Programming Manual

Rubidium™ MG362x1A Series Low-Noise RF/Microwave Signal Generators

Anritsu Company
490 Jarvis Drive
Morgan Hill, CA 95037-2809
USA

Part Number: 10370-10385
Revision: E
Published: 2023-6-9T13:08:14

Table of Contents

Chapter 1—General GPIB, Ethernet Information

Chapter 2—Programming with SCPI Commands

Chapter 3—Native Programming

Chapter 4—SCPI Commands

Chapter 4—MG362x1A Commands

Chapter 5—Error Messages

Appendix A—Command Reference

Chapter 1 — General GPIB, Ethernet Information

1-1 Introduction

This manual provides information for remote operation of the Series MG362x1A Synthesized High Performance Signal Generator using commands sent from an external controller via Ethernet Sockets and IEEE-488 General Purpose Interface Bus (GPIB).

1-2 Chapter Introduction

This chapter provides a general description of the GPIB and the bus data transfer and control functions. It also contains a listing of the MG362x1A's GPIB interface function subset capability and response to IEEE-488 interface function messages.

The GPIB information presented in this chapter is general in nature. For complete and specific information, refer to the following documents: ANSI/IEEE Std 488.1-1987 *IEEE Standard Digital Interface for Programmable Instrumentation* and ANSI/IEEE Std 488.2-1987 *IEEE Standard Codes, Formats, Protocols and Common Commands*. These documents precisely define the total specification of the mechanical and electrical interface, and of the data transfer and control protocols.

1-3 Scope of Manual

This manual includes the following:

- A general description of the GPIB and the bus data transfer and control functions
- A listing of the IEEE-488 Interface Function Messages recognized by the signal generator with a description of its response
- A complete listing and description of all the Standard Commands for Programmable Instruments (SCPI) commands that can be used to control signal generator operation with examples of command usage
- Error messages related to the MG362x1A operation

This manual is intended to be used in conjunction with the Series MG362x1A Synthesized Signal Generator Operation Manual, P/N: 10370-10383. Refer to that manual for general information about the MG362x1A, including equipment set up and front panel (manual mode) operating instructions.

1-4 Related Manuals

This is one of a four manual set that consists of an operation manual, programming manual, maintenance manual and a technical data sheet.

Operation Manual

The Series MG362x1A Synthesized Signal Generator Operation Manual provides general information, installation, and operating information for the Anritsu series MG362x1A synthesized signal generator. The Anritsu part number for the operation manual is 10370-10383.

Maintenance Manual

The Series MG362x1A Synthesized Signal Generator Maintenance Manual provides service information for all models in the MG362x1A series. The service information includes functional circuit descriptions, block diagrams, performance verification tests, calibration procedures, troubleshooting data, and assembly and component removal/replacement procedures. The Anritsu part number for the maintenance manual is 10370-10386.

Technical Data Sheet

The Series MG362x1A Synthesized Signal Generator technical data sheet provides performance specifications, operating conditions, list of options and accessories, and ordering information. The Anritsu part number for the maintenance manual is 11410-00928.

1-5 MG362x1A GPIB

All Series MG362x1A Synthesized Signal Generator functions, settings, and operating modes (except for power on/standby) are controllable using commands sent from an external controller via the GPIB. When in the remote (GPIB) mode, the signal generator functions as both a listener and a talker.

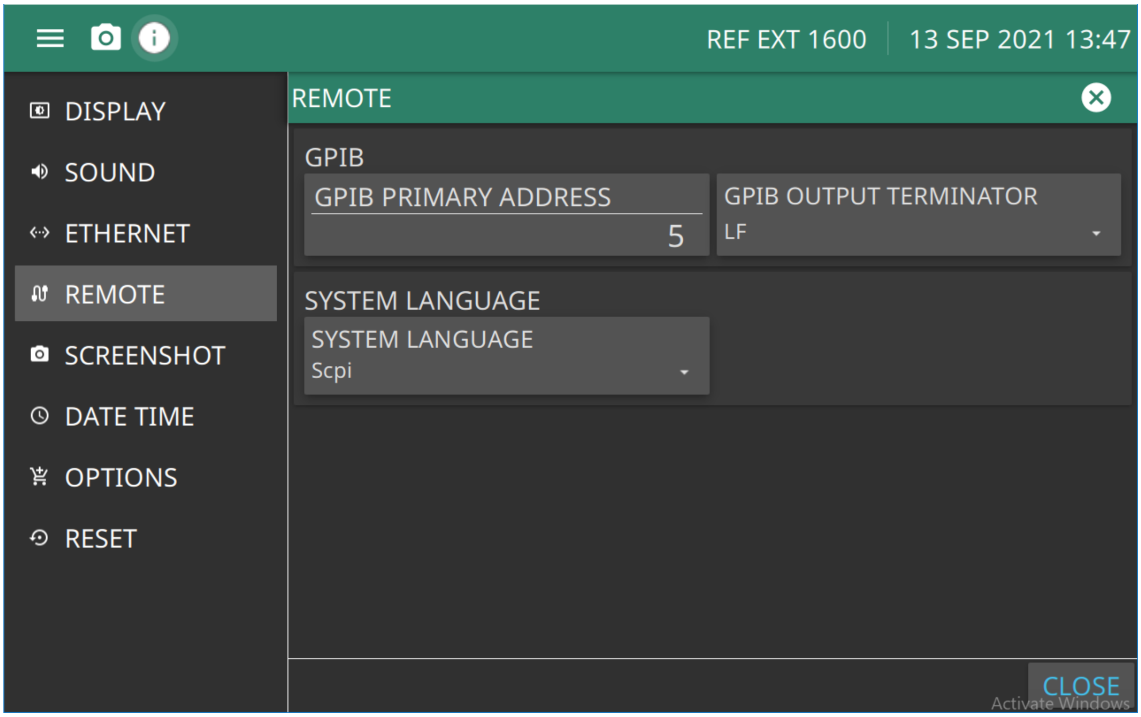
Setting GPIB Operating Parameters

The MG362x1A leaves the factory with the GPIB address value set to 5 and the data delimiting terminator can be set to Carriage Return or Line Feed.(CR/LF). A different address value can be entered from the front panel using the Configure GPIB menu. Using this same menu, the data delimiting terminator can be changed to carriage return (CR) only. To change these parameters, refer to Figure 1-1.

Selecting the Interface Language

The MG362x1A Synthesized Signal Generators can be remotely operated using one of two external interface languages—Native commands or SCPI. The Native language can be used over any remote bus to control the instrument. The SCPI interface language uses a set of the Standard Commands for Programmable Instruments (SCPI) commands to control the unit. Selecting which of these external interface languages to be used can be made from the front panel using the Remote menu as shown in Figure 1-1. To change the System language:

Press: 3-Line Icon | REMOTE | SYSTEM LANGUAGE.



GPIB

GPIB PRIMARY ADDRESS 5

GPIB OUTPUT TERMINATOR LF

SYSTEM LANGUAGE

SYSTEM LANGUAGE Scpi

GPIB: Select the address of the MG362x1A on the bus. Enter a new address, between 1 and 30. The new GPIB address will appear on the display.

GPIB OUTPUT TERMINATOR: select carriage return (CR), line feed, (LF) or carriage return and line feed (CR/LF) as the GPIB data delimiter. Consult the GPIB controller's manual to determine which data delimiter is required.

SYSTEM LANGUAGE: Select SCPI Native.

Figure 1-1. Configure GPIB Menu

Response to GPIB Interface Function Messages

Table 1-1 lists the GPIB interface function messages that the MG362x1A will recognize and respond to. With the exception of the Device Clear and Selected Device Clear messages, these messages affect only the operation of the MG362x1A GPIB interface. The signal generator's response for each message is indicated.

Interface function messages are transmitted on the GPIB data lines and interface management lines as either unaddressed or addressed commands. The manner in which these messages are invoked in programs is implementation dependent. For programming information, refer to the documentation included with the GPIB interface used for the external controller.

Table 1-1. MG362x1A Response to GPIB Interface Function Messages

Interface Function Message	Addressed Command	MG362x1A Response
Device Clear (DCL) Selected Device Clear (SDC)	No Yes	Resets the MG362x1A to its default state. (Equivalent to sending the `*RST` command.)
Go To Local (GTL)	Yes	Returns the MG362x1A to local (front panel) control.
Group Execute Trigger (GET)	Yes	Executes a string of commands, if programmed.
Interface Clear (IFC)	No	Stops the MG362x1A GPIB interface from listening or talking. (The front panel controls are not cleared.)
Remote Enable (REN)	No	Places the MG362x1A under remote (GPIB) control when it has been addressed to listen.
Serial-Poll Enable (SPE)	No	Outputs the serial-poll status byte.
Serial-Poll Disable (SPD)	No	Disables the serial-poll function.
Parallel-Poll Configure (PPC)	Yes	Responds to a parallel-poll message (PPOLL) by setting assigned data bus line to the logical state (1,0) that indicates its correct SRQ status.
Parallel-Poll Unconfigure (PPU)	No	Disables the parallel-poll function.

1-6 Remote Programming Setup and Interface

Remote programming and operation of the instrument is accomplished using HiSLIP with raw socket programming over Ethernet. The following paragraphs provide general information about interface language, utilizing VISA, interface connections, and cable requirements.

Refer to Section 1-7 for GPIB setup instructions.

Refer to Section 1-8 for Ethernet setup instructions.

VISA (Virtual Instrument System Architecture)

VISA is an I/O software standard for communicating with test instruments, like the MG362x1A, over any of the bus architectures which the MG362x1A supports. Anritsu recommends NI-VISA, the National Instruments implementation of the VISA I/O standard.

Note	Use the full version of NI-VISA (not just the runtime) for the best support of the latest .NET Framework, USB, and TCP/IP updates. Check with National Instruments for details.

The MG362x1A LabVIEW driver uses VISA to communicate. VISA uses connection strings to set up communication with the MG362x1A over various protocols. Some examples include:

```
//Raw Socket Connection string

"TCPIP0::192.168.1.7::9001::SOCKET";

//GPIB Connection string

"GPIB0::6::INSTR";
```

To communicate with the MG362x1A, note either the IP address or GPIB address of the signal generator and then set up a resource using the appropriate connection string.

Interface Connectors

When using Ethernet connectivity, the interface between the signal generator and other devices is through an RJ45 connector and a standard CAT-5 or CAT-7 Ethernet cable. For instruments on the GPIB, connectivity is established via a 24-wire interface cable. This cable uses connector shells having two connector faces. These double-faced connectors allow for the parallel connection of two or more cables to a single device.

1-7 GPIB Interconnection and Setup

GPIB Interconnection

The only interconnection required for GPIB operation is between the signal generator and the controller. This interconnection is via a standard GPIB cable. The Anritsu part number for such a cable is 2100-1, -2, or -4 (1, 2, or 4 meters in length).

GPIB Cable Length Restrictions

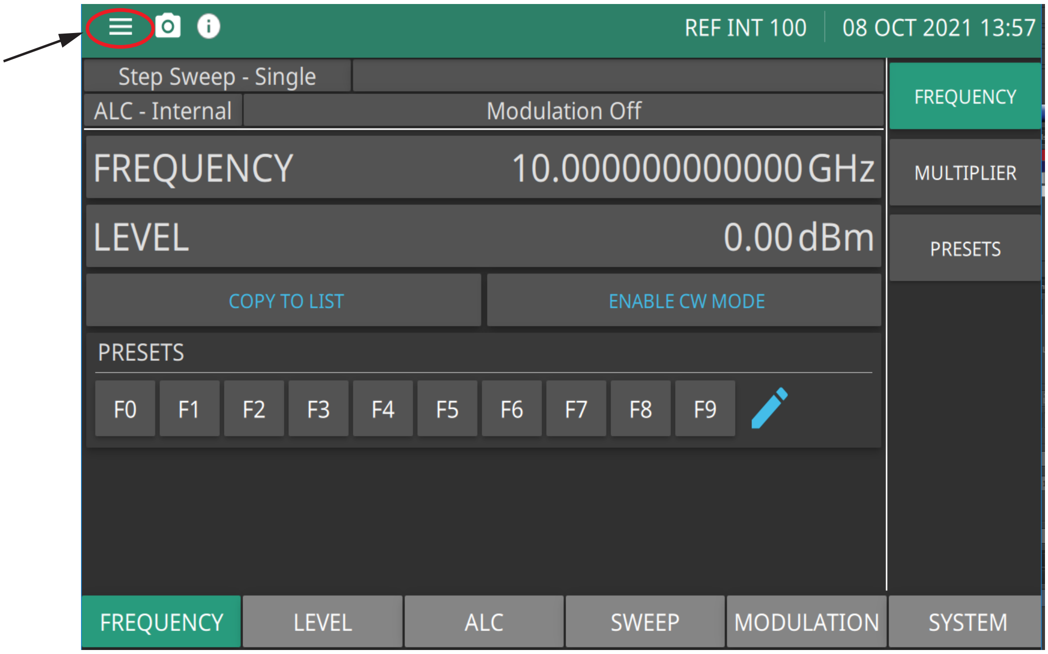
The GPIB can accommodate up to 15 instruments at any one time. To achieve design performance on the bus, proper timing and voltage level relationships must be maintained. If either the cable length between separate instruments or the cumulative cable length between all instruments is too long, the data and control lines cannot be driven properly and the system may fail to perform. Cable length restrictions are as follows:

- No more than 15 instruments may be installed on the bus.
- Total cumulative cable length (in meters) may not exceed two times the number of bus instruments or 20 meters—whichever is less.

Note	For low EMI applications, the GPIB cable should be a fully shielded type with well-grounded metal-shell connectors.

Setting the GPIB Address

The default GPIB address is five. If a different GPIB address is desired, it can be accessed from the 3-line icon shown in Figure 1-2.



REF INT 100

08 OCT 2021 13:57

Step Sweep - Single

ALC - Internal

Modulation Off

FREQUENCY 10.000000000000 GHz

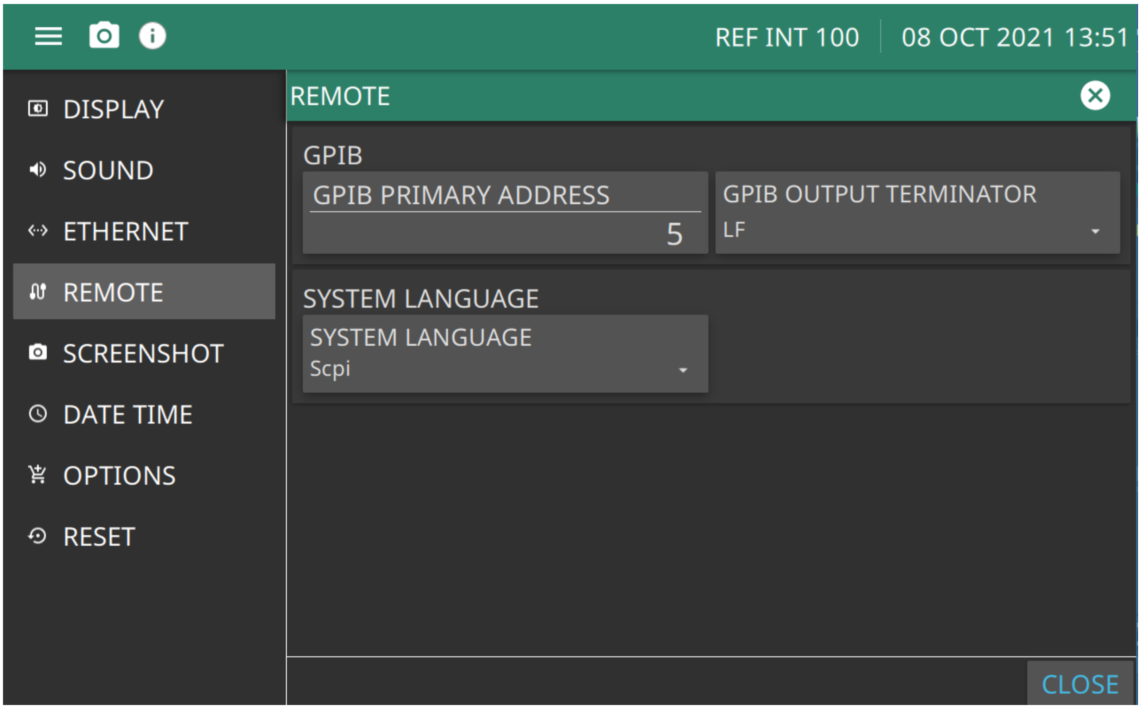
LEVEL 0.00 dBm

Figure 1-2. System Configuration Menu

To change the GPIB address:

Press: 3-line icon | SETTINGS | REMOTE |GPIB PRIMARY ADDRESS

The Configure GPIB menu is displayed in Figure 1-3.



GPIB PRIMARY ADDRESS: 5

GPIB OUTPUT TERMINATOR: LF

SYSTEM LANGUAGE

SYSTEM LANGUAGE: Scpi

Figure 1-3. GPIB Configuration Menu

1. Press GPIB PRIMARY ADDRESS to change the current GPIB address of the signal generator.
2. Enter a new address using the cursor control keys or the data entry keypad and the terminator soft key The new GPIB address will now appear on the display. The entry range is between 1 and 30.

Selecting the GPIB OUTPUT TERMINATOR

Data is delimited on the GPIB by either the carriage return (CR) ASCII character or both the carriage return and line feed (CR/LF) ASCII characters. Which character is used depends upon the requirements of the system controller. Most modern controllers can use either CR, LF or CR/LF, while many older controllers require one or the other. The GPIB Output Terminator only applies to GPIB responses from the instrument. It is not linked with the command controller coming from the controller. Consult the controller's manual for its particular requirements.

From the Configure GPIB menu display:

Press: 3-line icon | SETTINGS | REMOTE |GPIB OUTPUT TERMINATOR

This menu soft key toggles the GPIB terminator between CR, LF, and CR/LF. The current selection appears on the display.

1-8 Ethernet LAN TCP/IP Requirements and Setup

Information about Ethernet LAN general requirements, configuration, use, and restoring defaults are discussed in the following sections. Ethernet control programming is discussed with an example in the MG362x1A SCPI programming manuals.

- "TCP/IP General Requirements and Settings"
- "NI-VISA"

Note	This section is provided for general information about directly manually configuring an Ethernet connection. Consult the user local network administrator for the exact requirements and settings that are required for the network installation.

TCP/IP General Requirements and Settings

Transmission Control Protocol/Internet Protocol (TCP/IP) is a network protocol. TCP/IP is automatically installed and in most cases, installation, configuration, and communication are transparent to the user.

The TCP/IP setup requires the following:

- IP Address: Every computer and electronic device in a TCP/IP network requires an IP address. An IP address has four numbers (each between 0 and 255) separated by periods. For example: 128.111.122.42 is a valid IP address.
- Subnet Mask: The subnet mask distinguishes the portion of the IP address that is the network ID from the portion that is the station ID. The subnet mask 255.255.0.0, when applied to the IP address given above, would identify the network ID as 128.111 and the station ID as 122.42. All stations in the same local area network should have the same network ID, but different station IDs.
- Default Gateway: A TCP/IP network can have a gateway to communicate beyond the LAN identified by the network ID. A gateway is a computer or electronic device that is connected to two different networks and can move TCP/IP data from one network to the other. A single LAN that is not connected to another LAN requires a default gateway setting of 0.0.0.0. If you have a gateway, then the default gateway would be set to the appropriate value of your gateway.
- Ethernet Address: An Ethernet address, or *Media Access Control* (MAC) address, is a unique 48-bit value that identifies a network interface card to the rest of the network. Every network card has a unique Ethernet address permanently stored into its memory.
- Remote programming and operation between the instrument and remote program is accomplished via a TCP/IP raw socket connection to port 9001. The remote program must establish a TCP/IP raw socket connection at port 9001 of the MS362x1A.
- The remote application may connect to the instrument IP address or to its `HOSTNAME`. If using DHCP instead of a static IP, using the `HOSTNAME` may be more reliable for finding an instrument on a network.
- You may need to contact your network administrator to ensure network security policies, antivirus, and firewall settings do not block access to the controlling computer and its ports.

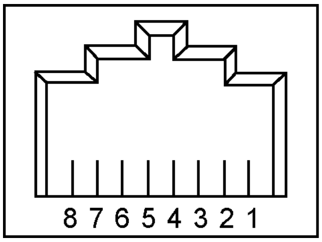
The MS362x1A can be configured for *Dynamic Host Configuration Protocol* (DHCP), an Internet protocol that automates the process of setting IP addresses for devices that use TCP/IP, and is the most common method of configuring a device for network use.

To determine if a network is set up for DHCP, connect the instrument to the network and select DHCP protocol. If the network is set up for DHCP, the assigned IP address should be displayed in the network settings.

Network Connection

Interface between the instrument and other devices on the network is via a category five (CAT-5) interface cable connected to a network. This cable uses four twisted pairs of insulated copper wires terminated into an RJ45 connector. CAT-5 cabling is capable of supporting frequencies up to 100 MHz and data transfer speeds up to 1 Gbps, which accommodates 1000Base-T, 100Base-T, and 10Base-T networks. CAT-5 cables are based on the EIA/TIA 568 Commercial Building Telecommunications Wiring Standard developed by the Electronics Industries Association. A pinout diagram is shown in Table 1-2.

Table 1-2. 8-pin Ethernet RJ45 Connector Pinout Diagram



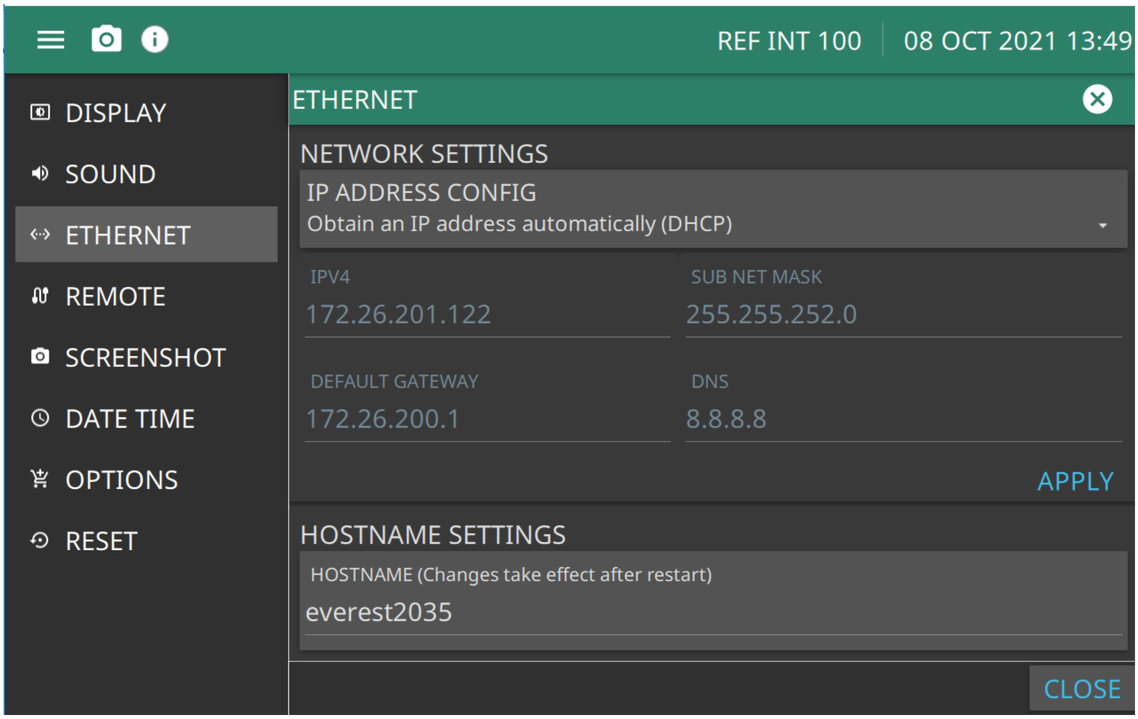
Pin	Name	Description	Wire Color
1	TX+ or BI_DA+	Transmit or bidirectional data	White/Green
2	TX– or BI_DA-	Transmit or bidirectional data	Green
3	RX+ or BI_DB+	Receive or bidirectional data	White/Orange
4	BI_DC+	Not used (common mode termination) or bidirectional	Blue
5	BI_DC-	Not used (common mode termination) or bidirectional	White/Blue
6	RX– or BI_DB-	Receive or bidirectional data	Orange
7	BI_DD+	Not used (common mode termination) or bidirectional	White/Brown
8	BI_DD-	Not used (common mode termination) or bidirectional	Brown

Integrated into the RJ45 connector are two LEDs. The amber LED indicates the presence of LAN voltages (a live LAN connection) while the green LED flashes to show that LAN traffic is present. The instrument IP address and its HOSTNAME are set via the System menu (upper left corner) and accessing the ETHERNET or WIFI settings menu.

TCP/IP connectivity requires setting up the parameters described at the beginning of this section. The following is a brief overview of how to set up a general LAN connection on the MG362x1.

Note	You may need to consult your network documentation or network administrator for assistance in configuring your network setup.

1. Access the System menu (three bars in upper left corner).
2. Press; 3-line icon | SETTINGS | ETHERNET displays the network settings as shown in Figure 1-4.
3. Change the IP address or Hostname (HOSTNAME SETTINGS).



REF INT 100 | 08 OCT 2021 13:49

ETHERNET

NETWORK SETTINGS

IP ADDRESS CONFIG: Obtain an IP address automatically (DHCP)

IPV4: 172.26.201.122

SUB NET MASK: 255.255.252.0

DEFAULT GATEWAY: 172.26.200.1

DNS: 8.8.8.8

HOSTNAME SETTINGS

HOSTNAME (Changes take effect after restart): everest2035

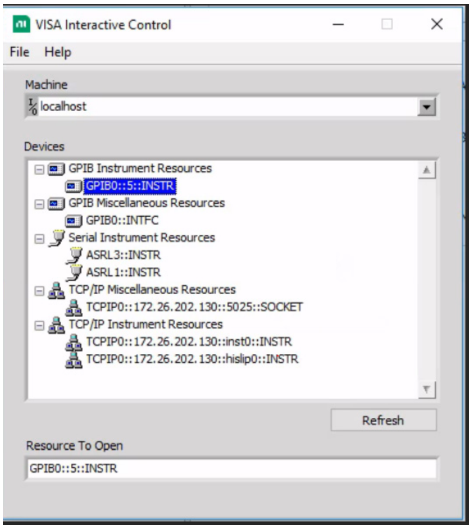
Figure 1-4. Network Settings

1-9 NI-VISA

SCPI commands can be sent to the instrument through any Virtual Instrument Software Architecture (VISA) controller. VISA is a commonly used API in the Test and Measurement industry for communicating with instruments from a PC. The physical connection between the PC and the instrument can be Ethernet or GPIB. NI-VISA is the National Instruments implementation of the VISA I/O standard. Information and downloads are available at http://www.ni.com/visa/.

The following example describes the verification that a VISA controller can interact with the instrument. The images shown and the instructions for your instrument and software may differ from the example shown here, so refer to the NI user instructions for more details.

1. On the PC, run VISA Interactive Control and double-click on the instrument listed in the GPIB Instrument Resources section as shown in Figure 1-5.



Machine: localhost

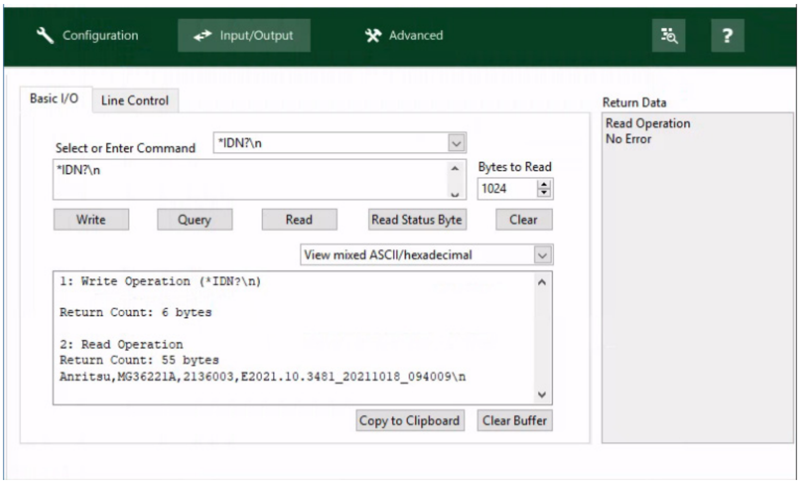
Devices

```
GPIB Instrument Resources
    GPIB0::5::INSTR
GPIB Miscellaneous Resources
    GPIB0::INTFC
Serial Instrument Resources
    ASRL3::INSTR
    ASRL1::INSTR
TCP/IP Miscellaneous Resources
    TCPIP0::172.26.202.130::5025::SOCKET
TCP/IP Instrument Resources
    TCPIP0::172.26.202.130::inst0::INSTR
    TCPIP0::172.26.202.130::hislip0::INSTR
```

Resource To Open: GPIB0::5::INSTR

Figure 1-5. VISA Interactive Control

2. Select the Write tab and execute the default *IDN? Write by clicking the Write button.
3. Click the Read tab. If the PC is connected to the instrument, the command returns the following information from the Buffer: manufacturer name (“Anritsu”), model number, serial number, and firmware package number as shown in Figure 1-6 on page 1-11.



Select or Enter Command: *IDN?\n

*IDN?\n

Bytes to Read: 1024

View mixed ASCII/hexadecimal

```
1: Write Operation (*IDN?\n)

Return Count: 6 bytes

2: Read Operation
Return Count: 55 bytes
Anritsu,MG36221A,2136003,E2021.10.3481_20211018_094009\n
```

Return Data

Read Operation
No Error

Figure 1-6. VISA Interactive Line Control Tab

Communicate Over GPIB – Example

This code demonstrates synchronous read and write commands to a GPIB instrument using VISA.

```
/********************************************************************/
/*              Read and Write to an Instrument Example             */
/*                                                                  */
/* This code demonstrates synchronous read and write commands to a */
/* GPIB instrument using VISA.                                      */
/*                                                                  */
/* The general flow of the code is                                  */
/*      Open Resource Manager                                       */
/*      Open VISA Session to an Instrument                          */
/*      Write the Identification Query Using viWrite                */
/*      Try to Read a Response With viRead                          */
/*      Close the VISA Session                                      */
/********************************************************************/

#if defined(_MSC_VER) && !defined(_CRT_SECURE_NO_DEPRECATE)
/* Functions like strcpy are technically not secure because they do */
/* not contain a 'length'. But we disable this warning for the VISA */
/* examples since we never copy more than the actual buffer size.   */
#define _CRT_SECURE_NO_DEPRECATE
#endif

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include "visa.h"

static ViSession defaultRM;
static ViSession instr;
static ViStatus status;
static ViUInt32 retCount;
static ViUInt32 writeCount;
static unsigned char buffer[100];
static char stringinput[512];

/*
* In every source code or header file that you use it is necessary to prototype your
VISA variables at the beginning of the file. You need to declare the VISA session,
VISA integers, VISA strings, VISA pointers, and VISA floating variables.
* Remember that if you are prototyping variables that are to be used as part of the
VISA session that need this prototyping. As an example, above retCount has been
prototyped as a static variable to this particular module.   It is an integer of bit
length 32. If you are uncertain how to declare your VISA prototypes refer to the VISA
help under the Section titled Type Assignments Table. The VISA help is located in
your NI-VISA directory or folder.
*/

int main(void)
{
    /*
     * First we must call viOpenDefaultRM to get the resource manager
     * handle. We will store this handle in defaultRM.
     */
   status=viOpenDefaultRM (&defaultRM);
   if (status < VI_SUCCESS)
   {
      printf("Could not open a session to the VISA Resource Manager!\n");
      exit (EXIT_FAILURE);
}

    /*
     * Now we will open a VISA session to a device at Primary Address 5.
     * You can use any address for your instrument. In this example we are
     * using GPIB Primary Address 5.
     *
     * We must use the handle from viOpenDefaultRM and we must
     * also use a string that indicates which instrument to open. This
     * is called the instrument descriptor. The format for this string
     * can be found in the NI-VISA User Manual.
     * After opening a session to the device, we will get a handle to
     * the instrument which we will use in later VISA functions.
     * The two parameters in this function which are left blank are
     * reserved for future functionality. These two parameters are
     * given the value VI_NULL.
     *
     */
   status = viOpen (defaultRM, "GPIB0::5::INSTR", VI_NULL, VI_NULL, &instr);
   if (status < VI_SUCCESS)
   {
        printf ("Cannot open a session to the device.\n");
        goto Close;
}
    /*
     * Set timeout value to 5000 milliseconds (5 seconds).
     */
   status = viSetAttribute (instr, VI_ATTR_TMO_VALUE, 5000);
    /*
     * At this point we now have a session open to the instrument
     * We can use this session handle to write
     * an ASCII command to the instrument. We will use the viWrite function
     * to send the string "*IDN?", asking for the device's identification.
     */
   strcpy(stringinput,"*IDN?");
   status = viWrite (instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput),
&writeCount);
   if (status < VI_SUCCESS)
   {
      printf("Error writing to the device\n");
      goto Close;
}
    /*
     * Now we will attempt to read back a response from the device to
     * the identification query that was sent. We will use the viRead
     * function to acquire the data. We will try to read back 100 bytes.
     * After the data has been read the response is displayed.
     */
   status = viRead (instr, buffer, 100, &retCount);
   if (status < VI_SUCCESS)
   {
      printf("Error reading a response from the device\n");
}
   else
   {
      printf("Data read: %*s\n",retCount,buffer);
}
   /*
    * Now we will close the session to the instrument using
    * viClose. This operation frees all system resources.
    */
Close:
   printf("Closing Sessions\nHit enter to continue.");
   fflush(stdin);
   getchar();
   status = viClose(instr);
   status = viClose(defaultRM);

   return 0;
}
```

Communicate Over Sockets – Example

This code demonstrates synchronous read and write commands to a Socket instrument using VISA.

```
/********************************************************************/
/*              Read and Write to an Instrument Example             */
/*                                                                  */
/* This code demonstrates synchronous read and write commands to a  */
/* Socket instrument using VISA.         */
/*                                                                  */
/* The general flow of the code is                                  */
/*      Open Resource Manager                                       */
/*      Open VISA Session to an Instrument                          */
/*      Write the Identification Query Using viWrite                */
/*      Try to Read a Response With viRead                          */
/*      Close the VISA Session                                      */
/********************************************************************/

#if defined(_MSC_VER) && !defined(_CRT_SECURE_NO_DEPRECATE)
/* Functions like strcpy are technically not secure because they do */
/* not contain a 'length'. But we disable this warning for the VISA */
/* examples since we never copy more than the actual buffer size.   */
#define _CRT_SECURE_NO_DEPRECATE
#endif

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include "visa.h"

static ViSession defaultRM;
static ViSession instr;
static ViStatus status;
static ViUInt32 retCount;
static ViUInt32 writeCount;
static unsigned char buffer[100];
static char stringinput[512];

/*
* In every source code or header file that you use it is necessary to prototype
* your VISA variables at the beginning of the file. You need to declare the VISA
* session, VISA integers, VISA strings, VISA pointers, and VISA floating variables.
* Remember that if you are prototyping variables that are to be used as part of the
* VISA session that need this prototyping. As an example, above retCount has been
* prototyped as a static variable to this particular module.   It is an integer of
* bit length 32. If you are uncertain how to declare your VISA prototypes refer
* to the VISA help under the Section titled Type Assignments Table. The VISA
* help is located in your NI-VISA directory or folder.
*/

int main(void)
{
    /*
     * First we must call viOpenDefaultRM to get the resource manager
     * handle. We will store this handle in defaultRM.
     */
   status=viOpenDefaultRM (&defaultRM);
   if (status < VI_SUCCESS)
   {
      printf("Could not open a session to the VISA Resource Manager!\n");
      exit (EXIT_FAILURE);
}
    /*
     * Now we will open a VISA session to a Socket device.
     * There are two examples, one using a TCP-IP address (commented out),
     * and one using an mDNS hostname. Your address and hostname will be different.
     *
     * We must use the handle from viOpenDefaultRM and we must
     * also use a string that indicates which instrument to open. This
     * is called the instrument descriptor. The format for this string
     * can be found in the NI-VISA User Manual.
     * After opening a session to the device, we will get a handle to
     * the instrument which we will use in later VISA functions.
     * The two parameters in this function which are left blank are
     * reserved for future functionality. These two parameters are
     * given the value VI_NULL.
     *
     */
   //status = viOpen(defaultRM, "TCPIP0::172.26.201.114::9001::SOCKET", VI_NULL,
VI_NULL, &instr);
   status = viOpen(defaultRM, "TCPIP0::MG36241A-2133003.local::9001::SOCKET",
VI_NULL, VI_NULL, &instr);
   if (status < VI_SUCCESS)
   {
        printf ("Cannot open a session to the device.\n");
        goto Close;
}
    /*
     * Set timeout value to 5000 milliseconds (5 seconds).
     */
   status = viSetAttribute (instr, VI_ATTR_TMO_VALUE, 5000);
   /*
   * Set the termination character to '\n' for reads, and enable the termination
character.
   * This is needed for raw socket devices as there is no other way to determine
termination.
   */
   status = viSetAttribute(instr, VI_ATTR_TERMCHAR, '\n');
   status = viSetAttribute(instr, VI_ATTR_TERMCHAR_EN, VI_STATE_ASSERTED);

    /*
     * At this point we now have a session open to the instrument
     * We can use this session handle to write
     * an ASCII command to the instrument. We will use the viWrite function
     * to send the string "*IDN?\n", asking for the device's identification.
     * The termination character '\n' is needed for raw socket devices.
     */
   strcpy(stringinput,"*IDN?\n");
   status = viWrite (instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput),
&writeCount);
   if (status < VI_SUCCESS)
   {
      printf("Error writing to the device\n");
      goto Close;
}
    /*
     * Now we will attempt to read back a response from the device to
     * the identification query that was sent. We will use the viRead
     * function to acquire the data. We will try to read back 100 bytes or
     * until the termination character is received.
     * After the data has been read the response is displayed.
     */
   status = viRead (instr, buffer, 100, &retCount);
   if (status < VI_SUCCESS)
   {
      printf("Error reading a response from the device\n");
}
   else
   {
      printf("Data read: %*s\n",retCount,buffer);
}


   /*
    * Now we will close the session to the instrument using
    * viClose. This operation frees all system resources.
    */
Close:
   printf("Closing Sessions\nHit enter to continue.");
   fflush(stdin);
   getchar();
   status = viClose(instr);
   status = viClose(defaultRM);

   return 0;
}
```

1-10 HiSLIP

The HiSLIP (high-speed LAN instrument protocol) is the successor protocol for VXI-11 for TCP-based instruments specified by the IVI foundation. HiSLIP Version 1 message types are supported.

1-11 Read and Write Example

```
/********************************************************************/
/*              Read and Write to an Instrument Example             */
/*                                                                  */
/* This code demonstrates synchronous read and write commands to a  */
/* Socket instrument using VISA.         */
/*                                                                  */
/* The general flow of the code is                                  */
/*      Open Resource Manager                                       */
/*      Open VISA Session to an Instrument                          */
/*      Write the Identification Query Using viWrite                */
/*      Try to Read a Response With viRead                          */
/*      Close the VISA Session                                      */
/********************************************************************/

#if defined(_MSC_VER) && !defined(_CRT_SECURE_NO_DEPRECATE)
/* Functions like strcpy are technically not secure because they do */
/* not contain a 'length'. But we disable this warning for the VISA */
/* examples since we never copy more than the actual buffer size.   */
#define _CRT_SECURE_NO_DEPRECATE
#endif

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include "visa.h"

static ViSession defaultRM;
static ViSession instr;
static ViStatus status;
static ViUInt32 retCount;
static ViUInt32 writeCount;
static unsigned char buffer[100];
static char stringinput[512];


/*
* In every source code or header file that you use it is necessary to prototype
* your VISA variables at the beginning of the file. You need to declare the VISA
* session, VISA integers, VISA strings, VISA pointers, and VISA floating variables.
* Remember that if you are prototyping variables that are to be used as part of the
* VISA session that need this prototyping. As an example, above retCount has been
* prototyped as a static variable to this particular module.   It is an integer of
* bit length 32. If you are uncertain how to declare your VISA prototypes refer
* to the VISA help under the Section titled Type Assignments Table. The VISA
* help is located in your NI-VISA directory or folder.
*/

int main(void)
```

```
{
    /*
     * First we must call viOpenDefaultRM to get the resource manager
     * handle.  We will store this handle in defaultRM.
     */
   status=viOpenDefaultRM (&defaultRM);
   if (status < VI_SUCCESS)
   {
      printf("Could not open a session to the VISA Resource Manager!\n");
      exit (EXIT_FAILURE);
}

    /*
     * Now we will open a VISA session to a Socket device.
     * There are two examples, one using a TCP-IP address (commented out),
     * and one using an mDNS hostname. Your address and hostname will be different.
     *
     * We must use the handle from viOpenDefaultRM and we must
     * also use a string that indicates which instrument to open.  This
     * is called the instrument descriptor.  The format for this string
     * can be found in the NI-VISA User Manual.
     * After opening a session to the device, we will get a handle to
     * the instrument which we will use in later VISA functions.
     * The two parameters in this function which are left blank are
     * reserved for future functionality.  These two parameters are
     * given the value VI_NULL.
     *
     */
   status = viOpen(defaultRM, "TCPIP0::172.26.201.114::hislip0::INSTR", VI_NULL,
VI_NULL, &instr);
   if (status < VI_SUCCESS)
   {
        printf ("Cannot open a session to the device.\n");
        goto Close;
}

    /*
     * Set timeout value to 5000 milliseconds (5 seconds).
     */
   status = viSetAttribute (instr, VI_ATTR_TMO_VALUE, 5000);

    /*
     * At this point we now have a session open to the instrument
     * We can use this session handle to write
     * an ASCII command to the instrument.  We will use the viWrite function
     * to send the string "*IDN?\n", asking for the device's identification.
     * The termination character '\n' is needed for raw socket devices.
     */
   strcpy(stringinput,"*IDN?");
   status = viWrite (instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput),
&writeCount);
   if (status < VI_SUCCESS)
   {
      printf("Error writing to the device\n");
      goto Close;
}
```

```
    /*
     * Now we will attempt to read back a response from the device to
     * the identification query that was sent.  We will use the viRead
     * function to acquire the data.  We will try to read back 100 bytes or
     * until the termination character is received.
     * After the data has been read the response is displayed.
     */
   status = viRead (instr, buffer, 100, &retCount);
   if (status < VI_SUCCESS)
   {
      printf("Error reading a response from the device\n");
}
   else
   {
      printf("Data read: %*s\n",retCount,buffer);
}


   /*
    * Now we will close the session to the instrument using
    * viClose. This operation frees all system resources.
    */
Close:
   printf("Closing Sessions\nHit enter to continue.");
   fflush(stdin);
   getchar();
   status = viClose(instr);
   status = viClose(defaultRM);

   return 0;
}
```

Chapter 2 — Programming with SCPI Commands

2-1 Introduction

This chapter provides an introduction to SCPI programming that includes descriptions of the command types, hierarchal command structure, data parameters, and notational conventions. Information on MG362x1A status system and trigger system programming is also provided.

2-2 Introduction to SCPI Programming

The Standard Commands for Programmable Instruments (SCPI) defines a set of standard programming commands for use by all SCPI compatible instruments. SCPI is intended to give the ATE user a consistent environment for program development. It does so by defining controller messages, instrument responses, and message formats for all SCPI compatible instruments. The set of SCPI commands implemented by the MG362x1A interfaces a comprehensive set of programming functions covering all the major functions of the MG362x1A signal generator.

SCPI Command Types

SCPI commands, which are also referred to as SCPI instructions, are messages to the instrument to perform specific tasks. The MG362x1A command set includes:

- "Common" commands (IEE488.2 mandated commands)
- SCPI required commands
- SCPI optional commands (per SCPI 1993.0)
- SCPI compliant commands that are unique to the MG362x1A

Common Commands

The required common commands are IEEE-488.2 mandated commands that are defined in IEEE-488.2 and must be implemented by all SCPI compatible instruments. These commands (Table 2-1) are identified by the asterisk (*) at the beginning of the command keyword. These commands are used to control instrument status registers, status reporting, synchronization, and other common functions. The common commands and their syntax are described in detail in Chapter 4, "SCPI Commands".

Table 2-1. Common Commands

*CLS	*IDN?	*RST	*STB?
*ESE	*OPC	*SAV	*TRG
*ESE?	*OPC?	*SRE	*TST?
*ESR?	*RCL	*SRE?	*WAI

Required and Optional SCPI Commands

The required SCPI commands are listed in Table 2-2. The optional SCPI commands and MG362x1A unique commands comprise the remainder (major portion) of the MG362x1A command set. They control the majority of the programmable functions of the MG362x1A. See Chapter 4, “SCPI Commands”.

Table 2-2. SCPI Required Commands

```
:STATus
   :OPERation
      [:EVENt]?
      :CONDition?
      :ENABle
   :PRESet
   :QUEStionable
      [:EVENt]?
      :CONDition?
      :ENABle
:SYSTem
   :ERRor?
   :VERSion?
```

Query Commands

All commands, unless specifically noted in the syntax descriptions in Chapter 4, have a query form. As defined in IEEE-488.2, a query is a command with a question mark symbol appended (examples: `*ESR?` and `:FREQuency:CENTer?`). When a query form of a command is received, the current setting associated with the command is placed in the output buffer.

Command Names

Typical SCPI commands consist of one or more keywords, parameters, and punctuation. SCPI command keywords can be a mixture of upper and lower case characters. Except for common commands, each keyword has a long and a short form. In this manual, the long form is presented with the short form in upper case and the remainder in lower case. For example, the long form of the command keyword to control the instrument display is: `DISPlay`.

The short form keyword is usually the first four characters of the long form For example: `DISP` is the short form of the command `DISPlay`. The exception to this is when the long form is longer than four characters and the fourth character is a vowel. In such cases, the vowel is dropped and the short form becomes the first three characters of the long form. Example: the short form of the keyword `POWer` is `POW`.

Some command keywords may have a numeric suffix to differentiate between multiple instrument features such as dual channel inputs. For example: keywords `EXTernal1` and `EXTernal2` (or `EXT1` and `EXT2`).

As with any programming language, the exact command keywords and command syntax must be used. The syntax of the individual commands is described in detail in Chapter 4. Unrecognized versions of long form or short form commands, or improper syntax, will generate an error. Error reporting is described in Chapter 5.

Hierarchical Command Structure

All SCPI commands, except the common commands, are organized in a hierarchical structure similar to the inverted tree file structure used in most computers. The SCPI standard refers to this structure as “the Command Tree.” The command keywords that correspond to the major instrument control functions are located at the top of the command tree. The command keywords for the MG362x1A SCPI command set are shown in Figure 2-1.

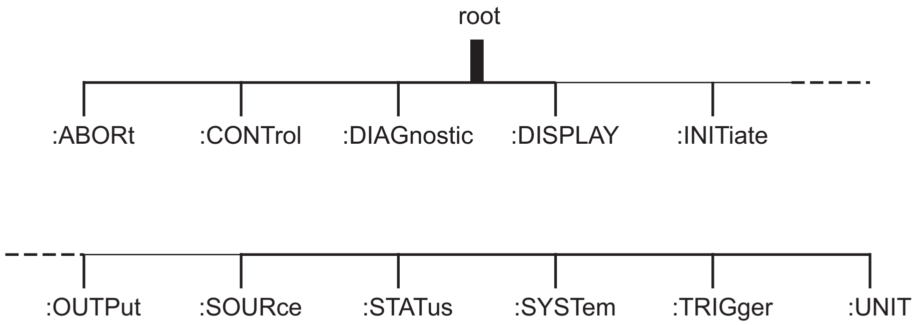


Figure 2-1. SCPI Command Tree

All MG362x1A SCPI commands, except the `ABORt` command, have one or more subcommands (keywords) associated with them to further define the instrument function to be controlled. The subcommand keywords may in turn also have one or more associated subcommands (keywords). Each subcommand level adds another layer to the command tree. The command keyword and its associated subcommand keywords form a portion of the command tree called a command *subsystem*.

Data Parameters

Data parameters, referred to simply as “parameters,” are the quantitative values used as arguments for the command keywords. The parameter type associated with a particular SCPI command is determined by the type of information required to control the particular instrument function. For example, Boolean (`ON|OFF`) type parameters are used with commands that control switch functions.

The command descriptions in Chapter 4 specify the type of data parameter to be used with each command. The most commonly used parameter types are numeric, extended numeric, discrete, and Boolean.

Numeric

Numeric parameters comprise integer numbers, or any number in decimal or scientific notation and may include polarity signs. This includes `<NR1>`, `<NR2>`, and `<NR3>` numeric data as defined in “Parameter Notations” on page 2-5. This type of numeric element is abbreviated as `<NRf>` throughout this document.

Extended Numeric

Extended numeric parameters include values such as `MAXimum` and `MINimum`.

Discrete

Discrete parameters, such as `INTernal` and `EXTernal`, are used to control program settings to a predetermined finite value or condition.

Boolean

Boolean parameters represent binary conditions and may be expressed as `ON`, `OFF` or `1`, `0`.

Unit Suffixes

Unit suffixes are not required for data parameters, provided the values are scaled for the global default units. The MG362x1A SCPI default units are: `Hz` (Hertz) for frequency related parameters and `S` (seconds) for time related parameters. For example, the command below sets the MG362x1A output frequency to 3 GHz.

```
:SOURce:FREQuency:CW 3000000000
```

The global default units may be changed via use of the `:UNIT` Subsystem commands.

2-3 Notational Conventions

The SCPI interface standardizes command syntax and style that simplifies the task of programming across a wide range of instrumentation. As with any programming language, the exact command keywords and command syntax must be used. Unrecognized commands, or improper syntax, will generate an error (refer to Chapter 5 for error reporting).

General Notations

The syntax conventions that are used for all SCPI command keywords and data parameter descriptions in this manual are described below:

: A colon links command keywords together to form commands. The colon is not an actual part of the keyword but is a signal to the SCPI interface parser. A colon must precede a root keyword immediately following a semicolon (See "Notational Examples" on page 2-6).

; A semicolon separates commands if multiple commands are placed on a single program line (See "Notational Examples" on page 2-6).

[ ] Square brackets enclose one or more *optional* parameters.

{ } Braces enclose one or more parameters *that may be included one or more times.*

| A vertical bar indicates "or" and is used to separate alternative parameter options. For example: ON|OFF is the same as ON or OFF.

<> Angle brackets enclose parameter descriptions.

::= Means "is defined as" For example: <a>::=<b><c> indicates that <b><c> can replace <a>.

Space(s), referred to as whitespace, must be used to separate keywords from their associated data parameters. It must not be used between keywords, or inside keywords.

For further information about SCPI command syntax and style, refer to the *Standard Commands for Programmable Instruments (SCPI) 1993.0* document.

Parameter Notations

The following syntax conventions are used for all data parameter descriptions in this manual

<arg> ::= a generic command argument consisting of one or more of the other data types.

<bNR1> ::= boolean values in <NR1> format; numeric 1 or 0 .

<boolean> ::= ON|OFF. Can also be represented as 1 or 0, where 1 means ON and 0 means OFF. Boolean parameters are always returned as 1 or 0 in <NR1> format by query commands.

<integer> ::= an unsigned integer without a decimal point (implied radix point).

<NR1> ::= a signed integer without a decimal point (implied radix point).

<NR2> ::= a signed number with an explicit radix point.

<NR3> ::= a scaled explicit decimal point numeric value with and exponent (e.g., floating point number).

<NRf> ::= <NR1>|<NR2>|<NR3>.

<nv> ::= SCPI numeric value: <NRf>|MIN|MAX|UP|DOWN|DEF|NAN|INF|NINF or other types.

<char> ::= <CHARACTER PROGRAM DATA> Examples: CW, FIXed, UP, and DOWN.

<string> ::= <STRING PROGRAM DATA> ASCII characters surrounded by double quotes, example: "OFF".

<block> ::= IEEE-488.2 block data format.

<NA> ::= Not Applicable.

Notational Examples

The following is an example showing command syntax (It is not an actual command):

```
[SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement] sp dBm|DOWN|UP
```

Command statements read from left to right and from top to bottom. In the command statement above, the `:STEP` keyword immediately follows the `:AMPLitude` keyword *with no separating space*. A space (***sp***) is used between the command string and its argument (a `<nv>` type data parameter).

Note that the first keyword in the command string does not require a leading colon; however, it is good practice to always use a leading colon for *all* keywords. Note also that the `:SOURce` keyword is optional. This is a SCPI convention for all voltage or signal source type instruments that allows shorter command statements to be used.

The following is an example of a multiple command statement that uses two separate commands in a single statement. Note the semicolon used to join the commands. (Also note the leading colon used immediately after the semicolon.)

```
:FREQuency:STARt 10E6;:FREQuency:STOP 20E9
```

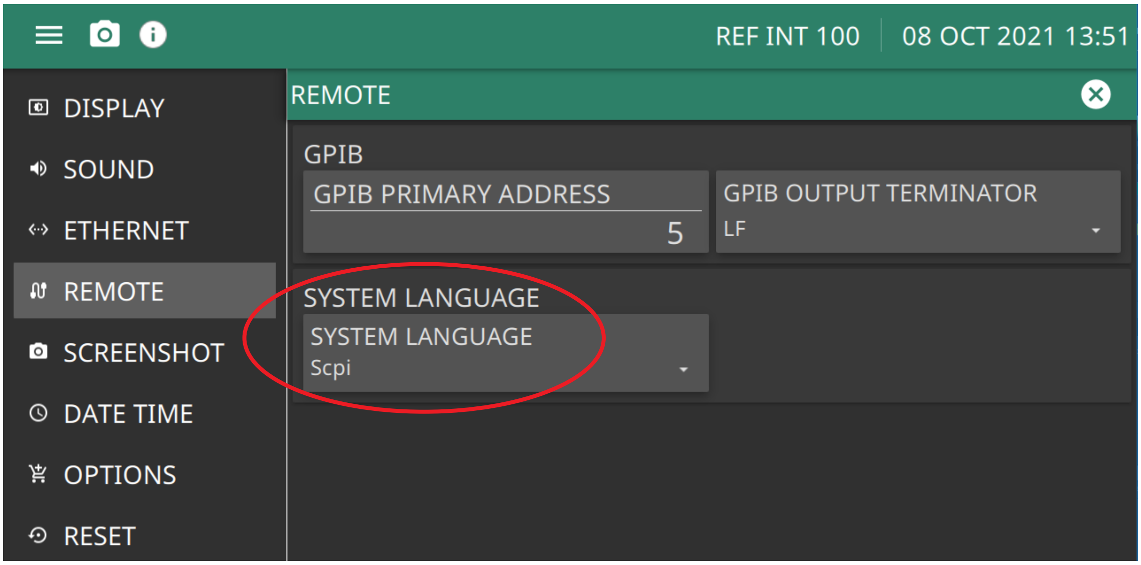
2-4 SCPI Interface Language Selection

The Series MG362x1A Synthesized Signal Generator can be remotely operated using one of two external interface languages—Native or SCPI. Before programming with SCPI commands, it is necessary to select SCPI as the external interface language.

Remote Selection

SCPI can be selected as the MG362x1A interface language during remote operations. To change the interface language:

Press: 3-line icon | SETTINGS | REMOTE |SYSTEM LANGUAGE



GPIB PRIMARY ADDRESS 5

GPIB OUTPUT TERMINATOR LF

SYSTEM LANGUAGE Scpi

Figure 2-2. SYSTEM LANGUAGE Button

2-5 Status System Programming

The MG362x1A status system (shown in Figure 2-3 on page 2-8) consists of the following SCPI-defined status-reporting structures:

- The Instrument Summary Status Byte Group
- The Standard Event Status Group
- The Operational Status Group
- The Questionable Status Group

The following paragraphs describe the registers that make up a status group and explain the status information that each status group provides.

Status Group Registers

In general, a status group consists of a condition register, a transition filter, an event register, and an enable register. Each component is briefly described in the following paragraphs.

Condition Register

The condition register is continuously updated to reflect the current status of the MG362x1A. There is no latching or buffering for this register, it is updated in real time. Reading the contents of a condition register does not change its contents.

Transition Filter

The transition filter is a special register that specifies which types of bit state changes in the condition register will set corresponding bits in the event register. Negative transition filters (NTR) are used to detect condition changes from True (1) to False (0); positive transition filters (PTR) are used to detect condition changes from False (0) to True (1). Setting both positive and negative filters True allows an event to be reported anytime the condition changes. Transition filters are read-write. Transition filters are unaffected by queries or `*CLS` (clear status) and `*RST` commands.

The command `:STATus:PRESet` sets all negative transition filters to all 0’s and sets all positive transition filters to all 1’s.

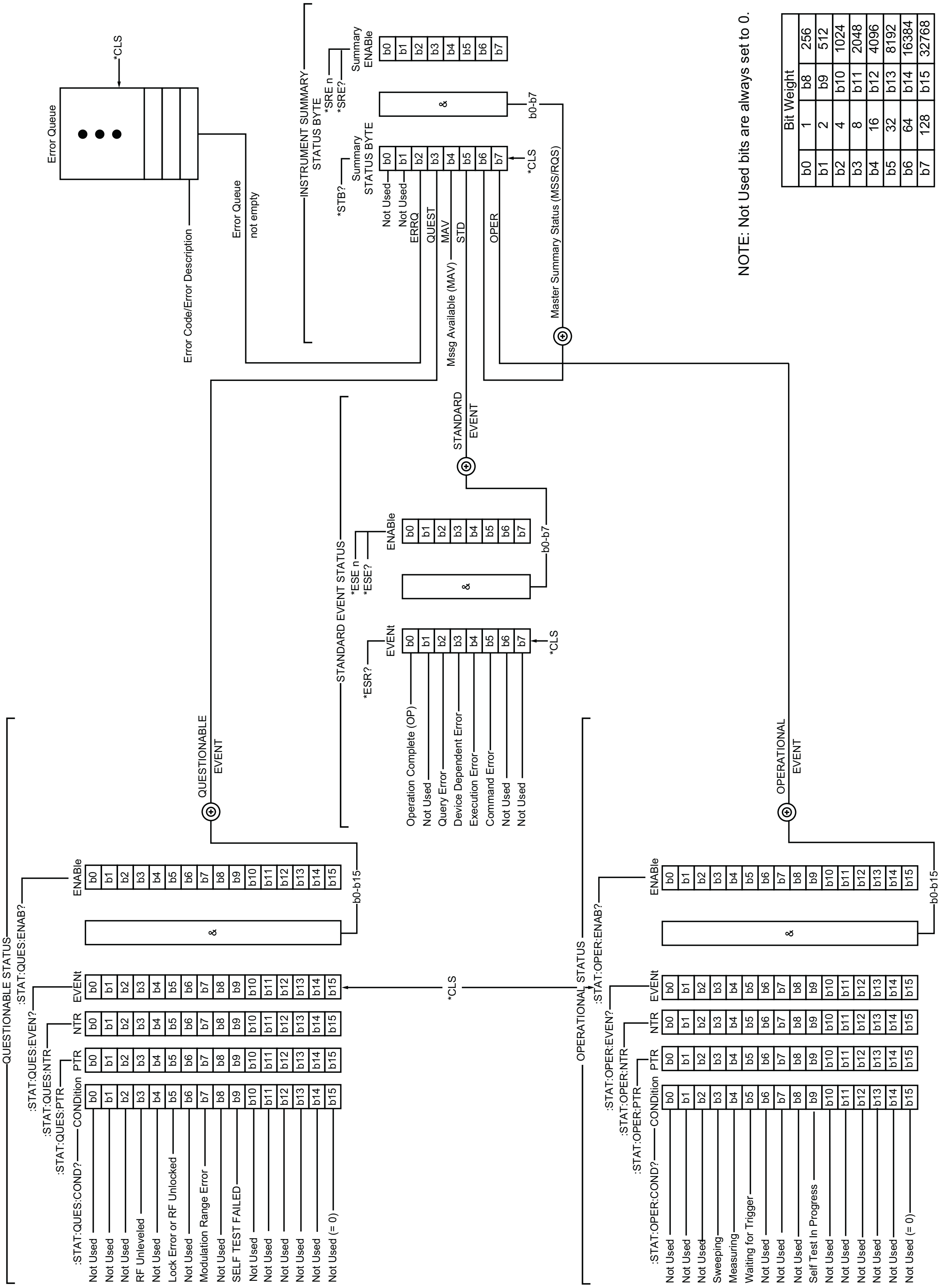
Event Register

The event register latches transition events from the condition register as specified by the transition filter. Bits in the event register are latched, and once set they remain set until cleared by a query or a `*CLS` command. Event registers are read only.

Enable Register

The enable register specifies the bits in the event register that can produce a summary bit. The MG362x1A logically ANDs corresponding bits in the event and enable registers, and ORs all the resulting bits to obtain a summary bit. Summary bits are recorded in the Summary Status Byte. Enable registers are read-write. Querying an enable register does not affect it.

The command `:STATus:PRESet` sets the Operational Status Enable register and the Questionable Status Enable register to all 0’s.



NOTE: Not Used bits are always set to 0.

Bit Weight			
b0	1	b8	256
b1	2	b9	512
b2	4	b10	1024
b3	8	b11	2048
b4	16	b12	4096
b5	32	b13	8192
b6	64	b14	16384
b7	128	b15	32768

Figure 2-3. MG362x1A Status-Reporting Structure

Status Group Reporting

The state of certain MG362x1A hardware and operational events and conditions can be determined by programming the status system. As shown in Figure 2-3, the three lower status groups provide status information to the Summary Status Byte group. The Summary Status Byte group is used to determine the general nature of an event or condition and the other status groups are used to determine the specific nature of the event or condition.

Note Programming commands for the status system, including examples of command usage, can be found in Chapter 4.

The following paragraphs explain the information that is provided by each status group.

Summary Status Byte Group

The Summary Status Byte group, consisting of the Summary Status Byte Enable register and the Summary Status Byte, is used to determine the general nature of a MG362x1A event or condition. The bits in the Summary Status Byte provide the following information:

Table 2-3. Summary Status Byte

Bit	Description
0, 1	Not Used. These bits are always set to 0.
2	Set to indicate the Error Queue contains data. The Error Query command can then be used to read the error message(s) from the queue.
3	Set to indicate the Questionable Status summary bit has been set. The Questionable Status Event register can then be read to determine the specific condition that caused the bit to be set.
4	Set to indicate that the MG362x1A has data ready in its output queue.
5	Set to indicate that the Standard Event Status summary bit has been set. The Standard Event Status register can then be read to determine the specific event that caused the bit to be set.
6	Set to indicate that the MG362x1A has at least one reason to require service. The individual bits in the Status Byte are ANDed with their corresponding Service Request Enable Register bits, then each bit value is ORed and input to this bit.
7	Set to indicate that the Operational Status summary bit has been set. The Operational Status Event register can then be read to determine the specific condition that caused the bit to be set.

Standard Event Status Group

The Standard Event Status group, consisting of the Standard Event Status register (an Event register) and the Standard Event Status Enable register, is used to determine the specific event that set bit 5 of the Summary Status Byte. The bits in the Standard Event Status register provide the following information:

Table 2-4. Standard Event Status

Bit	Description
0	Set to indicate that all pending MG362x1A operations were completed following execution of the "*OPC" command.
1	Not Used. The bit is always set to 0.
2	Set to indicate that a query error has occurred. Query errors have SCPI error codes from –499 to –400.
3	Set to indicate that a device-dependent error has occurred. Device-dependent errors have SCPI error codes from –399 to –300 and 1 to 32767.
4	Set to indicate that an execution error has occurred. Execution errors have SCPI error codes from –299 to –200.
5	Set to indicate that a command error has occurred. Command errors have SCPI error codes from –199 to –100.
6,7	Not Used. The bits are always set to 0.

Operational Status Group

The Operational Status group, consisting of the Operational Condition register, the Operational Positive Transition register, the Operational Negative Transition register, the Operational Event register, and the Operational Event Enable register, is used to determine the specific condition that set bit 7 in the Summary Status Byte. The bits in the Operational Event register provide the following information:

Table 2-5. Operational Status Event

Bit	Description
0–2	Not Used. The bits are always set to 0.
3	Set to indicate that a sweep is in progress.
4	Set to indicate that the MG362x1A is measuring.
5	Set to indicate that the MG362x1A is in an armed "wait for trigger" state.
6	Not Used. The bit is always set to 0.
7	Not Used. The bit is always set to 0.
8	Not Used. The bit is always set to 0.
9	Set to indicate that MG362x1A self-test is in progress.
10–14	Not Used. The bits are always set to 0.
*15	Always 0. The use of Bit 15 is not allowed by SCPI.

Questionable Status Group

The Questionable Status group, consisting of the Questionable Condition register, the Questionable Positive Transition register, the Questionable Negative Transition register, the Questionable Event register, and the Questionable Event Enable register, is used to determine the specific condition that set bit 3 in the Summary Status Byte. The bits in the Questionable Event register provide the following information:

Table 2-6. Questionable Event Status

Bit	Description
0–2	Not Used. The bits are always set to 0.
3	Set to indicate an RF unleveled condition.
4	Not Used. The bit is always set to 0.
5	Set to indicate a phase-lock error or RF unlocked condition.
6	Not Used. The bit is always set to 0.
7	Set to indicate a modulation range error.
8	Not Used. The bit is always set to 0.
9	Set to indicate that self-test failed.
10	Not Used. The bit is always set to 0.
11	Not Used. The bit is always set to 0.
12-14	Not Used. The bits are always set to 0.
15	Always 0. The use of Bit 15 is not allowed by SCPI.

2-6 Trigger System Programming

The MG362x1A trigger system is used to synchronize signal generator actions with software trigger commands. The MG362x1A follows the layered trigger model used in SCPI instruments. The following paragraphs describe operation and programming of the signal generator trigger system. The structure and components of the MG362x1A trigger model are shown in Figure 2-4.

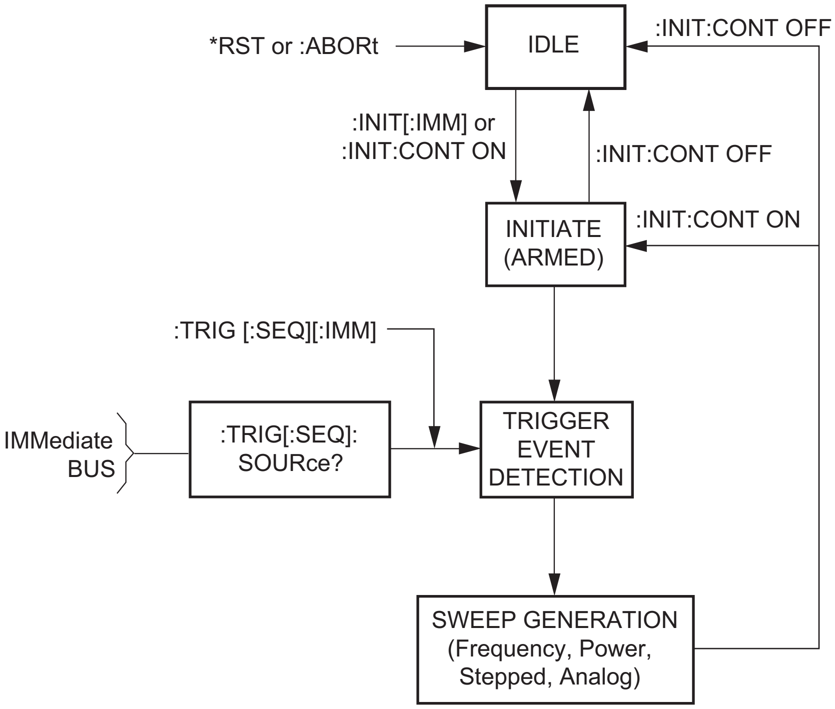


Figure 2-4. MG362x1A Trigger Model

Trigger System Operation

Turning power on, or sending `*RST` or `:ABORt` forces the trigger system into the *idle* state. The trigger system remains in the *idle* state until it is initiated. Trigger system initiation can happen on a continuous basis (`:INITiate:CONTinuous ON`) or on a demand basis (`:INITiate:CONTinuous OFF`). When the command `:INITiate:CONTinuous` is set to `OFF`, the trigger system is initiated by the `:INITiate[:IMMediate]` command. Note that `*RST` sets `:INITiate:CONTinuous` to `OFF`.

Once initiated, the trigger system enters an armed (wait for trigger) state. The trigger signal selected by the command `:TRIGger[:SEQuence]:SOURce` is examined until a TRUE condition is detected. The trigger signal selections are:

`IMMediate`	The trigger signal is always TRUE.
`BUS`	The trigger signal is either the GPIB `<GET>` (Group Execute Trigger) message or the `*TRG` command.
`HOLD`	The trigger signal is never TRUE.

When a TRUE condition is detected, sweep generation of the selected sweep starts.

The command `:TRIGger[:SEQuence][:IMMediate]` provides a one-time override of the normal downward path in the trigger-event-detection state by forcing a TRUE trigger signal regardless of the setting for `:TRIGger[:SEQuence]:SOURce`.

Upon sweep completion, if :INITiate:CONTinuous is set OFF, the trigger system returns to the *idle* state. If :INITiate:CONTinuous is set to ON, the trigger system returns to the *armed* (wait for trigger) state.

Auto Trigger Mode

Setting the command :INITiate:CONTinuous to ON and the command :TRIGger[:SEQuence]:SOURce to IMMediate, places the trigger system in an auto trigger mode. This causes continuous generation of the selected sweep.

ABORt

The :ABORt command resets any sweep in progress and immediately returns the trigger system to the idle state. Unlike *RST, :ABORt does not change the settings programmed by other commands.

Chapter 3 — Native Programming

3-1 Introduction

This chapter provides information for remote operation of MG362x1A Synthesized Signal Generator using MG362x1A native commands. All product-specific commands that are accepted and implemented by the MG362x1A are listed and described by function. Sample programs showing usage of the commands are also included. MG362x1A will be used to reference all of the covered models throughout this programming manual.

3-2 Command Codes

These native commands allow the user to program all front panel and menu functions (except for power on/standby). Each native command is mnemonic code that represents an instrument command or parameter; for example: RST (reset).

Table 3-1 is a listing of all MG362x1A native command mnemonic codes grouped into functional categories. The listing for each category includes references to the paragraph and page number in this chapter where a complete description of that group of commands can be found.

Table 3-1. MG362x1A Native Command Codes Listed by Function

Native Command Group Function	Command Codes
Data Entry/Data Terminator Commands	CLR
CW Frequency Commands	ACW, CF0, CF1, F1
Configuration Commands	RO0, RO1
Output Commands	*IDN?,
Parameter Entry Commands	PDY
Power Commands	AT0, AT1, ATT(xx), L0, L1, LO0, LO1, LOS, PDY, RF0, RF1
Modulation Commands	D1, IP, OPP, P0, PC4, PER, PMD1, PMD2, PMD3, PMD4, PTG1, PTG2, PTG3, PTG4, PTG5, PTG6, PW, XP
Service Request and Status Byte Commands	CSB, EL0, EL1, FB0, FB1, LE0, LE1, LS0, LS1, MB0, MB1, MB2, OEM, OES, OSB, PE0, PE1, SB0, SB1 SQ0, SQ1, UL0, UL1, W1
Self Test Commands	OSR, TST
Miscellaneous Commands	DS0, DS1, P3, P4, RL, RST, SYST:LANG "SCPI", SYST:LANG?

3-3 Data Input Restrictions

The MG362x1A signal generator does ***not*** accept parameter or data entries in an exponential or scientific notation format. The accepted data formats are as follows:

- A decimal or integer format for entering parameters and data
- A binary-byte format for entering the status byte mask commands

3-4 Parameter and Data Entry Commands

Open a parameter command for data entry. Parameter commands are listed in each corresponding parameter table in this chapter.

Data Entry

When a parameter is open for data entry, its value can be changed as follows:

- By sending a numeric value followed by the appropriate terminator code
- By incrementing or decrementing its value using an associated step size

An appropriate data terminator *must* be used to terminate a numeric-parameter entry, and it *must* immediately follow the numeric value. If it does not, a parameter entry error will result.

The parameter and data entry commands do not affect the signal generator's output unless the parameter being changed is also the current output parameter. The commands, therefore, may be used to change the preset values of parameters without altering the MG362x1A output.

Table 3-2. Data Entry Commands

Mnemonic Code	Description
CLR	Clear data entry
PDY	Opens the internal pulse delay1 parameter

3-5 Configuration Commands

Table 3-3 lists the configuration command mnemonic codes. The system configuration selections made with GPIB commands remain in effect when the instrument is returned to local control.

Table 3-3. Configuration Commands

Mnemonic Code	Function
RO0	Selects RF to be on at reset.
RO1	Selects RF to be off at reset.

3-6 CW Frequency Commands

Table 3-4 lists the CW frequency command mnemonic codes. These commands call up each of the preset (or previously set) CW frequencies. Each command causes its associated CW frequency to be output and opens that frequency's parameter for data entry.

Table 3-4. CW Frequency Commands

Mnemonic	Description
ACW	Set CW mode. Activates the currently open frequency parameter as CW.
CF0	Set CW mode at F0. Opens F0 parameter.
CF1	Set CW mode at F1. Opens F1 parameter.
F1	Opens the F1 parameter.

3-7 Output Power Leveling Commands

Table 3-5 lists the output power leveling command mnemonic codes that include electing an RF output power level, leveling the CW output power, and turning the output power leveling off. In addition, commands are provided for the level offset and step attenuator decoupling functions.

RF Output Power Level Selection

The command, L1 calls up each of the preset (or previously set) output power levels. Each command causes its associated RF power level to be output and opens that power levels parameter for data entry. Each command will also deselect any other previously programmed power level and will turn off an active power level sweep.

Attenuator Decoupling

The attenuator decoupling function provides for decoupling of the step attenuator (if equipped) from the ALC system. The AT1 command decouples the step attenuator, the ATT(xx) command provides for setting the step attenuator value for 0 to 110 dB in 10 dB increments, and the AT0 command deselects the attenuator decoupling function.

Table 3-5. Power Leveling Commands

Mnemonic Code	Function
AT0	Deselects step attenuator decoupling.
AT1	Selects step attenuator decoupling.
ATT(xx)	Sets the step attenuator value to xx (01 = 10dB, 02 = 20dB, etc...).
L0	Sets RF output power level to L0. Opens L0 parameter.
LO0	Turns off the Level Offset function.
LO1	Turns on the Level Offset function.
LOS	Opens the level offset parameter.
L1	Sets RF output power level to L1. Opens L1 parameter.
RF0	Turns off the RF output.
RF1	Turns on the RF output.

3-8 Modulation Commands

Table 3-6 on page 3-5 lists the modulation command mnemonic codes. These commands provide for ΦM and pulse modulation of the signal generator's output signal using modulating signals from the ΦM, pulse generators, or an external source.

Phase Modulation

Narrow and Wide ΦM modes are available. In Narrow ΦM mode, the maximum ΦM deviation is the lesser of ±3 radians or ±5 MHz/rate for DC to 8 MHz rates. In Wide ΦM mode, the maximum ΦM deviation is the lesser of ±400 radians or ±10 MHz/rate for DC to 1 MHz rates.

Note	If the ΦM function is not installed in the signal generator, the ΦM function commands produce syntax errors.

Pulse Modulation

Two pulse modulation modes are available—Internal and External. In Internal mode, pulse modulation of the output signal is accomplished by using a modulating signal from the internal pulse generator. In External mode, pulse modulation of the output signal is achieved using a modulating signal from an external source.

The internal pulse generator has four pulse modes that include single, doublet (double pulse), triplet (triple pulse), and quadruplet (quadruple pulse). Individual pulse widths and delays can be set for each of the pulses in a mode. The pulse generator has a 100 MHz clock rate. The 100 MHz clock rate produces higher resolution pulses and allows higher PRFs. The clock rate command is available only for backward compatibility so it doesn't create a syntax error. It is not operationally functional.

The internal pulse generator can be internally triggered, externally triggered, internally and externally triggered with delay, and gated. There is also a composite trigger mode in which an external pulse is summed with the internal pulse to pulse modulate the output signal. Refer to the MG362x1A technical data sheet (11410-00928) for pulse period, pulse width, and pule delay specifications.

Internal Pulse Modulation Function

The `IP` command turns on the internal pulse modulation function. The internal pulse mode is selected using the `PMD(x)` command and the internal pulse trigger is selected using the `PTG(x)` command. The pulse period can be set using the `PER` parameter entry command; the pulse frequency can be set using the `PR` parameter entry command. Individual pulse widths can be set using the `W1` (or `PW`) parameter entry commands. Individual pulse delays can be set using the `D1` (or `PDY`) parameter entry commands. The `P0` (or `SW0`) command turns off the pulse modulation function.

Square wave pulse modulation of the output signal by one of four internal modulating signals is available using the following commands:

Programming Example:

Programming "`IP PMD1 PTG2 PER 1 MS W1 2.5 US D1 10 US`" turns on the internal pulse modulation function; selects the doublet pulse mode, free run trigger mode, sets the pulse period to 1 ms, pulse width1 to 2.5 µs, and delay1 to 10 µs.

External Pulse Modulation Function

The `XP` command turns on the external pulse modulation function. The `PO` command turns off the pulse modulation function.

Modulation Command List

Table 3-6. Modulation Command List

Mnemonic Code	Function
D1	Set the internal pulse delay1 parameter.
IP	Turns on internal pulse modulation.
OPP	Returns the internal pulse period value (in ms) to the controller.
PC4	Set the 100 MHz internal pulse clock rate.
PDY	Set the power delay parameter for D1
PER	Set the internal pulse period parameter.
PW	Set the internal pulse width 1 parameter.
P0	Turns off the internal or external pulse modulation function (This is the default mode).
PMD(x)	Selects the internal pulse mode, where x = 1-single, 2-doublet, 3-triplet, 4-quadruplet.
PTG(x)	Selects the internal pulse trigger, where x = 1-free run, 2-gated, 3-delayed, 4-triggered, 5-triggered with delay, 6-composite.
W1	Set the internal pulse width 1 parameter.
XP	Turns on the external pulse modulation function. Disables the internal pulse modulation function, if previously programmed.

3-9 Output Commands

Table 3-7 lists the output command mnemonic codes. These commands provide for the output of data from the CW generator to the controller is enabled and detected.

Table 3-7. Output Commands

Mnemonic Code	Function
*IDN?	Requests device identification.
OEM	Returns the Extended SRQ Mask bytes (3 binary bytes) to the controller.
OES	Returns the GPIB Status bytes (3 binary bytes, no terminator) to the controller.
OPP	Returns the internal pulse period value (in µs) to the controller.
OSB	Each Primary byte status reporting bit returns the primary GPIB Status byte (1 binary byte, no terminator) to the controller. Each extended status reporting bit is enabled and detected.
OSR	Returns the self-test results (6 binary bytes, no terminator) to the controller.

3-10 SRQ and Status Byte Commands

Table 3-8 lists the Service Request (SRQ) and Status Byte command mnemonic codes. These commands enable the signal generator to request service from the controller when certain, predefined conditions exist.

Status Bytes

The MG362x1A has three GPIB status bytes that include the primary and two extended status bytes. Each status reporting bit, with the exception of primary status byte bit 6 (SRQ), is set when the condition on which it reports is detected. These changes in status byte bit settings can be read by the controller. The contents of the primary status byte is returned to the controller in response to a serial poll or the `OSB` command.

The contents of the primary status byte and the two extended status bytes are returned to the controller in response to the `OES` command.

SRQ Generation

The signal generator can generate GPIB service requests (SRQs) to report instrument status and syntax errors to the controller. The signal generator will generate an SRQ if:

1. The SRQ generation function has been enabled using the `SQ1` command *and*,
2. One (or more) of the status reporting functions is true *and*,
3. The primary status byte bit associated with the true status reporting function has been enabled.

Bits in the primary status byte can be enabled by either of two methods. The first uses the FB1/FB0, UL1/UL0, LE1/LE0, and SE1/SE0 commands.individually enable or disable each bit. The second method uses a single 8-bit status byte mask (MB0) to enable any or all of the primary status byte bits.

Primary and Extended Status Bytes

Note	All status byte bits are latched except for those indicated with the “*”. Once set, an OES or OSB command must be received before the condition will be reset. The primary status byte bit 6 (SRQ) is cleared by a serial poll only.

Primary Status Byte

Extended Status Byte 2	SRQ	Syntax Error	Parameter Range Error	Lock Error	RF Unleveled	Not Used	Extended Status Byte 1
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Primary status byte bit 0 is set whenever one of the status conditions reported by an extended status byte 1 is true and the associated status bit is enabled. This bit is cleared when the controller sends the `OES` command.

Primary status byte bit 7 is set whenever one of the status conditions reported by an extended status byte 2 is true and the associated status bit is enabled. This bit is cleared when the controller sends the `OES` command.

Primary status byte bit 6 (SRQ) is not maskable. This bit is enabled by the `SQ1` command and cleared by a serial poll.

Extended Status Byte 1

Not Used	Not Used	Not Used	Not Used	RF* Locked	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Extended status byte 1 bits are enabled by the extended status byte 1 mask command, `MB1`.

Extended status byte 1 bit 3 (RF Locked): the setting of this bit is blocked or unblocked by the commands, `LS0` and `LS1`.

Extended Status Byte 2

Not Used	Not Used	Not Used	RF* Unlocked	Not Used	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Extended status byte 2 bits are enabled by the extended status byte 2 mask command, `MB2`.

The setting of extended status byte 2 bit 4 (RF Unlocked) is blocked or unblocked by the commands, `EL0` and `EL1`.

Table 3-8. SRQ and Status Byte Commands

Mnemonic Code	Function
CSB	Clears all Native status bytes.
EL0	Inhibits the RF Unlocked bit update. ESB2 bit 4.
EL1	Enables the RF Unlocked bit update. ESB2 bit 4.
FB0	Inhibits Extended Status Byte 1 SRQ generation. Primary byte bit 0.
FB1	Enables Extended Status Byte 1 SRQ generation. Primary byte bit 0.
LE0	Inhibits Lock Error SRQ generation. Primary byte bit 3.
LE1	Enables Lock Error SRQ generation. Primary byte bit 3.
LS0	Inhibits RF Locked bit update. ESB1 bit 3.
LS1	Enables RF Locked bit update. ESB1 bit 3.
MB0	Sets the enable mask byte for the Primary Status Byte.
MB1	Sets the enable mask byte for Extended Status Byte 1.
MB2	Sets the enable mask byte for Extended Status Byte 2.
OEM	Returns the Extended SRQ Mask bytes (3 binary bytes) to the controller.
OES	Returns the GPIB Status bytes (3 binary bytes) to the controller.
OSB	Returns the Primary GPIB Status byte to the controller.
PE0	Inhibits Parameter Range Error SRQ generation. Primary byte bit 4.
PE1	Enables Parameter Range Error SRQ generation. Primary byte bit 4.
SB0	Inhibits Extended Status Byte 2 SRQ generation. Primary byte bit 7.
SB1	Enables Extended Status Byte 2 SRQ generation. Primary byte bit 7.
SQ0	Disables the SRQ generation function when a Primary byte bit is set.
SQ1	Enables SRQ generation when a Primary byte bit is enabled.
UL0	Inhibits RF Unleveled SRQ generation. Primary byte bit 2.
UL1	Enables RF Unleveled SRQ generation. Primary byte bit 2.

The MG362x1A has a software mask that permits manipulation of the three status bytes over the bus that are accomplished by sending the command codes `MB0`, `MB1`, `MB2`, or all three at once, followed by an argument that assigns an on/off condition for each bit in the byte. A status byte example is shown in Figure 3-1.

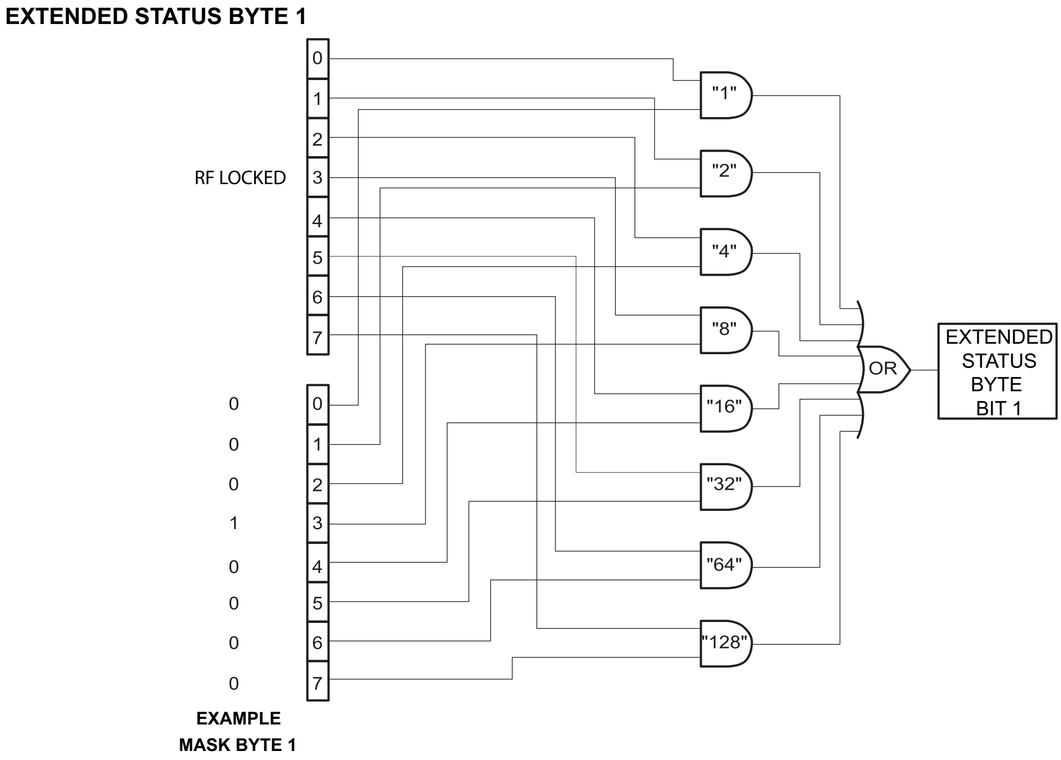


Figure 3-1. Status Byte Example

Example: `MB1\x08`

MB1 should be followed by a single binary byte. This example uses the hexadecimal escape sequence "\x08" to represent a single byte with value 0x08. Enables bit 3 in extended status byte 1. Same as the LS1 command.

3-11 Self Test Command

Table 3-9 lists the self test command mnemonic code. This command provides for executing a signal generator self test.

When a `TST` command is received, the signal generator performs a self test, then returns a single byte, unterminated. A "P" (for pass) or an "F" (for fail). It also generates six self test results bytes.

Table 3-9. Self Test Command

Mnemonic Code	Function
TST	Executes a signal generator self test. Extended Status Byte 1 bit 0 is set if self test fails; bit 2 is set when self test is complete.
OSR	Returns the six self test results bytes to the controller.

Refer to Appendix 5, "Self-Test Messages" for the self.test error and messages descriptions. The "P or "F" character placed on the bus by the signal generator self test must be cleared from the output buffer (read by the controller) before another output command, such as OSR, is sent. If it is not cleared, the first character of the next output will be missing.

Self Test Results

The six self test results bytes that identify the reporting function of each bit is shown below.

Self Test Results Byte 1

Not Used	Not Used	Power Supply Voltage(s) are Out of Reg	Not Used	Not Used	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Self Test Results Byte 2

Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Self Test Results Byte 3

Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Self Test Results Byte 4

Not Used	Not Used	Switch Filter Failed	Not Used	Not Used	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Self Test Results Byte 5

Not Used	Frequency Extension Module (FEM) Failed	Not Used	Not Used	Mod Processor Failed	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Self Test Results Byte 6

Self Test is Complete	Not Used	Not Used	RF Was Off When Self Test Started	Not Used	Not Used	Not Used	Not Used
Bit 7 (128)	Bit 6 (64)	Bit 5 (32)	Bit 4 (16)	Bit 3 (8)	Bit 2 (4)	Bit 1 (2)	Bit 0 (1)

Self Test Example

This code demonstrates synchronous read and write commands to a Socket instrument using VISA to perform and return self test results.

```
/********************************************************************/
/*              Read and Write to an Instrument Example             */
/*                                                                  */
/* This code demonstrates synchronous read and write commands to a  */
/* Socket instrument using VISA.         */
/*                                                                  */
/* The general flow of the code is                                  */
/*      Open Resource Manager                                       */
/*      Open VISA Session to an Instrument                          */
/*      Write the Identification Query Using viWrite                */
/*      Try to Read a Response With viRead                          */
/*      Close the VISA Session                                      */
/********************************************************************/

#if defined(_MSC_VER) && !defined(_CRT_SECURE_NO_DEPRECATE)
/* Functions like strcpy are technically not secure because they do */
/* not contain a 'length'. But we disable this warning for the VISA */
/* examples since we never copy more than the actual buffer size.   */
#define _CRT_SECURE_NO_DEPRECATE
#endif

#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#include "visa.h"

static ViSession defaultRM;
static ViSession instr;
static ViStatus status;
static ViUInt32 retCount;
static ViUInt32 writeCount;
static unsigned char buffer[100];
static char stringinput[512];

/*
* In every source code or header file that you use it is necessary to prototype
* your VISA variables at the beginning of the file. You need to declare the VISA
* session, VISA integers, VISA strings, VISA pointers, and VISA floating variables.
* Remember that if you are prototyping variables that are to be used as part of the
* VISA session that need this prototyping. As an example, above retCount has been
* prototyped as a static variable to this particular module. It is an integer of
* bit length 32. If you are uncertain how to declare your VISA prototypes refer
* to the VISA help under the Section titled Type Assignments Table. The VISA
* help is located in your NI-VISA directory or folder.
*/

int main(void)
{
    /*
     * First we must call viOpenDefaultRM to get the resource manager
     * handle. We will store this handle in defaultRM.
     */
```

```
   status=viOpenDefaultRM (&defaultRM);
   if (status < VI_SUCCESS)
   {
      printf("Could not open a session to the VISA Resource Manager!\n");
      exit (EXIT_FAILURE);
   }

    /*
     * Now we will open a VISA session to a Socket device.
     * There are two examples, one using a TCP-IP address (commented out),
     * and one using an mDNS hostname. Your address and hostname will be different.
     *
     * We must use the handle from viOpenDefaultRM and we must
     * also use a string that indicates which instrument to open. This
     * is called the instrument descriptor. The format for this string
     * can be found in the NI-VISA User Manual.
     * After opening a session to the device, we will get a handle to
     * the instrument which we will use in later VISA functions.
     * The two parameters in this function which are left blank are
     * reserved for future functionality. These two parameters are
     * given the value VI_NULL.
     *
     */
     status = viOpen(defaultRM, "TCPIP0::172.26.201.114::9001::SOCKET", VI_NULL,
     VI_NULL, &instr);
   if (status < VI_SUCCESS)
   {
        printf ("Cannot open a session to the device.\n");
        goto Close;
   }

    /*
     * Set timeout value to 10000 milliseconds (10 seconds).
     */
   status = viSetAttribute (instr, VI_ATTR_TMO_VALUE, 10000);

   /*
   * Set the termination character to '\n' for reads, and enable the termination
character.
   * This is needed for raw socket devices as there is no other way to determine
termination.
   */
   status = viSetAttribute(instr, VI_ATTR_TERMCHAR, '\n');
   status = viSetAttribute(instr, VI_ATTR_TERMCHAR_EN, VI_STATE_ASSERTED);

   /*
   * For Native selftest and results, we don't use the termination character during
reads.
   * We must unassert termination enabled in case any of the binary result data
happens to be 0x0A.
   * Native TST returns exactly 1 byte, no termination.
   * Native OSR returns exactly 6 bytes, no termination.
   */
   status = viSetAttribute(instr, VI_ATTR_TERMCHAR_EN, VI_STATE_UNASSERTED);
    /*
     * At this point we now have a session open to the instrument
     * We can use this session handle to write
     * an ASCII command to the instrument. We will use the viWrite function
     * to send the string "TST", asking the device to perform selftest and return a
result.
     */
   /*
   * This program assumes Native language mode is already selected on the instrument.
   * Native selftest, TST.
   */
   strcpy(stringinput,"TST\n");
   status = viWrite (instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput),
&writeCount);
   if (status < VI_SUCCESS)
   {
      printf("Error writing to the device\n");
      goto Close;
   }

    /*
     * Now we will attempt to read back a response from the device to
     * the selftest query that was sent. We will use the viRead
     * function to acquire the data. We will try to read back exactly 1 byte.
     * After the data has been read the response is displayed.
     */
   status = viRead (instr, buffer, 1, &retCount);
   if (status < VI_SUCCESS)
   {
      printf("Error reading a response from the device\n");
   }
   else
   {
      printf("Data read: %*s\n",retCount,buffer);
   }

   /*
    * This program assumes Native language mode is already selected on the
instrument.
    * Native selftest results, OSR.
    */
   strcpy(stringinput, "OSR\n");
   status = viWrite(instr, (ViBuf)stringinput, (ViUInt32)strlen(stringinput),
&writeCount);
   if (status < VI_SUCCESS)
   {
       printf("Error writing to the device\n");
       goto Close;
   }

   /*
    * Now we will attempt to read back a response from the device to
    * the selftest results query that was sent. We will use the viRead
    * function to acquire the data. We will try to read back exactly 6 bytes.
    * After the data has been read the binary response is displayed.
    */
   status = viRead(instr, buffer, 6, &retCount);
   if (status < VI_SUCCESS)
   {
       printf("Error reading a response from the device\n");
   }
   else
   {
       printf("Data read: 0x%02X 0x%02X 0x%02X 0x%02X 0x%02X 0x%02X \n", buffer[0],
buffer[1], buffer[2], buffer[3], buffer[4], buffer[5]);
   }


   /*
    * Now we will close the session to the instrument using
    * viClose. This operation frees all system resources.
    */
Close:
   printf("Closing Sessions\nHit enter to continue.");
   fflush(stdin);
   getchar();
   status = viClose(instr);
   status = viClose(defaultRM);

   return 0;
}
```

3-12 Miscellaneous Commands

Table 3-10 is a list of miscellaneous command mnemonic codes that do not fit into any of the other classifications. These commands provide the following operations:

- Secure Mode
- Returning the MG369xC to local control
- Instrument Resetl

Table 3-10. Miscellaneous Commands

Mnemonic Code	Function
DS0	Turns on the secure mode.
DS1	Turns off the secure mode.
RL	Returns the instrument to local (front panel) control.
RST	Resets the MG362x1A to its default settings. Remote settings are not reset, e.g. Remote language, Ethernet settings, GPIB settings are not reset.

3-13 Command Directory

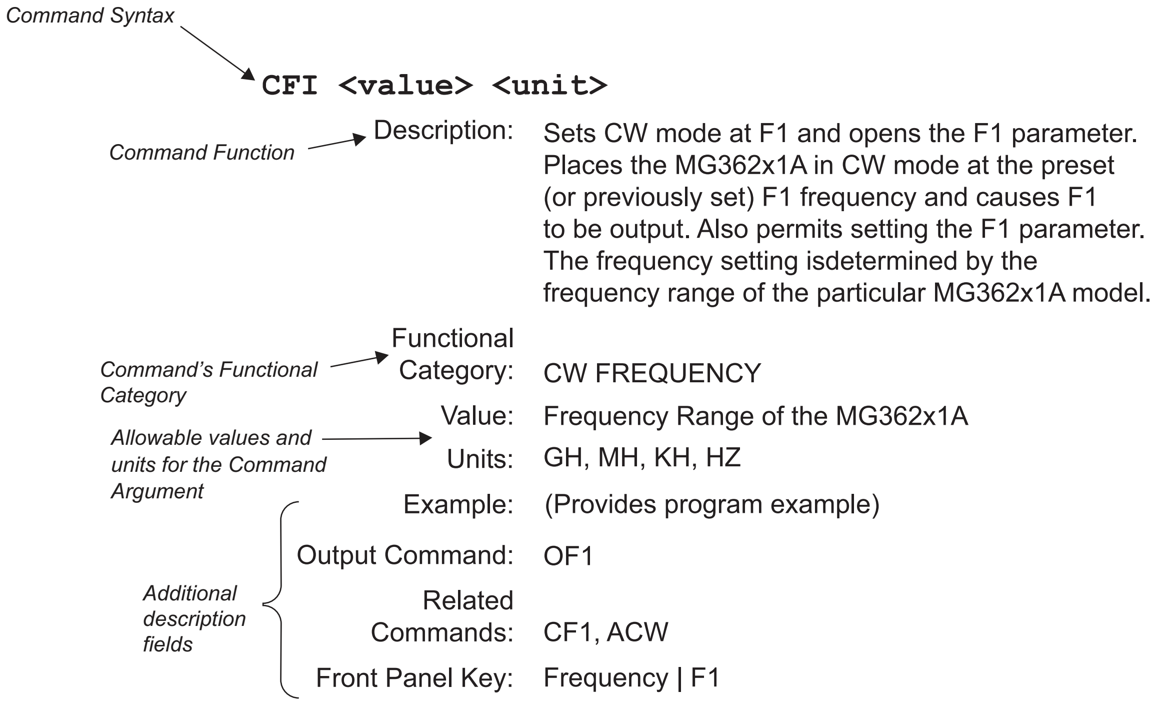
This section provides alphabetically-ordered listings and descriptions of all MG362x1A Native programming commands.

3-14 Typographic Conventions

The typographic conventions, abbreviations, and syntax legend used throughout this chapter to define the Native commands are described in Figure 3-2.

3-15 Functional Categories

The MG362x1A's Native functional categories provide descriptive details and tabular data that apply to the category as a whole.



Command Syntax
Command Function
Command's Functional Category
Allowable values and units for the Command Argument
Additional description fields

CFI <value> <unit>

Description: Sets CW mode at F1 and opens the F1 parameter. Places the MG362x1A in CW mode at the preset (or previously set) F1 frequency and causes F1 to be output. Also permits setting the F1 parameter. The frequency setting isdetermined by the frequency range of the particular MG362x1A model.

Functional Category: CW FREQUENCY

Value: Frequency Range of the MG362x1A

Units: GH, MH, KH, HZ

Example: (Provides program example)

Output Command: OF1

Related Commands: CF1, ACW

Front Panel Key: Frequency | F1

Figure 3-2. Typographic Conventions for the Native Command Listings

3-16 Commands

This section provides an alphabetical listing of the SCPI and Native commands used to program the series MG362x1A Synthesized Signal Generator.

*IDN?

Description: Requests device identification.

Returns an identification string in IEEE 488.2 specified <NR1> format (four fields separated by commas). The fields are <manufacturer>, <model>, <serial #>, <firmware version>.

Functional Category: OUTPUT

Front Panel Key: N/A

ACW

Description: Activates the currently open frequency parameter as CW.

Activates the frequency parameter that is currently open for data entry as CW, switches to CW if not in CW mode.

Functional Category: CW FREQUENCY

Example: `F1, ACW`

Related Commands: `CF1`

Front Panel Key: N/A

AT0

Description: Deselects step attenuator decoupling.

Deselects the step attenuator decoupling function. This returns the MG362x1A to its normal (coupled) leveling mode. In this mode, when the desired power level is set, the correct combination of level and step attenuator setting is determined by the instrument firmware.

Functional Category: POWER LEVELING

Related Commands: `AT1`, `ATT(xx)`

Front Panel Key: Level | Attenuator

AT1

Description: Selects ALC step attenuator decoupling.

This function decouples the step attenuator (if equipped) from the ALC system. This permits separate setting the ALC level and step attenuator.

Functional Category: POWER LEVELING

Related Commands: `AT0`, `ATT(xx)`

Front Panel Key: Level | Attenuator | Auto Attenuation

`ATT(xx)`

Description: Sets the step attenuator value to xx (times 10 dB).

Permits setting the step attenuator value to xx (times 10 dB) in the ALC step attenuator decoupling mode.

Functional Category: POWER LEVELING

Value: xx: unsigned integer between 00 (0 dB attenuation) and 11 (110 dB attenuation)

Example: `AT1, ATT08`

Decouples the step attenuator from the ALC system and sets the step attenuator value to 80 dB.

Related Commands: `AT0`, `AT1`

Front Panel Key: Level | Attenuator | Auto Attenuation to On | Attenuation

`CF0 or CF0 <value> <unit>`

Description: Sets CW mode at F0 and opens the F0 parameter.

Places the MG362x1A in CW mode at the preset (or previously set) F0 frequency and causes F0 to be output. Also permits setting the F0 parameter. The frequency setting is determined by the frequency range of the particular MG362x1A model.

Functional Category: CW FREQUENCY

Value: Frequency range of the MG362x1A model

Units: GH, MH, KH, HZ

Example: `CF0 5.5 GH`

Places the MG362x1A at the preset (or previously set) CW F0 frequency and opens the F0 parameter for data entry. Sets the F0 frequency to 5.5 GHz.

Output Command: `OF0`

Related Commands: `ACW`

Front Panel Key: Frequency | F(x)

If in swept mode:

Frequency | ENABLE CW MODE | F(x)

CF1 <value> <unit>

Description: Sets CW mode at F1 and opens the F1 parameter.

Places the MG362x1A in CW mode at the preset (or previously set) F1 frequency and causes F1 to be output. Also permits setting the F1 parameter. The frequency setting is determined by the frequency range of the particular MG362x1A model.

Functional Category: CW FREQUENCY

Value: Frequency range of the MG362x1A model

Units: GH, MH, KH, HZ

Output Command: `OF1`

Related Commands: `CF1`, `ACW`

Front Panel Key: Frequency | F1

CLR

Description: Clears the data entry.

Functional Category: DATA ENTRY

Front Panel Key: Clear Entry Key

CSB

Description: Clears all Native status bytes.

Functional Category: STATUS

Front Panel Key: N/A

D1 <value> <unit>

Description: Opens the internal pulse delay1 parameter.

Functional Category: PARAMETER ENTRY

Value: See MG362x1A TDS (11410-00928) for clock rates.

Units: SEC, MS, US, NS

Example: `PTG3; D1 100 US; IP`

Turns on internal pulse trigger delayed by 100 µs.

Output Command: `OD1` (µs)

Related Commands: `IP`, `PMD(x)`, `PTG(x)`

Front Panel Key: Modulation | Pulse (internal) | Arrow Key | Wdth/Dly List. . . | Edit Selected

DS0

Description: Turns on the secure mode.

In the secure mode, the front panel display of all frequency, power level, and modulation parameters is disabled. The instrument continues to function normally in all other respects.

Functional Category: MISCELLANEOUS

Front Panel Key: **System** | Secure Mode

DS1

Description: Turns off the secure mode.

Restores the front panel display of all frequency, power level, and modulation parameters.

Functional Category: MISCELLANEOUS

Front Panel Key: **System** | Reset

EL0

Description: Inhibits updating of the ESB2 bit 4.

Blocks updating of the Extended Status Byte two bit four (RF Unlocked). This is the default setting because it is normal for the RF to be momentarily unlocked during sweeps and sweep retrace.

Functional Category: STATUS

Related Commands: `SB0`, `SQ0`

Front Panel Key: N/A

EL1

Description: Enables RF Unlocked bit update. ESB2 bit 4.

Unblocks updating of the Extended Status Byte bit four (RF Unlocked).

Functional Category: STATUS

Example: C code snippet to enable unlocked SRQ:

```
strcpy(buf, “csb sb1 mb2 sq1 el1");
buf[11] = 16; /*mb2 command mask value*/snd(source_addr, buf);
```

Output Command: `OES` (outputs value of the three status bytes)

Related Commands: `SQ1`

Front Panel Key: `N/A`

F1 <value> <unit>

Description: Sets output frequency to F1. Opens the F1 parameter.

Permits setting the F1 parameter. The frequency setting is determined by the frequency range of the particular MG362x1A model.

Functional Category: PARAMETER ENTRY

Value: Frequency range of the MG362x1A model

Units: GH, MH, KH, HZ

Front Panel Key: Frequency | F1

Output Command: `F1` returns the F1 frequency value (in MHz) to the controller.

FB0

Description: Inhibits extended status byte 1 SRQ generation.

Inhibits an SRQ from being generated when the Extended Status Byte one bit is set. This is the default mode.

Functional Category: STATUS

Related Commands: `SQ0`, `FB1`

Front Panel Key: N/A

FB1

Description: Enables extended status byte 1 SRQ generation.

Enables an SRQ to be generated when Primary Status Byte bit zero (Extended Status Byte one) is set and SQ1 has been programmed. The Extended Status Byte one bit is set whenever one (or more) of the unmasked status reporting functions in Extended Status Byte one is true. SQ1 enables the SRQ generation function.

Functional Category: STATUS

Related Commands: `SQ1`, `MB1`, `FB0`

Example: C code snippet to enable RF locked SRQ:

```
strcpy(buf, “csb fb1 mb1 sq1 ls1");
buf[11] = 8; /*mb1 command mask value*/snd(source_addr, buf);
```

Output Command: `OSB` (output primary status byte)

Front Panel Key: N/A

IP

Description: Turns on internal pulse modulation.

Disables the external pulse modulation function, if previously programmed.

Functional Category: MODULATION

Related Commands: `P0`

Front Panel Key: Modulation | Pulse (internal) | State On

L0 or L0 <value> <unit>

Description: Sets RF output power level to L0. Opens L0 parameter.

Causes the RF output power level to be set to the preset (or previously set) L0 power level. Also permits setting the L0 parameter. The power level setting is determined by the power level range of the particular MG362x1A model.

Functional Category: POWER LEVELING

Value: Power level range of the MG362x1A model

Units: DM (log), VT (linear)

Example: `L0 5 DM`

Places the RF output power level at the preset (or previously set) L0 power level and opens the L0 parameter for data entry. Sets the L0 power level to 5 dBm.

Output Command: `OL0` returns the L0 power level (in dBm when in log mode; in mV when in linear mode) to the controller.

Related Commands: `LOG`, `LIN`, `PU0`, `PU1`, `PU2`

Front Panel Key: Level | Level | L0

L1 <value> <unit>

Description: Sets RF output power level to L1. Opens L1 parameter.

Causes the RF output power level to be set to the preset (or previously set) L1 power level. Also permits setting the L1 parameter. The power level setting is determined by the power level range of the particular MG362x1A model.

Functional Category: POWER LEVELING

Value: Power level range of the MG362x1A model

Units: DM (log), VT (linear)

Output Command: `OL1` returns the L1 power level (in dBm when in log mode; in mV when in linear mode) to the controller.

Related Commands: `L1`

Front Panel Key: Level | L1

LE0

Description: Inhibits lock error SRQ generation.

Inhibits an SRQ from being generated when the Lock Error bit (Primary Status Byte bit three) is set. This is the default mode.

Functional Category: STATUS

Front Panel Key: N/A

Functional Category: STATUS

LE1

Description: Enables lock error SRQ generation. Enables Lock Error bit update. Primary byte bit 3.

Enables an SRQ to be generated when Primary Status Byte bit three (Lock Error) is set and SQ1 has been programmed.

Example: `CSB, SQ1`

Enable an SRQ on lock error.

Front Panel Key: N/A

LO0

Description: Turns off the level offset function.

Functional Category: POWER LEVELING

Related Commands: `LO1`, `LOS`

Front Panel Key: Level | Offset | Offset to Off

LO1

Description: Turns on the level offset function. Can apply a constant offset to the displayed level value.

Functional Category: POWER LEVELING

Related Commands: `LO0`, `LOS`

Output Command: `LO1 returns the Level Offset value (in dB when in Log mode; in mV when in Linear mode) to the controller.`

Front Panel Key: Level | Offset | Offset to On

LOS <value> <unit>

Description: Opens the level offset parameter.

Functional Category: PARAMETER ENTRY

Value: –100dB to 100dB (logarithmic); +xxx mV to xxx mV (linear)

Units: DB (log), VT (linear)

Output Command: `LO1 returns the Level Offset value (in dB when in Log mode; in mV when in Linear mode) to the controller.`

Related Commands: `LO0`, `LO1`

Front Panel Key: Level | Offset | Level Offset

LS0

Description: Inhibits updating of the ESB1 bit 3.

Blocks updating of the Extended Status Byte one bit three (RF Locked). This is the default setting.

Functional Category: STATUS

Front Panel Key: N/A

LS1

Description: Enables updating of the ESB1 bit 3.

Unblocks updating of the Extended Status Byte one bit three (RF Locked).

Functional Category: STATUS

Example: C code snippet to enable the RF locked SRQ:

```
strcpy(buf, “csb fb1 mb1 sq1 ls1");
buf[11] = 8; /*mb1 command mask value*/snd(source_addr, buf);
```

Related Commands: `SQ1`, `MB1`, `LS0`

Front Panel Key: N/A

MB0

Description: Sets the enable mask byte for the primary status byte.

Sets an eight-bit data mask that is used to enable specific bits of the primary status byte. This enables any or all of the bits (except for bit six) in the primary status byte to generate an SRQ using one eight-bit byte. This command can be equivalent to sending FB1, LE1, PE1, SE1, SB1, and UL1.

Functional Category: STATUS

Front Panel Key: N/A

MB1

Description: Sets the enable mask byte for extended status byte 1.

Sets an eight-bit data mask that is used to enable specific bits of the extended status byte one. This enables any or all of the bits in the extended status byte one to generate an SRQ using one eight-bit byte.

Functional Category: STATUS

Example: C code snippet to enable RF locked bit:

```
strcpy(buf, “mb1 ”); buf[3] = 8; snd(source_addr, buf);
```

Front Panel Key: N/A

MB2

Description: Sets the enable mask byte for extended status byte 2.

Sets an eight-bit data mask that is used to enable specific bits of the extended status byte two. This enables any or all of the bits in the extended status byte two to generate an SRQ using one eight-bit byte.

Functional Category: STATUS

Example: C code snippet to enable the RF unlocked bit:
strcpy(buf, “mb2 ”); buf[3] = 16; snd(source_addr, buf);

Front Panel Key: N/A

OEM

Description: Returns the Extended SRQ Mask bytes (3 binary bytes) to the controller

Functional Category: OUTPUT

Related Commands: MB0, MB1, MB2

Front Panel Key: N/A

OES

Description: Returns the GPIB Status bytes (3 binary bytes) to the controller.
Returns the contents of the primary status byte and the two extended status bytes to the controller.

Functional Category: OUTPUT

Front Panel Key: N/A

OPP

Description: Returns the internal pulse period value (in ms) to the controller.
Returns the contents of the primary status byte and the two extended status bytes to the controller.

Functional Category: OUTPUT

Front Panel Key: N/A

OSB

Description: Returns the primary GPIB status byte to the controller.

Functional Category: OUTPUT

Front Panel Key: N/A

OSR

Description: Returns the self-test results (6 binary bytes) to the controller. Can apply a constant offset to the displayed level value.

Functional Category: OUTPUT

Related Commands: TST

Front Panel Key: N/A

P0

Description: Turns off the internal or external pulse modulation function (This is the default mode).

Functional Category: MODULATION

Related Commands: `XP`

Front Panel Key: Modulation | Pulse | On/Off to Off

P3

Description: Valid for the MG369xC product series and made available to avoid a syntax error when programming the MG362x1A product.

P4

Description: Valid for the MG369xC product series and made available to avoid a syntax error when programming the MG362x1A product.

PC4

Description: Selects the 100 MHz internal pulse clock rate.

Functional Category: MODULATION

Front Panel Key: Modulation | Pulse (internal) | Clock 100

PDY <value> <unit>

Description: Opens the internal pulse delay1 parameter (Same as D1).

Functional Category: PARAMETER ENTRY

Value: See MG362x1A TDS (11410-00928) for clock rates.

Units: SEC, MS, US, NS

Front Panel Key: Modulation | Pulse (internal) | Wdth/Dly List. . . | Edit Selected

PE0

Description: Inhibits parameter range error SRQ generation.

Inhibits an SRQ from being generated when the primary status byte bit four (parameter range error) is set. This is the default mode.

Functional Category: STATUS

Front Panel Key: N/A

PE1

Functional Category: STATUS

PE1

Description: Enables parameter range error SRQ generation.

Enables an SRQ to be generated when primary status byte bit four (parameter range error) is set and SQ1 has been programmed. SQ1 enables the SRQ generation function.

Related Commands: `SQ1`

Example: `CSB; SQ1; PE1`

Enables an SRQ on a parameter range error.

Front Panel Key: N/A

PER <value> <unit>

Description: Opens the internal pulse period parameter.

Functional Category: PARAMETER ENTRY

Value: See MG362x1A TDS (11410-00928) for clock rates.

Units: SEC, MS, US, NS

Example: `PER 500 US; W1 50 US; IP`

Output Command: `OPP` (in µs)

Turns internal pulse on with a period of 500 µs and pulse width of 50 µs.

Related Commands: `PR`

Front Panel Key: Modulation | Pulse (internal) | Edit Period

PMD(x)

Description: Selects the internal pulse mode.

Disables the internal ΦM function or the external ΦM function in Narrow mode, had either of these modes been previously programmed. If the ΦM function is not installed, this command produces a syntax error.

Functional Category: MODULATION

Value: x: 1 = Single, 2 = Doublet, 3 = Triplet, 4 = Quadruplet

Example: `PMD2; W1 10 US; IP`

Turns internal pulse on and sets pulse doublet, 10 µs width, 15 µs delay (from start of first pulse), and 10 µs width.

Output Command: `OMD` (text)

Front Panel Key: Modulation | Pulse (internal) | More > | Mode. . . | Select

PTG(x)

Description: Selects the internal pulse trigger.

Functional Category: MODULATION

Value: x: 1 = Free Run, 2 = Gated, 3 = Delayed, 4 = Triggered, 5 = Triggered with Delay

Front Panel Key: N/A

PW <value> <unit>

Description: Opens the internal pulse width1 parameter (Same as W1).

Functional Category: PARAMETER ENTRY

Value: See MG362x1A TDS (11410-00928) for clock rates.

Units: SEC, MS, US, NS

Related Commands: W1

Front Panel Key: Modulation | Pulse (internal) | Wdth/Dly List. . . | Edit Selected

Example: `PMD1; PTG1; PW 50 US; IP`

Turns internal pulse on with a single pulse, free run of width 50 µs.

Output Command: `OPW` (in µs)

RF0

Description: Turns off the RF output.

Functional Category: POWER LEVELING

Front Panel Key: Output - LED is Off

RF1

Description: Turns on the RF output. This is the default mode.

Functional Category: POWER LEVELING

Front Panel Key: Output - On (LED is green)

RL

Description: Returns the instrument to local (front panel) control.

Functional Category: MISCELLANEOUS

Front Panel Key: **System** | Return to Local
System | Reset

RO0

Description: Selects RF to be on at reset.

This is the default mode.

Functional Category: CONFIGURATION

Front Panel Key: **System** | Config | RF | Reset State to On

RO1

Description: Selects RF to be off at reset.

Functional Category: CONFIGURATION

Front Panel Key: **System** | Config | RF | Reset State to Off

RST

Description: Resets the MG362x1A to its default settings.

Sending this command clears the current instrument setup. If this setup is needed for future testing, save it as a stored setup using the command SSN(M1-9) or SAF before sending RST.

Remote settings are not reset, e.g. Remote language, Ethernet settings, GPIB settings are not reset.

Functional Category: MISCELLANEOUS

Front Panel Key: System | Reset | System Reset - Application Reset

SB0

Description: Inhibits extended status byte 2 SRQ generation.

Inhibits an SRQ from being generated when the Extended Status Byte two bit is set. This is the default mode.

Functional Category: STATUS

Front Panel Key: N/A

SB1

Description: Enables extended status byte 2 SRQ generation.

Enables an SRQ to be generated when primary status byte bit seven (extended status byte two) is set and SQ1 has been programmed. The extended status byte two bit is set whenever one (or more) of the unmasked status reporting functions in extended status byte two is true.

Functional Category: STATUS

Example: C code snippet to enable unlocked SRQ:

```
strcpy(buf, “csb sb1 mb2 sq1 el1");
buf[11] = 16; /*mb2 command mask value*/snd(source_addr, buf);
```

Related Commands: SQ1, MB2

Front Panel Key: N/A

SQ0

Description: Disables the SRQ generation function. This is the default mode.

Functional Category: STATUS

Front Panel Key: N/A

SQ1

Description: Enables the SRQ generation function. Enables SRQ generation when a Primary byte bit is set.

This command allow a status reporting function, that is true and enabled, to pull the SRQ line LOW (true) and request service from the controller.

Functional Category: STATUS

Front Panel Key: N/A

SYST:LANG "SCPI”

Description: Selects the instrument's external interface language. The double quotes are required and will be returned with the query reply.

"NATIVE" changes the instrument's external interface language to the Anritsu Native language mode. Any commands issued within 1 second of the change may be garbled or lost.

Note: When changing from NATIVE to SCPI interface language, use the command SYST:LANG "SCPI". Do not use the long form of the command, do not use a leading colon (:), and use exactly one space ( ) between the 'G' and the opening double quote ("). Deviations from this exact command may result in a syntax error.

Related Commands: SYST:LANG

Default Value: SCPI

TST

Description: Starts an instrument self-test.

When a `TST` command is received, the signal generator performs a self test, then returns a single byte, unterminated. A "P" (for pass) or an "F" (for fail). Extended Status Byte one bit zero is set if self test fails; bit two is set when self test is complete.

Functional Category: SELF TEST

Output Command: `OSR` returns the six self test results bytes to the controller.

Front Panel Key: System | Diagnostics | Self-Test

UL0

Description: Inhibits RF unleveled SRQ generation.

Inhibits an SRQ from being generated when the RF Unleveled bit is set. This is the default mode.

Functional Category: STATUS

Front Panel Key: N/A

UL1

Description: Enables RF unleveled SRQ generation. Enables RF Unleveled bit update.
Primary byte bit 2.

Enables an SRQ to be generated when Primary Status Byte bit two (RF Unleveled) is set and SQ1 has been programmed.

Functional Category: STATUS

Example: `SQ1; UL1`

Enables an unleveled SRQ generation.

Front Panel Key: N/A

W1 <value> <unit>

Description: Opens the internal pulse width 1 parameter.

Functional Category: PARAMETER ENTRY

Value: See MG362x1A TDS (11410-00928) for clock rates.

Units: SEC, MS, US, NS

Example: `W1 50 US; PER 100 US; PMD1; PTG1; IP`

Turns internal pulse on with a 50 µs pulse every 100 µs.

Output Command: `OW1` (in µs)

Front Panel Key: Modulation | Pulse (internal) | Wdth/Dly List. . . | Edit Selected

XP

Description: Turns on the external pulse modulation function. Disables the internal pulse modulation function, if previously programmed.

Turns on the pulse modulation function. If pulse modulation is not installed, this command produces a syntax error.

Functional Category: MODULATION

Related Commands: `PO`

Front Panel Key: Modulation | Pulse | On/Off to On

Chapter 4 — SCPI Commands

4-1 Introduction

This chapter contains information on all SCPI programming commands accepted and implemented by the Series MG362x1A Synthesized Signal Generator.

4-2 Parameter Ranges

The tables below list the MG362x1A parameters setting limits.

Table 4-1. Model MG362x1A Minimum and Maximum Frequencies

Model	Minimum Frequency*	Maximum Frequency
MG36221A	9 kHz	20 GHz
MG36241A	9 kHz	43.5 GHz
MG36271A	9 kHz	70 GHz

Table 4-2. Model MG362x1A Minimum and Maximum Settable Power Levels

	Without Step Attenuator		With Step Attenuator (Option 2)	
Model	Minimum	Maximum	Minimum	Maximum
MG36221A	–20 dBm	+33 dBm	–130 dBm	+30 dBm
MG36241A	–20 dBm	+33 dBm	–130 dBm	+30 dBm
MG36271A	–20 dBm	+33 dBm	–130 dBm	+30 dBm
With Option 15 (High Power)				
MG36221A	–20 dBm	+33 dBm	–130 dBm	+33 dBm
MG36241A	–20 dBm	+33 dBm	–130 dBm	+33 dBm
MG36271A	–20 dBm	+33 dBm	–130 dBm	+30 dBm

Chapter 4 — MG362x1A Commands

4-3 Common Commands

*CLS

Title: Clear Status Command

Description: This command clears all status data structures in the device (registers and error queue) and forces the Operation Complete state to Idle.

*ESE <numeric_value>
*ESE?

Title: Standard Event Status Enable

Description: This command provides access the Standard Event Status Enable Register.

Refer to the Status System Reporting section within this manual or IEEE 488.2, section 11.5.1 for more information about each supported bit.

Value ranges from 0 to 255.

Query returns a NR1 Numeric response.

Refer to *ESR? for the list of supported bit values.

Parameter(s): <numeric_value>

*ESR?

Title: Standard Event Status

Description: This command queries the Standard Event Status Register.

Refer to the Status System Reporting section within this manual or IEEE 488.2, section 11.5.1 for more information about each supported bit.

Query returns a NR1 Numeric response.

Supported bits:

bit 0: Operation Complete (OP)

bit 1: Not Used

bit 2: Query Error

bit 3: Device Dependent Error

bit 4: Execution Error

bit 5: Command Error

bit 6: Not Used

bit 7: Power On

*IDN?

Title: Identify

Description: This command returns the following information in <string> format separated by commas: manufacturer name ("Anritsu"), model number, serial number, and firmware package number.

*OPC
*OPC?

Title: Operation Complete

Description: The *OPC command causes the device to set the OPC bit of the Standard Event Status Register on the next transition of the No Operation Pending flag from false to true.

The *OPC? command returns 1 in the response whenever the No Operation Pending Flag is true, blocking until the operation is complete.

Note that for IQ streaming, *OPC? will return 1 immediately so that ABORt is not blocked.

*OPT?

Title: Option Identification

Description: The *OPT command queries the device to identify reportable device options. The response is a list of numeric options separated by commas. If no options are available, a value of "0" will be reported

*RCL <numeric_value>

Title: Recall Setup

Description: Recalls the saved setup from the desired slot, 1-9. If no setup exists at that slot, will produce an error.

Recalling from slot 0 will recall the instrument's last auto-saved setup.

Parameter(s): <numeric_value>

*RST

Title: Reset

Description: This command sets the instrument into a known state that is ready for SCPI command processing. Setting the instrument into a known state includes the following steps:

1. All applications beside the default application of the particular model will be turned off.

2. Settings for all applications (regardless of an application being active) are preset to default values.

3. System settings excluding settings associated with client connections (like Ethernet addresses or DNS) are preset to default values.

4. Pending operations are aborted and the pending operation flag will be reset. The instrument will be forced into the operation complete command idle state (OCIS) and the operation complete query idle state (OQIS) before the pending operation flag is reset.

*SAV <numeric_value>

Title: Save Setup

Description: Saves the current instrument configuration to internal storage. Differs from :MMEMory:STORe:STATe in that the setup file is not stored in a user-accessible location.

The command accepts a number 0-9, where numbers 1-9 will save a setup to that numbered slot, and *RCL <n> will recall the setup from that slot.

Entering 0 will instead manually trigger the instrument's auto-save functionality. The instrument saves settings every 5 minutes and when shut down safely, but this may be used for situations where you wish to have a guaranteed save.

The saved setups in slots 1-9 are deleted during a User File Reset and Reset All.

Parameter(s): <numeric_value>

*SRE <numeric_value>
*SRE?

Title: Service Request Enable

Description: This command provides access to the Service Request Enable Register.

Refer to the Status System Reporting section within this manual or IEEE 488.2, section 11.5.1 for more information about each supported bit.

Refer to *STB? for the list of supported mask bit values.

Bit 6 (MSS) cannot be set as per IEEE 488.2 standard.

Query returns a NR1 Numeric response.

Parameter(s): <numeric_value>

*STB?

Title: Status Byte Query

Description: This command queries the Status Byte Register.

Refer to the Status System Reporting section within this manual or IEEE 488.2, section 11.5.1 for more information about each supported bit.

Query returns a NR1 Numeric response.

Supported bits:

bit 0: Not Used

bit 1: Not Used

bit 2: Error Queue (ERRQ)

bit 3: Not Used

bit 4: Message Available (MAV)

bit 5: Standard Event Status (ESB)

bit 6: Master Summary Status (MSS)

bit 7: Not Used

*TRG

Title: Trigger

Description: Triggers instrument if :TRIGger:SOURce command data parameter is BUS. Refer to INITiate and TRIGger subsystem commands. It performs the same function as the Group Execute Trigger <GET> command defined in IEEE 488.1.

*TST?

Title: Self Test

Description: Causes the unit to perform a full internal self-test.

Status messages that indicate self-test results are placed in the error queue in the order they occur. Bits in the status register are also affected.

Returns the number of errors placed in the error queue. 0 means the unit passed self-test.

*WAI

Title: Wait-to-Continue Command

Description: This command causes the device to execute no further commands or queries until the No Operation Pending flag is TRUE.

4-4

:DIAGnostic:MEMory?

Title: Used Memory

Description: This command returns the total memory available on the instrument, the free memory, and the memory used by the backend process in kB.

If the instrument runs into an error while retrieving system memory data, "nan,nan,nan" will be returned as the response and a "-100" (SCPI command error) will be placed in the SCPI error queue.

:FETCh:GPS?

Title: Basic GPS Information

Description: This command returns the timestamp, latitude, and longitude of the device. The response is a comma-delimited ASCII response of one of the following forms: NO FIX or GOOD FIX,<timestamp>,<latitude>,<longitude>

If no GPS fix is currently available, the first response form (NO FIX) is returned.

If the GPS does have a fix, the second response form (GOOD FIX) is returned.

<timestamp> is in ISO8601 format. The timestamp provides the 24-hour time, and will include the year/date and/or UTC offset if the hardware supports it. If no UTC offset is provided, the time is in UTC time.

<latitude> and <longitude> are specified in decimal degrees.

:FETCh:GPS:ALL?

Title: Full GPS Information + Satellite System Information

Description: This command returns the fix timestamp, latitude, longitude, altitude and information on the satellites used for the last fix and the GNSS systems they are from. The response is in JSON format in the one of the following forms: {"fix":"GPS OFF"}

or {"fix":"GOOD FIX","timestamp":<timestamp>,"latitude":<latitude>,"longitude":<longitude>,"altitude":<altitude>, "satellitesUsed":<satellitesUsed>,"satellites":[{"name":"GPS","count":x}, {"name":"Galileo","count":x}, {"name":"GLONASS","count":x}, {"name":"BeiDou","count":x}]} {"name":"SBAS","count":x}]} <timestamp> is in ISO8601 format. The timestamp provides the 24-hour time, and will include the year/date and/or UTC offset if the hardware supports it. If no UTC offset is provided, the time is in UTC time.

<latitude> and <longitude> are specified in decimal degrees.

<altitude> specifies the current altitude relative to mean sea level, in meters\.

All satellite systems will be present with a count of 0 if they are not used in the fix.

:FETCh:GPS:FULL?

Title: Full GPS Information

Description: This command returns the timestamp, latitude, longitude, altitude, and satellite count of the device. The response is a comma-delimited ASCII response of one of the following forms: NO FIX or GOOD FIX,<timestamp>,<latitude>,<longitude>,<altitude>,<satellites>

If no GPS fix is currently available, the first response form (NO FIX) is returned.

If the GPS does have a fix, the second response form (GOOD FIX) is returned.

<timestamp> is in ISO8601 format. The timestamp provides the 24-hour time, and will include the year/date and/or UTC offset if the hardware supports it. If no UTC offset is provided, the time is in UTC time.

<latitude> and <longitude> are specified in decimal degrees.

<altitude> specifies the current altitude relative to mean sea level, in meters.

<satellites> specifies an integer count of the number of satellites currently used in the fix.

:INSTrument:APPLication:STATe <SG>,<0 | 1 | ON | OFF>

Title: Activate or Deactivate Application

Description: This command turns the specified application ON or OFF one at a time. When an application is turned ON (i.e. activated), it is loaded into memory and can be used to make measurements. An application must be activated before it will respond to commands. Please expect some delay in the application response to a command right after being turned on. The delay will vary depending on the application type and the options being installed. When an application is turned OFF (i.e. deactivated), any measurements it is making are stopped and it is unloaded from memory. Attempting to turn ON the same application twice or turn OFF an application that is not currently ON is not permitted and will result in an execution error (-200).

Attempting to turn on a different application (not the currently active application) requires turning the current application OFF. Otherwise, it will result in an execution error. To turn off an application uses the :INSTrument:APPLication:STATe command.

Unlike INSTrument:SELect, which selects a given application and makes it active if it is not already active, this command can be used to activate an application without selecting it.

If the requested application is the selected application (which can be queried with INSTrument:SELect?), turning it OFF will cause the current application selection to be reported as NONE via INSTrument:SELect?.

This command takes two parameters. The first parameter is a string value that specifies which application should be turned ON or OFF. The permitted values of this parameter are the same as those used by the INSTrument:SELect command. The second parameter is a Boolean value that specifies whether the application should be turned ON or OFF.

After a reset, the default selected application is the only active application.

Parameter(s): <SG>,<0 | 1 | ON | OFF>

:INSTrument:CATalog:ACTive?

Title: List Active Applications

Description: This command queries all active applications. The response is returned as a comma delimited list of application names. The application names are the same as those returned by the INSTrument:CATalog:FULL? query or used as parameters to the INSTrument:SELect command.

If no applications are currently active, this query will return NONE.

:INSTrument:CATalog:ALL?

Title: Query Available Applications with all information

Description: This query returns a JSON list of available applications with metadata including active status, appId modes and SCPI API version. [{"name":"SPA", "active":"true", "appId":"1","modes":["SPEC","NRAD"], "version": 1}, {"name":"VNA", "active":"false", "appId":"26","modes":["VNA"], "version": 1}, {"name":"SYSTEM", "version":1 }]

The list will always contain at least one item since the System application will always be included.

:INSTrument:CATalog:FULL?

Title: Query Available Applications

Description: This query returns a list of string - number pairs. The string contains the name of the available application. The immediately following NR1-formatted number is its associated application number. All response data elements are comma separated.

If no application is available, a null string followed by a zero is returned.

:INSTrument:NSELect <numeric_value>
:INSTrument:NSELect?

Title: Select Application by Number

Description: Sets the selected application based on the value of <integer>. The query version returns the number associated with the current application. The list of valid integers and the applications they correspond to can be queried with INSTrument:CATalog:FULL?

Selecting a different application (not the currently active application) requires turning the current application OFF. Otherwise, it will result in an execution error. To turn off an application uses the :INSTrument:APPLication:STATe command.

If no applications are active, the query version of this command will return 0.

Parameter(s): <numeric_value>

:INSTrument[:SELect] <SG>
:INSTrument[:SELect]?

Title: Select Application by Name

Description: Sets the selected application based on the application name specified by <identifier>. The <identifier> is a valid application name (character data). The list of valid application names can be queried with INSTrument:CATalog:FULL?

Selecting a different application (not the currently active application) requires turning the current application OFF. Otherwise, it will result in an execution error. To turn off an application uses the :INSTrument:APPLication:STATe command.

The query version returns the name of the current application.

If no applications are active, the query version of this command will return NONE.

Parameter(s): <SG>

Query Return: NONE | SPA | VNA | SG | HIPM | CAAUSB | CWSG | VVM | CAA

Default Value: NONE

:MMEMory:CATalog:DIRectory? <string>,<string>

Title: Memory Catalog Directory

Description: Returns the non-recursive contents of the memory specified by the MSUS and DIRECTORY parameters. Both parameters are case sensitive.

Parameter 1: Directory. Use "/" as a directory separator. Note that this parameter should NOT be an empty string. To access the root directory, pass in "/" as the first parameter.

Parameter 2: MSUS (Use :MMEMory:CATalog:MSUSs query command to retrieve the list of available storage devices).

For example:

MMEMory:CATalog:DIRectory? "/", "USB1-1"

The response is formatted as follows: <total space used on MSUS>,<total space available on MSUS>,{<file entry1>},..{<file entryN>} where <file entry> is: <file or directory name>,<file type>,<file size>

Parameter(s): <string>,<string>

:MMEMory:CATalog:DIRectory:SIZE? <string>,<string>

Title: MMemory Catalog Directory Size

Description: This command is query-only and returns the size of a given directory. It has two parameters. Both are case sensitive.

<directory>,<msus>

Parameter 1: The full path of a directory to query the size. Use "/" as a directory separator. Note that this parameter should NOT be an empty string. To access the root directory, pass in "/" as the first parameter.

Parameter 2: MSUS (Use :MMEMory:CATalog:MSUSs query command to retrieve the list of available storage devices).

For example:

MMEMory:CATalog:DIRectory:SIZE? "/", "USB1-1"

The response is a single ASCII string of the size of the given directory.

Parameter(s): <string>,<string>

:MMEMory:CATalog:MSUSs?

Title: Memory Catalog Msuss

Description: Returns a list of all available mass storage devices present, formatted as follows: <Device Name 1>,<Device Name 2>,...<Device Name N>

For example, sending:

MMEMory:CATalog:MSUSs?

could return:

Internal,USB1-1

:MMEMory:CDIRectory <string>
:MMEMory:CDIRectory?

Title: Default Mass Storage Directory

Description: This command selects the default directory on the default mass storage device (see :MMEMory:MSIS) for use with MMEMory commands. The parameter is case sensitive. Use "/" as a directory separator.

For example:

MMEMory:CDIRectory "/example_folder"

The set command will fail and an error is reported if the requested directory does not exist. Note that the query command returns the name of the default directory that was previously set and the device does not check whether the directory is still present.

Parameter(s): <string>

Default Value: /

:MMEMory:COPY <string>,<string>,<string>,<string>

Title: Copy File

Description: This command copies the specified source file into a new file located at the specified destination location. Four parameters are required for this command:

1. Source File Location: The file path to the file to be copied.

2. Source MSUS: The mass storage device the file is located in (i.e. Internal).

3. Destination File Location: The destination file path that the file should be copied to.

4. Destination MSUS: The mass storage device that the destination file should be written to (i.e. Internal).

For example:

MMEMory:COPY "/example_file.ext", "Internal", "/example_file.ext", "USB1-1"

Please note that if a file already exists at the destination location or if the source file does not exist, the command will fail to execute and add an execution error into the SCPI error queue. This command will only copy files. If a directory path is passed in as a parameter, the command will fail to execute.

Parameter(s): <string>,<string>,<string>,<string>

:MMEMory:CREate:DIRectory <string>,<string>

Title: Create Directory

Description: Creates a directory at the specified mass storage device. Both parameters are case sensitive.

Parameter 1: Directory. Use "/" as a directory separator.

Parameter 2: MSUS (Use :MMEMory:CATalog:MSUSs query command to retrieve the list of available storage devices)

For example:

MMEMory:CREate:DIRectory "/example_folder", "USB1-1"

If the intermediate directories in the path don't exist, the command will automatically create them. Common causes of the command failure are:

- The MSUS device is not present (Use :MMEMory:CATalog:MSUSs query command to retrieve the list of available storage devices)

- The total space available on MSUS device is not big enough (Use :MMEMory:CATalog:DIRectory query command to retrieve the total space available on MSUS)

Parameter(s): <string>,<string>

:MMEMory:DATA <string>,<string>,<block data>
:MMEMory:DATA? <string>,<string>

Title: File Transfer

Description: This command imports/exports a file to/from the instrument. Data is transferred to/from the instrument as an IEEE definite length arbitrary block response, which has the form <header><block>.

This set command takes three parameters:

1. File Path: The path of the file to be written

2. Msus: The mass storage device to write the file to

3. Block Data: The data to be written to the instrument in block data format

If any directories in the file path do not exist, the instrument will automatically create the required directories. Please note that the maximum file transfer size to the instrument is 25 MB.

The ASCII header specifies the number of data bytes of the file. It looks like #AX, where A is the number of digits in X and X is the number of bytes in the <block>. The first character is the literal ascii hash '#' 043. The second character (A) is a single ascii digit '1' to '9' describing the number of bytes in the length section (X). This number is called nlength. The next nlength bytes make up an ascii string of digits '1' to '9' describing the length of the <block> data.

For example, if the first 6 bytes are #49999, then the nlength is 4. The 4 bytes of length are 9999. After that follows the <block>, which would be 9999 bytes in size.

The query command takes two parameters.

1. File Path: The path to the file to be retrieved.

2. Msus: The mass storage device to retrieve the file from.

The file is returned in block data format with an ASCII header The query command will return a #10 if the file cannot be found.

Set Command Parameter(s): <string>,<string>,<block data>

Query Command Parameter(s): <string>,<string>

:MMEMory:DELete:DIRectory <string>,<string>

Title: Delete Directory

Description: Deletes the specified directory and all its content at the specified mass storage device. The two parameters for this command are directory name and mass storage device. Both parameters are case sensitive. Use "/" as a directory separator. To delete the root folder in a specified mass storage device, enter "/" for directory name parameter.

For example:

MMEMory:DELete:DIRectory "/example_folder", "USB1-1"

The available mass storage devices can be retrieved by using the :MMEMory:CATalog:MSUSs command. The command will fail if the directory doesn't exist or the mass storage device is not present.

Parameter(s): <string>,<string>

:MMEMory:DELete:FILe <string>,<string>

Title: Delete File

Description: Deletes the specified file at the specified mass storage device. The two parameters for this command are file name with relative path and mass storage device. Both parameters are case sensitive. Use "/" as a directory separator.

For example:

MMEMory:DELete:FILe "/example_folder/example_file.ext", "USB1-1"

The available mass storage devices can be retrieved by the :MMEMory:CATalog:MSUSs command. The command will fail if the file doesn't exist or the mass storage device is not present.

Parameter(s): <string>,<string>

:MMEMory:EJECt <string>

Title: Eject USB drive

Description: This command safely ejects (unmounts) a USB drive from the instrument. Following this command, a USB drive can be safely removed from the instrument. See :MMEMory:CATalog:MSUSs for more information on obtaining the name of the USB drive(s) currently mounted on the instrument.

For example:

MMEMory:EJECT "USB1-1"

Parameter(s): <string>

:MMEMory:LOAD:STATe <numeric_value>,<string>,<string>

Title: Load State

Description: This command recalls the specified setup. The file location is resolved using the MSUS and file path parameters. These parameters are case sensitive.

Parameters:

1. Numeric Value: currently unused. Send a 0.

2. File path: The file path (including file name) relative to the MSUS root directory. File extension is optional.

3. MSUS: Device to recall the file from. See MMEMory:CATalog:MSUSs? for information on obtaining the list of available devices.

For example:

MMEMory:LOAD:STATe 0,"/example_setup_file", "Internal"

The model of the device that the setup was created in must match the model of the device to recall in; otherwise, the recall will be rejected. The options of the device that the setup was created in must be enabled in the device to recall in; otherwise, the recall will be rejected.

Parameter(s): <numeric_value>,<string>,<string>

:MMEMory:MSIS <string>
:MMEMory:MSIS?

Title: Default Mass Storage Device

Description: This command selects the default device for use with MMEMory commands. The mass storage device parameter is case sensitive and must match a device returned from the :MMEMory:CATalog:MSUSs? command

For example:

MMEMory:MSIS "Internal"

Parameter(s): <string>

Default Value: Internal

:MMEMory:STORe:SCReen
:MMEMory:STORe:SCReen?

Title: Screen Shot

Description: The set command prompts a screenshot of the local UI to be taken and saved to local user storage. The naming convention and file path will be the same as what has been set up when taking a screenshot through the UI.

The query command will take a screenshot and return the image file as block data. No image will be saved in the local user storage.

The binary block header specifies the number of data bytes. It looks like #AX, where A is the number of digits in X and X is the number of bytes in the <block>.

The first character is the literal ascii hash '#' 043.

The second character (A) is a single ascii digit '1' to '9' describing the number of bytes in the length section (X). This number is called nlength.

The next nlength bytes make up an ascii string of digits '1' to '9' describing the length of the <block> data.

For example, if the first 6 bytes are #49999, then the nlength is 4. The 4 bytes of length are 9999. After that follows the <block>, which would be 9999 bytes in size.

:MMEMory:STORe:STATe <numeric_value>,<string>,<string>

Title: Store State

Description: This command saves the current setup to the specified file location. The file location is resolved using the MSUS and file path parameters. These parameters are case sensitive. If a file with the resolved name already exists, it will be overwritten.

Parameters:

1. Numeric Value: currently unused. Send a 0.

2. File path: The file path (including file name) relative to the MSUS root directory. File extension should not be specified.

3. MSUS: Device to save the file on. See MMEMory:CATalog:MSUSs? for information on obtaining the list of available devices.

For example:

MMEMory:STORe:STATe 0,"/example_setup_file", "Internal"

A common cause of the command failure is not having enough space available on MSUS to save the setup. Use :MMEMory:CATalog:DIRectory query command to retrieve the total space available on MSUS.

Parameter(s): <numeric_value>,<string>,<string>

:STATus:OPERation:CONDition?

Title: Status Operational Condition

Description: Returns the contents of the Operational Condition register in <NR1> format. When executed, this command does not clear the Operational Condition register.

:STATus:OPERation:ENABle <numeric_value>
:STATus:OPERation:ENABle?

Title: Status Operational Enable

Description: Sets the bits of the Operational Enable register associated with the Operational Event register to the binary weighted integer value specified by the mask parameter.

Parameter(s): <numeric_value>

:STATus:OPERation[:EVENt]?

Title: Status Operational Event

Description: Returns the contents of the Operational Event register in <NR1> format. When executed, this command clears the Operational Event register.

:STATus:OPERation:NTRansition <numeric_value>
:STATus:OPERation:NTRansition?

Title: Status Operational Negative Transition

Description: Sets the bits of the negative transition filter for the Operational Condition register to the binary weighted integer value specified by the mask parameter.

Parameter(s): <numeric_value>

:STATus:OPERation:PTRansition <numeric_value>
:STATus:OPERation:PTRansition?

Title: Status Operational Positive Transition

Description: Sets the bits of the positive transition filter for the Operational Condition register to the binary weighted integer value specified by the mask parameter.

Parameter(s): <numeric_value>

:STATus:PRESet

Title: Status Preset

Description: This command is an event that configures the SCPI and device-dependent status reporting structures so that device-dependent events are summarized and reported. This command performs the following functions:

Sets the Operational Enable register to all 0's.

Sets the Operational Positive Transition Filter to all 1's.

Sets the Operational Negative Transition Filter to all 0's.

Sets the Questionable Enable register to all 0's.

Sets the Questionable Positive Transition Filter to all 1's.

Sets the Questionable Negative Transition Filter to all 0's.

:STATus:QUEStionable:CONDition?

Title: Status Questionable Condition

Description: Returns the contents of the Questionable Condition register in <NR1> format. When executed, this command does not clear the Questionable Condition register.

:STATus:QUEStionable:ENABle <numeric_value>
:STATus:QUEStionable:ENABle?

Title: Status Questionable Enable

Description: Sets the bits of the Questionable Enable register associated with the Questionable Event register to the binary weighted integer value specified by the mask parameter.

Parameter(s): <numeric_value>

:STATus:QUEStionable[:EVENt]?

Title: Status Questionable Event

Description: Returns the contents of the Questionable Event register in <NR1> format. When executed, this command clears the Questionable Event register.

:STATus:QUEStionable:NTRansition <numeric_value>
:STATus:QUEStionable:NTRansition?

Title: Status Questionable Negative Transition

Description: Sets the bits of the negative transition filter for the Questionable Condition register to the binary weighted integer value specified by the mask parameter.

Parameter(s): <numeric_value>

:STATus:QUEStionable:PTRansition <numeric_value>
:STATus:QUEStionable:PTRansition?

Title: Status Questionable Positive Transition

Description: Sets the bits of the positive transition filter for the Questionable Condition register to the binary weighted integer value specified by the mask parameter.

Parameter(s): <numeric_value>

:SYSTem:COMMunicate:GPIB:ADDRess:PRIMary <numeric_value>
:SYSTem:COMMunicate:GPIB:ADDRess:PRIMary?

Title: Primary GPIB Address

Description: This command sets the Primary GPIB address of the instrument.

Parameter(s): <numeric_value>

Query Return: Numeric

Default Value: 5

Range: 0 to 30

:SYSTem:COMMunicate:GPIB:TERMinator:OUTPut <CRETurn|CRLFeed|LFEed>
:SYSTem:COMMunicate:GPIB:TERMinator:OUTPut?

Title: GPIB Output Terminator

Description: This command configures the line termination method for message data sent by the instrument. Selection is between line feed (LFeed), carriage return (CReturn), or carriage return and line feed (CRLFeed). Note that regardless of selection, response data is always terminated with EOI.

Parameter(s): <CRETurn|CRLFeed|LFEed>

Query Return: CRET|CRLF|LFE

Default Value: LFEed

:SYSTem:COMMunicate:LAN:CONFig <string>,[<string>],[<string>]
:SYSTem:COMMunicate:LAN:CONFig?

Title: Static LAN Configuration

Description: This command set and queries the static ethernet configuration of the device. The static configuration allows user to specify the ip, gateway, and subnet mask of the unit on a network. Parameters:

- Static IP Address: The desired IP address of the unit.

- Gateway: (optional)The network gateway

- Subnet Mask: (optional) The subnet mask of the network the device is connected to.

For a subnet mask of 255.255.255.0(/24), you only have a valid host IP range of x.x.x.1-x.x.x.254. The first IP in the subnet (x.x.x.0) is reserved for the network ID and the last one (x.x.x.255) is reserved as the broadcast ID. You can have a different range of valid host addresses if you change the subnet mask to a different number. For example, if a subnet mask of 255.255.254.0 is used, the gateway can be set to 192.168.101.0 since the valid host IP range is 192.168.100.1-192.168.101.254.

More information on subnetting can be found here:

https://docs.microsoft.com/en-us/troubleshoot/windows-client\networking/tcpip-addressing-and-subnetting

A calculator for finding valid host IP addresses in a subnet can be found here:

https://www.calculator.net/ip-subnet-calculator.html

The Gateway and Subnet Mask settings will remain no change if user has no input.

The dhcp will be turned off.

A new valid configuration will automatically be applied to the device. The user will be required to access the unit through the new configuration.

CAUTION: Consult with your network administrator when configuring the network interface to avoid potential loss of access or discovery of the device.

Parameter(s): <string>,[<string>],[<string>]

:SYSTem:COMMunicate:LAN:CONFig:CURRent?

Title: Current LAN Configuration

Description: This command queries the current ethernet configuration of the device.

:SYSTem:COMMunicate:LAN:DHCP <ON|OFF>
:SYSTem:COMMunicate:LAN:DHCP?

Title: DHCP Configuration

Description: This command sets and queries the DHCP configuration of the device. If the DHCP configuration is set to OFF, the device is configured to the static ethernet configuration (See :SYSTem:COMMunicate:LAN:CONFig). If the DHCP configuration is set to ON, the device will obtain its IP address, gateway, and subnet mask from the DHCP server in the network.

This set command should be used to caution, as changing the ethernet configuration will result in temporary loss of communication with the device.

CAUTION: Consult with your network administrator when configuring the network interface to avoid potential loss of access or discovery of the device.

Parameter(s): <ON|OFF>

Query Return: ON|OFF

Default Value: OFF

:SYSTem:COMMunicate:LAN:DNS <string>,<string>,<string>
:SYSTem:COMMunicate:LAN:DNS?

Title: Dynamic DNS Configuration

Description: This command sets and queries the Dynamic Domain Name Server (Dynamic DNS) configuration of the device. Currently, only http://www.noip.com (TM) is available for use. For support, please contact the vendor at http://www.noip.com. Parameters:

- Dynamic DNS NoIp.com Host Name: Desired host name of the device.

- Dynamic DNS NoIp.com Username: noip.com username.

- Dynamic DNS NoIp.com Password: noip.com password.

Parameter(s): <string>,<string>,<string>

:SYSTem:COMMunicate:LAN:DNSServer:APPend <string>, {<string>}, ...

Title: Domain Name System Server Append

Description: This command will add one or more Domain Name System Servers to the end of the device's current list of DNS Servers.

To view the entire list of DNS Servers, see SYSTem:COMMunicate:LAN:DNSServer:LIST. To remove a DNS Server, see SYSTem:COMMunicate:LAN:DNSServer:DELete.

This command takes a list of one or more string values. For example: SYSTem:COMMunicate:LAN:DNSServer:APPend "8.8.8.8", "123.125.81.6"

Parameter(s): <string>, {<string>}, ...

:SYSTem:COMMunicate:LAN:DNSServer:DELete <string>, {<string>}, ...

Title: Domain Name System Server Delete

Description: This command will delete the device's given Domain Name System Servers from the device's current list of DNS servers. To view the entire list of DNS Servers, see SYSTem:COMMunicate:LAN:DNSServer:LIST.

This command takes a list of one or more string values. For example: SYSTem:COMMunicate:LAN:DNSServer:DELete "8.8.8.8", "123.125.81.6"

Parameter(s): <string>, {<string>}, ...

:SYSTem:COMMunicate:LAN:DNSServer:LIST <string>, {<string>}, ...
:SYSTem:COMMunicate:LAN:DNSServer:LIST?

Title: Domain Name System Server List

Description: This command sets and retrieves a list of the device's currently set Domain Name System Servers.

The DNS Server is used to resolve a Domain Name. The list of DNS Servers will be contacted after any DNS Server assigned by DHCP Setting, if applicable, and will be contacted in the order given in the list.

This command takes a list of one or more string values. Setting the DNS server list with this command will override the currently set DNS server list.

An example of using the command: SYSTem:COMMunicate:LAN:DNSServer:LIST "8.8.8.8", "123.125.81.6"

This query version of this command returns a comma delimited list of strings. For example: 123.125.81.6,8.8.8.8

To remove DNS Server, see SYSTem:COMMunicate:LAN:DNSServer:DELete.

To add to the end of the current list, see SYSTem:COMMunicate:LAN:DNSServer:APPend.

Parameter(s): <string>, {<string>}, ...

:SYSTem:COMMunicate:LAN:FTP:STATe <0 | 1 | ON | OFF>
:SYSTem:COMMunicate:LAN:FTP:STATe?

Title: FTP Server State

Description: This command sets the system up to execute (ON) or bypass (OFF) the FTP server startup on the next instrument boot.

The query returns the state of the FTP server based on the bypass flag, 0 for bypass, 1 for start. The actual state of the FTP server is not check for the query.

Parameter(s): <0 | 1 | ON | OFF>

:SYSTem:COMMunicate:LAN:HOSTname <string>
:SYSTem:COMMunicate:LAN:HOSTname?

Title: Local Host Name

Description: This command sets and retrieves the local host name of the instrument. A valid hostname may contain only the ASCII letters 'a' through 'z' (in a case-insensitive manner), the digits '0' through '9', and the hyphen ('-'). They cannot start/end with '-'.

No other symbols, punctuation characters, or white space are permitted.

Parameter(s): <string>

:SYSTem:COMMunicate:LAN:HTTP:SECure:STATe <0 | 1 | ON | OFF> :SYSTem:COMMunicate:LAN:HTTP:SECure:STATe?

Title: HTTPS Port State

Description: This command sets the system up to allow (ON) or bypass (OFF) the HTTPS port setup on the next instrument boot.

The query returns the state of the HTTPS port based on the bypass flag, 0 for bypass, 1 for configure. The actual state of the HTTPS port is not check for the query.

Parameter(s): <0 | 1 | ON | OFF>

:SYSTem:DATE <numeric_value>,<numeric_value>,<numeric_value> :SYSTem:DATE?

Title: System Date

Description: This command sets and queries the system's internal calendar. Changing the system date does not affect the system time zone (if time zone is available). The three parameters for this command are <year>,<month> and <day>. The query response message shall consist of three fields separated by commas: <year>,<month>,<day>.

The year shall be entered as a four-digit number, including century and millennium information. This shall not be affected by a *RST command. Note: Some products only support the query version of this command.

Parameter(s): <numeric_value>,<numeric_value>,<numeric_value>

:SYSTem:DEFault:RESet:ALL

Title: Default Reset All

Description: This command performs a factory reset as well as deletes last saved settings, log files, and user files. The factory reset includes presetting parameters in all applications, presetting all system settings with the exception of ethernet settings (DHCP On/Off, static IP, Static Gateway, Static Subnet) and time zone settings, and cleaning up all previous access point connections if the instrument is equipped with WiFi.

:SYSTem:DEFault:RESet:DATA <USER|SYSTem>

Title: Reset System Files

Description: This command deletes the instrument data files as specified in the following table:

Parameter Value Description

USER Deletes all user files in the instrument's internal memory including measurements, setup files, and screen shots. User-customized system files will not be deleted.

SYSTem Deletes all user-customized system files including keyboard EZ names, cable lists, antenna lists, and log files.

Parameter(s): <USER|SYSTem>

:SYSTem:DEFault:RESet:FACTory

Title: Factory Default Reset

Description: This command presets parameters in all applications as well as system settings with the exception of ethernet settings (DHCP On/Off, static IP, Static Gateway, Static Subnet) and time zone setting to their factory default values, and restarts the instrument. Last saved settings will be deleted. If unit equips with WiFi, the previous access point connections will be clean up.

:SYSTem:ERRor:DISPlay <0 | 1 | ON | OFF>
:SYSTem:ERRor:DISPlay?

Title: SCPI Error Message State

Description: Sets or queries the state of SCPI error messages being displayed. By default, when a SCPI error enters the error queue it will be displayed on-screen. This command toggles whether those errors should be displayed. This command defaults to 0, meaning errors will NOT be displayed.

Usage: Suppress SCPI Error Messages: SYST:ERR:DISP 0 Show all SCPI Error Messages: SYST:ERR:DISP 1.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:SYSTem:ERRor[:NEXT]?

Title: System Error Queue

Description: If an error occurs, the error number and message are placed in the error queue, which can be read by this query command. Errors are cleared by reading them. Error code 0 is "No error".

Error codes from -100 to -199 belong to the Command error category and set bit 5 of the standard ESR register.

Error codes from -200 to -299 belong to the Command error category and set bit 4 of the standard ESR register.

Error codes from -400 to -499 belong to the Query error category and set bit 2 of the standard ESR register.

Error codes from -300 to -399 and 1 to 32767 belong to the Device-specific error category and set bit 3 of the standard ESR register.

Negative error numbers (command error, execution error, device-dependent error, query error) are standard SCPI errors. Positive error numbers are device specific errors, not standard SCPI errors.

The error queue is also cleared by *CLS, *RST, and when power is turned on.

If more errors have occurred than can fit in the buffer, the last error stored in the queue (the most recent error) is replaced with -350, Queue overflow. No additional errors are stored until removing errors from the queue. If no errors have occurred when reading the error queue, the instrument responds with 0, No error.

:SYSTem:FIRMware:UPDate <string>,<string>

Title: Firmware Update

Description: Initiate a firmware update from the MSUS whose path is specified as parameters. If the requested firmware package is not recognized, an appropriate error will be added to the error queue. Otherwise, the instrument will verify that the package is safe to install, update to the new firmware.

The two parameters for this command are file name with relative path and mass storage device. Both parameters are case sensitive. Use "/" as a directory separator.

For example:

SYSTem:FIRMware:UPDate "V2019.01.1.raucb", "USB4-1"

The available mass storage devices can be retrieved by the :MMEMory:CATalog:MSUSs command. The command will fail if the file doesn't exist or the mass storage device is not present.

Parameter(s): <string>,<string>

:SYSTem:FIRMware:UPDate:LIST?

Title: Firmware Package List

Description: This command queries the instrument for a list of firmware packages that can be installed from mounted USB block devices. Please note that firmware packages are only detected on the top level of a USB block device. Any files located in subfolders will be ignored.

The response is in JSON format and consists of the path of the package relative to the root of the MSUS, the MSUS that the package is located in, and the version of the package.

The following is an example of the response:
[{"msus":"USB1-1","path":"V2018.2.22.raucb","version":"V2018.2.22"},
{"msus":"USB2-1","path":"V2018.2.19.raucb","version":"V2018.2.19"}]

These package names enumerate the valid parameter values to the :SYSTem:FIRMware:UPDate command.

:SYSTem:FIRMware:UPDate:REMote <string>

Title: Remote Firmware Update

Description: Initiate a remote firmware update to the package whose name is specified as a parameter. Use :SYSTem:FIRMware:UPDate:REMote:LIST? to inspect the packages that can be installed. If the requested firmware package is not recognized, an appropriate error will be added to the error. Otherwise, the instrument will download the specified firmware package, verify that the package is safe to install, update to the new firmware.

The instrument will disconnect all remote clients and reboot after the new firmware has been installed.

If the <STRING DATA> argument begins with (case insensitive) any of

http://

https://

ftp://

Then the firmware update will be downloaded from that URL. The URL must be percent-encoded per RFC-2396 section 2.4.1.

Parameter(s): <string>

:SYSTem:FIRMware:UPDate:REMote:AUTO <0 | 1 | ON | OFF>
:SYSTem:FIRMware:UPDate:REMote:AUTO?

Title: Remote Firmware Update Auto Check

Description: This command sets and queries the state to enable automatically checking a remote server (see SYSTem:FIRMware:UPDate:REMote:SOURce?) for software updates. If this command is set to ON, The server will be checked for a newer version on bootup and every 24 hours thereafter. If this command is set to OFF, no automatic checking will take place, but a check can be initiated manually (see SYSTem:FIRMware:UPDate:REMote:LIST:REFResh command). If the check finds a newer version available, the SYSTem:FIRMware:UPDate:REMote:LATest? query will indicate the newer version. Use the SYSTem:FIRMware:UPDate:REMote command to initiate the download and install.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

:SYSTem:FIRMware:UPDate:REMote:LATest?

Title: Check For Firmware Update

Description: Check whether the instrument firmware is at the latest version. This command will access the package list specified by SYST:FIRM:UPD:REM:SOUR and return a response of the form <string>,<number> where <string> is the name of the latest firmware package, and <number> is 1 if the latest firmware package is newer than the version of firmware currently installed (a firmware update is available), or 0 if the current instrument firmware version matches the latest version (the instrument is up to date).

Clients can update the instrument to the latest firmware by passing the <string> component of the response as a parameter to the SYST:FIRM:UPD:REM command.

:SYSTem:FIRMware:UPDate:REMote:LIST?

Title: Remote Firmware Package List

Description: This command queries the instrument for a list of firmware packages that can be installed remotely. The package list will be populated from a resource file that is downloaded from a URL specified by the current value of :SYSTem:FIRMware:UPDate:REMote:SOURce.

The response is ASCII response consisting of one or more comma-delimited package names. For example: 2.0.0,1.0.0

These package names enumerate the valid parameter values to the :SYSTem:FIRMware:UPDate:REMote command.

:SYSTem:FIRMware:UPDate:REMote:LIST:REFResh

Title: Remote Firmware Package List Refresh

Description: This command causes the list of firmware packages that can be installed remotely (:SYSTem:FIRMware:UPDate:REMote:LIST?) to be refreshed from the remote source server immediately. Normally the list is automatically updated on a timer interval to limit the amount of network requests. The exact timer interval varies by model and could be anywhere from one minute to one day. Use this command if you do not want to wait for the next automatic timer-based update to the remote firmware package list.

Note that this command only initiates the request to the server to get the updated list of packages. It does not wait until the list (as visible by :SYSTem:FIRMware:UPDate:REMote:LIST?) is actually updated as the request is made asynchronously.

:SYSTem:FIRMware:UPDate:REMote:SOURce <string>
:SYSTem:FIRMware:UPDate:REMote:SOURce?

Title: Remote Firmware Package Source

Description: This command sets or queries the instrument's remote firmware update package list source. This source must be an http URL that is accessible to the instrument over the current network interface.

The set version of this command takes a string that spells the URL to the package list file. The query version returns that string. The default value points to the repository on anritsu.com where official firmware packages for this instrument are distributed. Most users will never need to change this setting from the default.

For example, the url value either set or query returned may be:

http://softwareupdates.anritsu.com/en-us/ms209xa.txt

The package list file must be a JSON file that has a minimum structure. The root object must contain a "version" string (which is reserved but currently unused), a "packages" array of zero or more package objects (each at minimum have a "name", "url", "version", and "model"), and a "default" string that matches one of the package names, or is "". The version and model attributes have semantic meaning, but the name and filename (url) can be anything. An example package list file is shown below:

```
{
"default": "V2021.11.2 (MS209xA)",
"packages": [
{
"name": "V2021.11.2 (MS209xA)",
"version": "V2021.11.2",
"model": "MS209xA",
"url": "http://softwareupdates.anritsu.com/en-us/MS2090A_V2021.11.2.raucb"
}
],
"version": "1.0.0"
}
```

or, as with MS2710xA:

```
{
"default": "V1.0.0 (MS2710xA)",
"packages": [
{
"name": "V1.0.0 (MS2710xA)",
"version": "V1.0.0",
"model": "MS2710xA",
"url": "http://softwareupdates.anritsu.com/en-us/MS2710xA_V1.0.0.tar"
}
],
"version": "1.0.0"
}
```

When the package source is set, the instrument will attempt to download the file and validate it according to the above rules before making the change permanent. If the file cannot be accessed or the file does not conform to the minimum criteria, a device specific error will be added to the error queue and the package source will remain unchanged.

Parameter(s): <string>

:SYSTem:FIRMware:VERSion?

Title: Firmware Version

Description: This command queries the device firmware version.

:SYSTem:GPS:SEARch:MODE <GGLonass|GGALileo|GBEidou|GGGalileo> :SYSTem:GPS:SEARch:MODE?

Title: GPS Satellite System Search Mode

Description: Sets and queries the GPS Satellite system search option.

The satellite system search options can be:

GPS: GPS only

GGLonass: GPS + GLONASS

GGALileo: GPS + GALILEO

GBEidou: GPS + BEIDOU

GGGalileo: GPS + GLONASS + GALILEO

Parameter(s): <GGLonass|GGALileo|GBEidou|GGGalileo>

Query Return: GGL|GGAL|GBE|GGG

Default Value: GGGalileo

:SYSTem:GPS:STATe <0 | 1 | ON | OFF> :SYSTem:GPS:STATe?

Title: GPS State

Description: Enable or disable the GPS. When the GPS is OFF, the GPS module is placed into standby to save power, and features which depend on the GPS may be unusable.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:SYSTem:LOG:ERRor?

Title: System Error Log

Description: This command retrieves the error log. The log is encrypted and can be saved to a file and sent to Anritsu Service if required.

:SYSTem:MACaddress?

Title: MAC Address

Description: Query the MAC address of the instrument.

Setting the MAC address is only available to customer service.

:SYSTem:MODel:OVERride <NONE|MG369xA|MG369xB|MG369xC>
:SYSTem:MODel:OVERride?

Title: System Model Override

Description: Allows the user to specify an override for the model returned by *IDN? for backwards compatibility. Defaults to NONE.

Parameter(s): <NONE|MG369xA|MG369xB|MG369xC>

Query Return: NONE|MG369xA|MG369xB|MG369xC

Default Value: NONE

:SYSTem:OPTions?

Title: Query Options

Description: This command retrieves the options that are currently set. The response is returned in "/" delimited form.

:SYSTem:OPTions:ALL?

Title: Query All Options

Description: This command returns all installed options with their metadata in JSON format. The expiration date time is in UTC time and in a string format of the form yyyy-MM-ddTHH:mm:ss corresponding to the ISO 8601 specification.

Example:

[{"expirationDate":"","option":"0090", "state":"active", "description":"Gated Sweep"},

{"expirationDate":"2022-07-05T19:00:42Z","option":"9103", "state":"pendingReboot", "description":"50 MHz Analysis Bandwidth" },

{"expirationDate":"","option":"9431", "state":"available", "description":"Coverage Mapping"}]

:SYSTem:OPTions:CONFig?

Title: Option Configuration

Description: This command returns a quoted string of characters readable only by Anritsu Customer Service. Only instrument configuration information is returned. No setup information is included.

:SYSTem:OPTions:TIMed:ACTivate <string>

Title: Activate Timed Options

Description: This command activates an existing Timed Option.

Parameter(s): <string>

:SYSTem:OPTions:UPDate:REMote

Title: Option Update Remote

Description: This command retrieves all available Licenses for given instrument and installs them.

:SYSTem:OPTions:UPGRade <string>

Title: Option Upgrade

Description: This command upgrades the options of the device with a valid license key provided as a quoted string parameter. The device must be rebooted immediately after this command for the installation of the new options to complete. The device must not be powered off while the command is executing. When the operation completes, a success message will be shown and request to reboot the device. The recommended way to restart the device safely is by sending the *RST command after this command. This command might report the following error code: -200: The command failed to complete. A detailed error message will be provided in the response of :SYStem:ERRor[:NEXT]? command. -2: The command failed because the options provided in the option file are incompatible with this instrument.

Parameter(s): <string>

:SYSTem:PRESet

Title: Preset

Description: This command presets parameters in the current application to their factory default values.

:SYSTem:REBoot

Title: System Reboot

Description: This command saves the current state of the instrument and reboots the instrument.

:SYSTem:SHUTdown

Title: System Shutdown

Description: This command shuts down the instrument. The current instrument state and settings will be saved.

:SYSTem:TEMPerature?

Title: Get CPU Temperature

Description: This command returns the CPU temperature in Celsius.

:SYSTem:TIME <numeric_value>,<numeric_value>,<numeric_value>
:SYSTem:TIME?

Title: System Time

Description: This command sets and queries the system's internal clock. Changing the system time does not affect the system time zone (if time zone is available). The three parameters for this command are <hour>,<minute> and <second>. The query response message shall consist of three fields separated by commas: <hour>,<minute>,<second>. Note: Some products only support the query version of this command.

Parameter(s): <numeric_value>,<numeric_value>,<numeric_value>

:SYSTem:TIME:SYNC <AUTO|MANual>
:SYSTem:TIME:SYNC?

Title: Sync System Time

Description: This command sets the system to either automatically synchronize with GPS/Internet server or manually adjust the time setting. Valid parameter values are AUTO|MANual. When AUTO is selected, GPS takes priority if it is present and turned on (see command :SYSTem:GPS:STATe to set GPS on or off). When the time setting is AUTO synchronized, the current time zone setting is not changed.

Parameter(s): <AUTO|MANual>

Query Return: AUTO|MAN

Default Value: AUTO

:SYSTem:TIME:SYNC:STATus?

Title: Sync Time Status

Description: This command query the last system time synchronization status. The return value range are INT, GPS, NONE, or NA.

INT - the system time was successfully synchronized with internet server.

GPS - the system time was successfully synchronized with GPS device time.

NONE - the system time was set to automatically synchronize with GPS/internet. However, it was unable to synchronize with the time source. It may indicates the time source connection is not available or time source is not ready such as GPS is off.

NA - the system time was set to manually update, thus synchronization status is not available.

Query Return: INT|GPS|NONE|NA

Default Value: INTernet

:SYSTem:TIME:ZONE <string>
:SYSTem:TIME:ZONE?

Title: System Time Zone

Description: This command sets and queries the system time zone. The time zone parameter is in string format of "(UTC+/-offset) Area/Location" as returned from the :SYSTem:TIME:ZONE:CATalog command. Changing the system time zone affects the system date/time.

Parameter(s): <string>

Default Value: (UTC-08:00) America/Los_Angeles

:SYSTem:TIME:ZONE:CATalog?

Title: System Time Zone Catalog

Description: This command queries the available system time zone catalog. The query returns comma delimited description for time zone sorted by UTC offset. Each item is in the format of "(UTC+/-offset) Area/Location" where Area/Location is defined in the tz database

:SYSTem:UNIT:NAMe <string>
:SYSTem:UNIT:NAMe?

Title: System Unit Name

Description: This command sets/queries the unit name of the current system.The name of the unit must not be greater than 45 ASCII characters

Parameter(s): <string>

:TEST:SELFtest?

Title: Self Test

Description: Perform a self-test and return the results. The response is formatted as a JSON (http://json.org/) array of name,value pair result objects. Some result objects also include a 'status' property that will be either 'pass' or 'fail' based on whether the criteria for that test was met.

For tests of voltages, the test passes if the measured voltage is within 10% of the expected voltage.

4-5

:ABORt

Title: Abort Sweep

Description: Forces the trigger system to the idle state. Any sweep in progress is aborted as soon as possible.

:DIAGnostic:CALibration:AMPLitude:INTernal:ABORt

Title: Internal Level Calibration Abort

Description: Aborts a currently running internal level calibration. Will fail if no calibration is in progress.

:DIAGnostic:CALibration:AMPLitude:INTernal:DATE?

Title: Internal Level Calibration Date

Description: Retrieve the date of the last successful internal level calibration. The date will be 'UNKNOWN' if there is no valid calibration data saved.

Default Value: UNKNOWN

:DIAGnostic:CALibration:AMPLitude:INTernal:INFO?

Title: Internal Level Calibration Info

Description: Query the information of the last successful internal level calibration. Data will be in the JSON format below:

{

"Date" : <string in MM/DD/YYYY>,

"Time" : <string in hh:mm>,

"Version" : <string version>,

"Sensor Model" : <string model>,

"Frequency Start MHz" : <start frequency in MHz>,

"Frequency Stop MHz" : <stop frequency in MHz>,

"Level Start dBm" : <start power in dBm>,

"Level Stop dBm" : <stop power in dBm>

}

:DIAGnostic:CALibration:AMPLitude:INTernal:PAD:LOAD <string>,<string>

Title: Internal Level Calibration Pad Load

Description: Imports the specified .s2p calibration file to be used by the external 20 dB pad required for the internal level calibration process.

Input parameters are the name of the storage device and the file path. Available devices can be seen with :MMEMory:CATalog:MSUSs?

Parameter(s): <string>,<string>

:DIAGnostic:CALibration:AMPLitude:INTernal:PAD:NAME?

Title: Internal Level Calibration Pad File Name

Description: Retrieves the name of the S2P file used to characterize the external 20 dB pad used by the internal level calibration process. Returns 'UNKNOWN' if there is no file loaded.

Default Value: UNKNOWN

:DIAGnostic:CALibration:AMPLitude:INTernal:PAD:STORe <string>,<string>

Title: Internal Level Calibration Pad Store

Description: Exports the current .s2p calibration file used by the internal level calibration's external pad.

Input parameters are the name of the storage device and the file path. Available devices can be seen with :MMEMory:CATalog:MSUSs?

Parameter(s): <string>,<string>

:DIAGnostic:CALibration:AMPLitude:INTernal:RANGe?

Title: Internal Level Calibration Range

Description: Retrieve the ranges of the last successful internal level calibration. The range will be 'UNKNOWN' if there is no valid calibration data saved.

Default Value: UNKNOWN

:DIAGnostic:CALibration:AMPLitude:INTernal:SENSor?

Title: Internal Level Calibration Sensor

Description: Retrieve the sensor used in the last successful internal level calibration. The sensor model will be 'UNKNOWN' if there is no valid calibration data saved.

Default Value: UNKNOWN

:DIAGnostic:CALibration:AMPLitude:INTernal:STARt

Title: Internal Level Calibration Start

Description: Begins the internal level calibration routine to correct the aging characteristics of the ALC components.

Will fail if:

There is no connected USB power sensor.

There is a calibration already in progress.

:DIAGnostic:CALibration:AMPLitude:INTernal:STATe <0 | 1 | ON | OFF>
:DIAGnostic:CALibration:AMPLitude:INTernal:STATe?

Title: Internal Level Calibration State

Description: Enable or disable the field calibration of ALC and related components.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:DIAGnostic:CALibration:ATTenuator:INTernal:ABORt

Title: Internal Attenuator Calibration Abort

Description: Aborts a currently running internal attenuator calibration. Will fail if no calibration is in progress.

:DIAGnostic:CALibration:ATTenuator:INTernal:DATE?

Title: Internal Attenuator Calibration Date

Description: Retrieve the date of the last successful internal level calibration. The date will be 'UNKNOWN' if there is no valid calibration data saved.

Default Value: UNKNOWN

:DIAGnostic:CALibration:ATTenuator:INTernal:INFO?

Title: Internal Attenuator Calibration Info

Description: Query the information of the last successful internal attenuator calibration. Data will be in the JSON format below:

```
{
"Date" : <string in MM/DD/YYYY>,
"Time" : <string in hh:mm>,
"Version" : <string version>,
"Sensor Model" : <string model>,
"Frequency Start MHz" : <start frequency in MHz>,
"Frequency Stop MHz" : <stop frequency in MHz>,
"Level Start dBm" : <start power in dBm>,
"Level Stop dBm" : <stop power in dBm>
}
```

:DIAGnostic:CALibration:ATTenuator:INTernal:RANGe?

Title: Internal Attenuator Calibration Range

Description: Retrieve the ranges of the last successful internal attenuator calibration. The range will be 'UNKNOWN' if there is no valid calibration data saved.

Default Value: UNKNOWN

:DIAGnostic:CALibration:ATTenuator:INTernal:SENSor?

Title: Internal Attenuator Calibration Sensor

Description: Retrieve the sensor used in the last successful internal attenuator calibration. The sensor model will be 'UNKNOWN' if there is no valid calibration data saved.

Default Value: UNKNOWN

:DIAGnostic:CALibration:ATTenuator:INTernal:STARt

Title: Internal Attenuator Calibration Start

Description: Begins the internal attenuator calibration routine to correct the aging characteristics of the attenuator components.

Will fail if:

There is no connected USB power sensor.

There is a calibration already in progress.

:DIAGnostic:CALibration:ATTenuator:INTernal:STATe <0 | 1 | ON | OFF>
:DIAGnostic:CALibration:ATTenuator:INTernal:STATe?

Title: Internal Attenuator Calibration State

Description: Enable or disable the field calibration of the attenuator.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:DIAGnostic:CALibration:INTernal:EXPort <string>,<string>

Title: Internal Calibration Export

Description: Bundles all self-calibration data together in a zip file for analysis by customer service.

Input parameters are the name of the storage device and the file path. Available devices can be seen with :MMEMory:CATalog:MSUSs?

Parameter(s): <string>,<string>

:DIAGnostic:CALibration:REFerence:ABORt

Title: Internal 100 MHz Reference Calibration Abort

Description: Abort a user calibration of the 100 MHz reference oscillator. The command will fail if there is no calibration in progress.

:DIAGnostic:CALibration:REFerence:DATE?

Title: Internal 100 MHz Reference Oscillator Calibration Date

Description: Returns the date and time of the last calibration of the 100 MHz oscillator, or "UNKNOWN" if there is no calibration data present.

Default Value: UNKNOWN

:DIAGnostic:CALibration:REFerence:INITiate

Title: Internal 100 MHz Reference Oscillator Calibration Initiate

Description: Initiates an internal user calibration for the 100 MHz reference oscillator.

This command is valid with and without the Low Phase Noise and Ultra High Stability option.

This calibration will generate an error if there is no 10 MHz reference or if the unit cannot lock to the 10 MHz reference.

The Low Phase Noise and Ultra High Stability calibration requires an external 10 MHz or 1PPS input. This calibration optimizes for an acceptable frequency error between the 100 MHz and the external input. This calibration will generate an error if initiated while the unit is not locked to an external 10 MHz or 1PPS.

:DIAGnostic:CALibration:REFerence:LFRequency:ABORt

Title: Internal 10 MHz Reference Calibration Abort

Description: Abort a user calibration of the 10 MHz reference oscillator. The command will fail if there is no calibration in progress. This command is only valid when the Low Phase Noise and Ultra High Stability option is installed.

:DIAGnostic:CALibration:REFerence:LFRequency:DATE?

Title: Internal 10 MHz Reference Oscillator Calibration Date

Description: Returns the date and time of the last calibration of the 10 MHz oscillator, or "UNKNOWN" if there is no calibration data present.

Default Value: UNKNOWN

:DIAGnostic:CALibration:REFerence:LFRequency:INITiate

Title: Internal 10 MHz Reference Oscillator Calibration Initiate

Description: Initiates an internal user calibration for the 10 MHz reference oscillator.

This command requires the Low Phase Noise and Ultra High Stability option. This calibration requires an external 10 MHz input, or, with the Internal Level and Frequency Cal option, a 1PPS input is also allowed.

This calibration searches for an optimized DAC value while attempting to lock to the given input. Being locked to the external input prior to initiating the calibration is not required, however an error will be generated if the unit cannot lock to the input after the calibration is completed.

:DIAGnostic:CALibration:REFerence:LFRequency:SOURce?

Title: Internal 10 MHz Reference Oscillator Calibration Source

Description: Returns the reference source used by the last calibration of the 10 MHz oscillator, or UNKNown if there is no calibration data present.

EXTernal = calibrated using external 10 MHz input

EPPS = calibrated using external 1PPS input

GPPS = calibrated using 1PPS from internal GNSS/GPS module

UNKNown = no calibration data present

Query Return: EXT | GPPS | EPPS | UNKN

Default Value: EXTernal

:DIAGnostic:CALibration:REFerence:LFRequency:STATus?

Title: Internal 10 MHz Reference Calibration Status

Description: This command queries the status of the 10 MHz reference oscillator calibration. This command is only valid when the Low Phase Noise and Ultra High Stability option is installed.

The return value is formatted similar to SYST:ERR:NEXT?: NUMBER,"TEXT". The text is a description of the numeric code, and may contain additional info about what the calibration is doing. You can expect the return to be one of these:

0,"No calibration data." 1,"Calibration running. - <X> percent" 2,"Calibration results in effect." 3,"Calibration routine failed. - <information about the failure>"

:DIAGnostic:CALibration:REFerence:STATus?

Title: Internal 100 MHz Reference Calibration Status

Description: This command queries the status of the 100 MHz reference oscillator calibration.

The return value is formatted similar to SYST:ERR:NEXT?: NUMBER,"TEXT". The text is a description of the numeric code, and may contain additional info about what the calibration is doing. You can expect the return to be one of these:

0,"No calibration data." 1,"Calibration running. - <X> percent" 2,"Calibration results in effect." 3,"Calibration routine failed. - <information about the failure>"

:DIAGnostic:MODule:INFOrmation?

Title: Module Information

Description: Report diagnostic information for all installed HW modules. The response is formatted as a JSON (http://json.org/) array of objects containing a "module" property specifying the name of the module and an "id" property specifying the HW board ID.

Additionally, some modules, when installed, may have extra properties containing more diagnostic information specific to the module. Switched Filter Module (SWF): type : 'standard' or 'high power'

YIG: type : 'esfera' or 'anritsu'

Synthesizer: type : 'Low Phase Noise Modulation Capable' or 'Modulation Capable' or 'Prototype'

Modulation Processing Module: fpgaVersion : FPGA version number

fpgaDebugVersion : FPGA test version, if any

fpgaTimestamp : FPGA build time

fpgaType : FPGA type string

fpgaCommitHash : FPGA build SHA

:DIAGnostic:SNUM?

Title: Serial Number

Description: This query returns the serial number of the instrument.

:DISPlay[:WINDow]:TEXT:STATe <0 | 1 | ON | OFF>
:DISPlay[:WINDow]:TEXT:STATe?

Title: Secure Mode

Description: Set or query the Display Text State. When enabled, the front panel

display will show all frequency, level, and modulation parameters.

When disabled, these parameters will be hidden. The instrument

will function normally even when this setting is disabled.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

:INITiate:CONTinuous <0 | 1 | ON | OFF>
:INITiate:CONTinuous?

Title: Initiate Continuous

Description: Continuously rearms the trigger system after completion of a triggered sweep.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

:INITiate[:IMMediate]

Title: Initiate Immediate

Description: Places the trigger system into the armed state from the idle state. If trigger system is not in idle state, or if :INITiate:CONTinuous is ON, will produce error -213.

:INITiate:SG:SELFtest?

Title: Self Test

Description: Perform a self-test and return the results. The response is formatted as a JSON (http://json.org/) array of name,value pair result objects. Some result objects also include a 'status' property that will be either 'pass' or 'fail' based on whether the criteria for that test was met.

:MEASure:AM[:EXTernal]:DEPTh?

Title: Measured AM Depth

Description: Queries the AM depth measurement. The voltage of the external AM input will be measured and used to calculate the percentage (linear mode) or dB (log mode) modulation value. The returned value is in percent if AM Type is LINear, or dB if AM Type is LOGarithmic. This command is only valid when the AM source is set to EXTernal, and will result in an execution error if the AM source is currently INTernal.

Query Return: Numeric

Default Value: 0

Range: 0 to 100

:MEASure:FM[:EXTernal]:DEViation?

Title: Fm Measure Deviation

Description: Queries the FM Deviation measurement in radians. The voltage of the external FM meter will be measured and used to calculate the FM Deviation measurement. This command is only valid when the FM State is On and FM source is set to EXTernal, and will result in an execution error if either condition is not met.

Query Return: Numeric

Default Value: 0

Range: -420000000 to 420000000

:MEASure:PM:DEViation?

Title: Pm Measure Deviation

Description: Queries the PM Deviation measurement in radians. The voltage of the external FM meter will be measured and used to calculate the PM Deviation measurement. This command is only valid when the PM State is On and PM source is set to EXTernal, and will result in an execution error if either condition is not met.

Query Return: Numeric

Default Value: 0

Range: -3220 to 3220

:MMEMory:LOAD:CORRection<n> <string>,<string>

Title: Correction Import

Description: Import a user level flatness correction power-offset table from the specified storage device and save it under the currently selected table.

Warning: this will overwrite the table set by [:SOURce]:CORRection:CSET:SELect The command will fail if there is no selected table.

Input parameters are the name of the storage device and the file path. Available devices can be seen with :MMEMory:CATalog:MSUSs?

Parameter(s): <string>,<string>

CORRection Suffix Range: 1-5, Default = 1

:MMEMory:STORe:CORRection<n> <string>,<string>

Title: Correction Export

Description: Export the current user level flatness correction power-offset table to the specified storage device.

Input parameters are the name of the storage device and the file path. Available devices can be seen with :MMEMory:CATalog:MSUSs?

Parameter(s): <string>,<string>

CORRection Suffix Range: 1-5, Default = 1

:OUTPut:IMPedance?

Title: RF Output Impedance

Description: Queries the RF output impedance. The impedance is nominally 50 ohms and is

not settable.

:OUTPut:PNAmplifier:STATe <0 | 1 | ON | OFF>
:OUTPut:PNAmplifier:STATe?

Title: Phase Noise Amplifier Enable

Description: Enables or disables routing of the RF through the PNA or the Front Panel. Requires having the Phase Noise Amplifier option installed.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:OUTPut:PRESet[:STATe] <0 | 1 | ON | OFF>
:OUTPut:PRESet[:STATe]?

Title: Select RF Output State on Preset

Description: Selects to turn RF output power on/off after a preset.

ON (1): RF output power is turned on after a preset.

OFF (0): RF output power is turned off after a preset.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:OUTPut:PROTection <0 | 1 | ON | OFF>
:OUTPut:PROTection?

Title: Enable RF Blanking

Description: Configures the RF output behavior during frequency transitions in CW or step sweep mode.

ON (1): RF output is turned off (blanked) during frequency transitions.

OFF (0): RF output remains on (unblanked) during frequency transitions.

Frequency transitions across band switch points always turn RF off (blank).

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:OUTPut:PROTection:RETRace <0 | 1 | ON | OFF>
:OUTPut:PROTection:RETRace?

Title: RF Sweep Retrace

Description: ON causes the RF output to be turned off during analog sweep retrace.

OFF leaves RF output turned on during analog sweep retrace.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

:OUTPut:ROSCillator:HFRequency[:STATe] <0 | 1 | ON | OFF>
:OUTPut:ROSCillator:HFRequency[:STATe]?

Title: Reference Oscillator 100 MHz Output

Description: Turn the reference oscillator external 100 MHz output on or off. Requires Low Phase Noise and Ultra High Stability option.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:OUTPut:ROSCillator:MULTiplied[:STATe] <0 | 1 | ON | OFF>
:OUTPut:ROSCillator:MULTiplied[:STATe]?

Title: Reference Oscillator 1600 MHz Output

Description: Turn the reference oscillator external 1600 MHz output on or off. Requires Low Phase Noise and Ultra High Stability option.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:OUTPut:ROSCillator:PPSecond[:STATe] <0 | 1 | ON | OFF>
:OUTPut:ROSCillator:PPSecond[:STATe]?

Title: Reference Oscillator 1PPS Output

Description: Turn the reference oscillator external 1PPS output on or off. Requires Low Phase Noise and Ultra High Stability option.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:OUTPut:ROSCillator[:STATe]?

Title: Reference Oscillator 10 MHz Output

Description: Get the reference oscillator external 10 MHz output state. The 10 MHz output is always on and this query always returns '1'.

Query Return: 0 | 1

Default Value: OFF

:OUTPut[:STATe] <0 | 1 | ON | OFF>
:OUTPut[:STATe]?

Title: Enable RF Output

Description: Turns RF output power on/off.

ON (1): RF output power is turned on.

OFF (0): RF output power is turned off.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:AM:ERRor?

Title: AM Error

Description: AM error status that can originate from a measurement limit violation or a rate vs. waveform type violation.

AM over range error = 0x1

AM rate exceeded error = 0x2

Query Return: Numeric

Default Value: 0

Range: 0 to 255

[:SOURce]:AM[:EXTernal]:LOGSens <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:AM[:EXTernal]:LOGSens? [DEFault | MINimum | MAXimum]

Title: AM Log Sensitivity

Description: Sets the AM sensitivity for the external AM Log mode.

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 3 dB/V

Default Unit: dB/V

Range: 0 dB/V to 25 dB/V

[:SOURce]:AM[:EXTernal]:SENSitivity <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:AM[:EXTernal]:SENSitivity? [DEFault | MINimum | MAXimum]

Title: AM Lin Sensitivity

Description: Sets the AM sensitivity for the external AM Linear mode.

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 50 PCT/V

Default Unit: PCT/V

Range: 0 PCT/V to 100 PCT/V

[:SOURce]:AM[:INTernal]:DEPTh <percent_value> | DEFault | MINimum | MAXimum
[:SOURce]:AM[:INTernal]:DEPTh? [DEFault | MINimum | MAXimum]

Title: AM Lin Depth

Description: Sets the modulation depth of the AM signal in the internal AM Linear mode.

Query Return: Percent

Set Command Parameter(s): <percent_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 50

Range: 0% to 100%

[:SOURce]:AM[:INTernal]:FREQuency <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:AM[:INTernal]:FREQuency? [DEFault | MINimum | MAXimum]

Title: AM Internal Frequency

Description: Sets the frequency of the modulating waveform for the internal AM function.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1000 Hz

Default Unit: Hz

Range: 0.1 Hz to 10000000 Hz

[:SOURce]:AM[:INTernal]:LOGDepth <numeric_value> {DB} | DEFault | MINimum | MAXimum
[:SOURce]:AM[:INTernal]:LOGDepth? [DEFault | MINimum | MAXimum]

Title: AM Log Depth

Description: Sets the modulation depth of the AM signal in the internal AM Log mode.

Query Return: Numeric (dB)

Set Command Parameter(s): <numeric_value> {DB} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 3 dB

Default Unit: dB

Range: 0 dB to 25 dB

[:SOURce]:AM[:INTernal]:WAVE <SINE|GAUSsian|RDOWn|RUP|SQUare|TRIangle|UNIForm>
[:SOURce]:AM[:INTernal]:WAVE?

Title: AM Internal Wave

Description: Selects the modulating waveform (from the internal AM generator) for the internal AM function, as follows:

SINE = Sine wave

GAUSsian = Guassian noise

RDOWn = Negative ramp

RUP = Positive ramp

SQUare = Square wave

TRIangle = Triangle wave

UNIForm = Uniform noise

Parameter(s): <SINE|GAUSsian|RDOWn|RUP|SQUare|TRIangle|UNIForm>

Query Return: SINE|GAUS|RDOW|RUP|SQU|TRI|UNIF

Default Value: SINE

[:SOURce]:AM:SOURce <INTernal|EXTernal>
[:SOURce]:AM:SOURce?

Title: AM Source

Description: Selects the source of the AM modulating signal, as follows:

INTernal = Internal AM generator

EXTernal selects the rear panel AM IN connector. EXTernal1 and EXTernal2 have the same functionality as EXTernal.

Parameter(s): <INTernal|EXTernal>

Query Return: INT|EXT

Default Value: INTernal

[:SOURce]:AM:STATe <0 | 1 | ON | OFF>
[:SOURce]:AM:STATe?

Title: AM State

Description: Enable/disable amplitude modulation of RF output signal.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:AM:TYPE <LINear|LOGarithmic> [:SOURce]:AM:TYPE?

Title:	AM Type
Description:	Selects the AM operating mode.
Parameter(s):	<LINear|LOGarithmic>
Query Return:	LIN|LOG
Default Value:	LINear

[:SOURce]:CORRection<n>:PMETer:OFFSet <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:CORRection<n>:PMETer:OFFSet? [DEFault | MINimum | MAXimum]

Title:	Correction Offset
Description:	Sets the offset for the power meter during user level flatness calibration. The command suffix indicates which table to update.
Query Return:	Numeric (Hz)
Set Command Parameter(s):	<numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
Query Command Parameter(s):	[DEFault | MINimum | MAXimum]
CORRection Suffix Range:	1-5, Default = 1
Default Value:	0 Hz
Default Unit:	Hz
Range:	-43500000000 Hz to 43500000000 Hz

[:SOURce]:CORRection<n>:PMETer:SCALe <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:CORRection<n>:PMETer:SCALe? [DEFault | MINimum | MAXimum]

Title:	Correction Scale
Description:	Sets the scale of the frequency of the power meter during user level flatness calibration. The command suffix indicates which table to update.
Query Return:	Numeric (Hz)
Set Command Parameter(s):	<numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
Query Command Parameter(s):	[DEFault | MINimum | MAXimum]
CORRection Suffix Range:	1-5, Default = 1
Default Value:	1 Hz
Default Unit:	Hz
Range:	0.01 Hz to 100 Hz

[:SOURce]:CORRection<n>:POINts <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:CORRection<n>:POINts? [DEFault | MINimum | MAXimum]

Title: Correction Points

Description: Sets the desired number of points to sample for the user level flatness correction power-offset table. This will change the step size to match.

The command suffix indicates which table to update.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

CORRection Suffix Range: 1-5, Default = 1

Default Value: 2

Range: 2 to 65535

[:SOURce]:CORRection<n>:STARt <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:CORRection<n>:STARt? [DEFault | MINimum | MAXimum]

Title: Correction Start

Description: Sets the start frequency for a user level flatness correction power-offset table. This will update the number of points or step size to accommodate the new range, preserving whichever was last changed by the user.

The command suffix indicates which table to update.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

CORRection Suffix Range: 1-5, Default = 1

Default Value: 0 Hz

Default Unit: Hz

Range: 0 Hz to 43500000000 Hz

[:SOURce]:CORRection<n>:STEP <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:CORRection<n>:STEP? [DEFault | MINimum | MAXimum]

Title: Correction Step

Description: Sets the step frequency for a user level flatness correction power-offset table. This will change the number of points to match.

The command suffix indicates which table to update.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

CORRection Suffix Range: 1-5, Default = 1

Default Value: 1 Hz

Default Unit: Hz

Range: 1 Hz to 43500000000 Hz

[:SOURce]:CORRection<n>:STOP <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:CORRection<n>:STOP? [DEFault | MINimum | MAXimum]

Title: Correction Stop

Description: Sets the stop frequency for a user level flatness correction power-offset table.

The command suffix indicates which table to update.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

CORRection Suffix Range: 1-5, Default = 1

Default Value: 0 Hz

Default Unit: Hz

Range: 0 Hz to 43500000000 Hz

[:SOURce]:CORRection:CALibrate:ABORt

Title: Correction Calibration Abort

Description: Abort a user level flatness correction calibration.

The command will fail if there is no calibration in progress.

[:SOURce]:CORRection:CALibrate:STARt

Title: Correction Calibration Start

Description: Initiates the user level flatness correction calibration, which will generate a power-offset table based on values read by the current correction device.

The command will fail if:

No table is selected with [:SOURce]:CORRection:CSET:SELect

No correction device is detected.

A calibration is already in progress.

The currently selected table is misconfigured (e.g. the points and step size cannot reach the stop frequency, or the frequency range is outside the correction device's capabilities).

[:SOURce]:CORRection:CALibrate:STATus?

Title: User Level Calibration Status

Description: This command queries the status of the user

level flatness correction calibration.

The return value is formatted similar to SYST:ERR:NEXT?: NUMBER,"TEXT". The text is a description of the numeric code, and may contain additional info about what the calibration is doing. You can expect the return to be one of these:

0,"No calibration data." 1,"Calibration running. - <X> percent" 2,"Calibration results in effect." 3,"Calibration routine failed. - <information about the failure>"

[:SOURce]:CORRection:CSET:SELect <NONE|USER{1}|USER{2}|USER{3}|USER{4}|USER{5}>
[:SOURce]:CORRection:CSET:SELect?

Title: Correction Setting

Description: Selects the user level flatness correction power-offset table to be applied to the output by the command [:SOURce]:CORRection:STATe ON.

Parameter(s): <NONE|USER{1}|USER{2}|USER{3}|USER{4}|USER{5}>

Query Return: NONE|USER1|USERUSER2|USER3|USERUSER4

Default Value: NONE

[:SOURce]:CORRection:DEVice <USB|ML2437|ML2438>
[:SOURce]:CORRection:DEVice?

Title: Correction Device

Description: Select the type of device to be used during a user level calibration.

Parameter(s): <USB|ML2437|ML2438>

Query Return: USB|ML2437|ML2438

Default Value: USB

[:SOURce]:CORRection[:STATe] <0 | 1 | ON | OFF>
[:SOURce]:CORRection[:STATe]?

Title:	Correction State
Description:	Turns the selected user level flatness correction power-offset table on/off.
Parameter(s):	<0 | 1 | ON | OFF>
Query Return:	0 | 1
Default Value:	OFF

[:SOURce]:FM:ERRor?

Title:	FM Error
Description:	FM error status that can originate from a measurement limit violation or a rate vs. waveform type violation. FM over range error = 0x1 FM rate exceeded error = 0x2 FM Deviation exceeded = 0x4
Query Return:	Numeric
Default Value:	0
Range:	0 to 255

[:SOURce]:FM[:EXTernal]:METer:INTerval <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
[:SOURce]:FM[:EXTernal]:METer:INTerval? [DEFault | MINimum | MAXimum]

Title:	FM Meter Interval
Description:	Sets the FM meter measuring interval.
Query Return:	Numeric (s)
Set Command Parameter(s):	<numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
Query Command Parameter(s):	[DEFault | MINimum | MAXimum]
Default Value:	0.01 s
Default Unit:	s
Range:	1e-06 s to 2 s

[:SOURce]:FM[:EXTernal]:SENSitivity <numeric_value> {HZ/V | KHZ/V | MHZ/V | GHZ/V} | DEFault | MINimum | MAXimum
[:SOURce]:FM[:EXTernal]:SENSitivity? [DEFault | MINimum | MAXimum]

Title: Fm Sensitivity

Description: Sets the FM sensitivity for the external FM function.

Set Command Parameter(s): <numeric_value> {HZ/V | KHZ/V | MHZ/V | GHZ/V} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1000000 Hz/V

Default Unit: Hz/V

Range: -420000000 Hz/V to 420000000 Hz/V

[:SOURce]:FM:HFILter <DC|HP63|HP1K>
[:SOURce]:FM:HFILter?

Title: FM High Pass Filter

Description: Selects high pass filter passband start frequency for the FM signal applied to either the FM external or internal input.

DC selects (0 Hz) no high pass filter.

HP63 selects the 63 Hz high pass filter.

HP1K selects the 1 kHz high pass filter.

Parameter(s): <DC|HP63|HP1K>

Query Return: DC|HP63|HP1K

Default Value: DC

[:SOURce]:FM[:INTernal]:DEViation <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:FM[:INTernal]:DEViation? [DEFault | MINimum | MAXimum]

Title: Fm Deviation

Description: Set the modulation deviation of the FM signal for the internal FM function.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1000000 Hz

Default Unit: Hz

Range: 10 Hz to 420000000 Hz

[:SOURce]:FM[:INTernal]:FREQuency <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:FM[:INTernal]:FREQuency? [DEFault | MINimum | MAXimum]

Title: Fm Internal Frequency

Description: Sets the frequency of the modulating waveform for the internal FM function.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1000 Hz

Default Unit: Hz

Range: 0.1 Hz to 10000000 Hz

[:SOURce]:FM[:INTernal]:WAVE <SINE|GAUSsian|RDOWn|RUP|SQUare|TRIangle|UNIForm>
[:SOURce]:FM[:INTernal]:WAVE?

Title: Fm Internal Wave

Description: Selects the modulating waveform (from the internal FM generator) for the internal FM function, as follows:

SINE = Sine wave

GAUSsian = Guassian noise

RDOWn = Negative ramp

RUP = Positive ramp

SQUare = Square wave

TRIangle = Triangle wave

UNIForm = Uniform noise

Parameter(s): <SINE|GAUSsian|RDOWn|RUP|SQUare|TRIangle|UNIForm>

Query Return: SINE|GAUS|RDOW|RUP|SQU|TRI|UNIF

Default Value: SINE

[:SOURce]:FM:MODE <LOCKed{[1]|2}|UNLocked>
[:SOURce]:FM:MODE?

Title: Fm Mode

Description: Sets the synthesis mode employed in generating the FM signal, as follows:

LOCKed[1|2] = Locked Low-Noise FM

UNLocked = Unlocked FM

If LOCKed1 or LOCKed2 is set, the YIG phase-locked loop is used in synthesizing the FM signal. If UNLocked is set, the YIG phase-lock loop is disabled and the FM signal is obtained by applying the modulating signal to the tuning coils of the YIG-tuned oscillator.

Parameter(s): <LOCKed{[1]|2}|UNLocked>

Query Return: LOCK|UNL

Default Value: LOCKed

[:SOURce]:FM:SOURce <INTernal|EXTernal>
[:SOURce]:FM:SOURce?

Title: Fm Source

Description: Selects the source of the FM modulating signal, as follows:

INTernal = Internal FM generator. EXTernal = Rear panel FM IN connector. EXTernal1 and EXTernal2 have the same functionality as EXTernal.

Parameter(s): <INTernal|EXTernal>

Query Return: INT|EXT

Default Value: INTernal

[:SOURce]:FM:STATe <0 | 1 | ON | OFF>
[:SOURce]:FM:STATe?

Title: Fm State

Description: Enable/disable frequency modulation of RF output signal.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:FREQuency:CENTer <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:FREQuency:CENTer? [DEFault | MINimum | MAXimum]

Title: Center Frequency

Description: Sets the center frequency for SWEep[1].

Center and span frequencies are coupled values. Entering the value for one will cause the other to be recalculated.

Setting the center frequency may affect sweep parameters start, stop, span, step, and points. Related commands:

[:SOURce]:FREQuency:MODE, [:SOURce]:FREQuency:STARt,

[:SOURce]:FREQuency:STOP, [:SOURce]:FREQuency:SPAN,

[:SOURce]:SWEep1:STEP, [:SOURce]:SWEep1:POINTs

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 40000004500 Hz

Default Unit: Hz

Range: 9000.0005 Hz to 79999999999.9995 Hz

[:SOURce]:FREQuency[:CW|FIXed] <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum | UP | DOWN [:SOURce]:FREQuency[:CW|FIXed]? [DEFault | MINimum | MAXimum | UP | DOWN]

Title: CW Frequency

Description: Sets the RF output frequency.

Parameters UP | DOWN increment/decrement the frequency of the RF output frequency by the value specified by the [:SOURce]:FREQuency:STEP:INCRement command.

Keywords :CW and :FIXed are equivalent and may be used interchangeably; they also are optional and may be omitted.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum | UP | DOWN

Query Command Parameter(s): [DEFault | MINimum | MAXimum | UP | DOWN]

Default Value: 10000000000 Hz

Default Unit: Hz

Range: 9000 Hz to 80000000000 Hz

[:SOURce]:FREQuency[:CW|FIXed]:STEP[:INCRement] <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:FREQuency[:CW|FIXed]:STEP[:INCRement]? [DEFault | MINimum | MAXimum]

Title:	Frequency Step
Description:	Sets the step increment size used with the :FREQuency:CW command.
Query Return:	Numeric (Hz)
Set Command Parameter(s):	<numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
Query Command Parameter(s):	[DEFault | MINimum | MAXimum]
Default Value:	100000000 Hz
Default Unit:	Hz
Range:	0.001 Hz to 10000000000 Hz

[:SOURce]:FREQuency[:CW]:SELect <F0|F1|F2|F3|F4|F5|F6|F7|F8|F9> [:SOURce]:FREQuency[:CW]:SELect?

Title:	Preset Frequency Select
Description:	Selects a Preset Frequency Index.
Parameter(s):	<F0|F1|F2|F3|F4|F5|F6|F7|F8|F9>
Query Return:	F0|F1|F2|F3|F4|F5|F6|F7|F8|F9
Default Value:	F1

[:SOURce]:FREQuency:DELTa <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:FREQuency:DELTa? [DEFault | MINimum | MAXimum]

Title: Delta F Setting

Description: Sets the value of the delta frequency setting which is utilized by the frequency sweep range select command.

The delta frequency represents a frequency range centered about the corresponding selected sweep range frequency preset (F5 or F6). When a frequency sweep range utilizing the delta frequency (DF) is selected, the start frequency and stop frequency are modified accordingly:

Start Frequency = Frequency Preset - (Delta Frequency / 2)

Stop Frequency = Frequency Preset + (Delta Frequency / 2).

The delta frequency is only applied to the Start and Stop Frequencies when a Sweep Range Selection is applied.

Related commands:

[:SOURce]:SWEep1:RANGe

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1000000000 Hz

Default Unit: Hz

Range: 0.002 Hz to 79999991000 Hz

[:SOURce]:FREQuency:F<n> <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:FREQuency:F<n>? [DEFault | MINimum | MAXimum]

Title: Preset Frequency Value

Description: Edits a selected preset frequency value.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

F Suffix Range: 0-9, Default = 0

Default Value: 10000000000 Hz

Default Unit: Hz

Range: 9000 Hz to 80000000000 Hz

[:SOURce]:FREQuency:MODE <CW|FIXed|SWEep{[1]}|LIST{[1]}|LIST{2}|LIST{3}|LIST{4}|LIST{5}|LIST{6} |LIST{7}|LIST{8}|LIST{9}|LIST{10}> [:SOURce]:FREQuency:MODE?

Title: Frequency Mode

Description: Specifies which command subsystem controls the frequency, as follows:

CW|FIXed = [:SOURce]:FREQuency:CW|FIXed

SWEep[1] = [:SOURce]:SWEep[1]

LIST<n> = [:SOURce]:LIST<n>

:SWEep and :SWEep1 may be used interchangeably.

In SWEep[1] mode, frequency will be determined by programmed values for the following :FREQuency subsystem commands:

:CENTer and :SPAN, or, :STARt and :STOP.

In LIST mode, frequency is determined by programmed values for :LIST<n>:FREQuency, where LIST<n> = LIST[1], LIST2, LIST3, ..., LIST10. :LIST and :LIST1 may be used interchangeably. List frequencies and powers are deleted on Factory Reset, User File Reset, and Reset All.

Related commands:

[:SOURce]:FREQuency:CENTer, [:SOURce]:FREQuency:SPAN,

[:SOURce]:FREQuency:STARt, [:SOURce]:FREQuency:STOP,

[:SOURce]:LIST<n>:FREQuency, [:SOURce]:FREQuency[:CW|FIXed]

Parameter(s):

<CW|FIXed|SWEep{[1]}|LIST{[1]}|LIST{2}|LIST{3}|LIST{4}|LIST{5}|LIST{6}|LIST{ 7}|LIST{8}|LIST{9}|LIST{10}>

Query Return:

CW|FIX|SWE1|SWELIST1|LIST2|LISTLIST3|LIST4|LISTLIST5|LIST6|LISTLIS T7|LIST8|LISTLIST9

Default Value: CW

[:SOURce]:FREQuency:SPAN <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:FREQuency:SPAN? [DEFault | MINimum | MAXimum]

Title: Frequency Span

Description: Sets the frequency span for SWEep1.

Span and center frequencies are coupled values. Entering the value for one will cause the other to be recalculated.

Setting the span frequency may affect sweep parameters start, stop, center, step, and points.

Related commands:

[:SOURce]:FREQuency:MODE, [:SOURce]:FREQuency:STARt,

[:SOURce]:FREQuency:STOP, [:SOURce]:FREQuency:CENTer,

[:SOURce]:SWEep1:STEP, [:SOURce]:SWEep1:POINTs

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 79999991000 Hz

Default Unit: Hz

Range: 0.001 Hz to 79999991000 Hz

[:SOURce]:FREQuency:SPAN:FULL

Title: Frequency Full Span

Description: Sets the frequency span for SWEep1 to the maximum span (MAX frequency – MIN frequency).

Setting the frequency span to full may affect sweep parameters start, stop, center, span, step size, and points.

Related commands:

[:SOURce]:FREQuency:SPAN, [:SOURce]:FREQuency:STARt,

[:SOURce]:FREQuency:STOP, [:SOURce]:FREQuency:CENTer,

[:SOURce]:SWEep1:STEP, [:SOURce]:SWEep1:POINTs

[:SOURce]:FREQuency:STARt <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:FREQuency:STARt? [DEFault | MINimum | MAXimum]

Title: Start Frequency

Description: Sets the start frequency for SWEep1.

Setting the start frequency may affect sweep parameters stop, center, span, step, and points.

Related commands:

[:SOURce]:FREQuency:MODE, [:SOURce]:FREQuency:STOP,

[:SOURce]:FREQuency:CENTer, [:SOURce]:FREQuency:SPAN,

[:SOURce]:SWEep1:STEP, [:SOURce]:SWEep1:POINTs

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 9000 Hz

Default Unit: Hz

Range: 9000 Hz to 79999999999.999 Hz

[:SOURce]:FREQuency:STOP <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:FREQuency:STOP? [DEFault | MINimum | MAXimum]

Title: Stop Frequency

Description: Sets the stop frequency for SWEep1.

Setting the stop frequency may affect sweep parameters start, center, span, step, and points.

Related commands:

[:SOURce]:FREQuency:MODE, [:SOURce]:FREQuency:STARt,

[:SOURce]:FREQuency:CENTer, [:SOURce]:FREQuency:SPAN,

[:SOURce]:SWEep1:STEP, [:SOURce]:SWEep1:POINTs

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 80000000000 Hz

Default Unit: Hz

Range: 9000.001 Hz to 80000000000 Hz

[:SOURce]:LIST<n>:FREQuency <numeric_value> {HZ | KHZ | MHZ | GHZ}, {<numeric_value> {HZ | KHZ | MHZ | GHZ}}, ...

[:SOURce]:LIST<n>:FREQuency? [<numeric_value>],[<numeric_value>]

Title: List Frequency

Description: Sets the list frequencies of the selected list starting at the list index specified by :LIST<n>:INDex or at list suffix 1 if no list suffix is specified.

There can be two optional parameters included with the command when performing a query.

The first parameter is the beginning index for the requested frequencies.

The second is the ending index for the requested frequencies.

If only the first parameter is included, the frequency at the beginning index will be returned. If both parameters are included, the inclusive frequencies from the beginning to ending index will be returned.

Querying with no optional parameters will return the frequency at the current index of the specified list.

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ}, {<numeric_value> {HZ | KHZ | MHZ | GHZ}}, ...

Query Command Parameter(s): [<numeric_value>],[<numeric_value>]

LIST Suffix Range: 1-10, Default = 1

[:SOURce]:LIST<n>:FREQuency:UPDate <numeric_value>,<numeric_value> {HZ | KHZ | MHZ | GHZ}, {<numeric_value> {HZ | KHZ | MHZ | GHZ}}, ...

Title: Update List Frequencies

Description: Update one or more frequencies in the specified list, starting at the given list index. Unlike :LIST<n>:FREQ this command can be used to modify list data independently of the currently selected index. The first parameter is the starting index in the list where the new frequencies will be applied.

Parameter(s): <numeric_value>,<numeric_value> {HZ | KHZ | MHZ | GHZ}, {<numeric_value> {HZ | KHZ | MHZ | GHZ}}, ...

LIST Suffix Range: 1-10, Default = 1

[:SOURce]:LIST<n>:INDex <numeric_value> | DEFault | MINimum | MAXimum

[:SOURce]:LIST<n>:INDex? [DEFault | MINimum | MAXimum]

Title: List Index

Description: Sets index of selected list.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

LIST Suffix Range: 1-10, Default = 1

Default Value: 0

Range: 0 to 3999

[:SOURce]:LIST<n>:POWer <numeric_value> {<amplitude_units>}, {<numeric_value> {<amplitude_units>}}, ... [:SOURce]:LIST<n>:POWer? [<numeric_value>],[<numeric_value>]

Title: List Power

Description: Sets the list power levels of the selected list beginning at the list index specified by :LIST<n>:INDex or at list suffix 1 if no list suffix <n> is specified.

There can be two optional parameters included with the command when performing a query.

The first parameter is the beginning index for the requested powers.

The second is the ending index for the requested powers.

If only the first parameter is included, the power at the beginning index will be returned. If both parameters are included, the inclusive powers from the beginning to ending index will be returned.

Querying with no optional parameters will return the power at the current index of the specified list.

See UNIT:POWer to query or set the current amplitude units.

Set Command Parameter(s): <numeric_value> {<amplitude_units>}, {<numeric_value> {<amplitude_units>}}, ...

Query Command Parameter(s): [<numeric_value>],[<numeric_value>]

LIST Suffix Range: 1-10, Default = 1

[:SOURce]:LIST<n>:POWer:UPDate <numeric_value>,<numeric_value> {<amplitude_units>}, {<numeric_value> {<amplitude_units>}}, ...

Title: Update List Powers

Description: Update one or more power levels in the specified list starting at the given list index. Unlike :LIST<n>:POW this command can be used to modify list data independently of the currently selected index. The first parameter is the starting index in the list where the new powers will be applied.

See UNIT:POWer to query or set the current amplitude units.

Parameter(s): <numeric_value>,<numeric_value> {<amplitude_units>}, {<numeric_value> {<amplitude_units>}}, ...

LIST Suffix Range: 1-10, Default = 1

[:SOURce]:LIST<n>:PRESet

Title: List Preset

Description: Resets the contents of the chosen list to default values. This will also clear the list data from internal storage.

LIST Suffix Range: 1-10, Default = 1

[:SOURce]:LIST<n>:STARt <numeric_value> | DEFault | MINimum | MAXimum [:SOURce]:LIST<n>:STARt? [DEFault | MINimum | MAXimum]

Title: List Start

Description: Sets list start index for a sweep of the selected list.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

LIST Suffix Range: 1-10, Default = 1

Default Value: 0

Range: 0 to 3999

[:SOURce]:LIST<n>:STOP <numeric_value> | DEFault | MINimum | MAXimum [:SOURce]:LIST<n>:STOP? [DEFault | MINimum | MAXimum]

Title: List Stop

Description: Sets list stop index for a sweep of the selected list.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

LIST Suffix Range: 1-10, Default = 1

Default Value: 3999

Range: 0 to 3999

[:SOURce]:LIST:COPY[:CW]

Title: List Copy CW

Description: Copies the current CW frequency and power settings to the current index of the current list sweep table. The current list is indicated by [:SOURce]:POWer:MODE and [:SOURce]:FREQuency:MODE, or, if no list is selected, the default destination is LIST1. Once the list has been updated, its point index is incremented by one.

[:SOURce]:LIST:DISPlay <numeric_value> | DEFault | MINimum | MAXimum [:SOURce]:LIST:DISPlay? [DEFault | MINimum | MAXimum]

Title: List Display

Description: Selects the list to be displayed on the GUI while in the List Sweep menu.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1

Range: 1 to 10

[:SOURce]:LIST:DWELl <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
[:SOURce]:LIST:DWELl? [DEFault | MINimum | MAXimum]

Title: List Dwell

Description: Sets the dwell time for each step in a sweep.

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 0.05 s

Default Unit: s

Range: 1e-05 s to 99 s

[:SOURce]:LIST:FREQuency:POINts?

Title: List Frequency Points

Description: Queries the number of frequency points in the selected list. The number of points is 4000 and is not settable.

[:SOURce]:LIST:MODE <SWEep|POINt|AUTO|MANual>
[:SOURce]:LIST:MODE?

Title: List Mode

Description: Selects the triggering mode for FREQuency:MODE LIST<n>. In SWEep LIST:MODE, the instrument does a complete sweep upon receiving a trigger.

In POINt LIST:MODE, the instrument will move to the next point in the list.

Parameter(s): <SWEep|POINt|AUTO|MANual>

[:SOURce]:LIST:POWer:POINts?

Title: List Power Points

Description: Queries the number of power level points in the selected list. The number of points is 4000 and is not settable.

[:SOURce]:LIST:PRESet:ALL

Title: List Preset

Description: Resets the contents of all list tables. See [:SOURce]:LIST<n>:PRESet.

[:SOURce]:PM:BWIDth <MINimum|MAXimum>
[:SOURce]:PM:BWIDth?

Title: PM Bandwidth

Description: Selects the phase modulation (ΦM) operating mode.

This SCPI command remains for backward compatibility and is replaced by SOURCE:PM:MODE. The MINimum/MAXimum input enums corresponds to LNOise/WDEViation inputs for PM Mode.

Parameter(s): <MINimum|MAXimum>

[:SOURce]:PM:ERRor?

Title: PM Error

Description: PM error status that can originate from a measurement limit violation or a rate vs. waveform type violation.

PM over range error = 0x1

PM rate exceeded error = 0x2

PM Deviation exceeded = 0x4

Query Return: Numeric

Default Value: 0

Range: 0 to 255

[:SOURce]:PM[:EXTernal]:SENSitivity <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:PM[:EXTernal]:SENSitivity? [DEFault | MINimum | MAXimum]

Title: PM Sensitivity

Description: Sets the ΦM sensitivity for the external phase modulation function. See related [:SOURce]:PM:BWIDth

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1 RAD/V

Default Unit: RAD/V

Range: -3220 RAD/V to 3220 RAD/V

[:SOURce]:PM[:INTernal]:DEViation <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:PM[:INTernal]:DEViation? [DEFault | MINimum | MAXimum]

Title: PM Deviation

Description: Set the modulation deviation of the ΦM signal for the internal phase modulation function.

See related [:SOURce]:PM:BWIDth

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1 RAD

Default Unit: RAD

Range: 1e-05 RAD to 3220 RAD

[:SOURce]:PM[:INTernal]:FREQuency <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:PM[:INTernal]:FREQuency? [DEFault | MINimum | MAXimum]

Title: PM Internal Frequency

Description: Sets the frequency of the modulating waveform for the internal phase modulation.

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1000 Hz

Default Unit: Hz

Range: 0.1 Hz to 10000000 Hz

[:SOURce]:PM[:INTernal]:WAVE <SINE|GAUSsian|RDOWn|RUP|SQUare|TRIangle|UNIForm>
[:SOURce]:PM[:INTernal]:WAVE?

Title: PM Internal Wave

Description: Selects the modulating waveform (from the internal ΦM generator) for the internal phase

modulation function, as follows:

SINE = Sine wave

GAUSsian = Gaussian noise

RDOWn = Negative ramp

RUP = Positive ramp

SQUare = Square wave

TRIangle = Triangle wave

UNIForm = Uniform noise

Parameter(s): <SINE | GAUSsian | RDOWn | RUP | SQUare | TRIangle | UNIForm>

Query Return: SINE | GAUS | RDOW | RUP | SQU | TRI | UNIF

Default Value: SINE

[:SOURce]:PM:MODE <LNOise|WDEViation>
[:SOURce]:PM:MODE?

Title: PM Mode

Description: Selects the phase modulation (ΦM) operating mode.

The Low Noise ΦM mode allows a deviation range of 0.00001 radians to 25.1 radians, and a sensitivity range of ±0.00001 RADS/V to ±25.1 RADS/V. The Wide deviation ΦM mode allows a deviation range of 0.001 radians to 3220 radians, and a sensitivity range of -0.001 RADS/V to -3220 RADS/V OR +0.001 RADS/V to +3220 RADS/V.

Parameter(s): <LNOise | WDEViation>

Query Return: LNO | WDEV

Default Value: LNOise

[:SOURce]:PM:SOURce <INTernal|EXTernal>
[:SOURce]:PM:SOURce?

Title: PM Source

Description: Selects the source of the ΦM modulating signal, as follows:

INTernal selects the Internal ΦM generator.

EXTernal selects the rear panel FM/ΦM IN connector. EXTernal1 and EXTernal2 have the same functionality as EXTernal.

Parameter(s): <INTernal | EXTernal>

Query Return: INT | EXT

Default Value: INTernal

[:SOURce]:PM:STATe <0 | 1 | ON | OFF>
[:SOURce]:PM:STATe?

Title: PM State

Description: Enable/disable phase modulation of the RF output signal.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:POWer:ALC:SEARch:LNOise [<[ONCE]>]

Title: Power Alc Search

Description: Performs a search at the current frequency and power settings for the optimum configuration for the lowest noise floor. No other commands are processed until the search is complete.

Parameter(s): [<[ONCE]>]

[:SOURce]:POWer:ALC:SOURce <INTernal|DIODe|DIODe1|DIODe2|FIXed>
[:SOURce]:POWer:ALC:SOURce?

Title: Power ALC Source

Description: Selects which system has control of instrument power level, and chooses the source of the ALC feedback signal.

FIXed places the instrument in a fixed gain power level mode (ALC off). RF output power is unleveled; use the [:SOURce]:POWer[:LEVel]:DAC command to control instrument output power.

The other parameters turn on the ALC function and select the source of the ALC feedback signal. INTernal specifies the ALC feedback signal from the instrument's internal level detector. The remaining parameter choices select an external detector or power meter signal as the feedback signal source for the ALC, as follows: DIODe = Detector output connected to the rear panel EXTERNAL ALC input. DIODe1 and DIODe2 exist for backwards compatibility and have the same effect as DIODe.

Parameter(s): <INTernal|DIODe|DIODe1|DIODe2|FIXed>

Query Return: INT|DIOD|DIOD1|DIOD2|FIX

Default Value: INTernal

[:SOURce]:POWer:ATTenuation <numeric_value> {DB}
[:SOURce]:POWer:ATTenuation? [<DEFault|MINimum|MAXimum>]

Title: Power Attenuation

Description: This command applies only to devices equipped with an internal Step Attenuator option. This command sets the step attenuator in 10 dB increments. This command decouples the step attenuator from the automatic leveling control (ALC) system (see the command :POWer:ATTenuation:AUTO OFF).

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (dB)

Set Command Parameter(s): <numeric_value> {DB}

Query Command Parameter(s): [<DEFault|MINimum|MAXimum>]

Default Value: 0 dB

Default Unit: dB

Range: 0 dB to 110 dB

[:SOURce]:POWer:ATTenuation:AUTO <0 | 1 | ON | OFF>
[:SOURce]:POWer:ATTenuation:AUTO?

Title: Power Attenuation State

Description: This command applies only to devices equipped with an internal Step Attenuator option. Setting to ON couples the step attenuator to the ALC system; setting to OFF decouples the step attenuator from the ALC system.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

[:SOURce]:POWer:CENTer <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum
[:SOURce]:POWer:CENTer? [DEFault | MINimum | MAXimum]

Title:	Power Center
Description:	Sets the center power for SWEep2. Center and span power level are coupled values. Entering the value for one will cause the other to be recalculated. Setting the center power level may affect sweep parameters start, stop, span, step, and points. Related commands: [:SOURce]:POWer:MODE, [:SOURce]:POWer:STARt, [:SOURce]:POWer:STOP, [:SOURce]:POWer:SPAN, [:SOURce]:SWEep2:STEP, [:SOURce]:SWEep2:POINTs See UNIT:POWer to query or set the current amplitude units.
Query Return:	Numeric (<amplitude_units>)
Set Command Parameter(s):	<numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum
Query Command Parameter(s):	[DEFault | MINimum | MAXimum]
Default Value:	-48.5 dBm
Default Unit:	dBm
Range:	-129.995 dBm to 32.995 dBm

[:SOURce]:POWer:DISPlay:OFFSet <numeric_value> {DB} | DEFault | MINimum | MAXimum
[:SOURce]:POWer:DISPlay:OFFSet? [DEFault | MINimum | MAXimum]

Title:	Power Display Offset
Description:	Sets the offset value for the power level display offset function (see :POWer:DISPlay:OFFSet:STATe). See UNIT:POWer to query or set the current amplitude units.
Query Return:	Numeric (dB)
Set Command Parameter(s):	<numeric_value> {DB} | DEFault | MINimum | MAXimum
Query Command Parameter(s):	[DEFault | MINimum | MAXimum]
Default Value:	0 dB
Default Unit:	dB
Range:	-100 dB to 100 dB

[:SOURce]:POWer:DISPlay:OFFSet:STATe <0 | 1 | ON | OFF>
[:SOURce]:POWer:DISPlay:OFFSet:STATe?

Title: Power Display Offset State

Description: Turns the power level display offset function on/off. When the function it turned on, the offset value, set by the command :POWer:DISPlay:OFFSet <arg>, is added to the displayed RF output power level. A negative offset value decreases the displayed power level.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:POWer[:LEVel]:DAC <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:POWer[:LEVel]:DAC? [DEFault | MINimum | MAXimum]

Title: Level DAC

Description: Manual setting of the Level DAC.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 0

Range: 0 to 65535

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum | UP | DOWN
[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]? [DEFault | MINimum | MAXimum | UP | DOWN]

Title: Power Amplitude

Description: This is the power level/amplitude setting/command. It has a SCPI command associated with it ([:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] <nv> [<term>]) as well as a setting class, which is autogenerated from the other qml information.

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (<amplitude_units>)

Set Command Parameter(s): <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum | UP | DOWN

Query Command Parameter(s): [DEFault | MINimum | MAXimum | UP | DOWN]

Default Value: 0 dBm

Default Unit: dBm

Range: -130 dBm to 33 dBm

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:L<n> <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:L<n>? [DEFault | MINimum | MAXimum]

Title: Preset Power Value

Description: Edits a selected preset power level value.

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (<amplitude_units>)

Set Command Parameter(s): <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

L Suffix Range: 0-9, Default = 0

Default Value: 0 dBm

Default Unit: dBm

Range: -130 dBm to 33 dBm

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:SELect <L0|L1|L2|L3|L4|L5|L6|L7|L8|L9> [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:SELect?

Title: Preset Power Select

Description: Selects a Preset Power Index.

Parameter(s): <L0|L1|L2|L3|L4|L5|L6|L7|L8|L9>

Query Return: L0|L1|L2|L3|L4|L5|L6|L7|L8|L9

Default Value: L1

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement] <numeric_value> {DB} | DEFault | MINimum | MAXimum [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:STEP[:INCRement]? [DEFault | MINimum | MAXimum]

Title: Power Step

Description: Sets the step increment size used with the :POWer:LEVel:IMMediate:AMPLitude command.

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (dB)

Set Command Parameter(s): <numeric_value> {DB} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 0.1 dB

Default Unit: dB

Range: 0.01 dB to 10 dB

[:SOURce]:POWer[:LEVel]:RANGe <HIGH|LOW>
[:SOURce]:POWer[:LEVel]:RANGe?

Title: Manual ALC Range

Description: Manual setting of the ALC Range for External Closed and Open Fixed Loops. This setting is used to configure particular ALC HW for either a High or Low user requested ALC range configuration.

Parameter(s): <HIGH|LOW>

Query Return: HIGH|LOW

Default Value: HIGH

[:SOURce]:POWer:LIMit[:AMPLitude] <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum | UP | DOWN
[:SOURce]:POWer:LIMit[:AMPLitude]? [DEFault | MINimum | MAXimum | UP | DOWN]

Title: Power Limit

Description: Sets the maximum power level when level limit state is enabled.

Related command: [SOURce:]POWer:LIMit:STATe.

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (<amplitude_units>)

Set Command Parameter(s): <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum | UP | DOWN

Query Command Parameter(s): [DEFault | MINimum | MAXimum | UP | DOWN]

Default Value: 30 dBm

Default Unit: dBm

Range: -130 dBm to 33 dBm

[:SOURce]:POWer:LIMit:STATe <0 | 1 | ON | OFF>
[:SOURce]:POWer:LIMit:STATe?

Title: Power Limit State

Description: Enable/disable power limit of the RF output signal.

When power limit is enabled the maximum power level is clamped to the power level limit value.

Related command: [SOURce:]POWer:LIMit[:AMPLitude].

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:POWer:MODE <CW|FIXed|SWEep{2}|LIST{[1]}|LIST{2}|LIST{3}|LIST{4}|LIST{5}|LIST{6}|LIST{7}|LIST{8}|LIST{9}|LIST{10}>
[:SOURce]:POWer:MODE?

Title: Power Mode

Description: Selects the method of controlling RF output power.

Below is a list of available modes, as well as the command or branch that controls output while in that mode, followed by the text displayed on the GUI for that mode.

CW|FIXed = [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] - CW

SWEep2 = [:SOURce]:SWEep2 - Level Sweep

LIST<n> = [:SOURce]:LIST<n> - List Sweep

In CW/Fixed mode, power output is controlled by the GUI's main Level dialog, or by the [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] command.

In SWEep2 (Level Sweep), power output is controlled programmatically by the level sweep subsystem [:SOURce]:SWEep2 with commands like [:SOURce]:SWEep2:POINts, [:SOURce]:SWEep2:STEP.

In LIST mode, power output is controlled by the list sweep subsystem, which programmatically controls both frequency and power output. See [:SOURce]:LIST<n> commands for more details. List frequencies and powers are deleted on Factory Reset, User File Reset, and Reset All.

Related commands:

[:SOURce]:POWer:CENTer, [:SOURce]:POWer:SPAN,

[:SOURce]:POWer:STARt, [:SOURce]:POWer:STOP,

[:SOURce]:LIST<n>:POWer, [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]

Parameter(s):

<CW|FIXed|SWEep{2}|LIST{[1]}|LIST{2}|LIST{3}|LIST{4}|LIST{5}|LIST{6}|LIST{7}|LIST{8}|LIST{9}|LIST{10}>

Query Return:

CW|FIX|SWE2|SWELIST1|LIST2|LISTLIST3|LIST4|LISTLIST5|LIST6|LISTLIST7|LIST8|LISTLIST9

Default Value: FIXed

[:SOURce]:POWer:PROTection:CLEar

Title: Power Protection Clear

Description: Clears the power protection tripped status.

[:SOURce]:POWer:PROTection:TRIPped?

Title: Power Protection Tripped

Description: Queries the status of a power trip.

0: power protection trip has not occurred.

1: power protection trip has occurred.

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:POWer:SLOPe <numeric_value> | DEFault | MINimum | MAXimum | UP | DOWN
[:SOURce]:POWer:SLOPe? [DEFault | MINimum | MAXimum | UP | DOWN]

Title: Power Slope

Description: Sets the value of the slope characteristic parameter for the ALC power slope function. This parameter is a relative number that defines the slope characteristic; a value of 128 produces a 0 dB/V slope. Parameters UP | DOWN increment/decrement the characteristic parameter by the value set by the [:SOURce]:POWer:SLOPe:STEP[:INCRement] command.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum | UP | DOWN

Query Command Parameter(s): [DEFault | MINimum | MAXimum | UP | DOWN]

Default Value: 128

Range: 0 to 255

[:SOURce]:POWer:SLOPe:STATe <0 | 1 | ON | OFF>
[:SOURce]:POWer:SLOPe:STATe?

Title: Power Slope State

Description: Turns ALC power slope function on/off.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:POWer:SLOPe:STEP[:INCRement] <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:POWer:SLOPe:STEP[:INCRement]? [DEFault | MINimum | MAXimum]

Title: Power Slope Step

Description: Sets the step increment size used with the :POWer:SLOPe command (ALC power slope function).

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1

Range: 0 to 255

[:SOURce]:POWer:SPAN <numeric_value> {DB} | DEFault | MINimum | MAXimum [:SOURce]:POWer:SPAN? [DEFault | MINimum | MAXimum]

Title: Power Span

Description: Sets the power span for SWEep2. Center and span power level are coupled values.

Entering the value for one will cause the other to be recalculated.

Setting the span power level may affect sweep parameters start, stop, center, step, and points.

Related commands:

[:SOURce]:POWer:MODE, [:SOURce]:POWer:STARt,

[:SOURce]:POWer:STOP, [:SOURce]:POWer:CENTer,

[:SOURce]:SWEep2:STEP, [:SOURce]:SWEep2:POINTs

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (dB)

Set Command Parameter(s): <numeric_value> {DB} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 163 dB

Default Unit: dB

Range: 0.01 dB to 163 dB

[:SOURce]:POWer:SPAN:FULL

Title: Power Span Full

Description: Sets the power span for SWEep2 to the maximum span (MAX power level – MIN power level).

Setting the power span to full may affect sweep parameters start, stop, center, span, step size, and points.

Related commands:

[:SOURce]:POWer:SPAN, [:SOURce]:POWer:STARt,

[:SOURce]:POWer:STOP, [:SOURce]:POWer:CENTer,

[:SOURce]:SWEep2:STEP, [:SOURce]:SWEep2:POINTs

[:SOURce]:POWer:STARt <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum
[:SOURce]:POWer:STARt? [DEFault | MINimum | MAXimum]

Title: Power Start

Description: Sets the start power for SWEep2.

Setting the start power may affect sweep parameters stop, center, span, step, and points.

Related commands:

[:SOURce]:POWer:MODE, [:SOURce]:POWer:STOP,

[:SOURce]:POWer:CENTer, [:SOURce]:POWer:SPAN,

[:SOURce]:SWEep2:STEP, [:SOURce]:SWEep2:POINTs

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (<amplitude_units>)

Set Command Parameter(s): <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: -130 dBm

Default Unit: dBm

Range: -130 dBm to 32.99 dBm

[:SOURce]:POWer:STOP <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum
[:SOURce]:POWer:STOP? [DEFault | MINimum | MAXimum]

Title: Power Start

Description: Sets the stop power for SWEep2.

Setting the stop power may affect sweep parameters start, center, span, step, and points.

Related commands:

[:SOURce]:POWer:MODE, [:SOURce]:POWer:STARt,

[:SOURce]:POWer:CENTer, [:SOURce]:POWer:SPAN,

[:SOURce]:SWEep2:STEP, [:SOURce]:SWEep2:POINTs

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (<amplitude_units>)

Set Command Parameter(s): <numeric_value> {<amplitude_units>} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 33 dBm

Default Unit: dBm

Range: -129.99 dBm to 33 dBm

[:SOURce]:PULM:ERRor? [DEFault | MINimum | MAXimum]

Title:	PULM Error
Description:	Pulsed Modulation error status. Pulse Usable Width Exceeded = 0x8
Query Return:	Numeric
Default Value:	0
Range:	0 to 255

[:SOURce]:PULM[:INTernal]:FREQuency <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
[:SOURce]:PULM[:INTernal]:FREQuency? [DEFault | MINimum | MAXimum]

Title:	Pulm Internal Frequency
Description:	Sets the pulse repetition frequency (PRF) of the internal pulse generator. The PRF range is determined by the pulse generator clock rate.
Query Return:	Numeric (Hz)
Set Command Parameter(s):	<numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum
Query Command Parameter(s):	[DEFault | MINimum | MAXimum]
Default Value:	1000 Hz
Default Unit:	Hz
Range:	0.597 Hz to 25000000 Hz

[:SOURce]:PULM:POLarity <NORMal|INVerted>
[:SOURce]:PULM:POLarity?

Title:	Pulm Polarity
Description:	Selects the polarity of the signal that turns the RF on during pulse modulation, as follows: NORMAL specifies positive-true operation; a TTL-high level will turn on the RF output signal. INVerted specifies negative-true operation; a TTL-low level will turn on the RF output signal.
Parameter(s):	<NORMal|INVerted>
Query Return:	NORM|INV
Default Value:	NORMal

[:SOURce]:PULM:SOURce <INTernal|EXTernal>
[:SOURce]:PULM:SOURce?

Title: Pulm Source

Description: Selects the pulse modulation signal source, as follows:

INTernal selects the signal from the internal pulse generator. INTernal1 and INTernal2 have the same functionality as INTernal.

EXTernal selects the rear panel external pulse input. EXTernal1 and EXTernal2 have the same functionality as EXTernal.

Parameter(s): <INTernal|EXTernal>

Query Return: INT|EXT

Default Value: INTernal

[:SOURce]:PULM:STATe <0 | 1 | ON | OFF>
[:SOURce]:PULM:STATe?

Title: Pulm State

Description: Turns the pulse modulation function on/off.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

[:SOURce]:PULSe:COUNt <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:PULSe:COUNt? [DEFault | MINimum | MAXimum]

Title: Pulse Count

Description: Sets the number of pulses generated by the internal pulse generator for each period of the pulsed waveform.

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 1

Range: 1 to 4

[:SOURce]:PULSe:DELay<n> <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
[:SOURce]:PULSe:DELay<n>? [DEFault | MINimum | MAXimum]

Title: Pulse Delay

Description: Sets the pulse delay for the selected pulse. The pulse delay range for each pulse is determined by the pulse generator clock rate.

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

DELay Suffix Range: 1-4, Default = 1

Default Value: 0.0002 s

Default Unit: s

Range: 0 s to 42.94967295 s

[:SOURce]:PULSe:PERiod <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
[:SOURce]:PULSe:PERiod? [DEFault | MINimum | MAXimum]

Title: Pulse Period

Description: Sets the pulse period of the internal pulse generator. The pulse period range is determined by the pulse generator clock rate.

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 0.001 s

Default Unit: s

Range: 1e-07 s to 42.94967295 s

[:SOURce]:PULSe:WIDTh<n> <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
[:SOURce]:PULSe:WIDTh<n>? [DEFault | MINimum | MAXimum]

Title: Pulse Width

Description: Sets the pulse width for the selected pulse. The pulse width range is determined by the pulse generator clock rate.

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

WIDTh Suffix Range: 1-4, Default = 1

Default Value: 0.0001 s

Default Unit: s

Range: 1e-08 s to 42.94967295 s

[:SOURce]:ROSCillator:GPS:LOCK?

Title: Reference Oscillator GPS Lock Status

Description: Returns the GPS reference 1PPS lock status. Query only. Requires GNSS/GPS Atomic Clock Receiver option.

LOCKed = The GPS reference is locked

ACQuiring = The GPS reference is acquiring lock

OFF = The GPS module in turned off

ERRor = The GPS reference is unable to acquire lock

Query Return: ACQ|LOCK|ERR|OFF

Default Value: ACQuiring

[:SOURce]:ROSCillator:HFRequency:BWIDth <WIDE|NARRow>
[:SOURce]:ROSCillator:HFRequency:BWIDth?

Title: 100 MHz Reference Bandwidth

Description: Toggle the 100 MHz reference bandwidth between wide and narrow.

WIDE = Low Offset Phase Noise, nominal, optimize for < 300 Hz

NARRow = High Offset Phase Noise, optimize for > 300 Hz

Parameter(s): <WIDE|NARRow>

Query Return: WIDE|NARR

Default Value: NARRow

[:SOURce]:ROSCillator:HFRequency:LOCK?

Title: Reference Oscillator 100 MHz Lock Status

Description: Returns the 100 MHz reference lock status. Query only.

ACQuiring = Attempting to lock the 100 MHz to an external 10 MHz reference (Low Phase Noise and Ultra High Stability option is not installed) OR Attempting to lock the 100 MHz to the internal 10 MHz reference (Low Phase Noise and Ultra High Stability option is installed)

LOCKed = 100 MHz is locked to the external 10 MHz (Low Phase Noise and Ultra High Stability option is not installed) OR 100 MHz is locked to the internal 10 MHz reference (Low Phase Noise and Ultra High Stability option required)

FREE = 100 MHz is freerunning

ERRor = 100 MHz is unable to lock to a connected 10 MHz (Low Phase Noise and Ultra High Stability option is not installed) OR 100 MHz is unable to lock to the internal 10 MHz reference (Low Phase Noise and Ultra High Stability option is installed)

Query Return: ACQ|LOCK|FREE|ERR

Default Value: FREE

[:SOURce]:ROSCillator:HFRequency:SOURce?

Title: 100 MHz Reference Source

Description: Returns the 100 MHz reference source. Query only.

INTernal = internal 100 MHz

EXTernal = external 100 MHz input (Low Phase Noise and Ultra High Stability option required) OR external 10 MHz input (Low Phase Noise and Ultra High Stability option not installed)

Query Return: INT|EXT

Default Value: INTernal

[:SOURce]:ROSCillator:LFRequency:SOURce?

Title: 10 MHz Reference Source

Description: Returns the 10 MHz reference source. Query only. Requires Low Phase Noise and Ultra High Stability option.

Note: When GNSS/GPS Atomic Clock Receiver option is installed

and SYSTem:GPS:STATe is ON, the source is always INTernal.

INTernal = internal 10 MHz free running

IRUBidium = internal 10 MHz Rubidium (Ultra Stable Rubidium Reference option required)

EXTernal = external 10 MHz input

EPPS = external 1PPS input (Ultra Stable Rubidium Reference option or GNSS/GPS Atomic Clock Receiver option required)

Query Return: INT|EXT|IRUB|EPPS|RPPS

Default Value: INTernal

[:SOURce]:ROSCillator:LOCK?

Title: Reference Oscillator Lock Status

Description: Returns the 10 MHz reference lock status if Low Phase Noise and Ultra High Stability option is installed. Returns the 100 MHz reference lock status if Low Phase Noise and Ultra High Stability option is NOT installed (standard reference). Query only. Possible values vary based on installed options.

ACQuiring = Attempting to acquire lock to the reference source indicated by ROSC:LFR:SOUR (Low Phase Noise and Ultra High Stability option is installed) OR Attempting to acquire lock to an external 10 MHz reference (Low Phase Noise and Ultra High Stability option is not installed)

LEXTernal = 100 MHz reference is locked to an external 10 MHz reference (Low Phase Noise and Ultra High Stability option is not installed) OR 10 MHz reference is locked to an external reference, either 10 MHz or 1PPS (Low Phase Noise and Ultra High Stability option required)

LRUBidium = 10 MHz reference is locked to an internal Rb reference (Ultra Stable Rubidium Reference option required)

FREE = 100 MHz reference is free running (Low Phase Noise and Ultra High Stability option is not installed) OR 10 MHz reference is free running (Low Phase Noise and Ultra High Stability option required)

EEXTernal = 100 MHz reference can't lock to a connected external 10 MHz input (Low Phase Noise and Ultra High Stability option not installed) OR 10 MHz reference can't lock to a connected external reference, either 10 MHz or 1PPS (Low Phase Noise and Ultra High Stability option required)

ERUBidium = 10 MHz reference can't lock to the internal Rb reference (Ultra Stable Rubidium Reference option required)

ECOLd = Reference is still cold (used during initialization)

Query Return: ACQ|LEXT|FREE|EEXT|LRUB|ERUB|ECOL

Default Value: FREE

[:SOURce]:ROSCillator:MULTiplied:SOURce?

Title: 1600 MHz Reference Source

Description: Returns the 1600 MHz multiplied reference source. Query only. Requires Low Phase Noise and Ultra High Stability option.

INTernal = internal 1600 MHz

EXTernal = external 1600 MHz input

Query Return: INT|EXT

Default Value: INTernal

[:SOURce]:ROSCillator:RUBidium:LOCK?

Title: Reference Oscillator Rubidium Lock Status

Description: Returns the Rubidium reference lock status. Query only. Requires Low Phase Noise and Ultra High Stability option and Ultra Stable Rubidium Reference option.

LOCKed = The Rb internal TCXO is locked to the vapor chamber

ACQuiring = The Rb internal TCXO is acquiring lock to the vapor chamber

OFF = The Rb module is turned off (see :SOURce:ROSCillator:RUBidium:STATe)

ERRor = The Rb internal TCXO is unable to acquire lock to the vapor chamber

Query Return: ACQ|LOCK|ERR|OFF

Default Value: ACQuiring

[:SOURce]:ROSCillator:RUBidium:STATe <0 | 1 | ON | OFF>
[:SOURce]:ROSCillator:RUBidium:STATe?

Title: Rubidium State

Description: Enable or disable the Rubidium reference module
(Ultra Stable Rubidium Reference option required).
When ON, the Rubidium module will be powered on,
and used as the active 10 MHz reference (provided
an external 10 MHz or 1PPS, which has higher priority,
is not connected). When OFF, the Rubidium module will be
powered off, and the 10 MHz OCXO will be left freerunning
in the absence of an external reference. This command makes
it possible to freerun the 10 MHz OCXO when the Rubidium
option is installed, as otherwise the Rubidium 10 MHz will
always be used in the absence of an external reference. The main use
case for turning the Rubidium state OFF is to utilize
the higher accuracy of the freerunning 10 MHz OCXO when it
has been calibrated using the GPS/GNSS receiver
(GNSS/GPS Atomic Clock Receiver option required).

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: ON

[:SOURce]:ROSCillator:SOURce?

Title: Reference Source

Description: Returns the state of the base system reference, internal or external. This command only returns INT or EXT for backwards compatibility to previous products. The state of each reference can be determined by other ROSCillator commands.

Query Return: INT|EXT

Default Value: INTernal

[:SOURce]:ROSCillator:SOURce:CURRent?

Title: Reference Oscillator Current Source

Description: Based on all current connected sources, will return the source that is determining the RF output accuracy.

Displayed by the GUI Title Bar Indicator.

Possible values are as follows, in order of highest to lowest priority:

"REF EXT 1600" (1600 MHz external)

"REF EXT 100" (100 MHz external)

"REF EXT 1PPS" (1PPS external, GNSS/GPS Atomic Clock Receiver option or Ultra Stable Rubidium Reference option)

"REF EXT 10" (10 MHz external)

"REF INT RB" (internal 10 MHz rubidium clock, Ultra Stable Rubidium Reference option only))

"REF INT 10" (internal 10 MHz freerunning, Low Phase Noise and Ultra High Stability option only)

"REF INT 100" (internal 100 MHz free running)

Note: For EXT 1PPS, EXT 10, or INT RB, the value may contain 'ACQ' instead

of 'REF' while the system is in the process of acquiring lock to this reference.

Default Value: REF INT 100

[:SOURce]:SWEep<n>:DIRection <UP|DOWN>
[:SOURce]:SWEep<n>:DIRection?

Title: Sweep Direction

Description: Selects the direction of a stepped sweep for sweep1 or sweep2. Sweep1 signifies frequency sweep; sweep2 signifies power sweep.

Parameter(s): <UP|DOWN>

Query Return: UP|DOWN

SWEep Suffix Range: 1-2, Default = 1

Default Value: UP

[:SOURce]:SWEep<n>:DWELl <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
[:SOURce]:SWEep<n>:DWELl? [DEFault | MINimum | MAXimum]

Title: Sweep Dwell

Description: Sets the dwell time for each step in a stepped frequency sweep or power level sweep. The value entered for dwell time cannot be less than :TIME/:POINts. Setting the sweep dwell may affect sweep parameters time, dwell auto, and time auto. Related commands:

[:SOURce]:SWEep1:GENeration, [:SOURce]:SWEep<n>:POINts, [:SOURce]:SWEep<n>:TIME, [:SOURce]:SWEep<n>:DWELl:AUTO, [:SOURce]:SWEep<n>:TIME:AUTO, [:SOURce]:SWEep<n>:TIME:LLIMit

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

SWEep Suffix Range: 1-2, Default = 1

Default Value: 0.001 s

Default Unit: s

Range: 1e-05 s to 100 s

[:SOURce]:SWEep<n>:DWELl:AUTO?

Title: Sweep Dwell Auto

Description: ON signifies that :DWELl ≈ :TIME/:POINts. Note: :DWELl = (:TIME / :POINts – internal settling time) :DWELl, :SPAN, :TIME, :STEP, and :POINts are all interrelated. Entering too large a value for :DWELl may invalidate the setting for :TIME. Related commands:

[:SOURce]:SWEep1:GENeration, [:SOURce]:SWEep<n>:POINts, [:SOURce]:SWEep<n>:DWELl, [:SOURce]:SWEep<n>:TIME, [:SOURce]:SWEep<n>:TIME:AUTO, [:SOURce]:SWEep<n>:TIME:LLIMit

Query Return: 0 | 1

SWEep Suffix Range: 1-2, Default = 1

Default Value: OFF

[:SOURce]:SWEep<n>:ERRor? [DEFault | MINimum | MAXimum]

Title: Sweep Range Errors

Description: Returns a bitflag containing all current sweep error codes for the selected sweep mode. The relevant sweep mode

must be enabled for this value to be anything other than 0. The errors for each bit number

are as follows:

1 - Span mismatched (sweep span is out of sync with start/stop)

2 - Off-centered (sweep center value is out of sync with start/stop)

3 - Span too large (sweep span is calculated to be larger than start/stop distance)

4 - Span too small (sweep span is calculated to be larger than start/stop distance)

5 - Step mismatch (sweep step size does not match the number of points and/or the span)

6 - Points mismatch (number of points not possible with current span and allowed dB resolution)

7 - Step too large (step size exceeds the distance between start and stop)

8 - Dwell mismatch (sweep dwell does not match the number of points and sweep time)

Query Return: Numeric

SWEep Suffix Range: 1-2, Default = 1

Default Value: 0

Range: 0 to 255

[:SOURce]:SWEep<n>:GENeration <ANALog|STEPped>
[:SOURce]:SWEep<n>:GENeration?

Title: Sweep Generation

Description: This command is only valid for sweep1, frequency sweep.

Selects the sweep1 frequency sweep type.

Analog sweep is only available if the analog sweep option is installed.

Sweep2, power sweep is always stepped.

Related commands:

[:SOURce]:FREQuency:MODE, [:SOURce]:POWer:MODE, [:SOURce]:SWEep<n>:DWELl, [:SOURce]:SWEep<n>:TIME, [:SOURce]:SWEep<n>:POINts, [:SOURce]:SWEep<n>:DWELl:AUTO, [:SOURce]:SWEep<n>:TIME:AUTO, [:SOURce]:SWEep<n>:TIME:LLIMit

Parameter(s): <ANALog|STEPped>

Query Return: ANAL|STEP

SWEep Suffix Range: 1-2, Default = 1

Default Value: STEPped

[:SOURce]:SWEep<n>:POINts <numeric_value> | DEFault | MINimum | MAXimum
[:SOURce]:SWEep<n>:POINts? [DEFault | MINimum | MAXimum]

Title: Sweep Points

Description: Sets the number of points in a stepped frequency sweep or stepped power sweep. Points and step are coupled values. Entering the value for one will cause the other to be recalculated, per the following formula: points = (span / step) + 1. Entering a new value for either parameter will not change the span. An error will be generated if the points entered results in a calculated step size less than the minimum resolution for a stepped frequency sweep or a stepped power level sweep. An error will also be generated if the entered step size results in the number of points exceeding its maximum. Note: The absolute maximum number of frequency points is 65535, and the absolute maximum of power points is calculated using: absMax = (([:SOURce]:POWer? MAX - [:SOURce]:POWer? MIN) / [:SOURce]:POWer:SPAN? MIN) + 1 Related commands:

[:SOURce]:FREQuency:SPAN, [:SOURce]:POWer:SPAN, [:SOURce]:SWEep<n>:STEP

Query Return: Numeric

Set Command Parameter(s): <numeric_value> | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

SWEep Suffix Range: 1-2, Default = 1

Default Value: 1000

Range: 2 to 65535

[:SOURce]:SWEep<n>:RANGe <F1F2|F3F4|F5DF|F6DF|L1L2|L3L4|L5L6|L7L8|L9L0>

Title: Sweep Range Select

Description: SWEep1: Select from 4 frequency ranges, F1-F2, F3-F4, F5-DF, F6-DF, where DF represents delta F. Input without '-' i.e. SWEep1:RANGe F1F2. SWEep2: select from 5 power ranges, L1-L2, L3-L4, L5-L6, L7-L8, L9-L0. Input without '-' i.e. SWEep2:RANGe L1L2. Setting a sweep range may affect sweep parameters start, stop, center, span, step size, and points.

When applying a Frequency Range Selection which utilizes the Delta Frequency (F5-DF, F6-DF), if the Delta Frequency is configured to a value which would result in a Start Frequency less than the minimum CW Frequency or a Stop Frequency greater than the maximum CW frequency, an execution error will be reported.

Related commands:

[:SOURce]:FREQuency:STARt, [:SOURce]:FREQuency:STOP,

[:SOURce]:FREQuency:CENTer, [:SOURce]:FREQuency:SPAN,

[:SOURce]:SWEep1:STEP, [:SOURce]:SWEep1:POINTs,

[:SOURce]:FREQuency:DELTa

[:SOURce]:POWer:STARt, [:SOURce]:POWer:STOP,

[:SOURce]:POWer:CENTer, [:SOURce]:POWer:SPAN,

[:SOURce]:SWEep2:STEP, [:SOURce]:SWEep2:POINTs

Parameter(s): <F1F2|F3F4|F5DF|F6DF|L1L2|L3L4|L5L6|L7L8|L9L0>

Query Return: F1F2|F3F4|F5DF|F6DF|L1L2|L3L4|L5L6|L7L8|L9L0

SWEep Suffix Range: 1-2, Default = 1

Default Value: F1F2

[:SOURce]:SWEep<n>:TIME <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum
[:SOURce]:SWEep<n>:TIME? [DEFault | MINimum | MAXimum]

Title: Sweep Time

Description: Sets the sweep time for sweep1 or sweep2. When :SWEep<n>:TIME <param> is implemented, :SWEep<n>:TIME:AUTO is set to OFF. :TIME, :DWELl, :SPAN, :STEP, and :POINts are all interrelated. Entering too large a value for any of the four other parameters may invalidate the entered value for :TIME. Setting the sweep time may affect sweep parameters dwell, dwell auto, and time auto. Related commands:

[:SOURce]:SWEep1:GENeration, [:SOURce]:SWEep1:POINts, [:SOURce]:SWEep1:DWELl, [:SOURce]:SWEep1:DWELl:AUTO, [:SOURce]:SWEep1:TIME:AUTO, [:SOURce]:SWEep1:TIME:LLIMit

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

SWEep Suffix Range: 1-2, Default = 1

Default Value: 8 s

Default Unit: s

Range: 0.00102 s to 6553906.317 s

[:SOURce]:SWEep<n>:TIME:AUTO?

Title: Sweep Time Auto

Description: ON specifies that the sweep time for the associated sweep (sweep1 or sweep2) is to be calculated internally and is dependent on the sweep span value. Related commands:

[:SOURce]:SWEep1:GENeration, [:SOURce]:SWEep1:POINts, [:SOURce]:SWEep1:DWELl, [:SOURce]:SWEep1:DWELl:AUTO, [:SOURce]:SWEep1:TIME, [:SOURce]:SWEep1:TIME:LLIMit

Query Return: 0 | 1

SWEep Suffix Range: 1-2, Default = 1

Default Value: OFF

[:SOURce]:SWEep<n>:TIME:LLIMit <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum [:SOURce]:SWEep<n>:TIME:LLIMit? [DEFault | MINimum | MAXimum]

Title: Sweep Time Lower Limit

Description: Sets the lower limit for :SWEep<n>:TIME. No sweep time will be shorter than this value. When :SWEep<n>:TIME:AUTO is set ON, the internally computed sweep time will not be smaller than the sweep time set with the :SWEep<n>:Time:LLIMit command. :SWEep2:TIME:LLIMit will be set to the same value as :SWEep:TIME:LLImit and vice versa. Setting the sweep timer lower limit may affect sweep parameters dwell, time, dwell auto, and time auto. Related commands:

[:SOURce]:SWEep1:GENeration, [:SOURce]:SWEep1:POINts, [:SOURce]:SWEep1:DWELl, [:SOURce]:SWEep1:DWELl:AUTO, [:SOURce]:SWEep1:TIME, [:SOURce]:SWEep1:TIME:AUTO

Query Return: Numeric (s)

Set Command Parameter(s): <numeric_value> {PS | NS | US | MS | S | MIN | HR} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

SWEep Suffix Range: 1-2, Default = 1

Default Value: 0.002 s

Default Unit: s

Range: 0.002 s to 98.998 s

[:SOURce]:SWEep[:FREQuency]:STEP <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum [:SOURce]:SWEep[:FREQuency]:STEP? [DEFault | MINimum | MAXimum]

Title: Sweep Frequency Step

Description: Sets the step size for SWEep[1] linear stepped sweep. Points and step are coupled values. Entering the value for one will cause the other to be recalculated; see notes under :SWEep:POINts command. The maximum frequency sweep step size is equal to the maximum frequency span for the particular model / (minimum points – 1). An error will be generated if the step size exceeds the span. Related commands:

[:SOURce]:FREQuency:SPAN, [:SOURce]:SWEep1:POINTs

Query Return: Numeric (Hz)

Set Command Parameter(s): <numeric_value> {HZ | KHZ | MHZ | GHZ} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

Default Value: 80080071.0710711 Hz

Default Unit: Hz

Range: 0.001 Hz to 79999991000 Hz

[:SOURce]:SWEep<n>:POWer:STEP <numeric_value> {DB} | DEFault | MINimum | MAXimum
[:SOURce]:SWEep<n>:POWer:STEP? [DEFault | MINimum | MAXimum]

Title: Sweep Power Step

Description: Sets the step size for each step in a power sweep. Points and step are coupled values. Entering the value for one will cause the other to be recalculated. An error will be generated if the step size exceeds the span. Related commands:

[:SOURce]:POWer:SPAN, [:SOURce]:SWEep2:POINTs

See UNIT:POWer to query or set the current amplitude units.

Query Return: Numeric (dB)

Set Command Parameter(s): <numeric_value> {DB} | DEFault | MINimum | MAXimum

Query Command Parameter(s): [DEFault | MINimum | MAXimum]

SWEep Suffix Range: 2, Default = 2

Default Value: 0.163163163163163 dB

Default Unit: dB

Range: 0.01 dB to 163 dB

[:SOURce]:SWEep:SPACing <LINear|LOGarithmic>
[:SOURce]:SWEep:SPACing?

Title: Sweep Spacing

Description: Selects the type of SWEep1 stepped frequency sweep. In LINear stepped sweep mode, the frequency sweep is linearly incremented by the step size from the start frequency to the stop frequency. The size of each step of the linear stepped sweep can be set using either the :SWEep[:FREQuency]:STEP command or the :SWEep:POINts command. In LOGarithmic stepped sweep mode, step size increases logarithmically with frequency and is determined by a logarithmic curve fitted between the sweep start and stop frequencies. The number of steps is set using the command :SWEep:POINts. Related commands:

[:SOURce]:SWEep1:STEP, [:SOURce]:SWEep1:POINTs

Parameter(s): <LINear|LOGarithmic>

Query Return: LIN|LOG

Default Value: LINear

:SYSTem:LANGuage
:SYSTem:LANGuage?

Title: System Language

Description: Selects the instrument's external interface language. The double quotes are required and will be returned with the query reply.

"NATIVE" changes the instrument's external interface language to the Anritsu Native language mode. Any commands issued within 1 second of the change may be garbled or lost.

Note: When changing from NATIVE to SCPI interface language, use the command SYST:LANG "SCPI". Do not use the long form of the command, do not use a leading colon (:), and use exactly one space ( ) between the 'G' and the opening double quote ("). Deviations from this exact command may result in a syntax error.

Parameter(s):

Default Value: SCPI

:SYSTem:LANGuage:NATive:CATalog?

Title: System Language Native Catalog

Description: Returns all currently supported Native language commands. This is a comma-delimited list of commands that are permitted when

the :SYSTem:LANGuage has been set to NATIVE

:SYSTem:PRESet:REFerence:STATe <0 | 1 | ON | OFF>
:SYSTem:PRESet:REFerence:STATe?

Title: Preset Reference State

Description: A true value means the reference module will be reset on a system preset (:SYSTem:PRESet), causing it to re-acquire lock. In some situations, (especially if the Low Phase Noise and Ultra High Stability option is installed), having to wait for re-lock every preset could be undesirable. A false value means the reference will be unaffected by the preset.

Parameter(s): <0 | 1 | ON | OFF>

Query Return: 0 | 1

Default Value: OFF

:SYSTem:VERSion?

Title: System Version

Description: Returns the SCPI version number in <NR2> format that the instrument software complies with.

:TRIGger[:IMMediate]

Title: Trigger Immediate

Description: Causes the previously selected sweep to be triggered immediately if trigger system is in armed state due to previous :INITiate:IMMediate command.

:TRIGger:MODE <SWEep|POINt>
:TRIGger:MODE?

Title: Trigger Mode

Description: Selects the triggering mode for sweeps. In SWEep mode, the entire sweep will be performed on trigger; otherwise, in POINt mode, each trigger will advance the sweep once (for list sweep, this proceeds to the next list entry).

Parameter(s): <SWEep|POINt>

Query Return: SWE|POIN

Default Value: SWEep

:TRIGger:SEQuence<n>:SLOPe <POSitive|NEGative>
:TRIGger:SEQuence<n>:SLOPe?

Title: Trigger Sequence Slope

Description: Selects whether the internal pulse generator is triggered on the rising edge (POSitive) or falling edge (NEGative) of the external trigger signal.

Parameter(s): <POSitive|NEGative>

Query Return: POS|NEG

SEQuence Suffix Range: 3, Default = 3

Default Value: POSitive

:TRIGger:SEQuence<n>:SOURce <EXTernal{[1]|2}>
:TRIGger:SEQuence<n>:SOURce?

Title: Trigger Sequence Source

Description: Selects the source of the external trigger signal for the internal pulse generator as follows:

EXTernal – the rear panel PULSE TRIGGER IN connector. EXTernal1 and EXTernal2 have the same functionality as EXTernal.

Parameter(s): <EXTernal{[1]|2}>

Query Return: EXT

SEQuence Suffix Range: 3, Default = 3

Default Value: EXTernal

:TRIGger:SEQuence<n>:TYPE <FREerun|GATed|DELayed|TRIGgered|TRGDelay|COMPosite> :TRIGger:SEQuence<n>:TYPE?

Title: Trigger Sequence Type

Description: Selects the mode of triggering the internal pulse generator. The selections are:

FREerun – the pulse generator produces pulses at the internal PRF.

GATed – an external pulse gates the internal pulse generator on and off. When gated on, the pulse generator produces a single pulse at the internal PRF.

DELayed – the pulse generator produces pulses delayed by pulse delay1 ([:SOURce]:PULSe:DELay1) at the internal PRF

TRIGgered – the pulse generator is triggered by an external trigger signal to produce pulses.

TRGDelay – the pulse generator is triggered by an external trigger signal to produce pulses delayed by pulse delay1 ([:SOURce]:PULSe:DELay1).

COMPosite – an external pulse triggers the internal pulse generator and also pulse modulates the output signal. The internal pulse generator produces a single pulse delayed by pulse delay1 ([:SOURce]:PULSe:DELay1) which also pulse modulates the output signal.

Parameter(s): <FREerun|GATed|DELayed|TRIGgered|TRGDelay|COMPosite>

Query Return: FRE|GAT|DEL|TRIG|TRGD|COMP

SEQuence Suffix Range: 3, Default = 3

Default Value: FREerun

:TRIGger:SOURce <BUS|IMMediate|EXTernal> :TRIGger:SOURce?

Title: Trigger Source

Description: Selects the trigger source for the previously selected sweep. The source selections are:

BUS – The source is the group execute trigger command from the GPIB. The trigger will occur when either a <GET> or *TRG command is received.

IMMediate – The trigger signal is always true.

EXTernal – trigger on an external trigger input.

Parameter(s): <BUS|IMMediate|EXTernal>

Query Return: BUS|IMM|EXT

Default Value: BUS

:TRIGger:TYPE <AUTO|SINGle|EXTernal|MANual>
:TRIGger:TYPE?

Title: Trigger Type

Description: The Trigger Type represents a convenience command to set the following settings: Trigger Mode, Initiate Continuous, and Trigger Source.

The following represent the Trigger Types and the valid configurations of the associated settings:

(TRIG:TYPE = TRIG:SOUR, INIT:CONT, TRIG:MODE) AUTO = IMMediate, ON, SWEep SINGle = BUS, ON, SWEep EXTernal = EXTernal, ON, SWEep MANual = BUS, ON, POINt

Settings with multiple values are valid values for that Trigger Type and the value enclosed in the square brackets ([]) is the value that the relevant setting will be set to upon setting the relevant Trigger Type.

If a value is changed for a dependent setting and the collection of dependent setting values no longer matches a valid configuration, the Trigger Type query will return a value of REMote. The REMote condition can be resolved by setting the Trigger Type or modifying the dependent settings so that they match a valid configuration.

Related commands:

:TRIGger:MODE, :INITiate:CONTinuous, :TRIGger:SOURce

Parameter(s): <AUTO|SINGle|EXTernal|MANual>

Query Return: AUTO|SING|EXT|MAN

Default Value: SINGle

:TSWeep

Title: Trigger Sweep

Description: The :TSWeep command is a convenience command. It is equivalent to sending :ABORt;:INITiate[:IMMediate].

:UNIT:FREQuency <HZ>
:UNIT:FREQuency?

Title: Unit Frequency

Description: Constrained to HZ unit selection. Frequency setting utilizes additional

unit types directly as input parameters.

Parameter(s): <HZ>

Query Return: HZ

Default Value: HZ

:UNIT:POWer <DBM|DBUV|V>
:UNIT:POWer?

Title: Measurement Units

Description: Sets the default amplitude units for input, output and display.

Parameter(s): <DBM|DBUV|V>

Query Return: DBM|DBUV|V

Default Value: DBM

:UNIT:TIME <S>
:UNIT:TIME?

Title: Unit Time

Description: Selects the global default for all time related parameters used with all SCPI commands. Currently only seconds (S) are supported.

Parameter(s): <S>

Query Return: S

Default Value: S

Chapter 5 — Error Messages

5-1 Introduction

This chapter lists and describes each of the error messages related to MG362x1A signal generator operation. In addition, it provides information about the error message elements, the error query command, the error queue, and the classes of error messages.

5-2 Error Query

The `:SYSTem:ERRor?` query command is a request for the next entry in the instrument's error queue. Error messages in the queue contain an integer in the range [–32768 to 32768] denoting an error code and associated descriptive text. Negative codes are reserved by the SCPI standard and defined first in this chapter. Positive error codes are instrument-dependent. An error code value of zero indicates that no error has occurred (see Section 5-3).

The `:SYSTem:ERRor?` query command is required of all SCPI implementations.

The instrument responds to the `:SYSTem:ERRor?` query command with an error message in the following format:

<error code>,"<error description>;<device-dependent info>"

The <error code> is a unique error descriptor. Certain standard error codes are described in this chapter. The <error description> is a short description of the error, (optionally) followed by further information about the error. Short descriptions of the standard error codes are given in this chapter. The <device-dependent information> part of the response may contain information which will allow the user to determine the exact error and context. For example:

–241,"Hardware missing;Attenuator"

The maximum string length of <error description> plus <device-dependent information> is 255 characters. The <error description> shall be sent exactly as indicated in this chapter including case.

5-3 Error Queue

As errors are detected, error messages are placed in a queue. This queue is first in, first out and can hold a maximum of 10 messages. If the queue overflows, the last error message in the queue is replaced with the error message:

–350, "Queue overflow"

Any time the queue overflows, the least recent error messages remain in the queue, and the most recent error message is discarded. Reading an error message from the head of the queue removes that error message from the queue, and opens a position at the tail of the queue for a new error message, if one is subsequently detected.

When all error messages have been read from the queue, further error queries shall return

0, "No error"

The error queue shall be cleared when any of the following occur (IEEE 488.2, section 11.4.3.4):

- Upon power up
- Upon receipt of a `*CLS` command
- Upon reading the last error message from the queue

5-4 Error Codes

The system-defined error codes are chosen on an enumerated ("1 of N") basis. The SCPI-defined error codes and the <error description> portions of the query response are listed here. The first error described in each class (for example –100, –200, –300, –400) is a "generic" error. In selecting the proper error code to report, more specific error codes are preferred, and the generic error is used if the others are inappropriate.

5-5 No Error

This message indicates that the device has no errors.

Table 5-1. No Errors

Error Code	Error Description [description/explanation/examples]
0	"No error" The queue is completely empty. Every error in the queue has been read or the queue was purposely cleared by power-on, `*CLS`, etc.

5-6 Command Errors

An <error code> in the range [–199, –100] indicates that an IEEE 488.2 syntax error has been detected by the instrument's parser. The occurrence of any error in this class should cause the command error bit (bit 5) in the standard event status register to be set. One of the following events has occurred:

- An IEEE 488.2 syntax error has been detected by the parser. That is, a controller-to-device message is received which is in violation of the IEEE 488.2 standard. Possible violations include a data element which violates the device listening formats or whose type is unacceptable to the device
- An unrecognized header was received. Unrecognized headers include incorrect device-specific headers and incorrect or unimplemented IEEE 488.2 common commands
- A Group Execute Trigger (GET) was entered into the input buffer inside of an IEEE 488.2<PROGRAM MESSAGE>

Events that generate command errors shall not generate execution errors, device-specific errors, or query errors; see the other error definitions in this chapter.

Table 5-2. Command Errors (1 of 4)

Error Code	Error Description [description/explanation/examples]
11	EEPROM read/write error EEPROM read and/or write failed
10	Hardware communication error Failed to communicate with device hardware.
2	Option mismatch Operation failed due to option mismatch.
1	Model mismatch Operation failed due to option mismatch.
0	No error The queue is completely empty. Every error/event in the queue has been read or the queue was purposely cleared by power-on, `*CLS,` etc.
-100	Command error
-101	Invalid character A syntactic element contains a character which is invalid for that type.
–102	"Syntax error" An unrecognized command or data type was encountered.
–104	"Data type error" The parser recognized a data element different than one allowed; for example, numeric or string data was expected but block data was encountered.
–105	"GET not allowed" A Group Execute Trigger was received within a program message (see IEEE 488.2, 7.7).
–108	"Parameter not allowed" More parameters were received than expected for the header.
–109	"Missing parameter" Fewer parameters were received than required for the header.
–110	"Command Header Error" An error was detected in the header. This error message should be used when the device cannot detect the more specific errors described for errors –111 through –119.

Table 5-2. Command Errors (2 of 4)

Error Code	Error Description [description/explanation/examples]
–111	“Header Separator Error” A character which is not a legal header separator was encountered while parsing the header; for example, no white space followed the header, thus `*GMC“MACRO”` is an error.
–112	“Program mnemonic too long” The header contains more than 12 characters (see IEEE 488.2, 7.6.1.4.1).
–113	“Undefined header” The header is syntactically correct, but it is undefined for this specific device; for example, *XYZ is not defined for any device.
–114	“Header suffix out of range” The value of the numeric suffix attached to a program mnemonic makes the header invalid.
–120	“Numeric data error” This error, as well as errors –121 through –129, are generated when parsing a data element which appears to be numeric, including the nondecimal types. This particular error message should be used if the device cannot detect a more specific error.
–121	“Invalid character in number” An invalid character for the data type being parsed was encountered; for example, an alpha in a decimal numeric or a “9” in octal data.
–123	“Exponent too large” The magnitude of the exponent was larger than 32000 (see IEEE 488.2, 7.7.2.4.1).
–124	“Too many digits” The mantissa of a decimal numeric data element contained more than 255 digits excluding leading zeros (see IEEE 488.2, 7.7.2.4.1).
–128	“Numeric data not allowed” A legal numeric data element was received, but the device does not accept one in this position for the header.
–130	“Suffix error” This error, as well as errors –131 through –139, are generated when parsing a suffix. This particular error message should be used if the device cannot detect a more specific error.
–131	“Invalid suffix” The suffix does not follow the syntax described in IEEE 488.2, 7.7.3.2, or the suffix is inappropriate for this device.
–134	“Suffix too long” The suffix contained more than 12 characters (see IEEE 488.2, 7.7.3.4).
–138	“Suffix not allowed” A suffix was encountered after a numeric element which does not allow suffixes.
–140	“Character data error” This error, as well as errors 141 through 149, are generated when parsing a character data element. This particular error message should be used if the device cannot detect a more specific error.
–141	“Invalid character data” Either the character data element contains an invalid character or the particular element received is not valid for the header.

Table 5-2. Command Errors (3 of 4)

Error Code	Error Description [description/explanation/examples]
–144	“Character data too long” The character data element contains more than 12 characters (see IEEE 488.2, 7.7.1.4).
–148	“Character data not allowed” A legal character data element was encountered where prohibited by the device.
–150	“String data error” This error, as well as errors –151 through –159, are generated when parsing a string data element. This particular error message should be used if the device cannot detect a more specific error.
–151	“Invalid string data” A string data element was expected, but was invalid for some reasons (see IEEE 488.2, 7.7.5.2); for example, an END message was received before the terminal quote character.
–158	“String data not allowed” A legal string data element was encountered but was not allowed by the device at this point in parsing.
–160	“Block data error” This error, as well as errors –161 through –169, are generated when parsing a block data element. This particular error message should be used if the device cannot detect a more specific error.
–161	“Invalid block data” A block data element was expected, but invalid for some reason (see IEEE 488.2, 7.7.6.2); for example, an END message was received before the length was satisfied.
–168	“Block data not allowed” A legal block data element was encountered but was not allowed by the device at this point in parsing.
–170	“Expression error” This error, as well as errors –171 through –179, are generated when parsing an expression data element. This particular error message should be used if the device cannot detect a more specific error.
–171	“Invalid expression” The expression data element was invalid (see IEEE 488.2, 7.7.7.2); for example, unmatched parentheses or an illegal character.
–178	“Expression data not allowed” A legal expression data element was encountered but was not allowed by the device at this point in parsing.
–180	“Macro error” This error, as well as errors –181 through –189, are generated when defining a macro or executing a macro. This particular error message should be used if the device cannot detect a more specific error.
–181	“Invalid outside macro definition” Indicates that a macro parameter placeholder ($<number) was encountered outside of a macro definition.

Table 5-2. Command Errors (4 of 4)

Error Code	Error Description [description/explanation/examples]
–183	“Invalid inside macro definition” Indicates that the program message unit sequence, sent with a `*DDT` or `*DMC` command, is syntactically invalid (see IEEE 488.2, 10.7.6.3).
–184	“Macro parameter error” Indicates that a command inside the macro definition had the wrong number or type of parameters.

5-7 Execution Errors

An <error code> in the range [–299,–200] indicates that an error has been detected by the instrument's execution control block. The occurrence of any error in this class should cause the execution error bit (bit 4) of the standard event status register to be set. One of the following events has occurred:

- A <PROGRAM DATA> element following a header was evaluated by the device as outside its legal input range or is otherwise inconsistent with the device's capability
- A valid program message could not be properly executed due to some device condition

Execution errors shall be reported by the device after rounding and expression evaluation operations have taken place. Rounding a numeric data element, for example, shall not be reported as an execution error. Events that generate execution errors shall not generate command errors, device-specific errors, or query errors; see the other error definitions in this chapter.

Table 5-3. Execution Errors (1 of 4)

Error Code	Error Description [description/explanation/examples]
–200	"Execution error" This is a generic syntax error for devices that cannot detect more specific errors. This code indicates only that an execution error as defined in IEEE 488.2, 11.5.1.1.5 has occurred.
–201	"Invalid while in local" Indicates that a command is not executable while the device is in local due to a hard local control (see IEEE 488.2, 5.6.1.5); for example, a device with a rotary switch receives a message which would change the switches state, but the device is in local so the message cannot be executed.
–202	"Settings lost due to rtl" Indicates that a setting associated with a hard local control (see IEEE 488.2, 5.6.1.5) was lost when the device was changed to LOCS from REMS or to LWLS from RWLS.
–210	"Trigger error" A trigger error occurred in the signal generator.
–211	"Trigger Ignored" Indicates that a GET, `*TRG`, or triggering signal was received and recognized by the device but was ignored because of device timing considerations; for example, the device was not ready to respond. *Note:* a DTO device always ignores GET and treats `*TRG` as a command error.
–212	"Arm ignored" Indicates that an arming signal was received and recognized by the device but was ignored.
–213	"Init ignored" Indicates that a request for measurement initiation was ignored as another measurement was already in progress.
–214	"Trigger deadlock" Indicates that the trigger source for the initiation of a measurement is set to GET and subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error.
–215	"Arm deadlock" Indicates that the arm source for the initiation of a measurement is set to GET and subsequent measurement query is received. The measurement cannot be started until a GET is received, but the GET would cause an INTERRUPTED error.
–220	"Parameter error" Indicates that a program data element related error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –221 to –229.

Table 5-3. Execution Errors (2 of 4)

Error Code	Error Description [description/explanation/examples]
–221	“Settings conflict” Indicates that a legal program data element was parsed but could not be executed due to the current device state (see IEEE 488.2, 6.4.5.3 and 11.5.1.1.5).
–222	“Data out of range” Indicates that a legal program data element was parsed but could not be executed because the interpreted value was outside the legal range as defined by the device (see IEEE 488.2, 11.5.1.1.5).
–223	“Too much data” Indicates that a legal program data element of block, expression, or string type was received that contained more data than the device could handle due to memory or related device-specific requirements.
–224	“Illegal parameter value” Used where exact value, from a list of possibles, was expected.
–230	“Data corrupt or stale” Possibly invalid data; new reading started but not completed since last access.
–231	“Data questionable” Indicates that measurement accuracy is suspect.
–240	“Hardware error” Indicates that a legal program command or query could not be executed because of a hardware problem in the device. Definition of what constitutes a hardware problem is completely device-specific. This error message should be used when the device cannot detect the more specific errors described for errors –241 through –249.
–241	“Hardware missing” Indicates that a legal program command or query could not be executed because of missing device hardware; for example, an option was not installed. Definition of what constitutes missing hardware is completely device specific.
–250	“Mass storage error” Indicates that a mass storage error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –251 through –259.
–251	“Missing mass storage” Indicates that a legal program command or query could not be executed because of missing mass storage; for example, an option that was not installed. Definition of what constitutes missing mass storage is device-specific.
–252	“Missing media” Indicates that a legal program command or query could not be executed because of missing media; for example, no disk. Definition of what constitutes missing media is device-specific.
–253	“Corrupt media” Indicates that a legal program command or query could not be executed because of corrupt media; for example, bad disk or wrong format. Definition of what constitutes corrupt media is device-specific.

Table 5-3. Execution Errors (3 of 4)

Error Code	Error Description [description/explanation/examples]
–254	“Media full” Indicates that a legal program command or query could not be executed because the media was full; for example, there is no room on the disk. The definition of what constitutes a full media is device-specific.
–255	“Directory full” Indicates that a legal program command or query could not be executed because the media directory was full. The definition of what constitutes a full media directory is device-specific.
–256	“File name not found” Indicates that a legal program command or query could not be executed because the file name on the device media was not found; for example, an attempt was made to read or copy a nonexistent file. The definition of what constitutes a file not being found is device-specific.
–257	“File name error” Indicates that a legal program command or query could not be executed because the file name on the device media was in error; for example, an attempt was made to copy to a duplicate file name. The definition of what constitutes a file name error is device-specific.
–258	“Media protected” Indicates that a legal program command or query could not be executed because the media was protected; for example, the write-protect tab on a disk was present. The definition of what constitutes protected media is device-specific.
–260	“Expression error” Indicates that an expression program data element related error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –261 through –269.
–261	“Math error in expression” Indicates that a syntactically legal expression program data element could not be executed due to a math error; for example, a divide-by-zero was attempted. The definition of math error is device-specific.
–270	“Macro error” Indicates that a macro-related execution error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –271 through –279.
–271	“Macro syntax error” Indicates that a syntactically legal macro program data sequence, according to IEEE 488.2, 10.7.2, could not be executed due to a syntax error within the macro definition (see IEEE 488.2, 10.7.6.3).
–272	“Macro execution error” Indicates that a syntactically legal macro program data sequence could not be executed due to a some error in the macro definition (see IEEE 488.2, 10.7.6.3).
–273	“Illegal macro label” Indicates that the macro label defined in the *DMC command was a legal string syntax, but could not be accepted by the device (see IEEE 488.2, 10.7.3 and 10.7.6.2); for example, the label was too long, the same as a common command header, or contained invalid header syntax.
–274	“Macro parameter error” Indicates that the macro definition improperly used a macro parameter placeholder (see IEEE 488.2, 10.7.3).

Table 5-3. Execution Errors (4 of 4)

Error Code	Error Description [description/explanation/examples]
–275	“Macro definition too long” Indicates that a syntactically legal macro program data sequence could not be executed because the string or block contents were too long for the device to handle (see IEEE 488.2, 10.7.6.1).
–276	“Macro recursion error” Indicates that a syntactically legal macro program data sequence could not be executed because the device found it to be recursive (see IEEE 488.2, 10.7.6.6).
–277	“Macro redefinition not allowed” Indicates that a syntactically legal macro label in the *DMC command could not be executed because the macro label was already defined (see IEEE 488.2, 10.7.6.4).
–278	“Macro header not found” Indicates that a syntactically legal macro label in the `*GMC?` query could not be executed because the header was not previously defined.
–280	“Program error” Indicates that a downloaded program-related execution error occurred. This error message should be used when the device cannot detect the more specific errors described for errors –281 through –289.
–281	“Cannot create program” Indicates that an attempt to create a program was unsuccessful. A reason for the failure might include not enough memory.
–282	“Illegal program name” The name used to reference a program was invalid; for example, redefining an existing program, deleting a nonexistent program, or in general, referencing a nonexistent program.
–283	“Illegal variable name” An attempt was made to reference a nonexistent variable in a program.
–284	“Program currently running” Certain operations dealing with programs may be illegal while the program is running; for example, deleting a running program might not be possible.
–285	“Program syntax error” Indicates that a syntax error appears in a downloaded program. The syntax used when parsing the downloaded program is device-specific.
–286	“Program runtime error”

5-8 Device-Specific Errors

An <error code> in the range [–399,–300] or [1, 32767] indicates that the instrument has detected an error which is not a command error, a query error, or an execution error; some device operations did not properly complete, possibly due to an abnormal hardware or firmware condition. These codes are also used for self-test response errors. The occurrence of any error in this class should cause the device-specific error bit (bit 3) in the standard event status register to be set. The meaning of positive error codes is device-dependent and may be enumerated or bit mapped; the <error message> string for positive error codes is not defined by SCPI and available to the device designer. Note that the string is not optional; if the designer does not wish to implement a string for a particular error, the null string should be sent (for example, 42,""). The occurrence of any error in this class should cause the device-specific error bit (bit 3) in the standard event status register to be set. Events that generate device-specific errors shall not generate command errors, execution errors, or query errors; see the other error definitions in this chapter.

Table 5-4. Device-Specific Errors

Error Code	Error Description [description/explanation/examples]
–300	“Device-specific error” This is the generic device-dependent error for devices that cannot detect more specific errors. This code indicates only that a Device-Dependent Error as defined in IEEE 488.2, 11.5.1.1.6 has occurred.
–310	“System error” Indicates that some error, termed “system error” by the device, has occurred. This code is device-dependent.
–311	“Memory error” Indicates that an error was detected in the device’s memory. The scope of this error is device-dependent.
–312	“PUD memory lost” Indicates that the protected user data saved by the *PUD command has been lost.
–313	“Calibration memory lost” Indicates that nonvolatile calibration data used by the *CAL? command has been lost.
–314	“Save/recall memory lost” Indicates that the nonvolatile data saved by the *SAV? command has been lost.
–315	“Configuration memory lost” Indicates that nonvolatile configuration data saved by the device has been lost. The meaning of this error is device-specific.
–330	“Self-test failed”
–350	“Queue overflow” A specific code entered into the queue in lieu of the code that caused the error. This code indicates that there is no room in the queue and an error occurred but was not recorded.

5-9 Query Errors

An <error code> in the range [–499,–400] indicates that the output queue control of the instrument has detected a problem with the message exchange protocol described in IEEE 488.2, Chapter 6. The occurrence of any error in this class should cause the query error bit (bit 2) in the standard event status register to be set. These errors correspond to message exchange protocol errors described in IEEE 488.2, Section 6.5. One of the following is true:

- An attempt is being made to read data from the output queue when no output is either present or pending
- Data in the output queue has been lost

Events that generate query errors shall not generate command errors, execution errors, or device-specific errors; see the other error definitions in this chapter.

Table 5-5.

Error Code	Error Description [description/explanation/examples]
–400	"Query error" This is the generic query error for devices that cannot detect more specific errors. This code indicates only that a Query Error as defined in IEEE 488.2, 11.5.1.1.7 and 6.3 has occurred.
–410	"Query INTERRUPTED" Indicates that a condition causing an INTERRUPTED Query error occurred (see IEEE 488.2, 6.3.2.3); for example, a query followed by a DAB or GET before a response was completely sent.
–420	"Query UNTERMINATED" Indicates that a condition causing as UNTERMINATED Query error occurred (see IEEE 488.2, 6.3.2.2); for example, the device was addressed to talk and an incomplete program message was received.
–430	"Query DEADLOCKED" Indicates that a condition causing a DEADLOCKED Query error occurred (see IEEE 488.2, 6.3.1.7); for example, both input and output buffer are full and the device cannot continue.
–440	"Query UNTERMINATED after indefinite response" Indicates that a query was received in the same program message after a query requesting an indefinite response was executed (see IEEE 488.2, 6.5.7.5.7).

5-10 Parser Errors

An <error code> in the range [201, 212] is generated by the instrument's parser in response to the error condition described.

Table 5-6.

Error Code	Error Description [description/explanation/examples]
201	"Query only" Indicates the command is a query command only.
202	"No query allowed" Indicates that a query form of the command is not available.
203	"Parameter(s) not expected" Indicates that the command does not take any parameter(s).
204	"Constant not allowed in STATUS subsystem"
207	"Enumeric value not in union" Indicates an illegal parameter was sent with the command.
208	"Illegal number of parameters" Indicates an illegal number of parameters was sent with the command.
210	"Run out of memory handle" Indicates an internal parser problem.
211	"Unit not matched" Indicates the parameter unit does not match any defined unit for this parameter.
212	"Unit not required" Indicates no unit is required with this parameter.

5-11 Self-Test Messages

The MG362x1A signal generator has built-in diagnostics that allow you to run a built-in self test. The test results can be viewed on the screen and saved to a log file. To run self test, use the 3-Line Icon to access DIAGNOSTICS, then select SELF TEST and RUN SELF TEST. Below is a sample list of self tests with passing results:

Table 5-7. System Messages

Name	Type	Actual	Expected	Tolerance	Status
System voltage	Voltage	13.02 V	13.00 V	4.00 V	pass
+1.0V digital supply	Voltage	0.99 V	1.00 V	0.10 V	pass
+3.3V digital supply	Voltage	3.28 V	3.30 V	0.33 V	pass
+5.0V digital supply	Voltage	5.00 V	5.00 V	0.50 V	pass
+1.8V digital supply	Voltage	1.83 V	1.80 V	0.18 V	pass
+0.85V PS transceiver AVCC	Voltage	0.84 V	0.85 V	0.09 V	pass
+1.8V PS transceiver AVTT	Voltage	1.80 V	1.80 V	0.18 V	pass
+2.5V DDR SODIMM supply	Voltage	2.49 V	2.50 V	0.25 V	pass
+1.2V DDR SODIMM supply	Voltage	1.20 V	1.20 V	0.12 V	pass
+0.85V FPGA core supply	Voltage	0.85 V	0.85 V	0.09 V	pass
CPU Temperature	Temperature	37.40 C	50.00 C	50.00 C	pass

Table 5-8. SG Messages

Name	Type	Actual	Expected	Tolerance	Status
MOD +2.5V supply	Voltage	2.49 V	2.50 V	0.25 V	pass
MOD +3.3V supply	Voltage	3.31 V	3.30 V	0.33 V	pass
ALC	Temperature	59.56 C	30.00 C	60.00 C	pass
YIG	Temperature	40.25 C	30.00 C	60.00 C	pass
SWF +2.5V supply	Voltage	2.48 V	2.50 V	0.25 V	pass
SWF +3.3V supply	Voltage	3.33 V	3.30 V	0.33 V	pass
SWF +5.0V supply	Voltage	5.02 V	5.00 V	0.50 V	pass
SWF +8.0V supply	Voltage	8.07 V	8.00 V	0.80 V	pass
SWF +10.0V HP supply	Voltage	9.65 V	10.00 V	1.00 V	pass
SWF +28.0V supply	Voltage	27.68 V	28.00 V	2.80 V	pass
SWF –2.5V HP supply	Voltage	–2.52 V	–2.50 V	0.25 V	pass
SWF –5.0V HP supply	Voltage	–5.06V	–5.00 V	0.50 V	pass
SWF –32.0V HP supply	Voltage	–31.80 V	–32.00 V	3.20 V	pass
SWF –37.0V HP supply	Voltage	–36.41 V	–37.00 V	3.70 V	pass
SWF –3.3V HP supply	Voltage	–3.38 V	–3.30 V	0.33 V	pass
SWF ALC1 current monitor	Current	0.05 A	0.40 A	0.40 A	pass
SWF ALC2 current monitor	Current	0.32 A	0.40 A	0.40 A	pass
SWF DA1 current monitor	Current	0.41 A	0.40 A	0.40 A	pass
SWF DA2 current monitor	Current	0.00 A	0.40 A	0.40 A	pass
SWF PA1 current monitor	Current	0.30 A	0.70 A	0.70 A	pass

Table 5-8. (Continued)SG Messages

Name	Type	Actual	Expected	Tolerance	Status
SWF PA2 current monitor	Current	0.25 A	0.70 A	0.70 A	pass
SWF PA3 current monitor	Current	0.50 A	0.70 A	0.70 A	pass
SWF Temperature	Temperature	32.00 C	30.00 C	60.00 C	pass
Frequency Extension Module Not Available For This Model	Voltage	0.00 V	0.00 V	0.00 V	pass
REG +13V Main Voltage	Voltage	13.04 V	13.00 V	1.30 V	pass
REG +6.0V Standby	Voltage	6.01 V	6.00 V	0.60 V	pass
REG +1.5V supply	Voltage	1.52 V	1.50 V	0.15 V	pass
REG +3.3V supply	Voltage	3.35 V	3.30 V	0.33 V	pass
REG +3.9V supply	Voltage	4.03 V	3.90 V	0.39 V	pass
REG +6.0V supply	Voltage	6.02 V	6.00 V	0.60 V	pass
REG +9.0V supply	Voltage	9.18 V	9.00 V	0.90 V	pass
REG +10.0V supply	Voltage	10.18 V	10.00 V	1.00 V	pass
REG +15.0V supply	Voltage	15.10 V	15.00 V	1.50 V	pass
REG +16.0V supply	Voltage	16.59 V	16.00 V	1.60 V	pass
REG +24.0V supply	Voltage	24.22 V	24.00 V	2.40 V	pass
REG +25.0V supply	Voltage	26.24 V	25.00 V	2.50 V	pass
REG +30.0V supply	Voltage	30.13 V	30.00 V	3.00 V	pass
REG –6.0V supply	Voltage	–5.86 V	–6.00 V	0.60 V	pass
REG –16.0V supply	Voltage	–15.76 V	–16.00 V	1.60 V	pass
REG –15.0V supply	Voltage	–14.61 V	–15.00 V	1.50 V	pass
REG –30.0V supply	Voltage	–28.59 V	–30.00 V	3.00 V	pass
REG –40.0V supply	Voltage	–38.39 V	–40.00 V	4.00 V	pass
REG –6.0V standby	Voltage	–5.81 V	–6.00 V	0.60 V	pass
REG –13.0V standby	Voltage	–12.57 V	–13.00 V	1.30 V	pass
Overall Status	Info value				pass

Appendix A — Command Reference

10370-10385 E

Anritsu Company
490 Jarvis Drive
Morgan Hill, CA 95037-2809
USA
http://www.anritsu.com