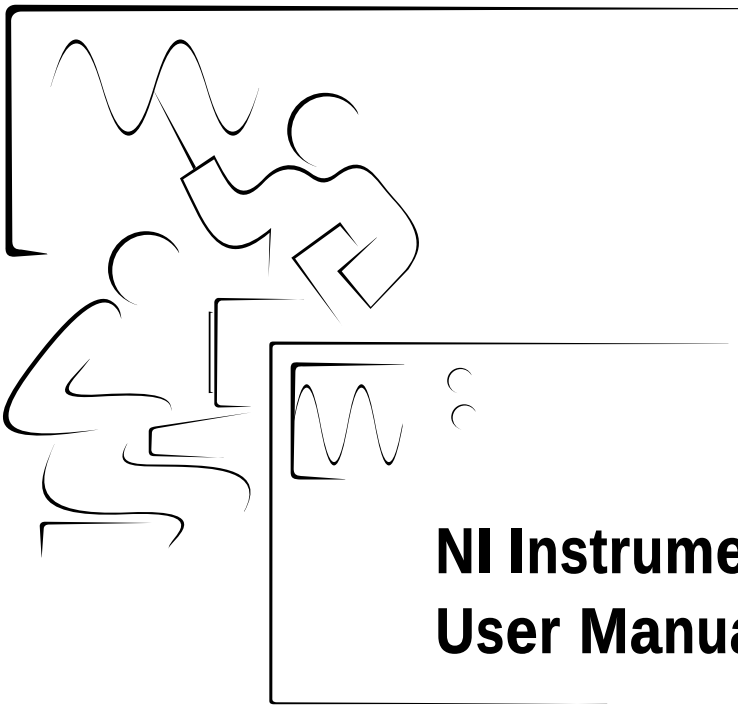


NI Instrument Simulator User Manual

August 1996 Edition
Part Number 320638B-01

Internet Support

GPIB: gpib.support@natinst.com
DAQ: daq.support@natinst.com
VXI: vxi.support@natinst.com
LabVIEW: lv.support@natinst.com
LabWindows: lw.support@natinst.com
HiQ: hiq.support@natinst.com
VISA: visa.support@natinst.com

E-mail: info@natinst.com
FTP Site: ftp.natinst.com
Web Address: http://www.natinst.com

Bulletin Board Support

BBS United States: (512) 794-5422 or (800) 327-3077
BBS United Kingdom: 01635 551422
BBS France: 1 48 65 15 59

FaxBack Support

(512) 418-1111

Telephone Support (U.S.)

Tel: (512) 795-8248
Fax: (512) 794-5678

International Offices

Australia 03 9 879 9422, Austria 0662 45 79 90 0, Belgium 02 757 00 20,
Canada (Ontario) 519 622 9310, Canada (Québec) 514 694 8521, Denmark 45 76 26 00,
Finland 90 527 2321, France 1 48 14 24 24, Germany 089 741 31 30, Hong Kong 2645 3186,
Italy 02 413091, Japan 03 5472 2970, Korea 02 596 7456, Mexico 95 800 010 0793,
Netherlands 0348 433466, Norway 32 84 84 00, Singapore 2265886, Spain 91 640 0085,
Sweden 08 730 49 70, Switzerland 056 200 51 51, Taiwan 02 377 1200, U.K. 01635 523545

National Instruments Corporate Headquarters

6504 Bridge Point Parkway Austin, TX 78730-5039 Tel: (512) 794-0100

Important Information

Warranty

The NI Instrument Simulator is warranted against defects in materials and workmanship for a period of two years from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this manual is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Trademarks

LabVIEW® and The Software is the Instrument® are trademarks of National Instruments Corporation.

Product and company names listed are trademarks or trade names of their respective companies.

WARNING REGARDING MEDICAL AND CLINICAL USE OF NATIONAL INSTRUMENTS PRODUCTS

National Instruments products are not designed with components and testing intended to ensure a level of reliability suitable for use in treatment and diagnosis of humans. Applications of National Instruments products involving medical or clinical treatment can create a potential for accidental injury caused by product failure, or by errors on the part of the user or application designer. Any use or application of National Instruments products for or involving medical or clinical treatment must be performed by properly trained and qualified medical personnel, and all traditional medical safeguards, equipment, and procedures that are appropriate in the particular situation to prevent serious injury or death should always continue to be used when National Instruments products are being used. National Instruments products are NOT intended to be a substitute for any form of established process, procedure, or equipment used to monitor or safeguard human health and safety in medical or clinical treatment.

FCC/DOC Radio Frequency Interference Compliance

This equipment generates and uses radio frequency energy and, if not installed and used in strict accordance with the instructions in this manual, may cause interference to radio and television reception. Classification requirements are the same for the Federal Communications Commission (FCC) and the Canadian Department of Communications (DOC). The FCC/DOC classification is indicated on a label on the devices. Look at the product to determine if your equipment has been tested and found to comply with Class A or Class B, then read the appropriate information below regarding the compliance of your product:

Class A

Federal Communications Commission

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Canadian Department of Communications

This Class A digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe A respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

Class B

Federal Communications Commission

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Canadian Department of Communications

This Class B digital apparatus meets all requirements of the Canadian Interference-Causing Equipment Regulations.

Cet appareil numérique de la classe B respecte toutes les exigences du Règlement sur le matériel brouilleur du Canada.

FCC Notices to User

Changes or modifications not expressly approved by National Instruments could void the user's authority to operate the equipment under the FCC Rules.

This device complies with the FCC rules only if used with shielded interface cables of suitable quality and construction. National Instruments used such cables to test this device and provides them for sale to the user. The use of inferior or nonshielded interface cables could void the user's authority to operate the equipment under the FCC rules.

If necessary, consult National Instruments or an experienced radio/television technician for additional suggestions. The following booklet prepared by the FCC may also be helpful: Interference to Home Electronic Entertainment Equipment Handbook. This booklet is available from the U.S. Government Printing Office, Washington, DC 20402.

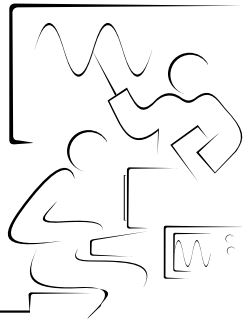
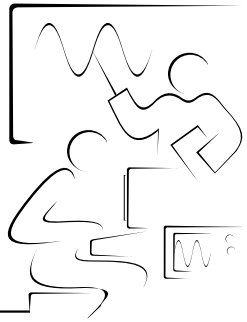


Table of Contents

NI Instrument Simulator Command Set

NI Instrument Simulator Command Set



The NI Instrument Simulator

The NI Instrument Simulator is a new way to learn GPIB communication protocol. Because the simulator can function as both a digitizing oscilloscope and a digital multimeter, it is flexible enough to be used in a classroom or industry setting. The Simulator is fully compatible with 488.2SRQ protocol and also supports a subset of SCPI-like commands. You can also use VISA to communicate with the Instrument Simulator through LabWindows/CVI or LabVIEW.

The NI Instrument Simulator is ideal for debugging or teaching purposes. Instead of carrying around your instruments to debug your GPIB system, use the Instrument Simulator and save a lot of time and effort. The NI Instrument Simulator is also used in conjunction with National Instruments customer education courses.

Data Formats

Waveform Format

The Simulator returns a 128-point waveform in either ASCII or binary. ASCII waveforms are preceded by the header CURVE. Binary waveforms are preceded by a pound sign (#) and the number of bytes that are in the waveform. All waveforms are terminated by a line feed <LF> character.

Floating Point ASCII (Default)

```
CURVE<space>num0,num1,...,num127<LF>
```

8-bit Unsigned Binary

```
#3128<Byte 0><Byte 1>...Byte<127><LF>
```

16-bit Signed Binary

```
#3256<MSB 0><LSB 0><MSB 1><LSB 1>...<MSB 127><LSB
127><LF>
```

Floating-Point Number Format

```
[+][-]1.2345E[+][-]0
```

Simulator Commands

The Simulator uses SCPI-like commands. The commands are shown in long form; however, the Simulator accepts only the short form of the command. In other words, send only the part of the command that is in uppercase characters. You can send multiple commands to the Simulator by separating them with a semicolon (;).

Address Command

`SADDRess primary, secondary`		Sets the address (power-on default—switch setting)
Example:	`SADDR 2`	Set the address to 2
	`SADDR 3, 4`	Set the primary address to 3 and the secondary address to 4

Waveform Format Commands

These commands format how the waveform data is returned by the Simulator.

`FORMat:DATA`	`ASCii`	Floating point (Default)
	`INTeger,8`	8-bit unsigned binary
	`INTeger,16`	16-bit signed binary
`FORMat:DATA?`		Returns the current waveform format

The following command changes the order of the bytes returned by `INTeger,16` encoding.

`FORMat:BORDer`	`NORMal`	Low byte first (Default)
	`SWAPped`	High byte first
`FORMat:BORDer?`		Returns the current format of the byte order

Example: `FORM:DATA INT,16` Set the waveform format as 16-bit integers

`FORM:DATA?` Query the current waveform format. For example, if the command was issued after the preceding command, it would return `FORM:DATA INT,16<LF>`

Waveform Generation Commands

These commands generate a 128-point waveform of the specified type. The number of cycles in the waveform is random. It can take 5 to 15 seconds to generate the waveform, depending on the format and type of the waveform. Typically, ASCII waveforms take longer than integer waveforms.

`SOURce:FUNCtion`	`SINusoid`	Sine waveform (Default)
	`SQUare`	Square waveform
	`NOISe`	Noisy sine waveform
	`RANDom`	Random noise waveform
	`PCHirp`	Chirp waveform

`SOURce:FUNCtion?` Returns the current waveform type

Example: `SOUR:FUNC SIN` Generate a sinusoid waveform

`SOUR:FUNC?` Query the current waveform type. For example, if the command was issued after the preceding command, it would return `SOUR:FUNC SIN<LF>`

Waveform Query Commands

`SENSe:DATA?` Returns the waveform data in the format specified by the waveform format commands

`SENSe:VOLTage:RANGe:OFFSet?` Returns the Y offset for the waveform in ASCII floating point

`SENSe:VOLTage:RANGe?` Returns the Y multiplier for the waveform in ASCII floating point

`SENSe:SWEep:TIME?`	Returns the X increment (1E-3) in ASCII floating point
`SENSe:VOLTage:HEADer?`	Returns all of the waveform scaling information in the format `OFFSET=x.xxxxE+x, RANGE=x.xxxxE+x, TIME=1E-3<LF>`

For integer-formatted waveforms, the offset and range are used to scale the raw integer data—for example,

```
scaled.point(i) = (waveform.point(i) + offset) * range
```

Example:	`SENS:DATA?`	Query Simulator for the waveform
	`SENS:VOLT:HEAD?`	Query Simulator for the waveform scaling information

"Multimeter Configuration" Commands

These commands simulate the operation of a meter. They return one value in ASCII floating point.

`MEASure:DC?`		Returns a random value between 0 to +x in floating point ASCII. The range of x depends on the `CONFigure:DC` command
`CONFigure:DC DEFault`		`MEASure:DC?` returns a number between 0 and 10
	`MIN`	`MEASure:DC?` returns a number between 0 and 1
	`MAX`	`MEASure:DC?` returns a number between 0 and 100
`CONFigure:DC?`		Returns the current configuration setting

Example:	`CONF:DC MAX`	Set the maximum range
	`CONF:DC?`	Query the current DC range. For example, if the command was issued after the command above, it would return `CONF:DC MAX<LF>`
	`MEAS:DC?`	Queries one value, for example `1.2308<LF>`

Other Commands

`*IDN?`	Returns `National Instruments GPIB and Serial Device Simulator Rev B.x <LF>`
`*RST`	Resets the Simulator to its default state
`*TRG`	Triggers the Simulator and returns one random reading (same as `MEAS:DC?`)
`*TST?`	Simulates testing the Simulator. Returns `OK`
`*OPC`	Sets the operation complete bit in the Standard Event Status Register (ESR)
`*OPC?`	Returns the value of the OPC bit in the ESR register
`*ESR?`	Returns value of Standard Event Status register as specified by `FORM:SREG`

Figure 1 illustrates the bits defined by the Simulator for the ESR register—bit 7 (Power On), bit 5 (Command Error), and bit 0 (Operation Complete). Bit 7 is set when the Simulator is powered on; bit 5 is set when the Simulator receives an invalid command; bit 0 is set when the Simulator receives the `*OPC` command. You can use the `*ESR?` command to query the value of the ESR register. The value returned is in either ASCII or HEX, as specified by the `FORMat:SREGister` command. The ESR register is cleared after you read it.

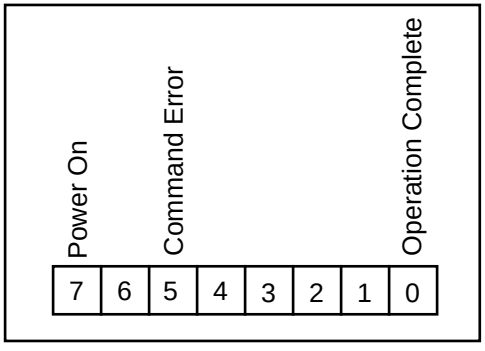


Figure 1. Three ESR Bits Set by the Simulator

`*ESE 0x##` (zero, x, mask in hex)	Sets value of Standard Event Status Enable register
`*ESE?`	Returns value of Standard Event Status Enable register as specified by `FORM:SREG`
`*STB?`	Returns value of Status Byte register as specified by `FORM:SREG`
`*SRE 0x##` (zero, x, mask in hex)	Sets value of Service Request Enable register
`*SRE?`	Returns value of Service Request Enable register as specified by `FORM:SREG`
`*WAI`	Does not do anything; included to make the Simulator IEEE 488.2 compatible
`FORMat:SREGister ASCii`	Specifies the output of ESR, ESE, STB, and SRE registers as an ASCII string (default)
`HEX`	Specifies the output of ESR, ESE, STB, and SRE registers in hex
`FORMat:SREGister?`	Returns the current format of the registers
`SYStem:HELP?`	Returns a list of all of the commands. Refer to *Command Summary* section.

Command Summary

```
SADDR
FORM:DATA ASC | INT,8 | INT,16 (?)
FORM:BORD NORM | SWAP (?)
SOUR:FUNC SIN | SQU | RAND | PCH (?)
SENS:DATA?
SENS:VOLT:RANG:OFFS?
SENS:VOLT:RANG?
SENS:SWE:TIME?
MEAS:DC?
CONF:DC MIN | MAX | DEF (?)
*IDN?
*RST
*TRG
*TST?
*OPC
*OPC?
*ESR?
*ESE 0x##
*ESE?
*STB?
*SRE 0x##
*SRE?
*WAI
FORM:SREG ASC | HEX (?)
SYS:HELP?
```

|— separates options for the command
(?)—indicates the command can be used to query the current state

Short Form GPIB Commands

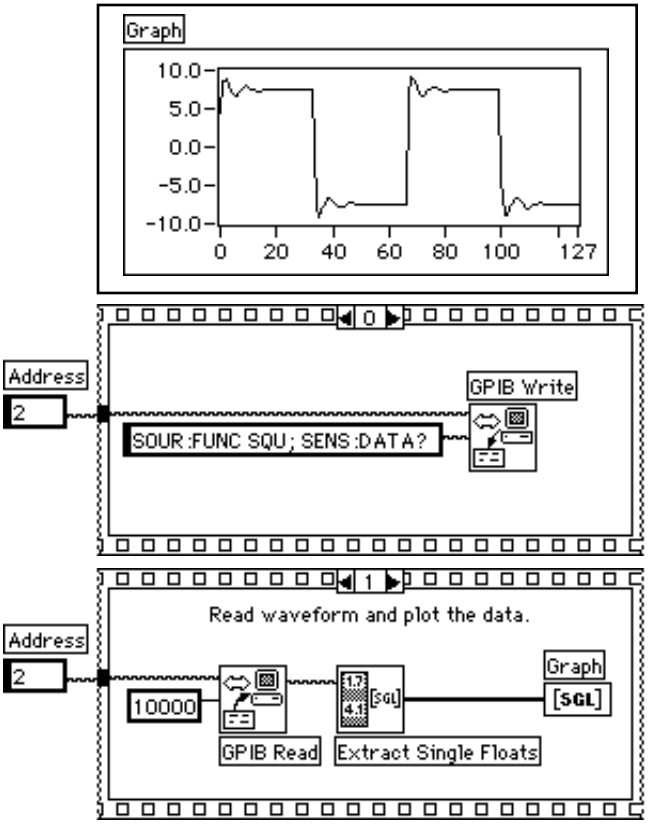
The new Simulator EPROM supports the following commands for compatibility with the older EPROM. However, if multiple commands are sent together, they must be separated using a semicolon (;).

`E0xh0`	(E zero, x, mask in hex) Causes the box to assert SRQ whenever it has finished generating data in response to a W command. The serial poll status is specified in h0.
`E0x0`	(E zero, x, zero) Disables asserting SRQ
`G0`	Output data as 2-byte integers
`G1`	Output data as ASCII floats separated by a comma
`G2`	Output data as ASCII floats separated by a comma
`W1`	Output a noisy square wave
`W2`	Output a sine wave
`W3`	Output a noisy sine wave
`W4`	Output random data
`W5`	Output a chirp waveform
`Od0`	(Letter O) Output d0 random 2-byte integers one at a time

LabVIEW Examples

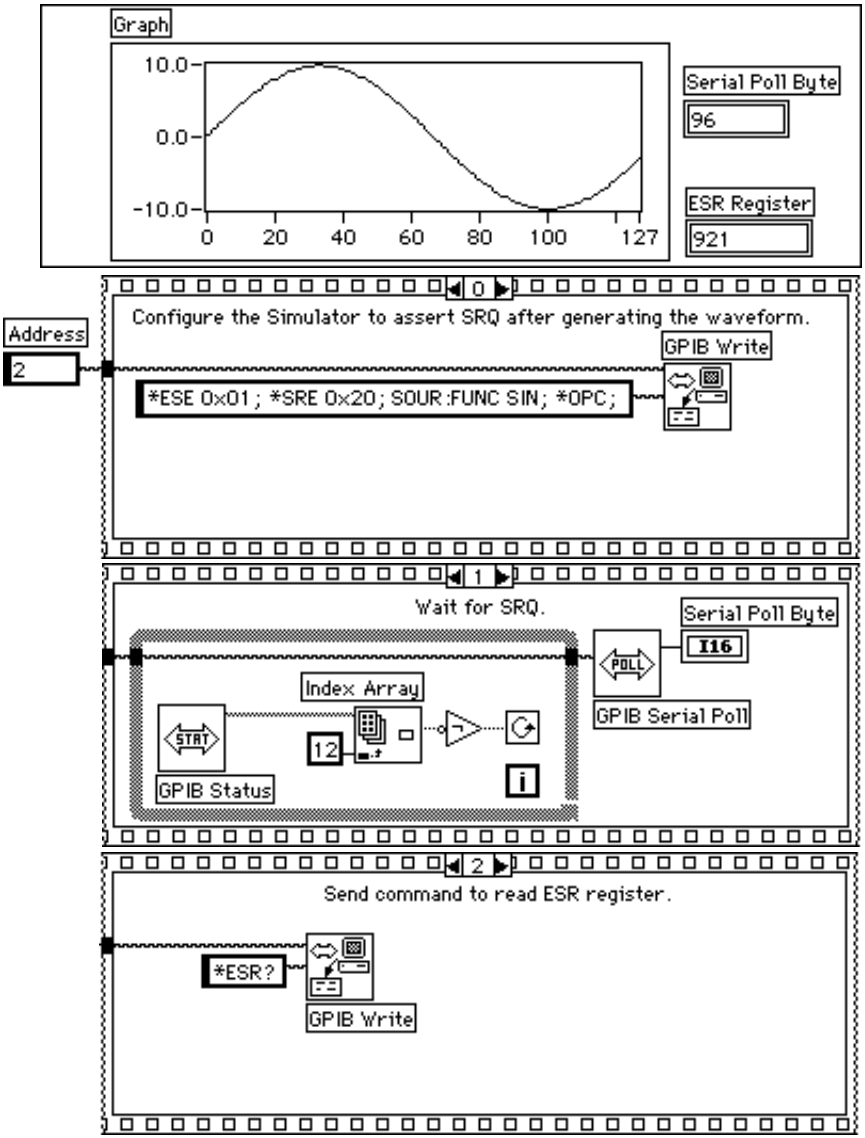
Example 1

The following LabVIEW example shows how to set up the Simulator to generate a square waveform, read the waveform, and plot the waveform on a graph.

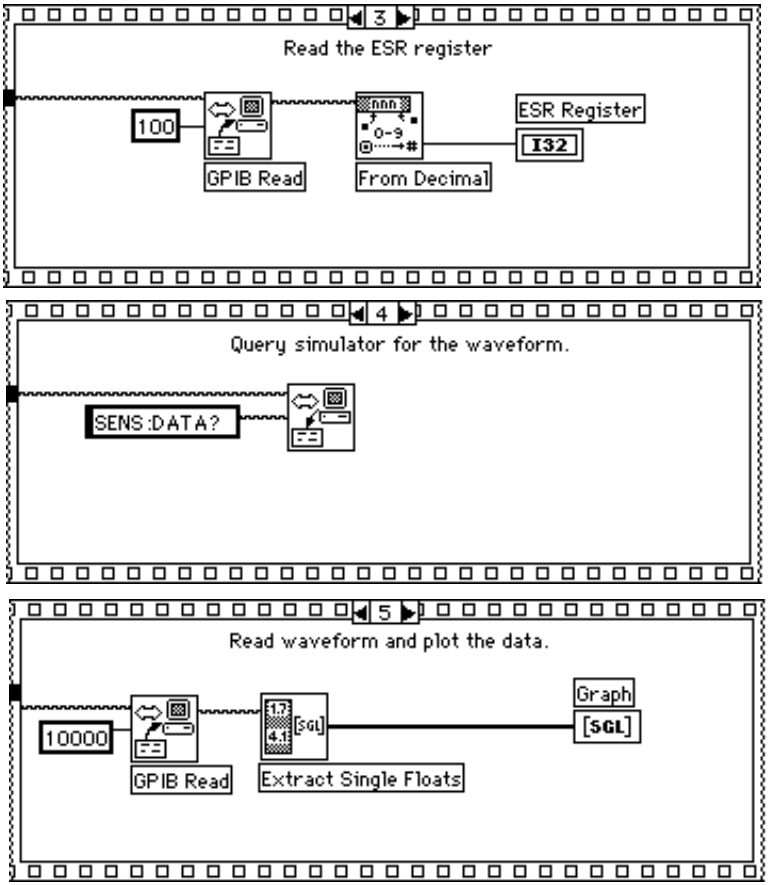


Example 2

The following LabVIEW example shows how to set up the Simulator to assert an SRQ after it generates a sine waveform, read the waveform, and plot the waveform on a graph.

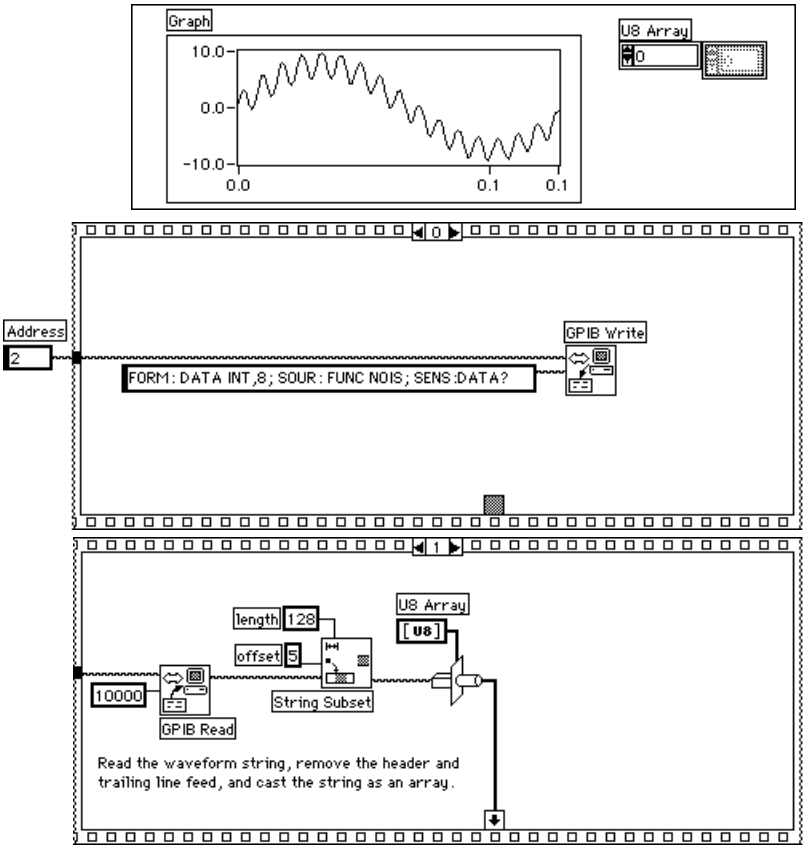


(Example continues on the next page.)

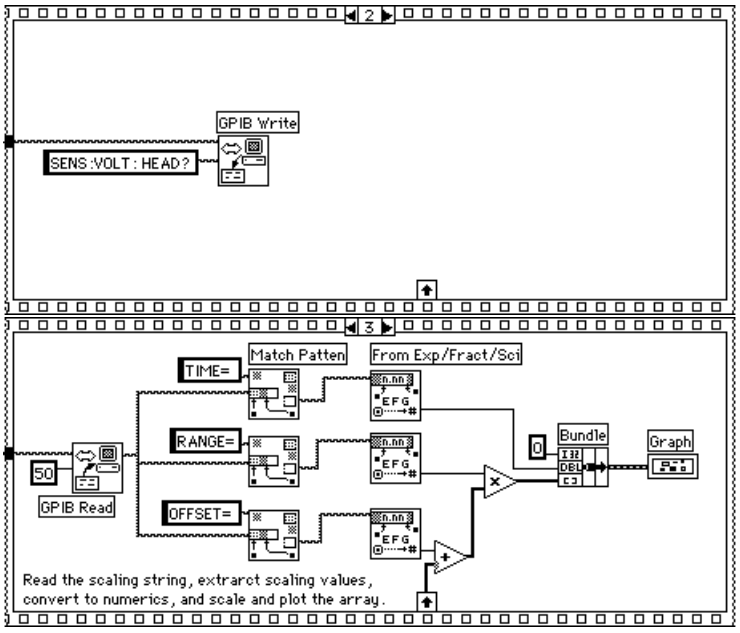

3
Read the ESR register
100
GPIB Read
From Decimal
ESR Register
I32
4
Query simulator for the waveform.
SENS:DATA?
5
Read waveform and plot the data.
10000
GPIB Read
Extract Single Floats
Graph
[SGL]

Example 3

The following LabVIEW example shows how to set up the Simulator to generate a noisy sine waveform in binary format, read the waveform, scale the waveform, and plot the waveform on a graph.



(Example continues on the next page.)


2
GPIB Write
SENS:VOLT:HEAD?
3
Match Patten
From Exp/Fract/Sci
TIME=
RANGE=
OFFSET=
50
GPIB Read
0
Bundle
Graph
Read the scaling string, extract scaling values,
convert to numerics, and scale and plot the array.

LabWindows/CVI Examples

Example 1

```
/*The following example shows how to set up the NI Instrument Simulator to
generate a square waveform, read the waveform, and plot it on a graph.*/

/*To generate another type of waveform, substitute the command in for SQU in
the ibwrt statement. For instance, NOIS for SQU will generate a noisy
waveform*/

#include <userint.h>
#include <formatio.h>
#include <gpib.h>

char buffer[2000];
double waveform[2000];
int ud0, ud1;

int main (int argc, char *argv[])
{
   /* initializes the gpib board */
   ud0 = ibfind ("gpib0");

   /* sets the board as controller in charge */
   ibsic (ud0);
   /* opens and initializes the device */
   ud1 = ibfind ("DEV3");

   /* writes the string */
   ibwrt (ud1, "SOUR:FUNC SQU; SENS:DATA?", 26);

   /* reads the data from the device */
   ibrd (ud1, buffer, 2000);

  /* Discards the header and converts ASCII data to a floating-point array*/
   Scan (buffer, "%s[i6]>%250f[x]", waveform);

   /* Plotting the data*/
   YGraphPopup ("Waveform Plot", waveform, 130, VAL_DOUBLE);

return 0;
}
```


Example 2

```
/*The following example shows how to set up the simulator to assert an SRQ
after it generates a sine wave, read the waveform and plot it on a graph*/

#include <formatio.h>
#include <userint.h>
#include <gpib.h>


int main (int argc, char *argv[])
{
   char buffer[2000];
   double waveform[2000];
   int ud0, ud1;
   static char SPR;

   /*Initializes the gpib board*/
   ud0 = ibfind ("GPIB0");

   /*Sets the board as controller in charge*/
   ibsic (ud0);

   /*Opens and initializes the device*/
   ud1 = ibfind ("DEV3");

   /*Changes the software configuration parameters*/
   ibconfig (ud0, IbcAUTOPOLL, 0);

   /*Writes data to the device*/
   ibwrt (ud1, "*ESE 0x01; *SRE 0x20; SOUR:FUNC SIN; *OPC", 41);

   /*Waiting for SRQ line to be asserted*/
   ibwait (ud0, SRQI);

   /*Conducting a serial poll*/
   ibrsp (ud1, &SPR);

   /*Writes for information on the Event
     Status Register*/
   ibwrt (ud1, "*ESR?", 5);

   /*Reads the value of the Event Status Register*/
```

```
   ibrd (ud1, buffer, 2000);

   /*Requests the waveform data*/
   ibwrt (ud1, "SENS:DATA?", 10);

   /*Reads the sine wave data*/
   ibrd (ud1, buffer, 2000);

   /*Discarding header and converting to
     floating point*/
   Scan (buffer, "%s[i6]>%128f[x]", waveform);

   /*Plots the returned sine wave*/
   YGraphPopup ("Waveform Plot", waveform, 128, VAL_DOUBLE);

return 0;
}
```

Documentation Comment Form

National Instruments encourages you to comment on the documentation supplied with our products. This information helps us provide quality products to meet your needs.

Title: *NI Instrument Simulator User Manual*

Edition Date: August 1996

Part Number: 320638B-01

Please comment on the completeness, clarity, and organization of the manual.

_____

_____

_____

_____

_____

_____

_____

If you find errors in the manual, please record the page numbers and describe the errors.

_____

_____

_____

_____

_____

_____

_____

Thank you for your help.

Name _____

Title _____

Company _____

Address _____

_____

Phone (___)_____ Fax (___) _____

Mail to: Customer Education
National Instruments Corporation
6504 Bridge Point Parkway
Austin, TX 78730-5039

Fax to: Customer Education
National Instruments Corporation
(512) 794-5678