Figure 3-1: Command message elements .....	 3-3
Figure 3-2: Example of SCPI subsystem hierarchy tree .....	3-4
Figure 3-3: Example of abbreviating a command .....	3-6
Figure 3-4: Example of chaining commands and queries .....	3-7
Figure 3-5: Example of omitting root and lower level nodes .....	3-7
 Figure 4-1: Error and event handling process .....	 4-2
Figure 4-2: The Status Byte Register (SBR) .....	4-5
Figure 4-3: The Standard Event Status Register (SESR) .....	4-6
Figure 4-4: Operation Condition Register (OCR) .....	4-7

**Figure 4-5: Questionable Condition Register (QCR) ..... 4-8**  
**Figure 4-6: Event Status Enable Register (ESER) ..... 4-9**  
**Figure 4-7: Service Request Enable Register (SRER) ..... 4-9**  
**Figure 4-8: Operation Enable Register (OENR) ..... 4-9**  
**Figure 4-9: Questionable Enable Register (QENR) ..... 4-10**  
  
**Figure A-1: Fuse adapter ..... A-3**

# List of Tables

Table 2-1: Sine/Square Menu .....	2-3
Table 2-2: Ramp Menu .....	2-4
Table 2-3: Pulse Menu .....	2-4
Table 2-4: Arb Menu .....	2-5
Table 2-5: More Menu .....	2-6
Table 2-6: Modulation Parameter Menu .....	2-7
Table 2-7: Sweep Parameter Menu .....	2-9
Table 2-8: Burst Parameter Menu .....	2-11
Table 2-9: Output Menu .....	2-12
Table 2-10: Save/Recall Menu .....	2-13
Table 2-11: Edit Menu .....	2-14
Table 2-12: Operations submenu .....	2-17
Table 2-13: Utility Menu .....	2-23
Table 3-1: BNF notation .....	3-1
Table 3-2: Command message elements .....	3-2
Table 3-3: Query response examples .....	3-3
Table 3-4: Parameter types used in syntax descriptions .....	3-5
Table 3-5: Calibration and Diagnostic commands .....	3-9
Table 3-6: Display commands .....	3-9
Table 3-7: Memory commands .....	3-9
Table 3-8: Mass Memory commands .....	3-10
Table 3-9: Output commands .....	3-10
Table 3-10: Source commands .....	3-10
Table 3-11: Status commands .....	3-12
Table 3-12: System commands .....	3-13
Table 3-13: Synchronization commands .....	3-13
Table 3-14: Trace commands .....	3-14
Table 3-15: Trigger commands .....	3-14
Table 3-16: AFG Control command .....	3-14
Table 4-1: SBR bit functions .....	4-5
Table 4-2: SESR bit functions .....	4-6
Table 4-3: OCR bit functions .....	4-7
Table 4-4: QCR bit functions .....	4-8
Table 4-5: Definition of event codes .....	4-11
Table 4-6: Command error messages .....	4-11
Table 4-7: Execution error messages .....	4-13
Table 4-8: Device-specific error messages .....	4-15
Table 4-9: Query errors .....	4-15
Table 4-10: Power-on events .....	4-15
Table 4-11: User request events .....	4-16
Table 4-12: Request control events .....	4-16
Table 4-13: Operation complete events .....	4-16
Table 4-14: Device errors .....	4-17

**Table A-1: Options ..... A-1**  
**Table A-2: Standard accessories ..... A-2**  
**Table A-3: Optional accessories ..... A-2**  
**Table C-1: SCPI conformance information ..... C-1**  
**Table D-1: Default settings ..... D-1**

# General Safety Summary

Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it. To avoid potential hazards, use this product only as specified.

Only qualified personnel should perform service procedures.

## To avoid Fire or Personal Injury

**Use Proper Power Cord.** Use only the power cord specified for this product and certified for the country of use.

**Ground the Product.** This product is grounded through the grounding conductor of the power cord. To avoid electric shock, the grounding conductor must be connected to earth ground. Before making connections to the input or output terminals of the product, ensure that the product is properly grounded.

**Observe All Terminal Ratings.** To avoid fire or shock hazard, observe all ratings and markings on the product. Consult the product manual for further rating information before making connections to the product. Do not apply a potential to any terminal, including the common terminal, that exceeds the maximum rating of that terminal.

**Power Off.** The power cord provides Mains disconnect.

**Do Not Operate Without Covers.** Do not operate this product with covers or panels removed.

**Do Not Operate With Suspected Failures.** If you suspect there is damage to this product, have it inspected by qualified service personnel.

**Do Not Operate in Wet/Damp Conditions.**

**Do Not Operate in an Explosive Atmosphere.**

**Keep Product Surfaces Clean and Dry.**

**Provide Proper Ventilation.** Refer to the manual's installation instructions for details on installing the product so it has proper ventilation.

## **Symbols and Terms**

**Terms in this Manual.** These terms may appear in this manual:



---

***WARNING.*** *Warning statements identify conditions or practices that could result in injury or loss of life.*

---



---

***CAUTION.*** *Caution statements identify conditions or practices that could result in damage to this product or other property.*

---

**Terms on the Product.** These terms may appear on the product:

DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

CAUTION indicates a hazard to property including the product.

# Preface

This manual provides operating information for the AFG3000 Series Arbitrary/Function Generators. The manual consists of the following sections:

- *Operating Basics* covers operating principles of the instrument. The operating procedures help you understand how your generator operates.
- *Reference* provides in-depth descriptions of the instrument menu structures and menu button functions.
- *Syntax and Commands* defines the command syntax and processing conventions, describes command notation.
- *Status and Events* explains the status information and event messages reported by the instrument.
- *Programming Examples* contains remote interface application programs to help you develop programs for your application.
- *Appendix A: Accessories & Options* describes the standard and optional accessories as well as the instrument options.
- *Appendix B: General Care and Cleaning* describes how to take care of the instrument.
- *Appendix C: SCPI Conformance Information* contains a list of commands and SCPI information.
- *Appendix D: Default Setup* contains a list of the menus and controls that are recalled when you push the front-panel **Default** button.

## Documentation

In addition to this *AFG3000 Series Arbitrary/Function Generators Reference Manual*, the following documentation is available for this instrument:

**AFG3000 Series Quick Start User Manual.** The quick start user manual provides information on installation, general features, operating the instrument, and user interface. It also provides electrical, environmental, and physical specifications for the instrument.

**Built in Help System.** The built-in help system that is integrated with the User Interface application that ships with this instrument.

**AFG3000 Series Service Manual.** A service manual is available as an optional accessory. The service manual includes procedures to service the instrument to the module level. The manual also includes performance verification procedures so that you can verify performance to the advertised specifications.

# Contacting Tektronix

Phone	1-800-833-9200*
Address	Tektronix, Inc. Department or name (if known) 14200 SW Karl Braun Drive P.O. Box 500 Beaverton, OR 97077 USA
Web site	<a href="http://www.tektronix.com">www.tektronix.com</a>
Sales support	1-800-833-9200, select option 1*
Service support	1-800-833-9200, select option 2*
Technical support	1-800-833-9200, select option 3*  6:00 a.m. - 5:00 p.m. Pacific time

---

\* This phone number is toll free in North America. After office hours, please leave a voice mail message.  
Outside North America, contact a Tektronix sales office or distributor; see the Tektronix web site for a list of offices.





# Operating Basics



# Operating Basics

The AFG3000 Series Arbitrary/Function Generators front panel is divided into easy to use functional areas. This section provides you with a quick overview of the controls. Figure 1-1 shows the front panel of dual-channel model.

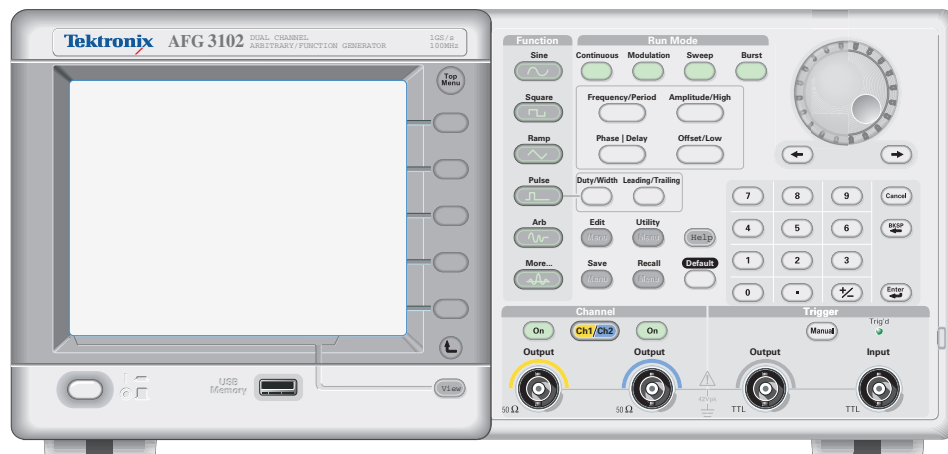
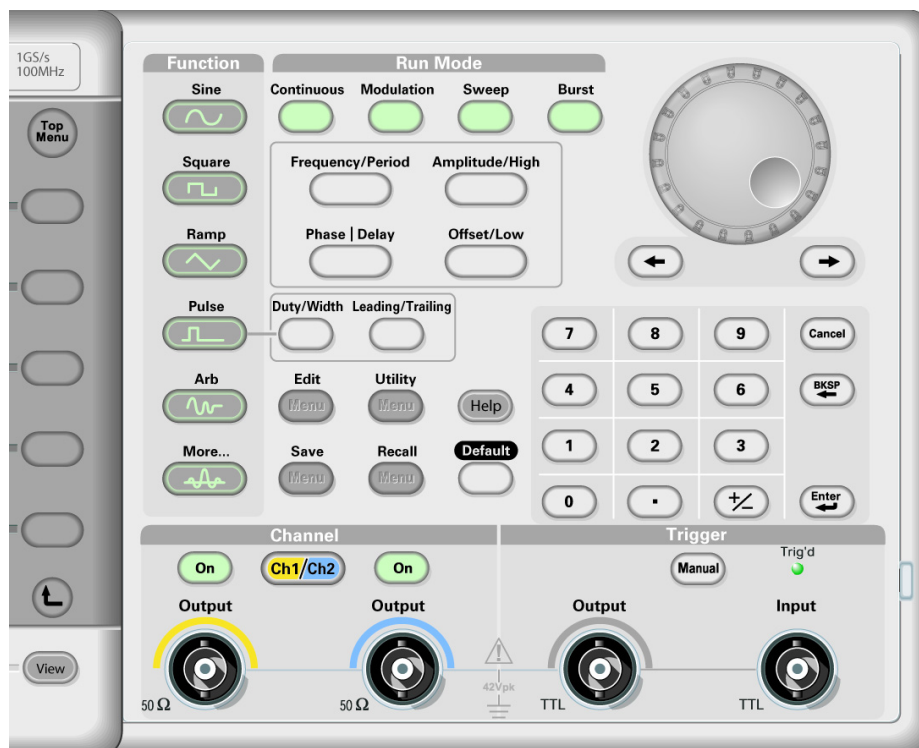


Figure 1-1: Dual-channel model

## Front Panel Controls

This section introduces you to the front panel controls of the instrument and provides a brief overview on how to use the front panel key controls.



**Figure 1-2: Front panel controls**


The AFG3000 Series Arbitrary/Function Generators front-panel key controls are divided into the following categories:


- Action buttons
- Menu buttons
- State buttons
- Function buttons
- Shortcut buttons


**Action Buttons**

The Power (not shown in Figure 1-2), Upper Menu, Top Menu, View, and Manual Trigger buttons are called action buttons. When you push these action buttons, it will cause an action.

**Power button.** Pushing the power button once turns the instrument on. Pushing the power button when the instrument is on will turn off the instrument.

**Upper Menu button.** Pushing the **Upper Menu**  button returns the currently displayed bezel menu to the upper level of the menu tree.

**Top Menu button.** The **Top Menu**  button is used to return to the top level of the menu tree from the current menu location.

**View button.** The **View**  button is used to toggle the screen view format. The arbitrary/function generator provides the following three screen view formats:

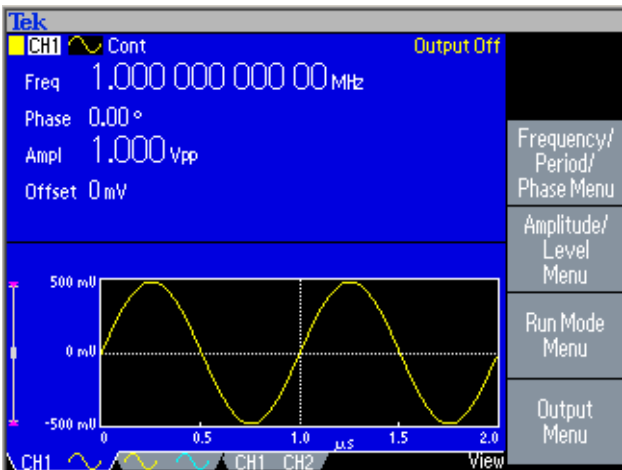
- View format 1: Waveform parameter and graph display, see Figure 1-3
- View format 2: Graph comparison, see Figure 1-4
- View format 3: Waveform parameter comparison, see Figure 1-5 and Figure 1-6

When the instrument is in one of the three display format, pushing the **View** button will toggle through the three views in a repeating cycle in the order described above.

When the instrument is in the Edit Menu, pushing the **View** button will toggle between Edit text and graphical views. This is the only function of the single-channel model view button.

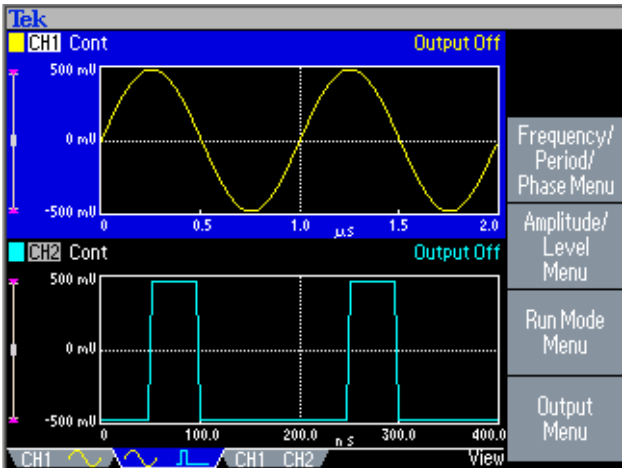
When the instrument is in the Save/Recall, Utility, Output, or Help menu, pushing the **View** button will have no effect.

**View Format 1.** Figure 1-3 is a single channel parameter and graph setup display. In this view, Channel 1 is selected with the Channel Select button. When Channel 2 is selected, the parameters and graph for Channel 2 will be displayed in this view. You can easily toggle between the information for Channel 1 and Channel 2 with this view.



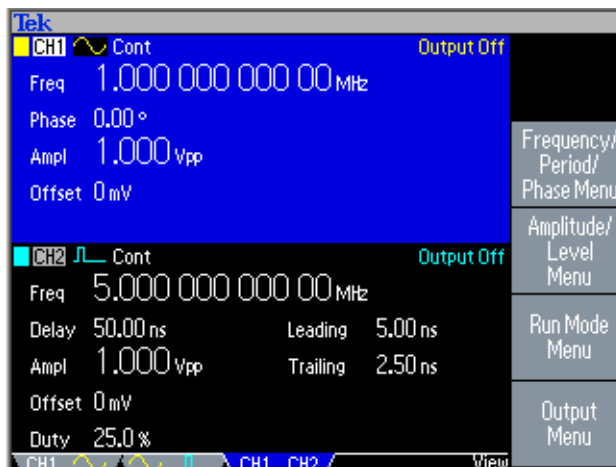
**Figure 1-3: Waveform parameter and graph display**

**View Format 2.** Figure 1-4 is a graph compare view. In this view, the Channel Select button selects the active graph. The selected channel is highlighted.



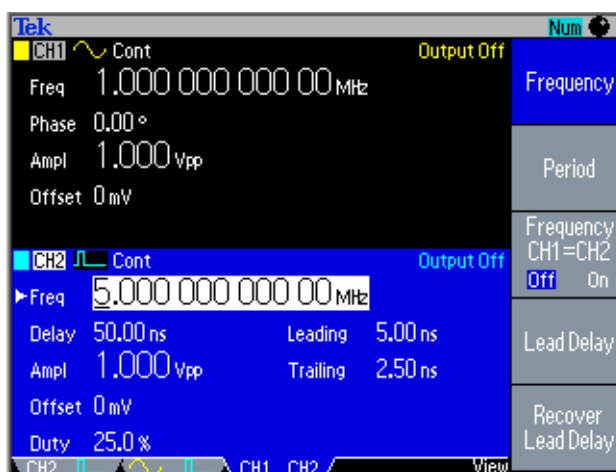
**Figure 1-4: Graph comparison**

**View Format 3.** Figure 1-5 is a channel compare view. In this view, Channel 1 is selected with the Channel Select button.



**Figure 1-5: Waveform parameter comparison (CH1 selected)**

In Figure 1-6, Channel 2 is selected with the Channel Select button.



**Figure 1-6: Waveform parameter comparison (CH2 selected)**

**Manual Trigger button.** Pushing the **Manual Trigger** button generates a trigger event manually. The Manual Trigger is one of the trigger sources for sweep or burst waveform. If you push the Manual Trigger button, the arbitrary/function generator will initiate one sweep in the Trigger Sweep mode, or output a waveform that has the specified number of cycles in the Burst mode. If the instrument is in Gated mode, it outputs a waveform while the Manual Trigger button is depressed.

The Trig'd LED lights whenever the instrument responds to a trigger.

**Other Action buttons.** The Enter button and the following related buttons are also classified as Action buttons.

- **Enter button.** The **Enter** button causes a numeric input to be updated.
- **+/- button.** This button is only active when you are setting a value. The +/- button changes the sign of the currently selected parameter from positive to negative. If the value is already negative, this button makes it positive.
- **Cancel button.** The **Cancel** button is active when you are setting a value. Pushing the **Cancel** button closes the Units menu and restores the previous value for the selected setting.  
  
When the front-panel controls are locked, you can use the **Cancel** button to unlock the front-panel controls. See page 1-10.
- **Backspace (BKSP) button.** The **BKSP** button is only active when you are setting a value in the Numeric Input Area. Pushing the **BKSP** button deletes the currently selected digit.

**Menu Buttons**      The Edit, Utility, Save, Recall, Help, and Default buttons are called Menu buttons.

**Edit button.** The **Edit** button opens Edit menu. See Table 2-11 on page 2-14 for the Edit menu map.

**Utility button.** The **Utility** button opens Utility menu. See Table 2-13 on page 2-23 for the Utility menu map.

**Save button.** The **Save** button opens the Save menu. See Table 2-10 on page 2-13 for the Save menu map.

**Recall button.** The **Recall** button opens the Recall menu. See Table 2-10 on page 2-13 for the Recall menu map.

**Help button.** The **Help** button opens the built-in Help.


**Default button.** The **Default** button restores the instrument settings to the default values. When you push this button, you will be prompted on the display with a pop-up window message requesting you to confirm that you want to restore the defaults. See *Default Setup* on page D-1 for the settings when you push the **Default** button.

**Run Mode buttons.** When one of four Run Mode buttons is selected, the menu for configuring the run mode is displayed on the screen. See *State buttons* on page 1-7.



**State Buttons**

The Channel Select, CH1/CH2 Output and Run Mode buttons are called State buttons.

**Channel Select button.** The Channel Select  button directly controls the display, toggling between the two channels. This button is used to select the channel that you are currently interacting with. Only one channel can be selected at a time.

When you push the Channel Select button on the Edit, Utility, Save, Recall, or Help screen, the arbitrary/function generator returns to previous display. The displayed channels do not toggle if you push the Channel Select button on those screen menus. After restoring the main display area, pushing the Channel Select button toggles between CH1 and CH2.

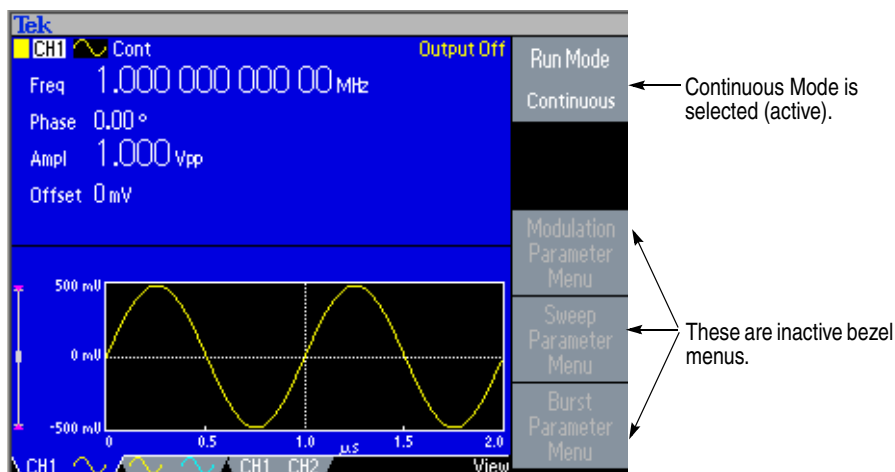
**CH1/CH2 Output On button.** The arbitrary/function generator allow you to turn on and off the signal output for CH1 and CH2 independently. You can configure the signal with the outputs off, to minimize the chance of sending a problematic signal to your device. You can select either one or both of these buttons. Each button is lit with an LED when in the On state.

**Run Mode buttons.** Only one Run Mode menu can be selected for each channel. The Run Mode buttons are unique because they are both State and Menu buttons. When one of four buttons is selected, that run mode is activated and the menu for configuring that run mode is displayed on the screen. The selected button is lit with an LED.

- Continuous
- Modulation
- Sweep
- Burst

If your instrument is a dual-channel model, Run Mode can be set independently for each channel.

Select the **Run Mode Menu** bezel button from the default screen (see page 1-14) to display the Run Mode menus. The Continuous mode is selected in Figure 1-7. If you select Modulation, Sweep, or Burst as the Run Mode, the corresponding bezel menu is highlighted.



**Figure 1-7: Run Mode menu (Continuous)**

## Function Buttons

Only one of the Function buttons can be selected for each channel at a time. The selected button will be lit with an LED.

**Sine button.** Pushing the **Sine** button selects the sine waveform, causing the Sine button LED to turn on. See Table 2-1 on page 2-3 for the menu map.

**Square button.** Pushing the **Square** button selects the square waveform, causing the Square button LED to turn on. See Table 2-1 on page 2-3 for the menu map.

**Ramp button.** Pushing the **Ramp** button selects the ramp waveform, causing the Ramp button LED to turn on. See Table 2-2 on page 2-4 for the menu map.

**Pulse button.** Pushing the **Pulse** button selects the pulse waveform, causing the Pulse button LED to turn on. See Table 2-3 on page 2-4 for the menu map.

**Arb button.** Pushing the **Arb** button causes the Arb waveform menu to be displayed on the screen, and causes the Arb LED to turn on. See Table 2-4 on page 2-5 for the menu map.

**More... button.** Pushing the **More...** button causes the More waveform menu to be displayed on the screen, and causes the More... LED to turn on. See Table 2-5 on page 2-6 for the menu map.

## Shortcut Buttons

The following six buttons are called Shortcut buttons and are provided as shortcuts for experienced users. You can push this button while viewing any of the display types. If you are not in view format 1, 2, or 3 (see page 1-3), pushing the shortcut button will take you to the last view you used and highlight the selected setting.

**Frequency/Period button.** This button selects the setting that was last used (**Frequency** or **Period**). If Frequency was selected, you can change the shortcut by selecting Period with the bezel menu. The next time you push the Frequency/Period button, Period will be selected. This shortcut button allows you to select the setting and enter their numeric value using the front panel, without requiring any bezel menu selection.

**Amplitude/High button.** This button selects the setting that was last used (**Amplitude** or **High Level**).

**Offset/Low button.** This button selects the setting which was last used (**Offset** or **Low Level**).

**Duty/Width button.** This button only operates when the **Pulse** function is selected for the current channel. Otherwise, the button does nothing when pushed. The instrument remembers which setting (**Duty** or **Width**) was last selected from the bezel menu and highlights that setting when this shortcut button is pushed.

**Leading/Trailing button.** This button only operates when the **Pulse** function is selected for the current channel, otherwise, the button does nothing when pushed. The instrument remembers which setting (**Leading Edge** or **Trailing Edge**) was last selected from the bezel menu and highlights that setting when this shortcut button is pushed.

**Phase | Delay button.** This shortcut button is different from the other buttons. This button does not toggle between two parameters. For example, if you push the **Phase | Delay** shortcut button in the pulse parameter menu, **Delay** becomes active. Pushing the **Phase | Delay** button again will have no effect, because there is no Phase parameter in the pulse parameter menu. Similarly, when you push the **Phase | Delay** button in the Sine, Square, or Ramp parameter menu, **Phase** becomes active. Pushing the button again will have no effect, because there is no Delay parameter in these menus.

## Knob and Arrow Keys

The general purpose knob can be used to increase and decrease selected numeric values. The arrow keys (digit select keys) are used to move the underbar to a field that contains an editable number. This will allow you to change the digit with the knob. Refer to page 1-15 for entering or changing numeric values using the knob and the arrow keys.

## BNC Connectors

Refer to Figure 1-2 on page 1-2 for the locations of the front panel BNC connectors.

**CH1 Output.** This BNC connector will output the Channel 1 signal. This connector will be deactivated when the Channel 1 output button is not selected. The load impedance for this connection can be set in the Output Menu.

**CH2 Output.** Same functionality as Channel 1 Output. This output is not present in the single channel instrument model.

**Trigger Output.** This connector provides a TTL level pulse synchronized with the Channel 1 output. The connector provides a signal that will allow an oscilloscope to synchronize with the arbitrary/function generator.

When you synchronize multiple arbitrary/function generators, the Trigger Output on the master instrument is connected to the Trigger Input of the slave instrument.

**Trigger Input.** When the arbitrary/function generator is a slave to another device, The Trigger Input connector will be used to synchronize the arbitrary/function generator with the master device. Trigger signals from other devices can also be input here.

## USB Memory

The USB Memory connector is a host connector, which allows a USB client memory device to be connected. You can perform the following tasks:

- Save or recall user-defined waveforms to/from a USB memory
- Save or recall setups to/from files on a USB memory
- Update your arbitrary/function generator firmware



---

**CAUTION.** Do not remove USB memory while writing or reading data. It may cause data loss and the USB memory may be damaged.

*Before reinsert a USB memory after removing it from the instrument, wait at least one second.*

*Do not repeat insertion and removal of USB memory without waiting one second each time to avoid damage to the instrument.*

---

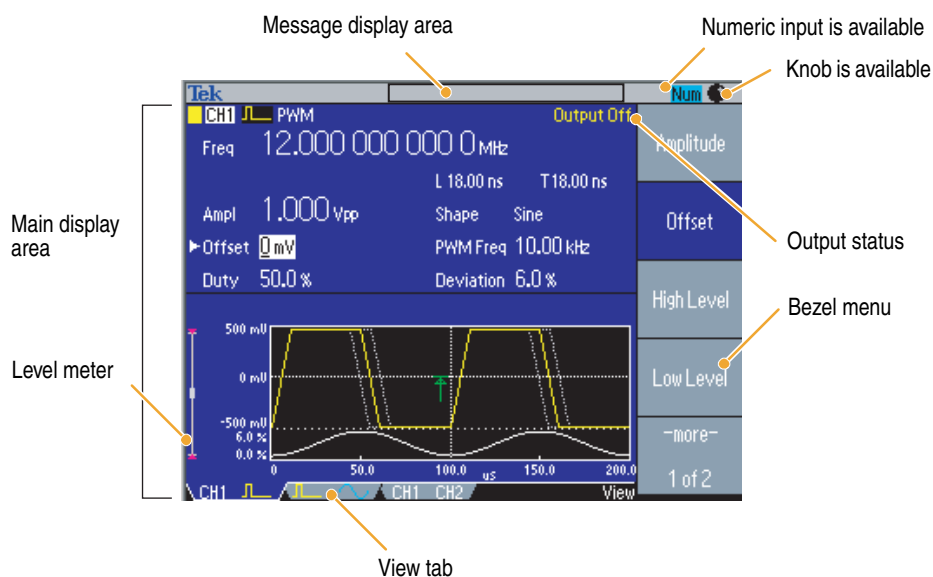
## To Unlock Front Panel Controls

The front panel may be locked by a remote user while the instrument is being remotely controlled via GPIB, USB or Ethernet. When the front panel is locked, all keys and buttons are disabled except the power switch. The “Lock” symbol at the top right of the screen indicates that the instrument front-panel controls are locked.

To unlock the front-panel, use the remote command or push the front-panel **Cancel** button twice in a row. This method is not applied if the arbitrary/function generator is in the GPIB LLO (Local Lockout) state.

## Display Area and Screen Interface

Figure 1-8 shows the main areas of the instrument display.



**Figure 1-8: Screen interface**

### Main Display Area

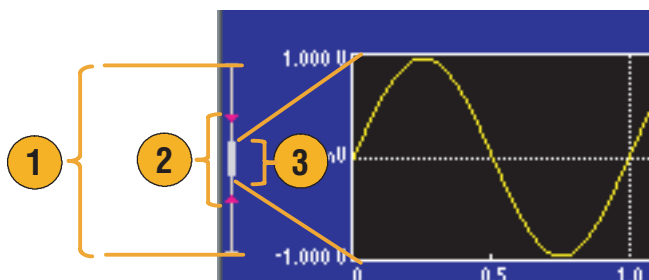
Pushing the front-panel **View** button changes the view format of the main display area. See page 1-3 for screen view formats.

**View Tab.** The view tabs correspond with the current view format.

**Output Status.** If the output is set to disable, the **Output Off** message is displayed in this area. When you push the front-panel channel output button to enable the output, the message will disappear.

From the Output menu, you can set the load impedance, invert a waveform, or add an external signal to the CH1 output. The status will change based on the output status.

**Level Meter.** Amplitude level is displayed. To protect your DUT (device under test), use the Output Menu to set the limit values for high level and low level. Figure 1-9 shows Level Meter.



**Figure 1-9: Level meter**

1. Shows maximum amplitude level of your instrument.
2. Shows the range of high limit and low limit that you have set.
3. Shows the amplitude level that is currently selected.

## Message Display Area

A message that monitors hardware status such as clock or trigger is displayed in this area.

The arbitrary/function generator displays a message at the top of the screen, which conveys the following types of information about hardware status:

- External Reference out of range
- Waiting for Trigger

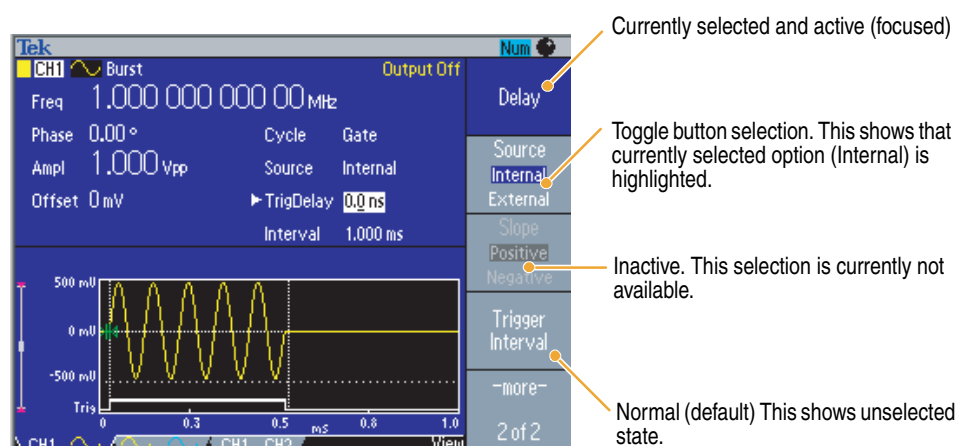
You can also display a text message in this area by using the following remote command. See page 3-20.

```
DISPlay[:WINDow]:TEXT[:DATA]
```

## Bezel Menu Display Area

When you push a front panel button, the instrument displays the corresponding menu on the right side of the screen. The menu shows the options that are available when you push the unlabeled bezel buttons directly to the right of the screen. (Some documentation may also refer to the bezel buttons as side-menu buttons or soft keys.)

The AFG3000 Series Arbitrary/Function Generators use four types of menu button status. See Figure 1-10.



**Figure 1-10: Graphical representation of button status**

- **Focused (active) – Blue background and white type**  
The bezel menu item is currently selected.
- **Non-Focused – Medium gray background and white type, blue box around type only**  
There are some toggle button selections within the bezel menus. For example, Internal and External. You can specify either one but not both of these parameters.
- **Inactive – Medium gray background and light gray type**  
This selection is currently not available because of the other instrument settings.
- **Normal (default) – Medium gray background and white type**  
This is the currently unselected state.

# Waveform Parameters and Numeric Input

This section explains how to set or change the waveform parameters of the arbitrary/function generator using the front-panel controls or bezel menu selection.

## Changing Parameters Using the Bezel Menu

The arbitrary/function generator outputs a sine waveform of 1 MHz frequency with 1 V<sub>p-p</sub> by default. You can use the following two methods to set or change the waveform parameters:

- Using the bezel menu selection
- Using the front-panel shortcut buttons (see page 1-17)

Figure 1-11 shows the default display of sine waveform.

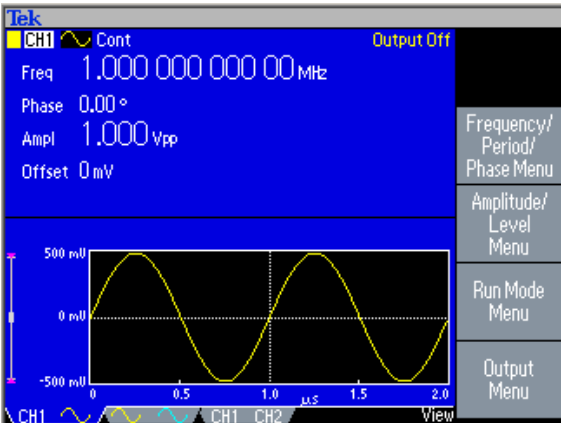


Figure 1-11: Default display

Select the **Frequency/Period/Phase Menu** bezel button from the default display, you can change the values of frequency, period, or phase.

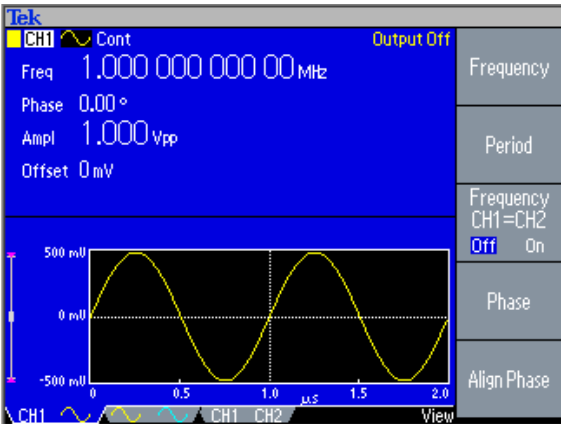
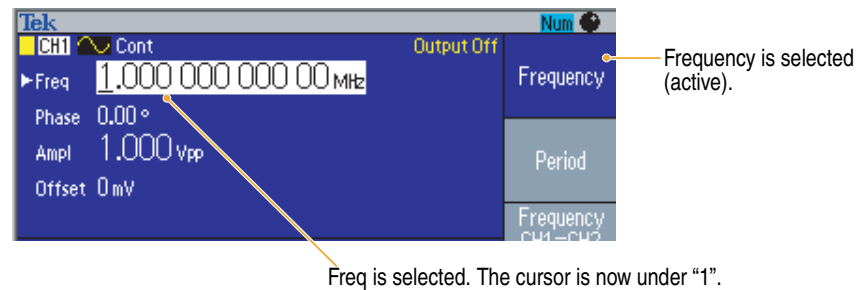


Figure 1-12: Frequency/Period/Phase Menu



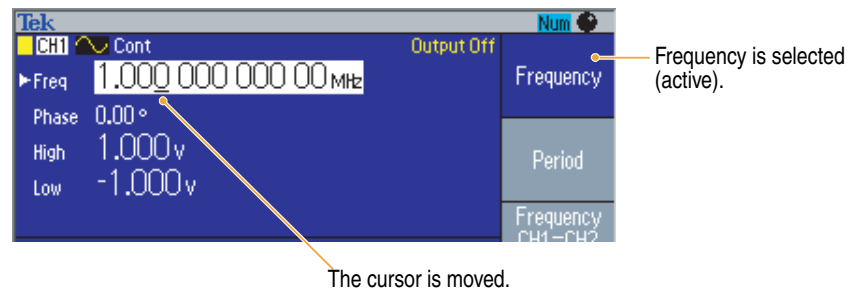
**Numeric Input**

If you want to change the frequency value, push the **Frequency** bezel button. The value of Freq in main display area changes to “selected status”. The Freq is displayed in black type inside a white box. See Figure 1-13.



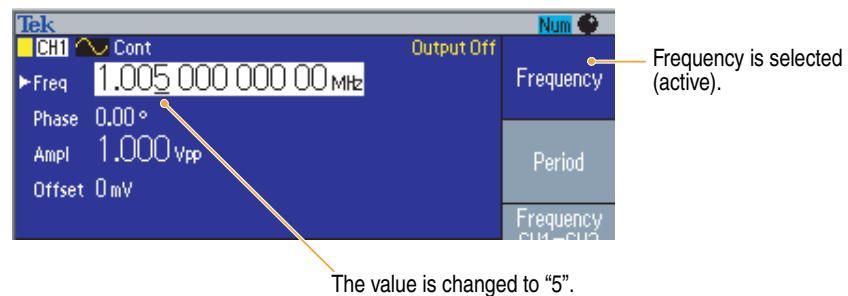
**Figure 1-13: Screen display with Frequency active (1)**

To move the cursor, use the arrow keys.



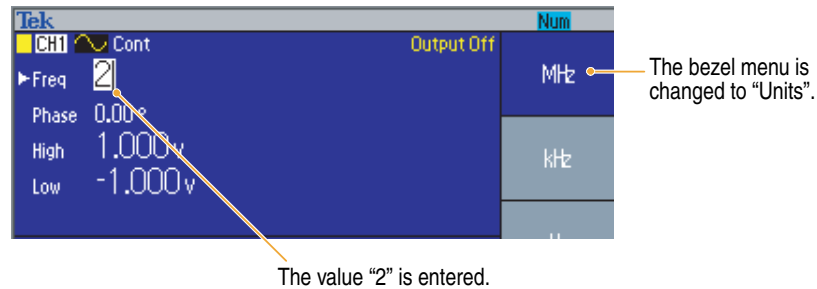
**Figure 1-14: Screen display with Frequency active (2)**

To change the value, use the general purpose knob. Turn the knob clockwise to increase the value; turn the knob counterclockwise to decrease the value.




**Figure 1-15: Screen display with Frequency active (3)**

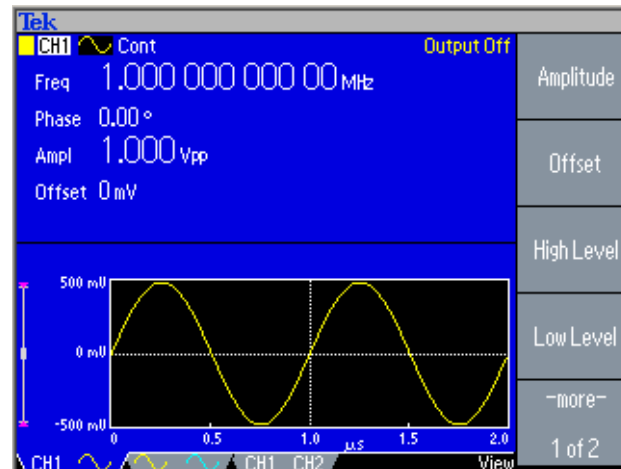
You can also change the value with the front-panel numeric key-pad. Entering any value from the numeric key-pad will automatically change the bezel menu to Units. See Figure 1-16.



**Figure 1-16: Screen display with Frequency active (4)**

After entering the frequency value, push the Units bezel button or the front-panel **Enter** button to complete the entry.

To change the amplitude value, push the Top Menu  button, and then select the **Amplitude/Level Menu** bezel button. You can change the values of amplitude, offset, high level, or low level.



**Figure 1-17: Amplitude/Level Menu**

## Changing Parameters Using the Shortcut Buttons

The shortcut buttons are provided for experienced users. The buttons allow you to select a setup parameter without using any bezel menu selection. The following example shows how the Frequency/Period shortcut button works.

You can use the shortcut buttons while viewing any of the display formats. Push a shortcut button to display the last view type and highlight the selected parameter setting. Figure 1-18 is a sample screen of Pulse Width Modulation parameter menu display.

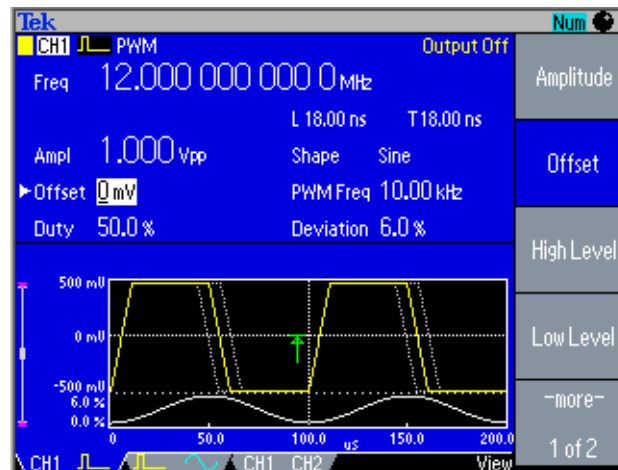


Figure 1-18: PWM sample screen

From Figure 1-18, pushing the **Frequency/Period** shortcut button will change the bezel menu to look like Figure 1-19.

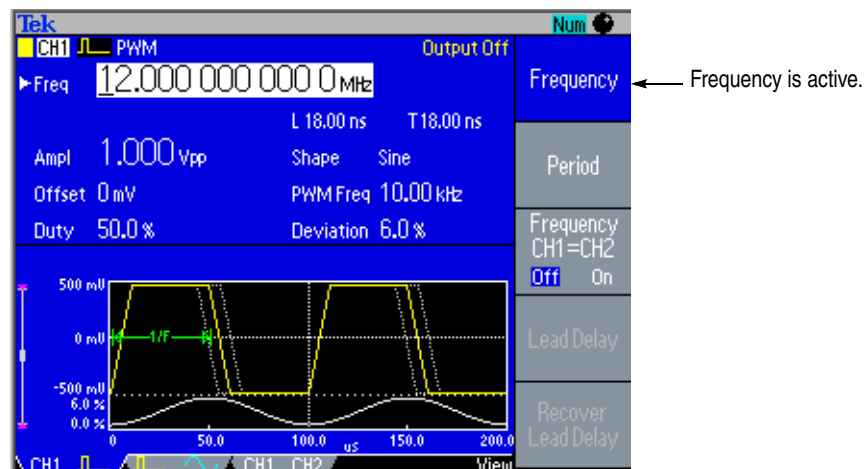
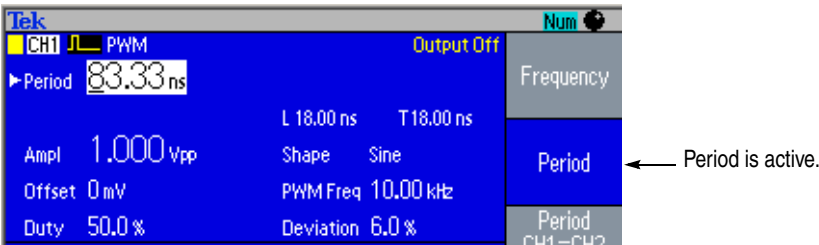


Figure 1-19: PWM parameter menu (Freq is selected)

You can now change the frequency value. If you push the **Frequency/Period** shortcut button again, the active parameter will change to Period. See Figure 1-20.



**Figure 1-20: PWM parameter menu (Period is selected)**

The **Frequency/Period** shortcut button is used to select the setting that was last used (**Frequency** or **Period**). If Frequency was selected, you can change the shortcut by pushing the shortcut button again. The next time you push the **Frequency/Period** button, **Period** will be selected.

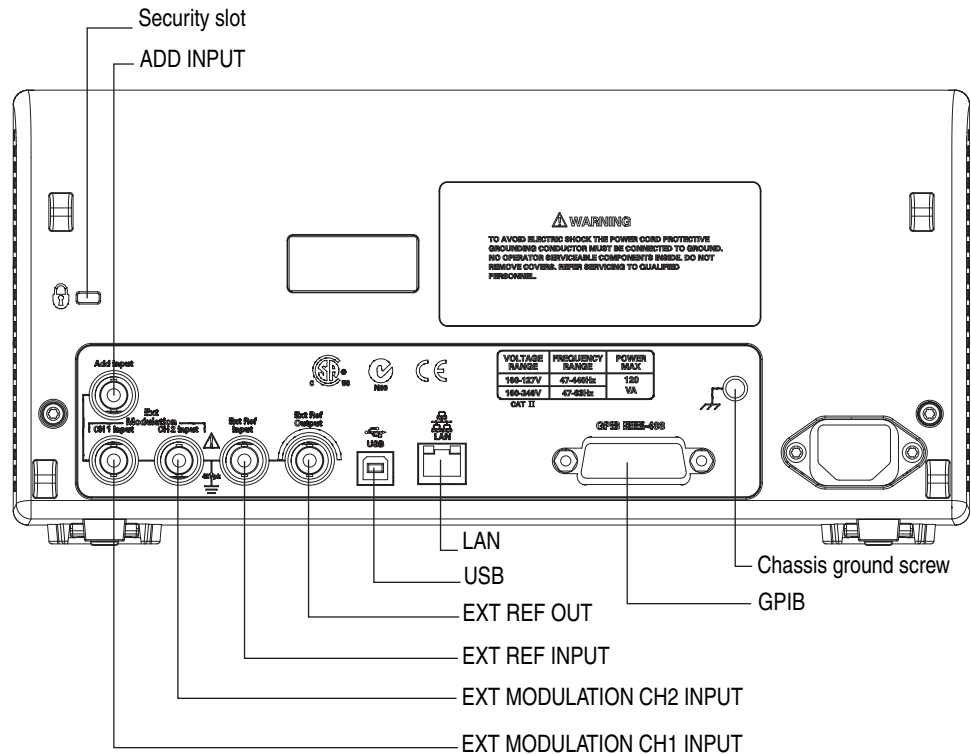
---

**NOTE.** The **Duty/Width** and **Leading/Trailing** shortcut buttons are operational only when **Pulse** is selected in the **Functional** button.

---

## Rear Panel

Figure 1-21 shows the locations of the instrument rear panel connectors.



**Figure 1-21: Rear panel connectors**

**EXT REF INPUT.** This input is used when synchronizing multiple arbitrary/function generators or an arbitrary/function generator and another device.

**EXT REF OUTPUT.** This output is used when synchronizing multiple arbitrary/function generators or an arbitrary/function generator and another device.

**ADD INPUT.** (Dual-channel model only) Additional Input connector allows you to input a signal from some other source and add that signal to CH 1 output.

**EXT MODULATION CH 1 INPUT.** A signal applied to the External Modulation CH 1 Input connector is used to modulate the CH 1 output signal. The signal input level applied to this connector will control the modulation depth.

**EXT MODULATION CH 2 INPUT.** (Dual-channel model only) The External Modulation CH 2 Input connector is used to apply an external modulating signal to the CH 2 output signal. The signal input level applied to this connector will control the modulation depth.

**USB.** The USB connector is used to connect a USB controller.

**LAN.** This connector is used to connect the arbitrary/function generator to a network. Connect a 10BASE-T or 100BASE-T cable here.

**GPIB.** The GPIB connector is used to control the arbitrary/function generator through remote commands.

**Security Slot.** Use a standard laptop computer security cable to secure your arbitrary/function generator to your location.

**Chassis Ground Screw.** The chassis ground screw is used to ground the arbitrary/function generator. Use a unified coarse screw (#6-32, 6.35 mm length or less).



# Reference





# Reference

This section describes the menus associated with each front-panel menu button or control.

## Menu System

The user interface of the AFG3000 Series Arbitrary/Function Generators was designed for easy access to specialized functions through the menu structure.

When you push a front panel-button, the arbitrary/function generator displays the corresponding menu on the right side of the screen. The menu shows the options that are available when you push the unlabeled bezel buttons directly to the right of the screen. (Some documentation may also refer to the bezel buttons as side-menu buttons or soft keys.)

### Using the Menu System

The arbitrary/function generator uses the following methods to display menu options:

**Submenu Selection.** For some menus, you can use the bezel button to choose the submenus. For example, when you push the **I/O Interface** bezel button in the Utility menu, the arbitrary/function generator displays the I/O Interface submenu (GPIB or Ethernet).

**Action.** The arbitrary/function generator displays the type of action that will immediately occur when you push an action bezel button. For example, when you push the front-panel **Save** button and then push the **Save** bezel button, the arbitrary/function generator saves the settings immediately.

**Option Buttons.** The arbitrary/function generator uses a different button for each option. The currently selected option is highlighted. For example, the instrument displays various language options when you push the **Utility** menu button and then push the **Language** bezel button. To select the option, push the corresponding button. Option button is also called radio button.

**Toggle.** For some menus, pushing a bezel button will toggle the corresponding menu options. For example, when you push the **Power On** bezel button in the Utility menu, it will toggle between **Default** and **Last**. Pushing the **Beeper** bezel button will toggle between **On** and **Off**.

## Menu Structure

This section describes the menus and operating details associated with each front-panel menu button.

- Sine/Square Menu, page 2-3
- Ramp Menu, page 2-4
- Pulse Menu, page 2-4
- Arb Menu, page 2-5
- More... Menu, page 2-6
- Run Mode Menus
  - Continuous (No bezel menu for Continuous mode)
  - Modulation Parameter Menu, page 2-7
  - Sweep Parameter Menu, page 2-9
  - Burst Parameter Menu, page 2-11
- Output Menu, page 2-12

---

**NOTE.** To access the Output menu, push the front-panel Top Menu button. The arbitrary/function generator does not have the Output menu button on the front panel. Refer to page 1-3 for the Top Menu button.

---

- Save/Recall Menu, page 2-13
- Edit Menu, page 2-14
- Utility Menu, page 2-23

**Sine/Square Menu** Table 2-1 shows the **Sine** and **Square** menu.

**Table 2-1: Sine/Square Menu**

First Level	Second Level	Third Level	Description
Frequency/Period/ Phase Menu	Frequency		Selects <b>Frequency</b> as a parameter to be changed.
	Period		Selects <b>Period</b> as a parameter to be changed.
	Frequency (Period) <sup>1</sup> CH1=CH2 Off On		Toggles between <b>Off</b> and <b>On</b> to disable/enable the function that sets the CH1 and CH2 to the same value.
	Phase		Selects <b>Phase</b> as a parameter to be changed.
	Align Phase <sup>1</sup>		Use to align the phase of two channel signals.
Amplitude/Level Menu	Amplitude		Selects <b>Amplitude</b> as a parameter to be changed.
	Offset		Selects <b>Offset</b> as a parameter to be changed.
	High Level		Selects <b>High Level</b> as a parameter to be changed.
	Low Level		Selects <b>Low Level</b> as a parameter to be changed.
	-more- (1 of 2)		
	Units [Vpp]	Vpp Vrms dBm <sup>2</sup>	Option buttons
	Level <sup>1</sup> CH1=CH2 Off On		Toggles between <b>Off</b> and <b>On</b> to disable/enable the function that sets the CH1 and CH2 to the same value.
	-more- (2 of 2)		
Run Mode Menu	Continuous		—
	Modulation		Refer to Table 2-6
	Sweep		Refer to Table 2-7
	Burst		Refer to Table 2-8
Output Menu			Refer to Table 2-9

1. Dual-channel model only.
2. dBm is used only for sine waveform.

**Ramp Menu** Table 2-2 shows the **Ramp** menu.

**Table 2-2: Ramp Menu**

First Level	Second Level	Description
Ramp Parameter Menu	Symmetry	Sets the Ramp waveform symmetry to your desired value.
	Set to 0%	Sets the Ramp waveform symmetry to 0%.
	Set to 50%	Sets the Ramp waveform symmetry to 50%.
	Set to 100%	Sets the Ramp waveform symmetry to 100%.
Frequency/Period/Phase Menu		Refer to Table 2-1
Amplitude/Level Menu		Refer to Table 2-1
Run Mode Menu	Continuous	—
	Modulation	Refer to Table 2-6
	Sweep	Refer to Table 2-7
	Burst	Refer to Table 2-8
Output Menu		Refer to Table 2-9

**Pulse Menu** Table 2-3 shows the **Pulse** menu.

**Table 2-3: Pulse Menu**

First Level	Second Level	Description
Pulse Parameter Menu	Duty	Selects <b>Duty</b> as a parameter to be changed.
	Width	Selects <b>Width</b> as a parameter to be changed.
	Leading Edge	Selects <b>Leading Edge</b> as a parameter to be changed.
	Trailing Edge	Selects <b>Trailing Edge</b> as a parameter to be changed.
Frequency/Period/Delay Menu	Frequency	Selects <b>Frequency</b> as a parameter to be changed.
	Period	Selects <b>Period</b> as a parameter to be changed.
	Frequency (Period) <sup>1</sup> CH1=CH2 Off On	Toggles between <b>Off</b> and <b>On</b> to disable/enable the function to set the CH1 and CH2 to the same value.
	Lead Delay	Selects <b>Lead Delay</b> as a parameter to be changed.
	Recover Lead Delay <sup>1</sup>	Use to align the lead delay of two channel signals.
Amplitude/Level Menu		Refer to Table 2-1
Run Mode Menu	Continuous	—
	Modulation	Refer to Table 2-6
	Burst	Refer to Table 2-8
Output Menu		Refer to Table 2-9

1. Dual-channel model only.

**Arb Menu** The arbitrary/function generator can output a user-defined waveform that is stored in the internal memory, Edit Memory, or a USB memory. Table 2-4 shows the **Arb** menu.

**Table 2-4: Arb Menu**

First Level	Second Level	Description
Arb Waveform Menu	Memory Internal USB	Selects a memory type. (Internal or USB) Refer to page 2-28 for file operations.
	(blank)	
	Change Directory	Opens a directory in a USB memory.
	OK	
	Cancel	
Frequency/Period/ Phase Menu		Refer to Table 2-1
Amplitude/Level Menu		Refer to Table 2-1
Run Mode Menu	Continuous	—
	Modulation	Refer to Table 2-6
	Sweep	Refer to Table 2-7
	Burst	Refer to Table 2-8
Output Menu		Refer to Table 2-9

To output a user-defined waveform, you have two options:

- Output the contents of Edit Memory
- Output one of user-defined waveforms stored in the internal or in a USB memory

To output a user-defined waveform, you must first save the file in the internal or USB memory. You can save up to four user-defined waveforms in the internal waveform memory. Waveform data in the Edit Memory can also be output, but the contents of Edit Memory is deleted when the instrument power is turned off.

---

**NOTE.** When the arbitrary/function generator outputs an Arb waveform, the peak-to-peak voltage of the measurement result is the same as the setup value if the waveform data is normalized.

---

**More... Menu** Table 2-5 shows the **More...** menu.

**Table 2-5: More Menu**

First Level	Second Level	Description
More Waveform Menu	Sin(x)/x	Option buttons
	Noise	
	DC	
	Gaussian	
	-more- (1 of 2)	
	Lorentz	Option buttons
	Exponential Rise	
	Exponential Decay	
	Haversine	
	-more- (2 of 2)	
Frequency/Period/ Phase Menu		Refer to Table 2-1
Amplitude/Level Menu		Refer to Table 2-1
Run Mode Menu	Continuous	—
	Modulation	Refer to Table 2-6
	Sweep	Refer to Table 2-7
	Burst	Refer to Table 2-8
Output Menu		Refer to Table 2-9

**NOTE.** When the arbitrary/function generator outputs Sin(x)/x, Gaussian, Lorentz, Exponential Rise, Exponential Decay, or Haversine waveforms, the peak-to-peak voltage is defined as twice the 0 to peak value.

## Modulation Parameter Menu

Table 2-6 shows the Modulation Parameter Menu.

**Table 2-6: Modulation Parameter Menu**

First Level	Second Level	Description
Modulation Type [AM]	AM FM PM FSK PWM	Selects modulation type.
[AM] Source Internal External		Selects modulation source (internal or external) for AM, FM, PM, FSK, or PWM.
AM, FM, PM, PWM Frequency, or FSK Rate		Sets modulation frequency or FSK rate.
Modulation Shape <sup>1</sup>	Memory Internal USB	Selects a memory type (internal or USB).
	(blank)	
	Change Directory	Opens a directory in a USB memory.
	OK	
	Cancel	
Depth/Deviation/ Hop Frequency		Sets modulation depth for AM, deviation for FM and PM, or Hop Frequency for FSK.

1. If you specify **Internal** as the memory type, you can select the modulation shape from the following:

Sine  
 Square  
 Triangle  
 Up Ramp  
 Down Ramp  
 Noise  
 Arb (User1 to User4)  
 Edit Memory

**Amplitude Modulation.** Amplitude modulation is a modulation technique in which the amplitude of the carrier frequency is modified by the amplitude of the modulating waveform (internal or external modulation source).

The modulation depth is expressed as a percentage and represents the extent of the amplitude variation.

**Frequency Modulation.** Frequency modulation is a modulation technique in which the carrier frequency is modified by the amplitude of the modulating waveform.

The carrier frequency must always be greater than or equal to the deviation. Also the sum of the carrier frequency and deviation must be less than or equal to the maximum frequency for the selected waveform.

**Phase Modulation.** Phase modulation is similar to FM (Frequency modulation), but in PM, the phase of the carrier waveform is varied by the amplitude of the modulating waveform.

**Frequency Shift Keying.** Frequency shift keying modulation is a modulation technique that shifts the output signal frequency between two frequencies: the carrier frequency and the FSK Hop frequency.

**Pulse Width Modulation.** Pulse width modulation is a modulation technique in which the width of a pulse waveform is varied by the voltage of the modulating waveform.



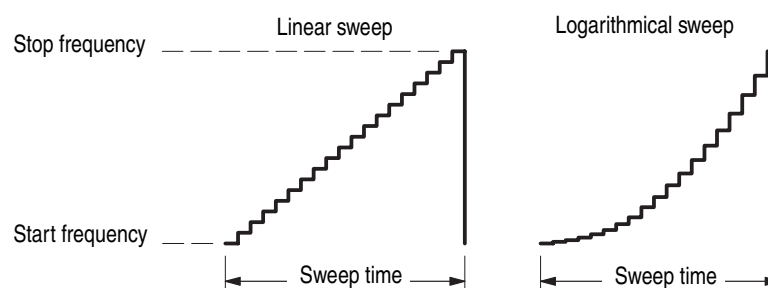
**Sweep Parameter Menu**

Table 2-7 shows the Sweep Parameter Menu.

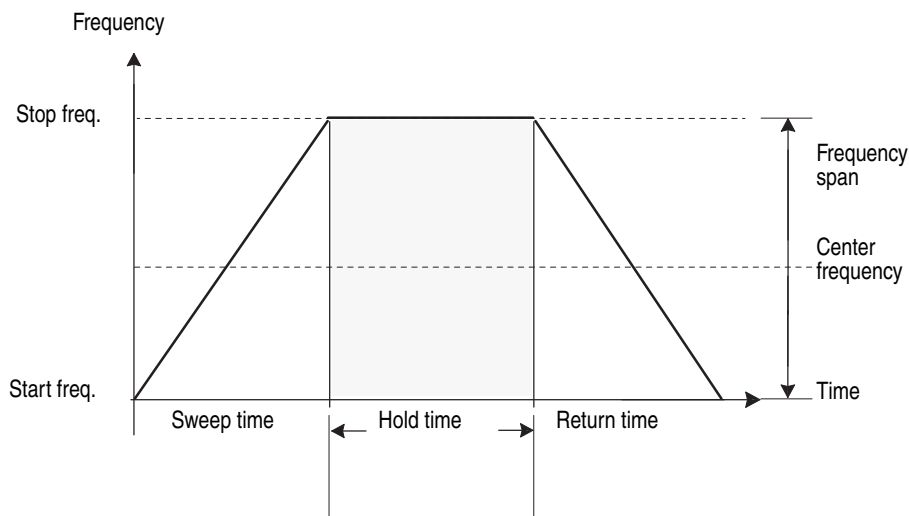
**Table 2-7: Sweep Parameter Menu**

First Level	Description
Start Frequency	Numeric input. Sets start frequency.
Stop Frequency	Numeric input. Sets stop frequency.
Sweep Time	Numeric input. Sets sweep time.
Return Time	Numeric input. Sets return time.
-more- (1 of 3)	
Center Frequency	Numeric input. Sets center frequency.
Span	Numeric input. Sets frequency span.
Hold Time	Numeric input. Sets hold time.
Type Linear Logarithm	Selects sweep type (Linear or Logarithm). See Figure 2-1.
-more- (2 of 3)	
Mode Repeat Trigger	Selects sweep mode (Repeat or Trigger).
Source Internal External	Selects trigger source (Internal or External).
Slope Positive Negative	Selects trigger slope (Positive or Negative). Available only when you select <b>External</b> as a trigger source.
Trigger Interval	Numeric input. Sets trigger interval. Available only when you select <b>Internal</b> as a trigger source.
-more- (3 of 3)	

The sweep function outputs a waveform with the output signal frequency varying linearly or logarithmically.

**Figure 2-1: Sweep type**

When sweep is taking place, the frequency sweeps from start frequency to stop frequency. Refer to Figure 2-2.



**Figure 2-2: Frequency sweep**

The instrument sweeps from a low frequency to a high frequency when start frequency is set to be lower than stop frequency (start frequency < stop frequency).

The instrument sweeps from a high frequency to a low frequency when start frequency is set to be higher than stop frequency.

The frequency range differs based on the selected waveform types. For frequency sweep, you can select a sine, square, ramp, or arbitrary waveform. Pulse, DC, and Noise waveforms cannot be selected.

You can select one of the following two operation modes:

**Repeat.** A continuous sweep is output at a rate specified by the sweep time, hold time, and return time.

**Trigger.** One sweep waveform is output when a trigger input is received.

You can select a trigger source from an external signal applied to the front-panel Trigger Input connector, the trigger signal generated internally, the manual trigger, or the remote interface.

**Burst Parameter Menu**

Table 2-8 shows the Burst Parameter Menu.

**Table 2-8: Burst Parameter Menu**

First Level	Description
1-Cycle	Sets burst count to 1.
N-Cycle	Sets burst count to N. (N=1 to 1,000,000)
Inf-Cycle	Sets burst count to Infinite.
Gate	Selects gated burst mode.
-more- (1/2)	
Delay	Sets trigger delay between the receipt of the trigger and the start of the burst waveform. The range is 0 to 500 s.
Source Internal External	Select trigger source (Internal or External).
Slope Positive Negative	Selects trigger slope (Positive or Negative). Available only when you select <b>External</b> as a trigger source.
Trigger Interval	Sets trigger interval. Available only when you select <b>Internal</b> as a trigger source.
-more- (2/2)	

**Triggered Burst Mode.** When a trigger event occurs in burst mode, the waveform is output for the number of cycles specified by the Burst Parameter Menu. Once the triggered burst mode is selected, the arbitrary/function generator will wait for a trigger event to occur.

**Gated Burst Mode.** In the gated burst mode, the output is enabled or disabled based on the internal gate signal or an external signal applied to the front panel Trigger Input connector. While the gate signal is true or the front panel Manual Trigger button is depressed, the instrument outputs a continuous waveform.

When the trigger source is set to **Internal**, the trigger interval can be set. The setting range of trigger interval is 1 ms to 500 s. The default is 1 ms.

When the trigger source is set to **External**, you can select Positive or Negative as the trigger slope. The instrument will trigger on the rising edge (Positive) or falling edge (Negative) of the signal applied to the front-panel Trigger Input connector.

**Output Menu**

Table 2-9 shows the Output Menu. To access the Output menu, push the front-panel Top Menu button, and then push the bottom bezel button.

**Table 2-9: Output Menu**

First Level	Second Level	Description
Load Impedance	50 $\Omega$	Option buttons. Set the load impedance to 50 $\Omega$
	Load <sup>1</sup>	Option buttons & Numeric input. You can set the load impedance to any value from 1 $\Omega$ to 10 k $\Omega$
	High Z	Option buttons. Select high impedance.
Invert Off On		Selects Invert <b>On</b> to invert a waveform.
Limit	High Limit	Numeric input.
	Low Limit	Specifies the higher and lower amplitude limit. See <i>Level Meter</i> on page 1-12.
Noise	Noise Add Off On	Selects Noise Add <b>On</b> to add the internal noise to a waveform.
	Noise Level <sup>2</sup>	Sets the noise level.
External Add <sup>3</sup> Off On		Selects External Add <b>On</b> to add an external signal to the CH 1 output signal.

1. Range: 1  $\Omega$  to 10 k $\Omega$   
When dBm is currently selected for output amplitude units, the amplitude units setting is automatically changed to V<sub>p-p</sub> if you select high impedance.
2. Range: 0% to 50% of amplitude setting (V<sub>p-p</sub>) of signal waveform, resolution 1%  
The maximum value of noise range is limited by waveform and amplitude range.
3. AFG3100 and AFG3200 series only.

**Save/Recall Menu**

The **Save** and **Recall** menus store or recall arbitrary/function generator setups. The setups are stored in either internal setup memory or USB memory.

**Table 2-10: Save/Recall Menu**

Save or Recall	First Level	Second Level	Description
Save	Memory Internal USB		Selects a memory type (Internal or USB).
	Save		Saves the instrument settings to the specified setup memory location.
	Save As	Enter Character	Refer to page 2-32 for entering a file name.
		OK	
		Cancel	
	Change Directory		Opens a directory.
	-more- (1/2)		
	Lock/Unlock		Locks or unlocks a file.
	Erase		Erases a file.
	(blank)		
	Make Directory	Enter Character	Refer to page 2-28 for file operations.
		OK	
		Cancel	
	-more- (2/2)		
Recall	Memory Internal USB		Selects a memory type (Internal or USB).
	Recall		Recalls the instrument settings from the specified file.
	(blank)		
	Change Directory		Opens a directory.
	(blank)		

The arbitrary/function generator has four storage locations in non-volatile memory to store the instrument setups. The locations are numbered 1 through 4. The instrument uses location 0, which is called “last setup memory”, to overwrite the instrument setups at power off.

You can select the instrument settings that are restored when you power on the instrument. If you select **Power On Last** in the Utility menu, the instrument will restore the last powered-off settings the next time you power on the instrument. (See the Utility Menu on page 2-23.)

The **Lock/Unlock** menu allows you to lock the file against accidental overwrite. When a memory location is locked, a lock icon appears on the screen.

**Edit Menu**

The **Edit** menu is used to create or edit user-defined waveforms and to import or export waveforms. To output a user-defined waveform, waveform data must be created in the Edit Memory and then stored in the internal or a USB memory. Although you can output the content of Edit Memory, the content of Edit Memory will be deleted when the instrument is powered off.

There are three ways to create a waveform in the Edit Memory:

- Create a waveform file using the Edit Menu
- Copy a waveform file using remote commands
- Import a waveform file using the ArbExpress software

**Table 2-11: Edit Menu**

First Level	Second Level	Description
Number of Points		Sets the number of points of waveform to be created or the length of edit memory.  The default is 1000.
New	Sine	Writes a standard waveform to Edit Memory. (The contents of Edit Memory will be overwritten.) You can select a waveform from five standard waveforms. The written waveform has the number of points specified by the <b>Number of Points</b> bezel menu.
	Square	
	Pulse	
	Ramp	
	Noise	
Operation	Line	Opens Line submenu. See Table 2-12.
	Data	Opens Data submenu. See Table 2-12.
	Cut	Opens Cut submenu. See Table 2-12.
	(blank)	
	(blank)	
Read from...	Memory Internal USB	Selects a memory type (Internal or USB).
	Read	Load one of the user-defined waveforms, which is stored in the internal or USB memory, to Edit Memory. Once <b>Read</b> is executed, the edit waveform is overwritten.
	(blank)	
	Change Directory	Opens a directory. This menu item is available when <b>USB</b> is selected.
	Cancel	
-more- (1/3)		

Table 2-11: Edit Menu (cont.)

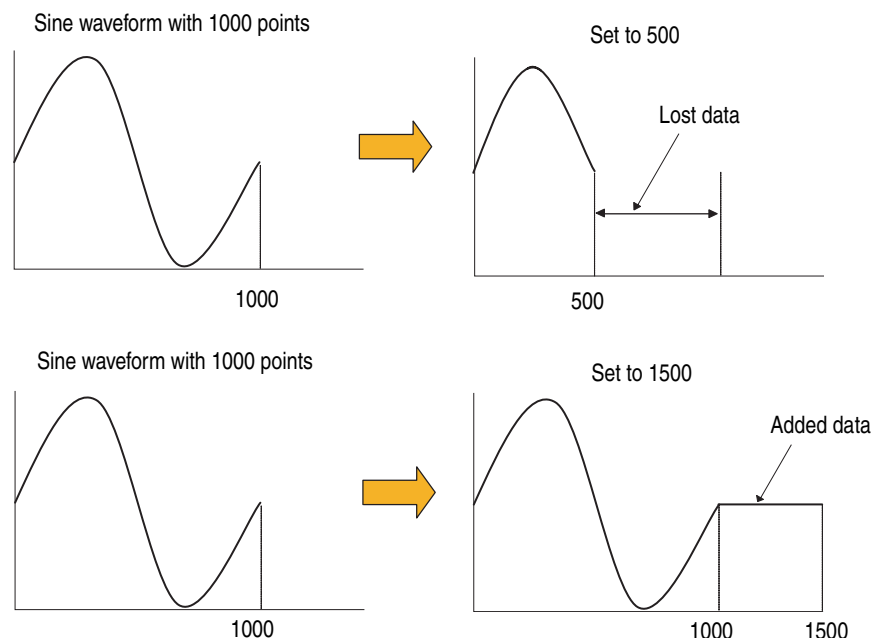
First Level	Second Level	Description
Paste at Beginning	Memory Internal USB	Selects a memory type (Internal or USB).
	Paste	Pastes a waveform at the beginning of the current waveform.
	(blank)	
	Change Directory	Opens a directory. This menu item is available when <b>USB</b> is selected.
	Cancel	
Paste at End	Memory Internal USB	Selects a memory type (Internal or USB).
	Paste	Pastes a waveform at the end of the current waveform.
	(blank)	
	Change Directory	Opens a directory. This menu item is available when <b>USB</b> is selected.
	Cancel	
(blank)		
Write to...	Memory Internal USB	Selects a memory type (Internal or USB).
	Write	Saves the current waveform to the internal or USB memory.
	Write As	Opens the submenu to enter a file name. See page 2-30.
	Change Directory	Opens a directory. This menu item is available when <b>USB</b> is selected.
	-more- (1/2)	
	Lock/Unlock	Locks or unlocks a file.
	Erase	Erase a file.
	(blank)	
	Make Directory	Creates a new directory. This menu item is available when <b>USB</b> is selected.
	-more- (2/2)	
-more- (2/3)		
Zoom Fit		Adjusts the graph scale and display the complete waveform cycle.
(blank)		
(blank)		
(blank)		
-more- (3/3)		

**Number of Points.** This bezel menu sets the number of points of the waveform to be created or the length of Edit Memory. The point can be set to a value in the following range:

- 2 to 65,536 (AFG3021 and AFG3022)
- 2 to 131,072 (AFG3101/AFG3102 and AFG3251/AFG3252)

If a waveform is already written to Edit Memory and **Number of Points** is set to a value smaller than the number of points in that waveform, then the data for the points starting at that value plus one will be lost. For example, if **Number of Points** is set to 500 in the state where a waveform with 1000 points has been written to Edit Memory, the waveform data for points 1 to 500 will remain without change but the data from point 501 will be lost.

If a waveform is already written to Edit Memory and **Number of Points** is set to a value greater than the number of points in that waveform, then the data for the points starting at that value plus one to the point identified by the **Number of Points** will be set to the zero level. For example, if **Number of Points** is set to 1500 in the state where a waveform with 1000 points has been written to Edit Memory, the waveform data for points 1 to 1000 will remain without change and the data from point 1001 to point 1500 will be set to zero level. See Figure 2-3 for an example using the **Number of Points** parameter.



**Figure 2-3: Number of Points**



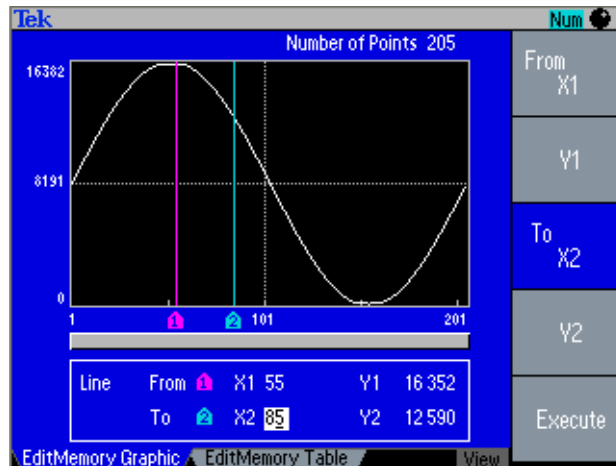
**New.** Pushing the **New** bezel button will display a standard waveform selection submenu. You can select a standard waveform from five waveform types (Sine, Square, Pulse, Ramp, and Noise). For example, selecting **Sine** will write a sine waveform into Edit Memory. The waveform has the number of points specified by the **Number of Points** bezel menu. If waveform data is already stored in the Edit Memory, the existing waveform data will be lost, and the new waveform will overwrite the old data.

**Operations Submenu (Line, Data, and Cut).** For each of the **Operations** submenus, you must push the **Execute** bezel button to implement the edit operation.

**Table 2-12: Operations submenu**

First level	Second level	Description
Line	From X1	Use to edit a line linearly by interpolating from a specified point in the waveform currently being edited to another specified point.
	Y1	
	To X2	
	Y2	
	Execute	
Data	X	Use to change the specified data point in the waveform currently being edited.
	Y	
	(blank)	
	(blank)	
	Execute	
Cut	From X1	Use to delete an area between specified points in the waveform currently being edited.
	(blank)	
	To X2	
	(blank)	
	Execute	
(blank)		
(blank)		

**Using the Cursors.** You can use cursors to edit a waveform. To use cursors, select the **Operations** submenu in the **Edit** menu. Figure 2-4 is a sample of **Line** edit screen.



**Figure 2-4: Using the Cursor (Line edit)**

In the **Line** edit screen, there are two cursor types: X1 cursor and X2 cursor. Use the **From X1** or **To X2** bezel button to select which cursor to move. To move the X1 cursor, push the **From X1** bezel button and then turn the general purpose knob. Or, you can enter a value for the point X1 using the numeric keypad. Entering a value and then pushing the front-panel **Enter** button will move the cursor to the specified point. To move the X2 cursor, push the **To X2** bezel button and then turn the knob or enter a value.

After specified two points, push the **Execute** bezel button to implement the edit operation. The **Line** edit is applied to the portion between the two cursors (X1 and X2).

When you select **Operations > Data**, only one cursor is displayed. The edit operation is applied to the specified point.

When you select **Operations > Cut**, two cursors are displayed like **Line** edit. The edit operation is applied to the portion between the two cursors.

**Line (Line Edit).** The **Line** edits by linearly interpolating from a specified point in the waveform currently edited to another specified point as shown in Figure 2-5.

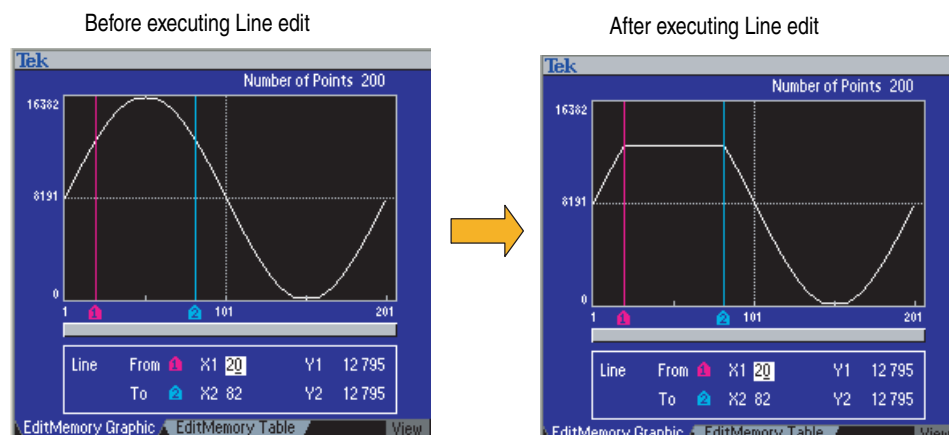


Figure 2-5: Line Edit example

**Data (Edit by Data Point).** The **Data** edits a specified point in the waveform currently being edited.

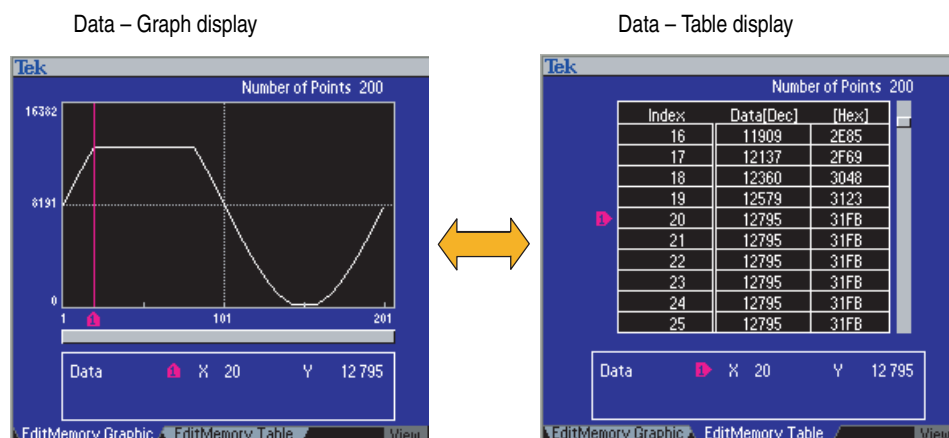
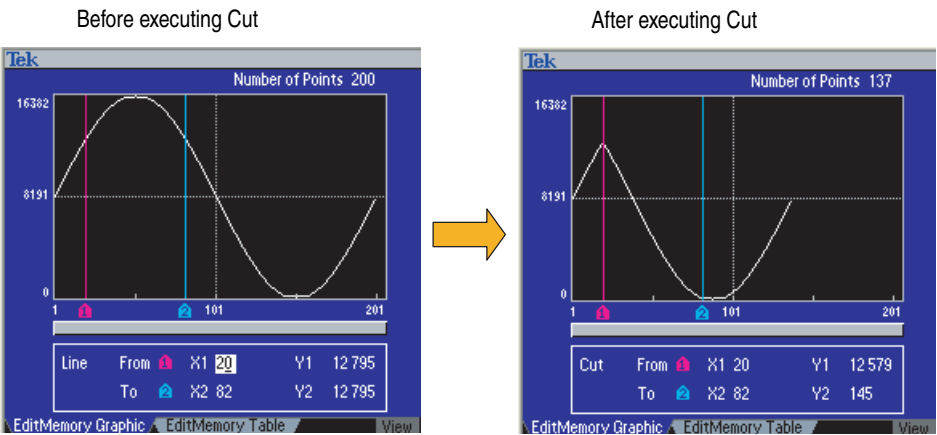


Figure 2-6: Operation – Edit by Data Point

If you enter a number that exceeds the number of points in the current waveform, the entered value will be automatically changed to the number of points in the current waveform.

If you push the **Execute** bezel button in the **Data** operation, the instrument will auto-increment to the next point. For example, you specify a data point X and then push **Execute**, the instrument will automatically move from point X to X + 1 in anticipation of your next input.

**Cut (Cut by Data Points).** The **Cut** deletes an area between specified points in the waveform. When **Cut** is selected, two values can be input (From X1 and To X2). See Figure 2-7.

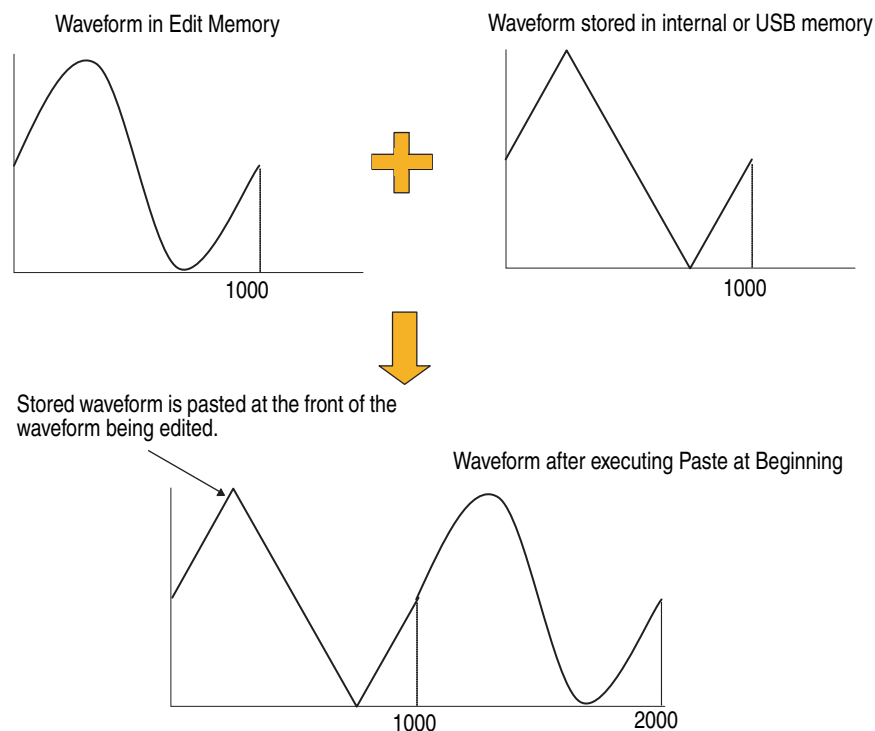


**Figure 2-7: Cut example**

**NOTE.** Data that has been deleted cannot be recovered.

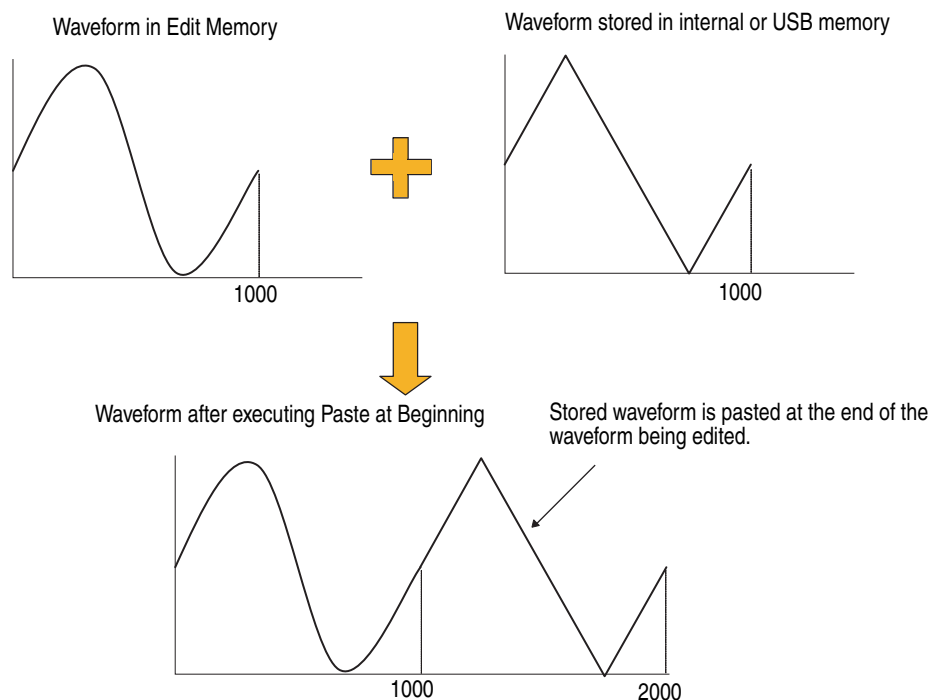
**Read from...** Push the front-panel **Edit** button and then push the **Read from... > Memory** bezel menu to load a waveform file from the internal memory or a USB memory into Edit Memory.

**Paste at Beginning.** The **Paste at Beginning** inserts the contents of the user waveform memory (User1, User2, User3, or User4), USB memory, or the current contents of Edit Memory at the beginning of the waveform that is currently being edited. When this operation is executed, the Number of Points setting is automatically modified. If this operation would cause the waveform to exceed 65,536 (AFG3021 and AFG3022) or 131,072 (AFG3101/AFG3102 and AFG3251/AFG3252) points, the waveform insertion will not be executed.



**Figure 2-8: Paste at Beginning**

**Paste at End.** The **Paste at End** appends the contents of the user waveform memory (User1, User2, User3, or User4), USB memory, or the current contents of edit memory at the end of the waveform currently being edited. When this operation is executed, the Number of Points setting is automatically modified. If this operation would cause the waveform to exceed 65,536 (AFG3021 or AFG3022) or 131,072 (AFG3101/AFG3102 and AFG3251/AFG3252) points, the waveform append will not be executed.



**Figure 2-9: Paste at End**

**Write to...** The currently edited waveform is in Edit Memory. If the instrument power is turned off, the contents of Edit Memory will be lost. To allow this waveform to be used the next time the power is turned on, the contents of Edit Memory must be saved into the user waveform memory or a USB memory.

For more information on saving a user-defined waveform file, refer to *Saving a Waveform File* on page 2-29.

**Utility Menu** The **Utility** menu provides access to utilities used by the arbitrary/function generator such as language selection, instrument diagnostics, and user preferences.

**Table 2-13: Utility Menu**

First Level	Second Level	Third Level	Description
I/O Interface <sup>1</sup>	GPIB	Address	Sets the instrument GPIB address.
		Configuration Talk/Listen Off Bus	Select <b>Talk/Listen</b> to enable the GPIB bus communication.  Select <b>Off Bus</b> to isolate the instrument from the GPIB bus communication.
	Ethernet <sup>2</sup>	IP Address	Sets an IP address manually.
		Subnet Mask	Sets a subnet mask manually.
		Default Gateway	Sets a default gateway manually.
		DHCP <sup>3</sup> Off On	Select <b>DHCP On</b> to assign an IP Address automatically.  Select <b>DHCP Off</b> to assign an IP Address manually.
	Language	English	Selects a language to display on the instrument screen.  After selecting a desired language, the bezel menu, pop-up message, and built-in help are displayed in the selected language. The main display area (see page 1-11) is not translated.
		French	
		German	
		Japanese	
		-more- (1 of 2)	
		Korean	
		Simplified Chinese	
		Traditional Chinese	
	(blank)		
	-more- (2 of 2)		

Table 2-13: Utility Menu (cont.)

First Level	Second Level	Third Level	Description
System	Trigger Out Trigger Sync	Select <b>Trigger</b> to generate the trigger signal for the oscilloscope.  Select <b>Sync</b> to synchronize multiple AFG3000 series instruments.	
	Clock Ref Internal External	Select <b>Internal</b> to accept the internal reference clock.  Select <b>External</b> to accept an external reference clock.	
	Power On Default Last	Select <b>Default</b> to set the instrument power-on status to the default settings.  Select <b>Last</b> to set the instrument power-on status to the last powered-off settings.	
	Secure	Select <b>Secure</b> to erase all data except Mac Address, calibration data, and the instrument serial number.	
	-more- (1 of 2)		
	Contrast	Select <b>Contrast</b> to change the instrument display contrast.	
	Screen Saver Off On	Select <b>Screen Saver Off</b> to deactivate the screen saver.  Select <b>Screen Saver On</b> to activate the screen saver.	
	Click Tone Off On	Select <b>Click Tone Off</b> to deactivate the click sound.  Select <b>Click Tone On</b> to activate the click sound.	
	Beeper Off On	Select <b>Beeper Off</b> to deactivate the beeper sound.  Select <b>Beeper On</b> to activate the beeper sound.	
	-more- (2 of 2)		
Status	Setup	Select <b>Setup</b> to list the CH1 and CH2 waveform parameter configuration.  Select <b>Misc.</b> to list the instrument network configuration and user preferences.	
	Misc.		
	(blank)		
	Page Up	Returns to previous page.	
	Page Down	Moves to next page.	
-more- (1 of 3)			



Table 2-13: Utility Menu (cont.)

First Level	Second Level	Third Level	Description
Diagnostics/ Calibration	Execute Diagnostics	Performs the instrument diagnostics.	
	Execute Calibration	Performs the instrument self calibration.	
Backup/ Restore <sup>4</sup>	Type Setup Arb	Selects file type (Setup or Arb).	
	Backup Internal -> USB Memory	Copies the contents of internal memory to a USB memory.	
	Restore Internal <- USB Memory	Copies the files in the USB memory to the internal memory.	
	Change Directory	Opens a directory in a USB memory.	
	Make Directory	Creates a new directory in a USB memory.	
Channel Copy	CH1 -> CH2	Copies the CH1 waveform parameter settings to CH2.	
	CH2 -> CH1	Copies the CH2 waveform parameter settings to CH1.	
(blank)			
-more- (2 of 3)			
Firmware Update <sup>4</sup>	Execute	Performs the instrument firmware update.	
	(blank)		
	(blank)		
	Change Directory	Opens a directory in a USB memory.	
	Cancel		
-more- (3 of 3)			

1. AFG3100 and AFG3200 series only.
2. To connect your instrument to a network, you must first obtain information from your network administrator. The procedure for entering the Ethernet network parameters depends on your network configuration.
3. If your network supports DHCP (Dynamic Host Configuration Protocol), select the DHCP On so that the instrument can automatically set its network address through DHCP.
4. These menus are available only when a USB memory is inserted correctly to the instrument.

**Connecting to a GPIB Network.** The GPIB Interface can be used for remotely controlling the arbitrary/function generator from an external device (such as a PC). Refer to *Syntax and Commands* on page 3-1 for information about remote control commands.

**Connecting to a Ethernet Network.** To connect the arbitrary/function generator to a LAN, use a 10 Base-T or 100 Base-T cable. Connect a LAN cable to the LAN port on the rear panel. See page 1-19 for the location of the LAN port.

**Synchronous Operation.** By synchronizing multiple arbitrary/function generators, you can configure the multichannel arbitrary/function generator. To synchronize multiple arbitrary/function generators, connect the instruments as follows:

- Use a BNC cable to connect the front-panel Trigger Output connector of the master unit and the Trigger Input connector of the slave unit.
- Use another BNC cable to connect the EXT REF OUTPUT connector on the rear panel of the master unit and the EXT REF INPUT connector of the slave unit.

Push the front-panel **Utility** button. Select **System > Trigger Out** bezel menu, and then select **Sync**.

---

**NOTE.** The EXT REF OUTPUT connector is not provided with the AFG3021 and AFG3022 arbitrary/function generator.

---

**Power-on Setting.** You can select the arbitrary/function generator settings that are restored when the instrument is powered on. By default, the arbitrary/function generator is set to restore the default settings at power-on. To change the power-on settings, use the **Utility > System > Power On** bezel menu. To restore the last powered-off settings next time you power on the instrument, select **Last**.

**Secure.** The **Secure** function erases all the waveform/setup data in the internal memory. This is sometimes useful when you are storing data that is confidential and you must transport the instrument for servicing or demonstrations.



---

**CAUTION.** Executing **Secure** will erase all setups and waveform data files in the internal memory. Make sure you want to remove all data before execution. You cannot recover the deleted files.

---

**Diagnostics and Calibration.** From the Utility menu, you can perform the self test and self calibration routine.

The self test can be performed by selecting the **Utility > -more- > Diagnostics/Calibration > Execute Diagnostics** bezel menu. The test checks whether the instrument is operating correctly. These diagnostics can be helpful when repairing the instrument.

The self calibration allows the instrument to operate with correct precision.

---

**NOTE.** The self calibration must be executed at an ambient temperature of between +20 °C and +30 °C (+68 °F to +86 °F), when a warm-up period of approximately 20 minutes has passed.

---



---

**CAUTION.** Do not power off the instrument during self calibration.

---

**Backup/Restore.** The **Backup** function is provided for copying the current arbitrary/function generator information (instrument setups or user-defined waveforms) to a USB memory with a single button push. Pushing the **Backup Internal -> USB Memory** button will copy the internal memory contents to a USB memory.

The **Restore** function is provided for copying the contents of USB memory to the internal memory.

**Firmware Update.** You can use the Utility menu to update your arbitrary/function generator instrument firmware. If Tektronix offers a newer version of firmware, download the firmware file to your USB memory from the Tektronix Web site.

1. Go to [www.tektronix.com](http://www.tektronix.com).
2. Download a newer version of firmware to your USB memory.
3. Connect the USB memory to your arbitrary/function generator.
4. Push the front-panel **Utility** button.
5. Push the **-more-** bezel button twice, and then select **Firmware Update**.
6. Select the downloaded firmware file and then push the **Execute** bezel button.

---

**NOTE.** To check your instrument firmware version, push the front-panel **Utility** button. The version information is displayed on the screen.

---



---

**CAUTION.** Updating your instrument firmware is a sensitive operation, which may damage your instrument if you do not follow the instructions carefully.

To prevent damage to the instrument, do not remove the USB memory or power off the instrument during the update process.

---

# File Operations

This section describes the basic file operations such as browsing files and saving or recalling the instrument setups or waveform data. The instrument setups or the waveform data can be saved in or recalled from files in the internal memory or a USB memory.

## Browsing Waveform Files

To browse for a waveform file in the internal memory or in a USB memory, use the **Arb Waveform** Menu. You can select either **Internal** or **USB** as a memory type. Figure 2-10 is a screen display example when **USB** is selected.

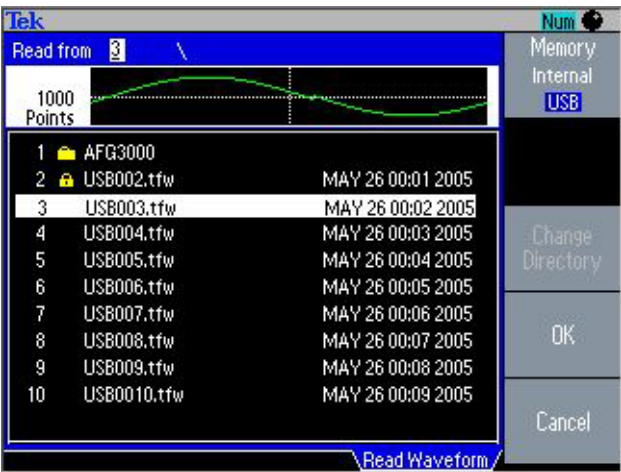


Figure 2-10: Browse waveform files (USB memory)

You can select a waveform to be output. The specified waveform is output if the output is enabled.

**USB Memory.** When you select **USB** from the **Memory** bezel menu, a directory of folder and files on the USB memory are displayed on the screen. You can select a folder or file using the general purpose knob to scroll up and down the rows of the list. The directory of contents are listed in alphabetical order on the screen.

The arbitrary/function generator displays the directory with an index number (the instrument numbers the directory contents from 1 to 500 files). You can also enter an item index number using the numeric keypad to get close to the item you want.

The currently selected item is highlighted. You can open a directory by selecting that row and then pushing the **Change Directory** bezel button. To open a file, select that row and then push **OK**.

**Browsing Waveform Files using Edit Menu.** You can also browse through waveforms using the **Edit** Menu. Select a file you want to edit and then push the **Read** bezel button. The specified waveform is loaded to the Edit Memory. You can now edit the waveform.

**Edit Memory.** The currently edited waveform is written to the Edit Memory. Written data in Edit Memory will be lost if the instrument is powered off. To allow this waveform to be used the next time the instrument is turned on, the Edit Memory waveform must be saved into the internal or a USB memory. To save a waveform file to the internal or USB memory, use the **Write to...** submenu.

### **Saving a Waveform File**

Use the **Edit** Menu to save waveform data as a file in the internal memory or a USB memory. To save a waveform data file, do the following:

1. Attach a USB memory to the front-panel USB connector.
2. Push the front-panel **Edit** button to display the Edit menu.
3. Push the **-more-** bezel button to display the second page, and then push the **Write to...** bezel button.
4. Select a memory location:
  - To save a waveform data file to the internal memory, select **Internal**.
  - To save a waveform data file to a USB memory, select **USB**.
5. Use the general purpose knob to scroll a file or directory.
  - Select **Write** to overwrite the existing file.
  - Select **Write As** to display the submenu to enter a file name. (USB only)
6. Select **Make Directory** to create a new directory. (USB only)
7. To lock a file or directory, select a file or directory and then push the **Lock/Unlock** bezel button. Once a file is locked, a lock icon appears next to the file name.

---

**NOTE.** When you lock a directory, a lock icon does not appear. Once you lock a directory, you cannot create a new file, new directory, or delete existing files.

---

To unlock a file or directory, select a locked file or directory and then push the **Lock/Unlock** bezel button. If the lock icon disappears, it indicates that the file is unlocked.

To delete a file or directory, select a file or directory and then push the **Erase** bezel button.

**Entering a File Name.** When you select **Write As** from the **Write to...** submenu, the character entry box is displayed as shown in Figure 2-11.



**Figure 2-11: Write As - Character entry box**

English alphabet characters are displayed below the file name box. A selected character is highlighted. You can move the highlighted selection using the general purpose knob. To enter the desired character, push the **Enter Character** bezel button or the front-panel **Enter** button. The selected character is displayed in the file name box.

- To move the cursor in the file name box, use the arrow keys.
- To delete a character in the file name box, push the front-panel **BKSP** key.
- To insert a character in the file name box, move the cursor using the arrow keys, then enter a character.

When the filename is specified, push the **OK** bezel button. To cancel the entry, push the front-panel **Cancel** button.

If you select **Make Directory**, you can create a new directory. The created directory name will be listed. The file listings are in alphabetical order. The specified filename will be highlighted.

---

**NOTE.** File names are displayed only in English characters regardless of current instrument language selection. If you use non-English characters to name a file, these characters are replaced by Roman symbols such as #, \$, % on the display screen.

---

## Browsing Instrument Setup Files

To browse instrument setup files in the internal memory or in a USB memory, push the front-panel **Recall** button, and then push the **Memory** bezel button to specify **Internal** or **USB** as a memory type. Figure 2-12 is an example of screen when **Internal** is selected.

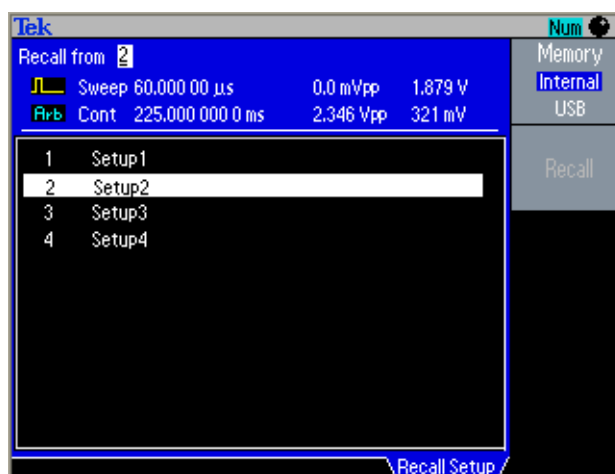


Figure 2-12: Recall menu (Internal)

Select a desired setup file, and then push **Recall**. The selected setup file is recalled.

## Saving an Instrument Setup File

Use the front-panel **Save** button to save instrument setups in the internal or in a USB memory.

1. Push the front-panel **Save** button to display the Save menu.
2. Select a memory location:
  - To save a setup to the internal memory, select **Internal**.
  - To save a setup to a USB memory, select **USB**.
3. Use the general purpose knob to scroll a file or directory.
  - Select **Save** to overwrite the existing file.
  - Select **Save As** to display the submenu to enter a file name. (USB only)
  - Select **Make Directory** to create a new directory. (USB only)

**Entering a File Name.** When you select **Save As**, the file name box is displayed as shown in Figure 2-13.



**Figure 2-13: Save As - Character entry box**

English alphabet characters are displayed below the file name box. A selected character is highlighted. You can move the highlighted selection using the general purpose knob. To enter the desired character, push the **Enter Character** bezel button or the front-panel **Enter** button. The selected character is displayed in the file name box.

- To move the cursor in the file name box, use the arrow keys.
- To delete a character in the file name box, push the front-panel **BKSP** key.
- To insert a character in the file name box, move the cursor using the arrow keys, then enter a character.

When the filename is specified, push the **OK** bezel button. To cancel the entry, push the front-panel **Cancel** button.

If you select **Make Directory**, you can create a new directory. The created directory name will be listed. The file listings are in alphabetical order. The specified filename will be highlighted.

## File Management Conventions

The arbitrary/function generator checks for available space on the USB memory before saving files, and displays a warning message if there is not enough memory for the files you are saving.

The arbitrary/function generator does not have the internal timer. If you create a file using the Edit menu of the instrument, the date and time information is displayed like "TEKAFG001.TFW JAN 01 00:00 1980".

If a file is created by other tools such as ArbExpress software, the date and time information is displayed on the arbitrary/function generator screen.





# Syntax and Commands



# Syntax and Commands

This section provides the following information:

- *Command Syntax* defines the command syntax and processing conventions.
- *Command Groups* describes command groups which lists the commands by function.
- *Command Descriptions* describes the notation of each of the commands in alphabetical order.

## Command Syntax

You can control the operations and functions of the arbitrary/function generator through the GPIB interface using commands and queries. The related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the instrument uses to process them. See the *Command Groups* on page 3-9 for a listing of the commands by command group, or use the index to locate a specific command.

**Backus-Naur Form Notation.** This manual describes the commands and queries using Backus-Naur Form (BNF) notation. Refer to Table 3-1 for the symbols that are used.

**Table 3-1: BNF notation**

Symbol	Meaning
< >	Defined element
::=	Is defined as
	Exclusive OR
{ }	Group; one element is required
[ ]	Optional; can be omitted
...	Previous element(s) may be repeated
( )	Comment

**Command and Query Structure**

Commands consist of set commands and query commands (usually simply called commands and queries). Commands change instrument settings or perform a specific action. Queries cause the instrument to return data and information about its status.

Most commands have both a set form and a query form. The query form of the command is the same as the set form except that it ends with a question mark. For example, the set command `DISPlay:CONTRast` has a query form `DISPlay:CONTRast?`. Not all commands have both a set and a query form; some commands are set only and some are query only.

A few commands do both a set and query action. For example, the `*CAL?` command runs a self-calibration program on the instrument, then returns the result of the calibration.

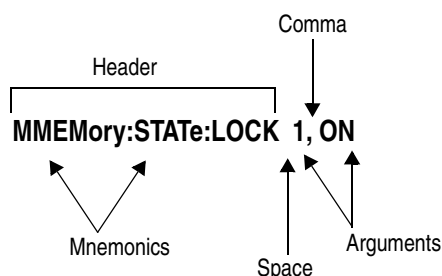
A command message is a command or query name, followed by any information the instrument needs to execute the command or query. Command messages consist of five element types.

Table 3-2 lists and describes the five different element types.

**Table 3-2: Command message elements**

Symbol	Meaning
<Header>	The basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character; if the command is concatenated with other commands the beginning colon is required. The beginning colon can never be used with command headers beginning with a star (*).
<Mnemonic>	A header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, they are always separated from each other by a colon (:) character.
<Argument>	A quantity, quality, restriction, or limit associated with the header. Not all commands have an argument, while other commands have multiple arguments. Arguments are separated from the header by a <Space>. Arguments are separated from each other by a <Comma>.
<Comma>	A single comma between arguments of multiple-argument commands. It may optionally have white space characters before and after the comma.
<Space>	A white space character between command header and argument. It may optionally consist of multiple white space characters.

Figure 3-1 shows the five command message elements.



**Figure 3-1: Command message elements**

**Commands.** Commands cause the instrument to perform a specific function or change one of its settings. Commands have the structure:

[:]<Header>[<Space><Argument>[<Comma><Argument>]...]

A command header is made up of one or more mnemonics arranged in a hierarchical or tree structure. The first mnemonic is the base or root of the tree and each subsequent mnemonic is a level or branch of the previous one. Commands at a higher level in the tree may affect those at a lower level. The leading colon (:) always returns you to the base of the command tree.

**Queries.** Queries cause the arbitrary/function generator to return information about its status or settings. Queries have the structure:

[:]<Header>?

[:]<Header>?[<Space><Argument>[<Comma><Argument>]...]

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level.

**Query Responses.** When a query is sent to the arbitrary/function generator, only the values are returned. When the returned value is a mnemonic, it is noted in abbreviated format, as shown in Table 3-3.

**Table 3-3: Query response examples**

Query	Response
SOURce:PULSe:DCYcle?	50.0
OUTPut:POLarity?	NORM

**Command Entry**

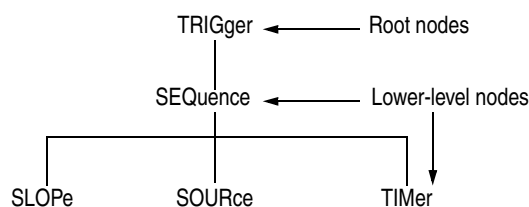
Follow these general rules when entering commands:

- Enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).
- The instrument ignores commands that consists of just a combination of white space characters and line feeds.

**SCPI Commands and Queries**

The arbitrary/function generator uses a command language based on the SCPI standard. The SCPI (Standard Commands for Programmable Instruments) standard was created by a consortium to provide guidelines for remote programming of instruments. These guidelines provide a consistent programming environment for instrument control and data transfer. This environment uses defined programming messages, instrument responses and data formats that operate across all SCPI instruments, regardless of manufacturer.

The SCPI language is based on a hierarchical or tree structure that represents a subsystem (see Figure 3-2). The top level of the tree is the root node; it is followed by one or more lower-level nodes.



**Figure 3-2: Example of SCPI subsystem hierarchy tree**

You can create commands and queries from these subsystem hierarchy trees. Commands specify actions for the instrument to perform. Queries return measurement data and information about parameter settings.

## Creating Commands

SCPI commands are created by stringing together the nodes of a subsystem hierarchy and separating each node by a colon.

In Figure 3-2, TRIGger is the root node and SEquence, SLOPe, SOURce, and TIMer are lower level nodes. To create an SCPI command, start with the root node TRIGger and move down the tree structure adding nodes until you reach the end of a branch. Most commands and some queries have parameters; you must include a value for these parameters. The command descriptions, which begin on page 3-15, list the valid values for all parameters.

For example, TRIGger:SEquence:SOURce EXternal is a valid SCPI command created from the hierarchy tree in Figure 3-2.

## Parameter Types

Parameters are indicated by angle brackets, such as <file\_name>. There are several different types of parameters, as listed in Table 3-4. The parameter type is listed after the parameter. Some parameter types are defined specifically for the arbitrary/function generator command set and some are defined by SCPI.

**Table 3-4: Parameter types used in syntax descriptions**

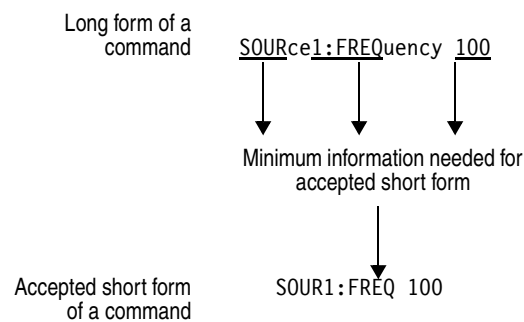
Parameter type	Description	Example
arbitrary block	A block of data bytes	#512234xxxx... where 5 indicates that the following 5 digits (12234) specify the length of the data in bytes; xxxxx... indicates the data  or  #0xxxxx...<LF><&EOI>
boolean	Boolean numbers or values	ON or $\neq$ 0 OFF or 0
discrete	A list of specific values	MIN, MAX
binary	Binary numbers	#B0110
octal	Octal numbers	#Q75, #Q3
hexadecimal	Hexadecimal numbers (0-9, A-F)	#HAA, #H1
NR1 numeric	Integers	0, 1, 15, -1
NR2 numeric	Decimal numbers	1.2, 3.141516, -6.5
NR3 numeric	Floating point numbers	3.1415E-9, -16.1E5
NRf numeric	Flexible decimal number that may be type NR1, NR2, or NR3	See NR1, NR2, NR3 examples in this table
string	Alphanumeric characters (must be within quotation marks)	"Testing 1, 2, 3"

### Special Characters

The Line Feed (LF) character or the New Line (NL) character (ASCII 10), and all characters in the range of ASCII 127-255 are defined as special characters. These characters are used in arbitrary block arguments only; using these characters in other parts of any command yields unpredictable results.

### Abbreviating Commands, Queries, and Parameters

You can abbreviate most SCPI commands, queries, and parameters to an accepted short form. This manual shows these commands as a combination of upper and lower case letters. The upper case letters indicate the accepted short form of a command, as shown in Figure 3-3. The accepted short form and the long form are equivalent and request the same action of the instrument.



**Figure 3-3: Example of abbreviating a command**

---

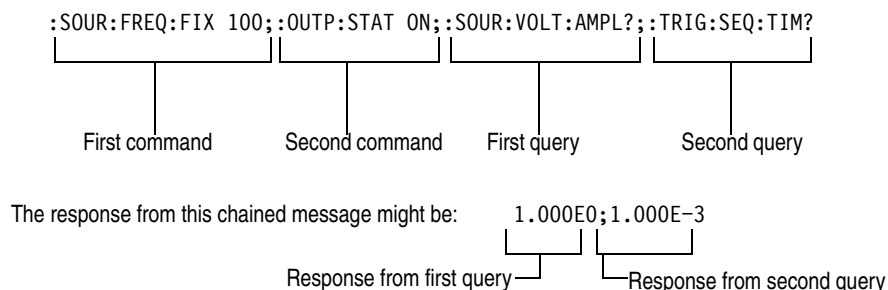
**NOTE.** The numeric suffix of a command or query may be included in either the long form or short form. The arbitrary/function generator will default to “1” if no suffix is used.

---



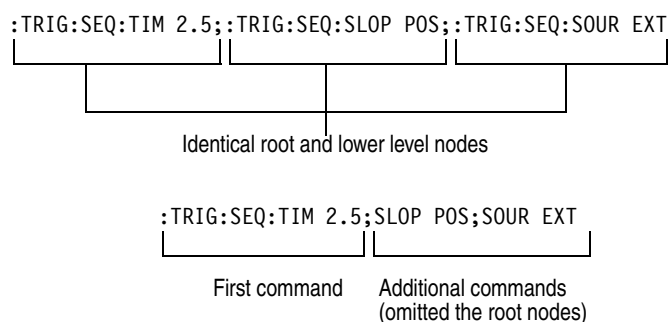
## Chaining Commands and Queries

You can chain several commands or queries together into a single message. To create a chained message, first create a command or query, then add a semicolon (;), and finally add more commands or queries and semicolons until you are done. If the command following a semicolon is a root node, precede it with a colon (:). Figure 3-4 illustrates a chained message consisting of several commands and queries. The chained message should end in a command or query, not a semicolon. Responses to any queries in your message are separated by semicolons.



**Figure 3-4: Example of chaining commands and queries**

If a command or query has the same root and lower-level nodes as the previous command or query, you can omit these nodes. In Figure 3-5, the second command has the same root node (SEquence) as the first command, so these nodes can be omitted.



**Figure 3-5: Example of omitting root and lower level nodes**

## Units and SI Prefix

If the decimal numeric argument refers to voltage, frequency, impedance, or time, you can express it using SI units instead of using the scaled explicit point input value format <NR3>. (SI units are units that conform to the System International d'Unites standard.) For example, you can use the input format 200 mV or 1.0 MHz instead of 200.0E-3 or 1.0E+6, respectively, to specify voltage or frequency.

## General Rules for Using SCPI commands

The following are three general rules for using SCPI commands, queries, and parameters:

- You can use single ( ' ') or double ( " ") quotation marks for quoted strings, but you cannot use both types of quotation marks for the same string.
  - correct      "This string uses quotation marks correctly."
  - correct      'This string also uses quotation marks correctly.'
  - incorrect    "This string does not use quotation marks correctly.'
- You can use upper case, lower case, or a mixture of both cases for all commands, queries, and parameters.
  - :SOURCE:FREQUENCY 10MHZ
  - is the same as
  - :source:frequency 100mhz
  - and
  - :SOURCE:frequency 10MHZ

---

**NOTE.** *Quoted strings are case sensitive.*

---

- No embedded spaces are allowed between or within nodes.
  - correct      :OUTPUT:FILTER:LPASS:FREQUENCY 200MHZ
  - incorrect    :OUTPUT: FILTER: LPASS:FREQUENCY 200MHZ

## Command Groups

This section lists the commands organized by functional group. The Command Descriptions section, starting on page 3-15, lists all commands alphabetically.

**Calibration and Diagnostic Commands.** Calibration and Diagnostic commands let you initiate the instrument self-calibration routines and examine the results of diagnostic tests. Table 3-5 lists Calibration and Diagnostic commands.

**Table 3-5: Calibration and Diagnostic commands**

Header	Description
*CAL?	Perform self-calibration and return result status
CALibration[:ALL]	Perform self-calibration
DIAGnostic[:ALL]	Perform self-test
*TST?	Perform self-test and return result status

**Display Commands.** Display commands let you change the graticule style, displayed contrast, and other display attributes. Table 3-6 lists and describes Display commands.

**Table 3-6: Display commands**

Header	Description
DISPlay:CONTrast	Set/query the LCD display contrast
DISPlay:SAVer[:STATe]	Set/query the screen saver settings
DISPlay[:WINDow]:TEXT[:DATA]	Set/query the text message display
DISPlay[:WINDow]:TEXT:CLear	Delete text message

**Memory Commands.** Memory commands let you change setup memory attributes. Table 3-7 lists and describes Memory commands.

**Table 3-7: Memory commands**

Header	Description
MEMory:STATe:VALid?	Query the availability of setup memory
MEMory:STATe:DELeTe	Delete the setup memory
MEMory:STATe:LOCK	Set/query the lock of setup memory overwrite and deletion
MEMory:STATe:RECall:AUTo	Set/query the recall of last set memory
*RCL	Recall instrument setting from setup memory
*SAV	Save instrument setting to setup memory

**Mass Memory Commands.** Mass Memory commands let you change mass memory attributes. Table 3-8 lists and describes Mass Memory commands.

**Table 3-8: Mass Memory commands**

Header	Description
MMEMory:CATalog?	Query the status of mass memory
MMEMory:CDIRectory	Set/query current directory
MMEMory:DELeTe	Delete file or directory in mass memory
MMEMory:LOAD:STATe	Copy instrument setting in mass memory to setup memory
MMEMory:LOAD:TRACe	Copy waveform data file in mass memory to edit memory
MMEMory:LOCK[:STATe]	Set/query the lock of mass memory overwrite and deletion
MMEMory:MDIRectory	Create directory in mass memory
MMEMory:STORe:STATe	Save the setup memory status to mass memory
MMEMory:STORe:TRACe	Save waveform data file in edit memory to mass memory

**Output Commands.** Output commands let you set output attributes. Table 3-9 lists and describes Output commands.

**Table 3-9: Output commands**

Header	Description
OUTPut[1 2]:IMPedance	Set/query impedance
OUTPut[1 2]:POLarity	Set/query polarity
OUTPut[1 2][:STATe]	Set/query output on or off
OUTPut:TRIGger:MODE	Set/query the mode of Trigger Output

**Source Commands.** Source commands let you set waveform output parameters. Table 3-10 lists and describes Source commands.

**Table 3-10: Source commands**

Header	Description
[SOURce]:ROSCillator:SOURce	Set/query clock reference input
[SOURce[1 2]]:AM:STATe	Set/query amplitude modulation status
[SOURce[1 2]]:AM:INTernal:FREQuency	Set/query internal modulation frequency
[SOURce[1 2]]:AM:INTernal:FUNCTion	Set/query modulation waveform setting
[SOURce[1 2]]:AM:INTernal:FUNCTion:EFILe	Set/query EFILe setting
[SOURce[1 2]]:AM:SOURce	Set/query amplitude modulation source
[SOURce[1 2]]:AM[:DEPT]h	Set/query amplitude modulation depth
[SOURce[1 2]]:BURSt:MODE	Set/query burst mode
[SOURce[1 2]]:BURSt:NCYCles	Set/query burst mode waveform output cycle
[SOURce[1 2]]:BURSt:TDELay	Set/query burst mode trigger delay time

Table 3-10: Source commands (cont.)

Header	Description
[SOURce[1 2]]:BURSt [:STATe]	Set/query burst mode status
[SOURce[1 2]]:COMBine:FEED	Set/query internal noise or external signal
[SOURce[1 2]]:FM:INTernal:FREQuency	Set/query internal modulation frequency
[SOURce[1 2]]:FM:INTernal:FUNCTion	Set/query internal modulation waveform
[SOURce[1 2]]:FM:INTernal:FUNCTion:EFILe	Set/query EFILe setting
[SOURce[1 2]]:FM:SOURce	Set/query frequency modulation source
[SOURce[1 2]]:FM:STATe	Set/query frequency modulation status
[SOURce[1 2]]:FM [:DEViation]	Set/query frequency deviation
[SOURce[1 2]]:FREQuency:CENTer	Set/query center frequency
[SOURce[1 2]]:FREQuency:CONCurent[:STATe]	Set/query concurrent change of frequency
[SOURce[1 2]]:FREQuency:MODE	Set/query sweep status
[SOURce[1 2]]:FREQuency:SPAN	Set/query sweep frequency span
[SOURce[1 2]]:FREQuency:STARt	Set/query sweep start frequency
[SOURce[1 2]]:FREQuency:STOP	Set/query sweep stop frequency
[SOURce[1 2]]:FREQuency [:CW]:FIXed]	Set/query output waveform frequency
[SOURce[1 2]]:FSKey:INTernal:RATE	Set/query FSK internal modulation rate
[SOURce[1 2]]:FSKey:SOURce	Set/query FSK source
[SOURce[1 2]]:FSKey:STATe	Set/query FSK status
[SOURce[1 2]]:FSKey [:FREQuency]	Set/query FSK hop frequency
[SOURce[1 2]]:FUNCTion:RAMP:SYMMetry	Set/query ramp waveform symmetry
[SOURce[1 2]]:FUNCTion [:SHAPE]	Set/query output waveform
[SOURce[1 2]]:FUNCTion:EFILe	Set/query EFILe name
[SOURce[1 2]]:PHASe:INITiate	Initiate output waveform phase synchronization
[SOURce[1 2]]:PHASe[:ADJust]	Set/query output waveform phase
[SOURce[1 2]]:PM:INTernal:FREQuency	Set/query internal modulation frequency
[SOURce[1 2]]:PM:INTernal:FUNCTion	Set/query internal modulation waveform
[SOURce[1 2]]:PM:INTernal:FUNCTion:EFILe	Set/query EFILe name
[SOURce[1 2]]:PM:SOURce	Set/query phase modulation source
[SOURce[1 2]]:PM:STATe	Set/query phase modulation status
[SOURce[1 2]]:PM [:DEViation]	Set/query phase modulation deviation
SOURce<3 4>:POWer [:LEVel][:IMMEDIATE] [:AMPLitude]	Set/query internal noise level
[SOURce[1 2]]:PULSe:DCYCLE	Set/query pulse waveform duty cycle
[SOURce[1 2]]:PULSe:DELay	Set/query pulse waveform lead delay
[SOURce[1 2]]:PULSe:HOLD	Set/query pulse waveform parameter
[SOURce[1 2]]:PULSe:PERiod	Set/query pulse waveform period
[SOURce[1 2]]:PULSe:TRANSition[:LEADing]	Set/query pulse waveform leading edge time
[SOURce[1 2]]:PULSe:TRANSition:TRAILing	Set/query pulse waveform trailing edge time
[SOURce[1 2]]:PULSe:WIDTh	Set/query pulse waveform width

**Table 3-10: Source commands (cont.)**

Header	Description
[SOURce[1 2]]:PWM:INTernal:FREQuency	Set/query pulse width modulation frequency
[SOURce[1 2]]:PWM:INTernal:FUNCTion	Set/query pulse width modulation waveform
[SOURce[1 2]]:PWM:INTernal:FUNCTion:EFILe	Set/query EFILE name
[SOURce[1 2]]:PWM:SOURce	Set/query pulse width modulation source
[SOURce[1 2]]:PWM:STATe	Set/query pulse width modulation status
[SOURce[1 2]]:PWM[:DEViation]:DCYCLe	Set/query pulse width modulation deviation
[SOURce[1 2]]:SWEep:HTIME	Set/query sweep hold time
[SOURce[1 2]]:SWEep:MODE	Set/query sweep mode
[SOURce[1 2]]:SWEep:RTIME	Set/query sweep return time
[SOURce[1 2]]:SWEep:SPACing	Set/query sweep spacing
[SOURce[1 2]]:SWEep:TIME	Set/query sweep time
[SOURce[1 2]]:VOLTage:CONCurent[:STATe]	Set/query concurrent change of amplitude level
[SOURce[1 2]]:VOLTage:LIMit:HIGH	Set/query output amplitude upper limit
[SOURce[1 2]]:VOLTage:LIMit:LOW	Set/query output amplitude lower limit
[SOURce[1 2]]:VOLTage:UNIT	Set/query output amplitude units
[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate]:HIGH	Set/query output amplitude high level
[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate]:LOW	Set/query output amplitude low level
[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate]:OFFSet	Set/query output offset voltage
[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]	Set/query output amplitude

**Status Commands.** Status commands let you determine the status of the instrument. Table 3-11 lists and describes Status commands.

**Table 3-11: Status commands**

Header	Description
*CLS	Clear all event registers and queues
*ESE	Set/query standard event status enable register
*ESR?	Return standard event status register
*PSC	Set/query power-on status clear
*SRE	Set/query service request enable register
*STB?	Read status byte
STATus:OPERation[:EVENT]?	Return operation event register
STATus:OPERation:CONDition?	Return operation condition register
STATus:OPERation:ENABLE	Set/query operation enable register
STATus:PRESet	Preset SCPI enable register

**Table 3-11: Status commands (cont.)**

Header	Description
STATus:QUESTionable[:EVENT]?	Return questionable event register
STATus:QUESTionable:CONDition?	Return questionable condition register
STATus:QUESTionable:ENABLE	Set/query questionable enable register

**System Commands.** System commands let you control miscellaneous instrument functions. Table 3-12 lists and describes System commands.

**Table 3-12: System commands**

Header	Description
*IDN?	Return identification information
*OPT?	Return option information
*RST	Reset
SYSTem:BEEPer:STATe	Set/query beeper state
SYSTem:BEEPer[:IMMEDIATE]	Generate an audible tone
SYSTem:ERRor[:NEXT]?	Return error event queue
SYSTem:KCLick[:STATe]	Set/query click sound
SYSTem:KLOCK[:STATe]	Set/query front panel lock/unlock
SYSTem:SECurity:IMMEDIATE	Reset to factory default
SYSTem:ULANguage	Set/query language for display screen
SYSTem:VERSion?	Return version information

**Synchronization Commands.** Synchronization commands let you synchronize the operation of the instrument. Table 3-13 lists and describes Synchronization commands.

**Table 3-13: Synchronization commands**

Header	Description
*OPC	Set/query operation complete
*WAI	Wait to continue

**Trace Commands.** Trace commands let you set the edit memory and user waveform memory. Table 3-14 lists and describes Trace commands.

**Table 3-14: Trace commands**

Header	Description
TRACe DATA:CATalog?	Return user waveform memory status
TRACe DATA:COpy	Copy edit memory (or user waveform memory) content to user waveform memory (or edit memory)
TRACe DATA[:DATA]	Set/query waveform data to edit memory
TRACe DATA:DELeTe[:NAME]	Delete user waveform memory contents
TRACe DATA:DEFine	Set edit memory content
TRACe DATA:LOCK[:STATe]	Set/query lock/unlock of user waveform memory
TRACe DATA:POINts	Set/query number of points for waveform data in edit memory
TRACe DATA[:DATA]:LINE	Write waveform data with interpolation
TRACe DATA[:DATA]:VALue	Set/query waveform data in edit memory

**Trigger Commands.** Trigger commands let you control all aspects of arbitrary/function generator triggering. Table 3-15 lists and describes Trigger commands.

**Table 3-15: Trigger commands**

Header	Description
ABORt	Initialize trigger system
*TRG	Force trigger event
TRIGger[:SEQuence]:SLOPe	Set/query the slope of trigger signal
TRIGger[:SEQuence]:SOURce	Set/query the source of trigger signal
TRIGger[:SEQuence]:TIMer	Set/query the period of internal clock
TRIGger[:SEQuence] [:IMMediate]	Generate a trigger event

**AFG Control.** AFG Control command copies setups between two channels.

**Table 3-16: AFG Control command**

Header	Description
AFGControl:CSCopy	Copy CH1 (or CH2) setup parameters to CH2 (or CH1)



## Command Descriptions

Commands either set or query instrument values. Some commands both set and query, some only set, and some only query.

### Manual Conventions

This manual uses the following conventions:

- No Query Form indicates set-only commands
- A question mark (?) appended to the commands and Query Only indicates query-only commands
- Fully spells out headers, mnemonics, and arguments with the minimal spelling shown in upper case; for example, to use the abbreviated form of the DISPLAY:CONTRast command, just type DISP:CONT
- Syntax of some commands varies, depending on the model of arbitrary/function generator you are using; differences are noted

## ABORt (No Query Form)

Initializes all the current trigger system parameters and resets all trigger sequences.

**Group** Trigger

**Syntax** ABORt

**Arguments** None

**Examples** ABORt  
resets the trigger system.

## AFGControl:CSCopy (No Query Form)

This command copies setup parameters for one channel to another channel.

If your arbitrary/function generator is a single-channel model, this command is not supported.

<b>Group</b>	AFG Control
<b>Syntax</b>	AFGControl:CSCopy {CH1 CH2},{CH1 CH2}
<b>Arguments</b>	CH1 CH2
<b>Examples</b>	AFGControl:CSCopy CH1,CH2 copies the CH1 setup parameters into CH2.

## \*CAL?

This command performs an internal calibration and returns 0 (Pass) or a calibration error code.

---

**NOTE.** The self-calibration can take several minutes to complete. During this time, the arbitrary/function generator does not execute any commands. Do not power off the instrument during the self-calibration.

---

<b>Group</b>	Calibration and Diagnostic
<b>Related Commands</b>	CALibration[:ALL]?
<b>Syntax</b>	*CAL?
<b>Arguments</b>	None
<b>Returns</b>	<NR1> where <NR1>=0 indicates that the internal calibration completed without errors.  <NR1>≠0 indicates that the arbitrary/function generator detected an error.
<b>Examples</b>	*CAL? performs an internal calibration and returns results. For example, it might return 0, which indicates that the calibration completed without any errors.

## CALibration[:ALL]

The CALibration[:ALL] command performs an internal calibration.

The CALibration[:ALL]? command performs an internal calibration and returns 0 (Pass) or a calibration error code.

---

**NOTE.** The self-calibration can take several minutes to complete. During this time, the arbitrary/function generator does not execute any commands. Do not power off the instrument during the self-calibration.

---

<b>Group</b>	Calibration and Diagnostic
<b>Related Commands</b>	*CAL?
<b>Syntax</b>	CALibration[:ALL]  CALibration[:ALL]?
<b>Arguments</b>	None
<b>Returns</b>	<NR1>
<b>Examples</b>	CALibration:ALL performs an internal calibration.  CALibration:ALL? performs an internal calibration and returns results. For example, it might return 0, which indicates that the calibration completed without any errors.

## \*CLS (No Query Form)

This command clears all the event registers and queues, which are used in the arbitrary/function generator status and event reporting system.

<b>Group</b>	Status
<b>Syntax</b>	*CLS
<b>Arguments</b>	None
<b>Examples</b>	*CLS clears all the event registers and queues.

## DIAGnostic[:ALL]

The `DIAGnostic[:ALL]` command performs a self-test. The `DIAGnostic[:ALL]?` command returns the results after executing the test.

---

**NOTE.** *The self-test can take several minutes to complete. During this time, the arbitrary/function generator does not execute any commands. Do not power off the instrument during the self-test.*

---

<b>Group</b>	Calibration and Diagnostic
<b>Related Commands</b>	*TST?
<b>Syntax</b>	<code>DIAGnostic[:ALL]</code>  <code>DIAGnostic[:ALL]?</code>
<b>Arguments</b>	None
<b>Returns</b>	<code>&lt;NR1&gt;</code> where <code>&lt;NR1&gt;=0</code> indicates that the self-test completed without errors.  <code>&lt;NR1&gt;≠0</code> indicates that the arbitrary/function generator detected an error.
<b>Examples</b>	<code>DIAGnostic:ALL</code> performs self-test.  <code>DIAGnostic:ALL?</code> performs self-test and might return an error code.

## DISPlay:CONTRast

This command sets or queries the contrast of the LCD display.

**Group** Display

**Syntax** DISPlay:CONTRast {<contrast>|MINimum|MAXimum}  
DISPlay:CONTRast? [MINimum|MAXimum]

**Arguments** <contrast>::=<NR2>  
where  
<NR2> is a range of display contrast from 0 through 1. The larger the value, the greater the screen contrast.

MINimum sets the display to the 0 contrast level.

MAXimum sets the display to the largest contrast level.

**Returns** <NR2>

**Examples** DISPlay:CONTRast MAXimum  
sets the display contrast to the largest contrast level.

## DISPlay:SAVer[:STATe]

This command sets or queries the screen saver setting of the LCD display. When enabled, the screen saver function automatically starts if no operations are applied to the instrument front panel for five minutes.

**Group** Display

**Syntax** DISPlay:SAVer[:STATe] {ON|OFF|<NR1>}  
DISPlay:SAVer[:STATe]?

**Arguments** ON or <NR1>≠0 enables the screen saver function.  
OFF or <NR1>=0 disables the screen saver function.

**Returns** <NR1>

**Examples** DISPlay:SAVer:STATe OFF  
disables the screen saver function.

## DISPlay[:WINDow]:TEXT[:DATA]

The DISPlay[:WINDow]:TEXT[:DATA] command displays a text message on the instrument screen.

The DISPlay[:WINDow]:TEXT[:DATA]? command returns a text string currently displayed on the instrument screen.

The displayable characters are ASCII code 32 through 126, and the instrument can display approximately 64 characters.

<b>Group</b>	Display
<b>Syntax</b>	DISPlay[:WINDow]:TEXT[:DATA] <string> DISPlay[:WINDow]:TEXT[:DATA]?
<b>Arguments</b>	<string>
<b>Returns</b>	<string>  A currently displayed text message is returned.
<b>Examples</b>	DISPlay:WINDow:TEXT:DATA? returns the currently displayed text message.

## DISPlay[:WINDow]:TEXT:CLEAr (No Query Form)

This command clears the text message from the display screen.

<b>Group</b>	Display
<b>Syntax</b>	DISPlay[:WINDow]:TEXT:CLEAr
<b>Arguments</b>	None
<b>Examples</b>	DISPlay:WINDow:TEXT:CLEAr clears the text message from the screen.

**\*ESE**

This command sets or queries the bits in the Event Status Enable Register (ESER) used in the status and events reporting system of the arbitrary/function generator. The query command returns the contents of the ESER.

**Group**      Status

**Related Commands**    \*CLS, \*ESR?, \*PSC, \*SRE, \*STB

**Syntax**      \*ESE <bit\_value>  
  
                 \*ESE?

**Arguments**    <bit\_value>::<<NR1>  
                 where:  
                 <NR1> is a value in the range of 0 through 255. The binary bits of the ESER are set according to this value.

**Returns**      <bit\_value>

**Examples**      \*ESE 177  
                 sets the ESER to 177 (binary 10110001), which sets the PON, CME, EXE and OPC bits.  
  
                 \*ESE?  
                 might return 186, indicating that the ESER contains the binary value 10111010.

**\*ESR?**

This query-only command returns the contents of the Standard Event Status Register (SESR) used in the status events reporting system in the arbitrary/function generator. \*ESR also clears the SESR (since reading the SESR clears it).

<b>Group</b>	Status
<b>Related Commands</b>	*CLS, *ESE?, *SRE, *STB
<b>Syntax</b>	*ESR?
<b>Arguments</b>	None
<b>Returns</b>	<NR1> indicates that the contents of the SESR as a decimal integer.
<b>Examples</b>	*ESR? might return 181, which indicates that the SESR contains the binary number 10110101.

**\*IDN?**

This query-only command returns identification information on the arbitrary/function generator.

<b>Group</b>	System
<b>Syntax</b>	*IDN?
<b>Arguments</b>	None
<b>Returns</b>	<Manufacturer>,<Model>,<Serial Number>,<Firmware Level>  where: <Manufacturer>::=TEKTRONIX <Model>::={AFG3021 AFG3022 AFG3101 AFG3102 AFG3251 AFG3252} <Serial Number> <Firmware Level>::=SCPI:99.0 FV:1.0
<b>Examples</b>	*IDN? might return the following response:  TEKTRONIX,AFG3102,C100101,SCPI:99.0 FV:1.0



## MEMory:STAtE:VALid?

This query-only command returns the availability of a setup memory.

**Group** Memory

**Syntax** MEMory:STAtE:VALid? {0|1|2|3|4}

**Arguments** 0, 1, 2, 3, or 4 specifies the location of setup memory.

**Returns** <NR1>

1 means that the specified setup memory has been saved.

0 means that the specified setup memory has been deleted.

**Examples** MEMory:STAtE:VALid? 0  
might return 1 if the specified setup memory has been saved.

## MEMory:STAtE:DELEte (No Query Form)

This command deletes the contents of specified setup memory. If a specified setup memory is not allowed to overwrite or delete, this command causes an error.

**Group** Memory

**Syntax** MEMory:STAtE:DELEte {0|1|2|3|4}

**Arguments** 0, 1, 2, 3, or 4 specifies the location of setup memory.

**Examples** MEMory:STAtE:DELEte 1  
deletes the contents of specified setup memory.

## MEMory:STATe:LOCK

This command sets or queries whether to lock the specified setup memory. If you lock a setup memory, you cannot overwrite or delete the setup file.

You cannot execute this command for the setup memory of location number 0 (last setup memory).

**Group** Memory

**Syntax** MEMory:STATe:LOCK {1|2|3|4},{ON|OFF|<NR1>}

MEMory:STATe:LOCK?{1|2|3|4}

**Arguments** ON or <NR1>≠0 locks the specified location of setup memory.

OFF or <NR1>=0 allows you to overwrite or delete the specified location of setup memory.

**Returns** <NR1>

**Examples** MEMory:STATe:LOCK 1,ON  
locks the setup memory of location number 1.

## MEMory:STATe:RECall:AUTO

This command sets or queries whether to enable the automatic recall of last setup memory when powered-on. The next time you apply the power, the arbitrary/function generator will automatically recall the settings you used when you powered off the instrument.

If you select OFF, the default setups are recalled when you power on the instrument.

**Group** Memory

**Syntax** MEMory:STATe:RECall:AUTO {ON|OFF|<NR1>}

MEMory:STATe:RECall:AUTO?

**Arguments** ON or <NR1>≠0 enables the recall of the setup memory you last used before powering off the instrument.

OFF or <NR1>=0 disables the last setup recall function.

**Returns** <NR1>

**Examples** MEMory:STATe:RECall:AUTO ON  
sets the instrument to recall the last setup memory when powered-on.

## MMEMory:CATalog?

This query-only command returns the current state of the mass storage system (USB memory).

**Group** Mass Memory

**Related Commands** MMEMory:CDIRectory

**Syntax** MMEMory:CATalog?

**Arguments** None

**Returns** <NR1>,<NR1>[,<file\_name>,<file\_type>,<file\_size>]...  
where:  
The first <NR1> indicates that the total amount of storage currently used, in bytes.  
The second <NR1> indicates that the free space of mass storage, in bytes.  
  
<file\_name> is the exact name of a file.  
<file\_type> is DIR for directory, otherwise it is blank.  
<file\_size> is the size of the file, in bytes.

**Examples** MMEMory:CATalog?  
might return the following response:  
  
32751616,27970560,"SAMPLE1.TFS,,5412"

## MMEMory:CDIRectory

This command changes the current working directory in the mass storage system.

**Group** Mass Memory

**Syntax** MMEMory:CDIRectory [<directory\_name>]

MMEMory:CDIRectory?

**Arguments** <directory\_name>::=<string> indicates that the current working directory for the mass storage system.

If you do not specify a parameter, the directory is set to the \*RST value.  
At \*RST, this parameter is set to the root.

**Returns** <directory\_name>::=<string>

**Examples** MMEMory:CDIRectory "/AFG/WORK0"  
changes the current directory to /AFG/WORK0.

## MMEMory:DELeTe (No Query Form)

This command deletes a file or directory from the mass storage system. If a specified file in the mass storage is not allowed to overwrite or delete, this command causes an error. You can delete a directory if it is empty. You cannot delete waveform data while it is outputting.

**Group** Mass Memory

**Syntax** MMEMory:DELeTe <file\_name>

**Arguments** <file\_name>::=<string> specifies a file to be deleted.

**Examples** MMEMory:DELeTe "TEK001.TFW"  
deletes the specified file from the mass storage.

## MMEMory:LOAD:STAtE (No Query Form)

This command copies a setup file in the mass storage system to an internal setup memory. If a specified internal setup memory is locked, this command causes an error.

When you power off the instrument, the setups are automatically overwritten in the setup memory 0 (last setup memory).

<b>Group</b>	Mass Memory
<b>Related Commands</b>	MEMory:STAtE:LOCK MEMory:STAtE:RECa11:AUTO MMEMory:STORe:STAtE
<b>Syntax</b>	MMEMory:LOAD:STAtE{0 1 2 3 4},<file_name>
<b>Arguments</b>	0, 1, 2, 3, or 4 specifies the location of setup memory. <file_name>::=<string> specifies a setup file to be copied.
<b>Examples</b>	MMEMory:LOAD:STAtE 1,"SETUP1.TFS" copies a file named SETUP1.TFS in the mass storage into the internal memory location 1.

## MMEMory:LOAD:TRACe (No Query Form)

This command copies a waveform data file in the mass storage system to Edit Memory. If the file format is different, this command causes an error.

<b>Group</b>	Mass Memory
<b>Related Commands</b>	MMEMory:STORe:TRACe
<b>Syntax</b>	MMEMory:LOAD:TRACe EMEMory,<file_name>
<b>Arguments</b>	<file_name>::=<string> specifies a waveform data file to be copied.
<b>Examples</b>	MMEMory:LOAD:TRACe EMEMory,"TEK001.TFW" copies a file named TEK001.TFW in the mass storage into Edit Memory.

## MMEMory:LOCK[:STATe]

This command sets or queries whether to lock a file or directory in the mass storage system. If you lock a file or directory, you cannot overwrite or delete it.

**Group** Mass Memory

**Syntax** MMEMory:LOCK[:STATe] <file\_name>, {ON|OFF|<NR1>}

MMEMory:LOCK[:STATe] <file\_name>?

**Arguments** ON or <NR1>≠0 locks a file or directory in the mass storage system.

OFF or <NR1>=0 allows you to overwrite or delete a file or directory in the mass storage system.

**Returns** <NR1>

**Examples** MMEMory:LOCK[:STATe] "SETUP1.TFS",ON  
locks the file "SETUP1.TFS".

## MMEMory:MDIRectory (No Query Form)

This command creates a directory in the mass storage system. If the specified directory is locked in the mass storage system, this command causes an error.

**Group** Mass Memory

**Syntax** MMEMory:MDIRectory <directory\_name>

**Arguments** <directory\_name>::=<string> specifies a directory name to be created.

**Examples** MMEMory:MDIRectory "SAMPLE1"  
creates a directory named "SAMPLE1" in the mass storage system.

## MMEMory:STORe:STATe (No Query Form)

This command copies a setup file in the setup memory to a specified file in the mass storage system. If the specified file in the mass storage system is locked, this command causes an error. You cannot create a new file if the directory is locked. If the setup memory is deleted, this command causes an error. <file\_name> is a quoted string that defines the file name and path.

<b>Group</b>	Mass Memory
<b>Related Commands</b>	MMEMory:LOAD:STATe MMEMory:LOCK[:STATe]
<b>Syntax</b>	MMEMory:STORe:STATe{0 1 2 3 4},<file_name>
<b>Arguments</b>	0, 1, 2, 3, or 4 specifies the location of setup memory. <file_name>::=<string> specifies a file name in the mass storage system. The <file_name> includes path. Path separators are forward slashes (/).
<b>Examples</b>	MMEMory:STORe:STATe 1,"SETUP1.TFS" Copies the setup file in the setup memory location 1 to a file named "SETUP1.TFS" in the mass storage system.

## MMEMory:STORe:TRACe (No Query Form)

This command copies a waveform data file in the Edit Memory to a file in the mass storage system. If the file in the mass storage is locked, this command causes an error. You cannot create a new file if the directory is locked.

<b>Group</b>	Mass Memory
<b>Related Commands</b>	MMEMory:LOCK[:STATe] MMEMory:LOAD:TRACe
<b>Syntax</b>	MMEMory:STORe:TRACe EMEMory,<file_name>
<b>Arguments</b>	<file_name>::=<string> specifies a file name in the mass storage system. The <file_name> includes path. Path separators are forward slashes (/).
<b>Examples</b>	MMEMory:STORe:TRACe EMEMory,"SAMPLE1.TFW" Copies the content of EMEMory to a file named "SAMPLE1.TFW" in the mass storage system.



## \*OPC

This command generates the operation complete message by setting bit 0 in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete.

The query command places the ASCII character “1” into the output queue when all such OPC commands are complete.

<b>Group</b>	Synchronization
<b>Syntax</b>	*OPC  *OPC?
<b>Arguments</b>	None
<b>Returns</b>	<execution complete>::=1 where “1” indicates that all pending operations are complete.
<b>Examples</b>	*OPC? might return 1 to indicate that all pending OPC operations are finished.

## \*OPT?

This query-only command returns a list of the options installed in your arbitrary/function generator.

<b>Group</b>	System
<b>Syntax</b>	*OPT?
<b>Arguments</b>	None
<b>Returns</b>	<OPT>[,<OPT>[,<OPT>[,<OPT>]]]
<b>Examples</b>	*OPT? might return 0, which indicates no option is installed in the instrument.

## OUTPut[1|2]:IMPedance

The OUTPut : IMPedance command sets the output load impedance for the specified channel. The specified value is used for amplitude, offset, and high/low level settings. You can set the impedance to any value from 1  $\Omega$  to 10 k $\Omega$ . The default value is 50  $\Omega$ .

The OUTPut : IMPedance? command returns the current load impedance setting in ohms. If the load impedance is set to INFINITY, the query command returns "9.9E+37".

**Group**     Output

**Syntax**     OUTPut[1|2]:IMPedance{<ohms>|INFINITY|MINimum|MAXimum}

OUTPut[1|2]:IMPedance?[MINimum|MAXimum]

**Arguments**     <ohms>::=<NR3>[<units>]  
                      where  
                      <units>::=OHM

INFINITY sets the load impedance to >10 k $\Omega$

MINimum sets the load impedance to 1  $\Omega$

MAXimum sets the load impedance to 10 k $\Omega$

**Returns**     <ohms>::=<NR3>

**Examples**     OUTPut1:IMPedance MAXimum  
                      sets the CH 1 load impedance to 10 k $\Omega$ .

## OUTPut[1|2]:POLarity

This command inverts a specified output waveform relative to the offset level. The query command returns the polarity for the specified channel.

**Group** Output

**Syntax** OUTPut[1|2]:POLarity{NORMal|INVerted}

OUTPut[1|2]:POLarity?

**Arguments** NORMal sets the specified output waveform polarity to Normal.

INVerted sets the specified output waveform polarity to Inverted.

**Returns** NORM|INV

**Examples** OUTPut1:POLarity NORMal  
sets the CH 1 waveform polarity to Normal.

## OUTPut[1|2][:STATe]

This command sets or query whether to enable the arbitrary/function generator output for the specified channel.

**Group** Output

**Syntax** OUTPut[1|2][:STATe] {ON|OFF|<NR1>}

OUTPut[1|2][:STATe]?

**Arguments** ON or <NR1>≠0 enables the arbitrary/function generator output.

OFF or <NR1>=0 disables the arbitrary/function generator output.

**Returns** <NR1>

**Examples** OUTPut1:STATe ON  
sets the arbitrary/function generator CH 1 output to ON.

## OUTPut:TRIGger:MODE

This command sets or queries the mode (trigger or sync) for Trigger Output signal.

When the burst count is set to Inf-Cycles in burst mode, TRIGger indicates that the infinite number of cycles of waveform will be output from the Trigger Output connector.

When the burst count is set to Inf-Cycles in burst mode, SYNC indicates that one pulse waveform is output from the Trigger Output connector when the Inf-Cycles starts.

When Run Mode is specified other than Burst Inf-Cycles, TRIGger and SYNC have the same effect.

**Group** Output

**Syntax** OUTPut:TRIGger:MODE {TRIGger|SYNC}

OUTPut:TRIGger:MODE?

**Arguments** TRIGger means TRIGger is selected for Trigger Out.

SYNC means SYNC is selected for Trigger Out.

**Returns** TRIG|SYNC

**Examples** OUTPut:TRIGger:MODE SYNC  
outputs one cycle waveform from the Trigger Output connector when Inf-Cycles starts.

**\*PSC**

This command sets and queries the power-on status flag that controls the automatic power-on execution of SRER and ESER. When \*PSC is true, SRER and ESER are set to 0 at power-on. When \*PSC is false, the current values in the SRER and ESER are preserved in nonvolatile memory when power is shut off and are restored at power-on.

**Group** Status

**Syntax** \*PSC <NR1>

\*PSC?

**Arguments** <NR1>=0 sets the power-on status clear flag to false, disables the power-on clear, and allows the instrument to possibly assert SRQ after power-on.

<NR1>≠0 sets the power-on status clear flag true. Sending \*PSC 1 therefore enables the power-on status clear and prevents any SRQ assertion after power-on.

**Returns** <NR1>

**Examples** \*PSC 0  
sets the power-on status clear flag to false.

**\*RCL (No Query Form)**

This command restores the state of the instrument from a copy of the settings stored in the setup memory. The settings are stored using the \*SAV command. If the specified setup memory is deleted, this command causes an error.

**Group** Memory

**Related Commands** \*SAV

**Syntax** \*RCL {0|1|2|3|4}

**Arguments** 0, 1, 2, 3, or 4 specifies the location of setup memory.

**Examples** \*RCL 3  
restores the instrument from a copy of the settings stored in memory location 3.

**\*RST (No Query Form)**

This command resets the instrument to the factory default settings. This command is equivalent to pushing the **Default** button on the front panel. The default values are listed in *Default Settings* on page D-1.

<b>Group</b>	System
<b>Syntax</b>	*RST
<b>Arguments</b>	None
<b>Examples</b>	*RST resets the arbitrary/function generator settings to the factory defaults.

**\*SAV (No Query Form)**

This command stores the current settings of the arbitrary/function generator to a specified setup memory location.

A setup memory location numbered 0 (last setup memory) is automatically overwritten by the setups when you power off the instrument.

If a specified numbered setup memory is locked, this command causes an error.

<b>Group</b>	Memory
<b>Related Commands</b>	*RCL
<b>Syntax</b>	*SAV {0 1 2 3 4}
<b>Arguments</b>	0, 1, 2, 3, or 4 specifies the location of setup memory.
<b>Examples</b>	*SAV 2 saves the current instrument state in the memory location 2.

**[SOURce[1|2]]:AM[:DEPTh]**

This command sets or queries the modulation depth of AM modulation for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:AM:DEPTh{<depth>|MINimum|MAXimum}

[SOURce[1|2]]:AM:DEPTh?[MINimum|MAXimum]

**Arguments** <depth>::=<NR2>[<units>]  
where  
<NR2> is the depth of modulating frequency.  
<units>::=PCT

MINimum sets the modulation depth to minimum value.

MAXimum sets the modulation depth to maximum value.

**Returns** <depth>

**Examples** SOURce1:AM:DEPth MAXimum  
sets the depth of modulating signal on CH 1 to the maximum value.

## [SOURce[1|2]]:AM:INTernal:FREQuency

This command sets or queries the internal modulation frequency of AM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

You can select the source of modulating signal by using the [SOURce[1|2]]:AM:SOURce [INTernal|EXTernal] command.

**Group** Source

**Related Commands** [SOURce[1|2]]:AM:SOURce

**Syntax** [SOURce[1|2]]:AM:INTernal:FREQuency{<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:AM:INTernal:FREQuency?[MINimum|MAXimum]

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the modulation frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:AM:INTernal:FREQuency 10kHz  
sets the CH 1 internal modulation frequency to 10 kHz.



## [SOURce[1|2]]:AM:INTernal:FUNCTion

This command sets or queries the modulating waveform of AM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

If you specify EFILE when there is no EFILE or the EFILE is not yet defined, this command causes an error.

**Group** Source

**Related Commands** [SOURce[1|2]]:AM:SOURce

**Syntax** [SOURce[1|2]]:AM:INTernal:FUNCTion {SINusoid|SQUare|TRIangle|RAMP|NRamp|PRNoise|USER[1]|USER2|USER3|USER4|EMEMory|EFILE}

[SOURce[1|2]]:AM:INTernal:FUNCTion?

**Arguments** SINusoid|SQUare|TRIangle|RAMP|NRamp|PRNoise  
One of six types of function waveform can be selected as a modulating signal.

USER[1]|USER2|USER3|USER4|EMEMory  
A user defined waveform saved in the user waveform memory or the EMEMory can be selected as a modulating signal.

EFILE  
EFILE is used as a modulating signal.

**Returns** SIN|SQU|TRI|RAMP|NRAM|PRN|USER1|USER2|USER3|USER4|EMEMory|EFILE

**Examples** SOURce1:AM:INTernal:FUNCTion SQUare  
selects Square as the shape of modulating waveform for the CH 1 output.

**[SOURce[1|2]]:AM:INTernal:FUNCTion:EFILe**

This command sets or queries an EFILE name used as a modulating waveform for AM modulation. A file name must be specified in the mass storage system. This command causes an error if there is no file in the mass storage.

**Group** Source

**Syntax** [SOURce[1|2]]:AM:INTernal:FUNCTion:EFILe <file\_name>  
[SOURce[1|2]]:AM:INTernal:FUNCTion:EFILe?

**Arguments** <file\_name>::=<string> specifies a file name in the mass storage system. The <file\_name> includes path. Path separators are forward slashes (/).

**Returns** <file\_name>

**Examples** SOURce1:AM:INTernal:FUNCTion:EFILe "SAMPLE1"  
sets a file named "SAMPLE1" in the mass storage.

**[SOURce[1|2]]:AM:SOURce**

This command sets or queries the source of modulating signal of AM modulation for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:AM:SOURce [INTernal|EXTernal]  
[SOURce[1|2]]:AM:SOURce?

**Arguments** INTernal means that the carrier waveform is modulated with an internal source.  
EXTernal means that the carrier waveform is modulated with an external source.

**Returns** INT|EXT

**Examples** SOURce1:AM:SOURce INTernal  
sets the CH 1 source of modulating signal to internal.

## [SOURce[1|2]]:AM:STATe

This command enables or disables AM modulation for the specified channel. The query command returns the state of AM modulation.

**Group** Source

**Syntax** [SOURce[1|2]]:AM:STATe {ON|OFF|<NR1>}

[SOURce[1|2]]:AM:STATe?

**Arguments** ON or <NR1>≠0 enables AM modulation.

OFF or <NR1>=0 disables AM modulation.

**Returns** <NR1>

**Examples** SOURce1:AM:STATe ON  
enables the CH 1 AM modulation.

## [SOURce[1|2]]:BURSt:MODE

This command sets or queries the burst mode for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:BURSt:MODE{TRIGgered|GATed}

[SOURce[1|2]]:BURSt:MODE?

**Arguments** TRIGgered means that triggered mode is selected for burst mode.

GATed means that gated mode is selected for burst mode.

**Returns** TRIG|GAT

**Examples** SOURce1:BURSt:MODE TRIGgered  
selects triggered mode.

**[SOURce[1|2]]:BURSt:NCYCles**

This command sets or queries the number of cycles (burst count) to be output in burst mode for the specified channel. The query command returns 9.9E+37 if the burst count is set to INFINITY.

**Group** Source

**Syntax** [SOURce[1|2]]:BURSt:NCYCles {<cycles>|INFINITY|MINimum|MAXimum}  
[SOURce[1|2]]:BURSt:NCYCles? {MINimum|MAXimum}

**Arguments** <cycles>::=<NRf>  
where  
<NRf> is the burst count.  
The burst count ranges from 1 to 1,000,000.  
  
INFINITY sets the burst count to infinite count.  
  
MINimum sets the burst count to minimum count.  
  
MAXimum sets the burst count to maximum count.

**Returns** <cycles>

**Examples** SOURce1:BURSt:NCYCles 2  
sets the CH 1 burst count to 2.

**[SOURce[1|2]]:BURSt[:STATe]**

This command enables or disables the burst mode for the specified channel. The query command returns the state of burst mode.

**Group** Source

**Syntax** [SOURce[1|2]]:BURSt:STATe {ON|OFF|<NR1>}

[SOURce[1|2]]:BURSt:STATe?

**Arguments** ON or <NR1>≠0 enables the burst mode.

OFF or <NR1>=0 disables the burst mode.

**Returns** <NR1>

**Examples** SOURce1:BURSt:STATe ON  
enables the burst mode for the CH 1.

**[SOURce[1|2]]:BURSt:TDElay**

This command sets or queries delay time in the burst mode for the specified channel. It specifies a time delay between the trigger and the signal output. This command is available only in the Triggered burst mode.

**Group** Source

**Syntax** [SOURce[1|2]]:BURSt:TDElay {<delay>|MINimum|MAXimum}  
[SOURce[1|2]]:BURSt:TDElay?{MINimum|MAXimum}

**Arguments** <delay>::=<NRf>[<units>]  
where  
<units>::=[s | ms |  $\mu$ s | ns]

MINimum sets the delay time to minimum value.

MAXimum sets the delay time to maximum value.

**Returns** <delay>

**Examples** SOURce1:BURSt:DElay 20ms  
sets the CH 1 delay time to 20 ms.

## [SOURce[1|2]]:COMBine:FEED

This command sets or queries whether to add the internal noise or an external signal to an output signal for the specified channel.

When you specify the internal noise, you can set or query the noise level by SOURce<3|4>:POWer[:LEVel][:IMMediate][:AMPLitude] command.

To disable the internal noise add or the external signal add function, specify “”.

You can add an external signal to the CH 1 output signal of the AFG3100 and AFG3200 series arbitrary/function generators.

Both the internal noise and an external signal can be added simultaneously to the arbitrary/function generator.

<b>Group</b>	Source
<b>Related Commands</b>	SOURce<3 4>:POWer[:LEVel][:IMMediate][:AMPLitude]
<b>Syntax</b>	[SOURce[1 2]]:COMBine:FEED [“NOISe” “EXTerna” “BOTH” “”] [SOURce[1 2]]:COMBine:FEED?
<b>Arguments</b>	<p>NOISe indicates that the internal noise is added to the output signal.</p> <p>EXTerna indicates that an external signal is added to the CH 1 output signal of the AFG3100 or AFG3200 series arbitrary/function generators.</p> <p>BOTH indicates that the internal noise and an external signal are added to the CH 1 output signal of the AFG3100 or AFG3200 series arbitrary/function generators.</p> <p>“” disables the internal noise add and external signal add function.</p>
<b>Returns</b>	“NOIS” “EXT” “BOTH” “”
<b>Examples</b>	<p>SOURce1:COMBine:FEED EXTerna</p> <p>adds an external signal to the CH 1 output signal.</p>

## [SOURce[1|2]]:FM[:DEVIation]

This command sets or queries the peak frequency deviation of FM modulation for the specified channel. The setting range of frequency deviation depends on the waveform selected as the carrier.

**Group** Source

**Syntax** [SOURce[1|2]]:FM:DEVIation {<deviation>|MINimum|MAXimum}  
[SOURce[1|2]]:FM:DEVIation?{MINimum|MAXimum}

**Arguments** <deviation>::=<NRf>[<units>]  
where  
<NRf> is the frequency deviation.  
<units>::=[Hz | kHz | MHz]

**Returns** <deviation>

**Examples** SOURce1:FM:DEVIation 1.0MHz  
sets the CH 1 frequency deviation to 1.0 MHz.

## [SOURce[1|2]]:FM:INTernal:FREQuency

This command sets or queries the internal modulation frequency of FM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

**Group** Source

**Syntax** [SOURce[1|2]]:FM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:FM:INTernal:FREQuency?[MINimum|MAXimum]

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the modulation frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:FM:INTernal:FREQuency 10kHz  
sets the CH 1 internal modulation frequency to 10 kHz.



## [SOURce[1|2]]:FM:INTernal:FUNCTion

This command sets or queries the modulating waveform of FM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

If you specify EFILE when there is no EFILE or the EFILE is not yet defined, this command causes an error.

**Group** Source

**Related Commands** [SOURce[1|2]]:FM:SOURce

**Syntax** [SOURce[1|2]]:FM:INTernal:FUNCTion{SINusoid|SQUare|TRIangle|RAMP|NRAMP|PRNoise|USER[1]|USER2|USER3|USER4|EMEMory|EFILE}

[SOURce[1|2]]:FM:INTernal:FUNCTion?

**Arguments** SINusoid|SQUare|TRIangle|RAMP|NRAMP|PRNoise  
One of six types of function waveform can be selected as a modulating signal.

USER[1]|USER2|USER3|USER4|EMEMory  
A user defined waveform saved in the user waveform memory or the EMEMory can be selected as a modulating signal.

EFILE  
EFILE is used as a modulating signal.

**Returns** SIN|SQU|TRI|RAMP|NRAMP|PRN|USER1|USER2|USER3|USER4|EMEMory|EFILE

**Examples** SOURce1:FM:INTernal:FUNCTion SQUare  
selects Square as the shape of modulating waveform for the CH 1 output.

**[SOURce[1|2]]:FM:INTernal:FUNCTion:EFILe**

This command sets or queries an EFILE name used as a modulating waveform for FM modulation. A file name must be specified in the mass storage system. This command causes an error if there is no file in the mass storage.

**Group** Source

**Syntax** [SOURce[1|2]]:FM:INTernal:FUNCTion:EFILe <file\_name>  
[SOURce[1|2]]:FM:INTernal:FUNCTion:EFILe?

**Arguments** <file\_name>::=<string> specifies a file name in the mass storage system. The <file\_name> includes path. Path separators are forward slashes (/).

**Returns** <file\_name>

**Examples** SOURce1:FM:INTernal:FUNCTion:EFILe "SAMPLE1"  
sets a file named "SAMPLE1" in the mass storage.

**[SOURce[1|2]]:FM:SOURce**

This command sets or queries the source of modulating signal of FM modulation for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:FM:SOURce [INTernal|EXTernal]  
[SOURce[1|2]]:FM:SOURce?

**Arguments** INTernal means that the carrier waveform is modulated with the internal source.  
EXTernal means that the carrier waveform is modulated with an external source.

**Returns** INT|EXT

**Examples** SOURce1:FM:SOURce INTernal  
sets the CH 1 source of modulating signal to internal.

**[SOURce[1|2]]:FM:STATe**

This command enables or disables FM modulation. The query command returns the state of FM modulation.

**Group** Source

**Syntax** [SOURce[1|2]]:FM:STATe {ON|OFF|<NR1>}

[SOURce[1|2]]:FM:STATe?

**Arguments** ON or <NR1>≠0 enables FM modulation.

OFF or <NR1>=0 disables FM modulation.

**Returns** <NR1>

**Examples** SOURce1:FM:STATe ON  
enables the CH 1 FM modulation.

**[SOURce[1|2]]:FREQuency:CENTer**

This command sets or queries the center frequency of sweep for the specified channel. This command is always used with the [SOURce[1|2]]:FREQuency:SPAN command. The setting range of center frequency depends on the waveform selected for sweep.

**Group** Source

**Related Commands** [SOURce[1|2]]:FREQuency:SPAN  
[SOURce[1|2]]:FREQuency:MODE

**Syntax** [SOURce[1|2]]:FREQuency:CENTer {<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:FREQuency:CENTer?

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the center frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:FREQuency:CENTer 550kHz  
sets the CH 1 center frequency to 550 kHz.

**[SOURce[1|2]]:FREQuency:CONCurent[:STATe]**

This command enables or disables the function to copy the frequency (or period) of one channel to another channel.

The[SOURce[1|2]]:FREQuency:CONCurent command copies the frequency (or period) of the channel specified by the header suffix to another channel. If you specify CH 1 with the header, the CH 1 frequency will be copied to CH 2.

The[SOURce[1|2]]:FREQuency:CONCurent? command returns “0” (off) or “1” (on).

If your arbitrary/function generator is single-channel model, this command is not supported.

**Group** Source

**Syntax** [SOURce[1|2]]:FREQuency:CONCurent {ON|OFF|<NR1>}  
[SOURce[1|2]]:FREQuency:CONCurent?

**Arguments** ON or <NR1>≠0 enables the concurrent copy function.  
OFF or <NR1>=0 disables the concurrent copy function.

**Returns** <NR1>

**Examples** SOURce1:FREQuency:CONCurent ON  
copies the frequency value of CH 1 to CH 2.

**[SOURce[1|2]]:FREQuency[:CW|:FIXed]**

This command sets or queries the frequency of output waveform for the specified channel. This command is available when the Run Mode is set to other than Sweep.

The setting range of output frequency depends on the type of output waveform. If you change the type of output waveform, it might change the output frequency because changing waveform types impacts on the setting range of output frequency.

<b>Group</b>	Source
<b>Syntax</b>	<p>[SOURce[1 2]]:FREQuency[:CW :FIXed] {&lt;frequency&gt; MINimum MAXimum}</p> <p>[SOURce[1 2]]:FREQuency[:CW :FIXed]?{MINimum MAXimum}</p>
<b>Arguments</b>	<p>&lt;frequency&gt;::=&lt;NRf&gt;[&lt;units&gt;]</p> <p>where</p> <p>&lt;NRf&gt; is the output frequency.</p> <p>&lt;units&gt;::=[Hz   kHz   MHz]</p>
<b>Returns</b>	<frequency>
<b>Examples</b>	<p>SOURce1:FREQuency:FIXed 500kHz</p> <p>sets the CH 1 output frequency to 500 kHz when the Run Mode is set to other than Sweep.</p>

**[SOURce[1|2]]:FREQuency:MODE**

This command sets or queries the frequency sweep state. You can select sine, square, ramp, or arbitrary waveform for sweep. The arbitrary/function generator automatically changes to the Continuous mode if any waveform is selected other than sine, square, ramp, or an arbitrary waveform.

**Group** Source

**Related Commands** [SOURce[1|2]]:FREQuency[:CW|:FIXed]  
[SOURce[1|2]]:FREQuency:CENTer  
[SOURce[1|2]]:FREQuency:SPAN  
[SOURce[1|2]]:FREQuency:START  
[SOURce[1|2]]:FREQuency:STOP

**Syntax** [SOURce[1|2]]:FREQuency:MODE {CW|FIXed|SWEep}  
  
[SOURce[1|2]]:FREQuency:MODE?

**Arguments** CW|FIXed means that the frequency is controlled by the [SOURce[1|2]]:FREQuency[:CW|:FIXed] command. The sweep is invalid.  
  
SWEep means that the output frequency is controlled by the sweep command set. The sweep is valid.

**Returns** CW|FIXed|SWEep

**Examples** SOURce1:FREQuency:MODE SWEep  
specifies the sweep command set for controlling the CH 1 output frequency.

## [SOURce[1|2]]:FREQuency:SPAN

This command sets or queries the span of frequency sweep for the specified channel. This command is always used with the [SOURce[1|2]]:FREQuency:CENTer command. The setting range of frequency span depends on the waveform selected for sweep.

**Group** Source

**Related Commands** [SOURce[1|2]]:FREQuency:CENTer  
[SOURce[1|2]]:FREQuency:MODE

**Syntax** [SOURce[1|2]]:FREQuency:SPAN {<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:FREQuency:SPAN? {MINimum|MAXimum}

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the frequency span.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:FREQuency:SPAN 900 kHz  
sets the CH 1 frequency span to 900 kHz.

**[SOURce[1|2]]:FREQuency:STARt**

This command sets or queries the start frequency of sweep for the specified channel. This command is always used with the [SOURce[1|2]]:FREQuency:STOP command. The setting range of start frequency depends on the waveform selected for sweep.

**Group** Source

**Related Commands** [SOURce[1|2]]:FREQuency:MODE  
[SOURce[1|2]]:FREQuency:STOP

**Syntax** [SOURce[1|2]]:FREQuency:STARt {<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:FREQuency:STARt? {MINimum|MAXimum}

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the start frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:FREQuency:STARt 10kHz  
sets the sweep start frequency of CH 1 to 10 kHz.



## [SOURce[1|2]]:FREQuency:STOP

This command sets or queries the start frequency of sweep for the specified channel. This command is always used with the [SOURce[1|2]]:FREQuency:START command. The setting range of stop frequency depends on the waveform selected for sweep.

**Group** Source

**Related Commands** [SOURce[1|2]]:FREQuency:MODE  
[SOURce[1|2]]:FREQuency:START

**Syntax** [SOURce[1|2]]:FREQuency:STOP {<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:FREQuency:STOP? {MINimum|MAXimum}

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the stop frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:FREQuency:STOP 100KHz  
sets the stop frequency of CH 1 to 100 kHz.

**[SOURce[1|2]]:FSKey[:FREQuency]**

This command sets or queries the hop frequency of FSK modulation for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:FSKey[:FREQuency] {<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:FSKey[:FREQuency]? {MINimum|MAXimum}

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the hop frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:FSKey:FREQuency 1.0MHz  
sets the hop frequency of CH 1 FSK modulation to 1.0 MHz.

**[SOURce[1|2]]:FSKey:INTernal:RATE**

This command sets or queries the internal modulation rate of FSK modulation for the specified channel. You can use this command only when the internal modulation source is selected.

**Group** Source

**Syntax** [SOURce[1|2]]:FSKey:INTernal:RATE {<rate>|MINimum|MAXimum}  
[SOURce[1|2]]:FSKey:INTernal:RATE? [MINimum|MAXimum]

**Arguments** <rate>::=<NRf>[<units>]  
where  
<NRf> is the modulation rate.  
<units>::=[Hz | kHz | MHz]

**Returns** <rate>

**Examples** SOURce1:FSKey:INTernal:RATE 50Hz  
sets the CH 1 internal modulation rate to 50 Hz.

**[SOURce[1|2]]:FSKey:SOURce**

This command sets or queries the source of modulation signal of FSK modulation for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:FSKey:SOURce [INTerna1|EXTerna1]

[SOURce[1|2]]:FSKey:SOURce?

**Arguments** INTerna1 means that the carrier waveform is modulated with an internal source.  
EXTerna1 means that the carrier waveform is modulated with an external source.

**Returns** INT|EXT

**Examples** SOURce1:FSKey:SOURce INTerna1  
sets the CH 1 source of modulating signal to internal.

**[SOURce[1|2]]:FSKey:STAtE**

This command enables or disables FSK modulation. The query command returns the state of FSK modulation. You can select a sine, square, ramp, or arbitrary waveform for the carrier waveform.

**Group** Source

**Syntax** [SOURce[1|2]]:FSKey:STAtE {ON|OFF|<NR1>}

[SOURce[1|2]]:FSKey:STAtE?

**Arguments** ON or <NR1>≠0 enables FSK modulation.  
OFF or <NR1>=0 disables FSK modulation.

**Returns** <NR1>

**Examples** SOURce1:FSKey:STAtE ON  
enables the CH 1 FSK modulation.

## [SOURce[1|2]]:FUNCTION:EFILe

This command sets or queries an EFILe name used as an output waveform. A file name must be specified in the mass storage system. This command causes an error if there is no file in the mass storage.

**Group** Source

**Syntax** [SOURce[1|2]]:FUNCTION:EFILe <file\_name>  
[SOURce[1|2]]:FUNCTION:EFILe?

**Arguments** <file\_name>::=<string> specifies a file name in the mass storage system. The <file\_name> includes path. Path separators are forward slashes (/).

**Returns** <file\_name>

**Examples** SOURce1:FUNCTION:EFILe "SAMPLE1"  
sets a file named "SAMPLE1" in the mass storage.

## [SOURce[1|2]]:FUNCTION:RAMP:SYMMetry

This command sets or queries the symmetry of ramp waveform for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:FUNCTION:RAMP:SYMMetry {<symmetry>|MINimum|MAXimum}  
[SOURce[1|2]]:FUNCTION:RAMP:SYMMetry? {MINimum|MAXimum}

**Arguments** <symmetry>::=<NR2>[<units>]  
where  
<NR2> is the symmetry.  
<units>::=PCT

**Returns** <symmetry>

**Examples** SOURce1:FUNCTION:RAMP:SYMMetry 80.5  
sets the symmetry of the CH 1 ramp waveform to 80.5 %.

## [SOURce[1|2]]:FUNction[:SHAPE]

This command sets or queries the shape of the output waveform. When the specified user memory is deleted, this command causes an error if you select the user memory.

**Group** Source

**Syntax** [SOURce[1|2]]:FUNction[:SHAPE] {SINusoid|SQUare|PULSe|RAMP|PRNoise|DC|SINC|GAUSSian|LOrentz|ERISe|EDEcay|HAVersine|USER[1]|USER2|USER3|USER4|EMEMory|EFILe}

[SOURce[1|2]]:FUNction[:SHAPE]?

**Arguments** SINusoid|SQUare|PULSe|RAMP|PRNoise|DC|SINC|GAUSSian|LOrentz|ERISe|EDEcay|HAVersine

The following table shows the combination of modulation type and the shape of output waveform.

	Sine, Square, Ramp, Arb, Sin(x)/x, Gaussian, Lorentz, Exponential Rise, Exponential Decay, Haversine	Pulse	Noise, DC
AM	√		
FM	√		
PM	√		
FSK	√		
PWM		√	
Sweep	√		
Burst	√	√	

If you select a waveform shape that is not allowed with a particular modulation, sweep, or burst, the Run mode will automatically be changed to Continuous.

If you specify EFILe when there is no EFILe or the EFILe is not yet defined, this command causes an error.

If you change the type of output waveform, it might change the output frequency because changing waveform types impacts the setting range of output frequency.

USER[1]|USER2|USER3|USER4|EMEMory

A user defined waveform saved in the user waveform memory or the EMEMory can be selected as an output waveform.

EFILe

EFILe is specified as an output waveform.

**Returns** SIN|SQU|PULS|RAMP|PRN|DC|SINC|GAUS|LOR|ERIS|EDEC|HARV|USER1|USER2  
|USER3|USER4|EMEMory|EFILe

**Examples** SOURce1:FUNctIon:SHApe SQUARE  
selects the shape of CH 1 output waveform to square waveform.

## [SOURce[1|2]]:PHASe[:ADJust]

This command sets or queries the phase of output waveform for the specified channel. You can set the value in radians or degrees. If no units are specified, the default is RAD. The query command returns the value in RAD.

This command is supported when you select a waveform other than DC, Noise, and Pulse.

**Group** Source

**Syntax** [SOURce[1|2]]:PHASe[:ADJust] {<phase>|MINimum|MAXimum}  
[SOURce[1|2]]:PHASe[:ADJust]? {MINimum|MAXimum}

**Arguments** <phase>::=<NR3>[<units>]  
where  
<NR3> is the phase of output frequency.  
<units>::=[RAD | DEG]

If <units> are omitted, RAD is specified automatically. The setting ranges are:

RAD -1 PI to +1 PI, relative to phase value  
DEG -180 to +180, in 1 degree steps, relative to phase value

**Returns** <phase>

**Examples** SOURce1:PHASe:ADJust MAXimum  
sets the maximum value for the phase of CH 1 output frequency.

**[SOURce[1|2]]:PHASe:INITiate (No Query Form)**

This command synchronizes the phase of CH 1 and CH 2 output waveforms. The arbitrary/function generator performs the same operation if you specify either SOURce1 or SOURce2.

If your arbitrary/function generator is single-channel model, this command is not supported.

<b>Group</b>	Source
<b>Syntax</b>	[SOURce[1 2]]:PHASe:INITiate
<b>Arguments</b>	None
<b>Examples</b>	SOURce1:PHASe:INITiate synchronizes the phase of CH 1 and CH 2 output signals.

**[SOURce[1|2]]:PM[:DEViation]**

This command sets or queries the peak frequency deviation of PM modulation for the specified channel.

<b>Group</b>	Source
<b>Syntax</b>	[SOURce[1 2]]:PM:DEViation {<deviation> MINimum MAXimum}  [SOURce[1 2]]:PM:DEViation? {MINimum MAXimum}
<b>Arguments</b>	<deviation>::=<NR3>[<units>] where <NR3> is the phase deviation. <units>::=[RAD   DEG]  If <units> are omitted, RAD is specified automatically. The setting ranges are:  RAD    -1 PI to +1 PI, relative to phase value DEG    -180 to +180, in 1 degree steps, relative to phase value
<b>Returns</b>	<deviation>
<b>Examples</b>	SOURce1:PM:DEViation MAXimum sets the maximum value for the CH 1 phase deviation.

**[SOURce[1|2]]:PM:INTernal:FREQuency**

This command sets or queries the internal modulation frequency of PM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

**Group** Source

**Related Commands** [SOURce[1|2]]:PM:SOURce

**Syntax** [SOURce[1|2]]:PM:INTernal:FREQuency {<frequency>|MINimum|MAXimum}  
[SOURce[1|2]]:PM:INTernal:FREQuency? [MINimum|MAXimum]

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the modulation frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:PM:INTernal:FREQuency 10kHz  
sets the CH 1 internal modulation frequency to 10 kHz.



## [SOURce[1|2]]:PM:INTernal:FUNCTion

This command sets or queries the modulating waveform of PM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

If you specify EFILE when there is no EFILE or the EFILE is not yet defined, this command causes an error.

**Group** Source

**Related Commands** [SOURce[1|2]]:PM:SOURce

**Syntax** [SOURce[1|2]]:PM:INTernal:FUNCTion {SINusoid|SQUare|TRIangle|  
RAMP|NRAMP|PRNoise|USER[1]|USER2|USER3|USER4|EMEMory|EFILE}  
  
[SOURce[1|2]]:PM:INTernal:FUNCTion?

**Arguments** SINusoid|SQUare|TRIangle|RAMP|NRAMP|PRNoise  
One of six types of function waveform can be selected as a modulating signal.

USER[1]|USER2|USER3|USER4|EMEMory  
A user defined waveform saved in the user waveform memory or the EMEMory can be selected as a modulating signal.

EFILE  
EFILE is used as a modulating signal.

**Returns** SIN|SQU|TRI|RAMP|NRAMP|PRN|USER1|USER2|USER3|USER4|EMEMory|EFILE

**Examples** SOURce1:PM:INTernal:FUNCTion SQUare  
selects Square as the shape of modulating waveform for the CH 1 output.

**[SOURce[1|2]]:PM:INTernal:FUNCTion:EFILe**

This command sets or queries an EFILE name used as a modulating waveform for PM modulation. A file name must be specified in the mass storage system. This command causes an error if there is no file in the mass storage.

**Group** Source

**Syntax** [SOURce[1|2]]:PM:INTernal:FUNCTion:EFILe <file\_name>  
[SOURce[1|2]]:PM:INTernal:FUNCTion:EFILe?

**Arguments** <file\_name>::=<string> specifies a file name in the mass storage system. The <file\_name> includes path. Path separators are forward slashes (/).

**Returns** <file\_name>

**Examples** SOURce1:PM:INTernal:FUNCTion:EFILe "SAMPLE1"  
sets a file named "SAMPLE1" in the mass storage.

**[SOURce[1|2]]:PM:SOURce**

This command sets or queries the source of modulation signal of PM modulation for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:PM:SOURce [INTernal|EXTernal]  
[SOURce[1|2]]:PM:SOURce?

**Arguments** INTernal means that the carrier waveform is modulated with an internal source.  
EXTernal means that the carrier waveform is modulated with an external source.

**Returns** INT|EXT

**Examples** SOURce1:PM:SOURce INTernal  
sets the CH 1 source of modulating signal to internal.

**[SOURce[1|2]]:PM:STATe**

This command enables or disables PM modulation. The query command returns the state of PM modulation. You can select a sine, square, ramp, or arbitrary waveform for the carrier waveform.

**Group** Source

**Syntax** [SOURce[1|2]]:PM:STATe {ON|OFF|<NR1>}

[SOURce[1|2]]:PM:STATe?

**Arguments** ON or <NR1>≠0 enables PM modulation.

OFF or <NR1>=0 disables PM modulation.

**Returns** <NR1>

**Examples** SOURce1:PM:STATe ON  
enables the CH 1 PM modulation.

**SOURce<3|4>:POWer[:LEVel][:IMMediate][:AMPLitude]**

This command sets or queries the internal noise level which applies to the output signal for the specified channel. The noise level represents the percent against current amplitude level. The setting range is 0 to 50%.

This command is available when Run Mode is set to Continuous, Burst, or Sweep.

You can set or query whether to add the internal noise to the output signal using the [SOURce[1|2]]:COMBine:FEED command.

<b>Group</b>	Source
<b>Related Commands</b>	[SOURce[1 2]]:COMBine:FEED
<b>Syntax</b>	<pre>SOURce&lt;3 4&gt;:POWer[:LEVel][:IMMediate][:AMPLitude] {&lt;percent&gt;  MINimum MAXimum}  SOURce&lt;3 4&gt;:POWer[:LEVel][:IMMediate][:AMPLitude]?[MINimum  MAXimum]</pre>
<b>Arguments</b>	<pre>&lt;percent&gt;::=&lt;NR2&gt;[&lt;units&gt;] where &lt;NR2&gt; is the noise level. &lt;units&gt;::=PCT</pre>
<b>Returns</b>	<percent>
<b>Examples</b>	<pre>SOURce3:POWer:LEVel:IMMediate:AMPLitude 50PCT sets the internal noise level that is added to the output signal to 50%.</pre>

## [SOURce[1|2]]:PULSe:DCYClE

This command sets or queries the duty cycle of the pulse waveform for the specified channel. The setting range is 0.1% to 99.9% in increments of 0.1.

The arbitrary/function generator will hold the settings of leading edge and trailing edge when the duty cycle is varied.

Refer to the [SOURce[1|2]]:PULSe:WIDTh command for the setting range.

<b>Group</b>	Source
<b>Related Commands</b>	[SOURce[1 2]]:PULSe:WIDTh
<b>Syntax</b>	[SOURce[1 2]]:PULSe:DCYClE {<percent> MINimum MAXimum} [SOURce[1 2]]:PULSe:DCYClE? {MINimum MAXimum}
<b>Arguments</b>	<percent>::=<NR2>[<units>] where <NRf> is the duty cycle. <units>::=PCT
<b>Returns</b>	<percent>
<b>Examples</b>	SOURce1:PULSe:DCYClE 80.5 sets the duty cycle of the pulse waveform on CH 1 to 80.5%.

## [SOURce[1|2]]:PULSe:DELay

This command sets or queries the lead delay of the pulse waveform for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:PULSe:DELay {<delay>|MINimum|MAXimum}

[SOURce[1|2]]:PULSe:DELay?{MINimum|MAXimum}

**Arguments** <delay>::=<NR2>[<units>]  
where <NR2> is the lead delay.  
<units>::=[ns |  $\mu$ s | ms | s]

Setting range: 0 ns to Pulse Period – {Pulse Width +  $0.8 \times$  (Leading Edge Time + Trailing Edge Time)}

**Returns** <delay>

**Examples** SOURce1:PULSe:DELay 20ms  
sets the CH 1 lead delay to 20 ms.

## [SOURce[1|2]]:PULSe:HOLD

The [SOURce[1|2]]:PULSe:HOLD command sets the arbitrary/function generator to hold either pulse width or pulse duty.

The [SOURce[1|2]]:PULSe:HOLD? command returns WIDTH or DUTY.

**Group** Source

**Syntax** [SOURce[1|2]]:PULSe:HOLD {WIDTH|DUTY}

[SOURce[1|2]]:PULSe:HOLD?

**Arguments** WIDTH means that the arbitrary/function generator holds the pulse width setting.

DUTY means that the arbitrary/function generator holds the pulse duty setting.

**Returns** WIDT|DUTY

**Examples** SOURce1:PULSe:HOLD WIDTH  
holds the CH 1 pulse width setting.

## [SOURce[1|2]]:PULSe:PERiod

This command sets or queries the period for pulse waveform.

**Group** Source

**Syntax** [SOURce[1|2]]:PULSe:PERiod {<period>|MINimum|MAXimum}  
[SOURce[1|2]]:PULSe:PERiod?{MINimum|MAXimum}

**Arguments** <period>::=<NRf>[<units>]  
where  
<NRf> is the pulse period.  
<units>::=[ns |  $\mu$ s | ms | s]

**Returns** <period>

**Examples** SOURce1:PULSe:PERiod 200ns  
sets the CH 1 pulse period to 200 ns.

## [SOURce[1|2]]:PULSe:TRANSition[:LEADIng]

This command sets or queries the leading edge time of pulse waveform.

**Group** Source

**Syntax** [SOURce[1|2]]:PULSe:TRANSition[:LEADIng] {<seconds>|MINimum|MAXimum}  
[SOURce[1|2]]:PULSe:TRANSition[:LEADIng]?{MINimum|MAXimum}

**Arguments** <seconds>::=<NRf>[<units>]  
where  
<NRf> is the leading edge time of pulse waveform.  
<units>::=[ns |  $\mu$ s | ms | s]

**Returns** <seconds>

**Examples** SOURce1:PULSe:TRANSition:LEADIng 200ns  
sets the CH 1 leading edge time to 200 ns.

**[SOURce[1|2]]:PULSe:TRANSition:TRAIling**

This command sets or queries the trailing edge time of pulse waveform.

**Group** Source

**Syntax** [SOURce[1|2]]:PULSe:TRANSition:TRAIling {<seconds>|MINimum|MAXimum}

[SOURce[1|2]]:PULSe:TRANSition:TRAIling?{MINimum|MAXimum}

**Arguments** <seconds>::=<NRf>[<units>]  
where  
<NRf> is the trailing edge of pulse waveform.  
<units>::=[ns |  $\mu$ s | ms | s]

**Returns** <seconds>

**Examples** SOURce1:PULSe:TRANSition:TRAIling 200ns  
sets the trailing edge time to 200 ns.



## [SOURce[1|2]]:PULSe:WIDTh

This command sets or queries the pulse width for the specified channel.

$$\text{Pulse Width} = \text{Period} \times \text{Duty Cycle} / 100$$

The pulse width must be less than the period. The setting range is 0.1% to 99.9% in terms of duty cycle.

AFG3021 and AFG3022: 30 ns to 999 s

AFG3101 and AFG3102: 8 ns to 999 s

AFG3251 and AFG3252: 4 ns to 999 s

$$\text{Pulse Width} \leq \text{Pulse Period} - 0.8 \times (\text{Leading Edge Time} + \text{Trailing Edge Time})$$

$$\text{Pulse Width} \geq 0.625 \times (\text{Leading Edge Time} + \text{Trailing Edge Time})$$

**Group** Source

**Related Commands** [SOURce[1|2]]:PULSe:DCYCl e

**Syntax** [SOURce[1|2]]:PULSe:WIDTh {<seconds>|MINimum|MAXimum}  
[SOURce[1|2]]:PULSe:WIDTh?{MINimum|MAXimum}

**Arguments** <seconds>::=<NRf>[<units>]  
where  
<NRf> is the pulse width.  
<units>::=[ns | μs | ms | s]

**Returns** <seconds>

**Examples** SOURce1:PULSe:WIDTh 200ns  
sets the CH 1 pulse width to 200 ns.

**[SOURce[1|2]]:PWM:INTernal:FREQuency**

This command sets or queries the internal modulation frequency of PWM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

**Group** Source

**Related Commands** [SOURce[1|2]]:PWM:SOURce

**Syntax** [SOURce[1|2]]:PWM:INTernal:FREQuency  
{<frequency>|MINimum|MAXimum}  
  
[SOURce[1|2]]:PWM:INTernal:FREQuency?[MINimum|MAXimum]

**Arguments** <frequency>::=<NRf>[<units>]  
where  
<NRf> is the modulation frequency.  
<units>::=[Hz | kHz | MHz]

**Returns** <frequency>

**Examples** SOURce1:PWM:INTernal:FREQuency 10kHz  
sets the CH 1 internal frequency to 10 kHz.

## [SOURce[1|2]]:PWM:INTernal:FUNCTion

This command sets or queries the modulating waveform of PWM modulation for the specified channel. You can use this command only when the internal modulation source is selected.

If you specify EFILE when there is no EFILE or the EFILE is not yet defined, this command causes an error.

**Group** Source

**Related Commands** [SOURce[1|2]]:PWM:SOURce

**Syntax** [SOURce[1|2]]:PWM:INTernal:FUNCTion {SINusoid|SQUare|TRIangle|  
RAMP|NRAMP|PRNoise|USER[1]|USER2|USER3|USER4|EMOMory|EFILE}  
  
[SOURce[1|2]]:PWM:INTernal:FUNCTion?

**Arguments** SINusoid|SQUare|TRIangle|RAMP|NRAMP|PRNoise  
One of six types of function waveform can be selected as a modulating signal.

USER[1]|USER2|USER3|USER4|EMEMory  
A user defined waveform saved in the user waveform memory or the EMEMory can be selected as a modulating signal.

EFILE  
EFILE is used as a modulating signal.

**Returns** SIN|SQU|TRI|RAMP|NRAMP|PRN|USER1|USER2|USER3|USER4|EMOMory|EFILE

**Examples** SOURce1:PWM:INTernal:FUNCTion SQUare  
selects Square as the shape of modulating waveform for the CH 1 output.

**[SOURce[1|2]]:PWM:INTernal:FUNCTion:EFILe**

This command sets or queries an EFILE name used as a modulating waveform for PWM modulation. A file name must be specified in the mass storage system. This command causes an error if there is no file in the mass storage.

**Group** Source

**Syntax** [SOURce[1|2]]:PWM:INTernal:FUNCTion:EFILe <file\_name>  
[SOURce[1|2]]:PWM:INTernal:FUNCTion:EFILe?

**Arguments** <file\_name>::=<string> specifies a file name in the mass storage system. The <file\_name> includes path. Path separators are forward slashes (/).

**Returns** <file\_name>

**Examples** SOURce1:PWM:INTernal:FUNCTion:EFILe "SAMPLE1"  
creates a file named "SAMPLE1" in the mass storage.

**[SOURce[1|2]]:PWM:SOURce**

This command sets or queries the source of modulating signal of PWM modulation for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:PWM:SOURce [INTernal|EXTernal]  
[SOURce[1|2]]:PWM:SOURce?

**Arguments** INTernal means that the carrier waveform is modulated with the internal source.  
EXTernal means that the carrier waveform is modulated with an external source.

**Returns** INT|EXT

**Examples** SOURce1:PWM:SOURce INTernal  
sets the source of modulating signal on CH 1 to internal.

**[SOURce[1|2]]:PWM:STATe**

This command enables or disables PWM modulation. The query command returns the state of PWM modulation. You can select only pulse waveform as a carrier waveform for PWM.

**Group** Source

**Syntax** [SOURce[1|2]]:PWM:STATe {ON|OFF|<NR1>}  
[SOURce[1|2]]:PWM:STATe?

**Arguments** ON or <NR1>≠0 enables PWM modulation.  
OFF or <NR1>=0 disables PWM modulation.

**Returns** <NR1>

**Examples** SOURce1:PWM:STATe ON  
enables the CH 1 PWM modulation.

**[SOURce[1|2]]:PWM[:DEViation]:DCYCle**

This command sets or queries the PWM deviation in percent for the specified channel.

The setting range must meet the following conditions:

$$\text{Deviation} \leq \text{Pulse Width} - \text{PWmin}$$

$$\text{Deviation} \leq \text{Pulse Period} - \text{Pulse Width} - \text{PWmin}$$

$$\text{Deviation} \leq \text{Pulse Width} - 0.8 \times (\text{Leading Edge Time} + \text{Trailing Edge Time})$$

$$\text{Deviation} \leq \text{Pulse Period} - \text{Pulse Width} - 0.8 \times (\text{Leading Edge Time} + \text{Trailing Edge Time})$$

where PWmin is the minimum pulse width.

<b>Group</b>	Source
<b>Syntax</b>	<code>[SOURce[1 2]]:PWM[:DEViation]:DCYCle {&lt;percent&gt; MINimum MAXimum}</code> <code>[SOURce[1 2]]:PWM[:DEViation]:DCYCle? {MINimum MAXimum}</code>
<b>Arguments</b>	<code>&lt;percent&gt;::=&lt;NR2&gt;[&lt;units&gt;]</code> where <code>&lt;NR2&gt;</code> is the PWM deviation. <code>&lt;units&gt;::=PCT</code>
<b>Returns</b>	<code>&lt;percent&gt;</code>
<b>Examples</b>	<code>SOURce1:PWM[:DEViation]:DCYCle 5.0</code> sets the CH 1 PWM deviation to 5.0%.

**[SOURce]:ROSCillator:SOURce**

This command sets the reference clock to either internal or external.

**Group** Source

**Syntax** [SOURce]:ROSCillator:SOURce {INTerna|EXTerna}  
[SOURce]:ROSCillator:SOURce?

**Arguments** INTerna means that the reference clock is set to internal.  
EXTerna means that the reference clock is set to external.

**Returns** INT|EXT

**Examples** SOURce:ROSCillator:SOURce INTerna  
selects the internal clock reference.

**[SOURce[1|2]]:SWEep:HTIME**

This command sets or queries the sweep hold time. Hold time represents the amount of time that the frequency must remain stable after reaching the stop frequency.

**Group** Source

**Syntax** [SOURce[1|2]]:SWEep:HTIME {<seconds>|MINimum|MAXimum}  
[SOURce[1|2]]:SWEep:HTIME? {MINimum|MAXimum}

**Arguments** <seconds>::=<NRf>[<units>]  
where  
<NRf> is the hold time in seconds.  
<units>::=[ns | μs | ms | s]

**Returns** <seconds>

**Examples** SOURce1:SWEep:HTIME 1ms  
sets the CH 1 hold time to 1 ms.

**[SOURce[1|2]]:SWEep:MODE**

The [SOURce[1|2]]:SWEep:MODE command selects auto or manual for the sweep mode for the specified channel.

The query command returns the sweep mode for the specified channel.

**Group** Source

**Related Commands** [SOURce[1|2]]:SWEep:HTIME  
[SOURce[1|2]]:SWEep:RTIME  
[SOURce[1|2]]:SWEep:TIME  
TRIGger[:SEquence]:SOURce  
TRIGger[:SEquence]:TIMer

**Syntax** [SOURce[1|2]]:SWEep:MODE {AUTO|MANual}

[SOURce[1|2]]:SWEep:MODE?

**Arguments** AUTO sets the sweep mode to auto. The instrument outputs a continuous sweep at a rate specified by Sweep Time, Hold Time, and Return Time.

MANual sets the sweep mode to manual. The instrument outputs one sweep when a trigger input is received.

**Returns** AUTO|MAN

**Examples** SOURce1:SWEep:MODE AUTO  
sets the CH1 sweep mode to auto. The instrument outputs a continuous sweep.



## [SOURce[1|2]]:SWEep:RTIME

This command sets or queries the sweep return time. Return time represents the amount of time from stop frequency through start frequency. Return time does not include hold time.

**Group** Source

**Syntax** [SOURce[1|2]]:SWEep:RTIME {<seconds>|MINimum|MAXimum}  
[SOURce[1|2]]:SWEep:RTIME? {MINimum|MAXimum}

**Arguments** <seconds>::=<NRf>[<units>]  
where  
<NRf> is the return time in seconds.  
<units>::=[ns |  $\mu$ s | ms | s]

**Returns** <seconds>

**Examples** SOURce1:SWEep:RTIME 1ms  
sets the CH 1 return time to 1 ms.

## [SOURce[1|2]]:SWEep:SPACing

The [SOURce[1|2]]:SWEep:SPACing command selects linear or logarithmic spacing for the sweep for the specified channel.

The query command returns the type for the sweep spacing for the specified channel.

**Group** Source

**Syntax** [SOURce[1|2]]:SWEep:SPACing {LINear|LOGarithmic}  
[SOURce[1|2]]:SWEep:SPACing?

**Arguments** LINear sets the sweep spacing to linear.  
LOGarithmic sets the sweep spacing to logarithmic.

**Returns** LIN|LOG

**Examples** SOURce1:SWEep:SPACing LINear  
sets the CH1 sweep spacing to linear.

**[SOURce[1|2]]:SWEep:TIME**

This command sets or queries the sweep time for the sweep for the specified channel. The sweep time does not include hold time and return time. The setting range is 10 ms to 100 s.

**Group** Source

**Syntax** [SOURce[1|2]]:SWEep:TIME {<seconds>|MINimum|MAXimum}

[SOURce[1|2]]:SWEep:TIME? {MINimum|MAXimum}

**Arguments** <seconds>::=<NRf>[<units>]  
where  
<NRf> is the sweep time in seconds.  
<units>::=[ns |  $\mu$ s | ms | s]

**Returns** <seconds>

**Examples** SOURce1:SWEep:TIME 100ms  
sets the CH 1 sweep time to 100 ms.

**[SOURce[1|2]]:VOLTage:CONCurrent[:STATe]**

This command enables or disables the function to copy the voltage level of one channel to another channel.

The [SOURce[1|2]]:VOLTage:CONCurrent[:STATe] command copies the voltage level of the channel specified by the header suffix to another channel. If you specify CH 1 with the header, the CH 1 voltage level will be copied to CH 2.

The query command returns “0” (off) or “1” (on).

If your arbitrary/function generator is a single-channel model, this command is not supported.

**Group** Source

**Syntax** [SOURce[1|2]]:VOLTage:CONCurrent[:STATe] {ON|OFF|<NR1>}  
[SOURce[1|2]]:VOLTage:CONCurrent[:STATe]?

**Arguments** ON or <NR1>≠0 enables the concurrent copy function.  
OFF or <NR1>=0 disables the concurrent copy function.

**Returns** <NR1>

**Examples** SOURce1:VOLTage:CONCurrent:STATe ON  
enables the concurrent copy function.

**[SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:HIGH**

This command sets or queries the high level of output amplitude for the specified channel. If your instrument is a dual-channel model and the [SOURce[1|2]]:VOLTage:CONCurrenT[:STATe] command is set to ON, then the high level of other channel is also the same value.

**Group** Source

**Related Commands** [SOURce[1|2]]:VOLTage:CONCurrenT[:STATe]

**Syntax** [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:HIGH {<voltage>|MINimum|MAXimum}

[SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:HIGH?{MINimum|MAXimum}

**Arguments** <voltage>::=<NRf>[<units>]  
where  
<NRf> is the high level of output amplitude.  
<units>::=[mV | V]

**Returns** <voltage>

**Examples** SOURce1:VOLTage:LEVel:IMMediate:HIGH 1V  
sets the high level of CH 1 output amplitude to 1 V.

**[SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:LOW**

This command sets or queries the low level of output amplitude for the specified channel. If your instrument is a dual-channel model and the [SOURce[1|2]]:VOLTage:CONCurent[:STATe] command is set to ON, then the low level of other channel is also the same value.

**Group** Source

**Related Commands** [SOURce[1|2]]:VOLTage:CONCurent[:STATe]

**Syntax** [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:LOW {<voltage>|MINimum|MAXimum}

[SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:LOW? {MINimum|MAXimum}

**Arguments** <voltage>::=<NRf>[<units>]  
 where  
 <NRf> is the low level of output amplitude.  
 <units>::=[mV | V]

**Returns** <voltage>

**Examples** SOURce1:VOLTage:LEVel:IMMediate:LOW -1V  
 sets the low level of CH 1 output amplitude to -1 V.

**[SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:OFFSet**

This command sets or queries the offset level for the specified channel. If your instrument is a dual-channel model and the [SOURce[1|2]]:VOLTage:CONCurent[:STATe] command is set to ON, then the offset level of the other channel is also the same value.

**Group** Source

**Related Commands** [SOURce[1|2]]:VOLTage:CONCurent[:STATe]

**Syntax** [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:OFFSet{<voltage>  
|MINimum|MAXimum}

[SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:OFFSet?{MINimum|  
MAXimum}

**Arguments** <voltage>::=<NRf>[<units>]  
where  
<NRf> is the offset voltage level.  
<units>::=[mV | V]

**Returns** <voltage>

**Examples** SOURce1:VOLTage:LEVel:IMMediate:OFFSet 500mV  
sets the CH 1 offset level to 500 mV.

## [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]

This command sets or queries the output amplitude for the specified channel. If your instrument is two channel model and the [SOURce[1|2]]:VOLTage:CONCurent[:STATe] command is set to ON, then the output amplitude of the other channel is the same value.

Units	Amplitude resolution
VPP	0.1 mV <sub>p-p</sub> or four digits
VRMS	0.1 mV <sub>rms</sub> or four digits
DBM	0.1 dBm

You can set the units of output amplitude by using either the bezel menu selection or the [SOURce[1|2]]:VOLTage:UNIT command. The selection by bezel menu has priority over the remote command.

<b>Group</b>	Source
<b>Related Commands</b>	[SOURce[1 2]]:VOLTage:CONCurent[:STATe]
<b>Syntax</b>	<p>[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]  {&lt;amplitude&gt; MINimum MAXimum}</p> <p>[SOURce[1 2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]? {MINimum MAXimum}</p>
<b>Arguments</b>	<p>&lt;amplitude&gt;::=&lt;NRf&gt;[&lt;units&gt;]  where  &lt;NRf&gt; is the output amplitude.  &lt;units&gt;::=[VPP   VRMS   DBM]</p>
<b>Returns</b>	<amplitude>
<b>Examples</b>	<p>SOURce1:VOLTage:LEVel:IMMediate:AMPLitude 1V  sets the CH 1 output amplitude to 1 V.</p>

**[SOURce[1|2]]:VOLTage:LIMit:HIGH**

This command sets or queries the higher limit of the output amplitude high level for the specified channel. If your instrument is a dual-channel model and the [SOURce[1|2]]:VOLTage:CONCurent[:STATe] command is set to ON, then the higher level limit of the other channel is the same value.

**Group** Source

**Related Commands** [SOURce[1|2]]:VOLTage:CONCurent[:STATe]

**Syntax** [SOURce[1|2]]:VOLTage:LIMit:HIGH {<voltage>|MINimum|MAXimum}

[SOURce[1|2]]:VOLTage:LIMit:HIGH? {MINimum|MAXimum}

**Arguments** <voltage>::=<NRf>[<units>]  
where  
<NRf> is the higher limit of output amplitude.  
<units>::=[mV | V]

**Returns** <voltage>

**Examples** SOURce1:VOLTage:LIMit:HIGH 1V  
sets the higher limit of CH 1 output amplitude to 1 V.



## [SOURce[1|2]]:VOLTage:LIMit:LOW

This command sets or queries the lower limit of the output amplitude low level for the specified channel. If your instrument is a dual-channel model and the [SOURce[1|2]]:VOLTage:CONCurent[:STATE] command is set to ON, then the low level lower limit of the other channel is the same value.

**Group** Source

**Related Commands** [SOURce[1|2]]:VOLTage:CONCurent[:STATE]

**Syntax** [SOURce[1|2]]:VOLTage:LIMit:LOW {<voltage>|MINimum|MAXimum}

[SOURce[1|2]]:VOLTage:LIMit:LOW? {MINimum|MAXimum}

**Arguments** <voltage>::=<NRf>[<units>]  
 where  
 <NRf> is the lower limit of output amplitude.  
 <units>::=[mV | V]

**Returns** <voltage>

**Examples** SOURce1:VOLTage:LIMit:LOW 10mV  
 sets the lower limit of CH 1 output amplitude to 10 mV.

**[SOURce[1|2]]:VOLTage:UNIT**

This command sets or queries the units of output amplitude for the specified channel. This command does not affect the offset, High level, or Low level of output. The setting of this command is not affected by the units setting of [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude] command.

$$V_{\text{rms}} = \frac{V_{\text{pp}}}{2\sqrt{2}} \text{ (sin)}$$

$$\text{dBm} = 10 \times \log_{10}\left(\frac{P}{0.001}\right) \quad P = \frac{V_{\text{rms}}^2}{R_L}$$

$$\text{RL load impedance } V_{\text{rms}} = \frac{V_{\text{pp}}}{2\sqrt{3}} \text{ (triangle)}$$

If your instrument is a dual-channel model and the [SOURce[1|2]]:VOLTage:CONCurent[:STATe] command is set to ON, then the units of the other channel are set the same.

<b>Group</b>	Source
<b>Related Commands</b>	[SOURce[1 2]]:VOLTage:CONCurent[:STATe] [SOURce[1 2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude]
<b>Syntax</b>	[SOURce[1 2]]:VOLTage:UNIT {VPP VRMS DBM}  [SOURce[1 2]]:VOLTage:UNIT?
<b>Arguments</b>	VPP sets the units of the output voltage to $V_{\text{p-p}}$ .  VRMS sets the units of the output voltage to $V_{\text{rms}}$ .  DBM sets the units of the output voltage to dBm. You cannot specify DBM if the load impedance is set to infinite.
<b>Returns</b>	VPP VRMS DBM
<b>Examples</b>	SOURce1:VOLTage:UNIT VPP sets the voltage units to $V_{\text{p-p}}$ .

**\*SRE**

This command sets and queries the bits in the Service Request Enable Register (SRER).

**Group**      Status

**Related Commands**      \*PSC

**Syntax**      \*SRE <bit\_value>  
  
                 \*SRE?

**Arguments**      <bit\_value>::=<NR1>  
                 where  
                 <NR1> is a value in the range from 0 through 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if \*PSC is set to 1. If \*PSC is set to 0, the SRER maintains the previous power cycle value through the current power cycle.

**Returns**      <bit\_value>

**Examples**      \*SRE 48  
                 sets the bits in the SRER to binary 00110000.  
  
                 \*SRE?  
                 might return 32, showing that the bits in the SRER have the binary value of 00100000.

## STATus:OPERation:CONDition?

This query-only command returns the contents of the Operation Condition Register.

<b>Group</b>	Status
<b>Syntax</b>	STATus:OPERation:CONDition?
<b>Arguments</b>	None
<b>Returns</b>	<bit_value>::=<NR1>
<b>Examples</b>	STATus:OPERation:CONDition? might return 32 which indicates that the OCR contains the binary number 00000000 00100000 and the CH 1 of the instrument is waiting for trigger.

## STATus:OPERation:ENABLE

This command sets or queries the mask for the Operation Enable Register.

<b>Group</b>	Status
<b>Syntax</b>	STATus:OPERation:ENABLE <bit_value>  STATus:OPERation:ENABLE?
<b>Arguments</b>	<bit_value>::=<NR1>
<b>Returns</b>	<bit_value>
<b>Examples</b>	STATus:OPERation:ENABLE 1 sets the CALibrating bit in the OENR to on.

**STATus:OPERation[:EVENT]?**

This query-only command returns the value in the Operation Event Register and clears the Operation Event Register.

<b>Group</b>	Status
<b>Syntax</b>	STATus:OPERation[:EVENT]?
<b>Arguments</b>	None
<b>Returns</b>	<NR1>
<b>Examples</b>	STATus:OPERation:EVENT? might return 1 which indicates that the OEVR contains the binary number 00000000 00000001 and the CALibrating bit is set to on.

**STATus:PRESet (No Query Form)**

This command presets the SCPI status registers (OENR and QENR).

<b>Group</b>	Status
<b>Syntax</b>	STATus:PRESet
<b>Arguments</b>	None
<b>Examples</b>	STATus:PRESet presets the SCPI status registers.

## STATus:QUESTionable:CONDition?

This query-only command returns the contents of the Questionable Condition Register.

<b>Group</b>	Status
<b>Syntax</b>	STATus:QUESTionable:CONDition?
<b>Arguments</b>	None
<b>Returns</b>	<bit_value>::=<NR1>
<b>Examples</b>	STATus:QUESTionable:CONDition? might return 32 which indicates that the QCR contains the binary number 00000000 00100000 and the accuracy of frequency is questionable.

## STATus:QUESTionable:ENABle

This command sets or queries the mask for the Questionable Enable Register.

<b>Group</b>	Status
<b>Syntax</b>	STATus:QUESTionable:ENABle <bit_value>  STATus:QUESTionable:ENABle?
<b>Arguments</b>	<bit_value>::=<NR1>
<b>Returns</b>	<bit_value>
<b>Examples</b>	STATus:QUESTionable:ENABle 32 sets the FREQuency bit in the QENR to on.

**STATus:QUESTionable[:EVENT]?**

This query-only command returns the value in the Questionable Event Register and clears the Questionable Event Register.

<b>Group</b>	Status
<b>Syntax</b>	STATus:QUESTionable[:EVENT]?
<b>Arguments</b>	None
<b>Returns</b>	<bit_value>::=<NR1>
<b>Examples</b>	STATus:QUESTionable[:EVENT]? might return 32 which indicates that the QEVr contains the binary number 00000000 00100000 and the FREQuency bit is set to on.

**\*STB?**

This query-only command returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit.

<b>Group</b>	Status
<b>Syntax</b>	*STB?
<b>Arguments</b>	None
<b>Returns</b>	<NR1>
<b>Examples</b>	*STB? might return 96, showing that the SBR contains the binary value 01100000.

## SYSTem:BEEPer[:IMMediate] (No Query Form)

This command causes the instrument to beep immediately.

<b>Group</b>	System
<b>Syntax</b>	SYSTem:BEEPer[:IMMediate]
<b>Arguments</b>	None
<b>Examples</b>	SYSTem:BEEPer:IMMediate causes a beep.

## SYSTem:BEEPer:STATe

The SYSTem:BEEPer:STATe command sets the beeper ON or OFF.

The SYSTem:BEEPer:STATe? command returns “0” (OFF) or “1” (ON).

When the beeper is set to ON, the instrument will beep when an error message or a warning message is displayed on the screen. The instrument does not beep when an error or warning caused by remote command execution.

<b>Group</b>	System
<b>Related Commands</b>	SYSTem:BEEPer[:IMMediate]
<b>Syntax</b>	SYSTem:BEEPer:STATe {ON OFF <NR1>}  SYSTem:BEEPer:STATe?
<b>Arguments</b>	ON or <NR1>≠0 enables the beeper.  OFF or <NR1>=0 disables the beeper.
<b>Returns</b>	<NR1>
<b>Examples</b>	SYSTem:BEEPer:STATe ON enables the beeper function.



## SYSTem:ERRor[:NEXT]?

This query-only command returns the contents of the Error/Event queue.

<b>Group</b>	System
<b>Syntax</b>	SYSTem:ERRor[:NEXT]?
<b>Arguments</b>	None
<b>Returns</b>	<Error/event number>::=<NR1> <Error/event description>::=<string>
<b>Examples</b>	SYSTem:ERRor:NEXT? might return the following response:  -410,"Query INTERRUPTED"

If the instrument detects an error or an event occurs, the event number and event message will be returned.

## SYSTem:KCLick[:STATe]

This command enables or disables the click sound when you push the front panel buttons or turn the general purpose knob. The query command returns "0" (OFF) or "1" (ON).

<b>Group</b>	System
<b>Syntax</b>	SYSTem:KCLick[:STATe] {ON OFF <NR1>}  SYSTem:KCLick[:STATe]?
<b>Arguments</b>	ON or <NR1>≠0 enables click sound.  OFF or <NR1>=0 disables click sound.
<b>Returns</b>	<NR1>
<b>Examples</b>	SYSTem:KCLick:STATe ON enables the click sound.

## SYSTem:KLOCK[:STATe]

This command locks or unlocks the instrument front panel controls. The query command returns “0” (OFF) or “1” (ON).

**Group** System

**Syntax** SYSTem:KLOCK[:STATe] {ON|OFF|<NR1>}

SYSTem:KLOCK[:STATe]?

**Arguments** ON or <NR1>≠0 locks front panel controls.

OFF or <NR1>=0 unlocks front panel controls.

**Returns** <NR1>

**Examples** SYSTem:KLOCK:STATe ON  
locks front panel controls.

## SYSTem:SECurity:IMMediate (No Query Form)

This command erases all the current instrument setups, setup memory, last setup memory, user waveform memory, and log content, and recalls the factory default settings. Calibration data is not erased.

The communication settings are initialized to the factory default settings. This might cause a remote communication error.

**Group** System

**Syntax** SYSTem:SECurity:IMMediate

**Arguments** None

**Examples** SYSTem:SECurity:IMMediate  
initializes the instrument.

## SYSTem:ULANguage

This command sets or queries the language that the instrument uses to display information on the screen.

**Group** System

**Syntax** SYSTem:ULANguage {ENGLish|FRENch|GERMan|JAPanese|KOREan|SCHinese|TCHinese}

SYSTem:ULANguage?

**Arguments** ENGLish|FRENch|GERMan|JAPanese|KOREan|SCHinese|TCHinese  
specifies which language will be used to display instrument information on the screen.

**Returns** ENGLish|FRENch|GERMan|JAPanese|KOREan|SCHinese|TCHinese

**Examples** SYSTem:ULANguage FRENch  
specifies that the instrument displays information in French.

## SYSTem:VERSion?

This query-only command returns the conformed SCPI version of the instrument.

**Group** System

**Syntax** SYSTem:VERSion?

**Arguments** None

**Returns** <SCPI Version>::=YYYY.V

where  
YYYY — indicates year.  
V — indicates the version number for that year.

**Examples** SYSTem:VERSion?  
might return 1999.0.

## TRACe|DATA:CATalog?

This query-only command returns the names of user waveform memory and edit memory.

**Group** Trace

**Syntax** TRACe|DATA:CATalog?

**Arguments** None

**Returns** <string>

A series of strings separated by commas is returned. Each string is enclosed within quotation marks.

**Examples** DATA:CATalog?  
might return "USER1","USER4","EMEM"

This example indicates that waveform data of USER2 and USER3 are deleted and not saved. Edit memory always has data.

## TRACe|DATA:COPY (No Query Form)

This command copies the contents of edit memory (or user waveform memory) to a specified user waveform memory (or edit memory).

**Group** Trace

**Syntax** TRACe|DATA:COPY <trace\_name>,EMEMory

TRACe|DATA:COPY EMEMory,{USER[1]|USER2|USER3|USER4}

**Arguments** <trace\_name>::={USER[1]|USER2|USER3|USER4}

This command is invalid when <trace\_name> is being output.

**Examples** DATA:COPY USER1,EMEMory  
copies the waveform data in the edit memory to the user waveform memory USER1.

DATA:COPY EMEMory,USER1  
copies the waveform data in the user waveform memory USER1 to the edit memory.

## TRACe|DATA[:DATA]

This command transfers the waveform data from the external controller to the edit memory in the arbitrary/function generator. The query command returns the binary block data.

**Group** Trace

**Syntax** TRACe|DATA[:DATA] EMEMoY,<binary\_block\_data>

TRACe|DATA[:DATA]? EMEMoY

**Arguments** <binary\_block\_data>  
where <binary\_block\_data> is the waveform data in binary format.

**Returns** <binary\_block\_data>

**Examples** DATA:DATA EMEMoY,#42000<DAB><DAB>...<DAB>  
transmits a waveform to the edit memory in the arbitrary/function generator. The block data element #42000 indicates that 4 is the number of digits in 2000 (byte count) and the 2000 bytes of binary data are to be transmitted.

## TRACe|DATA[:DATA]:LINE (No Query Form)

This command writes line data to the edit memory. The data between the specified points is interpolated linearly.

**Group** Trace

**Syntax** TRACe|DATA[:DATA]:LINE EMEMory,<start\_point>,<point\_data1>,<end\_point>,<point\_data2>

**Arguments** <start\_point>::=<NR1>  
where <NR1> is the first point from which the data is interpolated linearly.

<point\_data1>::=<NR1>  
where <NR1> is the data value at the start point.

<end\_point>::=<NR1>  
where <NR1> is the last point from which the data is interpolated linearly.

<point\_data2>::=<NR1>  
where <NR1> is the data value at the end point.

**Examples** DATA:DATA:LINE EMEMory,1,2047,250,4094  
sets a data value of 2047 for start point 1 and a data value of 4094 for end point 250, and interpolates linearly between these two points in the edit memory.

## TRACe|DATA[:DATA]:VALue

This command sets or queries the data value at the specified point in the edit memory.

**Group** Trace

**Syntax** TRACe|DATA[:DATA]:VALue EMEory,<point>,<data>

TRACe|DATA[:DATA]:VALue? EMEory,<points>

**Arguments** <point>::=<NR1>  
where <NR1> is the specified point number in the edit memory.

<data>::=<NR1>  
where <NR1> is the data value for the specified point number.

**Returns** <NR1>

**Examples** DATA:DATA:VALue EMEory,500,2047  
sets the data value to 2047 for the point number 500 in the edit memory.

DATA:DATA:VALue? EMEory,500  
might return “2047”

This example indicates that the data value of point number 500 is set to 2047.

## TRACe|DATA:DEFine (No Query Form)

This command resets the contents of edit memory.

**Group** Trace

**Syntax** TRACe|DATA:DEFine EMEMoRY[, {<points>|<trace\_name>}]

**Arguments** <points>::=<NR1>  
where <NR1> is the number of points for the waveform data in the edit memory that ranges from 2 to 65536 for AFG3021 and AFG3022, and 2 to 131072 for AFG3101, AFG3102, AFG3251, and AFG3252.

If the second parameter in the argument is a numerical value, the length of the edit memory will be the number of points specified by this number and each point will be initialized to the default value (8191).

<trace\_name>::={SINusoid|SQUare|PULSe|RAMP|NOISe}

If the second parameter in the argument is specified by <trace\_name>, the specified waveform data will be copied to the edit memory. The number of points for the specified waveform data is equal to the number of points for one period of current waveform data in the edit memory.

If the <points> and <trace\_name> parameters in the argument are omitted, the edit memory will be initialized to the default number of points (1000) and value (8191).

**Examples** DATA:DEFine EMEMoRY,1000  
sets the length of the edit memory to 1000 points and resets the data points to the default value.



## TRACe|DATA:DELeTe[:NAME] (No Query Form)

This command deletes the contents of specified user waveform memory.

**Group** Trace

**Syntax** TRACe|DATA:DELeTe <trace\_name>

**Arguments** <trace\_name>::={USER[1]|USER2|USER3|USER4}

This command is invalid when <trace\_name> is being output, or <trace\_name> is locked.

**Examples** DATA:DELeTe:NAME USER1  
deletes the contents of USER1 waveform memory.

## TRACe|DATA:LOCK[:STATe]

This command sets or queries whether to lock or unlock the user waveform memory.

**Group** Trace

**Syntax** TRACe|DATA:LOCK[:STATe] {USER[1]|USER2|USER3|USER4},{ON|OFF|<NR1>}

TRACe|DATA:LOCK[:STATe]?{USER[1]|USER2|USER3|USER4}

**Arguments** ON or <NR1>≠0 locks the specified user waveform memory.

OFF or <NR1>=0 unlocks the specified user waveform memory.

**Returns** <NR1>

**Examples** DATA:LOCK:STATe USER1,ON  
locks the USER1 waveform memory.

## TRACe|DATA:POINTs

This command sets or queries the number of data points for the waveform created in the edit memory.

**Group** Trace

**Syntax** TRACe|DATA:POINTs EMEMoRY[,<points>|MINimum|MAXimum]

TRACe|DATA:POINTs? EMEMoRY{,MIN|MAX}

**Arguments** <points>::=<NR1>  
where <NR1> sets the number of points for the waveform created in the edit memory that ranges from 2 to 65536 for the AFG3021 and AFG3022, and 2 to 131072 for the AFG3101, AFG3102, AFG3251, and AFG3252.

**Returns** <NR1>

**Examples** DATA:POINTs EMEMoRY, 500  
sets the waveform data points to 500 in the edit memory.

## \*TRG (No Query Form)

This command generates a trigger event.

**Group** Trigger

**Related Commands** TRIGger[:SEquence] [:IMMediate]

**Syntax** \*TRG

**Arguments** None

**Examples** \*TRG  
generates a trigger event.

## TRIGger[:SEquence]:SLOPe

This command sets or queries the slope of trigger signal.

**Group** Trigger

**Syntax** TRIGger[:SEquence]:SLOPe {POSitive|NEGative}

TRIGger[:SEquence]:SLOPe?

**Arguments** POSitive indicates that the event occurs on the rising edge of the external trigger signal.

NEGative indicates that the event occurs on the falling edge of the external trigger signal.

**Returns** POS|NEG

**Examples** TRIGger:SEquence:SLOPe POSitive  
sets the trigger slope to positive, which triggers on the rising edge of the signal.

## TRIGger[:SEquence]:SOURce

This command sets or queries the trigger source for an external trigger signal.

**Group** Trigger

**Syntax** TRIGger[:SEquence]:SOURce {TIMer|EXTernal}

TRIGger[:SEquence]:SOURce?

**Arguments** TIMer specifies an internal clock as the trigger source.

EXTernal specifies an external trigger input as the trigger source.

**Returns** TIM|EXT

**Examples** TRIGger:SEquence:SOURce EXTernal  
sets an external trigger input as the trigger source.

## TRIGger[:SEquence]:TImer

This command sets or queries the period of an internal clock when you select the internal clock as the trigger source with the TRIGger[:SEquence]:SOURce command.

**Group** Trigger

**Related Commands** TRIGger[:SEquence]:SOURce

**Syntax** TRIGger[:SEquence]:TImer <seconds>  
TRIGger[:SEquence]:TImer?

**Arguments** <seconds>::=<NRf>[<units>]  
where  
<units>::=[ $\mu$ s | ms | s]

**Returns** <seconds>

**Examples** TRIGger:SEquence:TImer 5ms  
sets the internal trigger rate to 5 ms.

## TRIGger[:SEquence][:IMMediate] (No Query Form)

This command forces a trigger event to occur.

**Group** Trigger

**Syntax** TRIGger[:SEquence][:IMMediate]

**Arguments** None

**Examples** TRIGger:SEquence:IMMediate  
generates a trigger event.

**\*TST?**

This command performs a self-test and returns the results.

---

**NOTE.** *The self-test can take several minutes to complete. During this time, the arbitrary/function generator does not execute any commands. Do not power off the instrument during the self-test.*

---

<b>Group</b>	Calibration and Diagnostic
<b>Related Commands</b>	DIAGnostic[:ALL]?
<b>Syntax</b>	*TST?
<b>Arguments</b>	None
<b>Returns</b>	<p>&lt;NR1&gt;            where            &lt;NR1&gt;=0 indicates that the self-test completed without errors.</p> <p>&lt;NR1&gt;≠0 indicates that the arbitrary/function generator detected an error.</p>
<b>Examples</b>	<p>*TST?            performs a self-test and returns 0 if no error is reported.</p>

**\*WAI (No Query Form)**

This command prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete.

<b>Group</b>	Synchronization
<b>Related Commands</b>	*OPC
<b>Syntax</b>	*WAI
<b>Arguments</b>	None
<b>Examples</b>	<p>*WAI            prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete.</p>





# Status and Events





# Status and Events

This section provides details about the status information and events the arbitrary/function generator reports.

## Status Reporting Structure

The arbitrary/function generator status reporting functions conform to IEEE-488.2 and SCPI standards. Use the status reporting function to check for instrument errors and to identify the types of events that have occurred on the instrument.

Figure 4-1 shows an outline of the instrument error and event reporting function.

The error and event reporting system consists of the following three blocks:

- Standard/Event Status
- Operation Status
- Questionable Status

The operations processed in these blocks are summarized in status bytes, which provide the error and event data.

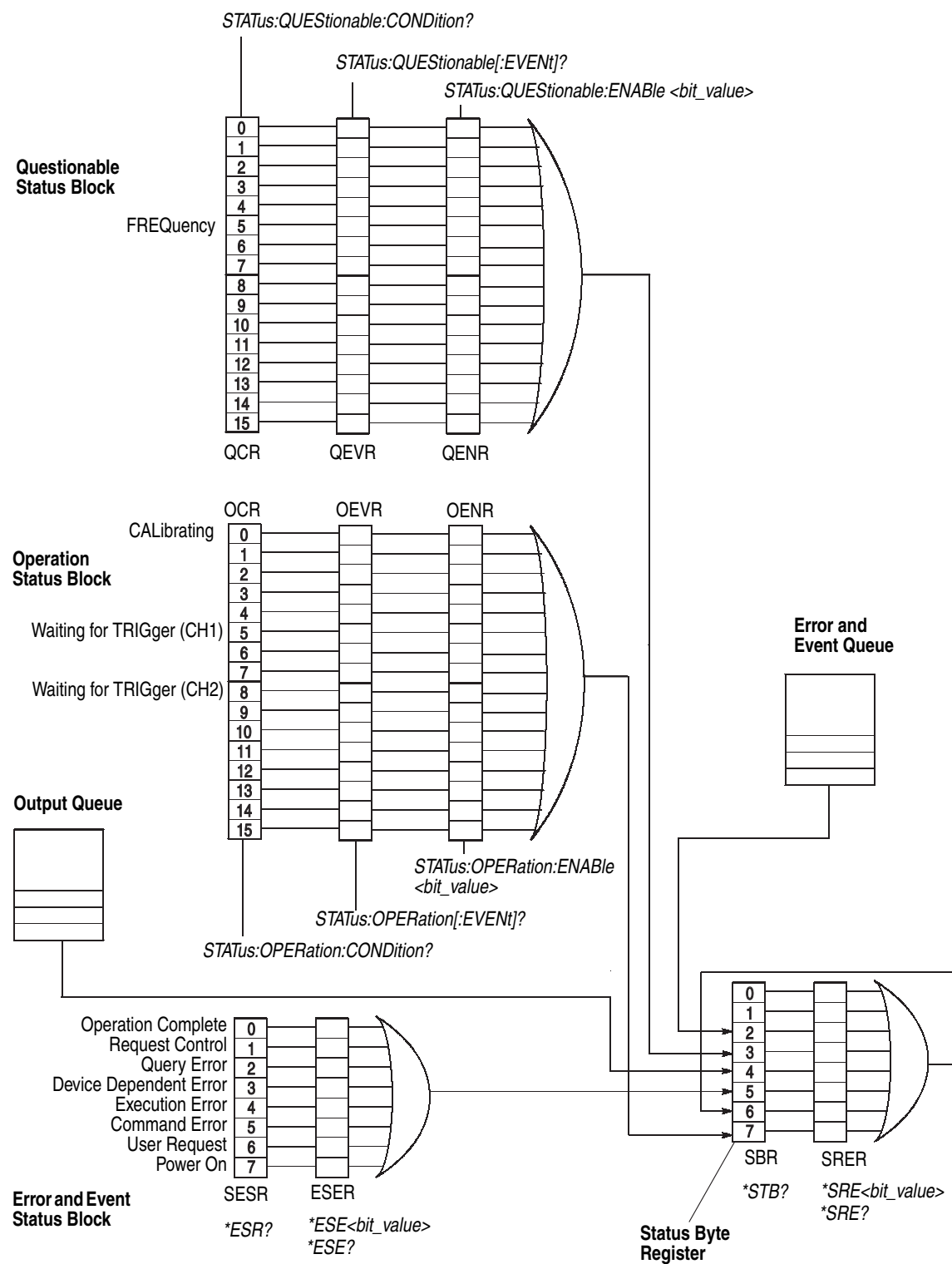


Figure 4-1: Error and event handling process

**Standard/Event Status Block**

This block is used to report power on/off, command error, and command execution status.

The block has two registers: the Standard Event Status Register (SESR) and the Event Status Enable Register (ESER). Refer to the Standard/Event Status Block shown at the bottom of Figure 4-1.

**Standard Event Status Register.** The SESR is an eight-bit status register. When an error or other type of event occurs on the instrument, the corresponding bit is set. You cannot write to this register.

**Event Status Enable Register.** The ESER is an eight-bit enable register that masks the SESR. You can set this mask, and take AND with the SESR to determine whether or not the ESB bit in the Status Byte Register (SBR) should be set.

**Operation Status Block**

This block is used to report on the status of several operations being executed by the arbitrary/function generator.

The block has three registers: the Operation Condition Register (OCR), the Operation Event Register (OEVR), and the Operation Enable Register (OENR). Refer to the Operation Status Block shown in Figure 4-1.

**Operation Condition Register.** When the instrument achieves a certain status, the corresponding bit is set to the OCR. It is not allowed for the user to write to this register.

**Operation Event Register.** The OCR bits that have changed from false (reset) to true (set) status are set in the OEVR.

**Operation Enable Register.** The function of the OENR is to mask the OEVR. You can set this mask and take AND with the OEVR to determine whether or not the OSS bit in the Status Byte Register (SBR) should be set.

**Questionable Status Block**

This block reports on the status of signals and data, such as the accuracy of entered data and signals generated by the instrument. The register configuration and process flow are the same as the Questionable Status Block.

## Registers

The registers in the event reporting system fall into two functional groups:

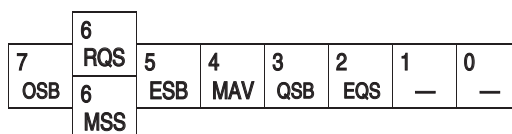
- The Status Registers contain information about the status of the instrument.
- Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue.

### **Status Registers**

There are six types of status registers:

- Status Byte Register (SBR), page 4-5
- Standard Event Status Register (SESR), page 4-6
- Operation Condition Register (OCR), page 4-7
- Operation Event Register (OEVR), page 4-7
- Questionable Condition Register (QCR), page 4-8
- Questionable Event Register (QEVr), page 4-8

**Status Byte Register (SBR).** The SBR is made up of 8 bits. Bits 4, 5 and 6 are defined in accordance with IEEE Std 488.2-1992 (see Figure 4-2). These bits are used to monitor the output queue, SESR, and service requests, respectively.



**Figure 4-2: The Status Byte Register (SBR)**

**Table 4-1: SBR bit functions**

Bit	Function	
7 (MSB)	OSB	Operation Status Bit. Indicates that an operation event has occurred.
6	RQS	Request Service. When the instrument is accessed using the GPIB serial poll command, this bit is called the Request Service (RQS) bit and indicates to the controller that a service request has occurred (in other words, that the GPIB bus SRQ line is LOW). The RQS bit is cleared when serial poll ends.
6	MSS	Master Status Summary. When the instrument is accessed using the *STB? query, this bit is called the Master Status Summary (MSS) bit and indicates that the instrument has issued a service request for one or more reasons. The MSS bit is never cleared to 0 by the *STB? query.
5	ESB	Event Status Bit. This bit indicates whether or not a new event has occurred after the previous Standard Event Status Register (SESR) has been cleared or after an event readout has been performed.
4	MAV	Message Available Bit. This bit indicates that a message has been placed in the output queue and can be retrieved.
3	QSB	Questionable Status Bit.
2	EQS	Error/Event Queue Summary.
1-0	—	Not used

**Standard Event Status Register (SESR).** The SESR records eight types of events that can occur within the instrument as shown in Figure 4-3.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

**Figure 4-3: The Standard Event Status Register (SESR)**

**Table 4-2: SESR bit functions**

Bit	Function	
7 (MSB)	PON	Power On. Indicates that the power to the instrument is on.
6	URQ	User Request. Indicates that an application event has occurred. The arbitrary/function generator does not use this bit.
5	CME	Command Error. Indicates that an error occurred while the arbitrary/function generator was parsing a command or query.
4	EXE	<p>Execution Error. Indicates that an error occurred while the arbitrary/function generator was executing a command or query.</p> <p>Execution errors occur for one of the following reasons:</p> <ul style="list-style-type: none"> <li>■ A value designated in the argument is outside the allowable range of the instrument, or is in conflict with the capabilities of the instrument.</li> <li>■ The command was not executed properly because the conditions for execution is differed from those required.</li> </ul>
3	DDE	Device Error. An instrument error has been detected.
2	QYE	<p>Query Error. Indicates that a query error has been detected by the output queue controller. Query errors occur for one of the following reasons:</p> <ul style="list-style-type: none"> <li>■ An attempt was made to retrieve messages from the output queue when the output queue is empty or in pending status.</li> <li>■ The output queue message was cleared while it was being retrieved from the output queue.</li> </ul>
1	RQC	Request Control. The arbitrary/function generator does not use this bit.
0	OPC	Operation Complete. Indicates that the operation is complete. This bit is set when all pending operations complete following the *OPC command.

**Operation Event Register (OEVR).** This register has the same content as the Operation Condition Register.

**Operation Condition Register (OCR).** The Operation Condition Register is made up of six bits, which note the occurrence of three different types of events as shown in Figure 4-4.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							WTRIG (CH2)			WTRIG (CH1)		SWE			CAL

**Figure 4-4: Operation Condition Register (OCR)**

**Table 4-3: OCR bit functions**

Bit	Function	
15 to 9	—	Not used
8	WTRIG CH2	Waiting for Trigger. Indicates whether the instrument is waiting for a trigger. This bit is set when CH 2 (in the case of dual-channel model) is waiting for a trigger. Bit is reset when the waiting-for-trigger status is canceled.
5	WTRIG CH1	Waiting for Trigger. Indicates whether the instrument is waiting for a trigger. This bit is set when CH 1 (in the case of dual-channel model) is waiting for a trigger. Bit is reset when the waiting-for-trigger status is canceled.
4	—	Not used
3	SWE	Sweep. Indicates whether the instrument is executing a frequency sweep. This bit is set when a frequency sweep is being executed on CH 1 or another channel (in the case of dual-channel model). Bit is reset when the execution stops.
2 to 1	—	Not used
0	CAL	Calibration. Indicates whether the instrument is being calibrated. This bit is set when calibration is in progress and is reset when the calibration is complete.

**Questionable Event Register (QEVR).** This register has the same content as the Questionable Condition Register.

**Questionable Condition Register (QCR).** The Questionable Condition Register is made up of sixteen bits which not the occurrence of only one type of event.

15	14	13	12	11	10	9	8	7	6	5 FREQ	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	-----------	---	---	---	---	---

**Figure 4-5: Questionable Condition Register (QCR)**

**Table 4-4: QCR bit functions**

Bit	Function
15 to 6	—— Not used
5	FREQ Frequency. Indicates whether frequency accuracy of the signal is of questionable quality.
4 to 0	—— Not used

## Enable Registers

There are four types of enable registers:

- Event Status Enable Register (ESER), page 4-8
- Service Request Enable Register (SRER), page 4-9
- Operation Enable Register (OENR), page 4-9
- Questionable Enable Register (QENR), page 4-10

Each bit in the enable registers corresponds to a bit in the controlling status register. By setting and resetting the bits in the enable register, you can determine whether or not events that occur will be registered to the status register and queue.

**Event Status Enable Register (ESER).** The ESER consists of bits defined exactly the same as bits 0 through 7 in the SESR register. You can use this register to control whether or not the Event Status Bit (ESB) in the SBR should be set when an event has occurred, and to determine if the corresponding SESR bit is set.

To set the ESB in the SBR (when the SESR bit has been set), set the ESER bit corresponding to that event. To prevent the ESB from being set, reset the ESER bit corresponding to that event.



Use the \*ESC command to set the bits in the ESER. Use the \*ESR? query to read the contents of the ESER. Figure 4-6 shows the ESER functions.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

**Figure 4-6: Event Status Enable Register (ESER)**

**Service Request Enable Register (SRER).** The SRER consists of bits defined exactly the same as bits 0 through 7 in the SBR. You can use this register to define which events will generate service requests.

The SRER bit 6 cannot be set. Also, the RQS is not maskable.

The generation of a service request with the GPIB interface involves changing the SRQ line to LOW, and making a service request to the controller. The result is that a status byte for which an RQS has been set is returned in response to serial polling by the controller.

Use the \*SRE command to set the bits of the SRER. Use the \*SRE? query to read the contents of the SRER. Bit 6 must be set to 0. Figure 4-7 shows the SRER functions.

	6						
	RQS	5	4	3	2	1	0
7		ESB	MAV	QSB	EQS	—	—
OSB	6						
	MSS						

**Figure 4-7: Service Request Enable Register (SRER)**

**Operation Enable Register (OENR).** The OENR consists of bits defined exactly the same as bits 0 through 15 in the OEVR (see Figure 4-8). You can use this register to control whether or not the Operation Status Bit (OSB) in the SBR is set when an event occurs and the corresponding OEVR bit is set.

Use the STATus:OPERation:ENABle command to set the bits in the OENR. Use the STATus:OPERation:ENABle? query to read the contents of the OENR.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
							WTRIG (CH2)			WTRIG (CH1)		SWE			CAL

**Figure 4-8: Operation Enable Register (OENR)**

**Questionable Enable Register (QENR).** The QENR consists of bits defined exactly the same as bits 0 through 15 in the QEVN register (see Figure 4-9). You can use this register to control whether the QSB in the SBR is set when an event occurs and the corresponding QEVN bit is set.

Use the `STATus:QUESTionable:ENABle` command to set the bits in the QENR. Use the `STATus:QUESTionable:ENABle?` query to read the contents of the QENR.

15	14	13	12	11	10	9	8	7	6	5 FREQ	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	-----------	---	---	---	---	---

**Figure 4-9: Questionable Enable Register (QENR)**

## Queues

There are two types of queues in the status reporting system: output queue and error/event queues.

### Output Queue

The output queue is an FIFO (first-in, first-out) queue that holds response messages to queries awaiting retrieval. When there are messages in the queue, the MAV bit in the SBR is set.

The output queue is emptied each time a command or query is received, so the controller must read the output queue before the next command or query is issued. If this is not done, an error occurs and the output queue is emptied; however, the operation proceeds even if an error occurs.

### Error/Event Queue

The event queue is an FIFO queue, which stores events as they occur in the instrument. If more than 64 events are stored, the 64th event is replaced with event code -350 (“Queue Overflow”).

The oldest error code and text are retrieved by using one of the following queries:

`SYSTem:ERRor[:NEXT]?`

First, issue the `*ESR?` query to read the contents of the SESR. The contents of the SESR are cleared after they are read. If an SESR bit is set, events are stacked in the Error/Event Queue. Retrieve the event code with the following command sequence:

`*ESR?`

`SYSTem:ERRor[:NEXT]?`

If you omit the `*ESR?` query, the SESR bit will remain set, even if the event disappears from the Error/Event Queue.

## Messages and Codes

Error and event codes with negative values are SCPI standard codes. Error and event codes with positive values are unique to the AFG3000 Series Arbitrary/Function Generators.

Table 4-5 lists event code definitions. When an error occurs, you can find its error class by checking for the code range in Table 4-6 through Table 4-14. Events in these tables are organized by event class.

**Table 4-5: Definition of event codes**

Event class	Code range	Description
No error	0	No event or status
Command errors	–100 to –199	Command syntax errors
Execution errors	–200 to –299	Command execution errors
Device-specific errors	–300 to –399	Internal device errors
Query errors	–400 to –499	System event and query errors
Power-on events	–500 to –599	Power-on events
User request events	–600 to –699	User request events
Request control events	–700 to –799	Request control events
Operation complete events	–800 to –899	Operation complete events
Extended device-specific errors	1 to 32767	Device dependent device errors
Reserved	other than above	not used

### Command Errors

Table 4-6 shows the error messages generated by improper command syntax. Check that the command is properly formed and that it follows the rules in the *Syntax and Commands* starting on page 3-1.

**Table 4-6: Command error messages**

Error code	Error message
–100	Command error
–101	Invalid character
–102	Syntax error
–103	Invalid separator
–104	Data type error
–105	GET not allowed
–108	Parameter not allowed
–109	Missing parameter
–110	Command header error
–111	Header separator error
–112	Program mnemonic too long

**Table 4-6: Command error messages (cont.)**

Error code	Error message
-113	Undefined header
-114	Header suffix out of range
-115	Unexpected number of parameters
-120	Numeric data error
-121	Invalid character in number
-123	Exponent too large
-124	Too many digits
-128	Numeric data not allowed
-130	Suffix error
-131	Invalid suffix
-134	Suffix too long
-138	Suffix not allowed
-140	Character data error
-141	Invalid character data
-144	Character data too long
-148	Character data not allowed
-150	String data error
-151	Invalid string data
-158	String data not allowed
-160	Block data error
-161	Invalid block data
-168	Block data not allowed
-170	Expression error
-171	Invalid expression
-178	Expression data not allowed
-180	Macro error
-181	Invalid outside macro definition
-183	Invalid inside macro definition
-184	Macro parameter error

**Execution Errors** Table 4-7 lists the errors that are detected during execution of a command.

**Table 4-7: Execution error messages**

Error code	Error message
-200	Execution error
-201	Invalid while in local
-202	Settings lost due to RTL
-203	Command protected
-210	Trigger error
-211	Trigger ignored
-212	Arm ignored
-213	Init ignored
-214	Trigger deadlock
-215	Arm deadlock
-220	Parameter error
-221	Settings conflict
-222	Data out of range
-223	Too much data
-224	Illegal parameter value
-225	Out of memory
-226	Lists not same length
-230	Data corrupt or stale
-231	Data questionable
-232	Invalid format
-233	Invalid version
-240	Hardware error
-241	Hardware missing
-250	Mass storage error
-251	Missing mass storage
-252	Missing media
-253	Corrupt media
-254	Media full
-255	Directory full
-256	File name not found
-257	File name error
-258	Media protected
-260	Expression error
-261	Math error in expression
-270	Macro error
-271	Macro syntax error
-272	Macro execution error
-273	Illegal macro label

**Table 4-7: Execution error messages (cont.)**

Error code	Error message
-274	Macro parameter error
-275	Macro definition too long
-276	Macro recursion error
-277	Macro redefinition not allowed
-278	Macro header not found
-280	Program error
-281	Cannot create program
-282	Illegal program name
-283	Illegal variable name
-284	Program currently running
-285	Program syntax error
-286	Program runtime error
-290	Memory use error
-291	Out of memory
-292	Referenced name does not exist
-293	Referenced name already exists
-294	Incompatible type

**Device Specific Errors**

Table 4-8 lists the device-specific errors that can occur during arbitrary/function generator operation. These errors may indicate that the instrument needs repair.

**Table 4-8: Device-specific error messages**

Error code	Message
-300	Device specific error
-310	System error
-311	Memory error
-312	PUD memory lost
-313	Calibration memory lost
-314	Save/recall memory lost
-315	Configuration memory lost
-320	Storage fault
-321	Out of memory
-330	Self-test failed
-340	Calibration failed
-350	Queue overflow
-360	Communication error
-361	Parity error in program message
-362	Framing error in program message
-363	Input buffer overrun
-365	Time out error

**Query Errors**

Table 4-9 lists the error codes that are returned in response to an unanswered query.

**Table 4-9: Query errors**

Error codes	Message
-400	query error
-410	query INTERRUPTED
-420	query UNTERMINATED
-430	query DEADLOCKED
-440	query UNTERMINATED after indefinite response

**Power-on Events**

These events occur when the instrument detects an off to on transition in its power supply.

**Table 4-10: Power-on events**

Event code	Event message
-500	Power on

**User Request Events**

These events are not used in this instrument.

**Table 4-11: User request events**

Event code	Event message
-600	User request

**Request Control Events**

These events are not used in this instrument.

**Table 4-12: Request control events**

Event code	Event message
-700	Request control

**Operation Complete Events**

These events occur when instrument's synchronization protocol, having been enabled by an \*OPC command, completes all selected pending operations.

**Table 4-13: Operation complete events**

Event code	Event message
-800	Operation complete



**Device Errors**

Table 4-14 lists the error codes that are unique to the AFG3000 Series Arbitrary/Function Generators.

**Table 4-14: Device errors**

Error code	Error message
1101	Calibration failed; CH1 Internal offset
1102	Calibration failed; CH2 Internal offset
1103	Calibration failed; CH1 Output offset
1104	Calibration failed; CH2 Output offset
1105	Calibration failed; CH1 Output gain
1106	Calibration failed; CH2 Output gain
1201	Calibration failed; CH1 x 3 dB attenuator
1202	Calibration failed; CH2 x 3 dB attenuator
1203	Calibration failed; CH1 x 6 dB attenuator
1204	Calibration failed; CH2 x 6 dB attenuator
1205	Calibration failed; CH1 x 10 dB attenuator
1206	Calibration failed; CH2 x 10 dB attenuator
1207	Calibration failed; CH1 x 20 dB 1 attenuator
1208	Calibration failed; CH2 x 20 dB 1 attenuator
1209	Calibration failed; CH1 x 20 dB 2 attenuator
1210	Calibration failed; CH2 x 20 dB 2 attenuator
1211	Calibration failed; CH1 Filter
1212	Calibration failed; CH2 Filter
1301	Calibration failed; CH1 Sine Flatness
1302	Calibration failed; CH2 Sine Flatness
1401	Calibration failed; CH1 ASIC TINT
1402	Calibration failed; CH2 ASIC TINT
1403	Calibration failed; CH1 ASIC SGEN
1404	Calibration failed; CH2 ASIC SGEN
1405	Calibration failed; CH1 ASIC clock duty
1406	Calibration failed; CH2 ASIC clock duty
2100	Self-test failed; Calibration data not found
2101	Self-test failed; Calibration data checksum
2102	Self-test failed; Calibration data invalid
2201	Self-test failed; ASIC 1 memory
2202	Self-test failed; ASIC 2 memory
2203	Self-test failed; ASIC 1 overheat
2204	Self-test failed; ASIC 2 overheat
2301	Self-test failed; CH1 Internal offset
2302	Self-test failed; CH2 Internal offset
2303	Self-test failed; CH1 Output offset
2304	Self-test failed; CH2 Output offset

**Table 4-14: Device errors (cont.)**

Error code	Error message
2305	Self-test failed; CH1 Output gain
2306	Self-test failed; CH2 Output gain
2401	Self-test failed; CH1 x 3 dB attenuator
2402	Self-test failed; CH2 x 3 dB attenuator
2403	Self-test failed; CH1 x 6 dB attenuator
2404	Self-test failed; CH2 x 6 dB attenuator
2405	Self-test failed; CH1 x 10 dB attenuator
2406	Self-test failed; CH2 x 10 dB attenuator
2407	Self-test failed; CH1 x 20 dB 1 attenuator
2408	Self-test failed; CH2 x 20 dB 1 attenuator
2409	Self-test failed; CH1 x 20 dB 2 attenuator
2410	Self-test failed; CH2 x 20 dB 2 attenuator
2411	Self-test failed; CH1 Filter
2412	Self-test failed; CH2 Filter
2501	Self-test failed; CH1 Sine Flatness
2502	Self-test failed; CH2 Sine Flatness
9112	Waveform error; invalid waveform length
9113	Waveform error; waveform length is too short



# Programming Examples



# Programming Examples

The following two example programs, which demonstrate methods that you can use to control the arbitrary/function generator through the General Purpose Interface Bus (GPIB), are included on the *AFG3000 Series Arbitrary/Function Generators Product CD*.

- Example 1: Set up a Waveform Output
- Example 2: Waveform Transfer and Copy

The example programs are written in Microsoft Visual Basic Version 6.0. The programs run on Windows PC compatible systems equipped with TekVISA and a National Instruments GPIB board with the associated drivers.

TekVISA is the Tektronix implementation of the VISA Application Programming Interface (API). TekVISA is industry-compliant software for writing interoperable instrument drivers in a variety of Application Development Environments (ADEs).

The example programs assume that the GPIB system recognizes the PC (external controller) as GPIB0, and the address number of the instrument as 11.

If you use an interface other than GPIB, change the resource name of source code. Refer to TekVISA manual for details about resource.

**Example 1** This is a sample program for setting the arbitrary/function generator outputs.

```
Private Sub Sample1_Click()
/
'Assign resource
/
Tvc1.Descriptor = "GPIB0::11::INSTR"
/
'Initialize of device setting
/
Tvc1.WriteString ("*RST")
/
'Set CH1 output parameters
/
Tvc1.WriteString ("FUNCTION SIN") 'Set output waveform SIN
Tvc1.WriteString ("FREQUENCY 10E3") 'Set frequency 10kHz
Tvc1.WriteString ("VOLTAGE:AMPLITUDE 2.00") 'Set amplitude 2Vpp
Tvc1.WriteString ("VOLTAGE:OFFSET 1.00") 'Set offset 1V
Tvc1.WriteString ("PHASE:ADJUST 0DEG") 'Set phase 0degree
/
'Set CH2 output parameters
/
Tvc1.WriteString ("SOURCE2:FUNCTION SIN") 'Set output waveform SIN
Tvc1.WriteString ("SOURCE2:FREQUENCY 10E3") 'Set frequency 10kHz
Tvc1.WriteString ("SOURCE2:VOLTAGE:AMPLITUDE 1.00") 'Set amplitude 1Vpp
Tvc1.WriteString ("SOURCE2:VOLTAGE:OFFSET 0.00") 'Set offset 0V
Tvc1.WriteString ("SOURCE2:PHASE:ADJUST 90DEG") 'Set phase 90degrees
/
'Save settings and output on
/
Tvc1.WriteString ("*SAV 1") 'Save settings to Setup1
Tvc1.WriteString ("*RCL 1") 'Recall settings from Setup1

End Sub
```

**Example 2** This is a sample program for sending an arbitrary waveform to the arbitrary/function generator's Edit Memory and copying the contents of Edit Memory to the user waveform memory.

```
Private Sub Sample2_Click()
    '
    'Assign resource
    '
    Tvc1.Descriptor = "GPIB0::11::INSTR"

    'Initialize of device setting
    '
    Tvc1.WriteString ("*RST")

    'Make arbitrary block data (2000 Points)
    '
    Dim wave(4000) As Byte

    For i = 0 To 499 'Leading edge (500 Points)
        Data = i * Int(16382 / 500) 'Data range is from 0 to 16382
        High = Int(Data / 256) 'AFG's Data Format is big endian
        Low = Data - (High * 256)

        wave(2 * i) = High
        wave(2 * i + 1) = Low
    Next i

    For i = 500 To 799 'Part of High Level (800 Points)
        Data = 16382
        High = Int(Data / 256)
        Low = Data - (High * 256)

        wave(2 * i) = High
        wave(2 * i + 1) = Low
    Next i

    For i = 800 To 999 'Trailing Edge (200 Points)
        Data = (1000 - i) * Int(16382 / 200)
        High = Int(Data / 256)
        Low = Data - (High * 256)

        wave(2 * i) = High
        wave(2 * i + 1) = Low
    Next i
End Sub
```

```

Next i

For i = 1000 To 1999 'Part of Low Level (1000 Points)
    Data = 0
    High = Int(Data / 256)
    Low = Data - (High * 256)

    wave(2 * i) = High
    wave(2 * i + 1) = Low
Next i

/
'Transfer waveform
' Transfer arbitrary block data to edit memory
/

Tvc1.SendEndEnabled = False
Tvc1.WriteString ("TRACE:DATA EMEMORY,#44000")

Tvc1.SendEndEnabled = True
Tvc1.WriteByteArray (wave)
/
'Copy contents of edit memory to USER1
/

Tvc1.WriteString ("TRAC:COPY USER1,EMEM")

/
'Set CH1 output parameters
/

Tvc1.WriteString ("FUNCTION USER1") 'Set output waveform USER1
Tvc1.WriteString ("FREQUENCY 8K") 'Set frequency 8kHz
Tvc1.WriteString ("OUTPUT ON") 'Set CH1 output on

End Sub

```





# Appendices



# Appendix A: Accessories and Options

This section lists the standard and optional accessories available for the instrument, as well as the product options.

## Options

The following options can be ordered for the instrument:

**Table A-1: Options**

Option	Description
Power cord options	
Option A0	North America
Option A1	Universal Euro
Option A2	United Kingdom
Option A3	Australia
Option A5	Switzerland
Option A6	Japan
Option A10	China
Option A99	No power cord or AC adapter
Manual options	
Option L0	English Manual
Option L1	French Manual
Option L2	Italian Manual
Option L3	German Manual
Option L4	Spanish Manual
Option L5	Japanese Manual
Option L7	Simplified Chinese Manual
Option L8	Traditional Chinese Manual
Option L9	Korean Manual
Option L10	Russian Manual
Option L99	No printed Manual
Service options	
Option C3	Calibration Service 3 Years
Option C5	Calibration Service 5 Years
Option D1	Calibration Data Report
Option D3	Calibration Data Report 3 Years (with Option C3)
Option D5	Calibration Data Report 5 Years (with Option C5)
Option R5	Repair Service 5 Years

## Accessories

All accessories (standard and optional) are available by contacting your local Tektronix field office. Table A-2 lists standard accessories for this instrument.

**Standard** The following accessories are shipped with the instrument:

**Table A-2: Standard accessories**

Accessory	Part number
AFG3000 Series Quick Start User Manual A single printed manual is included. Refer to Table A-3 for a complete list of available language manuals.	—
AFG3000 Series Product CD	063-3828-xx
ArbExpress Software CD	063-3763-xx
Power Cord	—

**Optional** The accessories in Table A-3 are orderable for use with the instrument at the time this manual was originally published. Check the Tektronix Web site ([www.tektronix.com](http://www.tektronix.com)) for the most current information.

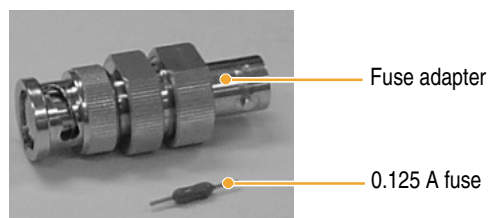
**Table A-3: Optional accessories**

Accessory	Part number
AFG3000 Series Quick Start User Manual	
English	071-1631-xx
French	071-1632-xx
Italian	071-1669-xx
German	071-1633-xx
Spanish	071-1670-xx
Japanese	071-1634-xx
Simplified Chinese	071-1635-xx
Traditional Chinese	071-1636-xx
Korean	071-1637-xx
Russian	071-1638-xx
AFG3000 Series Reference Manual (English)	071-1639-xx
AFG3000 Series Service Manual (English)	071-1640-xx
RM3100 Rackmount Kit	RM3100

**Table A-3: Optional accessories (cont.)**

Accessory	Part number
International Power Cord	
Option A0, North American	161-0066-00
Option A1, European	161-0099-09
Option A2, United Kingdom	161-0099-10
Option A3, Australian	161-0099-13
Option A5, Switzerland	161-0154-00
Option A6, Japanese	161-0298-00
Option A10, China	161-0304-00
50 $\Omega$ BNC Termination	011-0049-01
50 $\Omega$ BNC Cable, 91 cm (36 in), W shield	012-0482-00
50 $\Omega$ BNC Cable, 250 cm (98 in), W shield	012-1256-00
Fuse adapter, BNC-P to BNC-R	013-0345-00
Fuse set, 3 pcs, 0.125 A	159-0454-00

**Fuse Adapter.** The arbitrary/function generator will be damaged if a large DC or AC voltage is applied to the output or input connectors. To protect the output circuits, a fuse adapter is provided as an optional accessory. When the instrument is used by students or other inexperienced users, always attach the fuse adapter to the output connectors to avoid damage.

**Figure A-1: Fuse adapter**



# Appendix B: General Care and Cleaning

## General Care

Do not store or leave the arbitrary/function generator where the LCD display will be exposed to direct sunlight for long periods of time.



---

**CAUTION.** To avoid damage to the arbitrary/function generator, do not expose the instrument to sprays, liquids, or solvents.

---

## Cleaning

Inspect the arbitrary/function generator as often as operating conditions require. To clean the exterior surface, perform the following steps:

1. Remove loose dust on the outside of the instrument with a lint-free cloth. Use care to avoid scratching the clear plastic display filter.
2. Use a soft cloth dampened with water to clean the instrument. Use an aqueous solution of 75% isopropyl alcohol for more efficient cleaning.



---

**CAUTION.** To avoid damage to the surface of the instrument, do not use any abrasive or chemical cleaning agents.

---





## Appendix C: SCPI Conformance Information

All commands in the arbitrary/function generator are based on SCPI Version 1999.0. Table C-1 lists the SCPI commands the arbitrary/function generator supports.

**Table C-1: SCPI conformance information**

Command				Defined in SCPI 1999.0	Not defined in SCPI 1999.0
ABORt				√	
CALibration	[ALL](?)			√	
DIAGnostic	[ALL](?)				√
DISPlay	CONTrast(?)			√	
	SAVer	[STATe](?)			√
	[WINDow]	TEXT	[DATA](?)	√	
	CLEar			√	
AFGControl	CSCopy				√
MEMory	STATe	VALid?			√
		DELeTe			√
		LOCK(?)			√
		RECall	AUTo(?)		√
MMEMory	CATalog?			√	
	CDIRectory(?)			√	
	DELeTe			√	
	LOAD	STATe		√	
		TRACe		√	
	LOCK	[STATe](?)			√
	MDIRectory				√
	STORe	STATe		√	
		TRACe		√	
OUTPut[1 2]	IMPedance(?)			√	
	POLarity(?)			√	
	[STATe](?)			√	
	TRIGger	MODE(?)			√
[SOURce]	ROSCillator	SOURce(?)		√	
[SOURce[1 2]]	VOLTage	CONCurent	[STATe](?)		√
	AM	STATe(?)		√	
		INTernal	FREQuency(?)	√	
		FUNCTion(?)			√
			EFILe(?)		√

Table C-1: SCPI conformance information (cont.)

Command			Defined in SCPI 1999.0	Not defined in SCPI 1999.0
BURSt	SOURce(?)		√	
	[DEPTh](?)		√	
	MODE(?)			√
	NCYCles(?)			√
	TDELay(?)			√
	[STATe](?)			√
COMBine	FEED(?)		√	
FM	INTErnal	FREQuency(?)	√	
		FUNCtion(?)		√
		EFILe(?)		√
FREQuency	SOURce(?)		√	
	STATe(?)		√	
	[DEViation](?)		√	
	CENTer(?)		√	
	CONCurent	[STATe](?)		√
	MODE(?)		√	
	SPAN(?)		√	
	STARt(?)		√	
	STOP(?)		√	
	[CW FIXed](?)		√	
FSKey	INTErnal	RATE(?)		√
		SOURce(?)		√
		STATe(?)		√
		[FREQuency](?)		√
FUNCtion	EFILe(?)			√
	RAMP	SYMMetry(?)		√
	[SHAPE](?)		√	
PHASe	INITiate			√
	[ADJust](?)		√	
PM	INTErnal	FREQuency(?)	√	
		FUNCtion(?)		√
		EFILe(?)		√
	SOURce(?)		√	
	STATe(?)		√	
	[DEViation](?)		√	
	PULSe			
PULSe	DCYCLe(?)		√	
	DELay(?)		√	
	HOLD(?)		√	

Table C-1: SCPI conformance information (cont.)

Command					Defined in SCPI 1999.0	Not defined in SCPI 1999.0
PWM	PERiod(?)	TRANSition	TRAIling(?)	[LEADing](?)	√	
					√	
		WIDTh(?)	INTErnal	FREQuency(?)	√	
						√
	FUNction(?)	EFILe(?)	SOURce(?)	STATe(?)		√
						√
		[DEViation]	DCYCLe(?)	HTIME(?)		√
						√
	MODE(?)	RTIME(?)	SPACing(?)	TIME(?)	√	
					√	
	VOLTage	LIMit	HIGH(?)	LOW(?)	√	
					√	
		UNIT(?)	[LEVel]	[IMMediate]		√
SOURce<3 4>	POWER	[LEVel]	[IMMediate]	[AMPLitude](?)	√	
					√	
	STATus	OPERation	[EVENT]?	CONDition?	√	
					√	
		PRESet	QUESTionable	[EVENT]?	√	
					√	
	SYSTem	BEEPPer	STATe(?)	[IMMediate]	√	
						√
		ERRor	[NEXT]?	[STATe](?)	√	
						√
	KLOCK	SECurity	IMMediate	ULANguage(?)	√	
						√
		VERSion?			√	

**Table C-1: SCPI conformance information (cont.)**

Command			Defined in SCPI 1999.0	Not defined in SCPI 1999.0
TRACe DATA	CATalog?		√	
	COPY		√	
	[DATA](?)		√	
		LINE	√	
		VALue(?)	√	
	DEFine		√	
	DElete	[NAME]	√	
	LOCK	[STATE](?)		√
	POINts(?)		√	
TRIGger	[SEquence]	SLOPe(?)	√	
		SOURce(?)	√	
		TIMer(?)	√	
		[IMMediate]	√	
*CAL?				√
*CLS			√	
*ESE(?)			√	
*ESR?			√	
*IDN?			√	
*OPC(?)			√	
*OPT?				√
*PSC(?)				√
*RCL				√
*RST			√	
*SAV				√
*SRE(?)			√	
*STB?			√	
*TRG				√
*TST?			√	
*WAI			√	

## Appendix D: Default Setup

Table D-1 lists the settings that are restored when you push the front-panel **Default** button.

**Table D-1: Default settings**

Menu or System	Default setting
Output configuration	
Function	Sine
Frequency	1.000 000 000 00 MHz
Amplitude	1.000 V <sub>p-p</sub>
Offset	0 mV
Symmetry (Ramp)	50.0%
Duty (Pulse)	50.0%
Output Units	V <sub>p-p</sub>
Output Impedance	50 Ω
Output Invert	Off
Output Noise Add	Off
External Add	Off
Modulation	
Modulation Waveform	10.00 kHz, Sine (except FSK)
Modulation Waveform	10.00 kHz, Square (FSK)
AM Depth	50.0%
FM Deviation	1.000 000 MHz
PM Deviation	90.0 °
FSK Hop Frequency	1.000 000 MHz
FSK Rate	50.00 Hz
PWM Deviation	5.0%
Sweep	
Sweep Start Frequency	100.000 kHz
Sweep Stop Frequency	1.000 000 MHz
Sweep Time	10 ms
Sweep Hold Time	0 ms
Sweep Return Time	1 ms
Sweep Type	Linear
Sweep Mode	Repeat
Sweep Source	Internal
Trigger Slope	Positive
Trigger Interval	1.000 ms

**Table D-1: Default settings (cont.)**

Menu or System	Default setting
Burst	
Burst Mode	Triggered
Burst Count	5
Trigger Source	Internal
Trigger Delay	0.0 ns
Trigger Interval	1.000 ms
System-related settings	
Trigger Out	Trigger
Clock Reference	Internal

The front-panel **Default** button does not reset the following settings:

- Language option
- Power-on settings
- Display contrast
- Screen saver
- Click tone
- Beeper
- Saved setup files
- Saved waveform files
- Calibration data
- GPIB setup
- Ethernet setup



# Index





# Index

## Symbols

+/- button 1-6

## A

ABORt 3-15  
Accessories A-1  
Action buttons 1-3  
Add external signal, Output menu 2-12  
ADD INPUT connector, rear panel 1-19  
Add Noise, Output menu 2-12  
Address, Tektronix xii  
AFGControl:CSCopy 3-16  
Align Phase 2-3  
Amplitude/High button, shortcut button 1-9  
Arb button 1-8  
Arb menu 2-5  
ArbExpress 2-14

## B

Backup/Restore 2-27  
Beeper, Utility menu 2-24  
Bezel buttons 1-13, 2-1  
Bezel menu 1-11  
Bezel menu display area, screen interface 1-13  
BKSP button 1-6  
Burst Parameter Menu 2-11

## C

\*CAL? 3-16  
Calibration 2-26  
CALibration[:ALL] 3-17  
Cancel button 1-6  
    To unlock front panel controls 1-10  
CH1/CH2 Output On button 1-7  
Changing waveform parameters  
    Using the bezel menu 1-14  
    Using the shortcut buttons 1-17  
Channel Copy, Utility menu 2-25  
Channel Select button 1-7  
Chassis ground screw, rear panel 1-20  
Cleaning B-1  
Click Tone, Utility menu 2-24

Clock Ref, Utility menu 2-24

\*CLS 3-17

Command

    Abbreviating 3-6  
    Chaining 3-7  
    Message elements 3-2  
    Parameter types 3-5  
    Query response 3-3  
    Special characters 3-6  
    Units 3-7

Command entry 3-4

Command errors 4-11

Command groups 3-9

Command syntax 3-1

Connecting to a Ethernet Network 2-25

Connecting to a GPIB Network 2-25

Contacting Tektronix xii

Contrast, Utility menu 2-24

Cursors 2-18

Cut, Edit menu 2-20

## D

Data, Edit menu 2-19

DC, More waveform menu 2-6

Default button 1-6

    Control settings D-1

Delete a file 2-29

Device errors 4-17

Device specific errors 4-15

DIAGnostic[:ALL] 3-18

Diagnostics 2-26

DISPlay:CONTRast 3-19

DISPlay:SAVer[:STATe] 3-19

DISPlay[:WINDow]:TEXT:CLEar 3-20

DISPlay[:WINDow]:TEXT[:DATA] 3-20

Documentation xi

Duty/Width button, shortcut button 1-9

## E

Edit button 1-6

Edit Memory 2-5, 2-14, 2-29

Edit menu 2-14

Enter button 1-6

Erase

    Edit menu 2-15

- Save menu 2-13
- Error/Event Queue 4-10
- \*ESE 3-21
- ESER 4-8
- \*ESR? 3-22
- Event Status Enable Register 4-3, 4-8
- Execution errors 4-13
- Exponential Decay, More waveform menu 2-6
- Exponential Rise, More waveform menu 2-6
- EXT MODULATION CH 1 INPUT connector, rear panel 1-19
- EXT MODULATION CH 2 INPUT connector, rear panel 1-19
- EXT REF INPUT connector
  - Master-slave operation 2-26
  - Rear panel 1-19
- EXT REF OUTPUT connector
  - Master-slave operation 2-26
  - Rear panel 1-19
- External Add, Output menu 2-12

## F

- File management conventions 2-32
- File operations 2-28
  - To browse instrument setup files 2-31
  - To browse waveform files 2-28
  - To enter a file name (Save As) 2-32
  - To enter a file name (Write As) 2-30
- Firmware update 2-27
- Frequency/Period button, shortcut button 1-9
- Front panel controls 1-2
- Function buttons 1-8
- Fuse adapter A-3

## G

- Gated burst mode 2-11
- Gaussian, More waveform menu 2-6
- GPIO connector, rear panel 1-20

## H

- Haversine, More waveform menu 2-6
- Help button 1-6

## I

- I/O Interface, Utility menu 2-23
- \*IDN? 3-22
- Instrument options A-1

- Instrument setup file
  - Browsing 2-31
  - Saving 2-31
- Invert, Output menu 2-12

## L

- LAN connector, rear panel 1-20
- Language options A-1
- Language, Utility menu 2-23
- Last setup memory 2-13, 3-24, 3-25, 3-28, 3-36
- Leading/Trailing button, shortcut button 1-9
- Level meter 1-11, 1-12
- Limit, Output menu 2-12
- Line, Edit menu 2-19
- Load Impedance 1-10
  - Output menu 2-12
- Lock a directory 2-29
- Lock or unlock a file 2-29
- Lock/Unlock
  - Edit menu 2-15
  - Save menu 2-13
- Lorentz, More waveform menu 2-6

## M

- Main display area 1-11, 2-23
- Manual Trigger button 1-5
- MEMory:STATe:DELeTe 3-23
- MEMory:STATe:LOCK 3-24
- MEMory:STATe:RECall:AUTo 3-25
- MEMory:STATe:VALid? 3-23
- Menu buttons 1-6
- Menu structure 2-2
- Menu system 2-1
- Message display area 1-11, 1-12
- Messages and codes 4-11
- MMEMory:CATalog? 3-26
- MMEMory:CDIRectory 3-27
- MMEMory:DELeTe 3-27
- MMEMory:LOAD:STATe 3-28
- MMEMory:LOAD:TRACe 3-28
- MMEMory:MDIRectory 3-29
- MMEMory:STORE:STATe 3-30
- MMEMory:STORE:TRACe 3-30
- MMEMory:LOCK[:STATe] 3-29
- Modulation Parameter Menu 2-7
- More Waveform Menu 2-6
- More... button 1-8

## N

New, Edit menu 2-17  
 Noise  
   More waveform menu 2-6  
   Output menu 2-12  
 Number of Points, Edit menu 2-16  
 Numeric input 1-15

## O

OCR 4-7  
 OENR 4-9  
 OEVR 4-7  
 Offset/Low button, shortcut button 1-9  
 \*OPC 3-31  
 Operation complete events 4-16  
 Operation Condition Register 4-3, 4-7  
 Operation Enable Register 4-3, 4-9  
 Operation Event Register 4-3, 4-7  
 Operations submenu, Edit menu 2-17  
 \*OPT? 3-31  
 Option buttons 2-1  
 Optional accessories A-2  
 Options A-1  
 Output a user-defined waveform 2-5  
 Output menu 2-12  
   How to access 2-2  
 Output Queue 4-10  
 Output status 1-11  
 Output status, screen interface 1-11  
 OUTPut[1|2]:IMPedance 3-32  
 OUTPut[1|2]:POLarity 3-33  
 OUTPut[1|2]:STATe 3-33  
 OUTPut:TRIGger:MODE 3-34

## P

Paste at Beginning, Edit menu 2-21  
 Paste at End, Edit menu 2-22  
 Phase | Delay button, shortcut button 1-9  
 Phone, Tektronix xii  
 Power button 1-3  
 Power cord options A-1  
 Power On, Utility menu 2-13, 2-24  
 Power-on events 4-15  
 Power-on setting 2-26  
 Programming examples 5-1  
 \*PSC 3-35

Pulse button 1-8  
 Pulse menu 2-4

## Q

QCR 4-8  
 QENR 4-10  
 QEVR 4-8  
 Query errors 4-15  
 Questionable Condition Register 4-8  
 Questionable Enable Register 4-10  
 Questionable Event Register 4-8  
 Queues 4-10  
 Quick Start User Manual, part number A-2

## R

Radio button 2-1  
 Ramp button 1-8  
 Ramp menu 2-4  
 \*RCL 3-35  
 Read from..., Edit menu 2-20  
 Rear panel 1-19  
 Recall button 1-6  
 Recall menu 2-13  
 Recover Lead Delay, Pulse menu 2-4  
 Reference Manual, part number A-2  
 Registers 4-4  
 Request control events 4-16  
 \*RST 3-36  
 Run Mode buttons 1-6, 1-7

## S

Sales support, contact information xii  
 \*SAV 3-36  
 Save button 1-6  
 Save menu 2-13  
 SBR 4-5  
 SCPI  
   Conformance information C-1  
   Standard codes 4-11  
 SCPI commands 3-4  
   General rules 3-8  
 Screen interface 1-11  
 Screen Saver, Utility menu 2-24  
 Screen view formats 1-3  
 Secure 2-26  
   Utility menu 2-24

- Security slot, rear panel 1-20
- Select a channel 1-7
- Service Manual, part number A-2
- Service options A-1
- Service Request Enable Register 4-9
- Service support, contact information xii
- SESR 4-6
- Setup memory location 0 3-36
- Shortcut buttons 1-9
  - Changing waveform parameters 1-17
- Side-menu buttons 1-13, 2-1
- Sin(x)/x, More waveform menu 2-6
- Sine button 1-8
- Sine menu 2-3
- Soft keys 1-13, 2-1
  - [SOURce[1|2]]:AM[:DEPTH] 3-37
  - [SOURce[1|2]]:AM:INTernal:FREQuency 3-38
  - [SOURce[1|2]]:AM:INTernal:FUNCTion 3-39
  - [SOURce[1|2]]:AM:INTernal:FUNCTion:EFILe 3-40
  - [SOURce[1|2]]:AM:SOURce 3-40
  - [SOURce[1|2]]:AM:STATe 3-41
  - [SOURce[1|2]]:BURSt:MODE 3-41
  - [SOURce[1|2]]:BURSt:NCYCles 3-42
  - [SOURce[1|2]]:BURSt[:STATe] 3-43
  - [SOURce[1|2]]:BURSt:TDELay 3-44
  - [SOURce[1|2]]:COMBine:FEED 3-45
  - [SOURce[1|2]]:FM[:DEViation] 3-46
  - [SOURce[1|2]]:FM:INTernal:FREQuency 3-46
  - [SOURce[1|2]]:FM:INTernal:FUNCTion 3-47
  - [SOURce[1|2]]:FM:INTernal:FUNCTion:EFILe 3-48
  - [SOURce[1|2]]:FM:SOURce 3-48
  - [SOURce[1|2]]:FM:STATe 3-49
  - [SOURce[1|2]]:FREQuency:CENTer 3-49
  - [SOURce[1|2]]:FREQuency:CONCurrent[:STATe] 3-50
  - [SOURce[1|2]]:FREQuency[:CW]:FIXed 3-51
  - [SOURce[1|2]]:FREQuency:MODE 3-52
  - [SOURce[1|2]]:FREQuency:SPAN 3-53
  - [SOURce[1|2]]:FREQuency:STARt 3-54
  - [SOURce[1|2]]:FREQuency:STOP 3-55
  - [SOURce[1|2]]:FSKey[:FREQuency] 3-56
  - [SOURce[1|2]]:FSKey:INTernal:RATE 3-56
  - [SOURce[1|2]]:FSKey:SOURce 3-57
  - [SOURce[1|2]]:FSKey:STATe 3-57
  - [SOURce[1|2]]:FUNCTion:EFILe 3-58
  - [SOURce[1|2]]:FUNCTion:RAMP:SYMMetry 3-58
  - [SOURce[1|2]]:FUNCTion[:SHAPE] 3-59
  - [SOURce[1|2]]:PHASe[:ADJust] 3-60
  - [SOURce[1|2]]:PHASe:INITiate 3-61
  - [SOURce[1|2]]:PM[:DEViation] 3-61
  - [SOURce[1|2]]:PM:INTernal:FREQuency 3-62
  - [SOURce[1|2]]:PM:INTernal:FUNCTion 3-63
  - [SOURce[1|2]]:PM:INTernal:FUNCTion:EFILe 3-64
  - [SOURce[1|2]]:PM:SOURce 3-64
  - [SOURce[1|2]]:PM:STATe 3-65
  - [SOURce[1|2]]:PULSe:DCYCle 3-67
  - [SOURce[1|2]]:PULSe:DELay 3-68
  - [SOURce[1|2]]:PULSe:HOLD 3-68
  - [SOURce[1|2]]:PULSe:PERiod 3-69
  - [SOURce[1|2]]:PULSe:TRANSition[:LEADing] 3-69
  - [SOURce[1|2]]:PULSe:TRANSition:TRAILing 3-70
  - [SOURce[1|2]]:PULSe:WIDTh 3-71
  - [SOURce[1|2]]:PWM[:DEViation]:DCYCle 3-76
  - [SOURce[1|2]]:PWM:INTernal:FREQuency 3-72
  - [SOURce[1|2]]:PWM:INTernal:FUNCTion 3-73
  - [SOURce[1|2]]:PWM:INTernal:FUNCTion:EFILe 3-74
  - [SOURce[1|2]]:PWM:SOURce 3-74
  - [SOURce[1|2]]:PWM:STATe 3-75
  - [SOURce[1|2]]:SWEep:HTIME 3-77
  - [SOURce[1|2]]:SWEep:MODE 3-78
  - [SOURce[1|2]]:SWEep:RTIME 3-79
  - [SOURce[1|2]]:SWEep:SPACing 3-79
  - [SOURce[1|2]]:SWEep:TIME 3-80
  - [SOURce[1|2]]:VOLTage:CONCurrent[:STATe] 3-81
  - [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate][:AMPLitude] 3-85
  - [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:HIGH 3-82
  - [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:LOW 3-83
  - [SOURce[1|2]]:VOLTage[:LEVel][:IMMediate]:OFF-Set 3-84
  - [SOURce[1|2]]:VOLTage:LIMit:HIGH 3-86
  - [SOURce[1|2]]:VOLTage:LIMit:LOW 3-87
  - [SOURce[1|2]]:VOLTage:UNIT 3-88
  - SOURce<3|4>:POWER[:LEVel][:IMMediate][:AMPLitude] 3-66
  - [SOURce]:ROSCillator:SOURce 3-77
- Square button 1-8
- Square menu 2-3
- \*SRE 3-89
- SRER 4-9
- Standard accessories A-2

Standard Event Status Register 4-3, 4-6  
 State buttons 1-7  
 Status Byte Register 4-5  
 Status reporting structure 4-1  
 Status, Utility menu 2-24  
 STATus:OPERation:CONDition? 3-90  
 STATus:OPERation:ENABle 3-90  
 STATus:OPERation[:EVENT]? 3-91  
 STATus:PRESet 3-91  
 STATus:QUEStionable:CONDition? 3-92  
 STATus:QUEStionable:ENABle 3-92  
 STATus:QUEStionable[:EVENT]? 3-93  
 \*STB? 3-93  
 Sweep Parameter Menu 2-9  
 Synchronous operation 2-26  
 System related menu, Utility menu 2-24  
 SYSTem:BEEPer[:IMMediate] 3-94  
 SYSTem:BEEPer:STATe 3-94  
 SYSTem:ERRor[:NEXT]? 3-95  
 SYSTem:KCLick[:STATe] 3-95  
 SYSTem:KLOCK[:STATe] 3-96  
 SYSTem:SECurity:IMMediate 3-96  
 SYSTem:ULANguage 3-97

## T

Technical support, contact information xii  
 Toggle button selections 1-13  
 Top Menu button 1-3  
   To access the Output menu 2-2  
 TRACe|DATA:CATalog? 3-98  
 TRACe|DATA:COpy 3-98  
 TRACe|DATA[:DATA] 3-99  
 TRACe|DATA[:DATA]:LINE 3-100  
 TRACe|DATA[:DATA]:VALue 3-101  
 TRACe|DATA:DEFine 3-102  
 TRACe|DATA:LOCK[:STATe] 3-103  
 TRACe|DATA:POINts 3-104  
 \*TRG 3-104  
 Trigger Input connector 1-10  
 Trigger Out, Utility menu 2-24  
 Trigger Output connector 1-10  
 Triggered burst mode 2-11  
 TRIGger[:SEQuence][:IMMediate] 3-106  
 TRIGger[:SEQuence]:SLOPe 3-105  
 TRIGger[:SEQuence]:SOURce 3-105  
 TRIGger[:SEQuence]:TIMer 3-106

\*TST? 3-107

## U

Unified coarse screw 1-20  
 Unlock front panel controls 1-10  
 Upper Menu button 1-3  
 USB connector, rear panel 1-20  
 USB memory 2-28  
   Firmware Update 2-27  
 USB Memory connector, front panel 1-10  
 User request events 4-16  
 User-defined waveform 2-5  
 Using cursors 2-18  
 Using the menu system 2-1  
 Utility button 1-6  
 Utility menu 2-23

## V

View button 1-3  
 View tab, screen interface 1-11

## W

\*WAI 3-107  
 Waveform data file  
   Browsing 2-28  
   Saving 2-29  
 Waveform parameters, how to change 1-14  
 Web site, Tektronix xii  
 Write to..., Edit menu 2-22

## Z

Zoom Fit, Edit menu 2-15

