Remote Data Transfer Guide

Agilent Technologies
ESG-D & ESG-DP Series Signal Generators

Agilent Technologies

Agilent Part No. E4400-90348

Printed in USA
February 2000

Supersedes November 1999

Acknowledgements

Hewlett-Packard® is a registered trademark of Hewlett-Packard Company.

HP® is a registered trademark of Hewlett-Packard Company.

UNIX® is a registered trademark of the Open Group.

Windows® and MS Windows® are U.S. registered trademarks of Microsoft Corporation.

Contents

1. Overview of this Guide

2. Understanding the Remote System Interface

3. Understanding Digital Data Processing

4. Understanding Signal Generator Memory Structure

5. Download Methods and Data Formats

Contents

8. User FIR File Downloads

9. ARB Waveform Data Downloads

10. Data Transfer Troubleshooting

1 Overview of this Guide

This chapter contains an overview of the information presented in this guide. It covers advanced data transfer issues, and will help you transfer user-defined data from a remote computer to the signal generator.

Much of the information is designed to help individuals creating experimental data with software programs such as MathCad™ or MatLab™, transfer this user-defined data from a computer into the signal generator. For information on controlling the *functionality* of the signal generator from a remote system, see the ESG family signal generators programming guide.

There are nine chapters in this guide covering data transfer to the signal generator:

- Chapter 2, "Understanding the Remote System Interface," covers topics associated with choosing and implementing an interface between your system controller and the signal generator. If your interface is already connected, adjusted, and verified, continue to the following section, "Understanding Signal Generator Memory Structure."
- Chapter 3, "Understanding Digital Data Processing," covers topics associated with the signal generator's digital data processing. This will help you understand how your data will be processed by the signal generator.
- Chapter 4, "Understanding Signal Generator Memory Structure," covers topics associated with the signal generator's internal memory structure. This will help you understand how and where your data is saved and applied within the signal generator.
- Chapter 5, "Download Methods and Data Formats," covers topics associated with the different methods of downloading data to the signal generator and their associated *data requirements and limitations*. This will help you understand how to create data that can be used by the signal generator and the different methods for transferring your data to the instrument.
- Chapter 6, "Downloading Directly to Pattern RAM," shows you how to download data directly to pattern RAM and modulate the carrier signal with the data.
- Chapter 7, "User File Data Downloads," shows you how to download user files into the signal generator's memory catalog and modulate the carrier signal with them.
- Chapter 8, "User FIR File Downloads," shows you how to download user FIR filter coefficient files into the signal generator's memory catalog and filter the modulated data with the user-defined FIR filter.
- Chapter 9, "ARB Waveform Data Downloads," shows you how to download I/Q waveform files into the signal generator.
- Chapter 10, "Data Transfer Troubleshooting."

Throughout this manual, information is included for UNIX- and Windows NT™-based operating systems using GPIB or RS-232 system interfaces.

2 Understanding the Remote System Interface

This chapter contains information that will help you understand the remote system interface capabilities of the signal generator.

NOTE Downloads using a GPIB remote system interface are *significantly faster* than downloads using an RS-232 serial interface.

The signal generator can be interfaced to a computer using either the general purpose interface bus (GPIB) or a serial connection to the rear-panel RS-232 AUXILIARY INTERFACE connector. This section explains the GPIB and RS-232 capabilities of the signal generator, how to connect and adjust the system interface, and how to verify its functionality. It is divided into the following subsections:

"The GPIB Remote System Interface" on page 2-2

Connecting, adjusting, and verifying the GPIB interface, including:

- required equipment
- input/output descriptions (SICL and VISA)
- interface cable connector specifications (pin-outs, mnemonics, and cable limitations)
- connection instructions
- GPIB address modification
- remote interface verification program

"The RS-232 Remote System (Auxiliary) Interface" on page 2-8

Connecting adjusting, and verifying the serial interface, including:

- required equipment
- interface cable connector *specifications* (pin-outs, mnemonics, and cable limitations)
- connection instructions
- null modem cable operational parameters for serial interfaces
- baud rates, handshake settings and other parameters for serial interfaces
- remote interface verification command

The GPIB Remote System Interface

GPIB is a high-performance bus that allows individual instruments and computers to be combined into integrated test systems. The bus and its associated interface operations are defined by the IEEE 488.1 standard. The IEEE 488.2 standard defines the interface capabilities of instruments and controllers in a measurement system, including some frequently used commands.

NOTE All of the functionality provided by GPIB is also available using the rear-panel RS-232 AUXILIARY INTERFACE. For more information on using this type of system configuration, see "The RS-232 Remote System (Auxiliary) Interface" on page 2-8.

Commands are sent over the GPIB via a controller's language system. HP BASIC is the language used in the programming examples in this book. HP BASIC was selected because the majority of GPIB computers have HP BASIC language capability. However, other languages can also be used. The use of HP BASIC is explained in more detail in the ESG Family Signal Generators Programming Guide.

Additional Required Equipment

The following tables list the additional required equipment to implement an GPIB system interface between your computer and your signal generator.

Table 2-1 GPIB Required Equipment for PC-Based Systems

Interface Card	Operating System	I/O Library	Languages	Backplane	Max I/O (kB/sec)	Buffering
HP 82341C	Windows 3.1/95/NT	SICL/VISA	C/C++, Visual Basic, HP VEE	ISA/EISA, 16 bit	750	Built-in
HP 82340B	Windows 3.1/95/NT	SICL/VISA	C/C++, Visual Basic, HP VEE	ISA/EISA, 16 bit	520	None
HP 82335B	MS-DOS, Windows 3.1	Command Library/SICL	C/C++, PASCAL, BASIC for PC (including Visual Basic), HP VEE	ISA/EISA, 8 bit	355	None

Table 2-2 GPIB Required Equipment for HP Series 700 Workstations Running HP-UX

Interface Card	Operating System	I/O Library	Languages	Backplane	Max I/O (kB/sec)	Buffering
HP E2071C	HP-UX	SICL/VISA	ANSI C, HP VEE, HP BASIC	EISA	750	Built-in
HP E2070C	HP-UX	SICL/VISA	ANSI C, HP VEE, HP BASIC	EISA	230	None

Table 2-3 GPIB Cables

Model	HP 10833A	HP 10833B	HP 10833C	HP 10833D
Length	1 meter	2 meters	4 meters	5 meters

I/O Libraries for GPIB

SCPI (standard commands for programmable instruments) is a popular language used to communicate with the signal generator. Do not confuse SCPI with SICL and VISA, which are I/O libraries of functions used by programs that communicate through GPIB. SCPI is the actual language used to communicate with the signal generator itself.

Agilent Technologies Standard Instrument Control Library (SICL) and Virtual Instrument Software Architecture (VISA) are I/O libraries used to develop I/O applications for the GPIB interface. These functions are used in C or BASIC programs to simplify communication with the signal generator.

SICL is a modular instrument communications library that works with a variety of computer architectures, I/O interfaces, and operating systems. Applications written in C/C++ or Visual BASIC using this library can be ported at the source code level from one system to another without, or with very few, changes.

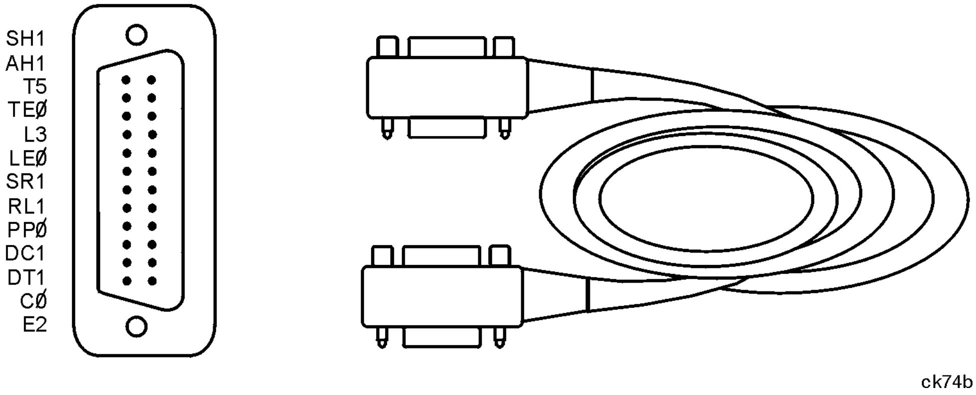
VISA is an I/O library that can be used to develop I/O applications and instrument drivers that comply with the VXI “plug & play” standards. Applications and instruments drivers developed with VISA can execute on VXI “plug & play” system frameworks that have the VISA I/O layer.

One or both of these libraries are included with the GPIB interface card. SICL/VISA for Hewlett-Packard Series 700 Controllers (model E2091D) and SICL/VISA for personal computers (model E2094E) may also be purchased. For additional information on SICL and VISA, see the user’s guides included with the SICL/VISA software package.

GPIB Interface Cables

The GPIB cable allows the signal generator to be connected to any other instrument or device on the GPIB interface bus. All GPIB instruments can be connected with GPIB cables. An example of a GPIB cable is shown in Figure 2-1.

Figure 2-1 Connector and GPIB Cable



The codes next to the connector, shown in Figure 2-1, describe the GPIB electrical capabilities of the signal generator, using IEEE Std. 488-1978 mnemonics (GPIB, IEEE-488, and IEC-625 are all electrically equivalent, though IEC-625 uses a unique connector). Briefly, the mnemonics translate as listed in Table 2-1.

Table 2-4 GPIB Cable Mnemonic Translations and Capabilities

Mnemonic	Signal Description	Capabilities
SH1	Source Handshake	Complete
AH1	Acceptor Handshake	Complete
T5	Talker	Capable of basic talker, serial poll, and unaddressed if MLA.
TE0	Talker, Extended Address	None
L3	Listener	Capable of basic listener, and unaddressed if MTA
LE0	Listener, Extended Address	None
SR1	Service Request	Complete
RL1	Remote Local	Complete
PP0	Parallel Poll	None
DC1	Device Clear	Complete
DT1	Device Trigger	Complete
C0	Controller	None
E2	Identifies electrical tristate drivers	

As many as 14 instruments can be connected to the signal generator via GPIB (15 total instruments in the system). The cables can be interconnected in a star pattern (one central instrument, with the GPIB cables emanating from that instrument like spokes on a wheel), or in a linear pattern (like boxcars on a train), or any combination pattern.

The following restrictions apply:

1. Each instrument must have a unique GPIB address ranging from 0 through 30 (decimal). See "Configuring the Interface" on page 2-6 for more information.
2. In a two-instrument system that uses just one GPIB cable, the cable length must not exceed 4 meters (9.13 ft.) between the two instruments.
3. When more than two instruments are connected on the bus, the cable length between each instrument must not exceed 2 meters (6.5 ft.) per unit.
4. The total cable length between all instruments must not exceed 20 meters (65 ft.).

Connecting the Interface

1. Install the required GPIB interface card in the computer.

Follow the installation instructions that arrived with the interface card. After you have completed the interface card installation procedure, continue with the next step.

2. Connect one end of the GPIB cable to the computer.

Insert one end of the cable into the GPIB connector on the computer. Use a slot screwdriver or your fingers to tighten the two retaining screws on the cable connector firmly against the connector on the computer.

3. Connect the other end of the GPIB cable to the signal generator.

Insert the other end of the GPIB cable into the GPIB connector on the signal generator. Use a slot screwdriver or your fingers to tighten the two retaining screws on the cable connector firmly against the connector on the signal generator.

Configuring the Interface

1. Activate the signal generator’s line power and set the signal generator’s GPIB address.

Each instrument in a GPIB network must have a unique address, ranging in value from 00 through 30 (decimal). The signal generator’s default address is 19. To adjust the address, press **Utility** > **HP-IB/RS-232** > **HP-IB Address**. If the address is set to another number, press **1** > **9** > **Enter** to set the address to 19.

NOTE Programming examples in this book assume that the signal generator’s GPIB address is 19. If necessary, modify the examples to correspond to your signal generator’s address.

2. Set the signal generator’s remote language.

Press **Utility** > **HP-IB/RS-232** > **Remote Language**. The default remote language is SCPI (standard commands for programmable instruments).

NOTE Though there are a number of remote languages compatible with other signal sources, SCPI is the only remote language that can implement all of the signal generators features. Unless you have reason to use one of the other remote languages (for example, programs previously written for another signal source that has been replaced by the ESG family signal generator), choose SCPI to maximize your control of the signal generator’s functionality. All programming commands throughout this documentation set are written in SCPI.

If SCPI is not highlighted, press **SCPI**. The signal generator’s remote language is now set to SCPI.

Verifying the GPIB System Interface

This program verifies that the GPIB connections and interface are functional. With the equipment set up as described in the previous section, clear and reset the controller by executing the following program:

```
10     Sig_gen=719
20     ABORT 7
30     LOCAL Sig_gen
40     CLEAR Sig_gen
50     OUTPUT Sig_gen;"*RST"
60     REMOTE Sig_gen
70     CLEAR SCREEN
80     PRINT "The source should now be in REMOTE."
90     PRINT "Verify that the `R’ annunciator is displayed."
100    END
```

Run the program and verify that the R (remote) annunciator is activated on the signal generator’s display. If it is not, verify that the signal generator address is set to 19 (see the section titled “Configuring the Interface”) and that the interface cable is properly connected.

If the controller display indicates an error message, a program line may have been entered incorrectly. If the controller accepts the remote statement but the signal generator’s remote annunciator does not appear on the signal generator’s display, refer to the service guide for troubleshooting information.

The RS-232 Remote System (Auxiliary) Interface

The signal generator can be remotely controlled using the rear-panel AUXILIARY INTERFACE, an RS-232 serial port.

NOTE All of the functionality provided by GPIB is available using the rear-panel AUXILIARY INTERFACE with the exception of indefinite blocks, serial polling, GET, non-SCPI remote languages, and remote mode.

Commands are sent over the serial interface via a controller's language system. HP BASIC is the language used in the programming examples in this book. HP BASIC was selected because the majority of GPIB computers have HP BASIC language capability. However, other languages can also be used. The use of HP BASIC is explained in more detail in the ESG family signal generators programming guide.

For information regarding RS-232 operating parameters such as handshake, baud rate, character format and parity, see “Overview of Serial (RS-232) Programming” in the ESG family signal generators programming guide.

Additional Required Equipment

The following table lists the additional items required when using a serial interface.

Table 2-5 RS-232 Remote System Interface Additional Required Equipment

Quantity	Description	HP/Agilent Part Number
1	Serial RS-232 Cable 9-pin (male) to 9-pin (female)	8120-6188
1	RS-232 Null Modem[1]	5158-6639
1	Serial RS-232 Adapter (female-female)	1252-7825

[1] Ensure proper pin connections between the computer and the signal generator.

Using the Null Modem

The null modem switches key transmission lines between the controller and the signal generator. When using the null modem in a direct connection, the signal generator appears to the controller as if it were a modem.

For personal computers, typically the connection is made to the COM2 RS-232 port, but COM1 may be used if it is available.

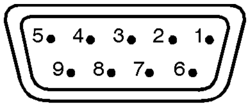
For UNIX workstations, typically the connection is made to the /dev/tty00 port, but /dev/tty01 may be used if it is available.

Some modification to the female-to-female adapter (1252-7825) is required. You must remove both standoffs from the adapter using a 5-mm nut driver.

RS-232 Serial (AUXILIARY INTERFACE) Interface Connector

The serial connector allows the signal generator to be connected to a serial port on a personal computer. The pin configuration of the signal generator’s rear-panel AUXILIARY INTERFACE is shown in Table 2-6. Only one instrument per RS-232 port on the computer may be connected at any given time.

Table 2-6 The Signal Generator’s AUXILIARY INTERFACE (RS-232) Connector



View looking into
rear panel connector

Pin Number	Signal Description	Signal Name
1	No Connection	
2	Receive Data	RECV
3	Transmit Data	XMIT
4	+5V	
5	Ground, 0V	
6	No Connection	
7	Request to Send	RTS
8	Clear to Send	CTS
9	No Connection	

Connecting the Interface

1. Connect the interface cable to the signal generator.

Attach the male end of the RS-232 cable to the signal generator’s AUXILIARY INTERFACE connector on the signal generator’s rear panel.

2. Connect the interface cable to the null modem.

Attach the female end of the RS-232 cable to the null modem.

3. Modify the female adapter.

Remove both standoffs from the female-to-female adapter using a 5-mm nut driver.

4. Connect one end of the modified adapter to the null modem and the other end to the selected port on the computer.

For personal computers, make the connection to the COM2 RS-232 port. COM1 is acceptable if available.

For UNIX workstations, connect to the /dev/tty00 port. Alternatively, /dev/tty01 may be used if it is available.

Configuring the Interface for Personal Computers

1. Set the signal generator's baud rate.

Press **Utility** > **HP-IB/RS-232** > **RS-232 Baud Rate**. The default RS-232 baud rate is 19200. Only use baud rates 19200 or lower. Press the appropriate softkey to adjust the signal generator's baud rate to the baud rate of your personal computer.

2. Set the signal generator's handshake.

Press **Utility** > **HP-IB/RS-232** > **RS-232 Pace**. To determine the hardware operating parameters, you need to know whether DSR (Data Set Ready) and CTS (Clear to Send) are active during communication with the controller, and the baud rate expected by the controller. Set the signal generator's handshake accordingly.

3. Set the signal generator's RS-232 echo.

Press **Utility** > **HP-IB/RS-232** > **RS-232 Echo Off On** to the desired operating state for your configuration.

4. Set the signal generator's remote language.

Press **Utility** > **HP-IB/RS-232** > **Remote Language**. The only remote language supported by RS-232 is SCPI (Standard Commands for Programmable Instruments).

If SCPI is not highlighted, press **SCPI**. The signal generator's remote language is now set to SCPI.

Configuring the Interface for UNIX Workstations

1. Set the signal generator's baud rate.

Press **Utility** > **HP-IB/RS-232** > **RS-232 Baud Rate**. The default RS-232 baud rate is 19200. Only use baud rates 19200 or lower. Press the appropriate softkey to adjust the signal generator's baud rate to the baud rate of your UNIX workstation.

2. Set the signal generator's handshake.

Press **Utility** > **HP-IB/RS-232** > **RS-232 Pace**. To determine the hardware operating parameters, you need to know whether DSR (Data Set Ready) and CTS (Clear to Send) are active during communication with the controller, and the baud rate expected by the controller. Set the signal generator's handshake accordingly.

3. Set the signal generator's RS-232 echo.

Press **Utility** > **HP-IB/RS-232** > **RS-232 Echo Off On** to the desired operating state for your configuration.

4. Set the signal generator's remote language.

Press **Utility** > **HP-IB/RS-232** > **Remote Language**. The only remote language supported is SCPI (Standard Commands for Programmable Instruments).

If SCPI is not highlighted, press **SCPI**. The signal generator's remote language is now set to SCPI.

Verifying the RS-232 System Interface

This command verifies that the serial interface is functional. With the equipment set up as described in the previous sections, clear and reset the controller. Execute the following command:

```
OUTPUT "*IDN?"
```

The signal generator should return a string similar to the following, depending on model:

<*instrument model name and number*>, `US37040098 B.03.00`

If it does not, verify that the RS-232 parameters are set correctly (see “Configuring the Interface” on page 2-6) and that the interface cable is properly connected.

3 Understanding Digital Data Processing

This chapter explains how the signal generator processes digital data. You will learn how data is digitally modulated through the signal generator (from pattern RAM memory in Option UN8/9 instruments or ARB memory in Option UND instruments), what types of data sources are available, and how data and control bits are defined within the signal generator's pattern RAM and ARB RAM. It is divided into the following subsections:

"RF Modulation with Option UN8/9" on page 3-2

The signal generator's basic Option UN8/9 digital modulation process, including:

- modulation block diagram
- order of data processing and modulation
- functional descriptions of each link in the digital modulation chain

"RF Modulation with Option UND" on page 3-4

The signal generator's basic Option UND digital modulation process, including:

- modulation block diagram
- order of data processing and modulation
- functional descriptions of each link in the digital modulation chain

"Data Sources" on page 3-6

Descriptions of the available data sources, including:

- internally generated data
- externally generated real-time data
- externally generated and downloaded data, including:
    - data downloaded directly to PRAM
    - data downloaded directly to volatile ARB memory
    - data downloaded into the signal generator's memory catalog, including:
        - user file data, user FIR file data, and ARB waveform data

"Data and Control Bits in Pattern RAM (Option UN8/9)" on page 3-9

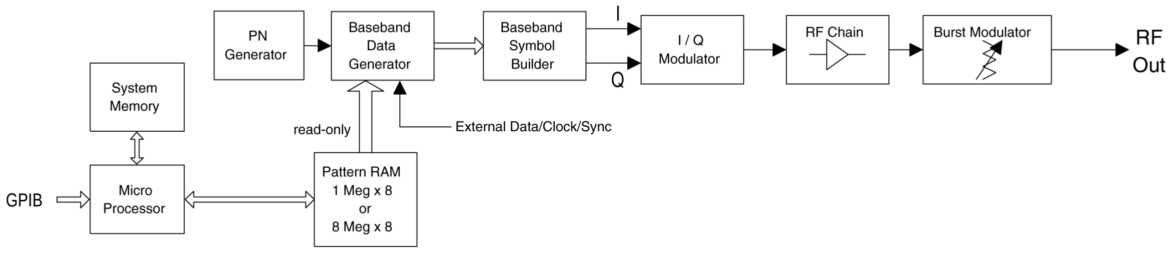
The data byte structure used by the signal generator's firmware to process data, including:

- bit order, bit function, bit settings, and bit description.

RF Modulation with Option UN8/9

The basic modulation of digital data in signal generators with Option UN8/9, from memory to RF output, occurs through the following path, as shown in the next figure. The process order is described following Figure 3-1.

Figure 3-1 ESG-D and ESG-DP Series Signal Generator Option UN8/9 Modulation Block Diagram



1. Pattern RAM

Internally generated or downloaded external data is loaded into pattern RAM (PRAM) memory by the signal generator’s firmware.

Pattern RAM not only contains the data bits to be modulated, but also contains the control bits for digitally modulating the carrier with burst. For detailed information, see “Data and Control Bits in Pattern RAM (Option UN8/9)” on page 3-9.

2. Baseband Data Generator

On each data clock, the baseband generator reads data and framing information from PRAM (pattern RAM), the PN (ITU pseudorandom data) generator, or external real-time DATA connectors (depending on instrument settings) and supplies formatted symbols to the baseband symbol builder.

For 1-bit-per-symbol modulation formats such as GMSK, one data value is read for each symbol clock period. For 2-bit-per-symbol modulation formats such as p/4DQPSK, two data values are read for each symbol clock period.

The PRAM address counter is incremented with every data clock. Therefore, each data clock cycles the PRAM address to the next byte (1 PRAM address = 1 byte). Since PRAM data is clocked into the baseband generator by the data clock and the PRAM address counter increments with the data clock, each address in PRAM can be described as an increment in time.

For continuous PN data with no framing, a dedicated hardware PN generator supplies data to the baseband generator, bypassing PRAM altogether.

3. Baseband Symbol Builder/Filtering Hardware

The symbol builder/filtering hardware generates the I/Q analog voltages corresponding to the selected modulation type and the selected filter type.

4. I/Q Modulator

The I/Q modulator supplies the modulating signal to the RF hardware.

5. RF Hardware

The RF hardware produces the carrier signal and modulates it with the I/Q modulator.

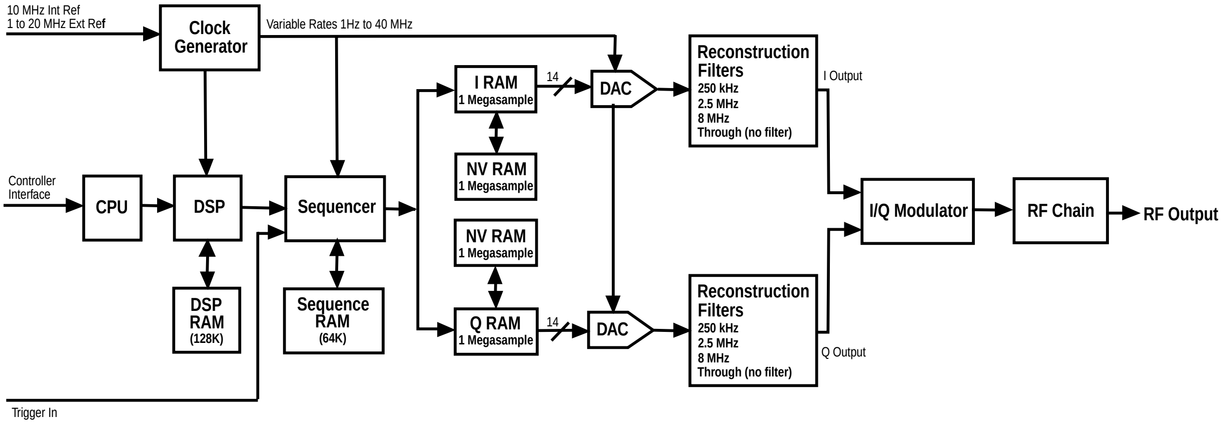
6. Burst Modulator

The burst modulator controls the amplitude of the RF signal.

RF Modulation with Option UND

The basic modulation of digital data in signal generators with Option UND, from memory to RF output, occurs through the following path, as shown in the next figure. The process order is described following Figure 3-2.

Figure 3-2 ESG-D and ESG-DP Series Option UND Modulation Block Diagram



1. Clock Generator

The arbitrary waveform generator sample clock features variable rates between 1 Hz and 40 MHz. The sample clock can be referenced to the internal 10 MHz reference frequency or to external reference signal between 1 and 20 MHz applied to the EXT REF IN connector.

2. Sequencer

The sequencer can be controlled using a simple table editor. You can save different waveforms as separate segments and sequence the waveform segments together to create a chain of repeating waveform types. You can also use the sequencer to control markers on the waveform segments.

3. I RAM, Q RAM, and Nonvolatile RAM

I RAM and Q RAM each provide memory space for 1 Megasample of data which can be clocked to the DACs for modulating the RF carrier. Data can be stored to the 1 Megasample nonvolatile RAM (the NVARB memory catalog), and loaded back into I RAM and Q RAM for subsequent playback.

4. Digital-to-Analog Converter and Filtering Hardware

14-bit DACs are used for superior fidelity. Reconstruction filters (250 kHz, 2.5 MHz, and 8 MHz) are provided, as well as a “through” filter which bypasses the internal filtering so that an external filter is applied.

5. I/Q Modulator

The I/Q modulator supplies the modulating signal to the RF hardware.

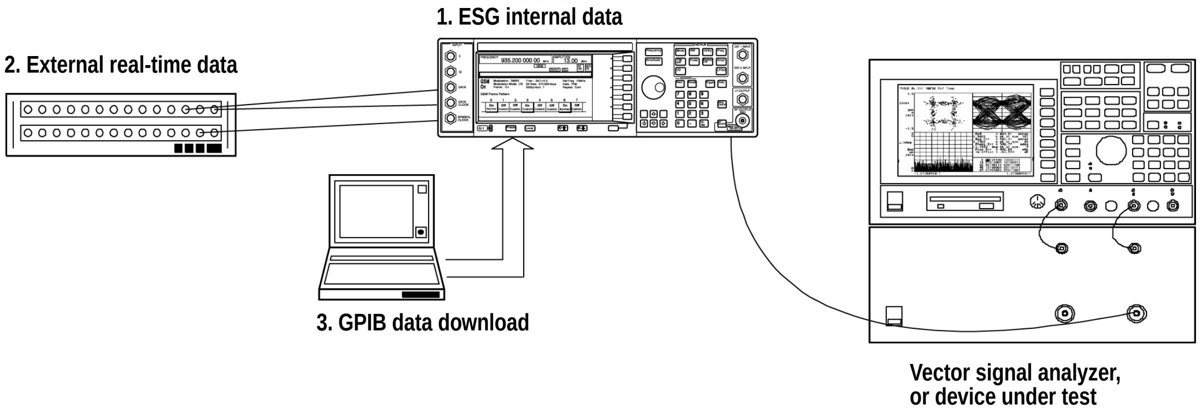
6. RF Hardware

The RF hardware produces the carrier signal and modulates it with the I/Q modulator.

Data Sources

To expand upon the signal generator's built-in flexibility, externally generated data patterns can be downloaded into the signal generator's internal memory, downloaded into pattern RAM (the memory read by the instrument's baseband generator for signal generators with Option UN8/9), or downloaded into ARB RAM for Option UND signal generators. For Option UN8/9 signal generators, data that has been downloaded into the memory catalog can subsequently be inserted into internally generated TDMA framing, including communications protocol, or transmitted as a continuous, unframed pattern.

Figure 3-3 ESG-D and ESG-DP Series Signal Generator Data Sources



There are three data sources available for digitally modulating the signal generator's RF carrier, either as a continuous or framed transmission. These three data sources are explained in the following subsections:

1. Internally Generated Data

The Option UN8/9 signal generator's firmware supplies:

- PN9 (an ITU 9-bit pseudorandom data pattern)
- PN11 (an ITU 11-bit pseudorandom data pattern)
- PN15 (an ITU 15-bit pseudorandom data pattern)
- PN20 (an ITU 20-bit pseudorandom data pattern)
- PN23 (an ITU 23-bit pseudorandom data pattern)
- FIX4 (a user-defined repeating 4-bit sequence)
- five different patterns of 1's and 0's (4 & 4, 8 & 8, 16 & 16, 32 & 32, and 64 & 64)

2. Externally Generated Real-Time Data

Real-time data and timing can be supplied by an external data source to the front-panel DATA, DATA CLOCK, and SYMBOL SYNC connectors. This data can be continuously transmitted, or can be framed by supplying a data-synchronous burst pulse to the EXT1 INPUT connector on the front panel. Additionally, the external data can be multiplexed into the internally generated framing.

NOTE This chapter does not cover externally generated real-time data. See the ESG Family Signal Generators User's Guide for details and timing diagrams.

3. External Data Downloaded via GPIB or RS-232 Remote Interface

There are three methods of downloading data to the signal generator: direct to pattern RAM, direct to the arbitrary waveform generator RAM, and to the signal generator's memory catalog.

Direct Pattern RAM Downloads (Option UN8/9 only) - For complete control over data, bursting, and protocol, you can write a block of data and control bits directly into pattern RAM, the memory location read by the signal generator's baseband generator, in order to produce formatted symbols for processing through the symbol builder/filtering hardware (Option UN8/9 required). Direct pattern RAM downloads are useful for generating non-standard framing for proprietary communications protocols, and for testing the limits of existing communications standards.

This method completely bypasses all internal firmware-generated framing and gives you the ability to control all time-domain framing parameters. However, pattern RAM is volatile and all data is lost when the signal generator is preset, overwritten by the selection of internally generated data, or powered off. For additional information, see "Data Downloads Directly into PRAM" on page 5-2.

Downloads to the Memory Catalog - Data may be downloaded to the signal generator's memory catalog, and subsequently applied to the RF output as needed. Depending on the options present in the signal generator, these data types include user files, FIR filter coefficients, and arbitrary waveform generator data.

User Files (Option UN8/9 only) provide a mechanism to download and store data patterns to the signal generator's microprocessor file system, the memory catalog. Subsequently, a user file can be selected as the data source for continuous modulation, or as a framed data source for any of the internally generated TDMA standards. Since the data is stored as a file in the signal generator's nonvolatile memory catalog, it is retained when the signal generator's line power is cycled or unplugged and can be recalled later. For additional information, see "User File Data Downloads" on page 5-4.

User FIR Files provide a mechanism to download and store Finite Impulse Response filter coefficient data to the signal generator's microprocessor file system, the memory catalog. Subsequently, a user FIR file can be selected as the filter source for any of the internally generated TDMA standards. Since the data is stored as a file in the signal generator's nonvolatile memory catalog, it is retained when the signal generator's line power is cycled or unplugged and can be recalled later. For additional information, see "FIR Filter Coefficient Data Downloads" on page 5-10.

Volatile ARB Memory Downloads (Option UND only) provide a mechanism to download to an Option UND signal generator's volatile arbitrary waveform generator memory. These waveforms can then be played individually or sequenced together and played as a waveform sequence. Once the data has been downloaded into volatile ARB memory, it can be copied or moved to nonvolatile arbitrary waveform generator memory for storage. For additional information, see "ARB Waveform Data Downloads" on page 5-11.

Nonvolatile ARB Memory Downloads (Option UND only) provide a mechanism to download and store ARB waveform data to the signal generator's microprocessor file system, the memory catalog. Subsequently, an ARB waveform can be selected from nonvolatile ARB memory as the data source for the arbitrary waveform generator. Since the data is stored as a file in the signal generator's nonvolatile memory catalog, it is retained when the signal generator's line power is cycled or unplugged and can be recalled later. For additional information, see "ARB Waveform Data Downloads" on page 5-11.

Data and Control Bits in Pattern RAM (Option UN8/9)

Pattern RAM not only contains the data bits to be modulated, but also contains the control bits for digitally modulating the carrier with burst.

The signal generator’s firmware adds seven control bits to each bit of data to be modulated. Therefore, each bit of user-defined data is contained within an 8-bit byte, or “address” in pattern RAM. Each byte in PRAM is organized as shown in the following table.

Table 3-1 Data and Control Bit Definitions for a Pattern RAM Address

Bit	Function	Value	Description
0	Data	0/1	This bit is the data to be modulated. This bit is a “don’t care” when burst (bit 2) is set to 0.
1	Reserved	0	Always 0.
2	Burst	0/1	Set to 1 = RF on. Set to 0 = RF off. For non-bursted, non-TDMA systems, this bit is set to 1 for all memory locations, leaving the RF output on continuously. For framed data, this bit is set to 1 for *on* timeslots and 0 for *off* timeslots
3	Reserved	0	Always 0.
4	Reserved	1	Always 1.
5	Reserved	0	Always 0.
6	Event 1 Output	0/1	Setting this bit to 1 causes a level transition at the EVENT 1 BNC connector. This can be used for many functions. For example, as a marker output to trigger external hardware when the data pattern has restarted, or to create a data-synchronous pulse train by toggling this bit in alternate addresses.
7	Pattern Reset	0/1	Set to 0 = continue to next sequential memory address. Set to 1 = end of memory and restart memory playback. This bit is set to 0 for all bytes except the last address of PRAM. For the last address (byte) of PRAM, it is set to 1 to restart the pattern.

When designing waveform data for subsequent direct download into pattern RAM, every bit of “modulation data” *must* be accompanied by these other seven control bits, forming an 8-bit byte that will occupy a specific address in PRAM. For further information, see “Data Downloads Directly into PRAM” on page 5-2.

4 Understanding Signal Generator Memory Structure

This section will help you understand the structure of the signal generator's internal memory, front-panel memory catalog navigation, the file naming conventions required by the signal generator, and instruction on manipulating files within the internal memory from a remote system. It is divided into the following topics:

"Memory Overview" on page 4-2

An overview of the signal generator's internal memory, including:

- default memory, motherboard RAM, firmware, arbitrary waveform generator memory, and pattern RAM
- volatile and nonvolatile ARB memory, their structures and differences

"Catalog Types" on page 4-3

A dictionary of memory catalog types, including:

- location within the memory structure
- data format
- file contents

"Using the Memory Catalog from the Front Panel" on page 4-5

Instruction in front-panel interface with the signal generator's memory catalog, including:

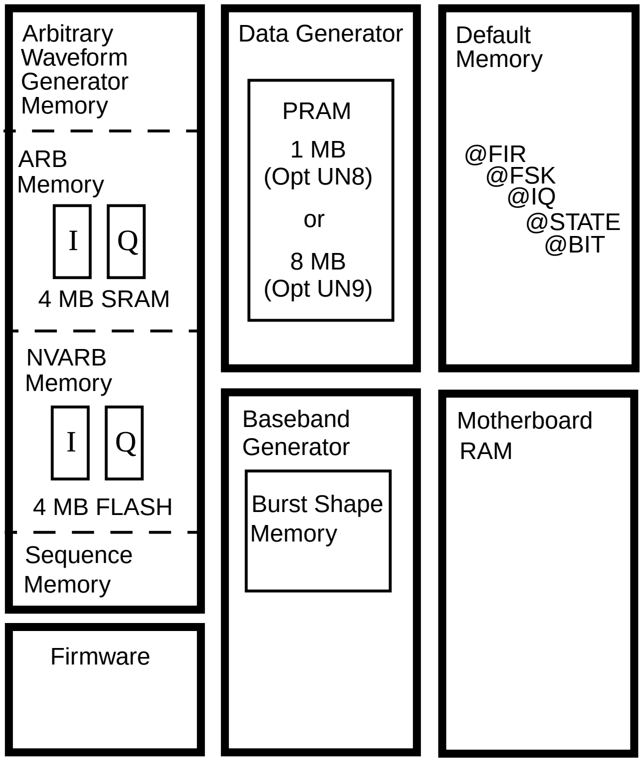
- catalog navigation
- viewing, copying, deleting and renaming files

"Using the Memory Catalog from a Remote System" on page 4-7

A discussion of remote access to the signal generator's internal memory structure, including:

- SCPI commands used for remotely storing, copying, moving, and deleting files
- SCPI command queries used for uploading data and querying catalog contents and open memory space

Memory Overview



The signal generator's memory, depending on the options present, consists of a default instrument memory, firmware memory, motherboard RAM, ARB memory (with Option UND), and data generator pattern RAM memory (with Option UN8/9).

Default instrument memory is used to store FIR filter coefficient data, modulation data, user file bit editor data, and instrument state data.

Firmware memory stores the code used to by the signal generator to manipulate internal data and the internal hardware.

Motherboard RAM stores data such as persistent state information, initialized and uninitialized variables, boot ROM, default directories, and default buffer.

Arbitrary waveform generator memory contains the 4 Mbyte static RAM-based volatile (ARB) memory, the 4 Mbyte FLASH RAM-based nonvolatile (NVARB) memory, and the sequencer memory. ARB memory holds the current data for playback using the arbitrary waveform generator. NVARB memory is used to store arbitrary waveform data for subsequent transfer to ARB memory for playback. Sequencer memory contains data related to waveform segments, waveform sequences, repetition, and marker settings.

The data generator contains the 1 MByte (Option UN8) or 8 MByte (Option UN9) pattern RAM memory. Data loaded into PRAM is read by the real-time I/Q baseband generator to produce the desired output. PRAM data includes the modulating data bits plus control information such as burst state, EVENT 1 output, and pattern reset.

The baseband generator contains the burst shape memory.

Catalog Types

There are three basic catalog types:

- Default
- ARB, for use by the arbitrary waveform generator
- Modulation

See the following table for a description of each catalog type, their accompanying subcatalogs, the types of information they contain, and any option the files are associated with.

All files are nonvolatile unless otherwise stated.

Table 4-1

Catalog Type	Subcatalogs	Information	Option
Default (binary format)	Binary	User files and any other binary files downloaded to the signal generator.	
	Bit	User file data created/modified using the bit editor or downloaded to the signal generator. Bit files contain a 10-byte header.	
	FIR	Finite impulse response filter files that contain FIR filter coefficient values.	
	List	Data associated with frequency and amplitude sweeps, such as frequency start/stop, dwell time, sweep type/repeat/trigger, point trigger, mode, direction, and trigger out polarity.	
	Shape	Data associated with specifying burst shape.	
	State	Data controlling instrument operating state parameters, such as frequency, power level, and mode.	
ARB (binary format)	ARB	Data residing in ARB memory (represents arbitrary waveform generator's static RAM). Default file name: `AUTO_GEN_WAVEFORM` File overwritten whenever a waveform is generated by the arbitrary waveform generator. Maximum file size: approximately 2 Mbytes	
	CDMA	IS-95A CDMA	UN5
	CDMOD	ARB digital modulation	
	DWCDMA	3GPP W-CDMA downlink channel data	101
	FCDMA	CDMA2000 forward channel data	101
	FWCDMA	W-CDMA forward channel data	100

Catalog Type	Subcatalogs	Information	Option
ARB *cont'd* (binary format)	MCDMA	Multicarrier CDMA IS-95A data	100
	MDMOD	Multicarrier ARB digital modulation data	
	MDWCDMA	Multicarrier 3GPP W-CDMA downlink data	100
	MFCDMA	Multicarrier CDMA2000 forward channel data	101
	MFWCDMA	Multicarrier W-CDMA forward channel data	100
	MTONE	Multitone data	
	NVARB	Data for use with the arbitrary waveform generator (represents arbitrary waveform generator's FLASH memory). Files copied into volatile ARB memory to be played. Maximum file size: approximately 2 Mbytes	
	RCDMA	CDMA2000 reverse channel data	101
	RWCDMA	W-CDMA reverse channel data	100
	Seq	Waveform sequence data, such as sequence contents (waveform segment file names), playing order, segment repetitions, marker settings, and sequence name.	
	UWCDMA	3GPP W-CDMA uplink channel data	100
	WCDMA	W-CDMA data	H97
Modulation	I/Q	User I/Q data, including I and Q values and differential mapping information.	
	FSK	User FSK data, including frequency deviation and differential mapping	

Using the Memory Catalog from the Front Panel

This section explains how to view, copy, rename and delete memory catalog files using the signal generator's front-panel interface.

Viewing Files

You can use the memory catalog to view the files in the signal generator's memory by file type, as follows:

1. Switch on the signal generator's line power.
2. If the signal generator is in remote mode, press the **Local** key to return the signal generator to local control, then press the **Utility** hardkey.
3. Press the softkey **Memory Catalog**.
4. Press **Catalog Type**. The default catalog type is **All**. When the catalog type is set to All, all files are listed in alphabetical order.

   To view a specific catalog types, choose that catalog's softkey.

NOTE The softkeys **ARB Catalog Types** and **Modulation Catalog Types** display an additional menu of softkeys where you choose the specific catalog files that you wish to view.

Copying Files

You can use the memory catalog to copy files within the signal generator's memory, as follows:

1. Highlight the desired file.
2. Press **Copy File** to open a text editor.
3. Press **Editing Keys** > **Clear Text**.
4. Using the alphabetical softkeys and the numeric keypad, input the new file name; pressing **Enter** to terminate the entry.

   The original file is copied to the new file using the name you specified.

Deleting Files

You can use the memory catalog to delete files within the signal generator's mass memory, as follows:

1. Highlight the desired file.
2. Press **Delete File**.

   The **Confirm Delete** softkey appears.
3. To delete the file from the memory catalog, press **Confirm Delete**.

Renaming Files

You can use the memory catalog to rename files within the signal generator's mass memory, as follows:

1. Highlight the desired file.
2. Press **More (1 of 2)** > **Rename File** to open a text editor.
3. Press **Editing Keys** > **Clear Text**.
4. Using the alphabetical softkeys and the numeric keypad, input the new file name, pressing **Enter** to terminate the entry.

   The original file is renamed as you specified.

Using the Memory Catalog from a Remote System

Querying the Memory Catalog Remotely

To query the contents of the memory catalog, amend the following SCPI command with the appropriate variable, then execute:

```
:MEMory:CATalog[:ALL]?
```

Executing this command appended with the `:ALL` variable returns the entire contents of the system memory. The following table lists the catalog variables.

Table 4-2 SCPI Variable and Associated Catalog for Remote Queries

SCPI Variable	Memory Catalog Queried
`:ALL?`	Entire contents of the default memory
`:BINary?`	Binary
`:BIT?`	Bit
`:CDMa?`	Option UN5 IS-95A CDMA
`:DMOD?`	ARB digital modulation
`:FCDMa?`	Forward channel Option 101 Multichannel CDMA2000
`:FIR?`	Finite Impulse Response
`:FSK?`	FSK modulation
`:FWCDma?`	Option 100 Multichannel Wideband CDMA
`:IQ?`	I/Q modulation
`:LIST?`	List
`:MDMod?`	Multicarrier ARB digital modulation
`:MCDMa?`	Option UN5 IS-95A Multicarrier CDMA
`:MFCDma?`	Multicarrier forward channel Option 101 Multichannel CDMA2000
`:MTONe?`	Multitone
`:RCDMa?`	Reverse channel Option 101 Multichannel CDMA2000
`:RWCDma?`	Reverse channel Option 100 Multichannel Wideband CDMA
`:SEQ?`	Sequence
`:SHAPe?`	Shape
`:STATe?`	State
`:WCDMa?`	Option H97 Wideband CDMA

Copying Memory Catalog Files Remotely

To copy the contents of one file to another, modify the following SCPI command with the appropriate file names and execute:

```
:MEMory:COPY[:NAME] "<file name>","<file name>"
```

Executing this command copis the contents of the first file to the second. For example, executing the command `:MEMory:COPY[:NAME] "FIR1","NEWFIR"` copies the data in the file `FIR1` to a new file named `NEWFIR`.

Deleting Individual Files Remotely

To delete a file, modify the following SCPI command with the appropriate file name and execute:

```
:MEMory:DELete[:NAME] "<file name>"
```

For example, executing the command `:MEMory:DELete[:NAME] "FIR1"` deletes the file `FIR1` from the memory catalog.

Deleting the Entire Contents of a Catalog Remotely

Execute the following SCPI command to delete a subcatalog.

```
:MEMory:DELete:<variable>
```

For example, `MEMory:DELete:FIR` deletes every file in the FIR catalog. The following table lists the variables and the catalogs they delete.

CAUTION Deleting the contents of a catalog is *irreversible*.

Table 4-3 SCPI Variable and Associated Catalog for Remote Deletions

Variable	Memory Catalog Deleted
`:ALL`	*Entire contents* of the default memory
`:BINary`	Binary
`:BIT`	Bit
`:DMOD`	ARB digital modulation
`:CDMa`	Option UN5 IS-95A CDMA
`:FCDMa`	Forward channel Option 101 Multichannel CDMA2000
`:FIR`	Finite Impulse Response
`:FSK`	FSK modulation
`:FWCDma`	Option 100 Multichannel Wideband CDMA
`:IQ`	I/Q modulation

Variable	Memory Catalog Deleted
`:LIST`	List
`:MCDMa`	Option UN5 IS-95A Multicarrier CDMA
`:MDMod`	Multicarrier ARB digital modulation
`:MFCDma`	Multicarrier forward channel Option 101 Multichannel CDMA2000
`:MTONe`	Multitone
`:RCDMa`	Reverse channel Option 101 Multichannel CDMA2000
`:RWCDma`	Reverse channel Option 100 Multichannel Wideband CDMA
`:SEQ`	Sequence
`:SHAPe`	Shape
`:STATe`	State
`:WCDMa`	Option H97 Wideband CDMA

Renaming Files Remotely

To rename a file, modify following SCPI command with the appropriate file names and execute:

```
:MEMory:MOVE "<source file name>","<destination file name>"
```

Executing this command renames the source file to the destination file name.

For example, executing the command `MEMory:MOVE "FIR1","NEWFIR"` renames the data contained within the file `FIR1` to `NEWFIR`.

5 Download Methods and Data Formats

The signal generator accepts data generated on a remote computer and subsequently downloaded directly into pattern RAM, ARB memory, or the signal generator's memory catalog. Depending on the options present, the signal generator accepts ARB waveform data, user file data, FIR filter coefficient data, or data downloaded directly to pattern RAM. This section explains the different types of download methods, and the data formatting required for each method. It is divided into the following subsections:

"Data Downloads Directly into PRAM" on page 5-2

An overview of direct-to-PRAM data downloads, including:

- data requirements and volatility
- file size limitations
- data and control bit definitions
- byte size

"User File Data Downloads" on page 5-4

An overview of user file data downloads, including:

- data requirements and volatility
- file size limitations
- user files as a data source for framed transmissions
- multiple user files selected as a data sources for different timeslots
- PN9, PN15, and user files as a data sources

"FIR Filter Coefficient Data Downloads" on page 5-10

An overview of user file data downloads, including:

- data requirements and volatility
- file size limitations
- byte size

"ARB Waveform Data Downloads" on page 5-11

An overview of arbitrary waveform generator data downloads, including:

- data requirements and volatility
- file size limitations
- data and control bit definitions (including EVENT 1 and EVENT 2 markers, and Even Second marker generation)
- I and Q data array definition
- waveform data bit-value-versus-power-output
- valid integer range

Data Downloads Directly into PRAM

Option UN8 or UN8/UN9 signal generators accept data downloaded directly into the data generator's pattern RAM. After downloading, this data can be used to stimulate the baseband generator's I/Q modulator. Direct downloads to PRAM allow you complete control over bursting, especially helpful for designing experimental or proprietary framing schemes.

Signal generators with Option UN8 contain a baseband generator assembly. This assembly builds modulation schemes by reading data stored to a pattern RAM assembly and constructing framing protocols according to the data patterns present in PRAM. PRAM data can be manipulated (types of protocols changed, standard protocols modified or customized, etc.) by the front-panel interface or by remote-command interface.

Typically, the signal generator's firmware generates the required data and framing structure and loads this data into PRAM. The data is subsequently read by the baseband generator, which in turn stimulates the input of the I/Q modulator. The signal generator is also equipped to accept data downloads directly into pattern RAM from a remote computer. After using a program such as MatLab™ or MathCad™ to generate the data, the data can be downloaded directly into PRAM in either a list format or a block format.

NOTE Due to the fact that there is no parsing involved, block data format downloads are *significantly faster* than list format downloads.

Data Requirements

There are two requirements for data downloaded directly into pattern RAM:

1. Data must be in binary form.

The data must be in binary form because the baseband generator is designed to receive data in binary form from the data generator.

2. For every bit of modulation data (bit 0), you must provide 7 bits of control information (bits 1-7).

The signal generator processes data in 8-bit bytes for each data bit. Each byte contains 1 bit of "data field" information, and seven bits of control information associated with the data field bit. The following table illustrates the required data and control bit definitions.

Table 5-1 Data and Control Bit Definitions

Bit	Function	Value	Comments
0	Data	0/1	This bit is the data to be modulated. This bit is a “don’t care” when burst (bit 2) is set to 0.
1	Reserved	0	Always 0.
2	Burst	0/1	Set to 1 = RF on. Set to 0 = RF off. For non-bursted, non-TDMA systems, this bit is set to 1 for all memory locations, leaving the RF output on continuously. For framed data, this bit is set to 1 for the valid data bits for *on* timeslots and 0 for all *off* timeslots.
3	Reserved	0	Always 0
4	Reserved	1	Always 1
5	Reserved	0	Always 0
6	Event 1 Output	0/1	Setting this bit to 1 causes a level transition at the EVENT 1 BNC connector. This can be used for many functions. For example, as a marker output to trigger external hardware when the data pattern has restarted, or to create a data-synchronous pulse train by toggling this bit in alternate addresses.
7	Pattern Reset	0/1	Set to 0 = continue to next sequential memory address. Set to 1 = end of memory and restart memory playback. This bit is set to 0 for all bytes except the last address of PRAM. For the last address (byte) of PRAM, it is set to 1 to restart the pattern.

Data Limitations

Total (data bits plus control bits) download size limitations are 1 Mbyte for Option UN8 and 8 Mbytes for Option UN9.

It is important to remember that a data pattern file containing 1 Mbyte of data for subsequent modulation must also contain another 7 Mbytes of control information. A file of this size would require a signal generator with Option UN9 and its 8 Mbyte pattern RAM. The largest amount of modulation data for a waveform in an Option UN8 signal generator is approximately 1 Mbit, leaving enough room for the required 7 Mbits control bits.

Data Volatility

PRAM data is *volatile.*

Data stored in PRAM will not survive a line power cycle. Also, it is overwritten whenever the mode is changed and a new TDMA format is activated.

User File Data Downloads

Option UN8 or UN8/UN9 signal generators accept user file data downloads. After downloading the data, the user files can be selected as the transmitting data source for the active digital communications standard.

In framed mode, user files data will be inserted into the data fields of an existing or user-defined, custom framed digital modulation format (DECT, PHS, TETRA, custom user-defined format, etc.).

The instrument's firmware generates the required framing structure and inserts the data contained within the user file into the data field(s) of the active format. For more information, see "User Files as Data Source for Framed Transmission" on page 5-6.

NOTE Unlike downloads to pattern RAM, user files contain "data field" information only. The control data bits required for files downloaded directly into PRAM are not required for user file data.

In pattern mode, the user file is modulated as a continuous, unframed stream of data, according to the modulation type, symbol rate, and filtering associated with the active digital communications format (DECT, TETRA, PHS, etc.).

When a user file is selected as the data source, the signal generator's firmware loads PRAM with the data specified in the user file, and sets the other seven control bits depending upon the operating mode selected, regardless of whether continuous or framed transmission is selected. In this manner, user files are mapped into PRAM such that one user file data bit equals one byte and occupies one address in PRAM.

Bit Memory versus Binary Memory

You can download user files to either the bit memory or binary memory catalog.

The bit memory catalog accepts data in any integer number of bits, up to the maximum available memory in the bit memory catalog (approximately 1 Mbit). The data length in bytes for files downloaded to the bit memory catalog is equal to the number of significant bits plus seven, divided by eight, then rounded down to the nearest integer. You must have enough bytes to contain the bits you specify. If the number of bits is not a multiple of 8, the least significant bits of the last byte will be ignored.

The bit memory catalog provides more versatility and is the preferred memory catalog for user file downloads.

The binary memory catalog requires data formatted in 8-bit bytes. Files stored or downloaded to the Binary catalog are converted to bit files prior to editing in the Bit File Editor. Afterward, these modified files from the binary memory catalog are stored in the bit memory catalog as bit files.

Data Requirements

1. Data must be in binary format.

Binary format is required because SCPI specifies the data in 8-bit bytes.

NOTE Not all binary values are ASCII characters that can be printed. In fact, only ASCII characters corresponding to decimal values 32 through 126 are printable keyboard characters. Typically, the ASCII character corresponding to an 8-bit pattern is not printable.

Because of this, the program written to download and upload user files *must correctly convert* the binary data into 8-bit characters.

2. Bit length must be a multiple of the data-field length of the active format.

Also, the bit length of a user file must be a multiple of the data-field length of the active format in order to completely fill the frame’s data field without leaving a remainder.

Remaining data is truncated by the signal generator’s firmware and is therefore not present in the resulting waveform at the RF output.

3. (Binary catalog downloads only) Bit length must be a multiple of 8.

The bit length of a user file must be a multiple of 8 because SCPI specifies the data in 8-bit bytes and because the binary catalog stores data in 8-bit bytes.

If the length (in bits) of the original data pattern is not a multiple of 8, you may need to:

a. Add additional bits to complete the ASCII character, or

b. replicate the data pattern to generate a continuously repeating pattern with no discontinuity, or

c. truncate and discard bits until you reach a string length that is a multiple of 8.

d. use a bit file and download to the bit memory catalog instead.

Data Limitations

The size of a user file that can be saved depends on the available memory in the memory catalog. The maximum amount of memory in the catalog is about 270 kbytes, but memory available for user files is reduced if memory is also being occupied by instrument state files or sweep list files that have been saved to the memory catalog.

Therefore, download size limitations are directly proportional to the memory catalog’s available memory space and the signal generator’s pattern RAM size (UN8 = 1 Mbyte, UN9 = 8 Mbyte). To determine the maximum user file size, you must consider the:

Framing overhead,

- pattern RAM size (1 MByte or 8 MByte), and
- available memory in the memory catalog.

It may be necessary to delete any superfluous files from the memory catalog before downloading larger files.

Data Volatility

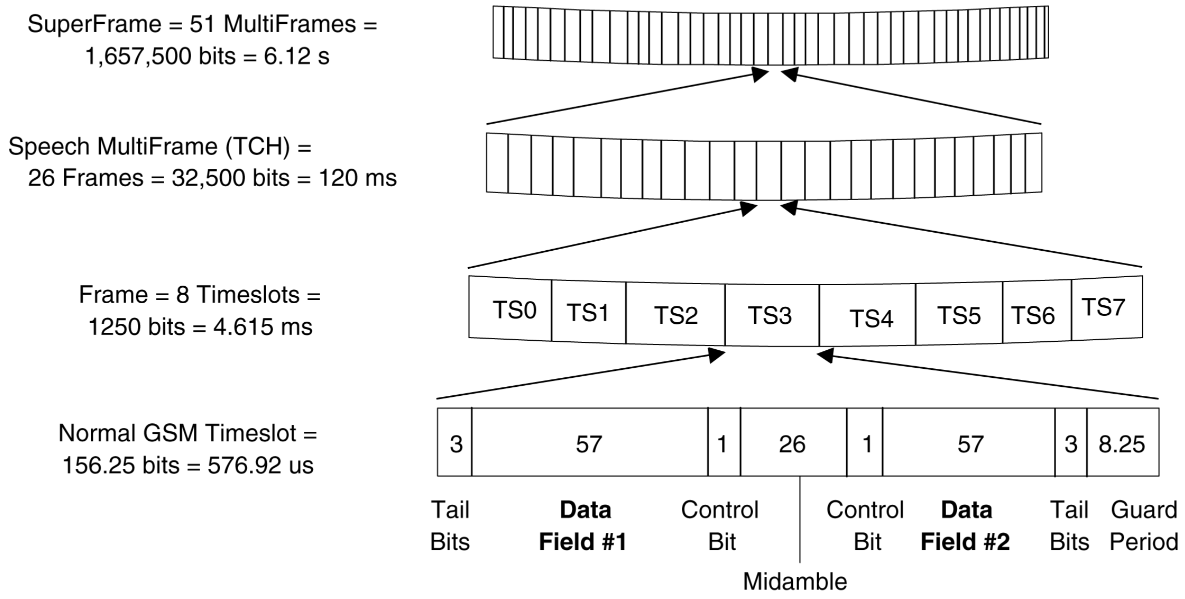
Downloaded user file data is *nonvolatile*. The data is stored to the instrument's memory catalog.

This data will survive a line power cycle. It will remain in the memory catalog until you delete the file, or until the signal generator's internal battery expires.

User Files as Data Source for Framed Transmission

Specifying a user file as the data source for a framed transmission provides you with an easy method to multiplex real data into internally-generated TDMA framing. The user file will fill the data fields of the active timeslot in the first frame, and continue to fill the same timeslot of successive frames as long as there is more data in the file. This functionality allows a communications system designer to download and modulate proprietary data sequences, specific PN sequences, or simulate multiframe transmission, such as those specified by some mobile communications protocols. As the example in the following figure shows, a GSM multiframe transmission requires 26 frames for speech.

Figure 5-1 GSM Multiframe Transmission



When a user file is selected as the data source for a framed transmission, the signal generator's firmware loads pattern RAM with the framing protocol of the active TDMA format. For all addresses corresponding to active (on) timeslots, burst bits are set to 1 and data bits are set with the contents of the user file for the data fields of the timeslot. Other bits are set according to the configuration selected. For inactive (off) timeslots, burst control bits are set to 0, and data is "unspecified." Pattern reset is set to 1 for the last byte in PRAM, causing the pattern to repeat after the last byte is read.

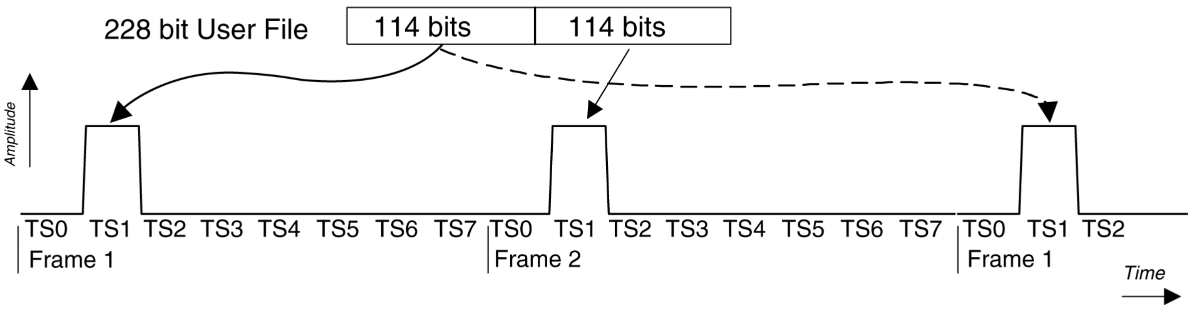
NOTE The data in pattern RAM is static. Firmware writes to PRAM once for the configuration selected and the hardware reads this data repeatedly. Firmware overwrites the volatile PRAM memory to reflect the desired configuration only when the data source or mode (digital communications format) is changed.

Take for example, transmitting a 228-bit user file for timeslot #1 (TS1) in a normal GSM transmission. Per the standard, a GSM normal channel is 156.25-bits long, with two 57-bit data fields (114 bits total per timeslot), and 42 bits for controlling or signalling purposes.

NOTE Compliant with the GSM standard, which specifies 156.25-bit timeslots, the signal generator uses 156-bit timeslots and adds an extra guard bit every fourth timeslot.

The seven remaining timeslots in the GSM frame are off. The user file will completely fill timeslot #1 in two consecutive frames, and will then repeat. See Figure 5-2.

Figure 5-2 Mapping User File Data to a Single Timeslot



For this protocol configuration, the signal generator's firmware loads pattern RAM with the bits defined in the following table.

Table 5-2 Pattern RAM Bit Definitions for GSM Normal Channel Transmission with a User File as the Data Source

Frame	Timeslot	PRAM Address	Data Bits	Burst Bits	Pattern Reset Bit
1	0	0 -155	0/1 (don't care)	0 (off)	0 (off)
1	1 (on)	156 - 311	42 bits set by GSM standard and first 114 bits of user file	1 (on)	0
1	2	312 - 467	0/1 (don't care)	0	0
1	3	468 - 624	0/1 (don't care)	0	0
1	4	625 - 780	0/1 (don't care)	0	0
1	5	781 - 936	0/1 (don't care)	0	0
1	6	937 - 1092	0/1 (don't care)	0	0

Frame	Timeslot	PRAM Address	Data Bits	Burst Bits	Pattern Reset Bit
1	7	1093 - 1249	0/1 (don't care)	0	0
2	0	1250 - 1405	0/1 (don't care)	0	0
2	1 (on)	1406 - 1561	42 bits set by GSM standard and remaining 114 bits of user file	1 (on)	0
2	2 through 6	1562 - 2342	0/1 (don't care)	0	0 (off)
2	7	2343 - 2499	0/1 (don't care)	0	0 (1 in address 2499 only)

Event 1 output is set to 0 or 1 depending on the sync out selection, which enables the Event 1 output at either the beginning of the frame, beginning of a specific timeslot, or at all timeslots.

Because timeslots are configured and enabled within the signal generator, a user file can be individually assigned to one or more timeslots. A timeslot cannot have more than one data source (PN sequence or user file) specified for it. The amount of user file data that can be mapped into hardware memory depends on both the amount of PRAM available on the baseband generator, and the number and size of each frame. the amount of PRAM required for a framed transmission is:

PRAM required = size of timeslot × timeslots per frame × number of frames

For example, to generate a superframe for GSM:

bits per timeslot × timeslots per frame × frames per multiframe × muliframes per superframe

= 156.25 × 8 × 26 × 51

= 1,657,5000 bytes.

The 1 Mbyte memory available with Option UN8 is not sufficient for this application. Option UN9 with 8 Mbytes is required.

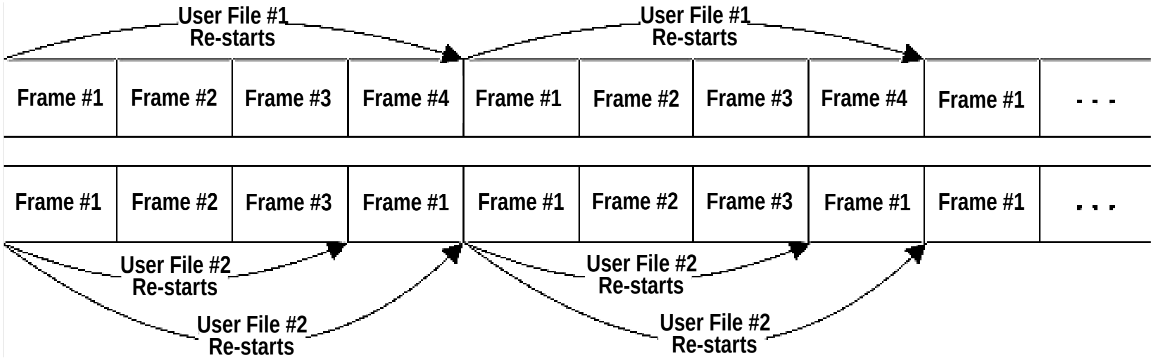
Multiple User Files Selected as Data Sources for Different Timeslots

If two or more user files are selected for a framed transmission, the amount of PRAM required is determined by the user file that generates the largest number of frames. In order to generate continuously repeating data patterns, each user file must be long enough to completely fill an integer number of timeslots. In addition, all user files must meet the "multiple of 8 bits" and "enough PRAM memory" requirements to be correctly modulated.

For example, user file #1 contains 114 bits and fills the data fields of a normal GSM timeslot, and user file #2 contains 148 bits for a custom GSM timeslot. In order to correctly transmit these data patterns as continuously repeating user files without discontinuities, both data patterns must be repeated four times. Therefore, user file #1 contains 456 bits, and user file 2 contains 592 bits. Each user file will then create exactly four frames in pattern RAM.

When two or more user files generate different numbers of complete frames, the user files will repeat on different cycles. All user files will restart when the user file that generates the largest number of frames repeats. For example, user file #1 needs four frames to completely transmit its data, and user file #2 needs only three. User file #2 will repeat after the third frame, and again when user file #1 repeats. See Figure 5-3. If these were integer multiples of each other, both user files would be continuous, and user file #2 would repeat after two frames.

Figure 5-3 Repeating Different Length User Files



PN9, PN15, and User File as Data Sources

If you select PN9 and/or PN15 to fill different timeslots without also selecting a user file, each PN sequence is hardware-generated to fill its designated timeslot. No discontinuity of the pseudorandom data occurs when the sequences are repeated.

When a PN sequence is selected to fill a timeslot and a user file is selected for another timeslot, the PN sequence is firmware-generated. The PN sequence is generated similarly to a user file.

During this process, PN9 sequences are handled differently than PN15.

A firmware-generated PN9 sequence always requires 511 frames to be continuous, regardless of the selected TDMA format. If the user file generates fewer than 511 frames, the user file will repeat as necessary to fill the data fields of the 511 frames. If the user file fills more than 511 frames, the PN9 sequence will repeat after the 511th frame, and repeat again when the user file repeats. In these cases, where the user file is not an integer multiple of 511 frames, either the user file or PN9 will be discontinuous.

Because of its length, a PN15 sequence will always be discontinuous regardless of whether a user file is selected for other timeslots, and regardless of the TDMA protocol selected. The PN15 behaves just like a user file of equal length (32,767 bits).

FIR Filter Coefficient Data Downloads

Options UN8 and UND signal generators accept finite impulse response (FIR) filter coefficient data downloads. After downloading the data, these user-defined FIR filter coefficient values can be selected as the filtering mechanism for the active digital communications standard.

Data Requirements

There are two requirements for FIR filter coefficient data downloads:

1. Data must be in ASCII format.

ASCII is required because the signal generator processes FIR filter coefficients as floating point numbers.

2. Data must be in List format.

List format is required because FIR filter coefficient data is processed as a list by the signal generator's firmware. See "Sample Command Line" on page 8-2.

Data Limitations

Filter lengths of up to 1024 taps are allowed. The oversample ratio (OSR) is the number of filter taps per symbol. Oversample ratios from 1 through 32 are possible.

The maximum combination of OSR and symbols allowed is 32 symbols with an OSR of 32. The signal generator's hardware achieves the filtering through a combination of digital and analog implementation.

Option UN8 is limited to 256 taps, 32 symbols and a 16-times oversample ratio. Therefore, any number of taps you insert are digitally resampled into a maximum of 16-times oversampling with 16 symbols, or a maximum 8-times oversampling with 32 symbols. FIR filters more than 32 symbols long cannot be used.

Option UND is limited to 512 taps and 24 symbols.

The sampling period (Δt) is equal to the inverse of the sampling rate (Fs). The sampling rate is equal to the symbol rate multiplied by the oversample ratio. For example, the GSM symbol rate is 270.83 ksps. With an oversample ratio of 4, the sampling rate is 1083.32 kHz and Δt (inverse of Fs) is 923.088 nsec.

Data Volatility

Downloaded FIR filter coefficient data is *nonvolatile*. It is stored to the instrument's memory catalog.

This data will survive a line power cycle. It will remain in the memory catalog until you delete the file, or until the signal generator's internal battery expires.

ARB Waveform Data Downloads

NOTE Downloads into volatile ARB memory are *significantly* faster than downloads into nonvolatile NVARB memory.

Option UND signal generators accept I/Q waveform data downloads. After downloading the data, these user-defined I/Q waveforms can be sequenced together with other waveforms and played as part of a waveform sequence.

Dual ARB waveforms are composed of two sets of data: I data and Q data. The I data drives the I port of the I/Q modulator and the Q data drives the Q port. These two sets of data give rise to the name *dual* arbitrary waveform generator.

I and Q data are arrays of 2-byte integers, in high-byte, low-byte order. The actual waveform shape is enclosed in the least-significant 14 bits of both the I and Q data. The other four bits (two for I and two for Q) are used for event markers and internal sequencing.

Table 5-3 I and Q Bit Delegation

Bit Number	I	Q
0 through 13	waveform data	waveform data
14	EVENT 2	reserved, set to 0
15	EVENT 1	reserved, set to 0

A 1 in bit position 14 or 15 of the I data will cause a TTL high to appear at the associated EVENT BNC connector on the signal generator’s rear panel.

Bit-value-versus-output-power:

- 0 gives negative full-scale output
- 8192 gives 0V output
- 16383 gives positive full-scale output

NOTE Because there are two files, one containing I data and another containing Q data, it is important that you remember to rewrite both the I and Q waveform data when downloading and overwriting a waveform in memory.

Volatile ARB Memory Versus Nonvolatile NVARB Memory

There are two types of arbitrary waveform generator memory: volatile static RAM, referred to as ARB memory, and nonvolatile FLASH RAM, referred to as NVARB memory.

Waveform data stored in ARB memory is volatile. The data in ARB memory is destroyed whenever the signal generator's line power is cycled. In ARB memory, waveform data may be downloaded, sequenced, and played back through the signal generator's I/Q baseband generator section.

Information stored in NVARB memory is nonvolatile. Waveforms stored in NVARB memory must first be moved to ARB memory in order to be sequenced and played. Waveform data stored in NVARB memory is not at risk when the signal generator's line power is cycled.

While waveforms may be directly downloaded to either ARB or NVARB memory, it is much faster to download to dynamic RAM-based volatile ARB memory. If memory space permits, first download the waveform data to ARB memory and then copy it in NVARB memory.

Data Requirements

There are six requirements for I/Q waveform data downloads:

1. Data must be in unsigned, offset (binary) format.

Data must be in unsigned, offset (binary) format because this is data type the signal generator is expecting to read.

2. Data must be in 2-byte integers.

Data must be in 2-byte integers because two bytes are needed to express the 14-bit waveforms. The signal generator accepts the Most Significant Byte first.

3. Input data must be between 0 and 16383.

Input data must be between 0 and 16383 based on the input specifications of the 14-bit DAC used to create the analog voltages for the I/Q modulator. Note that markers will increase the size of the I data beyond 16383.

4. Each I and Q waveform must contain at least 16 points.

Each I and Q waveform must have at least 16 points (a point equals an integer) in order to play in the waveform sequencer.

5. Each I and Q waveform must contain an even number of points.

Each I and Q waveform must contain an even number of points in order to play in the waveform sequencer.

6. Each I and Q waveform should be the same length.

Different length I and Q waveforms are allowed. The shorter waveform will give a 0V output from its end up to the length of the longer waveform.

Data Limitations

There are two types of arbitrary waveform generator memory: volatile, referred to as ARB memory, and nonvolatile, referred to as NVARB memory. The data limitations for the two different memory areas are explained separately.

For Volatile and Nonvolatile ARB Memory Downloads

ARB waveform memory space (approximately 1 Mpoints) is allocated in 4096-point (8192 byte) segments. Accordingly, regardless of how small the waveform is, it will always occupy at least 4096 points of volatile ARB memory. If a waveform file is too large to fit into a 4096-point memory segment, then additional memory space is allocated in multiples of 4096 points. For instance, if a waveform consists of 4100 points, it will be allocated to a 8192-point memory segment.

When played back through the sequencer, smaller waveforms play back only the number of points occupied by their waveform data, not the full 4096 points. See Table 5-4.

Table 5-4 Waveform Size Versus Memory Catalog File Size

Waveform	Number of Points	Memory Space (in points)
wfm1	32	4096
wfm2	128	4096
wfm3	4096	4096
wfm4	4098	8192
TOTAL POINTS	8354	20480

NOTE It is important to remember that, if there are many small waveform files stored in volatile ARB memory, the total memory usage may be much more than the sum of the points that make up the actual waveform data. There will not be as much memory available as perhaps expected, based on the combined sum of the individual waveform data files.

Data Volatility

Waveform data stored in ARB memory is *volatile*. This data is destroyed whenever the signal generator’s line power is cycled. Waveform data in ARB memory can be downloaded, sequenced, and played back through the signal generator’s I/Q baseband generator section.

Information stored in NVARB memory is *nonvolatile*. This data can survive a line power cycle, and remains in the memory catalog until you delete the file, or until the signal generator’s internal battery expires. To be sequenced and played, waveforms in NVARB memory must first be moved to ARB memory.

6 Downloading Directly to Pattern RAM

This chapter contains information that will help you transfer user-generated experimental or proprietary data from a system controller to the signal generator's pattern RAM. It explains how to download data directly into pattern RAM and modulate the carrier signal with the data. It is divided into the following subsections:

"Downloading in List Format" on page 6-2

Instructions for list format downloading, including:

- data requirements and limitations summary
- softkey presses and SCPI commands for preliminary signal generator set up
- SCPI commands and a sample command line for downloading data

"Downloading in Block Format" on page 6-4

Instructions for block format downloading, including:

- data requirements and limitations summary
- softkey presses and SCPI commands for preliminary signal generator set up
- using different number bases for downloading octal, hexadecimal, or binary data
- SCPI command and command query for downloading data and querying the contents of pattern RAM

"Modulating and Activating the Carrier" on page 6-6

Instructions for modulating the carrier with data in PRAM and activating the RF output, including:

- softkey presses for front panel interface to mode state, carrier frequency and power, modulation state and RF output state
- SCPI commands for remote interface to mode state, carrier frequency and power, modulation state and RF output state

Downloading in List Format

NOTE Because of parsing, list data format downloads are *significantly* slower than block format downloads.

Data Requirements and Limitations Summary

1. Data must be 8-bit, unsigned integers, from 0 to 255.

This is because list format downloads are parsed prior to being loaded into PRAM.

2. For every bit of modulation data (bit 0), you must provide 7 bits of control information (bits 1-7).

The signal generator processes data in 8-bit bytes. Each byte contains 1 bit of "data field" information, and seven bits of control information associated with the data field bit. See Table 5-1 on page 5-3 for the required data and control bits.

Total (data bits plus control bits) download size limitations are 1 Mbyte for Option UN8 and 8 Mbytes for Option UN9.

Because waveforms containing 1 Mbyte of data for subsequent modulation must also contain 7 Mbytes of control information, a file this size requires a signal generator with Option UN9 and its 8 Mbyte pattern RAM. The largest amount of modulation data for a waveform in an Option UN8 signal generator is approximately 1 Mbits, leaving enough room for the required 7 Mbits control bits.

Preliminary Setup

It is important to set up the digital communications format before downloading data. This allows the signal generator to define the modulation format, filter, and data clock. Activating the digital communications format after the data has been downloaded to PRAM may damage the downloaded data.

From the Front Panel Interface

To set up the TDMA format, press **Mode** > **desired format** and toggle the format on.

To set up the custom modulation format, press **Mode** > **Custom** and toggle the format on.

To adjust symbol rate, filtering, or other parameters, press the appropriate softkey and adjust the value.

Via the Remote Interface

For TDMA formats, send the SCPI command:

```
[:SOURce]:RADio:<desired format>[:STATe]:ON
```

Or for custom modulation send: `[:SOURce]:RADio:CUSTOm[:STATe]:ON`

To adjust symbol rate, filtering, or other parameters, execute the appropriate SCPI command.

SCPI Command to Download Data in List Format

```
:MEMory:DATA:PRAM:LIST <uint8>[,<uint8>,<...>]
```

This command downloads the list-formatted data directly into pattern RAM. The variable <uint8> is any of the valid 8-bit, unsigned integer values between 0 and 255, as specified by Table 5-1, “Data and Control Bit Definitions,” on page 5-3. Note that each value corresponds to a unique byte/address in PRAM.

Sample Command Line

For example, to burst a FIX4 data pattern of “1100” five times, then turn the burst off for 32 data periods (assuming a 1-bit/symbol modulation format), the command is:

```
:MEMory:DATA:PRAM:LIST 21,21,20,20,21,21,20,20,21,21,20,20,21,21,20,20,21,
21,20,20,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,16,
16,16,16,16,16,16,16,16,16,16,144
```

21 signifies data=1, burst = on (1).

20 signifies data=0, burst = on (1).

16 signifies data=unspecified, burst = off (0).

144 signifies data=unspecified, burst = off (0), pattern repeat = on (1).

Querying the Waveform Data

Use the following SCPI command line to determine whether there is a user-defined pattern in the PRAM:

```
:MEMory:DATA:PRAM?
```

Downloading in Block Format

NOTE Because there is no parsing, block data format downloads are *significantly* faster than list format downloads.

Data Requirements and Limitations Summary

1. Data must be in binary form.

This is because the baseband generator reads binary data from the data generator

2. For every bit of modulation data (bit 0), you must provide 7 bits of control information (bits 1-7).

The signal generator processes data in 8-bit bytes. Each byte contains 1 bit of "data field" information, and seven bits of control information associated with the data field bit. See Table 5-1 on page 5-3 for the required data and control bits.

Total (data bits plus control bits) download size limitations are 1 Mbyte for Option UN8 and 8 Mbytes for Option UN9.

Because a waveform containing 1 Mbit of data for subsequent modulation must also contain another 7 Mbits of control information, a file this size requires a signal generator with Option UN9 and its 8 Mbyte pattern RAM. The largest amount of modulation data for a waveform in an Option UN8 signal generator is approximately 125 kbits, which leaves enough room for the required 875,000 control bits.

Preliminary Setup

Before downloading data, set up the digital communications format to enable the signal generator to define the modulation format, filter, and data clock. Activating the digital communications format *after* data downloads to PRAM can damage the data.

From the Front Panel Interface

To set up the TDMA format, press **Mode** > **desired format** and toggle the format on.

To set up a custom modulation format, press **Mode** > **Custom** and toggle the format on.

To adjust symbol rate, filtering, or other parameters, press the appropriate softkey and adjust the value.

Via the Remote Interface

For TDMA formats, send the SCPI command:

```
[:SOURce]:RADio:<desired format>[:STATe]:ON
```

Or for custom modulation send: `[:SOURce]:RADio:CUSTom[:STATe]:ON`

To adjust symbol rate, filtering, or other parameters, execute the appropriate SCPI command.

SCPI Command to Download Data in Block Format

```
:MEMory:DATA:PRAM:BLOCK <datablock>
```

This command downloads the block-formatted data directly into pattern RAM.

Sample Command Line

A sample command line:

```
:MMEM:DATA:PRAM:BLOCK #ABC
```

A the number of decimal digits to follow in B.

B a decimal number specifying the number of data bytes in C.

C the binary user file data.

Example 1

```
:MMEM:DATA:PRAM:BLOCK #1912S407897
```

1 defines the number of decimal digits to follow in “B”.

9 denotes how many bytes of data are to follow.

12S407897 is the ASCII representation of the data downloaded to the signal generator. This variable is represented by C in the sample command line.

NOTE Not all binary values can be printed as ASCII characters. In fact, only ASCII characters corresponding to decimal values 32 to 126 are printable keyboard characters. The above example was carefully chosen for simplicity. Typically, the binary value corresponding to your 8-bit pattern is not printable.

Therefore, the program written to download and upload user files *must correctly convert* between binary and the visible representation of the data sequence.

Modulating and Activating the Carrier

The following section explains how to modulate the carrier with the data downloaded to PRAM, first from the front-panel interface, and then via remote SCPI commands.

From the Front Panel Interface

1. Set the carrier frequency to 2.5 GHz (**Frequency** > **2.5** > **GHz**).
2. Set the carrier amplitude –10.0 dBm (**Amplitude** > **–10** > **dBm**).
3. Turn modulation on (press **Mod On/Off** until the display annunciator reads `MOD ON`).
4. Activate the RF output (press **RF On/Off** until the display annunciator reads `RF ON`).

Via the Remote Interface

Execute the following SCPI commands to modulate and activate the carrier.

1. Set the carrier frequency to 2.15 GHz:

   ```
   [:SOURce]:FREQuency:FIXed 2.5GHZ
   ```

2. Set the carrier power to –10.0 dBm:

   ```
   [:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] -10.0DBM
   ```

3. Activate the modulation:

   ```
   :OUTPut:MODulation[:STATe] ON
   ```

4. Activate the RF output:

   ```
   :OUTPut[:STATe] ON
   ```

7 User File Data Downloads

This chapter contains information that will help you transfer user file data from a system controller to your signal generator.It explains how to download user files into the signal generator's memory catalog and modulate the carrier signal with them. It is divided into the following subsections:

"Downloading User File Data" on page 7-2

Instructions for downloading user files, including:

- data requirements and limitations summary
- SCPI command and sample command line for downloading user file data
- SCPI command for querying the user file data

"Selecting Downloaded User Files as the Transmitted Data" on page 7-6

Instructions for selecting user files as the transmitted data, including:

- softkey presses for user file data selection from the front panel interface
- SCPI commands for user file data selection from a remote interface
- softkey presses for TDMA format activation from the front panel interface
- SCPI command for TDMA format activation from a remote interface

"Modulating and Activating the Carrier" on page 7-7

Instructions for modulating the carrier with user file data and activating the RF output, including:

- softkey presses for carrier signal frequency and power definition, modulation state and RF output state from the front panel interface
- SCPI commands for carrier signal frequency and power definition, modulation state and RF output state from a remote interface

Downloading User File Data

This section includes information that explains how to download user file data. It includes data requirements and limitations, preliminary setup, SCPI commands and sample command lines for both downloads to the bit memory catalog and the binary memory catalog.

Data Requirements and Limitations Summary

1. Data must be binary.
2. Bit length must be multiple of the data-field length of the active TDMA format.
3. User file size is limited by the available memory in the memory catalog.
4. When designing user files, you must consider the signal generator's PRAM size (1 Mbyte or 8 Mbyte), framing overhead, and the available memory in the memory catalog.
5. For downloads to the binary memory catalog, bit length must be multiple of 8.

Preliminary Setup

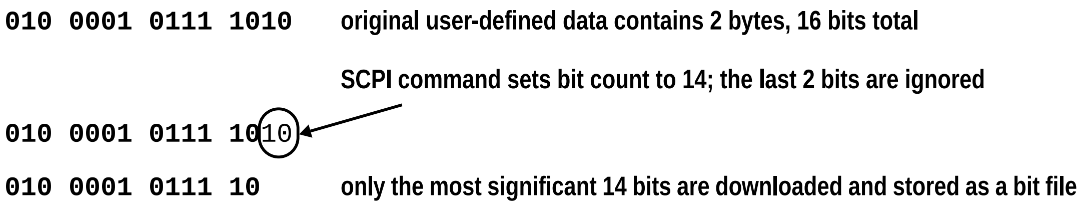
No preliminary setup is required for user file downloads.

Bit Memory Catalog Downloads

The bit memory catalog accepts data in any integer number of bits, up to the maximum available memory in the bit memory catalog (approximately 1 Mbit). The data length in bytes for files downloaded to the bit memory catalog is equal to the number of significant bits plus seven, divided by eight, then rounded down to the nearest integer.

You must have enough bytes to contain the bits you specify. If the number of bits is not a multiple of 8, the least significant bits of the last byte will be ignored.

For example, specifying 14 bits of a16-bit string using the command `:MEMory:DATA:BIT "userfile1", 14, #12Qz` results in the last 2 bits being ignored. See the following figure.


010 0001 0111 1010
original user-defined data contains 2 bytes, 16 bits total
SCPI command sets bit count to 14; the last 2 bits are ignored
010 0001 0111 1010
010 0001 0111 10
only the most significant 14 bits are downloaded and stored as a bit file

The bit memory catalog provides more versatility and is the preferred memory catalog for user file downloads.

SCPI Commands

```
:MEMory:DATA:BIT "<file name>", <bit count>, <datablock>
```

Execute this command to download the user file data into the signal generator's bit memory catalog. The variable `<file name>` denotes the name that will be associated with the downloaded user file within the signal generator.

Sample Command Line

```
:MEMory:DATA:BIT "<file name>", <bit count>, #ABC
```

`"<file name>"`	the name of the user file within the signal generator.
`<bit count>`	the number of significant bits in the data block.
`A`	the number of decimal digits to follow in B.
`B`	a decimal number specifying the number of data bytes in C.
`C`	the binary user file data.

Example 1

```
:MEMory:DATA:BIT "userfile1", 16, #12Qz
```

`userfile1` provides the user file name as it will appear in the signal generator's binary memory catalog.

`1` defines the number of decimal digits to follow in "B".

`2` denotes how many bytes of data are to follow.

`Qz` the ASCII representation of the 16 bits of data that are downloaded to the signal generator. This variable is represented by "C" in the sample command line.

NOTE Not all binary values can be printed as ASCII characters. In fact, only ASCII characters corresponding to decimal values 32 to 126 are printable keyboard characters. Typically, the binary value corresponding to your 8-bit pattern is not printable.

Therefore, the program written to download and upload user files *must correctly* convert between binary and the visible representation of the data sequence.

Querying the Waveform Data

Use the following SCPI command line to upload user file data from the binary memory catalog to the controller:

```
:MEMory:DATA:BIT? "<file name>"
```

The output format is the same as the input format.

Binary Memory Catalog Downloads

The binary memory catalog requires data formatted in 8-bit bytes. Files stored or downloaded to the Binary catalog are converted to bit files prior to editing in the Bit File Editor. Afterward, these modified files from the binary memory catalog are stored in the bit memory catalog as bit files.

The bit memory catalog provides more versatility and is the preferred memory catalog for user file downloads.

SCPI Commands

```
:MMEM:DATA "<file name>", <datablock>
```

Execute this command to download the user file data into the signal generator’s binary memory catalog. The variable `<file name>` denotes the name that will be associated with the downloaded user file within the signal generator.

Sample Command Line

```
:MMEM:DATA "<file name>", #ABC
```

`"<file name>"` the name of the user file within the signal generator.

`A` the number of decimal digits to follow in B.

`B` a decimal number specifying the number of data bytes in C.

`C` the binary user file data.

Example 1

```
:MMEM:DATA "userfile1", #1912S407897
```

`userfile1` provides the user file name as it will appear in the signal generator’s binary memory catalog.

`1` defines the number of decimal digits to follow in “B”.

`9` denotes how many bytes of data are to follow.

`12S407897` the ASCII representation of the data that is downloaded to the signal generator. This variable is represented by C in the sample command line.

NOTE Not all binary values can be printed as ASCII characters. In fact, only ASCII characters corresponding to decimal values 32 to 126 are printable keyboard characters. The previous example was carefully chosen for simplicity. Typically, the binary value corresponding to your 8-bit pattern is not printable.

Therefore, the program written to download and upload user files *must correctly* convert between binary and the visible representation of the data sequence.

Querying the Waveform Data

Use the following SCPI command line to upload user file data from the binary memory catalog to the controller:

```
:MMEM:DATA? "<file name>"
```

The output format is the same as the input format.

Selecting Downloaded User Files as the Transmitted Data

From the Front Panel

To select the user file as a continuous stream of unframed data for the active TDMA format or for a custom modulation, press **Mode** > **desired format** or **Custom**> **Data** > **User File**. Highlight the desired file in the catalog of user files and press **Select File.**

To activate the TDMA format press **Mode** > **desired format** and toggle the format on.

To activate custom modulation press **Mode** > **Custom** and toggle the **Custom Off On** softkey on.

Remotely

Execute the following SCPI command to select the user file from the bit memory catalog as a continuous stream of unframed data for the active TDMA format.

NOTE To select a user file from the binary memory catalog, execute the same commands shown in the following examples without BIT: preceding the filename.

For example, `[:SOURce]:RADio:<desired format>:DATA "<file name>"`

`[:SOURce]:RADio:<desired format>:DATA "BIT:<file name>"`

`[:SOURce]:RADio:<desired format>[:STATe] On` activates the desired TDMA format.

Execute the following SCPI command to select the user file as a continuous stream of data for the custom modulation format.

`[:SOURce]:RADio:CUSTom:DATA "BIT:<file name>"`

`[:SOURce]:RADio:CUSTom[:STATe] On` activates the custom modulation format.

From the Front Panel

To select the user file as framed data for the active TDMA format:, press **Mode** > **desired format** > **Data Format Pattern Framed** > **Configure Timeslots** > **Configure (current active timeslot)** > **Data** > **User File**. Highlight the desired file in the catalog of user files and press **Select File.**

To activate the TDMA format press **Mode** > **desired format** and toggle the format on.

Remotely

Execute the following SCPI command to select the user file as framed data for an NADC uplink traffic channel in timeslot 1. Modify the command as necessary for other formats.

`[:SOURce]:RADio:NADC:SLOT1:UTCHannel:DATA "BIT:<file name>"`

`[:SOURce]:RADio:NADC[:STATe] On` activates the NADC format.

Modulating and Activating the Carrier

The following section explains how to modulate the carrier, and activate the RF output; first from the front-panel interface and then via remote SCPI commands.

From the Front Panel

Press **Frequency** > **2.5** > **GHz** to set the carrier frequency to 2.5 GHz.

Press **Amplitude** > **-10** > **dBm** to set the carrier amplitude -10.0 dBm.

Press **Mod On/Off** until the display annunciator reads `MOD ON` to modulate the carrier.

Press **RF On/Off** until the display annunciator reads `RF ON` to activate the RF output.

Remotely

Execute the following SCPI commands to modulate and activate the carrier:

`[:SOURce]:RADio:<desired format>[:STATe] On` activates the desired TDMA format.

`[:SOURce]:FREQuency:FIXed 2.5GHZ` sets the carrier frequency to 2.15 GHz.

`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] -10.0DBM` sets the carrier amplitude to -10.0 dBm.

`:OUTPut:MODulation[:STATe] ON` modulates the carrier.

`:OUTPut[:STATe] ON` activates the RF output.

8 User FIR File Downloads

This chapter contains information that will help you transfer user Finite Impulse Response filter data from a system controller to your signal generator.

This section explains how to download user-defined FIR filter coefficient files into the signal generator’s memory catalog and activate them. It is divided into the following subsections:

“Downloading FIR Filter Coefficient Data” on page 8-2

Instructions for downloading FIR filter coefficient files, including:

- data requirements and limitations summary
- SCPI command and sample command line for downloading FIR filter coefficient data and querying the data

“Selecting Downloaded User FIR Filters as the Active Filter” on page 8-3

Instructions for filtering the signal with user FIR filters, including:

- softkey presses for front panel interface filter selection
- SCPI command for remote interface filter selection
- *softkey presses* for front panel interface TDMA format activation
- SCPI command for remote interface TDMA format activation

“Modulating and Activating the Carrier” on page 8-4

Instructions for modulating the carrier and activating the RF output, including:

- softkey presses for carrier signal frequency and power definition, modulation state and RF output state from the front panel interface
- SCPI commands for carrier signal frequency and power definition, modulation state and RF output state from a remote interface

Downloading FIR Filter Coefficient Data

Data Requirements and Limitations

1. Data must be ASCII.
2. Data must be transmitted in list format.
3. Filter length must be equal to or less than 1024 taps.
4. Oversample ratio must be between 1 and 32. (OSR is the number of filter taps per symbol.)
5. Option UN8 and Option UND have different limits on the maximum number of symbols allowed.

Preliminary Setup

No preliminary setup is required to download FIR filter coefficient data.

SCPI Command to Download FIR Filter Coefficients

Use the following SCPI command line to download list data from the controller to the signal generator’s FIR memory catalog:

```
:MEMory:DATA:FIR "<file name>",osr,coefficient{,coefficient}
```

Querying the FIR Filter Coefficient Data

Use the following SCPI command line to upload list data from the FIR memory catalog to the controller:

```
:MEMory:DATA:FIR? "<file name>"
```

Sample Command Line

For example, to download a typical set of FIR filter coefficient values, the command will look like this:

```
:MEMory:DATA:FIR "FIR1",4,0,0,0,0,0,0.000001,0.000012,0.000132,0.001101,
0.006743,0.030588,0.103676,0.265790,0.523849,0.809508,1,1,0.809508,0.523849,
0.265790,0.103676,0.030588,0.006743,0.001101,0.000132,0.000012,0.000001,0,0,
0,0,0
```

`FIR1` assigns the name FIR1 to the associated OSR and coefficient values. The file is then represented with this name in the FIR File catalog.

`4` specifies the oversample ratio.

`0,0,0,0,0,`
`0.000001,...` represent FIR filter coefficients.

Selecting Downloaded User FIR Filters as the Active Filter

From the Front Panel

To select the downloaded FIR filter data as the active filter for a TDMA format, press **Mode** > **TDMA** > **desired format** > **Modify Standard** > **Filter** > **Select** > **User FIR**. Highlight the desired file in the catalog of FIR files and press **Select File**.

To activate the TDMA format press **Mode** > **desired format** > and toggle the format on.

To select the downloaded FIR filter data as the active filter for a custom modulation, press **Mode** > **Custom** > **Filter** > **Select** > **User FIR**. Highlight the desired file in the catalog of FIR files and press **Select File**.

To activate the custom modulation, press **Mode** > **Custom** > **Custom Off On**.

To select the downloaded FIR filter data as the active filter for a CDMA modulation, press **Mode** > **CDMA Formats** > **desired format** > **CDMA Define** or **W-CDMA Define** > **Filter** > **Select** > **User FIR**. Highlight the desired file in the catalog of FIR files and press **Select File**.

To activate the CDMA modulation, press **Mode** > **CDMA Formats** > **desired format** > and toggle the format on.

Using the Remote Interface

Execute the following SCPI command to select the downloaded FIR filter data as the active filter for a TDMA modulation format.

`[:SOURce]:RADio:<desired format>:FILTer "<file name>"` `selects the user FIR filter you specify by file name as the active filter for the specified TDMA modulation format.`

`[:SOURce]:RADio:<desired format>[:STATe] On` activates the desired TDMA format.

Execute the following SCPI command to select the downloaded FIR filter data as the active filter for custom modulation.

`[:SOURce]:RADio:CUSTom:FILTer "<file name>"` `selects the user FIR filter you specify by file name as the active filter for the custom modulation format.`

`[:SOURce]:RADio:CUSTom[:STATe] On` activates the custom modulation format.

Execute the following SCPI command to select the downloaded FIR filter data as the active filter for custom modulation.

`[:SOURce]:RADio:<desired format>:ARB:FILTer "<file name>"` `selects the user FIR filter you specify by file name as the active filter for the CDMA modulation format.`

`[:SOURce]:RADio:<desired format>:ARB[:STATe] On` activates the CDMA modulation format.

Modulating and Activating the Carrier

The following section explains how to modulate the carrier, first from the front-panel interface, and then via remote SCPI commands.

From the Front Panel

To set the carrier frequency to 2.5 GHz, press **Frequency** > **2.5** > **GHz**.

To set the carrier amplitude -10.0 dBm, press **Amplitude** > **-10** > **dBm**.

To modulate the carrier, press **Mod On/Off** until the display annunciator reads `MOD ON`.

To activate the RF output, press **RF On/Off** until the display annunciator reads `RF ON`.

Using the Remote Interface

Execute the following SCPI commands to modulate and activate the carrier:

`[:SOURce]:FREQuency:FIXed 2.5GHZ` sets the carrier frequency to 2.15 GHz.

`[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] -10.0DBM` sets the carrier amplitude to -10.0 dBm.

`:OUTPut:MODulation[:STATe] ON` modulates the carrier.

`:OUTPut[:STATe] ON` activates the RF output.

9 ARB Waveform Data Downloads

This section explains how to download I/Q waveform files into the signal generator. It is divided into the following subsections:

"Downloading Waveforms into Volatile and Nonvolatile ARB Memory" on page 9-2

Instructions for downloading waveforms into volatile an nonvolatile ARB memory, including:

- data requirements and limitations summary
- SCPI command and command query for I-file downloads
- SCPI command and command query for Q-file downloads
- HP BASIC example programs (one for UNIX systems and one for Windows NT systems)

"Playing a Downloaded Waveform" on page 9-8

Instructions for playing waveforms that have been downloaded into volatile an nonvolatile ARB memory, including:

- how to load and select downloaded waveforms
- how to modulate the carrier and activate the RF output

Downloading Waveforms into Volatile and Nonvolatile ARB Memory

NOTE Downloads into volatile ARB memory are *significantly faster* than downloads into nonvolatile NVARB memory, due to the data parsing required by nonvolatile ARB memory.

There are two SCPI commands required to download I/Q data into ARB memory, one for the I data and one for the Q data. The signal generator will associate the I waveform values and the Q waveform values, and drive the I and Q modulators in the baseband generator with the stored waveform.

If only one of the two required commands is executed (I values only or Q values only), the missing data will be set to values corresponding to a 0 V output.

Data Requirements and Limitations

- Data must be in unsigned, offset (binary) format.
- Data must be in 2-byte integers (the signal generator accepts the MSB first).
- Input data must be between 0 and 16383.
- Each I and Q waveform must contain at least 16 points.
- Each I and Q waveform must contain an even number of points.
- Each I and Q waveform should be the same length.

Preliminary Setup

Before downloading into volatile ARB memory, turn off the ARB by pressing **Mode** > **Dual ARB** > **ARB Off On** until **Off** is highlighted or execute the SCPI command `[:SOURce]:RADio:ARB[:STATe] OFF`.

SCPI Commands

For Volatile ARB Memory:

```
:MMEM:DATA "ARBI:<waveform name>", <I waveform data>
```

This command downloads the I values for your waveform into volatile ARB memory. The variable <waveform_name> denotes the name that will be associated with the downloaded waveform data within the signal generator and therefore must be the same for both I and Q downloads.

```
:MMEM:DATA "ARBQ:<waveform name>", <Q_waveform data>
```

This command downloads the Q values for your waveform into volatile ARB memory.

For Nonvolatile ARB Memory:

```
:MMEM:DATA "NVARBI:<waveform name>", <I waveform data>
```

This command downloads the I values for your waveform into nonvolatile ARB memory. The variable <waveform_name> denotes the name that will be associated with the downloaded waveform data within the signal generator and therefore must be the same for both I and Q downloads.

`:MMEM:DATA "NVARBQ:<waveform name>",` <Q_waveform data>

This command downloads the Q values for your waveform into volatile ARB memory.

Sample Command Line

A sample command line:

```
:MMEM:DATA "ARBI:<waveform name>", #ABC
```

`"<waveform name>"` the name of the waveform file within the signal generator

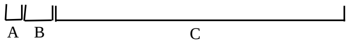
`A` the number of decimal digits to follow in B.

`B` a decimal number specifying the number of data bytes in C.

`C` the binary waveform data in 2-byte integers.

Example 1

```
:MMEM:DATA "ARBI:WAVEFORM1", #232<32 bytes of data>
```



`WAVEFORM1` the waveform name

`2` defines the number of decimal digits to follow in B. This variable is represented by A in the sample command line.

`32` denotes how many bytes of data are to follow. This variable is represented by B in the sample command line.

`<32 bytes of data>` the binary waveform data in 2-byte integers. Data order is defined as Most Significant Byte (first) and Least Significant Byte (last). This variable is represented by C in the sample command line.

NOTE This command line only sends the I-half of the I/Q waveform data. After sending the above command, you must send `MMEM:DATA "ARBQ:<waveform name>", <Q waveform data>` for the Q data.

Querying the Waveform Data

Use the following SCPI command line to upload waveform data from volatile ARB memory to the personal computer:

```
:MMEM:DATA? "ARBI:<waveform name>"
:MMEM:DATA? "ARBQ:<waveform name>"
```

Use the following SCPI command line to upload waveform data from nonvolatile ARB memory to the personal computer:

```
:MMEM:DATA? "NVARBI:<waveform name>"
:MMEM:DATA? "NVARBQ:<waveform name>"
```

Example Programs

Waveform Downloading Using HP BASIC for Windows™

The following program shows you how to download waveforms using HP BASIC for Windows™ into volatile ARB memory.

To download into nonvolatile ARB memory, replace line 80 with:

80 OUTPUT @ESG USING "#,K";":MMEM:DATA ""NVARBI:testfile"", #"

and replace line 130 with:

130 OUTPUT @ESG USING "#,K";":MMEM:DATA ""NVARBQ:testfile"", #"

First, the I waveform data is put into an array of integers called `Iwfm_data` and the Q waveform data is put into an array of integers called Qwfm_data. The variable `Nbytes` is set to equal the number of bytes in the I waveform data. This should be twice the number of integers in `Iwfm_data`, since an integer is 2 bytes. Input integers must be between 0 and 16383.

In the `Output` commands, `USING “#,K”` formats the data. The pound symbol (#) suppresses the automatic EOL (End of Line) output. This allows multiple output commands to be concatenated as if they were a single output. The “K” instructs HP BASIC to output the following numbers or strings in the default format.

```
5 Npoints=20
10 ALLOCATE INTEGER Iwfm_data(1:Npoints),Qwfm_data(1:Npoints)
15 DEG
20 FOR I=1 TO Npoints
25  Iwfm_data(I)=INT(8191*(COS(I*360/Npoints))+8192)
30  Qwfm_data(I)=INT(8191*(COS(I*360/Npoints))+8192)
35 NEXT I
40 Nbytes=2*Npoints
45 Assign @ESG to 719
50 Assign @ESGb to 719; FORMAT MSB FIRST
60 Nbytes$=VAL$(Nbytes)
70 Ndigits$=VAL$(LEN$(Nbytes$))
80 OUTPUT @ESG USING "#,K";":MMEM:DATA ""ARBI:testfile"", #"
90 OUTPUT @ESG USING "#,K";Ndigits$
100 OUTPUT @ESG USING "#,K";Nbytes
110 OUTPUT @ESGb;Iwfm_data(*)
120 OUTPUT @ESG;
130 OUTPUT @ESG USING "#,K";":MMEM:DATA ""ARBQ:testfile"", #"
140 OUTPUT @ESG USING "#,K";Ndigits$
```

```
150 OUTPUT @ESG USING "#,K";Nbytes$
160 OUTPUT @ESGb;Qwfm_data(*)
170 OUTPUT @ESG;
180 Assign @ESG TO *
190 Assign @ESGb TO *
200 END
```

Table 9-1 . Program Comments

5:	Sets the number of points in the waveform.
10:	Defines arrays for I and Q waveform points. Sets them to be integer arrays.
15:	Sets HP BASIC to use degrees for cosine and sine functions.
20:	Sets up loop to calculate waveform points.
25:	Calculates I waveform points.
30:	Calculates Q waveform points.
35:	End of loop.
40:	Calculates number of bytes in I or Q waveform.
45:	Opens an I/O path to the signal generator using GPIB. 7 is the address of the GPIB card in the computer, and 19 is the address of the signal generator. This I/O path is used to send ASCII data to the signal generator.
50:	Opens an I/O path for sending binary data to the signal generator. `FORMAT MSB FIRST` is needed to send the 2 bytes of each integer to the signal generator in the correct order.
60:	Creates an ASCII string representation of the number of bytes in the waveform.
70:	Finds the number of digits in Nbytes.
80:	Sends the I waveform SCPI download-to-ARBI command and the beginning of the ASCII header for the data. `testfile` is the waveform name that will be used in the signal generator.
90 to 100:	Sends the rest of the ASCII header.
110:	Sends the binary data. Note that `ESGb` is the binary I/O path.
120:	Sends an End-of-Line to terminate the transmission.
130 to 170:	Executes same commands for the Q waveform.
180 to 190:	Closes the connections to the signal generator.
200:	End the program.

Waveform Downloading Using HP BASIC for UNIX

The following program shows you how to download waveforms into volatile ARB memory using HP BASIC for UNIX.

First, the I waveform data is put into an array of integers called `Iwfm_data` and the Q waveform data is put into an array of integers called Qwfm_data. The variable `Nbytes` is set to equal the number of bytes in the I waveform data. This should be twice the number of integers in `Iwfm_data`, since an integer is 2 bytes. Input integers must be between 0 and 16383.

In the `Output` commands, `USING “#,K”` formats the data. The pound symbol (#) suppresses the automatic EOL (End of Line) output. This allows multiple output commands to be concatenated as if they were a single output. The “K” instructs HP BASIC to output the following numbers or strings in the default format.

```
5 Npoints=20
10 ALLOCATE INTEGER Iwfm_data(1:Npoint),Qwfm_data(1:Npoints)
15 DEG
20 FOR I=1 TO Npoints
25  Iwfm_data(I)=INT(8191*(COS(I*360/Npoints))+8192)
30  Qwfm_data(I)=INT(8191*(COS(I*360/Npoints))+8192)
35 NEXT I
40 Nbytes=2*Npoints
45 Assign @ESG to 719;FORMAT ON
50 Assign @ESGb to 719; FORMAT OFF
55 Nbytes$=VAL$(Nbytes)
60 Ndigits$=(LEN$(Nbytes$))
65 OUTPUT @ESG USING "#,K";":MMEM:DATA ""ARBI:<name>"",#"
70 OUTPUT @ESG USING "#,K";Ndigits$
75 OUTPUT @ESG USING "#,K";Nbytes
80 OUTPUT @ESGb;Iwfm_data(*)
85 OUTPUT @ESG;
90 OUTPUT @ESG USING "#,K";":MMEM:DATA ""ARBQ:<name>"",#"
95 OUTPUT @ESG USING "#,K";Ndigits$
100 OUTPUT @ESG USING "#,K";Nbytes
105 OUTPUT @ESGb;Qwfm_data(*)
110 OUTPUT @ESG;";"
115 Assign @ESGb to *
120 Assign @ESG to *
125 END
```

Table 9-2 Program Comments

5:	Sets the number of points in the waveform.
10:	Defines arrays for I and Q waveform points. Sets them to be integer arrays.
15:	Sets HP BASIC to use degrees for cosine and sine functions.
20:	Sets up loop to calculate waveform points.
25:	Calculates I waveform points.
30:	Calculates Q waveform points.
35:	End of loop.
40:	Calculates number of bytes in I or Q waveform.
45:	Opens an I/O path to the signal generator using GPIB. 7 is the address of the GPIB card in the computer, and 19 is the address of the signal generator. This I/O path is used to send ASCII data to the signal generator.
50:	Opens an I/O path for sending binary data to the signal generator.
55:	Creates an ASCII string representation of the number of bytes in the waveform.
60:	Finds the number of digits in Nbytes.
65:	Sends the I waveform SCPI download-to-ARBI command and the beginning of the ASCII header for the data. The variable `<name>` is the waveform name that will be used in the signal generator.
70 to 75:	Sends the rest of the ASCII header.

80:	Sends the binary data. Note that ESGb is the binary I/O path.
85:	Sends an End-of-Line to terminate the transmission.
90 to 110:	Executes same commands for the Q waveform. The variable <name> that appears in program line 100 must be identical to that in program line 50.
115 to 120:	Closes the connections to the signal generator.
125:	End the program.

Playing a Downloaded Waveform

The following section describes how to output a downloaded waveform. The following commands are for waveforms downloaded to ARB memory.

1. Load and select the downloaded I and Q files for playback.

Via the Front Panel Interface: If you have downloaded the data to ARB memory, this step is unnecessary because the files already reside in the playback memory. If the files were downloaded to non-volatile ARB memory (NVARB) press **Mode** > **Arb Waveform Generator** > **Dual ARB** > **Select Waveform** > **Waveform Segments** > **Load Store**, highlight the downloaded file in the NVARB catalog and press **Load Segment From NVARB Memory**. This loads your downloaded data into ARB memory for playback. If the proper waveform is not highlighted, use the cursor to highlight the waveform and press **Select Waveform**. This selects the proper waveform for playback.

Via the Remote Interface: If you have downloaded the data to ARB memory, this step is unnecessary because the files already reside in the playback memory. If the files were downloaded to non-volatile ARB memory (NVARB) press send the following SCPI commands:

```
[:SOURce]:RADio:ARB:WAVeform "ARBI:"<waveform name>""
[:SOURce]:RADio:ARB:WAVeform "ARBQ:"<waveform name>""
```

These commands load ARB memory with the I and Q files of your downloaded data. This loads your downloaded data into ARB memory for playback.

2. Generate and output the waveform.

Via the Front Panel Interface: Press **ARB Off On** until On is highlighted to generate the waveform. If necessary, press **Mod On/Off** until the display reads MOD ON (modulation is set to on at preset) and **RF On/Off** to activate the RF output (RF output is set to off at instrument preset). The waveform is now modulating the carrier frequency at the RF output.

Via the Remote Interface: To generate the waveform, send the following SCPI command:

```
[:SOURce]:RADio:ARB[:STATe] ON
```

To activate the modulation, send the following command:

:OUTPut:MODulation[:STATe] ON

To activate the RF output, send the following command:

:OUTPut[:STATe] ON

The waveform is now modulating the carrier frequency at the RF output.

10 Data Transfer Troubleshooting

This chapter contains information for troubleshooting remote data transfer problems:

- list of symptoms and possible causes to typical problems encountered while downloading data to the signal generator
- reminders regarding special considerations, file requirements and data limitations
- tips on creating data, transferring data, data application and memory usage

This chapter is divided into four sections by data transfer method:

Direct PRAM Download Problems

Table 10-1 Direct-to-PRAM Download Trouble - Symptoms and Causes

Symptom	Possible Cause
The pattern is transmitted interspersed with random unwanted data.	**Pattern reset bit not set.** **Insure that the pattern reset bit (bit 7, value 128) is set on the last byte of your downloaded data.**
ERROR -223, Too much data.	**PRAM download exceeds the size of PRAM memory.** **Either use a smaller pattern or get more memory by ordering the appropriate hardware option.**

Data Requirement Reminders

To avoid direct-download-to-PRAM problems, the following conditions *must* be met:

1. The data must be in binary form.
2. For every bit of modulation data (bit 0), you must provide seven bits of control information (bits 1-7).

Table 10-2 Data and Control Bit Definitions

Bit	Function	Value	Comments
0	Data	0/1	This bit is the data to be modulated. This bit is a “unspecified” when burst (bit 2) is set to 0.
1	Reserved	0	Always 0.
2	Burst	0/1	Set to 1 = RF on. Set to 0 = RF off. For non-bursted, non-TDMA systems, this bit is set to 1 for all memory locations, leaving the RF output on continuously. For framed data, this bit is set to 1 for *on* timeslots and 0 for *off* timeslots
3	Reserved	0	Always 0.
4	Reserved	1	Always 1.
5	Reserved	0	Always 0.
6	Event 1 Output	0/1	Setting this bit to 1 causes a level transition at the EVENT 1 BNC connector. This can be used for many functions. For example, as a marker output to trigger external hardware when the data pattern has restarted, or to create a data-synchronous pulse train by toggling this bit in alternate addresses.
7	Pattern Reset	0/1	Set to 0 = continue to next sequential memory address. Set to 1 = end of memory and restart memory playback. This bit is set to 0 for all bytes except the last address of PRAM. For the last address (byte) of PRAM, it is set to 1 to restart the pattern.

User File Download Problems

Table 10-3 User File Download Trouble - Symptoms and Causes

Symptom	Possible Cause
No data modulated.	Not enough data to fill a single timeslot. If a user file does not completely fill a single timeslot, the firmware will not load any data into the timeslot. For example, if a timeslot’s data field should contain 114 bits, and only 100 bits are provided in the user file, no data will be loaded into the data field of the timeslot. Therefore, no will be detected at the RF output.
Some data modulated, some data missing at RF output.	Data does not completely fill an integer number of timeslots. If a user file fills the data fields of more than one timeslot in a continuously repeating framed transmission, the user file will be restarted after the last timeslot containing completely filled data fields. For example, if the user file contains enough data to fill the data fields of 3.5 timeslots, firmware will load 3 timeslots with data and restart the user file after the third timeslot. The last 0.5 timeslot worth of data will never be modulated.

Data Requirement Reminders

To avoid user file data download problems, the following conditions *must* be met:

1. The user file selected must entirely fill the data field of each timeslot.
2. For Binary memory catalog downloads, the user file must be a multiple of 8 bits, so that it can be represented in ASCII characters.
3. Available PRAM must be large enough to support both the data field bits and the framing bits.

Requirement for Continuous User File Data Transmission

“Full Data Field” Requirements

If a user file does not *completely* fill a single timeslot, the firmware does not load *any* data into that timeslot. For example, if a timeslot’s data field should contain 114 bits, and only 100 bits are provided in the user file, no data is loaded into the timeslot data field, and no data is transmitted at the RF output.

To solve this problem, add bits to the user file until the it completely fills the data field of the active protocol.

“Integer Number of Timeslots” Requirement for Multiple-Timeslots

If a user file fills the data fields of more than one timeslot in a continuously repeating framed transmission, the user file is restarted after the last timeslot containing completely filled data fields. For example, if the user file contains enough data to fill the data fields of 3.5 timeslots, firmware loads 3 timeslots with data and restart the user file after the third timeslot. The last 0.5 timeslot worth of data is never modulated.

To solve this problem, add or subtract bits from the user file until it completely fills an integer number of timeslots

“Multiple-of-8-Bits” Requirement

For downloads to the binary memory catalog, user file data must be downloaded in multiples of 8 bits, since SCPI specifies data in 8-bit bytes. Therefore, if the original data pattern’s length is not a multiple of 8, you may need to:

- Add additional bits to complete the ASCII character,
- replicate the data pattern to generate a continuously repeating pattern with no discontinuity, or
- truncate the remaining bits.

NOTE The “multiple-of-8-bits” data length requirement (for binary memory catalog downloads) is in *addition* to the requirement of completely filling the data field of an integer number of timeslots.

The following method can be used to compute the number of data pattern repetitions required in order to form a continuous data stream.

In this example, a modified PN9, 511-bit data pattern is to be applied as the data source for a 114-bit data field in a GSM Normal timeslot.

Set up a spreadsheet containing:

A = number of repetitions of the original data pattern
B = user file length = number of repetitions × original data pattern length
C = Number of characters = user file length ÷ 8 (8 bits-per-character)
D = number of frames = user file length ÷ timeslot data field size (114)
E = total required PRAM = number of frames × number of bits-per-frame (1250 for GSM)

Table 10-4 Data Pattern Repetitions for Continuous Data Stream

A Number of reps	B Data Pattern Length × Repetitions	C Number of Characters (B ÷ 8)	D Number of frames needed to end on a timeslot boundary (B ÷ timeslot data field size)	E Total PRAM required (D × number of bits-per-frame)
1	511	63.88	4.48	5,603.07
2	1,022	127.75	8.96	11,206.14
3	1,533	191.63	13.45	16,809.21
4	2,044	255.50	17.93	22,412.28
5	2,555	319.38	22.41	28,015.35
6	3,066	383.25	26.89	33,618.42
7	3,577	447.13	31.38	39,221.49
8	4,088	511	35.86	44,824.56
9	4,599	574.88	40.34	50,427.63
...	...	...	...	...
455	232,505	29,063.13	2,039.52	2,549,396.92
456	233.016	29,127	2,044	2,555,000

The first row where both columns C and D are integers (the shaded row at the bottom of the table) is the minimum number of repetitions required to transmit the user file without discontinuity. In this example, in order to correctly generate the modified PN9 and download it to a user file, the user file must contain 456 repetitions of the 511-bit pattern. 233,016 total bits will be downloaded to the signal generator, for a total of 29,127 characters.

“Pattern RAM Memory Space” Requirement

It is possible to exhaust the available PRAM with a large, continuous user file transmitted across a large number of frames.

In the previous example, selecting the 233,016-bit user file as the data source for the GSM Normal timeslot will cause the firmware to compute 2,044 frames of data, filling 2,555,000 bytes of PRAM space. Option UN9 (8 Mbyte pattern RAM) is required for this configuration. Trying to load this data on an Option UN8 (1 Mbyte pattern RAM) signal generator will cause an error, because there is not enough PRAM to hold the required data.

If PN11 was used instead of PN9, 456 repetitions of the data pattern would require a 933,432-bit user file, requiring 8,188 frames and 10,235,000 bytes of PRAM. Because the size of this data exceeds the limits of Option UN9 (8 Mbyte PRAM), you would need to supply a file this size via the external DATA connector.

Using Externally-Generated, Real-Time Data for Large Files

If the data fields absolutely must be continuous data streams, and the size of the data exceeds the available PRAM, real-time data and synchronization can be supplied by an external data source to the front-panel DATA, DATA CLOCK, and SYMBOL SYNC connectors. This data can be continuously transmitted, or can be framed by supplying a data-synchronous burst pulse to the EXT1 INPUT connector on the front panel. Additionally, the external data can be multiplexed into the internally-generated framing.

NOTE This chapter does not cover externally-generated real-time data. See the ESG Family Signal Generator User's Guide for details and timing diagrams.

User FIR Filter Coefficient File Download Problems

Table 10-5 User FIR File Download Trouble - Symptoms and Causes

Symptom	Possible Cause
ERROR -321, Out of memory	There is not enough space in the memory catalog for the FIR coefficient file being downloaded. To solve the problem, either reduce the file size of the FIR file or delete unnecessary files from the memory catalog.
ERROR -223, Too much data.	User FIR filter has too many symbols. Option UN8 cannot use a filter that has more than 32 symbols (24 symbols maximum for Option UND). You may have specified an incorrect oversample ratio in the filter table editor.

Data Requirement Reminders

To avoid user FIR filter coefficient data download problems, the following conditions *must* be met:

1. Data must be in ASCII format.
2. Downloads must be in list format.
3. Filters containing more symbols than the hardware allows (32 for Option UN8 and 24 for Option UND) will not be selectable for that configuration.

ARB Waveform Data Download Problems

Table 10-6 I/Q Waveform Data Download Trouble - Symptoms and Causes

Symptom	Possible Cause
ERROR 224, Text file busy.	Attempting to download a waveform that has the same name as the waveform currently being played by the signal generator. To solve the problem, either change the name of the waveform being downloaded or turn off the ARB.
ERROR -321, Out of memory.	There is not enough space in the ARB memory for the waveform file being downloaded. To solve the problem, either reduce the file size of the waveform file or delete unnecessary files from ARB memory.

Data Requirement Reminders

To avoid I/Q waveform data download problems, the following six conditions *must* be met:

1. Data must be in unsigned, offset (binary) format.
2. Data must be in 2-byte integers, ordered MSB first to LSB last.
3. Input integers must be between 0 and 16383.
4. Each I and Q waveform must have at least 16 points (a point equals an integer).
5. Each I and Q waveform must contain an even number of points.
6. I and Q waveforms should be the same length. (Different length I and Q waveforms are allowed. The shorter waveform will give a 0V output from its end up to the length of the longer waveform.)

Index