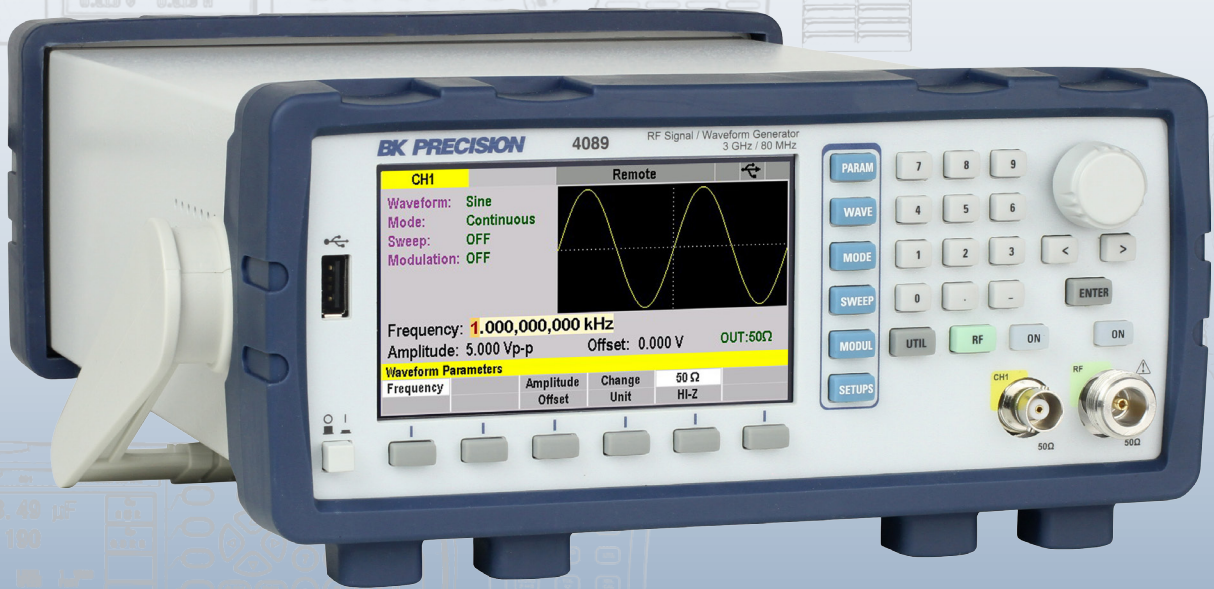


# Programming Manual

Models 4088 / 4089

RF Signal / Waveform Generator



# Contents

1	About this Manual	7
1.1	Intended Audience	8
1.2	Related Documents	8
2	Syntax Convention	9
2.1	Introduction	9
2.1.1	Types of SCPI Messages	9
2.2	Types of SCPI Commands	9
2.2.1	Common Commands	10
2.3	Syntax of Program Messages	11
2.3.1	Command Structure	11
2.3.2	Command Forms	11
2.3.3	Square Brackets	11
2.3.4	Separators and Terminators	12
2.3.5	Automatic Parameter Range Selection	12
3	SCPI Data Types	13
3.1	Data Types	14
3.1.1	Character Data Types	14
3.1.2	Numeric Data Types (<NRf>)	14
3.1.3	Boolean Data Type (<Boolean>)	14
3.1.4	String Data Type (<String>)	15
4	About Commands & Queries	16
4.1	How They are Listed	17
4.2	How They are Described	17
4.3	When can they be used?	17
4.4	Command Notation	17
5	Common Commands	18
5.1	*CLS	19
5.2	*ESE?	19
5.3	*ESR?	19
5.4	*IDN?	20
5.5	*OPC	20
5.6	*RST	20
5.7	*SRE?	21
5.8	*STB?	21

6	Source Subsystem	22
6.1	Frequency	23
6.2	Function	23
6.3	Amplitude	23
6.4	Offset	24
6.5	Clock Reference Source	24
6.6	Source Duty Cycle	25
7	AM Modulation	26
7.1	Source AM State	27
7.2	Source AM Depth	27
7.3	Source AM Shape	27
7.4	Source AM Frequency	28
7.5	Source AM Reference	28
8	FM Modulation	29
8.1	Source FM State	30
8.2	Source FM Deviation	30
8.3	Source FM Shape	30
8.4	Source FM Frequency	31
8.5	Source FM Reference	31
9	Frequency Shift Keying Modulation	32
9.1	Source FSK State	33
9.2	Source FSK Low Frequency	33
9.3	Source FSK High Frequency	33
9.4	Source FSK Rate	34
9.5	Source FSK Reference	34
10	Binary Shift Keying Modulation	35
10.1	Source BPSK State	36
10.2	Source BPSK Low Phase	36
10.3	Source BPSK High Phase	36
10.4	Source BPSK Rate	37
10.5	Source BPSK Source	37
11	Phase Modulation	38
11.1	Source PM State	39
11.2	Source PM Deviation	39
11.3	Source PM Shape	39
11.4	Source PM Frequency	40
11.5	Source PM Reference	40

12	SWEEP	41
12.1	Sweep State .....	42
12.2	Sweep Spacing .....	42
12.3	Sweep Start .....	42
12.4	Sweep Stop .....	43
13	Pulse Settings	44
13.1	Pulse Period .....	45
13.2	Pulse Width .....	45
13.3	Pulse Edge .....	45
13.4	Pulse Rise .....	46
13.5	Pulse Fall .....	46
14	Output Subsystem	47
14.1	Output State .....	48
14.2	Output Impedance .....	48
15	Trigger Subsystem	49
15.1	Trigger Mode .....	50
15.2	Trigger Source .....	50
15.3	Trigger Level .....	51
15.4	Burst Count .....	51
15.5	Internal Trigger Rate .....	51
16	Arbitrary Subsystem	52
16.1	Point Rate .....	53
16.2	Address .....	53
16.3	Data .....	54
16.4	Line Draw .....	54
16.5	Clear .....	55
16.6	Copy .....	55
16.7	Predefined Waveforms .....	56
16.8	Start Address .....	57
16.9	Wavelength .....	57
16.10	Marker Address .....	57
16.11	Marker Length .....	58
16.12	Marker State .....	58
16.13	Save .....	58
16.14	Load .....	59
17	Status Subsystem	60
17.1	Status Preset .....	61
17.2	Error Queue Read .....	62
17.3	Error Queue Enable .....	63

17.4	Questionable Status .....	63
17.5	Questionable Condition .....	64
17.6	Positive Transition Filter .....	64
17.7	Negative Transition Filter .....	65
17.8	Event Register .....	65
17.9	Event Enable Register .....	66
17.10	Status Diagram .....	66
18	System Subsystem .....	68
18.1	GPIB Address Change .....	69
18.2	Error Queue Reading .....	70
18.3	SCPI Version .....	71
18.4	Security .....	71
18.5	Power-On Buffer .....	71
19	RF Subsystem .....	72
19.1	RF Frequency .....	73
19.2	RF Amplitude .....	73
20	RF Amplitude Modulation .....	74
20.1	RF AM State .....	75
20.2	RF AM Depth .....	75
20.3	RF AM Shape .....	75
20.4	RF AM Frequency .....	76
20.5	RF AM Reference .....	76
21	RF Frequency Modulation .....	77
21.1	RF FM State .....	78
21.2	RF FM Deviation .....	78
21.3	RF FM Shape .....	78
21.4	RF FM Frequency .....	79
21.5	RF FM Reference .....	79
22	RF Phase Modulation .....	80
22.1	RF PM State .....	81
22.2	RF PM Deviation .....	81
22.3	RF PM Shape .....	81
22.4	RF PM Frequency .....	82
22.5	RF PM Reference .....	82
23	RF Pulse Modulation .....	83
23.1	RF Pulse State .....	84
23.2	RF Pulse Frequency .....	84
23.3	RF Pulse Reference .....	84

24	RF Sweep	85
24.1	Sweep State .....	86
24.2	Sweep Spacing .....	86
24.3	Sweep Time .....	86
24.4	Sweep Start .....	87
24.5	Sweep Stop .....	87

# About this Manual

This manual describes how to use the Standard Commands for Programming Instruments (SCPI) to communicate with the 4080 Series Benchtop Data Acquisition System. It provides a comprehensive reference for the command syntax, parameters, and functioning of the instrument's remote control capabilities.

1.1	Intended Audience .....	8
1.2	Related Documents .....	8

---

## 1.1 Intended Audience

---

This document is designed for instrument programmers and engineers tasked with creating SCPI-based software applications for the 4080 Series. Users should have a basic understanding of remote communication interfaces (such as USB, LAN, or GPIB) and the SCPI programming language.

---

## 1.2 Related Documents

---

For additional information regarding the operation and standards of the instrument, refer to the following documents:

- **4080 Series User's Manual:** Describes the front-panel operation, installation, and general features of the 4080 Series.
- **Standard Commands for Programming Instruments (SCPI):** Volume 1-4, Version 1990.0 (May 1999), SCPI Consortium.
- **IEEE Std 488.2-1992:** IEEE Standard Codes, Formats, Protocols, and Common Commands. The Institute of Electrical and Electronics Engineers, Inc..

# Syntax Convention

---

## 2.1 Introduction

---

SCPI (Standard Commands for Programmable Instruments) serves as a universal programming language designed for electronic test and measurement instruments. It is grounded in the IEEE 488.1 and IEEE 488.2 standards. The 4080 series aligns with the SCPI language and incorporates the IEEE 488.2 STD status structure.

The commands can be issued over USB, LAN, or GPIB interfaces.

### 2.1.1 Types of SCPI Messages

---

In order to program a 4080 instrument, it is necessary to create a program message. This message comprises one or more appropriately formatted SCPI commands transmitted from the controller to the 4080 instrument.

The program message, which can be sent at any time, requests the instrument to execute a specific action or provide data or status information. These requests are also referred to as *queries*.

Upon receiving a query, the 4080 instrument responds by sending a response message back to the controller. This response message contains data formatted in a specific SCPI format.

---

## 2.2 Types of SCPI Commands

---

Two types of SCPI commands are available: common commands, described below, and device-specific subsystem commands.

### 2.2.1 Common Commands

Common SCPI commands, as defined by IEEE 488.2, are responsible for controlling and managing generic system functions like reset, self-test, configuration storage, and device identification. Typically, common commands start with an asterisk (\*), have a length of four to five characters, and may involve one or more parameters.

The command keyword is separated from the initial parameter by a space. Multiple commands can be separated using a semicolon (;), as demonstrated below:

```
*RST; *CLS; *ESE 32; *OPC?
```

Refer to Table 2.1 for a summary of the common SCPI commands applicable to the 4080 series.

Command	Description
*CLS	Clears all Event Registers summarized in the status byte.
*ESE?	Returns an <NR1>, representing the value of the Standard Event Status Enable Register. Reading the value of the register will result in its clearance.
*ESR?	Returns an <NR1>, representing the value of the Standard Event Status Register. Reading the value of the register will result in its clearance.
*IDN?	Returns the unique identification string of the instrument.
*OPC?	Returns a "+1", when all pending selected device operations have been finished.
*RST	The Reset command performs a device reset. The Reset command is the third level of reset in a three-level reset strategy.
*SRE?	The Service Request Enable query allows the programmer to determine the current contents of the Service Request Enable Register.
*STB?	Returns the current binary value of the Status Byte Register.

**Table 2.1** Common SCPI Commands

## 2.3 Syntax of Program Messages

A program message consists of one or more properly formatted SCPI commands, a parameter (if necessary), and a terminator sent from the controller to the 4080 instrument.

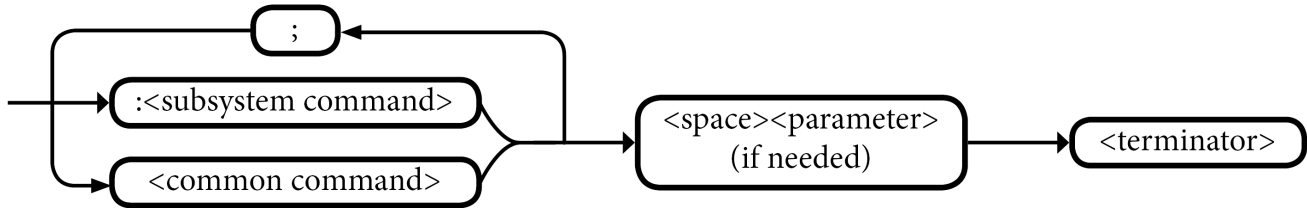


Figure 2.1 Syntax of Program Messages

### 2.3.1 Command Structure

SCPI commands follow a tree-like structure, organized into nodes. Each level of the command tree is a node. Each keyword (node) of a SCPI command is separated by a colon (:).

CONFigure:VOLTage:DC

### 2.3.2 Command Forms

Commands and queries have two different forms: long and short.

- **Long Form:** The full word (e.g., CONFigure).
- **Short Form:** An abbreviated version, typically the first three or four letters (e.g., CONF).

The commands can be written in either uppercase or lowercase. The instrument is case-insensitive. For example, CONFIGURE, configure, and Conf are all valid.

### 2.3.3 Square Brackets

Commands that contain square brackets [] indicate that the enclosed keywords or parameters are optional. The function of the command is the same with or without the bracketed items.

[SENSe:] VOLTage:DC:RANGe

Both SENSe:VOLTage:DC:RANGe and VOLTage:DC:RANGe are valid.

### **2.3.4 Separators and Terminators**

---

- **Command Separator:** A semicolon (;) is used to link multiple commands within a single program message string.
- **Parameter Separator:** A space is used to separate the command header from the first parameter. A comma (,) is used to separate multiple parameters.
- **Message Terminator:** The termination character marks the end of a command line. The 4080 series accepts the Line Feed (LF) character (<\n>) as the terminator. The IEEE 488.2 standard also defines CR, CR+LF, and LF+CR as valid terminators.

### **2.3.5 Automatic Parameter Range Selection**

---

The instrument may automatically set a command parameter to the next available value if the entered value is not exact. For example, if a specific range is requested that does not exist, the instrument selects the next highest available range.

# SCPI Data Types

SCPI defines various data types for use in program messages and response messages. The 4080 Series uses a subset of these standard data types. This section summarizes the data types accepted by the instrument.

3.1	Data Types .....	14
3.1.1	Character Data Types .....	14
3.1.2	Numeric Data Types (<NRf>) .....	14
3.1.3	Boolean Data Type (<Boolean>) .....	14
3.1.4	String Data Type (<String>) .....	15

## 3.1 Data Types

The following data types are used in SCPI commands and queries.

### 3.1.1 Character Data Types

Character data types are short, mnemonic keywords used to represent specific settings.

- They can be entered in either long or short form (e.g., IMMEDIATE or IMM).
- They are case-insensitive.
- Response messages always return the short form in uppercase.

Example: DISPLAY:PAGE {MEAS | MANUMEM | INTF}

### 3.1.2 Numeric Data Types (<NRf>)

SCPI defines three formats for numeric values, collectively referred to as <NRf>. The instrument accepts any of these formats for numeric parameters.

Data	Description
<NR1>	Integer values (e.g., 0, 100, -10).
<NR2>	Decimal values (e.g., 1.5, 10.0, 0.01).
<NR3>	Floating point values with exponents (e.g., 1.5E+3, 4.5E-6).
<NRf>	Any of the above formats (Flexible Numeric Representation).

**Table 3.1** Numeric Data Formats

### 3.1.3 Boolean Data Type (<Boolean>)

Boolean parameters represent a single binary condition (True or False).

- **\*\*ON\*\*** or **\*\*1\*\***: Represents True (Enabled).
- **\*\*OFF\*\*** or **\*\*0\*\***: Represents False (Disabled).

Example: OUTPUT:STATE ON

### 3.1.4 String Data Type (<String>)

String parameters are enclosed in double quotes (e.g., "Label Name").

- Used for labels, messages, or filenames.
- The quotes are part of the parameter syntax.

Example: DISPlay:TEXT "Testing"

# About Commands & Queries

This section lists and describes the remote control commands and queries recognized by the instrument. All commands and queries can be executed in either local or remote state.

The description, command syntax, query syntax, example, and response are provided for each command. The commands are given in both long and short forms. All examples are shown in the short form. Queries perform actions such as obtaining information and are recognized by the question mark (?) following the header.

4.1	How They are Listed .....	17
4.2	How They are Described .....	17
4.3	When can they be used? .....	17
4.4	Command Notation .....	17

---

## 4.1 How They are Listed

---

The commands are listed by subsystem and in alphabetical order according to their short form.

---

## 4.2 How They are Described

---

In the descriptions themselves, a brief explanation of the function performed is given. This is followed by a presentation of the formal syntax, with the header given in Upper-and-Lower-Case characters and the short form derived from it in ALL UPPER-CASE characters. Where applicable, the syntax of the query is given with the format of its response.

---

## 4.3 When can they be used?

---

The commands and queries listed here can be used for the 4080 Series.

---

## 4.4 Command Notation

---

The following notation is used in the command descriptions:

Notation	Description
< >	Angular brackets enclose words that are used as placeholders, of which there are two types: the header path and the data parameter of a command.
:=	A colon followed by an equals sign separates a placeholder from the description of the type and range of values that may be used in a command instead of the placeholder.
{ }	Braces enclose a list of choices, one of which must be made.
	A vertical bar separates multiple parameter choices (e.g., {ON   OFF}).
[ ]	Square brackets enclose optional items.
...	An ellipsis indicates that the items both to its left and right may be repeated a number of times.

**Table 4.1** Command Notation Symbols

# Common Commands

This section describes the IEEE 488.2 common commands supported by the instrument.

5.1	*CLS	19
5.2	*ESE?	19
5.3	*ESR?	19
5.4	*IDN?	20
5.5	*OPC	20
5.6	*RST	20
5.7	*SRE?	21
5.8	*STB?	21

---

## 5.1 \*CLS

---

**Description:** Clears the following registers: Standard Event Register, Query Event Register, Status Bit Group Register.

**Syntax:**

Command \*CLS

**Parameters:** None

**Example:**

Command \*CLS

---

---

## 5.2 \*ESE?

---

**Description:** Queries the value of the Standard Event Enable Register.

**Syntax:**

Query \*ESE?

**Parameters:** <NR1> : 0 to 255

**Example:**

Query \*ESE?

---

---

## 5.3 \*ESR?

---

**Description:** Reads the value of the Standard Event Register. After the command is executed, the value of the Standard Event Register is cleared. The bit definition is the same as the Standard Event Enable Register.

**Syntax:**

Query \*ESR?

**Parameters:** None

**Example:**

Query \*ESR?

---

---

## 5.4 \*IDN?

---

**Description:** Queries the model number, SN number, and version number of the instrument.

**Syntax:**

Query \*IDN?

**Parameters:** None

**Example:**

Query \*IDN?

---

### NOTICE

**Example return: Tonghui, TH7110, SHF-888-8888, V1.0.0 Copyright(c) 2017-09-01.**

---

---

## 5.5 \*OPC

---

**Description:** Sets the OPC (Operation Complete) bit of the Standard Event Register to 1 when all commands preceding this command have been executed. Sending the query command returns "1" to the output buffer.

**Syntax:**

Command \*OPC

Query \*OPC?

**Parameters:** None

**Example:**

Command \*OPC

Query \*OPC?

---

---

## 5.6 \*RST

---

**Description:** Resets the power supply to the factory setting state.

**Syntax:**

Command \*RST

**Parameters:** None

**Example:**

Command \*RST

---

---

## 5.7 \*SRE?

---

**Description:** Queries the value of the Status Enable Register. Returns a decimal number representing the binary weighted sum of all the bits in the enable register.

**Syntax:**

Query \*SRE?

**Parameters:** None

**Example:**

Query \*SRE?

---

---

## 5.8 \*STB?

---

**Description:** Reads the value of the Status Byte Register.

**Syntax:**

Query \*STB?

**Parameters:** None

**Example:**

Query \*STB?

---

# Source Subsystem

The Source subsystem commands control the frequency, voltage, amplitude, modulation, and clock source of the output waveform.

6.1	Frequency .....	23
6.2	Function .....	23
6.3	Amplitude .....	23
6.4	Offset .....	24
6.5	Clock Reference Source .....	24
6.6	Source Duty Cycle .....	25

## 6.1 Frequency

**Description:** Sets and queries the frequency of the output waveform.

**Syntax:**

Command `SOURce:FREQuency[:<mode>] <NRf>`

Query `SOURce:FREQuency?`

**Parameters:** <mode> : CW | FIXed

<NRf> : .001 mHz to 50 MHz

**Example:**

Command `SOURC:FREQ:FIX 2E6`

Query `SOURC:FREQ?`

### NOTICE

**FIXed is an alias for CW. The MIN and MAX arguments refer to the currently settable minimum or maximum values.**

## 6.2 Function

**Description:** Sets the type of waveform to be generated by the instrument.

**Syntax:**

Command `SOURce:FUNction[:SHAPE] <shape>`

Query `SOURce:FUNction?`

**Parameters:** <shape> : SINsoid | SQUare | TRlangle | ARBitary | PULSE

**Example:**

Command `SOUR:FUNC SQU`

Query `SOUR:FUNC?`

## 6.3 Amplitude

**Description:** Sets the peak-to-peak amplitude of the output waveform. The amplitude and offset are limited by the relation:  $\text{Peak Amplitude} + |\text{Offset}| \leq 5\text{V}$ .

**Syntax:**

Command `SOURce:VOLTage[:AC] [:LEVel] [:IMMediate] [:AMPLitude] <NRf>`

Query `SOURce:VOLTage?`

**Parameters:** <NRf> : 1 mVp-p to 10 Vp-p | 1 mVrms to 5 Vrms

**Example:**

Command SOUR:VOLT 5  
Query SOUR:VOLT?

---

**NOTICE**

The MAXimum amplitude is dependent on the offset. Do not use MAX/MIN arguments in a program message containing an OFFSet command.

---

---

## 6.4 Offset

---

**Description:** Sets the DC offset of the output waveform. The amplitude and offset are limited by the relation: Peak Amplitude + |Offset| ≤ 5V.

**Syntax:**

Command SOURce:VOLTage[:AC][:LEVel][:IMMediate]:OFFSet <NR2>  
Query SOURce:VOLTage:OFFSet?

**Parameters:** <NR2> : -2.500 V to 2.500 V

**Example:**

Command SOUR:VOLT:OFFS 1  
Query SOUR:VOLT:OFFS?

---

**NOTICE**

The MAXimum offset is dependent on the amplitude. Do not use MAX/MIN arguments in a program message containing an AMPLitude command.

---

---

## 6.5 Clock Reference Source

---

**Description:** Selects the source of the arbitrary waveform clock.

**Syntax:**

Command SOURce:REFerence <source>  
Query SOURce:REFerence?

**Parameters:** <source> : INTernal | EXTernal

**Example:**

Command SOUR:REF INT  
Query SOUR:REF?

---

---

## 6.6 Source Duty Cycle

---

**Description:** Sets the duty cycle of the square wave.

**Syntax:**

Command `SOURce:DCYCl e <NR1>`

Query `SOURce:DCYCl e?`

**Parameters:** <NR1> : 5% to 95%

**Example:**

Command `SOUR:DCYC 25`

Query `SOUR:DCYC?`

---

# AM Modulation

The following chapter describes the SCPI commands that control the AM modulation settings.

7.1	Source AM State .....	27
7.2	Source AM Depth .....	27
7.3	Source AM Shape .....	27
7.4	Source AM Frequency .....	28
7.5	Source AM Reference .....	28

---

## 7.1 Source AM State

---

**Description:** Activate or deactivate AM modulation.

**Syntax:**

Command [:SOURce]:AM[:STATe] <bool>

Query [:SOURce]:AM[:STATe]?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command [:SOUR]:AM ON

Query [:SOUR]:AM?

---

---

## 7.2 Source AM Depth

---

**Description:** Sets the AM modulation depth in percentage.

**Syntax:**

Command [:SOURce]:AM:DEPTH <NR1>

Query [:SOURce]:AM:DEPTH?

**Parameters:** <NR1> : 0% to 100%

**Example:**

Command [:SOUR]:AM:DEPT 50

Query [:SOUR]:AM:DEPT?

---

---

## 7.3 Source AM Shape

---

**Description:** Select the AM modulating waveform shape.

**Syntax:**

Command [:SOURce]:AM:SHAPE <shape>

Query [:SOURce]:AM:SHAPE?

**Parameters:** <shape> : SIN | SQU | TRI

**Example:**

Command [:SOUR]:AM:SHAP SIN

Query [:SOUR]:AM:SHAP?

---

---

## 7.4 Source AM Frequency

---

**Description:** Set the AM modulating waveform frequency.

**Syntax:**

Command [:SOURce]:AM:FREQuency <NRf>

Query [:SOURce]:AM:FREQuency?

**Parameters:** <NRf> : 10 mHz to 20 kHz

**Example:**

Command [:SOUR]:AM:FREQ 1e3

Query [:SOUR]:AM:FREQ?

---

---

## 7.5 Source AM Reference

---

**Description:** Select the AM modulation source as either internal or external. If set to internal, the instrument's settings are effective. If set to external, the external waveform determines the depth, shape, and frequency of modulation.

**Syntax:**

Command [:SOURce]:AM:SOURce <source>

Query [:SOURce]:AM:SOURce?

**Parameters:** <source> : INTernal | EXTernal

**Example:**

Command [:SOUR]:AM:SOUR INT

Query [:SOUR]:AM:SOUR?

---

# FM Modulation

The following commands set the FM modulation parameters.

8.1	Source FM State .....	30
8.2	Source FM Deviation .....	30
8.3	Source FM Shape .....	30
8.4	Source FM Frequency .....	31
8.5	Source FM Reference .....	31

---

## 8.1 Source FM State

---

**Description:** Activate or deactivate FM modulation.

**Syntax:**

Command [:SOURce]:FM[:STATe] <bool>

Query [:SOURce]:FM[:STATe]?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command [:SOUR]:FM ON

Query [:SOUR]:FM?

---

---

## 8.2 Source FM Deviation

---

**Description:** Set the FM modulation deviation.

**Syntax:**

Command [:SOURce]:FM:DEVIation <NRf>

Query [:SOURce]:FM:DEVIation?

**Parameters:** <NRf> : 10 mHz to 500 Hz

**Example:**

Command [:SOUR]:FM:DEV 50

Query [:SOUR]:FM:DEV?

---

---

## 8.3 Source FM Shape

---

**Description:** Select the FM modulating waveform shape.

**Syntax:**

Command [:SOURce]:FM:SHAPE <string>

Query [:SOURce]:FM:SHAPE?

**Parameters:** <string> : SIN | SQU | TRI

**Example:**

Command [:SOUR]:FM:SHAP SIN

Query [:SOUR]:FM:SHAP?

---

---

## 8.4 Source FM Frequency

---

**Description:** Set the FM modulating waveform frequency.

**Syntax:**

Command [:SOURce]:FM:FREQuency <NRf>

Query [:SOURce]:FM:FREQuency?

**Parameters:** <NRf> : 10 mHz to 20 kHz

**Example:**

Command [:SOUR]:FM:FREQ 1e3

Query [:SOUR]:FM:FREQ?

---

---

## 8.5 Source FM Reference

---

**Description:** Selects the FM modulation source as either internal or external. If internal, the above settings are effective. If external, the external waveform determines the deviation, shape, and frequency of modulation.

**Syntax:**

Command [:SOURce]:FM:SOURce <string>

Query [:SOURce]:FM:SOURce?

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command [:SOUR]:FM:SOUR INT

Query [:SOUR]:FM:SOUR?

---

# Frequency Shift Keying Modulation

The following commands set the FSK modulation parameters.

9.1	Source FSK State .....	33
9.2	Source FSK Low Frequency .....	33
9.3	Source FSK High Frequency .....	33
9.4	Source FSK Rate .....	34
9.5	Source FSK Reference .....	34

---

## 9.1 Source FSK State

---

**Description:** Enable or disable FSK modulation.

**Syntax:**

Command [:SOURce]:FSK[:STATE] <bool>

Query [:SOURce]:FSK[:STATE]?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command [:SOUR]:FSK ON

Query [:SOUR]:FSK?

---

---

## 9.2 Source FSK Low Frequency

---

**Description:** Set the lower of the two frequencies used in FSK modulation.

**Syntax:**

Command [:SOURce]:FSK:LOWFrequency <NRf>

Query [:SOURce]:FSK:LOWFrequency?

**Parameters:** <NRf> : .001 mHz to 50.00 kHz

**Example:**

Command [:SOUR]:FSK:LOWF 1e3

Query [:SOUR]:FSK:LOWF?

---

---

## 9.3 Source FSK High Frequency

---

**Description:** Set the higher of the two frequencies used in FSK modulation.

**Syntax:**

Command [:SOURce]:FSK:HIFrequency <NRf>

Query [:SOURce]:FSK:HIFrequency?

**Parameters:** <NRf> : .001 mHz to 50.00 kHz

**Example:**

Command [:SOUR]:FSK:HIF 1e3

Query [:SOUR]:FSK:HIF?

---

---

## 9.4 Source FSK Rate

---

**Description:** Set the rate of switching between the two frequencies of the modulation.

**Syntax:**

Command [:SOURce]:FSK:RATE <NRf>

Query [:SOURce]:FSK:RATE?

**Parameters:** <NRf> : 10 mHz to 1 kHz

**Example:**

Command [:SOUR]:FSK:RATE 1e3

Query [:SOUR]:FSK:RATE?

---

---

## 9.5 Source FSK Reference

---

**Description:** Select the FSK modulation source as either internal or external. If internal, the above settings are effective. If external, the external waveform determines the frequency of modulation.

**Syntax:**

Command [:SOURce]:FSK:SOURce <string>

Query [:SOURce]:FSK:SOURce?

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command [:SOUR]:FSK:SOUR INT

Query [:SOUR]:FSK:SOUR?

---

# Binary Shift Keying Modulation

The following commands set the BPSK modulation parameters.

10.1	Source BPSK State .....	36
10.2	Source BPSK Low Phase .....	36
10.3	Source BPSK High Phase .....	36
10.4	Source BPSK Rate .....	37
10.5	Source BPSK Source .....	37

---

## 10.1 Source BPSK State

---

**Description:** Activates or deactivates BPSK modulation.

**Syntax:**

Command `[ :SOURce ] :BPSK [ :STATe ] <bool>`

Query `[ :SOURce ] :BPSK [ :STATe ] ?`

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command `[ :SOUR ] :BPSK ON`

Query `[ :SOUR ] :BPSK ?`

---

---

## 10.2 Source BPSK Low Phase

---

**Description:** Sets the first phase (Low Phase) used in BPSK modulation.

**Syntax:**

Command `[ :SOURce ] :BPSK :LOWPhase <NR3>`

Query `[ :SOURce ] :BPSK :LOWPhase ?`

**Parameters:** <NR3> : 0 to 360 degrees

**Example:**

Command `BPSK :LOWP 80`

Query `BPSK :LOWP ?`

---

---

## 10.3 Source BPSK High Phase

---

**Description:** Sets the second phase (High Phase) used in BPSK modulation.

**Syntax:**

Command `[ :SOURce ] :BPSK :HIPHase <NR3>`

Query `[ :SOURce ] :BPSK :HIPHase ?`

**Parameters:** <NR3> : 0 to 360 degrees

**Example:**

Command `BPSK :HIP 220`

Query `BPSK :HIP ?`

---

---

## 10.4 Source BPSK Rate

---

**Description:** Sets the rate of switching between the two phases of the modulation.

**Syntax:**

Command `[ :SOURce ] :BPSK :RATE <NRf>`

Query `[ :SOURce ] :BPSK :RATE?`

**Parameters:** <NRf> : Frequency Value (e.g., 5KHZ)

**Example:**

Command `BPSK :RATE 5KHZ`

Query `BPSK :RATE?`

---

---

## 10.5 Source BPSK Source

---

**Description:** Selects the BPSK modulation source as either internal or external. If internal, the above settings are effective. If external, the external waveform determines the phase switching.

**Syntax:**

Command `[ :SOURce ] :BPSK :SOURce <string>`

Query `[ :SOURce ] :BPSK :SOURce?`

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command `[ :SOUR ] :BPSK :SOUR INT`

Query `[ :SOUR ] :BPSK :SOUR?`

---

# Phase Modulation

The following commands control the PM modulation parameters.

11.1	Source PM State .....	39
11.2	Source PM Deviation .....	39
11.3	Source PM Shape .....	39
11.4	Source PM Frequency .....	40
11.5	Source PM Reference .....	40

---

## 11.1 Source PM State

---

**Description:** Enable or disable phase modulation.

**Syntax:**

Command [:SOURce]:PM[:STATe] <bool>

Query [:SOURce]:PM[:STATe]?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command [:SOUR]:PM:STAT ON

Query [:SOUR]:PM:STAT?

---

---

## 11.2 Source PM Deviation

---

**Description:** Set the PM modulation deviation.

**Syntax:**

Command [:SOURce]:PM:DEVIation <NR1>

Query [:SOURce]:PM:DEVIation?

**Parameters:** <NR1> : 0 to 360 degrees

**Example:**

Command [:SOUR]:PM:DEV 50

Query [:SOUR]:PM:DEV?

---

---

## 11.3 Source PM Shape

---

**Description:** Select the PM modulating waveform shape.

**Syntax:**

Command [:SOURce]:PM:SHAPE <string>

Query [:SOURce]:PM:SHAPE?

**Parameters:** <string> : SIN | SQU | TRI

**Example:**

Command [:SOUR]:PM:SHAP SIN

Query [:SOUR]:PM:SHAP?

---

---

## 11.4 Source PM Frequency

---

**Description:** Set the PM modulating waveform frequency.

**Syntax:**

Command [:SOURce]:PM:FREQuency <NRf>

Query [:SOURce]:PM:FREQuency?

**Parameters:** <NRf> : 10 mHz to 20 kHz

**Example:**

Command [:SOUR]:PM:FREQ 1e3

Query [:SOUR]:PM:FREQ?

---

---

## 11.5 Source PM Reference

---

**Description:** Select the PM modulation source as either internal or external. If internal, the above settings are effective. If external, the external waveform determines the deviation, shape, and frequency of modulation.

**Syntax:**

Command [:SOURce]:PM:SOURce <string>

Query [:SOURce]:PM:SOURce?

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command [:SOUR]:PM:SOUR INT

Query [:SOUR]:PM:SOUR?

---

# SWEEP

The following commands control the sweep functionality.

12.1	Sweep State .....	42
12.2	Sweep Spacing .....	42
12.3	Sweep Start .....	42
12.4	Sweep Stop .....	43

---

## 12.1 Sweep State

---

**Description:** Enable or disable the sweep function.

**Syntax:**

Command `SOURce:SWEEP:STATe <bool>`

Query `SOURce:SWEEP:STATe?`

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command `SOUR:SWEEP:STAT ON`

Query `SOUR:SWEEP:STAT?`

---

---

## 12.2 Sweep Spacing

---

**Description:** Sets the sweep spacing as either LINear or LOGarithmic.

**Syntax:**

Command `SOURce:SWEEP:SPACing <string>`

Query `SOURce:SWEEP:SPACing?`

**Parameters:** <string> : LINear | LOGarithmic

**Example:**

Command `SOURce:SWEEP:SPACing LIN`

Query `SOURce:SWEEP:SPACing?`

---

---

## 12.3 Sweep Start

---

**Description:** Set the start frequency of the sweep.

**Syntax:**

Command `SOURce:SWEEP:START <NRf>`

Query `SOURce:SWEEP:START?`

**Parameters:** <NRf> : .001 mHz to 50 MHz

**Example:**

Command `SOURce:SWEEP:START 1e3`

Query `SOURce:SWEEP:START?`

---

---

## 12.4 Sweep Stop

---

**Description:** Set the stop frequency of the sweep.

**Syntax:**

Command `SOURce:SWEEP:STOP <NRf>`

Query `SOURce:SWEEP:STOP?`

**Parameters:** <NRf> : .001 mHz to 50 MHz

**Example:**

Command `SOURce:SWEEP:STOP 10e3`

Query `SOURce:SWEEP:STOP?`

---

# Pulse Settings

The following commands control the pulse function parameters such as period, width, and edge transitions.

13.1	Pulse Period	45
13.2	Pulse Width	45
13.3	Pulse Edge	45
13.4	Pulse Rise	46
13.5	Pulse Fall	46

---

## 13.1 Pulse Period

---

**Description:** Sets the pulse period to the specified value. This effectively sets the pulse frequency.

**Syntax:**

Command `SOURce:PULse:PERiod <NRf>`

Query `SOURce:PULse:PERiod?`

**Parameters:** <NRf> : 40 ns to 1000 s

**Example:**

Command `SOURce:PULse:PERiod 1`

Query `SOURce:PULse:PERiod?`

---

---

## 13.2 Pulse Width

---

**Description:** Sets the pulse width to the specified value.

**Syntax:**

Command `SOURce:PULse:WIDth <NRf>`

Query `SOURce:PULse:WIDth?`

**Parameters:** <NRf> : 20 ns minimum (Maximum defined by period and transition)

**Example:**

Command `SOUR:PUL:WID 20 ns`

Query `SOUR:PUL:WID?`

---

### NOTICE

**The width must satisfy the following conditions to be valid:**

**Width < Period - 10 ns**

**Width > 0.6 \* (Rise + Fall)**

---

---

## 13.3 Pulse Edge

---

**Description:** Sets both the rising and falling edge of the pulse to the specified value simultaneously.

**Syntax:**

Command `SOURce:PULse:EDGe <NRf>`

Query `SOURce:PULse:EDGe?`

**Parameters:** <NRf> : 10 ns minimum (Maximum defined by period and width)

**Example:**

Command `SOURce:PULse:EDGe 10 ns`

Query `SOURce:PULse:EDGe?`

---

## 13.4 Pulse Rise

---

**Description:** Sets the rising edge time of the pulse.

**Syntax:**

Command `SOURce:PULse:RISe <NRf>`

Query `SOURce:PULse:RISe?`

**Parameters:** <NRf> : 10 ns minimum (Maximum defined by period and width)

**Example:**

Command `SOURce:PULse:RISe 1e-6`

Query `SOURce:PULse:RISe?`

---

---

## 13.5 Pulse Fall

---

**Description:** Sets the falling edge time of the pulse.

**Syntax:**

Command `SOURce:PULse:FALl <NRf>`

Query `SOURce:PULse:FALl?`

**Parameters:** <NRf> : 10 ns minimum (Maximum defined by period and width)

**Example:**

Command `SOURce:PULse:FALl 1e-6`

Query `SOURce:PULse:FALl?`

---

# Output Subsystem

The Output Subsystem controls characteristics of the source's output.

14.1	Output State .....	48
14.2	Output Impedance .....	48

---

## 14.1 Output State

---

**Description:** Enable or disable the output.

**Syntax:**

Command `OUTPut:STATe <bool>`

Query `OUTPut:STATe?`

**Parameters:** `<bool>` : 0 | 1 | OFF | ON

**Example:**

Command `OUTP:STAT 1`

Query `OUTP:STAT?`

---

---

## 14.2 Output Impedance

---

**Description:** Set whether the 50 Ohm terminator is ON or OFF.

**Syntax:**

Command `OUTPut:TERM <bool>`

Query `OUTPut:TERM?`

**Parameters:** `<bool>` : 0 | 1 | OFF | ON

**Example:**

Command `OUTP:TERM 1`

Query `OUTP:TERM?`

---

# Trigger Subsystem

The Trigger Subsystem is used to control the waveform triggering.

15.1	Trigger Mode	50
15.2	Trigger Source	50
15.3	Trigger Level	51
15.4	Burst Count	51
15.5	Internal Trigger Rate	51

## 15.1 Trigger Mode

**Description:** Sets the trigger mode. This command is not a standard SCPI command.

**Syntax:**

Command TRIGger:MODE <string>

Query TRIGger:MODE?

**Parameters:**

Mode	Description
CONTInuous	Continuous output
TRIGger	Triggered output
GATE	Gated output
BURSt	Burst output

**Table 15.1** Trigger Modes

**Example:**

Command TRIG:MODE CONT

Query TRIG:MODE?

## 15.2 Trigger Source

**Description:** Selects the trigger source for use in the Trigger, Gate, and Burst trigger modes.

**Syntax:**

Command TRIGger:SOURce <string>

Query TRIGger:SOURce?

**Parameters:**

Source	Description
MANual	Front panel MAN key
BUS	GPIB trigger (GET or *TRG)
INTernal	Internal trigger
EXTernal	External trigger

**Table 15.2** Trigger Sources

**Example:**

Command TRIG:SOUR INT

Query TRIG:SOUR?

---

## 15.3 Trigger Level

---

**Description:** Selects the trigger level/polarity for use in the Trigger, Gate, and Burst external trigger modes.

**Syntax:**

Command TRIGger:LEVel <string>

Query TRIGger:LEVel?

**Parameters:** <string> : High | LOW

**Example:**

Command TRIG:LEV LOW

Query TRIG:LEV?

---

---

## 15.4 Burst Count

---

**Description:** Sets the number of cycles to be output in the BURST mode. This is not a standard SCPI command.

**Syntax:**

Command TRIGger:BURSt <NR1>

Query TRIGger:BURSt?

**Parameters:** <NR1> : 2 to 999999

**Example:**

Command TRIG:BURS 10

Query TRIG:BURS?

---

---

## 15.5 Internal Trigger Rate

---

**Description:** Sets the rate of the internal trigger.

**Syntax:**

Command TRIGger:TIMer <NRf>

Query TRIGger:TIMer?

**Parameters:** <NRf> : 1  $\mu$ s to 100 s

**Example:**

Command TRIG:TIM 1e-3

Query TRIG:TIM?

---

# Arbitrary Subsystem

The Arbitrary subsystem is not part of the SCPI standard. It was developed to suit the needs of the instrument. Within this subsystem are commands to control the point rate, start address, wavelength, marker address, and synchronization pulse address; set values of the arbitrary waveform (discretely or using predefined, copy, or draw functions); protect an area of waveform memory; set the state of automatic update; and update the waveform.

The unit has 16,777,216 memory locations. (Optional 33,554,432 locations). The limits in the following text are 16,000,000.

16.1	Point Rate	53
16.2	Address	53
16.3	Data	54
16.4	Line Draw	54
16.5	Clear	55
16.6	Copy	55
16.7	Predefined Waveforms	56
16.8	Start Address	57
16.9	Wavelength	57
16.10	Marker Address	57
16.11	Marker Length	58
16.12	Marker State	58
16.13	Save	58
16.14	Load	59

---

## 16.1 Point Rate

---

**Description:** Sets the point rate. It is coupled with the frequency of the waveform by the relation:  $Frequency = 1 / (PointRate \times Wavelength)$ .

**Syntax:**

Command ARbitrary:PRATe <NRf>

Query ARbitrary:PRATe?

**Parameters:** <NRf> : 4 ns to 100 s

**Example:**

Command ARB:PRAT 6e-6

Query ARB:PRAT?

---

---

## 16.2 Address

---

**Description:** Sets the current address of the waveform. It is used to determine what address the arbitrary data is written to.

**Syntax:**

Command ARbitrary:ADDRESS <NR1>

Query ARbitrary:ADDRESS?

**Parameters:** <NR1> : 1 to 16,000,000

**Example:**

Command ARB:ADDR 1e3

Query ARB:ADDR?

---

## 16.3 Data

**Description:** Sets the values of the waveform. The values can be set numerically or transferred through a definite or indefinite form arbitrary block.

**Syntax:**

Command ARbitrary:DATA <data>

Query ARbitrary:DATA? <points>, <format>

**Parameters:**

Type	Syntax / Range
Numeric	ARbitrary:DATA 100,200,1000...
Definite Block	ARbitrary:DATA #<digits> <count> <data>
Indefinite Block	ARbitrary:DATA #0<data> <terminator>
ASCII Range	-32767 to +32767
Hex Range	0001 to 7FFD

**Table 16.1** Arbitrary Data Formats

**Example:**

Command ARB:DATA 100,200,1000

Query ARB:DATA? 100, ASC

**NOTICE**

**Block Transfer is only available in GPIB. Since indefinite format does not contain a byte count, the command must be terminated with EOI (GPIB) or CR/LF (RS232).**

## 16.4 Line Draw

**Description:** Generates a straight line between two points in the arbitrary waveform memory.

**Syntax:**

Command ARbitrary:DRAW <start>, <stop>

**Parameters:** <start/stop> : 1 to 16,000,000

**Example:**

Command ARB:DRAW 1,1000

**NOTICE**

**The data values at the start and end points must first be set using ARB:DATA. The range cannot overlap with protected memory. Stop address must be greater than start address.**

---

## 16.5 Clear

---

**Description:** Clears all or a portion of waveform memory. The memory is set to the value zero.

**Syntax:**

Command ARbitrary:CLEar <start>, <stop>

**Parameters:** <start/stop> : 1 to 16,000,000

**Example:**

Command ARB:CLE 1,1000

---

### NOTICE

The clear range cannot overlap with protected memory. Stop address must be greater than start address.

---

---

## 16.6 Copy

---

**Description:** Copies a section of the waveform to a different location in waveform memory.

**Syntax:**

Command ARbitrary:COPY <start>, <length>, <dest>

**Parameters:** <start> : 1 to 16,000,000

<length> : Length of segment

<dest> : 1 to 16,000,000

**Example:**

Command ARB:COPY 1,1000,1001

---

### NOTICE

Destination range cannot overlap with protected memory or source range. Addresses must not exceed 16,000,000.

---

## 16.7 Predefined Waveforms

**Description:** Loads the waveform memory with a specific type of waveform.

**Syntax:**

Command ARbitrary:PREDefined <shape>, <start>, <length>, <scale>

**Parameters:**

Shape	Length Constraint	Data
SINusoid	16-65k (div by 4)	0
SQUare	2-65k (div by 2)	0
TRlangle	16-65k (div by 4)	0
NOISE	16-65k	0
URAMP	16-65k	-32767
DRAMP	16-65k	+32767
EXPUp	16-65k	-32767
EXPDn	16-65k	+32767

**Table 16.2** Predefined Waveform Parameters

<scale> : 1 to 100 percent

**Example:**

Command ARB:PRED SIN,1,1000,100

### NOTICE

Start address + Length - 1 must be  $\leq 16,000,000$ . 'Scale' determines the percentage of full scale (-32767 to +32767).

---

## 16.8 Start Address

---

**Description:** Sets the start address of the waveform to be run.

**Syntax:**

Command ARbitrary:START <NR1>

Query ARbitrary:START?

**Parameters:** <NR1> : 1 to 15,999,999

**Example:**

Command ARB:STAR 100

Query ARB:STAR?

---

### NOTICE

**Condition: Start Address + Length - 1  $\leq$  16,000,000.**

---

---

## 16.9 Wavelength

---

**Description:** Sets the length of the waveform being run.

**Syntax:**

Command ARbitrary:LENGth <NRf>

Query ARbitrary:LENGth?

**Parameters:** <NRf> : 2 to 16,000,000

**Example:**

Command ARB:LENG 1E3

Query ARB:LENG?

---

---

## 16.10 Marker Address

---

**Description:** Sets the address of the marker.

**Syntax:**

Command ARbitrary:MARKer:ADDRess <NRf>

Query ARbitrary:MARKer:ADDRess?

**Parameters:** <NRf> : 1 to 16,000,000

**Example:**

Command ARB:MARK:ADDR 45

Query ARB:MARK:ADDR?

---

---

## 16.11 Marker Length

---

**Description:** Sets the marker length.

**Syntax:**

Command ARbitrary:MARKer:LENGth <NRf>

Query ARbitrary:MARKer:LENGth?

**Parameters:** <NRf> : 1 to 4,000

**Example:**

Command ARB:MARK:LENG 5

Query ARB:MARK:LENG?

---

---

## 16.12 Marker State

---

**Description:** Enables or disables the marker.

**Syntax:**

Command ARbitrary:MARKer:STATe <bool>

Query ARbitrary:MARKer:STATe?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command ARB:MARK:STAT ON

Query ARB:MARK:STAT?

---

---

## 16.13 Save

---

**Description:** Saves all unsaved arbitrary waveform data into one of the non-volatile memory locations.

**Syntax:**

Command ARbitrary:SAVe <NR1>

**Parameters:** <NR1> : 1 to 8

**Example:**

Command ARB:SAV 1

---

---

## 16.14 Load

---

**Description:** Loads all arbitrary waveform data from one of the non-volatile memory locations.

**Syntax:**

Command ARbitrary:LOAD <NR1>

**Parameters:** <NR1> : 1 to 8

**Example:**

Command ARB:LOAD 1

---

# Status Subsystem

This subsystem controls the SCPI-defined status reporting structures, which are the QUESTIONable and OPERATION status registers, and the error/event queue. The OPERATION status registers are mandated by SCPI, and so are implemented, but are not used by the hardware. No status is ever reported through them, and they are not detailed in this manual.

17.1	Status Preset .....	61
17.2	Error Queue Read .....	62
17.3	Error Queue Enable .....	63
17.4	Questionable Status .....	63
17.5	Questionable Condition .....	64
17.6	Positive Transition Filter .....	64
17.7	Negative Transition Filter .....	65
17.8	Event Register .....	65
17.9	Event Enable Register .....	66
17.10	Status Diagram .....	66

---

## 17.1 Status Preset

---

**Description:** Set certain status values to default values.

**Syntax:**

Command STATus:PRESet

**Parameters:** None

**Example:**

Command STAT:PRES

---

### NOTICE

The OPERation and QUEstionable enable registers are cleared.

The Positive transition filters are set to 32767.

The Negative transition filters are set to 0.

Only errors in the range (-440:-100) are enabled to be reported in the event queue.

---

## 17.2 Error Queue Read

**Description:** Returns the first entry in the error queue, and removes that entry from the queue. Identical function as SYSTem:ERRor?.

**Syntax:**

Command STATus:QUEue?

**Parameters:**

Error Code	Description
0	No error
-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 separate error
-113	Undefined header
-131	Invalid suffix
-238	Suffix not allowed
-203	Command protected
-221	Settings conflict
-222	Data out of range
-223	Too much data
-240	Hardware error
-350	Error queue overflow

**Table 17.1** Error Definitions

**Example:**

Command STAT:QUE?

**NOTICE**

**Returns:** <Error number>, <error description>

---

## 17.3 Error Queue Enable

---

**Description:** Enable individual errors to be placed in the queue when they occur. Those errors not specified in the command are disabled from being reported in the error queue. Errors and events enabled to be reported at power on depends on the power on status clear status (set with the \*PSC command). If \*PSC is set, the status is cleared on power on, and the range of errors enabled is as set by the STATUS:PRESet command (i.e., -440 to -110). If \*PSC is cleared, the status is not cleared on power on, and the errors and events enabled are those that were enabled before the last power down.

**Syntax:**

Command STATUS:QUEue:ENABle <code>

Query STATus:QUEue:ENABle?

**Parameters:** <code> : Error code list

**Example:**

Command STAT:QUE:ENAB (-440:-410,-258:-220,402,-110)

Query STAT:QUE:ENAB?

---

---

## 17.4 Questionable Status

---

**Description:** The Questionable status data structure is used to alert the user to instrument conditions that affect the signal quality. Two types of conditions are defined in the AWG:

- Frequency - Trigger rate conflict
- Output overload condition

Each condition is reported separately for each channel. Thus, a total of four conditions may be reported. The data structure is comprised of five 16-Bit registers. Each bit represents a different status condition. In the AWG, bits 9 and 11 are used as follows:

- Bit 9: Frequency - trigger rate conflict.
- Bit 11: Output overload.

The existence of these conditions is indicated in the CONDition register. Bit 3 of the status byte is used to indicate the occurrence of a questionable status condition. The conditions cause this bit to be set depending on the values of the other four registers. The positive transition filter enables a bit in the event register to be set when a condition changes from false to true. The negative transition register enables a bit in the event register to be set when a condition changes from true to false. In order for the bit in the event register to be set, the corresponding bit in the transition register must be set. Bit 3 in the status byte will be set if a bit in the event register is set while the corresponding bit in the enable register is set.

---

## 17.5 Questionable Condition

---

**Description:** Read the condition register.

**Syntax:**

Query STATUS:QUEstionable:CONDition?

**Parameters:** None

**Example:**

Query STAT:QUES:COND?

---

**NOTICE**

**Returns:** <NR1>

---

---

## 17.6 Positive Transition Filter

---

**Description:** Set and query the value of the positive transition filter.

**Syntax:**

Command STATUS:QUEstionable:PTR <NR1>

Query STATUS:QUEstionable:PTR?

**Parameters:** <NR1> : 0 to 131,072

**Example:**

Command STATUS:QUEstionable:PTR 100

Query STATUS:QUEstionable:PTR?

---

**NOTICE**

**Returns:** <NR1>

---

---

## 17.7 Negative Transition Filter

---

**Description:** Set and query the value of the negative transition filter.

**Syntax:**

Command `STATUS:QUESTIONABLE:NTR <NR1>`

Query `STATUS:QUESTIONABLE:NTR?`

**Parameters:** <NR1> : 0 to 131,072

**Example:**

Command `STATUS:QUESTIONABLE:NTR 0`

Query `STATUS:QUESTIONABLE:NTR?`

---

**NOTICE**

**Returns:** <NR1>

---

---

## 17.8 Event Register

---

**Description:** Read the event register. Reading the register will clear it.

**Syntax:**

Query `STATUS:QUESTIONABLE:EVENT?`

**Parameters:** None

**Example:**

Query `STAT:QUES:EVEN?`

---

**NOTICE**

**Returns:** <NR1>

---

## 17.9 Event Enable Register

**Description:** Set and query the value of the enable register.

**Syntax:**

Command `STATus:QUESTIONable:ENABLE <numeric values>`

Query `STATus:QUESTIONable:ENABLE?`

**Parameters:** <numeric value>: 0 to 131,072

**Example:**

Command `STATus:QUESTIONable:ENABLE 0`

Query `STAT:QUES:ENAB?`

**NOTICE** Returns: <NR1>

## 17.10 Status Diagram

**Description:** The following table shows the capability of the status reporting.

**Parameters:**

Bit	Bit Name	Decimal	Definition
0	Q_VOLT	1	Bit is 1 when power supply is operating as voltage
1	Q_CURR	2	Bit is 1 when power supply is operating as current
2	Q_TIME	4	Bit is 1 when OVP (Over Voltage Protection)
3	Q_POWER	8	Bit is 1 when OVP (Over Current Protection)
4	Q_TEMP	16	Bit is 1 when OTP (Over Temperature Protection)
5	Q_FREQ	32	Bit is 1 when output is unregulated.
6	Q_PHAS	64	Bit is 1 when remote sense is not connected.
7	Q_MOD	128	Bit is 1 when DLOG is waiting for a trigger.
8	Q_CAL	256	Bit is 1 when instrument is waiting for a trigger.
9	Q_TRIG_RATE_SHORT_1	512	Bit is 1 when inhibit occurs.
10	Q_TRIG_RATE_SHORT_2	1024	Bit is 1 when inhibit occurs.

**Table 17.2** Questionable Status Summary

Bit	Bit Name	Decimal	Definition
11	Q_OUT_OVERLOAD_1	2048	Bit is 1 when inhibit occurs.
12	Q_OUT_OVERLOAD_2	4096	Bit is 1 when inhibit occurs.
13	Q_ARB_LOAD	8192	Bit is 1 when inhibit occurs.
14	Q_ARB_SAVE	16384	Bit is 1 when inhibit occurs.
15	Q_ARB_SUMMARY	32768	Bit is 1 when inhibit occurs.
16	Q_ARB_BINARY_SET	65536	Bit is 1 when inhibit occurs.
17	Q_ARB_PREDIFINE	131072	Bit is 1 when inhibit occurs.
18	Q_SRC_SUMMARY	262144	Bit is 1 when inhibit occurs.
19	Q_RCL	524288	Bit is 1 when inhibit occurs.
20	Q_SAV	1048576	Bit is 1 when inhibit occurs.
21	Q_SHAPE	2097152	Bit is 1 when inhibit occurs.
22	Q_ADC_CALIB	4194304	Bit is 1 when inhibit occurs.

**Table 17.3** Questionable Status Summary Continued

# System Subsystem

The SYSTem subsystem collects the functions that are not related to instrument performance. The functions implemented in the AWG are security, GPIB address changing, error queue reading, SCPI version reading, and power-on buffer setting.

18.1	GPIB Address Change	69
18.2	Error Queue Reading	70
18.3	SCPI Version	71
18.4	Security	71
18.5	Power-On Buffer	71

---

## 18.1 GPIB Address Change

---

**Description:** Sets the GPIB address.

**Syntax:**

Command `SYSTem:COMMunicate:GPIB:ADDRess <NR1>`

Query `SYSTem:COMMunicate:GPIB:ADDRess?`

**Parameters:** <NR1> : 0 to 31

**Example:**

Command `SYSTEM:COMM:GPIB:ADDR 1`

Query `SYSTEM:COMM:GPIB:ADDR?`

---

### NOTICE

Setting the address to 31 puts the instrument in the off-bus state.  
Using the MAX option sets the address to 30, not 31.  
Default Power-on is address 9.

---

## 18.2 Error Queue Reading

**Description:** Returns the first entry in the error queue, and moves that entry from the queue. Identical function to the STATUS:QUEUE:NEXT? query.

**Syntax:**

Query SYSTEM:ERROR?

**Parameters:**

Error Code	Description
0	No error
-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 separate error
-113	Undefined header
-131	Invalid suffix
-238	Suffix not allowed
-203	Command protected
-221	Settings conflict
-222	Data out of range
-223	Too much data
-240	Hardware error
-350	Error queue overflow

**Table 18.1** Error Codes

**Example:**

Query SYSTEM:ERROR?

**NOTICE**

**Returns:** <Error number>, <error description>

---

## 18.3 SCPI Version

---

**Description:** Read the SCPI version to which the instrument complies.

**Syntax:**

Query `SYSTem:VERSion?`

**Parameters:** None

**Example:**

Query `SYSTem:VERSion?`

---

### NOTICE

**Returns:** 1992.0

---

---

## 18.4 Security

---

**Description:** Clears the instrument memory. The stored settings and the arbitrary waveform memory are cleared when the security state is changed from ON to OFF. (Factory reset).

**Syntax:**

Command `SYSTem:SECurity:STATe <bool>`

Query `SYSTem:SECurity:STATe?`

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command `SYSTem:SECurity:STATe ON`

Query `SYSTem:SECurity:STATe?`

---

---

## 18.5 Power-On Buffer

---

**Description:** Sets the power on buffer settings. The instrument will power-on with the settings stored in that buffer.

**Syntax:**

Command `SYSTem:POBuffer <NR1>`

Query `SYSTem:POBuffer?`

**Parameters:** <NR1> : 0 to 49

**Example:**

Command `SYSTem:POBuffer 10`

Query `SYSTem:POBuffer?`

---

# RF Subsystem

The RF Subsystem controls all the variables of the RF function. These commands are valid when the instrument is set to the RF function.

19.1	RF Frequency .....	73
19.2	RF Amplitude .....	73

---

## 19.1 RF Frequency

---

**Description:** Sets the frequency of the output waveform.

**Syntax:**

Command RF:FREQuency <NRf> [units]

Query RF:FREQuency?

**Parameters:** <NRf> : 50 MHz to 3 GHz | MIN | MAX

**Example:**

Command RF:FREQ 3G

Query RF:FREQ?

---

---

## 19.2 RF Amplitude

---

**Description:** Sets the amplitude of the output waveform.

**Syntax:**

Command RF:LEVel <NRf> <unit>

Query RF:LEVel?

**Parameters:**

Unit	Range
dBm	-110.00 dBm to 13.00 dBm
Vrms	7.07 $\mu$ Vrms to 998.81 mVrms

**Table 19.1** RF Amplitude Ranges

**Example:**

Command RF:LEV -110 dBm

Query RF:LEV?

---

# RF Amplitude Modulation

The following chapter describes the SCPI commands that control the RF AM modulation settings.

20.1	RF AM State .....	75
20.2	RF AM Depth .....	75
20.3	RF AM Shape .....	75
20.4	RF AM Frequency .....	76
20.5	RF AM Reference .....	76

---

## 20.1 RF AM State

---

**Description:** Enables or disables AM modulation.

**Syntax:**

Command RF:AM:STATE <bool>

Query RF:AM:STATE?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command RF:AM:STAT ON

Query RF:AM:STAT?

---

---

## 20.2 RF AM Depth

---

**Description:** Sets the AM modulation depth in percentage.

**Syntax:**

Command RF:AM:DEPTH <NR1>

Query RF:AM:DEPTH?

**Parameters:** <NR1> : 0% to 100%

**Example:**

Command RF:AM:DEPT 50

Query RF:AM:DEPT?

---

---

## 20.3 RF AM Shape

---

**Description:** Selects the AM modulating waveform shape.

**Syntax:**

Command RF:AM:SHAPE <string>

Query RF:AM:SHAPE?

**Parameters:** <string> : SIN | SQU | TRI

**Example:**

Command RF:AM:SHAP SIN

Query RF:AM:SHAP?

---

---

## 20.4 RF AM Frequency

---

**Description:** Sets the AM modulating waveform frequency.

**Syntax:**

Command RF:AM:FREQuency <NRf>

Query RF:AM:FREQuency?

**Parameters:** <NRf> : 10 mHz to 20 kHz

**Example:**

Command RF:AM:FREQ 1e3

Query RF:AM:FREQ?

---

---

## 20.5 RF AM Reference

---

**Description:** Sets the AM modulation RF as either internal (then the above settings are effective) or external (and then the external waveform determines depth, shape, and frequency of modulation).

**Syntax:**

Command RF:AM:RF <string>

Query RF:AM:SOURce?

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command RF:AM:SOUR INT

Query RF:AM:SOUR?

---

RF Frequency Modulation

# RF Frequency Modulation

The following commands control the RF FM modulation parameters.

21.1	RF FM State .....	78
21.2	RF FM Deviation .....	78
21.3	RF FM Shape .....	78
21.4	RF FM Frequency .....	79
21.5	RF FM Reference .....	79

---

## 21.1 RF FM State

---

**Description:** Activates or deactivates FM modulation.

**Syntax:**

Command RF:FM:STATe <bool>

Query RF:FM:STATe?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command RF:FM:STAT ON

Query RF:FM:STAT?

---

---

## 21.2 RF FM Deviation

---

**Description:** Sets the FM modulation deviation.

**Syntax:**

Command RF:FM:DEVIation <NRf>

Query RF:FM:DEVIation?

**Parameters:** <NRf> : 10 mHz to 1 MHz | MIN | MAX

**Example:**

Command RF:FM:DEV 1MHz

Query RF:FM:DEV?

---

---

## 21.3 RF FM Shape

---

**Description:** Selects the FM modulating waveform shape.

**Syntax:**

Command RF:FM:SHAPE <string>

Query RF:FM:SHAPE?

**Parameters:** <string> : SIN | SQU | TRI

**Example:**

Command RF:FM:SHAP SIN

Query RF:FM:SHAP?

---

---

## 21.4 RF FM Frequency

---

**Description:** Sets the FM modulating waveform frequency.

**Syntax:**

Command RF:FM:FREQuency <NRf>

Query RF:FM:FREQuency?

**Parameters:** <NRf> : 10 mHz to 100 kHz | MIN | MAX

**Example:**

Command RF:FM:FREQ 1 KHz

Query RF:FM:FREQ?

---

---

## 21.5 RF FM Reference

---

**Description:** Selects the FM modulation source as either internal (then the above settings are effective) or external (and then the external waveform determines deviation, shape, and frequency of modulation).

**Syntax:**

Command RF:FM:SOURce <string>

Query RF:FM:SOURce?

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command RF:FM:SOUR INT

Query RF:FM:SOUR?

---

# RF Phase Modulation

The following commands control the RF phase modulation (PM) parameters.

22.1	RF PM State .....	81
22.2	RF PM Deviation .....	81
22.3	RF PM Shape .....	81
22.4	RF PM Frequency .....	82
22.5	RF PM Reference .....	82

---

## 22.1 RF PM State

---

**Description:** Activates or deactivates PM modulation.

**Syntax:**

Command RF:PM:STATe <bool>

Query RF:PM:STATe?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command RF:PM:STAT ON

Query RF:PM:STAT?

---

---

## 22.2 RF PM Deviation

---

**Description:** Sets the PM modulation deviation.

**Syntax:**

Command RF:PM:DEVIation <NR2>

Query RF:PM:DEVIation?

**Parameters:** <NR2> : 0 to 6.2 | MIN | MAX

**Example:**

Command RF:PM:DEV 3.14

Query RF:PM:DEV?

---

---

## 22.3 RF PM Shape

---

**Description:** Selects the PM modulating waveform shape.

**Syntax:**

Command RF:PM:SHAPE <string>

Query RF:PM:SHAPE?

**Parameters:** <string> : SIN | SQU | TRI

**Example:**

Command RF:PM:SHAP SIN

Query RF:PM:SHAP?

---

---

## 22.4 RF PM Frequency

---

**Description:** Sets the PM modulating waveform frequency.

**Syntax:**

Command RF:PM:FREQuency <NRf>

Query RF:PM:FREQuency?

**Parameters:** <NRf> : 10 mHz to 100 kHz | MIN | MAX

**Example:**

Command RF:PM:FREQ 1kHz

Query RF:PM:FREQ?

---

---

## 22.5 RF PM Reference

---

**Description:** Selects the PM modulation source as either internal (then the above settings are effective) or external (and then the external waveform determines deviation, shape, and frequency of modulation).

**Syntax:**

Command RF:PM:SOURce <string>

Query RF:PM:SOURce?

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command RF:PM:SOUR INT

Query RF:PM:SOUR?

---

# RF Pulse Modulation

The following commands control the RF pulse modulation.

23.1	RF Pulse State	84
23.2	RF Pulse Frequency	84
23.3	RF Pulse Reference	84

---

## 23.1 RF Pulse State

---

**Description:** Enables or disables pulse modulation.

**Syntax:**

Command RF:PULse:STATe <bool>

Query RF:PULse:STATe?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command RF:PUL:STAT ON

Query RF:PUL:STAT?

---

---

## 23.2 RF Pulse Frequency

---

**Description:** Sets the RF pulse modulating waveform's frequency.

**Syntax:**

Command RF:PULse:FREQuency <NRf>

Query RF:PULse:FREQuency?

**Parameters:** <NRf> : 10 mHz to 20 kHz

**Example:**

Command RF:PUL:FREQ 1e3

Query RF:PUL:FREQ?

---

---

## 23.3 RF Pulse Reference

---

**Description:** Sets the RF pulse modulation source as either internal (then the above settings are effective) or external (and then the external waveform determines depth, shape, and frequency of modulation).

**Syntax:**

Command RF:PULse:SOURce <string>

Query RF:PULse:SOURce?

**Parameters:** <string> : INTernal | EXTernal

**Example:**

Command RF:PUL:SOUR INT

Query RF:PUL:SOUR?

---

# RF Sweep

The following commands control the RF sweep functionality.

24.1	Sweep State .....	86
24.2	Sweep Spacing .....	86
24.3	Sweep Time .....	86
24.4	Sweep Start .....	87
24.5	Sweep Stop .....	87

---

## 24.1 Sweep State

---

**Description:** Enable or disable the sweep function.

**Syntax:**

Command RF:SWEEP:STATe <bool>

Query RF:SWEEP:STATe?

**Parameters:** <bool> : 0 | 1 | OFF | ON

**Example:**

Command RF:SWEEP:STAT ON

Query RF:SWEEP:STAT?

---

---

## 24.2 Sweep Spacing

---

**Description:** Sets the sweep spacing as either LINear or LOGarithmic.

**Syntax:**

Command RF:SWEEP:SPACing <string>

Query RF:SWEEP:SPACing?

**Parameters:** <string> : LINear | LOGarithmic

**Example:**

Command RF:SWEEP:SPACing LIN

Query RF:SWEEP:SPACing?

---

---

## 24.3 Sweep Time

---

**Description:** Sets the time for one complete sweep.

**Syntax:**

Command RF:SWEEP:TIME <NRf> [units]

Query RF:SWEEP:TIME?

**Parameters:** <NRf> : 10 ms to 500 s | MIN | MAX

**Example:**

Command SWEEP:TIME 10e-3

Query SWEEP:TIME?

---

---

## 24.4 Sweep Start

---

**Description:** Sets the start frequency of the sweep.

**Syntax:**

Command `RF:SWEEP:START <NRf> [units]`

Query `RF:SWEEP:START?`

**Parameters:** <NRf> : 50 MHz to 3 GHz | MIN | MAX

**Example:**

Command `RF:SWEEP:START 3GHz`

Query `RF:SWEEP:START?`

---

---

## 24.5 Sweep Stop

---

**Description:** Sets the stop frequency of the sweep.

**Syntax:**

Command `RF:SWEEP:STOP <NRf> [units]`

Query `RF:SWEEP:STOP?`

**Parameters:** <NRf> : 50 MHz to 3 GHz | MIN | MAX

**Example:**

Command `RF:SWEEP:STOP 3GHz`

Query `RF:SWEEP:STOP?`

---