TEK **Programmer Reference**

Part No. 070-7251-00
Product Group 47

THE DSA 601 & DSA 602

DIGITIZING SIGNAL ANALYZERS

Please check for CHANGE INFORMATION at the rear of this manual.

Tektronix®
COMMITTED TO EXCELLENCE

Scans
By
Artek Media

Artek Media
1042 Plummer Cir. SW
Rochester, MN 55902

www.artekmedia.com

"High resolution scans of obsolete technical manuals"

If your looking for a quality scanned technical manual in PDF format please visit our WEB site at www.artekmedia.com or drop us an email at manuals@artekmedia.com and we will be happy to email you a current list of the manuals we have available.

If you don't see the manual you need on the list drop us a line anyway we may still be able to point you to other sources. If you have an existing manual you would like scanned please write for details, This can often be done very reasonably in consideration for adding your manual to our library.

Typically the scans in our manuals are done as follows;

1) Typed text pages are typically scanned in black and white at 300 dpi.
2) Photo pages are typically scanned in gray scale mode at 600 dpi
3) Schematic diagram pages are typically scanned in black and white at 600 dpi unless the original manual had colored high lighting (as is the case for some 70's vintage Tektronix manuals).

If you purchased this manual from us (typically through our Ebay name of ArtekMedia) thank you very much. If you received this from a well-meaning "friend" for free we would appreciate your treating this much like you would "share ware". By that we mean a donation of at least $5-10 per manual is appreciated in recognition of the time (a manual can take as much as 40 hours to reproduce, book, link etc.), energy and quality of effort that went into preserving this manual. Donations via PayPal go to: manuals@artekmedia.com or can be mailed to us the address above.

Thanks

Dave & Lynn Henderson
Artek Media

Instrument Serial Numbers

Each instrument manufactured by Tektronix has a serial number on a panel insert or tag, or stamped on the chassis. The first letter in the serial number designates the country of manufacture. The last five digits of the serial number are assigned sequentially and are unique to each instrument. Those manufactured in the United States have six unique digits. The country of manufacture is identified as follows:

B010000	Tektronix, Inc., Beaverton, Oregon, USA
G100000	Tektronix Guernsey, Ltd., Channel Islands
E200000	Tektronix United Kingdom, Ltd., London
J300000	Sony/Tektronix, Japan
H700000	Tektronix Holland, NV, Heerenveen, The Netherlands

Instruments manufactured for Tektronix by external vendors outside the United States are assigned a two digit alpha code to identify the country of manufacture (e.g., JP for Japan, HK for Hong Kong, etc.).

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

Copyright © Tektronix, Inc., 1989. All rights reserved. Tektronix products are covered by U.S. and foreign patents, issued and pending. The following are registered trademarks:
TEKTRONIX, TEK, TEKPROBE, SCOPEMOBILE and

Centronics® is a registered trademark of Data Computer Corporation.

Epson is a registered trademark of Seiko Epson Corporation.

GPIB-PC is a trademark of National Instruments Corporation.

HP-GL, LaserJet, and ThinkJet are products of Hewlett Packard.

IBM® is a registered trademark of International Business Machines.

Microsoft® is a registered trademark of Microsoft Corporation.

First Printing MAR 1989
Revised MAY 1989

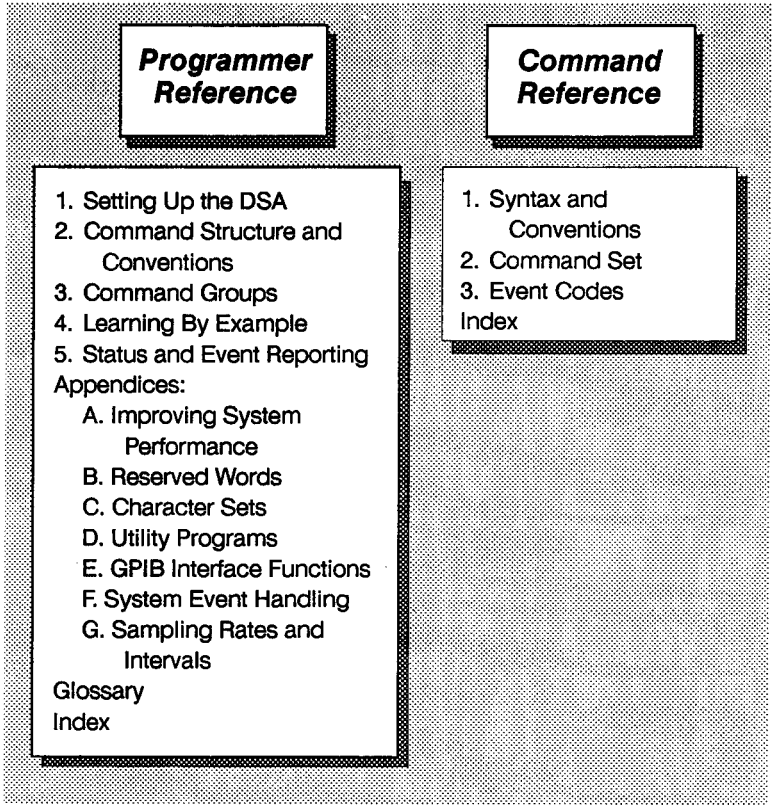
Printed in U.S.A.

About This Manual

This document, along with to the*DSA 601 and DSA 602 Command Reference* provide comprehensive, easily accessible information to aid you in operating your DSA via the General Purpose Interface Bus (GPIB) or the RS-232-C interface.

How the Programmer Manuals are Organized

The following illustration shows how the programmer documentation is organized; a list of related documents is on the next page.



DSA 601 and DSA 602 Programmer Documentation

Note that commands are fully described in the *Command Reference* which accompanies this manual. Event codes are listed in the *Command Reference* but are fully described in this manual.

Related Manuals

You may want to refer to the other manuals that complete the documentation set for this DSA:

- The *DSA 601 and DSA 602 Tutorial* (Tektronix part number 070-7249-00) describes the basic operation of the DSA 601 and DSA 602. Many of the examples in the *DSA 601 and DSA 602 Tutorial* are paralleled in the Learning By Example section of this manual.

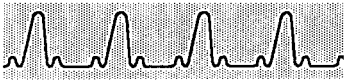
- The *DSA 601 and DSA 602 QuickStart Package* (U.S.A. Tektronix part number 020-1769-00, European 020-1770-00) is a complete learning laboratory, including a signal generating board and a workbook. A videotape for the *DSA 601 and DSA 602 QuickStart Package* is included with your DSA. These show you how to use the power of the DSA to get the types of measurements you need. The *QuickStart Package* is available at no charge, but you need to mail in the postage-paid card included with the DSA.

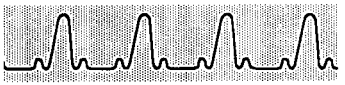
- The *DSA 601 and DSA 602 User Reference* (Tektronix part number 070-7250-00) covers all aspects of front panel operation. Use this manual to quickly gain information about a specific topic, or to get an overview of the menu system.

- The *DSA 601 and DSA 602 Command Reference* (Tektronix part number 070-7250-00) describes the commands used to program the DSA.

- The *DSA 601 and DSA 602 Service Reference* (Tektronix part number 070-7254-00) provides information to maintain and service components of the DSA, and provides a complete board-level description of the DSA operation.

Table of Contents





Setting Up the DSA

This section describes the implementation of each interface on the DSA 601 and DSA 602, shows how to connect your DSA to other instruments that have either a GPIB or an RS-232-C interface, and explains how to set up the DSA's front panel for remote operation.

Connecting the DSA to a GPIB Network

Before connecting devices to the GPIB, you should be aware of some rules concerning GPIB networks, cables, and connectors.

GPIB Interface Requirements

GPIB networks can be connected in any configuration, subject to the following rules:

- No more than 15 devices (including the controller) can be included on a single bus.
- In order to maintain bus electrical characteristics, one device load must be connected every two meters (six feet) of cable length. Generally, each instrument represents one device load on the bus.
- The total cumulative cable length must not exceed 20 meters.
- At least two-thirds of the device loads must be powered on when the network is in operation.
- There must be only one cable path from each device to each other device on the network; loop configurations are not allowed.

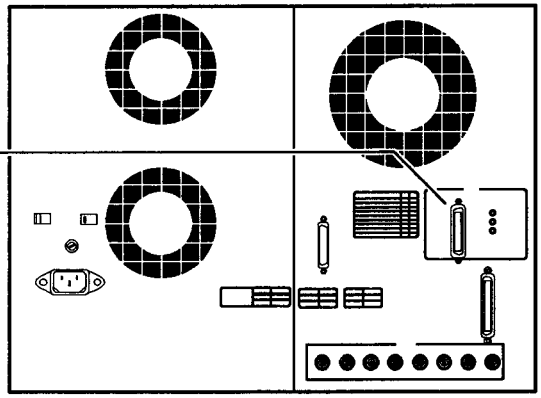
Cables

An IEEE STD 488 GPIB cable (available from Tektronix) is required to connect two GPIB devices.

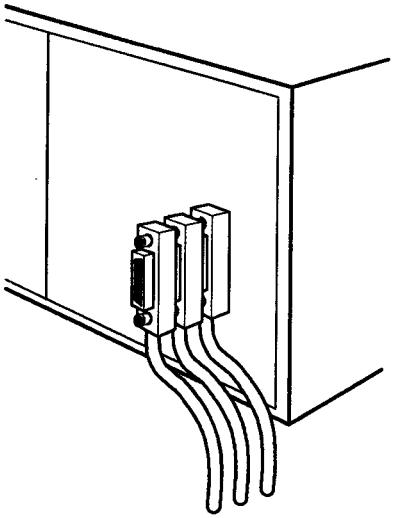
Connector

A 24-pin GPIB connector is located on the rear panel of the oscilloscope. The connector has a D-type shell and conforms to IEEE STD 488. GPIB connectors can be stacked on each other. See the illustration on the following page.

GPIB
Connector



Location of **GPIB** *Connector on Rear Panel*



How **GPIB** *Connectors can be Stacked Together*

Setting Up GPIB Parameters

The following steps tell how to set up the GPIB parameters at the front panel.

☐ Step 1: Press the **UTILITY** major menu button to the right of the display area. The Utility major menu appears in the major menu area at the bottom of the display.

☐ Step 2: Touch the **Page to Utility 2** selector in the major menu area. Touch the **GPIB Parameters** selector in the major menu area. The **GPIB Parameters** pop-up menu appears in the display area, with the following selectors:

- **Mode** sets the DSA to be a talker/listener, to be a talker-only, or to be off the bus. Set the DSA to be a talker-only when the only device to be connected to it is a listen-only device, such as a printer or plotter. Otherwise, set it to be a talker/listener.

The settings for the address and terminator parameters *must* match those of your controller. See the operating manual for your controller to select the appropriate parameters for its GPIB interface.

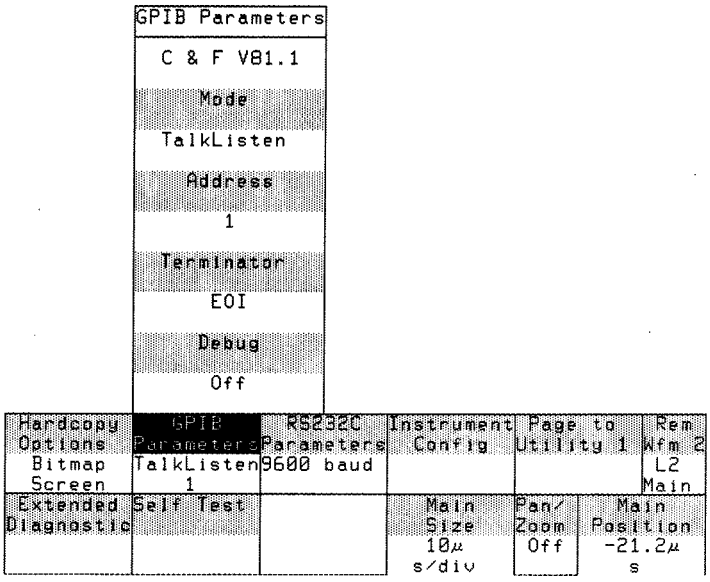
- **Address** sets the primary communication address of the DSA. The address range is 0 to 30.
- **Terminator** sets the method of indicating the end of device-dependent messages sent between the controller and the DSA. The choices are EOI (assert EOI line with transmission of last byte of message) or EOI/LF (send line feed character and assert EOI line with its transmission).

When debug is on, input/output processing is slowed.

- **Debug** specifies whether or not GPIB device-dependent messages (DSA commands) appear at the top of the DSA display.

☐ Step 3: Repeatedly touch the selector for each parameter (except **Address**) until the value you want appears.

☐ Step 4: Touch the **Address** selector to assign the knobs to address selection. Rotate either knob to change the address.



Typical GPIB Settings on the **GPIB Parameters** *Pop-Up Menu*

After these parameters are set, the GPIB interface is ready to operate.

For more information, refer to the explanation of the **GPIB Parameters** pop-up menu in the *DSA 601 and DSA 602 User Reference*.

Connecting the DSA to an RS-232-C Device

The RS-232-C interface provides a point-to-point connection between two items of equipment, such as a computer or terminal and the DSA. The remainder of this section tells how to connect and set up the DSA for communication over the RS-232-C interface.

RS-232-C Interface Requirements

The RS-232-C standard defines two types of devices: Data Terminal Equipment (DTE) and Data Communications Equipment (DCE).

The DSA is configured as a DCE device. A 25-pin female D-type-shell RS-232-C connector is located on its rear panel. In industry-standard usage, a 25-pin male D-connector appears on DTE devices, and a 25-pin female D-connector appears on DCE devices. A straight-through male-to-female cable (at least 9-wire) of less than 50 feet is typically used for local DTE-to-DCE connection.

Note, however, that some DTE devices may have female connectors. Also, the RS-232-C ports of many personal computers are configured as DCE devices, with either a 25-pin or a 9-pin connector. Refer to the documentation that accompanying your computer or terminal to determine if it is a DTE or a DCE device.

The following table shows how the pins map when connecting the DSA to another device in any of three common configurations:

- Oscilloscope to a 25-pin DTE device (most terminals)
- Oscilloscope to a 25-pin DCE device (for example, an IBM PC or compatible with a 25-pin COM port)
- Oscilloscope to a 9-pin DCE device (for example, an IBM PC or compatible with a 9-pin COM port).

In most cases, this pin mapping information given will allow you to connect the devices in these configurations.

RS-232-C Pin Mappings

DSA	25-Pin DTE	25-Pin DCE	9-Pin DCE
1	1	1	NC
2	2	3	3
3	3	2	2
4	4	8	7
5	5	20	8
6	6	6	6
7	7	7	5
8	8	4	1
20	20	5	4

For more complicated cases (such as when working with non-standard devices or cables), the pin-out information in the table below should allow you to wire an appropriate connector. The following suggestions may help:

- Pay special attention to the input signal requirements of the external device (many devices require a constant high signal at one or more input pins).
- For DCE-to-DCE connections, do not connect the output line of one DCE to the output line of the other. *Disregarding this restriction may damage one or both devices.*
- Ensure that the signal ground of the DSA is connected to the signal ground of the external device's.
- Ensure that the chassis ground of the DSA is connected to the chassis ground of the external device's.

DSA RS-232-C Pin-out

Pin Number	Function	Mnemonic	Direction †
1	Chassis Ground	-none-	
2	Transmit Data	TxD	Input
3	Received Data	RxD	Output
4	Request to Send	RTS	Input
5	Clear to Send	CTS	Output
6	Data Set Ready	DSR	Output
7	Signal Ground	-none-	
8	Data Carrier Detect	DCD	Output
20	Data Terminal Ready	DTR	Input

† Direction is from the perspective of the controller or terminal.

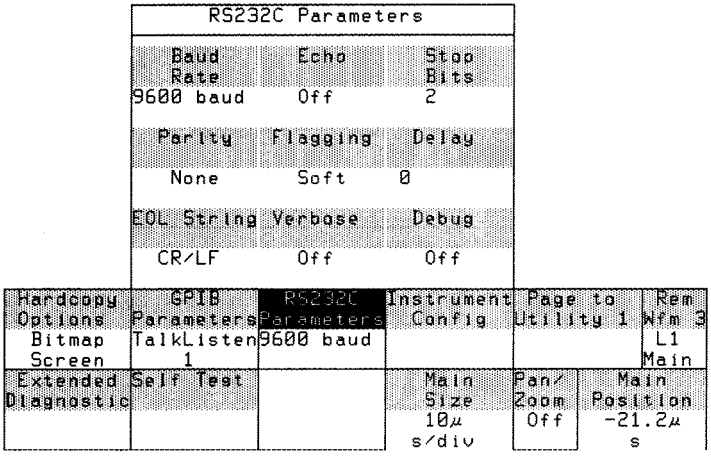
Setting Up RS-232-C Parameters

This subsection lists the parameters of the RS-232-C interface that are implemented in the DSA. These parameters can be set from the front panel (using the Utility major menu and the steps described here), or from within a program (using the RS232 command). After these parameters are set, the RS-232-C interface is ready to operate.

The following steps tell how to set up the RS-232-C parameters at the DSA front panel for remote operation.

☐ Step 1: Press the **UTILITY** major menu button to the right of the display area. The Utility major menu appears in the major menu area toward the bottom of the display.

☐ Step 2: Touch the **Page to Utility 2** selector in the major menu area, then touch the **RS232C** selector. The **RS232C Parameters** pop-up menu is now displayed.



Typical Settings on the **RS-232-C Parameters** *Pop-Up Menu*

☐ Step 3: Repeatedly touch the selector for each parameter except **Baud Rate** and **Delay** until the value you want appears. Touching **Baud Rate** or **Delay** activates the knobs to control these parameters. The RS-232-C selectors are:

The baud rate, stop bits, and parity settings must match those of the controller or terminal, or RS-232-C data communication will be impossible. Also, the controller or terminal's RS-232-C port must be set to use 8-bit characters.

- **Baud Rate** sets the data transmission rate. The selections are 110,150, 300, 600, 1200, 2400, 4800, 9600, or 19200 baud.
- **Stop Bits** sets the number of stop bits sent after each character. The selections are 1, 1.5, or 2 bits.
- **Parity** sets the error check bit for each character. The selections are none, even, or odd. When the parity setting is odd or even, the DSA generates the selected parity on output and checks incoming data against the selected parity on input. When the parity setting is none, no input parity error checking is performed, and no output parity is generated.
- **Echo** allows characters sent to the DSA to be echoed. When echo is turned on, all characters sent to the RS-232-C port are echoed; when echo is turned off, input characters are not echoed.

  Turn echo off when a computer program is transmitting data to the DSA (for example, when a BASIC program on a small computer is being used to control the DSA via the RS-232-C port). The computer program will not expect to see its commands echoed back, and the program will fail. The first command your program sends the DSA should be "ECHO OFF;VERBOSE OFF;INIT".

  Turn echo on when using a CRT or hardcopy terminal, or a computer with a terminal emulation program. Turning echo on in this case allows you to see what you have just typed on your computer or terminal screen.

- **Flagging** sets the method of controlling the flow of data between devices. Flagging is a way for the device receiving data to tell the transmitting device when to stop or resume sending data. The selections are none, hard, or soft. When flagging is set to none, the DSA does not use or recognize any flagging.

  When flagging is set to hard, the DSA uses the DTR (Data Terminal Ready) and CTS (Clear To Send) lines to control data transmission. On output, the DSA transmits data only when the DTR line is asserted. When the DTR line is not asserted, the DSA stops transmitting data. On input, the DSA unasserts the CTS line to stop transmission when its input buffer is three-quarters full, and asserts the CTS line to restart transmission when its input buffer is three-quarters empty.

Soft flagging is usually not used with binary data transfer, since the data may contain XON and XOFF character equivalents. Use hard flagging for binary data transfer.

  When flagging is set to soft, on output the DSA stops transmitting data when it receives an XOFF (DC3) character, and begins transmitting again when it receives an XON (DC1) character. On input, the DSA sends an XOFF character to halt transmission when its input buffer is three-quarters full, and sends an XON character to resume transmission when its input buffer is three-quarters empty.

- **Delay** sets the minimum delay time for the DSA to respond to a query. The delay range is 0 to 60 seconds, in multiples of 20 milliseconds.

- **Verbose** displays status and event messages as commands are executed. When verbose is turned on, each command sent to the DSA returns a response; for example, successfully executed commands return a response of "OK", successfully executed queries return their query data, and events return a response of "EVENT XXX", where XXX is an event code. When verbose is turned off, the controller must query the DSA to receive the message.

Turn verbose off when a computer program is transmitting data to the DSA (for example, when a BASIC program on a small computer is controlling the DSA with the RS-232-C interface). The first command your program sends the DSA should be "ECHO OFF;VERBOSE OFF;INIT".

Turn verbose on when using a CRT or hardcopy terminal, or a computer with a terminal emulation program. Turning verbose on in this case gives you feedback on the execution of commands you have typed.

- **EOL String** sets the end-of-line message terminator for the response to a query. The selections are CR (carriage return), LF (line feed), CR/LF (carriage-return-followed-by-line-feed), or LF/CR (line-feed-followed-by-carriage-return).

When debug is turned on, input/output processing is slowed.

- **Debug** controls whether or not RS-232-C commands appear at the top of the DSA display as they are executed.

Command Structure and Conventions

This section explains the syntax and command processing conventions of the command set. The complete command set, with a summary of the syntax, can be found in the *DSA 601 and DSA 602 Command Reference*.

A large set of commands can be used to control the operations and functions of the DSA from the external interfaces (GPIB and RS-232-C). The same command syntax is used for both interfaces.

Contact your Tektronix field representative for information on the Tek Codes and Formats Standard.

The GPIB and RS-232-C command messages conform to the Tektronix Codes, Formats, Conventions, and Features Standard, or "Tek Codes and Formats" for short. It defines the format of program elements and statements for the command language.

You transmit commands to the DSA using an enhanced American Standard Code for Information Interchange (ASCII) character encoding. The DSA supports both the standard ASCII character set and an additional "escape" character set that includes graphic elements. The character sets are described in the appendices.

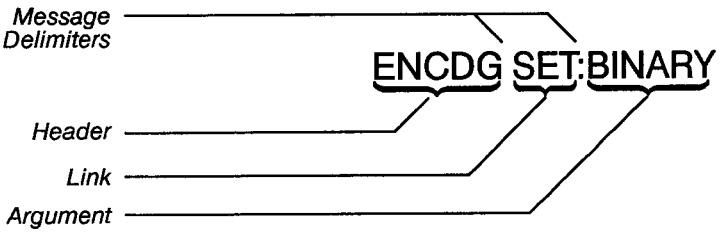
Command Structure

Command language messages are composed of set commands (also referred to as simply *commands*) and query commands (also referred to as *queries*). Set commands tell the DSA to take a specific action. Queries ask the DSA to return information about its state.

Commands are composed of four syntactic elements:

<*header*>	::=	The command name; if it ends with a question mark, the command is a query.
<*delimiter*>	::=	A space, colon (:), comma (,), or semicolon (;) which breaks the message into segments for the DSA to process.
<*link*>	::=	A command sub-function. Not all commands have links.
<*argument*>	::=	A quantity, quality, restriction, or limit associated with the header or link.

The following illustration shows the four syntactic elements:



Example of Syntax Elements

You can use most commands to set or query. However, some commands and/or links can only be used to set, and other commands and/or links can only be used to query. Attempting to make a set-only command or link a query always results in a syntax error.

Set Commands

Set commands cause the DSA to perform a function or change a setting or mode. There are four basic types of set commands:

- Set commands with no arguments
- Set commands with arguments that have no links
- Set commands with link arguments
- Set commands with a mix of link and non-link arguments

Set commands with no arguments—are formed with a header only. Only two commands (INIT and FEOI) have no arguments.

The INIT (initialize) command is an example of a set command that has no arguments. It is also a set-only command.

```
INIT
```

Set commands with arguments that have no links—are formed by specifying a header, following it with a delimiter (space or comma), and then adding its argument.

The AVG command is an example of a set command that has only an argument, in this case, ON or OFF:

```
AVG ON
```

The rules governing quoted strings begin on page 2-12.

Some commands take quoted string arguments (indicated by the notation *<qstring>* in the *Command Reference*). This means that the argument must be enclosed in single or double quotes. The DATE command is an example:

```
DATE '14-FEB-89'
```

Numeric data types are defined in the *Command Reference.*

Other commands take numeric arguments. Numeric arguments are not enclosed in quotes, but simply follow the header and delimiter, which is usually a space. The NAVG command is an example:

```
NAVG 64
```

Set commands with link arguments—are formed by specifying a header, following it with a delimiter, and then adding a link, a colon and an argument. Multiple sets of link arguments can be specified by placing a comma between each link argument. Links can follow a command header in any order.

The ENCDG command is an example of a set command that takes link arguments. ENCDG has two links, WAVFRM and SET:

```
ENCDG WAVFRM:BINARY
```

```
ENCDG SET:ASCII
```

```
ENCDG WAVFRM:BINARY,SET:ASCII
```

Set commands with a mix of link and non-link arguments—are formed by specifying a header, following it with a delimiter, and then adding comma-delimited link arguments and/or non-link arguments (usually in any order).

For example, the following COPY command sets the PRINTER link to TEK4696 and specifies the START argument for the COPY header:

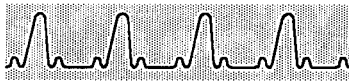
```
COPY PRINTER:TEK4696;START
```

Queries

Queries cause the DSA to return a measurement, waveform (trace) data, or a status condition (e.g., a current setting or mode). The DSA puts the response message in its output buffer.

Query commands have three basic structures:

- Set command headers that are made into queries.



These are formed by placing a question mark (?) after the header and omitting the set command argument.

In query examples, the query command is shown in **bold** type and the DSA response is in regular type.

```
AVG?
AVG ON
```

- Set command headers with links that are made into queries.

  These are formed by placing a question mark (?) after the header, omitting the colon and argument for each link, and separating each link being queried by a comma. Query links may follow a query header in any order.

```
ENCDG? WAVFRM
ENCDG WAVFRM:BINARY
```

  When query links are omitted, all links and their current arguments are returned.

```
ENCDG?
ENCDG SET:ASCII;WAVFRM:ASCII
```

- Queries that have no set-command equivalent (these are called "query-only.")

  An example of a query-only command is NVRAM?, which returns the number of bytes of nonvolatile RAM available for storing front panel settings:

```
NVRAM?
NVRAM 104376
```

Multiple Query Forms

Most set/query commands have only two basic forms of query: specific (<*header*>? <*link*>[{,<*link*>}...]) and general (<*header*>?). However, some commands have additional query forms because of their ability to specify a particular waveform, channel, or color. These commands are:

<*ui*> is an unsigned integer value which in these examples represents a trace (waveform) number (1 to 8) or a channel (1 to 4).

ADJTRACE[<*ui*>]?

CH[<*slot*>[<*ui*>]]?

COLOR<*ui*>?

TRACE[<*ui*>]?

The multiple query forms of these commands return information ranging from most specific to most general.

<*slot*> <*ui*> represents the left, center, or right plug-in compartment and its associated channel number.

For example, the CH[<*slot*>[<*ui*>]] command can produce queries in four basic forms:

- CH<*slot*> <*ui*>? <*link*>[{,<*link*>}...]

  This form returns the specified links and arguments for the specified plug-in unit channel (<*slot*> <*ui*>) only.

- CH<*slot*> <*ui*>?

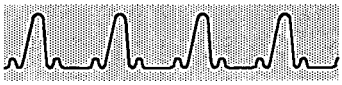
  This form returns all links and arguments for the specified plug-in unit channel.

- CH<*slot*>?

  This form returns all links and arguments for all channels in the specified plug-in unit.

- CH?

  This form returns query responses for all channels installed in the current configuration.



The ADJTRACE<*ui*>, COLOR<*ui*>, and TRACE<*ui*> commands are similar, in that you can specify the waveform number (<*ui*>) or query all waveforms by omitting <*ui*>.

Using Query Responses as Commands

Any response from a query that has a corresponding set command can always be returned to the DSA as a valid command. This makes it easy to save a response from the DSA in a string variable, then send the contents of the string variable back to the DSA as a command when you want to return the DSA to its previous state.

For example, the response from the following query could be used as a set command:

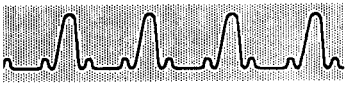
```
ENCDG? WAVFRM,SET
ENCDG WAVFRM:BINARY,SET:ASCII
```

The response from a query that has no corresponding set command (a query-only command) may not be returned to the DSA as a set command string. Any such attempt always results in a syntax error.

For example, if you sent the following query to the DSA and then tried to send the response (NVRAM104376) as a set command, you would get a syntax error.

```
NVRAM?
NVRAM 104376
```

However, a query response that includes a mixture of set, set-only, and query-only links can be returned to the DSA as a set command without generating a syntax error. In such cases, the DSA simply ignores the attempted modification of the query-only link(s).



For example, the response from the CHL1? query (or from any form of the CH command query where you don't specify a link), includes a response from the UNITS query-only link. However, the DSA lets you return the query response from such a command as a set command without causing a syntax error.

Concatenating Commands

Any combination of set and/or query commands may be joined with a semicolon. Thus:

```
RQS ON
```

```
ENCDG?
```

```
UPTIME?
```

may be combined as:

```
RQS ON;ENCDG?;UPTIME?
```

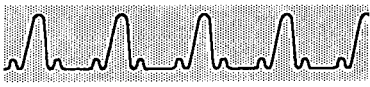
The response to a command message containing more than one query consists of two or more messages, separated by semicolons.

Thus, for the query command:

```
RQS?;EVENT?
```

the query response might be:

```
RQS ON;EVENT 0,"RQS is ON,...SRQ pending"
```



Defining New Command Strings

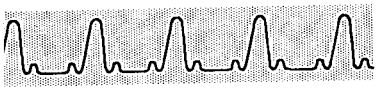
The DSA provides a command (DEF) that enables you to create new command names. That is, you can rename an existing DSA command function, or you can concatenate several existing commands under a single, new command name.

For example, to create a command that gives you the date and time, you could give the following command:

```
DEF "DATIME?","DATE?;TIME?"
```

Also, there is an UNDEF command to remove any or all new commands that you have created. So, to remove the previous example, give this command:

```
UNDEF "DATIME?"
```



Command Processing Conventions

Command processing conventions are rules that specify how the DSA interprets commands you send to it, or how it handles input/output to or from a GPIB or an RS-232-C interface port. Command processing conventions relate to such things as: abbreviating commands being sent to the DSA, getting "long-form" or "short-form" responses from the DSA, using upper or lower case characters in commands, using quoted command strings, and special considerations for each type of interface.

Abbreviating Commands

Each command reserved word (header, link, or argument) that is transmitted to the DSA has an abbreviation. Abbreviations are used in examples in the *Command Reference*; the abbreviated spelling is shown boldface in the header, link, and argument syntax blocks. The complete list of reserved words and their abbreviations is in Appendix B, Reserved Words.

For example, the reserved word:

```
TBMAIN
```

may be abbreviated to:

```
TBM
```

Getting Long-form or Short-form Responses from the DSA

Long form is easier to read; short form is more efficient during data transfers.

The LONGFORM command determines whether the DSA responds to queries in long form or short form. In long form, queries return fully spelled reserved words, and an event query returns both the numeric event code and its associated message string. In short form, queries return abbreviations of reserved words and event queries return only the numeric event code.

The following two examples are with LONGFORM set to ON:

```
ENCDG? WAVFRM
ENCDG WAVFRM:ASCII
```

```
EVENT?
EVENT 250,"NO TRACES DEFINED"
```

Here are the same examples with LONGFORM set to OFF:

Note that the LONGFORM command only affects DSA *responses*; set commands and queries can always be sent to the DSA in either long or short form.

```
ENCDG? WAVFRM
ENC WAV:ASC
```

```
EVENT?
EVENT 250
```

Using Upper and Lower Case

The DSA accepts both upper and lower case alphabetic input data. Thus, the following two commands are recognized as identical:

```
RQS ON
```

```
rqs on
```

The DSA returns the same case of alphabetic data to the GPIB or RS-232-C ports as you enter from within a quoted string.

Using Quoted Strings

Some commands accept or return quoted string (<*qstring*>) data; for example, the <*qstring*> argument to the DATE command contains the day, month, and year. The rules for quoted string usage are described below.

- The maximum length of any quoted string is 127 characters, excluding delimiters (unless noted otherwise in the *Command Reference*).
- An apostrophe (') or a quotation mark (") is a valid string delimiter. However, quoted string data returned to the controller (query data) is delimited solely with quotation marks.
- The same type of delimiter that opens a quoted string must close that string. Examples:

  `"this is a quoted string"` and `'so is this'`

  `'But this is not a quoted string"`

- You can mix quotation marks and apostrophes within a string if the previous rule is followed. For example:

  `"this is an 'acceptable' string"` and `'so "is" this'`

- A delimiter may be included within a string by simply repeating the delimiter. Examples:

  `"double "" quote"` and `'single '' quote'`

- For GPIB transfers, a quoted string may not be terminated with an EOI interface signal prior to the closing delimiter.

  For example, a missing end-quote after the command TEST in the command below produces an invalid string.

  `"test`<*EOI*>

- A carriage return or line feed embedded within a quoted string does not terminate that string but is considered to be just another character in the string.

- A quoted string may not include an embedded ASCII NULL character (0).

Terminating Messages

Message terminators are transmitted by a sending device to let receiving devices know that message transmission is complete. The DSA allows you to select a message terminator that is compatible with the controller or terminal you are using.

Terminators for the RS-232-C Interface – are selected through the front panel using the **RS232C Parameters** pop-up menu from the Utility 2 major menu or through the interface using the RS232 command. RS-232-C terminators are:

- CR (carriage return)

- LF (line feed)

- CR/LF (carriage return followed by line feed)

- LF/CR (line feed followed by carriage return)

Line feeds and carriage returns embedded within binary block (<*bblock*>) data are treated as data bytes, not as message terminators. Once the DSA begins reading a binary block, line feeds and carriage returns are not processed as terminators until the byte count of the block is satisfied.

Terminators for the GPIB interface – are selected through the front panel only, using the **GPIB Parameters** pop-up menu from the Utility 2 major menu. GPIB terminators are:

- EOI (assert EOI management line with the last byte of transmission)
- EOI/LF (transmit line feed as last character *and* assert EOI line)

Using White Space

All command elements (headers, links, arguments, or punctuation) may be preceded or followed by white space,(blank characters).

Thus, the following example:

```
ENCDG  WAVFRM  :  ASCII  ,  SET  :  BINARY
```

is equivalent to:

```
ENCDG WAVFRM:ASCII,SET:BINARY
```

Null Commands

Commands consisting solely of any combination of blank characters, carriage returns, and line feeds are called null commands. Null commands are ignored by the DSA and do not produce an error.

I/O Buffers

The following information pertains to both GPIB and RS-232-C input/output buffers, except as noted otherwise.

I/O buffer sizes—are 256 bytes for the GPIB Input buffer, 1024 bytes for the RS-232-C Input buffer, and 1024 bytes for the GPIB and RS-232-C Output buffer.

Data that exceeds the sizes of the GPIB and RS-232-C input/output buffers (256/1024 bytes) can be accepted. The DSA parses input data as soon as it is received at either port, thereby continuously emptying the input buffers while processing commands.

If an external controller fills an input buffer before the DSA has an opportunity to process the contents, the DSA holds off the external controller (with GPIB interface signals or RS-232-C flagging) until the buffer has been processed, leaving room for more input data.

Likewise, if a query response fills an output buffer, the DSA stops sending data to the buffer until some of the data are read by the external controller or terminal.

When a new message is received at the GPIB port—the DSA unconditionally clears its GPIB output buffer (no error is reported). This means that the GPIB output buffer of the DSA must be read by the controller after each message containing a query, or the response will be lost (overwritten).

When GPIB input and output message buffers are full—the DSA unconditionally clears the GPIB output buffer. An execution error is also reported (event code 203, "I/O buffers full").

If GPIB buffers are empty—and the DSA address talk addressed and not currently processing a GPIB command, it returns a Talked-With-Nothing-To-Say (TWNTS) message to the controller. This message is one byte with all eight bits set, ended by a message terminator (FF<*EOI*>). It is then up to the controller program to take appropriate action.

If a "hang" condition occurs, consult your controller or terminal operator manual for restart instructions.

If the RS-232-C output buffer is empty—and an external device attempts to read data from the RS-232-C port, the external device will "hang" the interface (no further input/output operations will be possible). This condition cannot occur when using a computer or terminal to send commands interactively to the DSA over the RS-232-C interface.

This condition may occur when executing a program that expects input from the DSA's RS-232-C port. In such cases, it is up to the program to recognize a "timeout" condition for expected input and take appropriate action.

GPIB Specific Conventions

When the DSA receives a Device Clear (DCL) or Selected Device Clear (SDC) interface message from the GPIB, it does the following:

1. Clears any service requests and all pending events except power on.

2. Clears the GPIB input and output buffers.

3. Restarts GPIB message processing in the DSA.

DCL and SDC interface messages do not change DSA settings or stored data, and do not interrupt front panel control or non-programmable functions.

RS-232-C Specific Conventions

You should be aware of the processing conventions that are specific to the RS-232-C interface. These conventions pertain to:

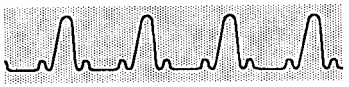
- Transferring binary block data
- Echoing character input
- Using Verbose mode
- Processing "break" characters
- RS-232-C I/O errors

When transferring binary block data – to the DSA via the RS-232-C port, note the following points:

- Do not transmit binary block data to the DSA when ECHO is set to ON. Attempting to do so causes the input block to be discarded and generates event code 164.
- Do not use binary data transfers with soft flagging unless you can ensure that the data does not contain XON or XOFF characters. Using DTR/CTS (hard) flagging guarantees correct data transfer.
- All eight bits of a binary block data byte contain meaningful information. To ensure that all eight bits are received or transmitted, an RS-232-C device must be configured to receive and transmit eight-bit characters (set the RS-232-C word length to eight bits).

  For example, a Tektronix PEP Controller (or PC compatible) uses this MS-DOS command to set up its RS-232-C interface:

```
MODE COM1:9600,n,8,1
```



Echoing character input—means that all characters received at the RS-232-C port are echoed back to the command source when ECHO is set to ON.

You can turn echo on or off from the front panel by selecting **RS232C** from the Utility 2 major menu and touching the **Echo** selector, or you can send the commands: RS232 ECHO:ON or RS232 ECHO:OFF.

When you are using a computer program to transmit commands to the DSA (for example, when a BASIC program is being used to control the DSA via the RS-232-C port), ECHO should be set to OFF.

When you are using a CRT or hardcopy terminal, or a computer running terminal emulation software to send commands interactively to the DSA via the RS-232-C port, ECHO should be set to ON.

When ECHO is set to ON, it has the following effects on command input:

- The DSA solicits command input with a ">" prompt. When this prompt appears on an RS-232-C device, enter a valid command and terminator.

- All command input is buffered. Therefore, commands will not be analyzed or executed until a terminator is received at the RS-232-C port. (As you may recall, RS-232-C I/O is normally not buffered, which means that each input character is processed as soon as it is received at the RS-232-C port.)

- Until the command is terminated, it may be edited with any of the following special characters:

  CONTROL-R retypes the current input command and places the cursor to the right of the last character of the command.

  CONTROL-U deletes the current command and returns the cursor to the start of the line.

  BACKSPACE erases the character to the left of the cursor (the effect of the backspace character is compatible with CRT terminals, but not with hardcopy terminals). If a character has been erased with the backspace key, the newly edited command can be seen by using the CONTROL-R character (applies to both CRT and hardcopy terminals).

  DEL or RUBOUT same function as BACKSPACE.

  BACKSLASH (\) use the backslash to place special editing characters (CR, LF, BACKSPACE, DEL, CONTROL-R, or CONTROL-U) in a quoted string. To place a backslash character in a quoted string, enter two consecutive backslashes ("\\" is interpreted as "\").

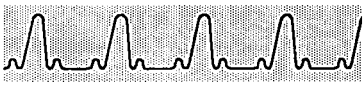
- Command input is discarded if it exceeds 256 bytes (the input buffer size) before a terminator is entered. If this happens, a command error (event code 163) is posted to the RS-232-C port and the input buffer is emptied.

- Non-printable ASCII characters are echoed with the visual representations shown in the following table:

Non-printable ASCII Character Representations

ASCII Character	Echoed Character	ASCII Character	Echoed Character
NUL (0)	^@	DC2 (18)	^R †
SOH (1)	^A	DC3 (19)	^S ††
STX (2)	^B	. .	.
. .	.	. .	.
. .	.	NAK (21)	^U †
BS (8)	^H †	. .	.
HT (9)	^I	. .	.
LF (10)	^J †	SUB (26)	^Z
. .	.	ESC (27)	^[
. .	.	FS (28)	^\
CR (13)	^M †	GS (29)	^]
. .	.	RS (30)	^^
. .	.	US (31)	^_
DC1 (17)	^Q ††	DEL (127)	^? †

† *Only echoed when preceded with a backslash.*
†† *Only echoed when soft flagging is disabled.*



Using verbose mode—causes the DSA to return a response for each command sent. When VERBOSE is set to OFF, only valid queries return a response from the DSA.

You can turn verbose on or off from the front panel by selecting **RS232C** from the Utility 2 major menu and touching the **Verbose** selector, or you can send the commands RS232 VERBOSE:ON or RS232 VERBOSE:OFF.

When RS-232-C VERBOSE is set to ON, each semicolon or terminated input command causes the DSA to return one of these responses:

OK	Returned for a successfully executed set command
<query response>	Returned for a successfully executed query
EVENT *<NR1>*[,*<qstring>*]	Returned when the DSA detects an error while parsing or executing a query/set command, where the *<NR1>* value represents an event code, and the optional *<qstring>* is an event code description string that describes the numerical event code. The event code description string *<qstring>* is only returned when LONGFORM is set to ON. If more than one error is detected while parsing a query or set command, only one EVENT response is returned to the RS-232-C port. All other errors are stacked and may be polled with the STBYTE? or EVENT? commands.

The following table demonstrates typical DSA response behavior with VERBOSE mode set to ON.

Examples of Responses with VERBOSE ON

Input Command(s)	DSA Response
LONGFORM OFF	OK
INPUT STO1;RS232? BAUD	OK;RS232 BAUD:9600
JUNK;INIT;INPUT?	EVENT 156;OK;INPUT STO1
JUNK;INIT	EVENT 156;OK

When RS-232-C VERBOSE is set to OFF, only valid queries cause the DSA to return responses to the RS-232-C. Valid set commands, invalid set commands, and invalid queries elicit no response event messages from the DSA. Errors associated with invalid commands are not discarded; they are stacked and may be polled at any time by using the STBYTE? or EVENT? commands.

The following table demonstrates typical DSA response behavior with VERBOSE set to OFF.

Examples of Responses with VERBOSE OFF

Input Command(s)	DSA Response
INPUT STO1;RS232? BAUD	RS232 BAUD:9600
JUNK;INIT;INPUT?	INPUT STO1
JUNK;INIT	*(none)*

The factory default state for verbose mode is off.

Verbose mode affects event communication at power-on. When the DSA is turned on and completes its power-on cycle, the DSA communicates events differently depending on the state of the verbose function.

- When VERBOSE is set to ON at power-on, an asynchronous message is written to the RS-232-C port. This message reports either that the instrument is operating satisfactorily (Event 401, "Power on"), or that diagnostics have discovered a fault (Event 394, "Test completed and failed").

- When VERBOSE is set to OFF at power-on, no asynchronous messages are written to the RS-232-C port. Instead, power-on events are stacked in the usual manner.

When the DSA senses a BREAK signal – at the RS-232-C port, it returns a special message that acknowledges this transmission. The form of the acknowledgement message depends on whether ECHO is set to ON or OFF.

- When ECHO is set to ON, the DSA signals that it has processed the BREAK signal by echoing a new prompt symbol (for example, >) for command input.

- When ECHO is set to OFF, the DSA signals that it has processed the BREAK signal by sending the following character string to the RS-232-C device:

  DCL<*terminator*>

Reception of the BREAK signal clears the RS-232-C input and output buffers and restarts the DSA's RS-232-C message processing. BREAK signals do not change DSA settings or stored data, and do not interrupt front panel operation or non- programmable functions.

RS-232-C I/O errors – are reported when there is a problem with parity, framing, or input buffer overruns.

To report RS-232-C errors, the DSA prints an error message on the display and posts an event code to both the GPIB and the RS-232-C ports:

RS-232-C I/O Errors

I/O Error	Event Code	Information
Parity	653	Check to identify transmission errors (PARITY ON)
Framing	654	A stop bit was not detected when data was received at RS-232-C port (indicates baud-rate mismatch)
Input Buffer Overrun	655	Software or hardware input buffer overflowed with data (caused by improper or nonexistent flagging)

To recover from I/O errors, the DSA RS-232-C interface takes the following actions:

During these I/O error recovery steps (when ECHO is set to OFF), the DSA may process incomplete commands, causing spurious syntax or semantic errors to be reported.

- When ECHO is set to OFF, all unparsed input buffer data are discarded until a semicolon or <*terminator*> character is encountered. Command processing resumes or resynchronizes from the point at which the semicolon or <*terminator*> is found.
- When ECHO is set to ON, all buffered but unparsed input data are discarded and you are prompted again for input.

Command Groups

This section presents the command set arranged by function. The *DSA 601 and DSA 602 Command Reference* presents complete command information arranged alphabetically.

Functional Groups

The table below lists the groups and their function(s). The following two pages show all the commands grouped by function.

Functional Groups in the Command Set

Group	Functions Controlled
Acquisition	Acquisition (digitizing) of waveforms
Calibration/Enhanced Accuracy	Enhanced accuracy functions
Channel/Vertical	Plug-in amplifier vertical parameters
Cursor	Waveform cursor selection and positioning
Data Transfer	Transfer of waveform data and front panel settings to and from the DSA
Diagnostics	Self-tests diagnostics and extended diagnostics
Display and Color	Front panel display parameters and colors
External I/O	Printer parameters, debug functions, and RS-232-C parameters
Label and Text	Placement of user-defined labels and text
Measurement	Oscilloscope measurement functions
Miscellaneous/System	System and front panel functions
Status and Event	Instrument event reporting, hardware identification, and configuration information
Time Base/ Horizontal	Main and window record length and position
Triggering	Triggering parameters
Waveform and Settings	Waveform creation and modification, and front panel settings commands

Acquisition

AUTOSET
AVG
CONDACQ
DELTA
DIGITIZER
ENV
FILTER
FFT
INCACQ
INTERLEAVE
NAVG
NENV
NREPTRIG

Calibration / Enhanced Accuracy

CALPROBE
CALSTATUS?
CALTEMPDELTA?
CCALCONSTANTS
CHSKEW?
LCALCONSTANTS
MCALCONSTANTS
RCALCONSTANTS
SELFCAL

Channel / Vertical

CH<*slot*> <*ui*>
CH<*slot*>?
CH?

Label and Text

LABABS
LABEL
LABREL
TEXT

Measurement

BASELINE
COMPARE
DAINT
DISTAL
DLYTRACE
LMZONE
MEAS?
<*meas*>?
MESIAL
MSLIST
MSLOPE
MSNUM?
MSYS
MTRACK
PROXIMAL
REFLEVEL
REFSET
RMZONE
SNRATIO
TOPLINE
TTAVERAGE
<*meas*> ::=
CROSS
DELAY
FALLTIME
FREQ
GAIN
MAX
MEAN
MID
MIN
PDELAY
PERIOD
PHASE
PP
RISETIME
RMS
TTRIG
WIDTH
YTENERGY
YTMNS_AREA
YTPLS_AREA

Miscellaneous / System

ABSTOUCH
CALIBRATOR
DATE
DEF
DSYMENU?
DSYSTOTD
FEOI
FPANEL
FPUPDATE
HSBATT?
INIT
LONGFORM
OPTIONS?
PATH
POWERON?
PROBE
SCLOCKD
SPEAKER
TIME
UNDEF
UPTIME?
USERID

DSA 601 and DSA 602 Command Groups

Cursor

CURSOR
DOT1ABS
DOT2ABS
DOT1REL
DOT2REL
H1BAR
H2BAR
V1BAR
V2BAR

Data Transfer

ABBWFMPRE
BYT.OR
CURVE
ENCDG
INPUT
OUTPUT
SET?
WAVFRM?
WFMPRE

Diagnostics

DIAG?
TEST

Display and Color

COLOR
DISPLAY

External I/O

ALTINK
BITMAP
COPY
DEBUG
HPGL
PIN8
PIN24
RS232
TEK4692
TEK4696

Status and Event

CONFIG?
EVENT?
ID?
IDPROBE?
PIVERSION?
RQS
SRQMASK
STBYTE?
UID?

Time Base / Horizontal

ADJTRACE<*ui*>
MAINPOS
TBMAIN
TBWIN
WIN1POS
WIN2POS

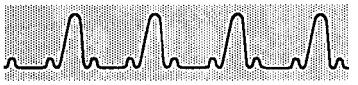
Triggering

TR?
TRMAIN
TRWIN
TSMAIN?
WTMODE

Waveform and Settings

ADJTRACE<*ui*>
CLEAR
DELETE
FPSLIST?
FPSNUM?
NVRAM?
PZMODE
RECALL
REMOVE
SCANSTOWFM
SELECT
SETSEQ
STOLIST?
STONUM?
STORE
TRACE<*ui*>
TRACE[<*ui*>]?
TRANUM?
WFMSCALING

DSA 601 and DSA 602 Command Groups (cont)



Acquisition Commands

The Acquisition commands control waveform digitizing. The commands are presented in three groups: commands that control signal acquisition, such as DIGITIZER, commands that control waveform functions, such as AVG, and commands that affect acquisition parameters, such as NAVG.

Signal Acquisition Commands

Command	Meaning
AUTOSET	Adjusts the waveform signal for optimal display.
CONDACQ	Controls the condition(s) on which the acquisition of waveforms stops.
DIGITIZER	Starts and stops waveform acquisition.

Waveform Function Commands

Command	Meaning
AVG	Turns waveform averaging on or off. (Averaging can also be defined in the waveform description; use the TRACE DESCRIPTION command in the Waveform and Settings group.)
DELTA	Compares an acquired waveform with an enveloped reference waveform and performs various actions under specified conditions.
ENV	Turns waveform enveloping on or off. (Enveloping can also be defined in the waveform description using the TRACE DESCRIPTION command.)

Acquisition Parameter Commands

Command	Meaning
FILTER	Limits the digitizer bandwidth for antialias filtering.
FFT	Controls FFT (Fast Fourier Transform) parameters. Use the TRACE <*ui*> DESCRIPTION command to enable the FFT function.
INCACQ	Controls digitizer incremental acquire mode.
INTERLEAVE	Controls digitizer interleave mode, which must be enabled to attain the maximum DSA sample rate.
NAVG	Sets the number of acquisitions to be used in waveform averaging.
NENV	Sets the number of acquisitions to be used in waveform enveloping.
NREPTRIG	Sets the number of waveforms to be acquired when conditional acquisition is set to repetitive trigger.

Calibration/ Enhanced Accuracy Commands

The Calibration/Enhanced Accuracy commands initiate DSA self-calibration features and report on their condition. Calibration refers to the state of accuracy of the instrument. There are three states of calibration:

- Normal (or non-enhanced accuracy), when the DSA is warming up and has not yet entered enhanced accuracy.
- Enhanced accuracy, when the DSA has warmed up and entered enhanced accuracy.
- New configuration, when a new plug-in unit has been installed and the DSA has not yet warmed up and entered enhanced accuracy.

The Calibration/Enhanced Accuracy commands are presented in two groups: commands that execute calibration routines or return calibration status, and commands that control calibration constants.

System Calibration and Status Commands

Command	Meaning
CALPROBE	Initiates the probe calibration routine.
CALSTATUS?	Returns the state of DSA enhanced accuracy.
CALTEMPDELTA?	Returns the change in degrees Celsius since the last calibration.
CHSKEW?	Returns the skew (time delay) measured by the probe calibration routine for each active channel.
SELFCAL	Determines whether enhanced accuracy calibration is performed manually or automatically, and initiates manual self-calibration.

Calibration Constants Commands

Command	Meaning
CCALCONSTANTS	Controls the calibration constants for the center plug-in unit.
LCALCONSTANTS	Controls the calibration constants for the left plug-in unit.
MCALCONSTANTS	Controls the calibration constants for the DSA
RCALCONSTANTS	Controls the calibration constants for the right plug-in unit.

Channel/ Vertical Commands

The Channel/Vertical commands set and query the vertical parameters of an input channel. The CH<*slot*> <*ui*> command has a large number of links, some of which are specific to an amplifier type. The Channel/Vertical commands are shown in the following table:

Channel/Vertical Commands

Command	Meaning
CH<*slot*> <*ui*>	Sets parameters for the specified plug-in unit channel.
CH<*slot*>?	Returns parameter information for all channels in a plug-in compartment.
CH?	Returns parameter information for all installed channels.

Cursor Commands

Cursor commands control the creation and placement of cursors on waveforms. Cursors are positioned with the dot or bar cursor commands. Use the CURSOR command to display the cursors and control the readout in the Cursor menu. (The readouts show the difference (Δ) between the cursors.) The four cursor types are split dot, paired dot, horizontal bar, and vertical bar cursors. The Cursor commands are shown in the following table:

Cursor Commands

Command	Meaning
CURSOR	Selects cursor operating characteristics.
DOT1ABS DOT2ABS	Positions the first and second split or paired cursors to specified absolute locations.
DOT1REL DOT2REL	Positions the first and second split or paired cursors relative to the DOT1ABS and DOT2ABS locations, respectively.
H1BAR H2BAR	Positions the first and second horizontal bar cursors to specified absolute locations.
V1BAR V2BAR	Positions the first and second horizontal bar cursors to specified absolute locations.

Cursors and the Selected Waveform

All cursor commands and queries apply to the selected waveform. When no waveforms are defined, there is no selected waveform. Cursor parameters cannot be set or queried without a selected waveform.

By default, the most recently defined waveform is the selected waveform. Use the SELECT command to select a different waveform when more than one waveform appears on the display.

Each displayed waveform has a unique set of cursor parameters. Therefore, cursor queries and set commands access cursor data that apply only to the selected waveform. Altering the cursor parameters of the selected waveform has no effect on the cursor data of any other waveform.

Cursor Positioning Methods

Dot cursors may be horizontally positioned by graticule divisions, percentage of the waveform record length, or horizontal units of the selected waveform. Dot cursors can also be positioned relative to their current position.

Bar cursors may be horizontally or vertically positioned by graticule divisions or the units of the selected waveform.

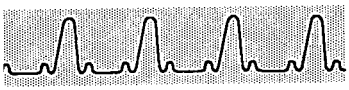
When cursors are positioned or queried by graticule divisions, use the illustration shown in the DOT1ABS command entry in the *DSA 601 and DSA 602 Command Reference* to interpret the dimensions of single and dual graticules.

Data Transfer Commands

Data transfer commands transfer waveform information and front panel settings to and from the DSA through the external interfaces. The Data Transfer commands are presented in two groups: data transfer execution commands and data transfer parameter commands.

Data Transfer Execution Commands

Command	Meaning
CURVE	Transfers unscaled waveform data. Scaling information is included in the waveform preamble.
SET?	Returns the front panel settings to the controller.
WAVFRM?	Returns the waveform preamble and waveform data points for the waveform specified by OUTPUT. WAVFRM? is equivalent to entering WFMPRE?;CURVE?
WFMPRE	Contains the links of the waveform preamble.



Data Transfer Parameter Commands

Command	Meaning
ABBWFMPRE	Controls whether a WFMPRE? query returns all links or an abbreviated set of links.
BYT.OR	Sets the byte order for binary data transfer.
ENCDG	Selects ASCII or binary format for data transfers.
INPUT	Selects the memory location in which to store a waveform transferred to the DSA.
OUTPUT	Selects the stored or displayed waveform to be transferred from the DSA.

Retrieving Waveform Data From the Oscilloscope

In general, the controller retrieves waveform data from the DSA in the following sequence:

1. Use the ENCDG command to select waveform encoding (ASCII or binary).

2. If binary encoding is in effect, use the BYT.OR command to select the binary data transmission order (LSB or MSB) appropriate for the target controller.

3. Use the OUTPUT command to select the stored or displayed waveforms that are to be retrieved.

4. Finally, use WFMPRE? and CURVE? (or WAVFRM?) to query the DSA for the preamble and unscaled waveform data of the waveform selected by OUTPUT.

Notice that once you establish waveform encoding and the binary data transmission order, steps 1 and 2 can be omitted for all subsequent retrievals.

Sending Waveform Data to the Oscilloscope

The controller normally sends waveform data that came from a previous query.

In general, waveform data are sent back to the DSA in the following sequence:

1. If the waveform data are binary encoded, use BYT.OR to specify the transmission order of the returned data.
2. Use the INPUT command to select the stored waveform location where the data will be written.
3. Use the WFMPRE command to return the waveform preamble to the DSA.
4. Finally, use the CURVE command to return unscaled waveform data to the DSA.

Diagnostics Commands

The Diagnostics commands invoke self-tests diagnostics or extended diagnostics. The Diagnostics commands are shown in the following table:

Diagnostics Commands

Command	Meaning
DIAG?	Returns the result of the diagnostic tests.
TEST	Performs self-test or extended-test diagnostics.

Display and Color Commands

Display commands control how waveforms appear on the display. Color commands determine the colors used on the display. The Display and Color commands are shown in the following table:

Display and Color Commands

Command	Meaning
COLOR	Determines the colors used in the display.
DISPLAY	Controls display intensity, number of graticules, and whether waveforms are displayed as dots or vectors.

External I/O Commands

The External I/O commands produce hardcopy printout of the display, control the debug feature for both interfaces, and set the parameters of the RS-232-C interface. The External I/O commands are presented in two groups: interface commands and printer commands.

Interface Commands

Command	Meaning
DEBUG	Displays the ASCII commands on the front panel as they are executed via the specified interface.
RS232	Sets the parameters of the RS-232-C interface.

Printer Commands

Command	Meaning
ALTINKJET	Controls HP Thinkjet and HP LaserJet printers.
BITMAP	Controls screen capture by an external computer.
COPY	Produces a printout of the display.
HPGL	Controls Tektronix HC100 plotters and other devices conforming to the HP-GL format.

Printer Commands (Cont.)

Command	Meaning
PIN8	Controls standard Epson 8-pin bit image graphics printers, such as the Tektronix 4644.
PIN24	Controls extended Epson 24-pin dot graphics printers.
TEK4692	Controls Tektronix 4692 Color Graphics Copiers and Tektronix 4693D Color Image Printers in 4692 emulation mode.
TEK4696	Controls Tektronix 4696 and Tektronix 4695 Color Ink-Jet Printers.

Label and Text Commands

Label and Text commands control how user text is placed on the display. Labels are text you create to identify waveforms and front panel settings. The Label and Text commands are shown in the following table:

Label and Text Commands

Command	Meaning
LABABS	Positions the label associated with the selected waveform to an absolute location.
LABEL	Defines and deletes labels, and controls whether they are displayed.
LABREL	Positions the label associated with the selected waveform to a location relative to its former absolute location.
TEXT	Defines and positions a text string on the display.

Measurement Commands

The Measurement commands take waveform measurements and control the parameters with which these measurements are taken. In this discussion, measurement commands are defined first, followed by measurement parameter commands, and finally measurement execution commands.

Measurements and the Selected Waveform

All measurement commands and queries apply to the selected waveform. When no waveforms are defined, there is no selected waveform. Measurement parameters cannot be set or queried without a selected waveform.

By default, the most recently defined waveform is the selected waveform. Use the SELECT command to select a different waveform when more than one waveform appears on the display.

Each displayed waveform has a unique set of measurement parameters. Therefore, measurement queries and set commands access measurement data that apply only to the selected waveform. Altering the measurement parameters of the selected waveform has no effect on the measurement data of any other waveform.

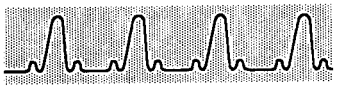
Measurement (<*meas*>) Commands

No measurements can be taken on XY waveforms, on FFT waveforms, or on waveforms displayed in point accumulate mode.

Three types of measurements can be taken: timing, amplitude, and area/energy. The measurements are defined in the following tables. Measurements are taken on the selected waveform.

The symbol <*meas*> represents one or more measurements. A <*meas*> measurement can be individually queried or <*meas*> can represent one or more measurements in a measurement execution command.

<meas>	Meaning
CROSS	The time from the trigger point to a specified reference level crossing.
DELAY	The time between the first and last mesial crossing of a waveform within the measurement zone.
FALLTIME	The transition time of a falling pulse edge, from the distal to proximal levels.
FREQ	Frequency (reciprocal of the period measurement).
PDELAY	Propagation delay between mesial crossings on two waveforms (used with the DLYTRACE command).
PERIOD	The time between the first and next mesial crossing of the same slope.
PHASE	The phase relationship of the reference waveform to the selected waveform, expressed as a value from -360 to +360 degrees.
RISETIME	The transition time of a rising pulse edge, from the proximal to distal levels.
TTRIG	The time between the main trigger point and the window trigger point.
WIDTH	The time between the first and next mesial crossing of the opposite slope.



Amplitude Measurements

<meas>	**Meaning**
GAIN	Ratio of the peak-to-peak amplitude of the reference waveform to the peak-to-peak amplitude of the selected waveform.
MAX	Maximum amplitude (most positive peak voltage).
MEAN	Average amplitude (arithmetic mean voltage).
MID	Amplitude midpoint, halfway between the maximum amplitude and the minimum amplitude.
MIN	Minimum amplitude (most negative peak voltage).
PP	Peak-to-peak value; the voltage difference between the maximum amplitude and minimum amplitude.
RMS	True root-mean-square voltage.

Area/Energy Measurements

<meas>	**Meaning**
YTENERGY	The energy represented under the curve of a Yt waveform (this integral of the squared voltages can be divided by the resistance of the circuit to yield power measurements).
YTMNS_AREA	The difference between the area under a Yt curve above a specified reference level, and the area under the curve below that level.
YTPLS_AREA	The total, absolute value of all areas between a Yt waveform and a user-specified reference level.

Measurement Parameter Commands

Measurement parameters apply to the selected waveform. The following commands set the measurement parameters.

Measurement Parameter Commands

Command	Meaning
BASELINE	Sets the absolute value of the baseline when measurement tracking (MTRACK) is turned off.
DAINT	Sets the data interval (one waveform period or the entire measurement zone) for taking measurements.
DISTAL	Sets the distal (farthest from origin) reference level, typically 90% of the baseline-to-topline value.
DLYTRACE	Sets the reference (delayed) waveform used with the PDELAY, GAIN, and PHASE measurement.
LMZONE	Sets the left limit of the measurement zone.
MESIAL	Sets the mesial (middle) reference level (the end point of the waveform period), typically 50% of the baseline-to-topline value.
MSLOPE	Sets the crossing slope for measurements.
MTRACK	Controls measurement tracking (whether you or the DSA set the baseline and topline values).
PROXIMAL	Sets the proximal (nearest to origin) reference level, typically 10% of the baseline-to-topline value.
REFLEVEL	Sets a user-defined signal reference level.
REFSET	Sets the reference value(s) used in comparison mode (COMPARE is set to ON).
RMZONE	Sets the right limit of the measurement zone.

Measurement Parameter Commands

Command	Meaning
SNRATIO	Sets the amplitude of a noise rejection band centered on the MESIAL level.
TOPLINE	Sets the absolute value of the topline when measurement tracking (MTRACK) is turned off.
TTAVERAGE	Sets the number of samples used by the TTRIG measurement.

Measurement parameter interactions—in the following table show how various parameters affect the measurement system.

Measurement Parameter Interactions

Parameter(s)	Effects
MTRACK	Determines whether BASELINE and TOPLINE are set dynamically by the DSA using measurement tracking (histograms) or are set by you.
BASELINE TOPLINE	Used to calculate range values for PROXIMAL, MESIAL, and DISTAL, when MTRACK is set to OFF.
DISTAL PROXIMAL	Sets the measurement end points (voltage levels) for RISETIME and FALLTIME <*meas*>.
MESIAL	Sets measurement endpoints for DELAY, FREQ, PDELAY, PERIOD, PHASE, and WIDTH <*meas*>; and when DAINT is set to SINGLE, for MEAN, RMS, YTENERGY, YTMNS_AREA and YTPLS_AREA <*meas*>.
LMZONE RMZONE	Establishes a measurement zone to isolate part of a waveform.
DAINT	Limits the measurement zone for MEAN, RMS, YTENERGY, YTMNS_AREA, and YTPLS_AREA.
REFLEVEL	Sets the crossing level for CROSS, and provides a comparison level for YTMNS_AREA <*meas*>.

Measurement Parameter Interactions

Parameter(s)	Effects
MSLOPE	Sets the crossing slope for CROSS <*meas*>.
SNRATIO	Noise filter for DELAY, FREQ, PDELAY, PERIOD, PHASE, and WIDTH <*meas*>; and when DAINT is set to SINGLE, for MEAN, RMS, YTENERGY, YTMNS_AREA and YTPLS_AREA <*meas*>.
DLYTRACE	Selects the delayed waveform used with the PDELAY, GAIN, and PHASE <*meas*>.

Measurement Execution Commands

The following commands control the taking of measurements and the measurement execution modes.

Measurement Execution Commands

Command	Meaning
COMPARE	Controls comparison mode. When comparison mode is on, reference values set with the REFSET parameter are subtracted from measurement values; the difference is returned.
MEAS?	Executes and returns the values of the measurements in the measurement list (MSLIST).
<*meas*>?	Executes and returns the value of the specified measurement (<*meas*>).
MSLIST	Selects the measurements (<*meas*>) which are executed continuously in the Measure major menu, and are executed once at a MEAS? query.
MSNUM?	Returns the number of measurements in the measurement list (MSLIST).
MSYS	Controls display of the Measure major menu.

Miscellaneous/ System Commands

The Miscellaneous/System commands include front panel commands, system status queries, and other useful functions. The Miscellaneous/System commands are presented in two groups: front panel commands and system commands.

Front Panel Commands

Command	Meaning
ABSTOUCH	Mimics a touch to the front panel display area or the major menu buttons.
CALIBRATOR	Controls the front panel calibrator output.
DSYMENU?	Returns which major menu is currently displayed.
DSYSTOTD	Controls the display of stored waveform time and date strings.
FPANEL	Controls front panel lockout.
FPUPDATE	Controls when front panel readouts are updated.
PROBE	Determines the result of a probe button press.
SPEAKER	Controls whether the DSA beeps when the display is touched.

System Commands

Command	Meaning
DATE	Sets the date on the system calendar.
DEF	Defines logical names for command strings.
FEOI	Forces a message terminator in a command string.
HSBATT?	Returns the status of Option 3C, Acquisition Memory External Power Input.
INIT	Initializes the system.
LONGFORM	Controls whether the DSA returns full or abbreviated query responses and event information.
OPTIONS?	Returns a list of installed options.

System Commands (Cont.)

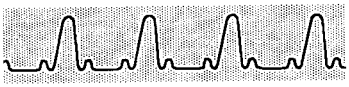
Command	Meaning
PATH	Determines whether queries return link-argument information or only the arguments.
POWERON?	Returns the number of times the DSA has been powered on.
SCLOCKD	Controls whether the sampling clock is dithered.
TIME	Sets the time on the system clock.
UNDEF	Deletes logical names previously defined with DEF.
UPTIME?	Returns the number of hours the DSA has been powered on.
USERID	Saves a quoted string in nonvolatile RAM.

Status Queries and Event Commands

Status Queries and Event commands report identifying information and operating status of the DSA to an external controller or device. Status queries are presented in one group and Event commands in another.

Status Queries

Command	Meaning
CONFIG?	Returns which type of plug-in units are installed.
ID?	Returns version numbers of system firmware.
IDPROBE?	Returns the channel number of the probe ID button last pressed.
PIVERSION?	Returns version numbers of plug-in unit firmware.
UID?	Returns serial numbers of the DSA and its plug-in units.



Event Commands

Command	Meaning
EVENT?	Returns event code information.
RQS	Sets whether the DSA asserts the SRQ line after an event occurs (GPIB only).
SRQMASK	Controls (masks) reporting of certain classes of event.
STBYTE?	Returns status byte information (RS-232-C only).

Time Base/ Horizontal Commands

The Time Base/Horizontal commands control horizontal position, establish acquisition scaling, and select a time base. The Time Base/Horizontal group contains these commands:

Time Base/Horizontal Commands

Command	Meaning
ADJTRACE <*ui*>	Controls the horizontal magnification and position of the selected waveform.
MAINPOS	Sets the horizontal position of the main waveform record with respect to the main trigger.
TBMAIN	Sets the main horizontal (time base) parameters.
TBWIN	Sets the window horizontal parameters.
WIN1POS WIN2POS	Sets the horizontal position of the window 1 or window 2 waveform record with respect to the window trigger.

Triggering Commands

The Trigger commands select and define the operation of the triggering system. You can define triggers on main or window waveforms. The Trigger commands are:

Triggering Commands

Command	Meaning
TR?	Returns the same information as: TRMAIN?;TRWIN?
TRMAIN	Sets the main trigger parameters.
TRWIN	Sets the window trigger parameters.
TSMAIN?	Returns the time from the trigger point to the 0 point, for real time single-shot acquisitions only.
WTMODE	Sets the window triggering mode.

Waveform and Settings Commands

Waveform and Setting commands select, store, remove, and specify waveform and front panel setting (FPS) characteristics. Waveform and Settings commands are presented in two groups: waveform commands and front panel settings commands.

Waveform Commands

Command	Meaning
ADJTRACE <*ui*>	Controls pan/zoom mode, vertical size and position, and window trace separation.
CLEAR	Discards acquired data for displayed waveforms.
DELETE	Deletes stored waveforms or front panel settings.
PZMODE	Controls multiple waveform pan/zoom mode.
REMOVE	Discards displayed waveforms and descriptions.
SCANSTOWFM	Controls scanning of stored waveforms.
SELECT	Designates the selected waveform.
STOLIST?	Returns a list of all stored waveforms.
STONUM?	Returns the number of stored waveforms.

Waveform Commands

Command	Meaning
STORE	Copies displayed waveforms to memory.
TRACE<*ui*>	Defines a waveform and its characteristics.
TRANUM?	Returns the number of displayed waveforms.
WFMSCALING	Sets a scaling flag to create new waveforms in fast (integer) or high-precision (floating-point) mode.

Front Panel Settings Commands

Command	Meaning
DELETE	Deletes stored waveforms or front panel settings.
FPSLIST?	Returns a list of stored front panel settings.
FPSNUM?	Returns the number of stored front panel settings.
NVRAM?	Returns the amount of available nonvolatile RAM.
RECALL	Recalls stored front panel settings from memory.
SETSEQ	Controls sequencing of front panel settings.
STORE	Stores front panel settings in nonvolatile RAM.

Learning by Example

The eight examples in this section demonstrate how to program typical DSA operations. These examples parallel those given in the *DSA 601 and DSA 602 Tutorial Manual*.

To run the examples, you need a basic knowledge of how to use the DSA from the front panel, how to use an IBM PC/XT/AT or compatible computer, and how to write programs in BASIC.

Each example begins with a brief explanation of its purpose and a listing of new DSA commands in the example. A listing of each program line (GPIB version) follows.

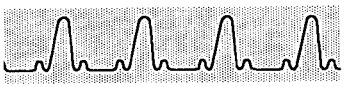
Required Hardware and Software

The programs included in this section are written for a Tektronix PEP301 Instrument Controller, or an IBM PC/XT/AT or other PC-compatible computer configured with a GPIB interface and running the GURU II GPIB controller software from Tektronix.

If you are using a different controller, or different software to control the GPIB, or prefer to use the RS-232-C interface, the examples should still prove useful. Since most of the calls to the GURU II software are invocations of IBWRT (send a message over the GPIB) or IBRD (receive a message over the GPIB), it is a simple matter to translate the programs to work with different hardware or software. To translate the examples for use over the RS-232-C interface, for example, most of the calls to IBWRT and IBRD can simply be converted to PRINT and INPUT statements referencing an appropriate device number.

The programs for this section are contained in the Learning by Example Software, on a single IBM-formatted $5^1/_4$-inch floppy disk located in the disk sleeve in the front of this manual. This software runs under most common BASIC language implementations, including:

- IBM BASIC.COM or BASICA.COM



The examples disk contains two directories, GPIB and RS232. The GPIB directory contains example programs to control the DSA using the GPIB interface. The RS232 directory contains programs to control the DSA using the RS-232-C interface.

Other hardware you will need includes the Pocket Signal Generator, which generates square waves used in the *DSA 601 and DSA 602 Tutorial*, and a GPIB or RS-232-C cable, as appropriate.

Installing the Learning by Example Software

Before running the examples, you should copy them onto your hard disk or to another floppy.

Hard Disk Installation

To install the example programs onto a hard disk, create a directory on the hard disk in which to store the example programs (600EXMPL might be a good choice for a name), using the MKDIR command from MS-DOS:

```
MKDIR 600EXMPL
```

Then, make that directory the current directory:

```
CD 600EXMPL
```

Insert the examples disk into drive A in your computer. If you want to use the GPIB programs, execute the following command:

```
COPY A:\GPIB\*.*
```

If you want to use the RS-232-C programs instead, execute the following command:

```
COPY A:\RS232\*.*
```

Once installation is complete, put the examples disk into the disk jacket in the manual for safekeeping.

Floppy Disk Installation

To install the examples onto a floppy disk in a dual-floppy-drive system, insert the examples disk into drive A:, and a formatted target disk into drive B:. Create a directory on the target floppy disk to store the examples (600EXMPL might be a good choice for a name), using the MKDIR command from MS-DOS:

```
MKDIR B:600EXMPL
```

Then, make that directory the current directory:

```
CD B:600EXMPL
```

If you want to use the GPIB programs, execute the following command:

```
COPY A:\GPIB\*.* B:600EXMPL
```

If you want to use the RS-232-C programs instead, execute the following command:

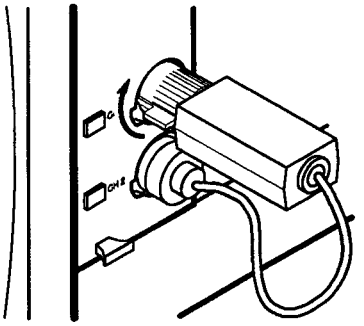
```
COPY A:\RS232\*.* B:600EXMPL
```

If you intend to use the target floppy disk as a start-up disk (it must be formatted with the /S option in order to do this), copy the following additional files from your original start-up disk onto the target disk:AUTOEXEC.BAT, CONFIG.SYS, and BASIC.COM (or BASICA.COM, or whatever the name of your BASIC program file is.)

Once installation is complete, put the examples disk into the disk jacket in the manual for safekeeping.

Attaching the Pocket Signal Generator

To run the examples, you must connect the pocket signal generator to channels 1 and 2 of the left plug-in amplifier. Connect the large end of the pocket signal generator to channel 1; connect the other end to channel 2. See the following illustration.



How the Pocket Signal Generator Should Be Connected

Running the Learning by Example Software

You can run the programs in either of two ways: from the MENU program, or individually.

To run the programs from the menu, make sure that the current directory is the directory where the MENU program resides, for example,

```
600EXMPL\GPIB
```

Then, enter:

```
BASIC MENU
```

or BASICA MENU, or whatever the name of your BASIC application happens to be.

The MENU program displays a list of programs for you to choose from. Type the number of the program you want to run, and press ENTER.

To run an individual program, make sure that the current directory is the directory where the program resides, for example,

```
600EXMPL\GPIB
```

Then, type the name of the program, followed by *<ENTER>*, for example,

```
BASIC SINGLE.BAS
```

or BASICA SINGLE.BAS, or whatever the name of your BASIC application happens to be.

Exiting the Learning by Example Software

When a program completes, you can type:

- *<ENTER>* (which returns you to the MENU program)
- **Q** (which exits the program and leaves you in BASIC)
- **S** (which exits the program and returns control to MS-DOS

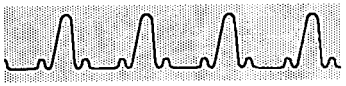
To exit a program without completing it, simply press CTRL-C. This will leave you in BASIC (most likely with a disabled front panel; see below). After re-enabling front panel operation, execute a SYSTEM command to return to MS-DOS.

Enabling Front Panel Operation

The programs disable front panel operation as part of the first command sent by the program, and re-enable it with the last command. Thus, if a program is allowed to complete, front panel operation is always enabled.

However, if you exit a program prematurely, front panel operation will most likely be disabled. To re-enable front panel operation, do one of the following:

- Cycle power on the DSA



- Execute the following BASIC commands, if your computer is communicating with the DSA over GPIB:

  ```
  WRT$ = "FPANEL ON"
  CALL IBWRT(TEKDEV1%,WRT$)
  ```

- Execute the following command, if your computer is communicating with the DSA over RS-232-C:

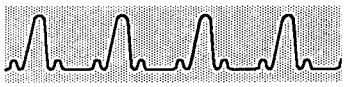
  ```
  PRINT #1,"FPANEL ON"
  ```

Setting GPIB Device 0 to "TEKDEV1"

In using the examples with a PEP controller or GURU II software, it is important that the name of GPIB device 0 be set to "TEKDEV1." Use the IBCONF program that came with your PEP controller or GURU II software to check for this name, and change GPIB device 0 to be "TEKDEV1" if necessary.

MENU.BAS Program

The MENU.BAS program is the starting point for the examples. After you invoke it, a menu appears on the screen that lists the number and title of each example. Type the number of the example you wish to run, then press *<ENTER>*.



MENU.BAS Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Examples Menu"
130 PRINT
140 PRINT "   1) Displaying a Single Waveform"
150 PRINT "   2) Managing Multiple Waveforms"
160 PRINT "   3) Defining Complex Waveforms"
170 PRINT "   4) Using Signal Processing"
180 PRINT "   5) Taking Automated Measurements"
190 PRINT "   6) Comparing to a Reference Measurement"
200 PRINT "   7) Taking Delay Measurements Using Cursors"
205 PRINT "   8) Comparing Waveforms to Stored Waveforms"
210 PRINT
220 PRINT "Enter the number of the example you wish to run,"
230 PRINT "press 'Q' to quit without exiting BASIC,"
240 PRINT "or press 'S' to quit and exit BASIC."
250 INPUT PROGNUM$
260 IF LEFT$(PROGNUM$,1)="Q" OR LEFT$(PROGNUM$,1)="q" THEN
    END
270 IF LEFT$(PROGNUM$,1)="S" OR LEFT$(PROGNUM$,1)="s" THEN
    SYSTEM
280 PROGNUM=VAL(PROGNUM$)
290 IF PROGNUM<1 OR PROGNUM>8 THEN GOTO 220
300 ON PROGNUM GOTO 310,320,330,340,350,360,370,380
310 LOAD "SINGLE.BAS",R
320 LOAD "MULTW.BAS",R
330 LOAD "COMPLEX.BAS",R
340 LOAD "SIGPRO.BAS",R
350 LOAD "AUTOMEAS.BAS",R
360 LOAD "REFMEAS.BAS",R
370 LOAD "DLAYMEAS.BAS",R
380 LOAD "STOWFM.BAS",R
```

Example 1: Displaying a Waveform

Example 1 demonstrates how to display a waveform on the screen, size it, and move it vertically and horizontally.

Commands and links introduced in this example include:

- AUTOSET START automatically scales the waveforms on the display.
- CH<*slot*><*ui*> IMPEDANCE sets the impedance of a specified channel.
- CH<*slot*><*ui*> OFFSET sets the vertical offset of a specified channel (moves a waveform up or down).
- CH<*slot*><*ui*> SENSITIVITY sets the "sensitivity," or vertical volts/division, of a specified channel. Use this command to adjust the size of a waveform on the display.
- DEBUG GPIB, when set to ON, causes commands input to the GPIB port to be displayed at the top of the screen.
- FPANEL, when set to OFF, disables front panel operation; when set to ON, enables it again.
- INIT returns the DSA to default and settings.
- LONGFORM when set to ON, causes query responses to contain full header and link spellings; when set to OFF, query responses are in abbreviated form.
- MAINPOS sets the horizontal position of the Main waveform record with respect to the Main trigger. This command acts like a horizontal-axis "offset" for waveforms.
- TBMAIN TIME sets the horizontal scale (time/division).
- TRACE<*ui*> DESCRIPTION defines the source description of a specified waveform.
- TRMAIN ANLEVEL sets the main trigger level value.

Example 1 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 1: Displaying a Waveform"
130 PRINT "(GPIB Version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m);
210 IBINIT1 = 58900!        'a smaller-than-calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":END
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to display a waveform from CH 1 of
    the left plug-in amplifier.",A$
330 WRT$="trace1 description:'L1'"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to position the waveform.",A$
360 WRT$="ch11 imp:50;autoset start"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to set the main time base to
    20 usec/div.",A$
390 WRT$="tbmain time:20e-6"
400 CALL IBWRT(TEKDEV1%,WRT$)
410 INPUT "Press ENTER to set the main time base to
    5 usec/div.",A$
420 WRT$="tbmain time:5e-6"
430 CALL IBWRT(TEKDEV1%,WRT$)
440 INPUT "Press ENTER to set the main time base to
    10 usec/div.",A$
450 WRT$="tbmain time:10e-6"
460 CALL IBWRT(TEKDEV1%,WRT$)
```

```
470 INPUT "Press ENTER to set the vertical size of channel
    L1 to 500mV/div.",A$
480 WRT$="ch11 sen:500e-3"
490 CALL IBWRT(TEKDEV1%,WRT$)
500 INPUT "Press ENTER to set the vertical size of channel
    L1 to 100mV/div.",A$
510 WRT$="ch11 sen:100e-3"
520 CALL IBWRT(TEKDEV1%,WRT$)
530 INPUT "Press ENTER to set the vertical size of channel
    L1 to 200mV/div.",A$
540 WRT$="ch11 sen:200e-3"
550 CALL IBWRT(TEKDEV1%,WRT$)
560 PRINT "Press ENTER to set the vertical offset of channel
    L1 to 1V."
570 INPUT "(Screen will indicate waveform not
    triggered.)",A$
580 WRT$="ch11 offs:1"
590 CALL IBWRT(TEKDEV1%,WRT$)
600 PRINT "Press ENTER to set the vertical offset of channel
    L1 to -400mV."
610 INPUT "(Screen will indicate waveform not trig
    gered.)",A$
620 WRT$="ch11 offs:-400e-3"
630 CALL IBWRT(TEKDEV1%,WRT$)
640 INPUT "Press ENTER to set the vertical offset of channel
    L1 to 400mV.",A$
650 WRT$="ch11 offs:400e-3"
660 CALL IBWRT(TEKDEV1%,WRT$)
670 INPUT "Press ENTER to re-enable the front panel.",A$
680 WRT$="fpanel on"
690 CALL IBWRT(TEKDEV1%,WRT$)
700 PRINT "End of example 1."
710 PRINT "Press ENTER to return to the Examples menu,"
720 PRINT "press 'Q' to quit the program without exiting
    BASIC,"
730 INPUT "or press 'S' to quit the program and exit
    BASIC.",A$
740 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
750 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
    LOAD "menu.bas",R
```

Example 2: Managing Multiple Waveforms

Example 2 demonstrates how to display, label, and control multiple waveforms, how to divide the display into two graticules, and how to return the display to a single graticule.

Commands and links introduced in this example include:

- DISPLAY GRATICULE selects single or dual display graticules.
- LABABS PCTG sets the horizontal position of a label on the selected waveform as a percentage of the waveform record.
- LABEL TRACE<*ui*> defines a label for a specified waveform.
- REMOVE TRACE<*ui*> removes a specified waveform from the display.
- SELECT TRACE<*ui*> selects a specified waveform.
- TRACE<*ui*> GRLOCATION positions the selected waveform to the upper or lower graticule.

Example 2 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 2: Managing Multiple Waveforms"
130 PRINT "(GPIB Version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m);
210 IBINIT1 = 58900!        'a smaller-than-calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":END
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to display a waveform from CH 1 of
    the left plug-in amplifier.",A$
330 WRT$="trace1 description:'L1'"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to position the waveform.",A$
360 WRT$="tbmain time:10e-6;mainpos -8e-6;ch11 imp:50;ch11
    sen:200e-3;ch11 offs:400e-3;trm anl:500e-3,vol"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to display a waveform from CH 2 of
    the left plug-in amplifier.",A$
390 WRT$="trace2 description:'L2'"
400 CALL IBWRT(TEKDEV1%,WRT$)
410 INPUT "Press ENTER to position the waveform.",A$
420 WRT$="ch12 imp:50;ch12 sen:200e-3;ch12 offs:600e-3"
430 CALL IBWRT(TEKDEV1%,WRT$)
440 INPUT "Press ENTER to select trace 1.",A$
450 WRT$="sel tra1"
460 CALL IBWRT(TEKDEV1%,WRT$)
```

```
470 INPUT "Press ENTER to label trace 1 and position its
    label.",A$
480 WRT$="label display:on;label tra1:'Trace 1';lababs
    pctg:10"
490 CALL IBWRT(TEKDEV1%,WRT$)
500 INPUT "Press ENTER to select trace 2.",A$
510 WRT$="sel tra2"
520 CALL IBWRT(TEKDEV1%,WRT$)
530 INPUT "Press ENTER to label trace 2 and position its
    label.",A$
540 WRT$="label tra2:'Trace 2';lababs pctg:25"
550 CALL IBWRT(TEKDEV1%,WRT$)
560 INPUT "Press ENTER to display two graticules.",A$
570 WRT$="disp gra:dua"
580 CALL IBWRT(TEKDEV1%,WRT$)
590 INPUT "Press ENTER to move trace 2 to the upper
    graticule.",A$
600 WRT$="tra2 grl:upper"
610 CALL IBWRT(TEKDEV1%,WRT$)
620 INPUT "Press ENTER to move trace 1 to the lower
    graticule.",A$
630 WRT$="tra1 grl:lower"
640 CALL IBWRT(TEKDEV1%,WRT$)
650 INPUT "Press ENTER to return the display to a single
    graticule.",A$
660 WRT$="disp gra:sin"
670 CALL IBWRT(TEKDEV1%,WRT$)
680 INPUT "Press ENTER to remove trace 2.",A$
690 WRT$="rem tra2"
700 CALL IBWRT(TEKDEV1%,WRT$)
710 INPUT "Press ENTER to re-enable the front panel.",A$
720 WRT$="fpanel on"
730 CALL IBWRT(TEKDEV1%,WRT$)
740 PRINT "End of example 2."
750 PRINT "Press ENTER to return to the Examples menu,"
760 PRINT "press 'Q' to quit the program without exiting
    BASIC,"
770 INPUT "or press 'S' to quit the program and exit
    BASIC.",A$
780 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
790 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
    LOAD "menu.bas",R
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
```

Example 3: Defining Complex Waveforms

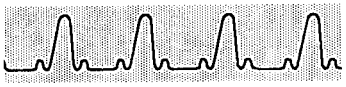
Example 3 demonstrates how to create waveforms as combinations of other waveforms, and how to display window waveforms.

Commands and links introduced in this example include:

- ADJTRACE<*ui*> TRSEP moves a specified window waveform up or down.
- TBWIN TIME sets the horizontal scale (time/division) of a window on the display.
- TRMAIN MODE selects the main triggering mode: AUTOLEVEL, AUTO, or NORMAL.
- WIN1POS sets the position of the window 1 acquisition record with respect to the window trigger.

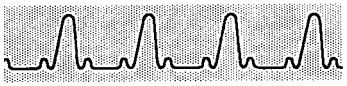
Example 3 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 3: Defining Complex Waveforms"
130 PRINT "(GPIB version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m)
210 IBINIT1 = 58900!        'a smaller than calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":PRINT
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to set the trigger mode to
    AUTO-LEVEL.",A$
330 WRT$="trm mod:autol"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to display a waveform from CH 1 of
    the left plug-in amplifier.",A$
360 WRT$="trace1 description:'L1'"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to set up waveform 1.",A$
390 WRT$="tbm tim:10e-6;mainpos -7e-6;ch11 imp:50;ch11
    sens:0.4;ch11 offs:-0.8"
400 CALL IBWRT(TEKDEV1%,WRT$)
410 INPUT "Press ENTER to display a waveform from CH 2 of
    the left plug-in amplifier.",A$
420 WRT$="trace2 description:'L2'"
430 CALL IBWRT(TEKDEV1%,WRT$)
440 INPUT "Press ENTER to set up waveform 2.",A$
450 WRT$="ch12 imp:50;ch12 sens:0.4;ch12 offs:1.6"
460 CALL IBWRT(TEKDEV1%,WRT$)
```



```
470 INPUT "Press ENTER to create a new trace, equal to the
    sum of channels L1 and L2.",A$
480 WRT$="tra3 des:'L1+L2'"
490 CALL IBWRT(TEKDEV1%,WRT$)
500 PRINT "Press ENTER to change the offset of CH L1."
510 INPUT "(Note the effect on trace 3.)",A$
520 REM
530 REM save the original offset of CH L1
540 REM
550 WRT$="ch11? offs"
560 CALL IBWRT(TEKDEV1%,WRT$)
570 RD$=SPACE$(64)
580 CALL IBRD(TEKDEV1%,RD$)
590 CHL1OFFS$=RD$
600 WHILE RIGHT$(CHL1OFFS$,1)=" "
610 CHL1OFFS$=LEFT$(CHL1OFFS$,LEN(CHL1OFFS$)-1)
620 WEND
630 WRT$="ch11 offs:-0.4"
640 CALL IBWRT(TEKDEV1%,WRT$)
650 PRINT "Press ENTER to change the offset of CH L1 again."
660 INPUT "(Note the effect on trace 3 again.)",A$
670 WRT$="ch11 offs:0"
680 CALL IBWRT(TEKDEV1%,WRT$)
690 INPUT "Press ENTER to change the offset of CH L1 back to
    its original value.",A$
700 WRT$=CHL1OFFS$
710 CALL IBWRT(TEKDEV1%,WRT$)
720 INPUT "Press ENTER to change the offset of CH L2.",A$
730 REM save the original offset of CH L2
740 WRT$="ch12? offs"
750 CALL IBWRT(TEKDEV1%,WRT$)
760 RD$=SPACE$(64)
770 CALL IBRD(TEKDEV1%,RD$)
780 CHL2OFFS$=RD$
790 WHILE RIGHT$(CHL2OFFS$,1)=" "
800 CHL2OFFS$=LEFT$(CHL2OFFS$,LEN(CHL2OFFS$)-1)
810 WEND
820 WRT$="ch12 offs:2.1"
830 CALL IBWRT(TEKDEV1%,WRT$)
840 INPUT "Press ENTER to change the offset of CH L2 back to
    its original value.",A$
850 WRT$=CHL2OFFS$
860 CALL IBWRT(TEKDEV1%,WRT$)
870 INPUT "Press ENTER to remove waveforms 1 and 2.",A$
880 WRT$="rem tra1;rem tra2"
890 CALL IBWRT(TEKDEV1%,WRT$)
900 INPUT "Press ENTER to display part of waveform 3 in a
    window.",A$
910 WRT$="tra4 des:'L1+L2 on win1'"
920 CALL IBWRT(TEKDEV1%,WRT$)
```

```
930 INPUT "Press ENTER to change the window position.",A$
940 WRT$="win1pos 15.5e-6"
950 CALL IBWRT(TEKDEV1%,WRT$)
960 INPUT "Press ENTER to change the size of window 1.",A$
970 WRT$="tbwin tim:2e-6"
980 CALL IBWRT(TEKDEV1%,WRT$)
990 INPUT "Press ENTER to display another part of waveform 3
    in a window.",A$
1000 WRT$="tra5 des:'L1+L2 on win2'"
1010 CALL IBWRT(TEKDEV1%,WRT$)
1020 INPUT "Press ENTER to move the 2nd window waveform up
     by two graticule divisions.",A$
1030 WRT$="adj5 trs:2"
1040 CALL IBWRT(TEKDEV1%,WRT$)
1050 INPUT "Press ENTER to re-enable the front panel.",A$
1060 WRT$="fpanel on"
1070 CALL IBWRT(TEKDEV1%,WRT$)
1080 PRINT "End of example 3."
1090 PRINT "Press ENTER to return to the Examples menu,"
1100 PRINT "press 'Q' to quit the program without exiting
     BASIC,"
1110 INPUT "or press 'S' to quit the program and exit
     BASIC.",A$
1120 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
1130 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
     LOAD "menu.bas",R
```



Example 4: Using Signal Processing

CAUTION
This program should NOT be run with an 11A52 or 11A72 Amplifier. Only run with an 11A32 or 11A34 Amplifier.

Example 4 demonstrates how to use the signal processing features of the DSA to provide more information about a waveform than is available from a "normal" view.

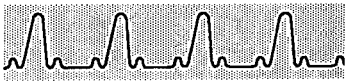
Commands and links introduced in this example include:

- CH<*slot*> <*ui*> BW sets the bandwidth of the specified channel.
- CONDACQ TYPE sets the acquisition type, such as averaging, continuous, enveloping, repetitive trigger, etc.
- NAVG sets the number of waveform samples to be used for averaging.
- TBMAIN LENGTH sets the record length of the Main time base in points per waveform.
- TRACE<*ui*> ACCUMULATE sets point accumulate mode ON or OFF for the specified waveform.

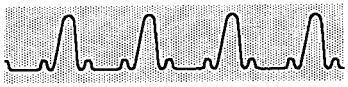
Example 4 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 4: Using Signal Processing"
130 PRINT "(GPIB version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m)
210 IBINIT1 = 58900!        'a smaller than calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":PRINT
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to display a waveform from CH 2 of
    the left plug-in amplifier.",A$
330 WRT$="trace1 description:'L2'"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to set up the waveform.",A$
360 WRT$="tbm tim:10e-6;mainpos -1.2e-6;chl2 imp:1e6;chl2
    sens:1;chl2 offs:4;trm anl:2.7,vol"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to re-enable the front panel.",A$
390 WRT$="fpanel on"
400 CALL IBWRT(TEKDEV1%,WRT$)
410 PRINT "The front panel is now re-enabled."
420 PRINT "Set the Vertical Offset resolution to FINE"
430 PRINT "(touch the vertical arrows selector,"
440 PRINT "touch the Vertical Offset selector, then press
    FINE)"
450 PRINT "and adjust the vertical offset until you find a
    point"
460 PRINT "near the vertical center of the waveform"
```

```
470 PRINT "where triggering is unstable"
480 PRINT "(i.e., where the waveform moves around on the
    screen)."
490 INPUT "Then press ENTER to continue the program.",A$
500 WRT$="fpanel off"
510 CALL IBWRT(TEKDEV1%,WRT$)
520 INPUT "Press ENTER to set CH 2's Coupling to DC High-
    Frequency Reject.",A$
530 WRT$="trm cou:dchf"
540 CALL IBWRT(TEKDEV1%,WRT$)
550 INPUT "Press ENTER to set CH 2's Coupling back to
    DC.",A$
560 WRT$="trm cou:dc"
570 CALL IBWRT(TEKDEV1%,WRT$)
580 INPUT "Press ENTER to set main size to 50ns/div, main
    position to -105ns.",A$
590 WRT$="tbm tim:50e-9;mainpos -105e-9"
600 CALL IBWRT(TEKDEV1%,WRT$)
610 INPUT "Press ENTER to turn Point Accumulate mode on.",A$
620 WRT$="tra1 acc:on"
630 CALL IBWRT(TEKDEV1%,WRT$)
640 INPUT "Press ENTER to turn Point Accumulate mode
    off.",A$
650 WRT$="tra1 acc:off"
660 CALL IBWRT(TEKDEV1%,WRT$)
670 INPUT "Press ENTER to set the trigger level voltage one
    division higher.",A$
680 REM
690 REM get the original sensitivity value
700 REM
710 WRT$="ch12? sens"
720 CALL IBWRT(TEKDEV1%,WRT$)
730 RD$=SPACE$(63)
740 CALL IBRD(TEKDEV1%,RD$)
750 SENS$=RD$
760 REM
770 REM process the returned string
780 REM
790 WHILE RIGHT$(SENS$,1)=" "
800 SENS$=LEFT$(SENS$,LEN(SENS$)-1)
810 WEND
820 REM
830 REM find the position of the ":" character
840 REM
850 COLPOS=INSTR(1,SENS$,":")
860 REM
870 REM extract the numeric value
880 REM
890 SENS=VAL(MID$(SENS$,COLPOS+1,LEN(SENS$)))
900 REM
```



```
910 REM get the original main trigger level
920 REM
930 WRT$="trm? anl"
940 CALL IBWRT(TEKDEV1%,WRT$)
950 RD$=SPACE$(63)
960 CALL IBRD(TEKDEV1%,RD$)
970 TRM$=RD$
980 REM
990 REM process the returned string
1000 REM
1010 WHILE RIGHT$(TRM$,1)=" "
1020 TRM$=LEFT$(TRM$,LEN(TRM$)-1)
1030 WEND
1040 REM
1050 REM find the position of the "," character
1060 REM
1070 COMPOS=INSTR(1,TRM$,",")
1080 REM
1090 REM extract the units string
1100 REM
1110 UNIT$=MID$(TRM$,COMPOS+1,LEN(TRM$))
1120 TRM$=MID$(TRM$,1,COMPOS-1)
1130 REM
1140 REM find the position of the ":" character
1150 REM
1160 COLPOS=INSTR(1,TRM$,":")
1170 REM
1180 REM extract the numeric value
1190 REM
1200 TRM=VAL(MID$(TRM$,COLPOS+1,LEN(TRM$)))
1210 IF UNIT$<>"VOLTS" THEN TRM=TRM*SENS
1220 REM
1230 REM calculate new trigger level
1240 REM
1250 TRM=TRM+SENS
1260 REM
1270 REM put a new command string together and send it
1280 REM
1290 TRM$=STR$(TRM)
1300 WRT$="trm anl:"+TRM$+",volts"
1310 CALL IBWRT(TEKDEV1%,WRT$)
1320 INPUT "Press ENTER to display the average value of
     CH 2.",A$
1330 WRT$="tbm tim:10e-6;tra1 desc:'avg(L2)'"
1340 CALL IBWRT(TEKDEV1%,WRT$)
1350 INPUT "Press ENTER to set the number of points to be
     averaged to 128.",A$
1360 WRT$="navg 128"
1370 CALL IBWRT(TEKDEV1%,WRT$)
```



```
1380 INPUT "Press ENTER to stop acquisition after 128 data
     points.",A$
1390 WRT$="cond typ:avg"
1400 CALL IBWRT(TEKDEV1%,WRT$)
1410 INPUT "Press ENTER to re-start continuous acquisi
     tion.",A$
1420 WRT$="cond typ:conti"
1430 CALL IBWRT(TEKDEV1%,WRT$)
1440 INPUT "Press ENTER to display the original waveform
     from CH 2.",A$
1450 WRT$="tra1 desc:'L2'"
1460 CALL IBWRT(TEKDEV1%,WRT$)
1470 INPUT "Press ENTER to set the main record length to
     512.",A$
1480 REM
1490 REM save the original record length
1500 REM
1510 WRT$="tbm? len"
1520 CALL IBWRT(TEKDEV1%,WRT$)
1530 RD$=SPACE$(64)
1540 CALL IBRD(TEKDEV1%,RD$)
1550 TBMLEN$=RD$
1560 WHILE RIGHT$(TBMLEN$,1)=" "
1570 TBMLEN$=LEFT$(TBMLEN$,LEN(TBMLEN$)-1)
1580 WEND
1590 WRT$="tbm len:512"
1600 CALL IBWRT(TEKDEV1%,WRT$)
1610 INPUT "Press ENTER to set the main record length to its
     upper limit.",A$
1620 WRT$="tbm len:32768"
1630 CALL IBWRT(TEKDEV1%,WRT$)
1640 INPUT "Press ENTER to set the main record length back
     to its original value.",A$
1650 WRT$=TBMLEN$
1660 CALL IBWRT(TEKDEV1%,WRT$)
1670 INPUT "Press ENTER to re-enable the front panel.",A$
1680 WRT$="fpanel on"
1690 CALL IBWRT(TEKDEV1%,WRT$)
1700 PRINT "End of example 4."
1710 PRINT "Press ENTER to return to the Examples menu,"
1720 PRINT "press 'Q' to quit the program without exiting
     BASIC,"
1730 INPUT "or press 'S' to quit the program and exit
     BASIC.",A$
1740 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
1750 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
     LOAD "menu.bas",R
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
```

Example 5: Taking Automated Measurements

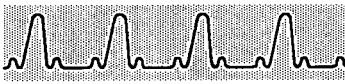
Example 5 demonstrates how to take measurements on a waveform.

Commands and links introduced in this example include:

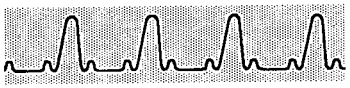
- MEAS? returns the current values of the measurements in the current measurements list.
- MSLIST selects the measurements to be included in the current measurements list.
- MSYS ON, when set to ON, displays the Measurements Major Menu at the bottom of the front panel display.

Example 5 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 5: Taking Automated Measurements"
130 PRINT "(GPIB version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m)
210 IBINIT1 = 58900!        'a smaller than calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":PRINT
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to display a waveform from CH 1 of
    the left plug-in amplifier.",A$
330 WRT$="trace1 description:'L1'"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to set up the waveform.",A$
360 WRT$="tbm tim:10e-6;mainpos -1.2e-6;ch11 imp:50;ch11
    sens:0.2;ch11 offs:0.8;trm anl:0.4,vol"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to add Peak-to-Peak and Frequency to
    measurements list.",A$
390 REM
400 REM get and print the current measurements list
410 REM
420 WRT$="mslist?"
430 CALL IBWRT(TEKDEV1%,WRT$)
440 RD$=SPACE$(64)
450 CALL IBRD(TEKDEV1%,RD$)
460 MSLIST$=RD$
470 WHILE RIGHT$(MSLIST$,1)=" "
```



```
480 MSLIST$=LEFT$(MSLIST$,LEN(MSLIST$)-1)
490 WEND
500 PRINT "The current measurements list is:"
510 PRINT MSLIST$
520 REM
530 REM add P-P and freq to measurements list
540 REM
550 WRT$="mslist pp,freq"
560 CALL IBWRT(TEKDEV1%,WRT$)
570 REM
580 REM get and print the new measurements list
590 REM
600 WRT$="mslist?"
610 CALL IBWRT(TEKDEV1%,WRT$)
620 RD$=SPACE$(64)
630 CALL IBRD(TEKDEV1%,RD$)
640 MSLIST$=RD$
650 WHILE RIGHT$(MSLIST$,1)=" "
660 MSLIST$=LEFT$(MSLIST$,LEN(MSLIST$)-1)
670 WEND
680 PRINT "The new measurements list is:"
690 PRINT MSLIST$
700 INPUT "Press ENTER to display the measurements menu.",A$
710 WRT$="msys on"
720 CALL IBWRT(TEKDEV1%,WRT$)
730 INPUT "Press ENTER to take a measurement.",A$
740 WRT$="meas?"
750 CALL IBWRT(TEKDEV1%,WRT$)
760 RD$=SPACE$(255)
770 CALL IBRD(TEKDEV1%,RD$)
780 MEAS$=RD$
790 WHILE RIGHT$(MEAS$,1)=" "
800 MEAS$=LEFT$(MEAS$,LEN(MEAS$)-1)
810 WEND
820 PRINT "The measurement results are:"
830 PRINT MEAS$
840 INPUT "Press ENTER to re-enable the front panel.",A$
850 WRT$="fpanel on"
860 CALL IBWRT(TEKDEV1%,WRT$)
870 PRINT "End of example 5."
880 PRINT "Press ENTER to return to the Examples menu,"
890 PRINT "press 'Q' to quit the program without exiting
    BASIC,"
900 INPUT "or press 'S' to quit the program and exit
    BASIC.",A$
910 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
920 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
    LOAD "menu.bas",R
```



Example 6: Comparing to a Reference Measurement

Example 6 demonstrates how to set up a reference waveform and compare it to other waveforms.

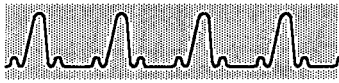
Commands and links introduced in this example include:

- COMPARE, when set to ON, returns the difference between the measurement (made on the selected waveform) and measurement's reference value (made on a reference waveform); when set to OFF, measurements return the value of the measurement.
- REFSET CURRENT executes a specified measurement and stores the result as a reference value.

Example 6 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 6: Comparing to a Reference Measurement"
130 PRINT "(GPIB version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m);
210 IBINIT1 = 58900!        'a smaller-than-calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":END
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to display a waveform from CH 1 of
    the left plug-in amplifier.",A$
330 WRT$="trace1 description:'L1'"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to set up the waveform.",A$
360 WRT$="tbm tim:10e-6;mainpos -1.2e-6;ch11
    imp:50,sens:0.2,offs:0.8;trm anl:0.4,vol"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to display a waveform from CH 2 of
    the left plug-in amplifier.",A$
390 WRT$="trace2 description:'L2'"
400 CALL IBWRT(TEKDEV1%,WRT$)
410 INPUT "Press ENTER to set up the waveform.",A$
420 WRT$="ch12 imp:50,sens:0.2,offs:0.2"
430 CALL IBWRT(TEKDEV1%,WRT$)
440 INPUT "Press ENTER to add Peak-to-Peak to the measure
    ments list.",A$
450 REM
460 REM get and display the current measurements list
```

```
470 REM
480 WRT$="mslist?"
490 CALL IBWRT(TEKDEV1%,WRT$)
500 RD$=SPACE$(64)
510 CALL IBRD(TEKDEV1%,RD$)
520 MSLIST$=RD$
530 WHILE RIGHT$(MSLIST$,1)=" "
540 MSLIST$=LEFT$(MSLIST$,LEN(MSLIST$)-1)
550 WEND
560 PRINT "The current measurements list is:"
570 PRINT MSLIST$
580 REM
590 REM add P-P to measurements list
600 REM
610 WRT$="mslist pp"
620 CALL IBWRT(TEKDEV1%,WRT$)
630 REM
640 REM get and display the new measurements list
650 REM
660 WRT$="mslist?"
670 CALL IBWRT(TEKDEV1%,WRT$)
680 RD$=SPACE$(64)
690 CALL IBRD(TEKDEV1%,RD$)
700 MSLIST$=RD$
710 WHILE RIGHT$(MSLIST$,1)=" "
720 MSLIST$=LEFT$(MSLIST$,LEN(MSLIST$)-1)
730 WEND
740 PRINT "The new measurements list is:"
750 PRINT MSLIST$
760 INPUT "Press ENTER to display the measurements menu.",A$
770 WRT$="msys on"
780 CALL IBWRT(TEKDEV1%,WRT$)
790 INPUT "Press ENTER to take a measurement.",A$
800 GOSUB 1040
810 INPUT "Press ENTER to use Peak-to-Peak as a reference
    measurement.",A$
820 WRT$="refset current:pp"
830 CALL IBWRT(TEKDEV1%,WRT$)
840 INPUT "Press ENTER to turn COMPARE on.",A$
850 WRT$="compare on"
860 CALL IBWRT(TEKDEV1%,WRT$)
870 INPUT "Press ENTER to take a comparison measurement.",A$
880 WRT$="select tra1"
890 CALL IBWRT(TEKDEV1%,WRT$)
900 GOSUB 1040
910 INPUT "Press ENTER to turn COMPARE off.",A$
920 WRT$="compare off"
930 CALL IBWRT(TEKDEV1%,WRT$)
940 INPUT "Press ENTER to re-enable the front panel.",A$
950 WRT$="fpanel on"
```



```
960 CALL IBWRT(TEKDEV1%,WRT$)
970 PRINT "End of example 6."
980 PRINT "Press ENTER to return to the Examples menu,"
990 PRINT "press 'Q' to quit the program without exiting
    BASIC,"
1000 INPUT "or press 'S' to quit the program and exit
    BASIC.",A$
1010 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
1020 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
     LOAD "menu.bas",R
1030 REM
1040 REM subroutine to take measurements
1050 REM
1060 WRT$="meas?"
1070 CALL IBWRT(TEKDEV1%,WRT$)
1080 RD$=SPACE$(64)
1090 CALL IBRD(TEKDEV1%,RD$)
1100 MEAS$=RD$
1110 WHILE RIGHT$(MEAS$,1)=" "
1120 MEAS$=LEFT$(MEAS$,LEN(MEAS$)-1)
1130 WEND
1140 PRINT "The measurement results are:"
1150 PRINT MEAS$
1160 RETURN
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
```

Example 7: Taking Delay Measurements Using Cursors

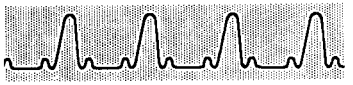
Example 7 demonstrates how to use cursors adjusted from the front panel to control automated measurements.

Commands and links introduced in this example include:

- AUTOSET VERT determines and sets the vertical size and offset of the input signal.
- CURSOR TYPE selects the cursor type.
- DOT1ABS? XCO, DOT2 ABS? XCO returns the position of the first or second dot cursor with respect to the horizontal units of the selected waveform.
- V1BAR? XCO, V1BAR? XCO returns the position of the first or second vertical bar cursor with respect to the horizontal units of the selected waveform.

Example 7 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 7: Taking Delay Measurements Using
    Cursors"
130 PRINT "(GPIB version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m);
210 IBINIT1 = 58900!        'a smaller-than-calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":END
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to display a waveform from CH 1 of
    the left plug-in amplifier.",A$
330 WRT$="trace1 description:'L1'"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to set up the waveform.",A$
360 WRT$="tbm tim:5e-6;mainpos -2e-6;ch11
    imp:50,sens:0.2,offs:0.8;trm anl:0.5,vol"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to display a waveform from CH 2 of
    the left plug-in amplifier.",A$
390 WRT$="trace2 description:'L2'"
400 CALL IBWRT(TEKDEV1%,WRT$)
410 INPUT "Press ENTER to set up the waveform.",A$
420 WRT$="ch12 imp:50,sens:0.2,offs:0.2"
430 CALL IBWRT(TEKDEV1%,WRT$)
440 INPUT "Press ENTER to display a pair of vertical-bar
    cursors.",A$
450 WRT$="cursor type:vbars,readout:on"
```



```
460 CALL IBWRT(TEKDEV1%,WRT$)
470 INPUT "Press ENTER to re-enable the front panel.",A$
480 WRT$="fpanel on"
490 CALL IBWRT(TEKDEV1%,WRT$)
500 PRINT "The front panel is now re-enabled."
510 PRINT "Use the left knob to move the left cursor"
520 PRINT "to the first rising edge of the selected trace."
530 PRINT "Use the right knob to move the right cursor"
540 PRINT "to the first falling edge after the first rising
    edge."
550 INPUT "Press ENTER when ready.",A$
560 WRT$="fpanel off"
570 CALL IBWRT(TEKDEV1%,WRT$)
580 REM
590 REM get cursor positions, calculate delta and 1/delta
600 REM
610 WRT$="v1bar? xco"
620 CALL IBWRT(TEKDEV1%,WRT$)
630 RD$=SPACE$(64)
640 CALL IBRD(TEKDEV1%,RD$)
650 PRINT RD$
660 GOSUB 1220
670 V1VAL=VALUE
680 WRT$="v2bar? xco"
690 CALL IBWRT(TEKDEV1%,WRT$)
700 RD$=SPACE$(64)
710 CALL IBRD(TEKDEV1%,RD$)
720 PRINT RD$
730 GOSUB 1220
740 V2VAL=VALUE
750 DELTA=V2VAL-V1VAL
760 PRINT "DELTA=";DELTA
770 PRINT "1/DELTA=";1/DELTA
780 INPUT "Press ENTER to display split-dot cursors.",A$
790 WRT$="cursor type:split,refe:trace1"
800 CALL IBWRT(TEKDEV1%,WRT$)
810 INPUT "Press ENTER to re-enable the front panel.",A$
820 WRT$="fpanel on"
830 CALL IBWRT(TEKDEV1%,WRT$)
840 PRINT "The front panel is now re-enabled."
850 PRINT "Use the left knob to move the left cursor"
860 PRINT "to the first rising edge of the selected trace."
870 PRINT "Use the right knob to move the right cursor"
880 PRINT "to the next rising edge of the non-selected
    trace."
890 INPUT "Press ENTER when ready.",A$
900 WRT$="fpanel off"
910 CALL IBWRT(TEKDEV1%,WRT$)
920 REM
930 REM get cursor positions, calculate delta and 1/delta
```

```
940 REM
950 WRT$="dot1abs? xco"
960 CALL IBWRT(TEKDEV1%,WRT$)
970 RD$=SPACE$(64)
980 CALL IBRD(TEKDEV1%,RD$)
990 PRINT RD$
1000 GOSUB 1220
1010 DOT1VAL=VALUE
1020 WRT$="dot2abs? xco"
1030 CALL IBWRT(TEKDEV1%,WRT$)
1040 RD$=SPACE$(64)
1050 CALL IBRD(TEKDEV1%,RD$)
1060 PRINT RD$
1070 GOSUB 1220
1080 DOT2VAL=VALUE
1090 DELTA=DOT2VAL-DOT1VAL
1100 PRINT "DELTA=";DELTA
1110 PRINT "1/DELTA=";1/DELTA
1120 INPUT "Press ENTER to re-enable the front panel.",A$
1130 WRT$="fpanel on"
1140 CALL IBWRT(TEKDEV1%,WRT$)
1150 PRINT "End of example 7."
1160 PRINT "Press ENTER to return to the Examples menu,"
1170 PRINT "press 'Q' to quit the program without exiting
     BASIC,"
1180 INPUT "or press 'S' to quit the program and exit
     BASIC.",A$
1190 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
1200 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
     LOAD "menu.bas",R
1210 REM
1220 REM subroutine to remove rightmost blanks, get link
     values
1230 REM
1240 WHILE RIGHT$(RD$,1)=" "
1250 RD$=LEFT$(RD$,LEN(RD$)-1)
1260 WEND
1270 COLPOS=INSTR(RD$,":")
1280 RD$=MID$(RD$,COLPOS+1,LEN(RD$))
1290 VALUE=VAL(RD$)
1300 RETURN
```

Example 8: Comparing Waveforms to Stored Waveforms

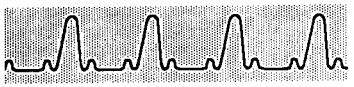
Example 8 demonstrates how to store and recall waveforms.

The commands and link introduced in this example is:

- STORE TRA<*ui*> stores a specified trace.

Example 8 Program Listing

```
100 CLS
110 PRINT "DSA 600 Series Digitizing Signal Analyzer"
120 PRINT "Example 8: Comparing Waveforms to Stored
    Waveforms"
130 PRINT "(GPIB version)"
140 PRINT
150 REM
160 REM decl.bas
170 REM
180 REM GURU initialization code; declarations
190 REM
200 CLEAR ,58900!           'IBM BASICA Declarations; =BYTES
    FREE -size(bib.m);
210 IBINIT1 = 58900!        'a smaller-than-calculated # is
    OK
220 IBINIT2 = IBINIT1 + 3   'these lines (thru CALL state
    ments below)
230 BLOAD "bib.m",IBINIT1   'MUST be included in your program
240 CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,
    IBPPC,IBBNA,IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,
    IBDMA,IBEOS,IBTMO,IBEOT,IBRDF,IBWRTF,IBTRAP)
250 CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,
    IBCMD,IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,
    IBXTRC,IBRDI,IBWRTI,IBRDIA,IBWRTIA,IBSTA%,IBERR%,IBCNT%)
260 BDNAME$ = "TEKDEV1"
270 CALL IBFIND (BDNAME$,TEKDEV1%)
280 IF TEKDEV1% < 0 THEN PRINT "IBFIND ERROR":END
290 INPUT "Press ENTER to set up the DSA.",A$
300 WRT$="init;longform on;fpanel off;debug gpib:on"
310 CALL IBWRT(TEKDEV1%,WRT$)
320 INPUT "Press ENTER to display a waveform from CH 1 of
    the left plug-in amplifier.",A$
330 WRT$="trace1 description:'L1'"
340 CALL IBWRT(TEKDEV1%,WRT$)
350 INPUT "Press ENTER to set up the waveform.",A$
360 WRT$="tbm tim:10e-6;mainpos -10e-6;ch11
    imp:50,sens:0.2,offs:0.4;trm anl:0.5,vol"
370 CALL IBWRT(TEKDEV1%,WRT$)
380 INPUT "Press ENTER to store the waveform.",A$
390 WRT$="store tra1:sto1"
400 CALL IBWRT(TEKDEV1%,WRT$)
410 INPUT "Press ENTER to remove trace 1 from the
    display.",A$
420 WRT$="rem tra1"
430 CALL IBWRT(TEKDEV1%,WRT$)
440 INPUT "Press ENTER to display the stored waveform.",A$
450 WRT$="tra1 desc:'sto1'"
460 CALL IBWRT(TEKDEV1%,WRT$)
```



```
470 INPUT "Press ENTER to display a waveform from CH 2 of
    the left plug-in amplifier.",A$
480 WRT$="trace2 description:'L2'"
490 CALL IBWRT(TEKDEV1%,WRT$)
500 INPUT "Press ENTER to set up the waveform.",A$
510 WRT$="ch12 imp:50,sens:0.2,offs:0.4;mainpos -4e-6"
520 CALL IBWRT(TEKDEV1%,WRT$)
530 PRINT "Press ENTER to clear the screen and display a new
    waveform,"
540 INPUT "showing the difference between the current and
    stored waveforms.",A$
550 WRT$="remove all;tra3 desc:'L2-STO1'"
560 CALL IBWRT(TEKDEV1%,WRT$)
570 INPUT "Press ENTER to take an RMS measurement.",A$
580 WRT$="mslist rms;msys on;rms?"
590 CALL IBWRT(TEKDEV1%,WRT$)
600 RD$=SPACE$(64)
610 CALL IBRD(TEKDEV1%,RD$)
620 PRINT RD$
630 INPUT "Press ENTER to re-enable the front panel.",A$
640 WRT$="fpanel on"
650 CALL IBWRT(TEKDEV1%,WRT$)
660 PRINT "End of example 8."
670 PRINT "Press ENTER to return to the Examples menu,"
680 PRINT "press 'Q' to quit the program without exiting
    BASIC,"
690 INPUT "or press 'S' to quit the program and exit
    BASIC.",A$
700 IF LEFT$(A$,1)="Q" OR LEFT$(A$,1)="q" THEN END
710 IF LEFT$(A$,1)="S" OR LEFT$(A$,1)="s" THEN SYSTEM ELSE
    LOAD "menu.bas",R
```

Status & Event Reporting

The DSA 601 and DSA 602 Digital Signal Analyzer provides a status and event reporting system for the GPIB and RS-232-C interfaces. The status and event system alerts you to significant conditions and events that occur within the DSA.

The status and event system has two principal subsystems:

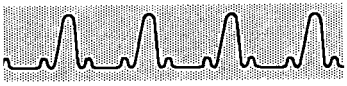
- The status reporting subsystem is based on the service request (SRQ) function defined by IEEE STD 488 for the GPIB interface. It provides a single byte of general status information. For the RS-232-C interface, the STBYTE? query command provides essentially the same function.
- The event reporting subsystem is defined by the Tektronix Codes and Formats Standard using the EVENT? query command. This query provides more detailed information about the specific event that has occurred. The EVENT? response may be reported to either the GPIB or the RS-232-C interface.

A controller always has the option of reading or ignoring the event code(s) associated with a given status byte.

Status Reporting

The status reporting subsystem includes:

- Status Byte for conveying the type of event that has occurred
- RQS command for GPIB asynchronous service requests and status messages
- SRQMASK command for masking event conditions
- STBYTE? query for RS-232-C polled status messages
- RS232 VERBOSE command for RS-232-C asynchronous status messages
- System Status Conditions the categories of events that are reported, such as command errors and internal warnings.



Status Byte Definition

The table below describes the individual bits in the status byte. Bit 8 is the most significant bit of the status byte. DIO is an IEEE STD 488 abbreviation for Data Input Output.

Status Byte Definitions

DIO Bit #	Meaning
1 2 3 4	System status bits. The state of these four bits varies with the type of event that is reported.
5	Busy bit. Asserted only when DSA diagnostics are in progress.
6	Error bit. Asserted when an internal or external error condition generates an event.
7	RQS (request service) bit. Asserted when the DSA requests service from a GPIB controller.
8	Never asserted (bit DIO8 is always 0).

RQS Command

The IEEE STD 488 Service Request function (SRQ) permits a device to asynchronously request service from a GPIB controller whenever the device detects some noteworthy event. A GPIB controller services the request by serial polling each active device on the bus. A device responds to the serial poll by placing an 8-bit status byte on the bus. The controller determines which device asserted SRQ by serially reading the status byte of each device and examining bit 7. If a particular device has requested service, bit 7 of its status byte is set. Otherwise, bit 7 is clear. (Refer to the Binary and Decimal Status Byte Codes table). The RQS command turns on the SRQ function in the DSA.

RQS only affects status and event reporting at the GPIB port. RQS has two major effects:

- It controls bit DIO7 of the status byte. The RQS ON command enables DIO7 assertion. The RQS OFF command disables assertion for all conditions except power-on. At instrument power-on, RQS is on at the GPIB port and off at the RS-232-C port.

- The RQS command also controls whether or not the DSA is permitted to request service from a GPIB controller. The RQS OFF command disables service requests. The RQS ON command enables service requests.

RQS for GPIB service requests—causes the DSA to assert the SRQ signal line whenever a new event occurs and RQS is on. A GPIB controller may then interrogate the DSA **with** an IEEE STD 488 serial poll and obtain a status byte that describes the event that occurred.

When RQS is off, the only new event that will cause the DSA to assert SRQ is power-on. Thus, a GPIB controller will not be informed asynchronously (with SRQ) that an event has occurred. In this situation, a controller may still interrogate the DSA with an IEEE STD 488 serial poll to read the most recent status byte from the serial poll register of the DSA.

RQS for RS-232-C service requests—is always off at the RS-232-C port. There is no SRQ signal line for the RS-232-C interface. No asynchronous messages are sent to the controller. Thus, an RS-232-C controller is required to query (poll) the DSA to determine the latest status condition that has occurred in the DSA.

SRQMASK Command

Regardless of whether RQS is on or off, there may be occasions when you want to disable event reporting for a specific class of system conditions. Use the SRQMASK command to disable (mask off) a specific category of events. The event tables later in this section include the SRQMASK for each event type.

STByte? Query-only Command

The STBYTE? query allows RS-232-C controllers to read the status byte of the most recent event reported to the RS-232-C port.

The response to the STBYTE? query is:

STBYTE <*NR1*>

where <*NR1*> is a decimal number representing a status condition. (Status byte conditions are defined on page 5-6.)

RS232 Verbose Mode

RQS is always off for the RS-232-C interface. Therefore, no new instrument event will cause the DSA to request service.

However, in addition to polling the DSA using the STBYTE? query, the RS-232-C interface includes another means to synchronously report status messages, RS232 VERBOSE mode. This mode is turned on or off by using either the **RS232** pop-up menu in the front panel Utility 2 major menu, or the or the RS232 VERBOSE command.

When VERBOSE is set to ON, each command sent to the DSA always returns an appropriate status message. (For more information on verbose mode, see the discussion on page 2-21.)

System Status Conditions

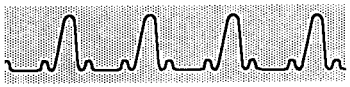
The status byte indicates nine system status conditions. System status conditions are divided into two categories: normal (DIO6 clear) and abnormal (DIO6 set).

There are five normal conditions defined:

- **No Status To Report** reports when there is no event or device dependent status to report.
- **Power On** reports when the DSA has finished its power-on sequence.
- **Operation Complete** tells the controller that a time-consuming task has been completed.
- **User Request** reports when the RQS icon is selected at the front panel.
- **Calibration Due** reports when self-calibration is due.

There are five abnormal conditions defined:

- **Command Error** reports when a message cannot be parsed or lexically analyzed.
- **Execution Error** reports when a message is parsed but cannot be executed.
- **Internal Error** reports if the DSA malfunctions.
- **Execution Warning** reports when the DSA is operating, but you should be aware of possibly inaccurate results.
- **Internal Warning** reports when the DSA detects a problem. The instrument remains operational, but the problem should be corrected.



A list of the binary and decimal codes that correspond to the previously described system status conditions is provided in the following table.

Binary and Decimal Status Byte Codes

	BINARY Status Bits		DECIMAL RQS	
Condition	**8765**	**4321**	**ON**	**OFF**
Normal:				
No Status to Report	0000	0000	0	0
Power On	0R00	0001	65	1
Operation Complete	0R00	0010	66	2
User Request	0R00	0011	67	3
Calibration Due	0R00	0110	70	6
Abnormal:				
Command Error	0R10	0001	97	33
Execution Error	0R10	0010	98	34
Internal Error	0R10	0011	99	35
Execution Warning	0R10	0101	101	37
Internal Warning	0R10	0110	102	38

DIO7, shown as "R," is asserted when specifically enabled with the RQS command (GPIB only). Otherwise, the "R" bit is 0 (zero).

Event Reporting

The second subsystem is event code reporting. Event messages expand the description of the status condition reported by the status byte to more clearly specify the event that has occurred.

GPIB and RS-232-C controllers may read event codes generated by the DSA by using the EVENT? query-only command.

The response to an EVENT? is either:

EVENT <*NR1*>

or

EVENT <*NR1*>,<*qstring*>

where <*NR1*> represents the numerical value of an event code, and <*qstring*> is a quoted string that describes the returned event code.

The response that includes the quoted description string is returned only when the LONGFORM command is set to ON.

Event Code Descriptions

All event codes and event code description strings for all event classes are listed beginning on page 5-9. The event code and event code description is in boldface. Commands that can generate the event code are listed immediately after.

Formatting Symbols—such as %A are combined in some of the description strings in the event code tables. When the event is queried, the formatting symbol(s) are expanded, as described on the next page.

Each formatting symbol begins with a percent sign (%). The symbols indicate that variable information will be substituted when LONGFORM is set to ON.

The formatting symbols and their meanings are shown in the following table.

Formatting Symbols

Symbol	Expand With:
%a	Plug-in channel number or unsigned integer
%A	Argument name
%b	Plug-in compartment indicator: L, C, or R
%B	Plug-in compartment indicator: LEFT, CENTER, or RIGHT
%C	Calibration request string: "Calibration due"
%d	Time base string: "Main" or "Window"
%D	Record length integer
%I	Calibration request string: "Calibration due"
%M	A calibration fault string for the main oscilloscope. If no error occurred, %M is replaced by "Pass"; otherwise %M is replaced by a short descriptive string of what caused the mainframe failure; for example, "Main Fine Holdoff."
%O	Option description string (e.g., "Option 4C - Nonvolatile RAM"
%P	Plug-in compartment fault list. If there are no plug-in unit failures, %P is replaced with "NONE". Otherwise %P will be replaced with a comma-delimited list of plug-in compartments, "LEFT," "CENTER," or "RIGHT," according to which compartments reported failures.
%T	Time, as "X minutes and Y seconds." If X is 0, then "X minutes" is omitted. If Y is 0, then "Y seconds" is omitted.
%W	Calibration request string: "Calibration due"
%?	Event code value

Command Errors

Command errors are reported when a message cannot be parsed or lexically analyzed. Command errors have event codes from 100 to 199. The SRQMASK for command errors is SRQMASK CMDERR. The status byte for a command error returns **97** (decimal) with RQS set to ON, and **33** (decimal) with RQS set to OFF. All command errors are listed on the following pages.

Command Errors

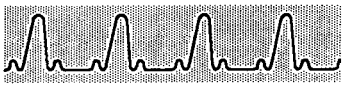
Event Code	Description	Commands that Generate Code	Explanation
108	**Checksum error in binary block transfer**	SET <*bblock*>	Checksum comparison of binary settings failed. Settings are discarded.
109	**Illegal byte count value on a binary block transfer**	SET <*bblock*>	Binary block byte count of binary settings returned to the DSA exceeds maximum size of frontpanel settings.
154	**Invalid number input**		Floating-point value too large or excessively long.
155	**Invalid string input**		String is too long, is not properly terminated, or contains a NULL character.
156	**Symbol not found**		**Oscilloscope** is unable to find the input symbol in its table.
157	**Syntax error**	Any command	Command was typed incorrectly.
		DELTA	<*qstring*> argument specified incorrectly
		RQS	Attempted to turn RQS on at RS-232-C port.
		STBYTE?	Attempted to use STBYTE? query from GPIB port.
		TEST	Set or query command appended to TEST command. TEST command is ignored; all other commands are processed normally.
		TRACE<*ui*>	Syntax error found in TRACE expression (for example, "L1 +", or attempted to create non-acquired trace component (for example, STO <*ui*>, <*NRx*>, or combinations) on WIN1 or WIN2 time base.

Command Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
157 (cont)	**Syntax error**	Trigger Source Expressions	Improper syntax in a trigger source expression; for example, any input sequence that cannot be parsed due to missing arguments, links, or delimiters, or incorrect links with commands. Note that this syntax does not permit RIGHT plug-in channels to be added to (or subtracted from) LEFT or CENTER plug-in channels
160	**Expression too complex**	TRACE <*ui*>	Trace description exceeds 54 characters for either the vertical or horizontal description, or cannot be parsed due to insufficient stack space.
		DELTA	DELTA description cannot be parsed due to insufficient stack space.
161	**Excessive number of points in binary CURVE data input**	Waveform Retrieval and Scaling (Data Transfer)	More binary data points were sent than were specified with the WFMPRE NR.PT link.
162	**Excessive number of points in ASCII CURVE data input**	Waveform Retrieval and Scaling (Data Transfer)	More ASCII data points were sent than were specified with the WFMPRE NR.PT link.
163	**No input terminator seen**		RS-232-C input type-ahead buffer has overflowed. All input is discarded.
164	**Binary block input not allowed with ECHO ON**	CURVE <*bblock*> SET <*bblock*>	Attempted to send binary block data through RS-232-C port with echo on. The data are discarded.

Command Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
167	**Insufficient data to satisfy binary block byte count**	SET?	Binary settings returned to GPIB port prematurely terminated (for example, binary block byte count not satisfied when EOI line is asserted).
168	**Unsupported constant**		
169	**Unsupported function**	TRACE<*ui*>	TRACE expression includes unsupported function.



Execution Errors

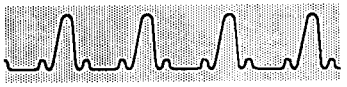
Execution errors are reported when a message is parsed but cannot be executed. Execution errors have event codes from 200 to 299. The SRQMASK for execution errors is SRQMASK EXERR. The status byte for an execution error returns **98** (decimal) with RQS set to ON, and **34** (decimal) with RQS set to OFF. All execution errors are listed on the following pages.

Event Code	Description	Commands that Generate Code	Explanation
203	**I/O buffers full**		Both input and output buffers are full. Output buffer is cleared.
205	**%A out of range - value ignored**	ABSTOUCH	Out-of-range ABSTOUCH argument.
		MCALCONSTANTS	Out-of-range <*ui*> values.
214	**That function is incompatible with %O**	DELTA, FFT	Attempted DELTA or FFT command with Option 3C-Acquisition Memory External Power Supply installed.
215	**Can't undo autoset**	AUTOSET	Attempted AUTOSET UNDO with no previous AUTOSET performed.
216	**Can't spool hardcopy**	COPY	Attempted COPY START when printer spooler is full.
217	**Can't keep scan waveform**	SCANSTOWFM	No current scan waveform exists to keep, or scanning was never started, or the template waveform has changed, or too many traces already exist to creat another.
218	**Can't start scanning**	SCANSTOWFM	There are no stored waveforms, or eight traces are already defined, or Repetitive Trigger or Delta is in effect.
219	**Record length of delta description test wfm cannot be greater than record length of test wfm**	DELTA	Attempted to enter new delta description or attempted to increase record length which conflicts with current delta description.
220	**Connect probe to calibrator and restart operation**	CALPROBE	Attempted to run probe calibration on a channel with no input signal.
221	**Illegal delta description**	DELTA	<*qstring*> argument not specified correctly (such as a non-envelope waveform).
222	**%O needed to support that function**	HSBATT?	Attempted to use an non-existent option.
223	**Illegal base label**	LABEL	Numerals specified as part of base label.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
224	**Function not available in selected plug-in range**	CALPROBE	Attempted probe calibration on an 11A33 plug-in unit with CH<*slot*> <*ui*> PROTECT: ON.
225	**Cannot change label while current acquisition mode is running**	DELTA	Attempted to change stored waveform labels or base label with Act-on-Delta running.
		LABEL	Attempted to change stored waveform labels or base lable with Repetitive Single Trigger running.
226	**Trigger timer not available**	Trigger Source Expressions	Trigger description specifies use of a timer more than once.
227	**Not available with Extended Triggering**	TRMAIN, TRWIN, Trigger Source Expressions	Attempted to set MODE to AUTOLEVEL, SLOPE to MINUS, COUPLING to AC, ACHF, or ACNOISE, or WTMODE to EVHOLDOFF or TIHOLDOFF with extended triggering mode active (i.e., a WHILE, AND, OR, or XOR operator appearing in the Main trigger expression).
228	**Label not found**	CLEAR, DELETE, INPUT, OUTPUT, RECALL, REMOVE, SELECT, STORE, SCANSTOWFM	No matching label found with <*qstring*> syntax used.
229	**No stored waveforms**	LABEL? STO, SCANSTOWFM	No waveforms were stored when LABEL was queried or SCANSTOWFM FROM or SCANSTOWFM TO link was issued.
230	**Can't set front panel calibrator amplitude**	CALIBRATOR	Attempted to set AMPLITUDE when frequency (FREQ) is 1KHz or 1MHz.
231	**Autoset – not functional with this waveform type**	AUTOSET	Attempted to autoset window waveform, which has no "parent" main waveform, when main waveform time base not triggered.

Signature Not Verified
ARTEK MEDIA
Digitally signed by ARTEK MEDIA
DN: cn=ARTEK MEDIA, o=DC Henderson, c=US
Date: 2005.03.07 10:25:54 -06'00'

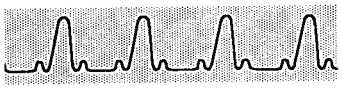


Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
232	**That XY waveform has incompatible components**	TRACE <*ui*>	Attempted to create XY trace with horizontal and vertical components with incompatible scaling modes.
233	**Delayed trace must not be the selected trace**	DLYTRACE	Attempted to specify currently selected trace as PDELAY delayed trace.
234	**Unsupported printer function**	COPY	Format unsupported for currently selected printer.
235	**Duplicate label – label not changed**	LABEL, FPS <*ui*>, TRACE <*ui*>, STO <*ui*>, WFMPRE	Attempted to create a duplicate label.
236	**Illegal color number**	COLOR, HPGL, TEK4692, TEK4696	Out-of-range color index.
237	**No labels defined**	LABEL?	No labels defined for specified links.
238	**Label not defined**	LABEL?	No label is defined for FPS <*ui*>, STO <*ui*>, or TRACE <*ui*> links.
239	**Improper version number**	SET <*bblock*>	Version number of received binary settings block not the same as current firmware version number.
240	**Can't accumulate nonacquired waveform**	TRACE <*ui*>	Attempted to enable point accumulate with non-acquired trace.
241	**Too many acquisitions**	TRACE <*ui*>	Trace definition would cause the DSA to acquire more than 14 total acquisitions.
242	**ENHANCED ACCURACY available after %T**	SELFCAL	Attempted SELFCAL FORCE before 20-minute warmup elapsed.
243	**That function is disabled by a hardware strap**	CCALCONSTANTS, LCALCONSTANTS, RCALCONSTANTS	Attempted to set plug-in calibration constants with hardware strap disabled.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
243 (cont)	**That function is disabled by a hardware strap**	MCALCONSTANTS	Attempted to set DSA calibration constants with hardware strap disabled.
		UID	Attempted to modify serial number with hardware strap disabled.
244	**%B plug-in channel(s) used differently in main and window sources**	Trigger Source Expressions	A channel was "chopped" between the main and window trigger sources, when WTMODE is set to EVHOLDOFF or TIHOLDOFF.
245	**Autoset – only functional with 11K plug-ins**	AUTOSET	Attempted to autoset waveform containing a 7000-series plug-in channel.
246	**Can't sequence settings**	RECALL	Attempted to sequence settings with SETSEQ OFF.
247	**No settings defined**	LABEL?	Attempted to query LABEL? FPS when no settings exist.
		PROBE	Attempted to assign probes to SETSEQ with no settings defined.
		RECALL	Attempted to sequence settings with no settings defined.
		SETSEQ	Attempted to turn SETSEQ on with no settings defined
248	**Misuse of AVG/ENV function**	AVG, ENV	Attempted to turn AVG or ENV on when selected trace is XY, or when selected trace is composed only of stored and scalar components. Or attempted to turn AVG or ENV off when selected waveform's vertical description not enclosed by the AVG or ENV function.
		TRACE<*ui*>	Trace description includes AVG or ENV function with a non-acquired argument, such as AVG (ST01) or ENV(1000).



Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
249	**Illegal use of trace positioning function**	ADJTRACE <*ui*>	Attempted to modify HMAG, HPOSITION, HVPOSITION, HVSIZE, TRSEP, VPOSITION, or VSIZE values when modification is not permitted (for example, when PANZOOM is off).
250	**No traces defined**	ADJTRACE?, LABEL?, TRACE?	Query attempted with no traces displayed.
		LABEL	Attempted to label or delete a label on a trace when no traces are currently displayed.
		AVG, CURSOR, DOT1ABS, DOT2ABS, DOT1REL, DOT2REL, ENV, H1BAR, H2BAR, V1BAR, or V2BAR	Attempted to set or query one of these commands with no traces defined.
		BASELINE, DAINT, DISTAL, DLYTRACE, LMZONE, MESIAL, MSLOPE, MTIME,PROXIMAL, REFLEVEL, REFTRACE, RMZONE, SNRATIO, TOPLINE	Attempted to set or query one of these measurement commands with no selected trace.
251	**Illegal trace number**	ADJTRACE, CLEAR, CURSOR, DELTA, LABEL, OUTPUT, REMOVE, SELECT, STORE, TRACE	Out-of-range <*ui*> argument with one of these commands.
252	**Illegal stored settings number**	DELETE, LABEL, RECALL, STORE	Out-of-range <*ui*> argument with one of these commands.

Execution Errors (Cont.)

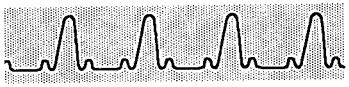
Event Code	Description	Commands that Generate Code	Explanation
255	**Out of memory**	DELTA	Insufficient memory to create DELTA DESCRIPTION: <*qstring*>.
		DIGITIZER	Insufficient memory to save repetitive trigger on delta waveform.
		STORE	Insufficient memory to store a trace or insufficient NVRAM to store settings.
		SCANSTOWFM	KEEP link or MODE:SCAN link sent; insufficient memory to create another displayed trace.
		TRACE	Insufficient memory to create a new trace.
		Waveform Retrieval and Scaling (Data Transfer)	INPUT command references nonexistent stored waveform, insufficient memory to create stored waveform record.
		WFMPRE	Insufficient memory to create stored waveform record for preamble.
257	**Illegal stored waveform number**	DELETE, DELTA, INPUT, LABEL, OUTPUT, STORE, TRACE	Out-of-range STO <*ui*> argument for one of these commands.
		SCANSTOWFM	Argument of FROM or TO link not valid stored waveform.
263	**Illegal channel number**	CH <*slot*> <*ui*>	Attempted to set query parameters of plug-in channel that does not exist.
		TRACE <*ui*>	TRACE expression references illegal plug-in channel.
		Trigger Source Expressions	Trigger Source Expression references nonexistent 11000-series channel number.
264	**No further XY waveforms may be defined**	TRACE<*ui*>	Attempted to define more than the maximum permissible number of XY traces.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
265	**Illegal DATE/TIME**	DATE	Illegal date value or syntax specified.
		TIME	Illegal time value or syntax specified.
		WFMPRE	Invalid date or time string entered. The date or time is set to the current clock value.
266	**DEF expansion overflow**	DEF	Expansion string overflowed internal expansion buffer.
267	**Illegal DEF string**	DEF	Illegal logical name specified.
268	**Illegal DEF recursion**	DEF	Unacceptable DEF recursion detected. Recursive logical names are acceptable only when recursion occurs to the right of an unquoted semicolon.
269	**No such trace**	ADJTRACE, CLEAR, CURVE, DELTA, LABEL, REMOVE, SELECT, STORE, TRACE, WAVFRM?, WFMPRE	Referenced, or attempted to set or query parameters of a nonexistent trace using one of these commands.
270	**No such stored waveform**	CURVE?	CURVE? query attempted, OUTPUT references nonexistent stored waveform.
		DELETE	Attempted to delete nonexistent stored waveform.
		DELTA	Attempted to reference a nonexistent stored waveform.
		LABEL	Attempted to label or query for a label of a nonexistent stored waveform.
		TRACE <*ui*>	TRACE expression referenced legal but undefined stored waveform.
		WAVFRM?	WAVFRM? query attempted, OUTPUT referenced nonexistent stored waveform.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
270 (cont)	**No such stored waveform**	WFMPRE	WFMPRE? query attempted, OUTPUT referenced nonexistent stored waveform.
271	**No such DEF**	UNDEF	UNDEF argument not defined in current list of logical names.
272	**That function is not supported by this plug-in**	CH <*slot*> <*ui*>	Attempted to set or query AMPOFFSET, MNSCOUPLING, MNSOFFSET, MNSPROBE, PLSCOUPLING, PLSOFFSET, PLSPROBE, PROTECT, or VCOFFSET links of nondifferential amplifier, or attempted to set or query COUPLING or PROBE linked of differential amplifier.
			Attempted to set or query BWHI or BWLO parameters of plug-in unit that does not support the high/low bandwidth limit function, or attempted to set or query BW parameter of plug-in unit that supports high/low bandwidth limits.
		Trigger Source Expressions	Attempted to invert the trigger channel for an 11A71 plug-in unit trigger channel.
273	**No such FPS**	DELETE	Attempted to delete undefined stored settings number.
		LABEL	Attempted to label or query undefined stored settings number.
		RECALL	Attempted to recall undefined stored settings number. In this context, "undefined" refers to previously deleted settings or settings that have never been initialized.
274	**No appropriate 11K plug-ins loaded**	CH?	CH? query attempted, DSA has no plug-in units that support 11000-series generic plug-in unit interface.
275	**%B slot not loaded with appropriate 11K plug-in**	CCALCONSTANTS, LCALCONSTANTS, RCALCONSTANTS	Attempted to set or query calibration constants of plug-in compartment not loaded with 11000-series plug-in unit.



Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
275 (cont)	**%B slot not loaded with appropriate 11K plug-in**	CH <*slot*> <*ui*>	Attempted to set or query parameters of plug-in compartment not loaded with 11000-series generic plug-in unit.
		TRACE <*ui*>	TRACE expression references 7000-series or missing plug-in unit as 11000-series signal.
		Trigger Source Expressions	Trigger Source Expression references plug-in compartment not loaded with 11000-series generic plug-in unit as 11000-series trigger channel.
276	**%B slot not loaded with 7K plug-in amplifier**	TRACE <*ui*>	TRACE expression references 11000-series plug-in unit as 7000-series signal.
		Trigger Source Expressions	Attempted to reference 11000-series plug-in unit as 7000-series channel; for example, if the left compartment is loaded with an 11000-series plug-in unit, any attempted to refer to the left compartment as an "L" trigger channel returns this event code.
277	**Misuse of 7K plug-in**	Trigger Source Expressions	Same 7000-series channel used more than once in trigger expression, or attempted to invert 7000-series trigger channel in expression (for example, TRM SOU:"-C" or TRW SOU:"L-C").
278	**Plug-in channel used more than once in trigger source**	Trigger Source Expressions	Same 11000-series channel used more than once in trigger expression.
279	**Line trigger not available for window trigger source**	Trigger Source Expressions	Attempted to set source of window time base to line.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
281	**Can't delete active stored waveform**	DELETE	Attempted to delete stored waveform that is a component of a combined active trace.
		WFMPRE	Returning WFMPRE data would cause deletion of a stored waveform that is not the sole component of a waveform description of a displayed trace. The WFMPRE data are discarded.
282	**Can't store trace**	STORE	Attempted to store XY trace, or attempted to copy a trace over an existing stored waveform when the two waveforms do not have equal record lengths.
283	**Can't clear nonacquired waveform**	CLEAR	Attempted to clear trace that has only stored waveform components (for example, TRACE1 DESCRIPTION:"STO3").
284	**Requested coupling for channel %a not available on %B plug-in**	CH <*slot*> <*ui*>	Attempted to set coupling to value not supported by plug-in unit, or attempted to set plug-in channel's coupling to value not allowed because a Level 2 TekProbe is connected to that channel, or attempted to set coupling to value that would increase overload on input channel.
285	**Requested input impedance for channel %a not available on %B plug-in**	CH <*slot*> <*ui*>	Attempted to set a plug-in channel to impedance value not allowed because Level 2 TekProbe is connected to that channel, or attempted to set impedance to value that would increase overload on input channel.
286	**Too many measurements specified**	MSLIST	More than six measurements specified.
287	**Hardcopy absent or off line**	COPY	CENTRONICS port specified as COPY output port, printer not connected to port or currently connected printer is offline.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
288	**Inappropriate trigger level units**	TRMAIN	Improper ANLEVEL units specified.
		TRWIN	Improper NLEVEL units specified.
289	**Split cursors not permitted on XY trace**	CURSOR	Attempted to SPLIT cursors across XY trace.
290	**Current reference measurement failed**	REFSET	CURRENT reference cannot be computed due to one of the following conditions: ▪ No waveforms are defined (regardless of measurement). ▪ Selected waveform is XY or point accumulate trace (regardless of measurement). ▪ Reference measurement specified as DUTY, FREQ, or PERIOD; no period can be found within specified measurement zone. ▪ Reference measurement specified is MEAN, RMS, YTENERGY, YTMNS_AREA, or YTPLS_AREA while DAINT is set to SINGLE; no period can be found within specified measurement zone. ▪ Reference measurement specified is CROSS and REFLEVEL does not fall between computed maximum and minimum of specified measurement zone. ▪ Reference measurement specified is RISETIME and measurement system cannot compute valid proximal and distal time within specified measurement zone.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
290 (cont)	**Current reference measurement failed**	REFSET	▪ Reference measurement specified is FALLTIME and measurement system cannot compute valid distal and proximal time within specified measurement zone. ▪ Reference measurement specified is WIDTH and two mesial crossings of opposite slope cannot be found within specified measurement zone. ▪ Reference measurement specified is GAIN, PHASE, or PDELAY and only one trace is defined.
291	**TEXT not permitted when acquired XY trace is active**	TEXT	Attempted to place text on display when acquired XY trace is active.
292	**%B slot not loaded with 11K plug-in**	UID	Attempted to set or query serial number of plug-in compartment not loaded with 11000-series plug-in unit.
294	**Dual graticules not permitted with XY trace**	DISPLAY	Attempted to change screen from single to dual graticules when XY trace is displayed.
295	**Record length too long for Point Accumulate waveform**	SELECT	Attempted to select point accumulate waveform whose time base length is greater than 2048 points. The point accumulate waveform becomes the selected trace but is displayed in normal Yt format.
		TBMAIN/TBWIN	Attempted is made to increase time base length of point accumulate waveform to more than 2048 points.
		TRACE <*ui*>	Attempted to enable point accumulate with time base record length greater than 2048 points.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
296	**Point accumulate and XY waveforms are mutually exclusive**	TRACE <*ui*>	Attempted to simultaneously define point accumulate and XY trace on the same graticule.
297	**Panzoom may not be enabled**	ADJTRACE <*ui*>	Attempted to enable PANZOOM for XY trace.
298	**Panzoom may not be disabled**	ADJTRACE <*ui*>	Attempted to disable PANZOOM for stored or scalar trace, or for FFT magnitude phase traces.
299	**CONDACQ function not available**	CONDACQ	AVG or ENV conditional acquisition specified, but no traces include AVG or ENV function in trace descriptions.
			CONDACQ set to BOTH, but the following condition does not exist: ▪ At least one waveform description includes the AVG function and at least one other waveform description includes the ENV function. ▪ One waveform includes both AVG and ENV in its description.
			Conditional acquisition of any type except CONTINUOUS specified, with no traces defined.
			DELTA conditional acquisition specified, but no valid delta description exists.

Internal Errors

Internal errors are reported if the DSA malfunctions. Internal errors have event codes from 300 to 399. The SRQMASK for internal errors is SRQMASK INERR. The status byte for internal errors returns **99** (decimal) with RQS set to ON, and **35** (decimal) with RQS set to OFF. All internal errors are described on the following pages.

Internal Errors

Event Code	Description	Commands that Generate Code	Explanation
308	**Bad level 2 probe checksum on channel %b%a**	CH<*slot*> <*ui*>	Level 2 TekProbe improperly connected to input channel, or Level 2 TekProbe properly connected, but malfunctioned and needs repair.
327	**DIG probe compensation failed**	SELFCAL	Digitizer internal error.
328	**DIG plug-in ENHANCED ACCURACY failed**	SELFCAL	Digitizer plug-in calibration failed. Internal digitizer error.
329	**Deskew failed: %c**	SELFCAL	Digitizer internal error.
330	**ENHANCED ACCURACY failed. Mainframe: %M Plug-in: %P**	SELFCAL	ENHANCED ACCURACY initiated, but failed.
331	**Probe calibration failed: %c**	CALPROBE	Probe calibration initiated, but subsequently failed.
332	**Partial ENHANCED ACCURACY failed. Plug-in: %P**	Power on	Automatic calibration of new plug-in configuration failed.
394	**Test completed and failed**	TEST	Self-tests diagnostics or extended diagnostics completed and failed.
395	**General DIG failure detected (code = %a)**		Digitizer detected an internal error.

Internal Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
396	**%B plug-in communication failure**	CH <*slot*> <*ui*>	Oscilloscope detects that communication is no longer possible with a particular plug-in unit. The DSA may continue to operate, depending on the type of message that was in progress when communication failure occurred. Any of the following problems may exist: failed hardware, a software bug, or a plug-in unit was removed after the DSA was powered up, and communication was then attempted with the empty plug-in compartment.
397	**Internal DAC overflow on channel %a of %B plug-in**	CH <*slot*> <*ui*>	A plug-in unit detects that the requested setting overflowed an internal DAC. The plug-in unit sets the DAC to the limit nearest the requested setting. This event code usually indicates failed hardware.
398	**Invalid DIG table ID detected**		Digitizer detected an invalid table ID.
399	**Invalid DIG field ID detected**		Digitizer detected an invalid field ID.

System Events

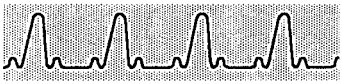
System events are normal conditions of the system and are listed on the following pages. System events have event codes from 400 to 499. The SRQMASK for each event is included in the table.

Note: Event 400 (system function normal) and event 401 (power on) cannot be masked with SRQMASK.

Event Code	Description	SRQMASK	Status Bytes		Commands that Generate Code
400	System function normal	-none-	0	0	
401	Power on	-none-	65	1	
403	Front panel RQS icon selected	USER	67	3	
450	Conditional acquire complete	OPCMPL	66	2	DIGITIZER
451	Abstouch	ABSTOUCH	67	3	ABSTOUCH
457	Probe %a ID button pressed on %B plug-in	IDPROBE	67	3	
458	Hardcopy aborted	OPCMPL	66	2	COPY
460	Test completed and passed	OPCMPL	66	2	TEST
461	ENHANCED ACCURACY completed and passed	OPCMPL	66	2	SELFCAL
462	Hardcopy complete	OPCMPL	66	2	COPY
463	Measurements complete	OPCMPL	66	2	
464	Autoset complete	OPCMPL	66	2	AUTOSET
465	Warmup complete - %C	CALDUE	70	6	
466	New configuration - partial ENHANCED ACCURACY occurring	CALDUE	70	6	
467	Warmup complete with new configuration - %C	CALDUE	70	6	
468	Warmup complete with new configuration - automatic ENHANCED ACCURACY occurring	CALDUE	70	6	
469	Temperature change - automatic ENHANCED ACCURACY occuring	CALDUE	70	6	
470	Temperature change - %C	CALDUE	70	6	
471	Warmup complete - ENHANCED ACCURACY in effect	CALDUE	70	6	
472	Warmup complete - automatic ENHANCED ACCURACY occurring	CALDUE	70	6	

System Events (Cont.)

Event Code	Description	SRQMASK	Status Bytes		Commands that Generate Code
473	Front panel recall complete	OPCMPL	66	2	RECALL
474	INIT complete	OPCMPL	66	2	INIT
475	Probe calibration completed and passed	OPCMPL	66	2	CALPROBE
476	Temperature change - %I	OPCMPL	66	2	
477	Warmup complete with new configuration - %W	OPCMPL	66	2	
478	Warmup complete - ENHANCED ACCURACY in effect. Compensate probe to use max Real Time sample rate	OPCMPL	66	2	
479	Partial ENHANCED ACCURACY completed and passed	OPCMPL	66	2	



Execution Warnings

Execution warnings are reported when the DSA is operating, but may produce inaccurate results. Execution warnings have event codes from 500 to 599. The SRQMASK for execution warnings is SRQMASK EXWARN. The status byte returns **101** (decimal) with RQS set to ON, and **37** (decimal) with RQS set to OFF. All execution warnings are listed on the following pages.

Event Code	Description	Commands that Generate Code	Explanation
550	**%A out of range – limit set**	ADJTRACE <*ui*>	HMAG, HPOSITION, HVSIZE, HVPOSITION, TRSEP, VPOSITION, or VSIZE link argument out of range.
		BASELINE	Out-of-range BASELINE argument.
		CALIBRATOR	Out-of-range AMPLITUDE or FREQ link argument
		CH <*slot*> <*ui*>	AMPOFFSET, MNSOFFSET, OFFSET, PLSOFFSET, SENSITIVITY, or VCOFFSET link argument out of range.
		COLOR <*ui*>	HUE, LIGHTNESS, or SATURATION link argument out of range.
		CONDACQ	FILL link argument out of range.
		DELTA	CONSECPTS or TOTALPTS link argument out of range.
		DISPLAY	INTENSITY link argument out of range.
		DISTAL	Out-of-range DISTAL argument.
		DLYTRACE	Out-of-range DLYTRACE argument.
		DOT1ABS, DOT2ABS	PCTG, XDIV, or XCOORD link argument would position dot cursors off waveform record of selected trace.
		DOT1REL, DOT2REL	PCTG, XDIV, or XCOORD link argument would position dot cursors outside limits specified for corresponding DOT1ABS/DOT2ABS links.
		H1BAR, H2BAR	Out-of-range YCOORD or YDIV link argument.
		HPGL	Out-of-range plotter pen assignment.
		LABABS	Out-of-range <*NRx*> value.
		LABEL	Link argument would position label outside limits specified for LABABS command.

Execution Warnings (Cont.)

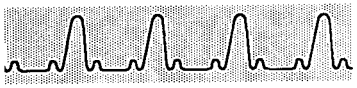
Event Code	Description	Commands that Generate Code	Explanation
550 (cont)	**%A out of range – limit set**	LMZONE	Out-of-range LMZONE argument.
		MAINPOS, WIN1POS, WIN2POS	Out-of-range MAINPOS, WIN1POS, or WIN2POS argument. The valid range of WIN1POS and WIN2POS depend upon the current value of MAINPOS. Thus, changing the value of MAINPOS can cause the current value of WIN1POS or WIN2POS to be out of range. In this case, WIN1POS or WIN2POS are set to the closest legal value.
		MESIAL	Out-of-range MESIAL argument.
		NAVG	Out-of-range NAVG value.
		NENV	Out-of-range NENV value.
		NREPTRIG	Out-of-range NREPTRIG argument.
		PROXIMAL	Out-of-range PROXIMAL argument.
		REFLEVEL	Out-of-range REFLEVEL argument.
		RMZONE	Out-of-range RMZONE argument
		RS232	Out-of-range BAUD, DELAY, or STOPBITS link argument.
		SCANSTOWFM	Out-of-range FROM, RATE, or TO link argument.
		SNRATIO	Out-of-range SNRATIO argument.
		TBMAIN, TBWIN	Out-of-range LENGTH or TIME link argument.
		TEK4692	Out-of-range RGB value.
		TEK4696	Out-of-range color inkjet selection.
		TEXT	X or Y link argument out of range.
		TOPLINE	Out-of-range TOPLINE argument.

Execution Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
550 (cont)	**%A out of range - limit set**	TRMAIN	ALEVEL, ANLEVEL, or TIHOLDOFF link argument out of range.
		TRWIN	ALEVEL, EVHOLDOFF, NLEVEL, or TIHOLDOFF link argument out of range.
		TTAVERAGE	TTAVERAGE argument out of range.
		V1BAR, V2BAR	XCOORD or XDIV link argument out of range.
		WFMPRE	NR.PT, XINCR, XZERO, YMULT, or YZERO link argument out of range.
551	**Insufficient data to satisfy binary block byte count**	Waveform Retrieval and Scaling (Data Transfer)	Binary waveform data sent to GPIB port prematurely terminated (for example, binary block byte count not satisfied when EOI line asserted). The waveform is filled out with NULL points.
552	**Checksum error in binary block transfer**	Waveform Retrieval and Scaling (Data Transfer)	Checksum of received binary waveform data does not match checksum of original binary block. The waveform data is retained, regardless of the outcome of the test. **Note:** *If the binary data was created with a NULL checksum, the checksum test is almost certain to fail. Since the returned data is not discarded, this failure is not important.*
553	**Window trigger source set equal to main trigger source**	Trigger Source Expressions	WTMODE changed from MAIN to TIHOLDOFF or EVHOLDOFF, and window source is incompatible with main source. The window trigger source is set equal to main source.

Execution Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
554	**Autoset – no signal detected**	AUTOSET	AUTOSET initiated with no traces defined and no signal source can be found (for example, no plug-in units are loaded in the plug-in compartments), or the signal being autoset is DC (it has no AC component).
555	**Binary curve odd data byte discarded**	Waveform Retrieval and Scaling (Data Transfer)	An odd number of data bytes was sent to the DSA.
556	**No acquisitions active – digitizer remains stopped**	DIGITIZER	Attempted to start digitizer when no traces are defined, or when no defined traces contain "active" components (as opposed to scalar and stored components).
557	**Hardcopy aborted**	COPY	COPY operation aborted.
558	**Nothing to abort**	COPY	COPY ABORT attempted, nothing to abort.
559	**XY PT.FMT not permitted, PT.FMT not changed**	WFMPRE	WFMPRE PT.FMT set to XY. The data format is not changed.
560	**AUTOSET – vertical search failed**	AUTOSET	AUTOSET failure for any reason other than those that generate event code 554.
561	**Base label index greater than 999, waveform not stored**	DIGITIZER	Digitizer reached maximum base label index during repetitive single trigger; digitizer stopped.
562	**AUTOSET – trigger search failed**	AUTOSET	AUTOSET could not find a valid trigger signal.
563	**AUTOSET – horizontal search failed**	AUTOSET	Horizontal autoset algorithm cannot correctly calculate period of selected trace.
564	**AUTOSET – ac signal too large**	AUTOSET	Vertical AUTOSET algorithm detects signal whose AC component is too large for least-sensitive gain setting of plug-in channel.



Execution Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
565	**AUTOSET - dc signal too large**	AUTOSET	Vertical AUTOSET algorithm detects signal whose DC component is larger than offset range of least-sensitive gain setting of plug-in channel.
566	**Interleave Enabled - Press ENHANCED ACCURACY then compensate probe to use the max Real Time sample rate**	INTERLEAVE	ENHANCED ACCURACY must be in effect and probes must be calibrated to get 1 Gsample/sec in DSA 601 or 2 Gsample/sec in DSA 602.
567	**Trigger timer2 value modified due to change to timer1**	TRMAIN, TRWIN	Change in trigger timer1 caused change in trigger timer2.
568	**Trigger mode changed to Normal**	CONDACQ	The trigger mode is set to Normal when CONDACQ TYPE is set to SEQUENCE, SINGLE, or DELTA.
569	**Argument out of range. Limit set. Valid smoothing range is: 3 - 999**	TRACE<*ui*>	Trace description contains a SMOOTH function with an out of range argument.
570	**Argument out of range. Limit set. Valid dejitter range is: 0 - 9**	TRACE<*ui*>	Trace description contains a DEJITTER function with an out of range argument.
571	**Interleave Enabled - Compensate probe to use the maximum Real Time sample rate**	INTERLEAVE	When INTERLEAVE is enabled, probe calibration must be run to achieve maximum accuracy.

Execution Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
572	%d record length changed to %D	CONDACQ	When CONDACQ TYPE is DELTA, attempted to increase record length which conflicts with current delta description.
573	FFT record length must be a power of 2	TBMAIN, TBWIN	The record length of an FFT function must be a power of 2.
574	Delta description no longer valid	COMPARE, DELETE, REMOVE	

Internal Warnings

Internal warnings are reported when a problem has been detected. The DSA remains operational, but the problem should be corrected. Internal warnings have event codes from 600 to 699. The SRQMASK for internal warnings is SRQMASK INWARN. The status byte for internal warnings returns **102** (decimal) with RQS set to ON, and **38** (decimal) with RQS set to OFF. All internal warnings are listed on the following pages.

Event Code	Description	Commands that Generate Code	Explanation
651	**Input channel %a overload on %B plug-in**	CH <*slot*> <*ui*>	Input signal overloads the low-impedance terminal on resistor of a plug-in unit. The plug-in unit changes impedance to protect against this condition and returns the event code.
652	**Input channel %a overdrive on %B plug-in**	CH <*slot*> <*ui*>	Input signal of plug-in unit is overdriven in a way that might distort the displayed signal.
653	**RS-232 input parity error**		
654	**RS-232 input framing error**		
655	**RS-232 input buffer overrun**		
656	**Internal table search failed**		
657	**Probable nonvolatile RAM battery failure. Nonvolatile RAM completely reset**		

Internal Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
659	Cannot report unknown error code (%?)		
660	Digitizer stopped – timebase settings exceeded available acquisition memory		
665	Teksecure Erase Memory Status: Erased; Instrument ID, on-time, and number of power-ups retained		

Appendix A: Improving System Performance

Optimum system performance means acquiring accurate data with the fastest system throughput. This appendix discusses the components of system performance and suggests techniques to improve each.

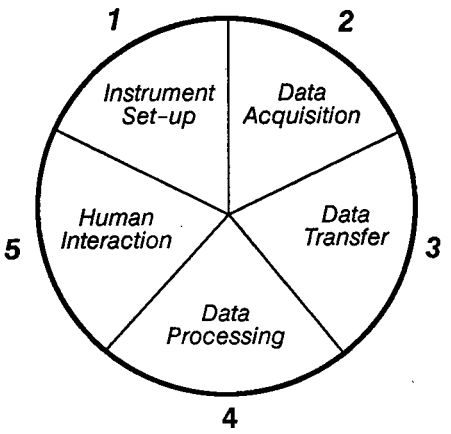
First, you must be familiar with your instrument controller, measurement instruments, data recorders, and with your chosen software (operating system, programming language, device drivers, etc.). When you know the capabilities of your system, you are better prepared to write efficient application programs.

Then you must decide which interface (GPIB or RS-232-C) best suits your application needs.

A good way to develop a thorough understanding of your system instruments is to study their manuals. In particular, learn about the command vocabulary and data formats (for example, ASCII or binary), for each instrument and learn how each device buffers data and executes commands. This gives you information about which hardware configurations and program algorithms will be most efficient for your application.

Components of System Performance

Five major components affect the overall system performance, as summarized in the following illustration. The sum of these components is the total time required to execute your application.



System Performance Components

The contribution of each component to the total execution time varies, based on your specific system configuration.

For example, a data logging system generally requires little time to set up and doesn't require operator intervention. However, significant time is spent acquiring and transferring data. In contrast, a production test system may spend less time acquiring data, but more time processing data and interacting with the operator. Each situation requires a different focus for optimizing system performance.

The best way to determine the time that each component contributes to system performance is to measure it. You can use a real time clock in your controller to do this.

For example, to measure the time it takes to execute a PP? (peak-to-peak amplitude) measurement query, turn on your controller real-time clock before the command, then read the elapsed time immediately after reading the PP? response. Repeating this measurement a few times under varying system configurations will produce typical values you can use to judge the impact of each component on system performance.

Instrument Setup Time

Instrument setup time can be divided into two parts: the time required to decode and execute a setting command, and the time required for new settings to stabilize.

The time it takes to decode and execute a single DSA command is usually short, but if a command initiates a complex or lengthy operation, it can increase the setup time.

For instance, some commands require the DSA to check whether any settings associated with the command function have changed prior to the command. If any associated settings have changed, the DSA must load the new settings into its hardware.

The second part of the setup time is the time it takes the DSA to settle to the specified setting. For example, when vertical size is set automatically, the DSA takes a reading of the input voltage, tests for under- or over-voltage conditions, steps the vertical scale range up or down, and takes another reading. Several readings might have to be made until the correct range is determined. The process stops once the reading is within the new vertical scale range. Thus, a single change in test conditions can cause a significant change in setup time.

Optimizing setup time—requires reducing the number of setting changes or reducing the time required for the DSA to execute the setting changes.

Here are some suggestions to optimize setup time:

- Group tests that use common settings.

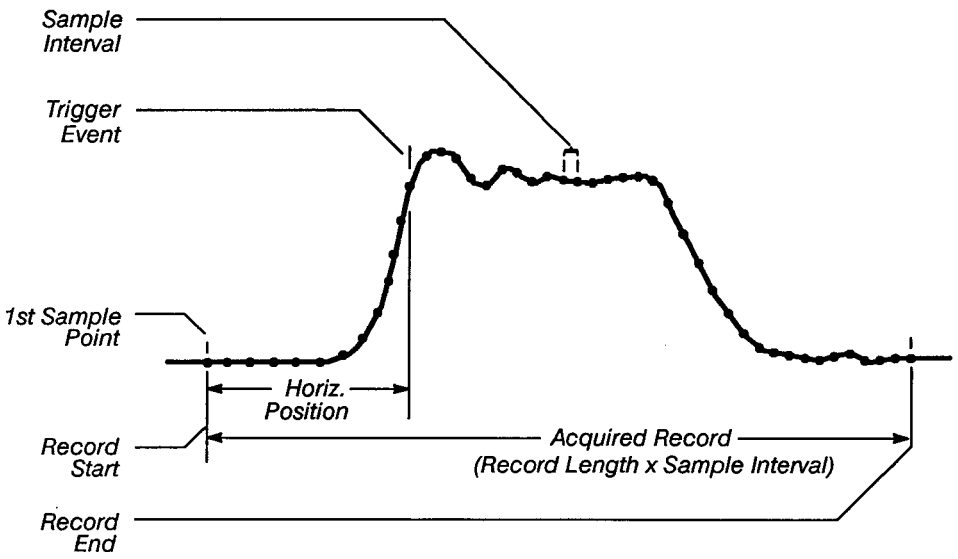
- Set your ranges explicitly. Generally, autoset takes more time.
- First set up instruments that require more settling time. While setting, you can be setting up other devices.
- Use the store setting features. Reconstructing a setting takes more time.
- Use low byte-count and less complex commands. For example, use the LONGFORM OFF command for abbreviated responses to queries. This can significantly reduce the byte count for data transfers.

Data Acquisition Time

The second component of system performance is the time required to acquire a full record of the input source (the selected waveform). This is the data acquisition time.

The DSA has two acquisition modes: real time and equivalent time. In real time acquisition, all waveform samples are taken at one trigger event. Equivalent time acquisition takes several consecutive trigger events to fill a waveform record.

In real time mode, the data acquisition time corresponds to the duration of the record (the record length times the sample interval). In equivalent time mode, several factors affect the data acquisition time: frequency of trigger events, horizontal size, and the waveform record length.



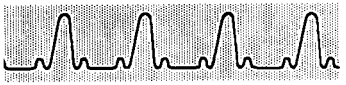
Components of Data Acquisition Time

Optimizing data acquisition time—requires careful attention in setting up the acquisition.

Here are some suggestions to optimize data acquisition time:

- Faster digitizing can be achieved by increasing the repetition rate of the input signal (if possible), or by changing the time base setting. The fastest digitizing rate occurs at the maximum real time sample rate. At slower sample rate, the DSA takes longer to acquire a waveform record. Faster effective sample rates use equivalent time random sampling, which is slower than real time acquisition.

- Use an operation-complete SRQ interrupt instead of waiting for the acquisition to finish. You can continue processing while the acquisition completes.



Data Transfer Time

The third component of system performance is the time it takes to transfer data from one instrument to another. The data transfer time depends on two factors: the number of bytes being transferred and the time it takes to transfer each byte.

The number of bytes transferred depends on the size of the message (number of characters) and the data format (for example, ASCII or binary). For GPIB transfers, the transfer rate depends on the speed of the slowest addressed device on the bus. For RS-232-C transfers, the data transfer rate depends on the baud rate setting of the DSA and controller.

Understanding the processing of GPIB and RS-232-C I/O statements is the key to estimating data transfer times.

GPIB I/O execution time – consists of five parts:

- Addressing sequence
- Unaddressing sequence
- Statement overhead
- Buffer overhead
- Data overhead

The addressing and unaddressing sequences are composed of GPIB interface messages that make the DSA talk or listen to the controller. The time required depends on the data handshake rate of the slowest device connected to the bus.

Statement overhead is the time required to examine the I/O statement for content and syntax (parsing). For the controller, this includes evaluating the statement's I/O function(s) and other expressions, and the statement clauses (DSA commands).

Buffer overhead is the time it takes to fill or empty an I/O memory register with the I/O statement. This depends on the the amount of data (how many characters), and the type of data (string or numeric, ASCII or binary).

Data overhead is the time it takes to transfer data over the interface bus. Again, the time depends on the data transfer rate of the slowest device involved in the transfer, and on the amount and type of data transferred (for example, numeric arrays are a little faster than an equivalent number of scalar variables). This includes the spaces and formatting characters for each message. The total data transmission time is the number of bytes being transferred divided by the data transfer rate (in bytes/second).

RS-232-C I/O execution time—consists of five parts, similar to the GPIB:

- Statement overhead
- Buffer overhead
- Start message
- Data overhead
- Stop message

The RS-232-C statement and buffer overheads consist of the same elements as in GPIB I/O.

The start and stop message time consists of the time required to send one or two bits (depending on the configuration of the RS-232-C interface) before and after each byte of the message in order to synchronize the transmission.

The RS-232-C data overhead time is determined by the baud rate setting of the RS-232-C port on each device.

Since data are sent serially over the RS-232-C interface, additional time is required to convert information from serial-to-parallel, for input data, and from parallel-to-serial for output data. Thus, throughput for an RS-232-C message tends to be slower than throughput for the same GPIB message.

Optimizing data transfer time – involves two major areas. The first is optimizing the system configuration, and the second is the program that controls the transfer.

These suggestions will help you optimize the system configuration:

- Choose instruments that have an optimum transfer rate as near as possible to the bus capacity.
- If your controller has more than one GPIB port, use frequently interacting devices on one bus, or put faster devices together on one bus.
- Use direct-memory access (DMA) transfers whenever possible and keep the faster instruments on this bus.
- Be sure to unaddress slow devices when they are not required in the transfer.
- If you have two ports, put a device under test (DUT) on one bus, and the test equipment on the other bus. Then, if the DUT has an error or malfunction, it won't affect the test equipment.

Follow these suggestions to optimize transfer program parameters:

- Choose the most efficient I/O statements that your controller provides. In most cases high-level commands are fastest, except where long strings are encountered. Then use low-level transfer commands (if provided).
- Minimize bus traffic by reducing the number of bytes being sent. You can do this by abbreviating command names, deleting unnecessary spaces, and omitting unnecessary zeros.

- Minimize buffer overhead. This can be done by defining buffer size (usually possible for most controllers) to accommodate the entire data transfer. You may also store the data within a string variable; string variables store data directly from the I/O buffer and reduce overhead time.

- Use binary block data transfers if possible. Binary data is a little more complicated to handle than ASCII data, but binary transfers tend to be much faster because they involve fewer bytes than an equivalent ASCII transfer.

Data Processing Time

The fourth component of system performance is the time required to manipulate the acquired data for a desired result.

The data processing time is composed of the time it takes the DSA to manipulate the data, plus the time required by the controller to further process the data. The DSA can deliver raw, semi-processed data, or completely processed data, depending on the requirements of the application. The processing speed of the DSA depends on the type or complexity of the operation performed.

Optimizing data processing time – involves using faster algorithms and distributed processing.

These suggestions will help you optimize data processing time:

- Evaluate your choice of algorithms to ensure that the most efficient operations are used for your application and system configuration.

- Use implied array operations instead of for/next loops in your controller programs. This allows numeric operations to be performed much faster. The implied array operation creates temporary arrays to perform the implicit operation (for example, add a scalar to the array) rather than an element-by-element operation.

- Carefully select the data type for your controller programs. Try to group integer, short floating-point, and long floating-point operations. It takese less time to process each as a group, rather than to do mixed data type operations that require conversion from one format to another.

- Evaluate your measurement needs to identify the most effective device for each data processing task. For example, would the DSA best perform a given function on a waveform, or would your controller perform that function more quickly?

Human Interaction Time

The fifth major component of system performance is determined by operator intervention required to enter test parameters or to make adjustments to a device under test (DUT).

This component can easily become the largest part of tht total operating time for a system. Direct measurement of this component is the best way to determine its effect on system performance.

Optimizing human interaction time – can be difficult. The best advice is to avoid the need for human interaction with the system as much as possible.

Follow these suggestions to optimize human interaction time:

- Use programmable interfaces and switches to route signal connections wherever possible. These include programmable relay scanners, multi-function interfaces, and signal multiplexers.

- Keep the user interface simple. The DSA is designed especially for this purpose. User menus are quick and easy to use, so you can make changes quickly.

Appendix B: Reserved Words

Reserved words represent the entire set of predefined command words for the DSA, including headers, links, and arguments.

In this section, reserved words appear in mixed case, with the required minimum substring in uppercase.

A

ABBwfmpre
ABOrt
ABORTIng
ABStouch
AC
ACCumulate
ACHf
ACLf
ACNoise
ACState
ADJtrace
ALEvel
ALL
ALL_Wavfrm
ALTinkjet
ALWays
AMPLitude
AMPoffset
AMPS
ANBlevel
ANLevel
ARMed
ASCii
AUTO
AUTOLevel
AUTOSet
AVG

B

BASELAbel
BASeline
BAUd
BINary
BINHex
BIT/nr
BITMap
BLAckman
BLHarris
BN.fmt
BOTh
BW
BWHi
BWLo
BYPassed
BYT/nr
BYT.or

C

CALDue
CALIbrator
CALProbe
CALStatus
CALTempdelta
CCAlconstants
CENter
CENTRonics
CH
CHIme
CHKsm0
CLEar
CMDerr
COLor
COMpare
CONDacq
CONFig
CONSecpts
CONTInuous
COPy
COUpling
CR
CRLf
CROss
CRVchk
CURRent
CURSor
CURVe

D

DAInt
DATACompress
DATAFormat
DATE
DB
DC
DCHf
DCNoise
DEBug
DEF
DEFAult
DEGrees
DELAy
DELete
DELTa
DEScription
DIAg
DIGitizer
DIRection
DISAble
DISPlay
DISTal
DIThered
DIVS
DLYtrace
DOT1Abs
DOT1Rel
DOT2Abs
DOT2Rel
DOTs
DRAft
DSYmenu
DSYStotd
DUAl

E

ECHo
ECL
EMPty
ENAble
ENCdg
ENHanced
ENV
EOL
EQ
EVEN
EVENT
EVENTS
EVHoldoff
EXErr
EXWarn

F

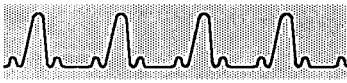
FAlled
FALltime
FASt
FEOi
FFT
FILl
FILTer
FLAgging
FORce
FORMat
FPAnel
FPNext
FPS
FPSList
FPSNum
FPUpdate
FREq
FROm
FULl

G

GAIn
GPIb
GRAticule
GRLocation
GRType
GT

H

H1Bar
H2Bar
HAMming
HANning
HARd
HBArs
HERtz
HIPrec
HIRes
HMAg
HORiz
HPGl
HPOsition
HSBatt
HUE
HVPosition
HVSize



I

ID
IDLe
IDProbe
IMPedance
INCAcq
INErr
INIt
INPut
INTensity
INTERleave
INWarn

K

KEEp

L

LABAbs
LABel
LABRel
LCAlconstants
LEFt
LENgth
LEVel
LF
LFCr
LIGhtness
LINear
LMZone
LONgform
LOWer
LSB
LT

M

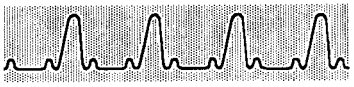
MAIn
MAINPos
MANual
MAX
MCAlconstants
MEAN
MEAS
MESial
MID
MIN
MINUs
MNSCoupling
MNSOffset
MNSProbe
MODe
MSB
MSLIst
MSLOpe
MSNum
MSYs
MTRack
MULTitrace

N

NAVg
NENHanced
NENv
NEVer
NEWconfig
NEXt
NEXTRep
NLEvel
NONe
NORmal
NOTrg
NR.pt
NREptrig
NT
NTAuto
NULl
NVRam

O

ODD
OFF
OFFSet
OHMs
ON
OPCmpl
OPTional
OPTIONS
OUTput



P

PAIred
PANzoom
PARity
PASsed
PATh
PCTg
PDElay
PERiod
PHAse
PIN8
PIN24
PIVersion
PIVOt
PLOtter
PLSCoupling
PLSOffset
PLSProbe
PLUs
PORt
POWeron
PP
PREvious
PRInter
PRINTIng
PROBe
PROTect
PROXimal
PT.fmt
PZMode

R

RATe
RCAIconstants
REAdout
RECall
RECTangular
REDuced
REFErence
REFLevel
REFset
REMAining
REPeat
REMove
RI
RIGht
RISetime
RMS
RMZone
RQS
RS232
RUN

S

SATuration
SAVe
SAVEFactory
SCAn
SCANStowfm
SCLockd
SCReen
SEConds
SELect
SELFcal
SENsitivity
SEQuence
SET
SETSeq
SHOrt
SINgle
SLOpe
SNRatio
SOFt
SOUrce
SPEaker
SPLit
SPOoling
SRQ
SRQMask
STARt
STAtus
STByte
STO
STOList
STONum
STOP
STOPBits
STORe
STORE_Recall
STRing

T

TBMain
TBWin
TEK4692
TEK4696
TESt
TEXt
TIHoldoff
TIMe
TIMER1
TIMER2
TO
TOPline
TOTalpts
TR
TRAce
TRANUm
TRG
TRIAngular

TRIgger
TRMain
TRSep
TRWin
TSMain
TTAverage
TTL
TTRig
TYPe

U

UID
UN
UNDEF
UNDO
UNIts
UPPer
UPTime
USEr
USERId
USIng
UTIlity
UTILITY1
UTILITY2

V

V1Bar
V2Bar
VBArs
VC
VCOffset
VECtors
VERBose
VERt
VOLts
VPOsition
VSIze

W

WATts
WAVfrm
WFId
WFMCalc
WFMpre
WFMScaling
WFMSCAN
WHOle
WIDth
WIN1Pos
WIN2Pos
WINDow
WTMode

X

X
XCOord
XDIv
XINcr
XMUlt
XQUal
XTNd
XUNit
XY
XZEro

Y

Y
YCOord
YDIv
YMUlt
YQUal
YTEnergy
YTMns_area
YTPls_area
YUNit
YZEro

Appendix C: Character Sets

The character sets include standard ASCII characters and a special set of characters that include math, Greek, European, Spanish, and graphic symbols.

The special "escape" characters are formed by putting an ASCII escape character (decimal 27) in front of another ASCII character. For example, to place an integral math symbol (∫) on the DSA display, enter an escape character (represented by *<ESC>*) followed by the letter **d**.

```
TEXT STRING:"<ESC>d"
```

For more information on placing characters on the display, see the TEXT command entry in the *Command Reference*.

The character-set tables begin on the following page.

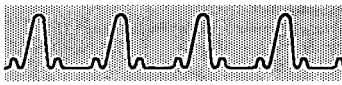
ASCII Character Set

	0	1	2	3	4	5	6	7
0	0 NUL	16 DLE	32	48 0	64 @	80 P	96 `	112 p
1	1 SOH	17 DC1	33 !	49 1	65 A	81 Q	97 a	113 q
2	2 STX	18 DC2	34 "	50 2	66 B	82 R	98 b	114 r
3	3 ETX	19 DC3	35 #	51 3	67 C	83 S	99 c	115 s
4	4 EOT	20 DC4	36 $	52 4	68 D	84 T	100 d	116 t
5	5 ENQ	21 NAK	37 %	53 5	69 E	85 U	101 e	117 u
6	6 ACK	22 SYN	38 &	54 6	70 F	86 V	102 f	118 v
7	7 BEL	23 ETB	39 '	55 7	71 G	87 W	103 g	119 w
8	8 BS	24 CAN	40 (	56 8	72 H	88 X	104 h	120 x
9	9 HT	25 EM	41)	57 9	73 I	89 Y	105 i	121 y
A	10 LF	26 SUB	42 *	58 :	74 J	90 Z	106 j	122 z
B	11 VT	27 ESC	43 +	59 ;	75 K	91 [	107 k	123 {
C	12 FF	28 FS	44 ,	60 <	76 L	92 \	108 l	124 |
D	13 CR	29 GS	45 -	61 =	77 M	93]	109 m	125 }
E	14 SO	30 RS	46 .	62 >	78 N	94 ^	110 n	126 ~
F	15 SI	31 US	47 /	63 ?	79 O	95 _	111 o	127 DEL

Escape Character Set

	0	1	2	3	4	5	6	7
0	0 Ä	16 Ñ	32	48	64 Π	80 π	96 ↓	112 ···
1	1 ä	17 ñ	33	49	65 α	81 θ	97 ↑	113 Ä
2	2 Ö	18 ȷ	34	50	66 γ	82 ρ	98 →	114 E
3	3 ö	19 ¡	35	51	67 δ	83 Σ	99 ←	115 R
4	4 Ü	20 Ã	36	52	68 Δ	84 τ	100 ∫	116 T
5	5 ü	21 ã	37	53	69 ε	85 υ	101 ÷	117 ┘
6	6 à	22 À	38	54	70 φ	86 ν	102 °	118 ┐
7	7 è	23 Õ	39	55	71 Γ	87 ω	103 √	119 ┌
8	8 á	24 õ	40	56	72 Θ	88 χ	104 ¬	120 └
9	9 é	25 É	41	57	73 ι	89 ξ	105 ±	121 ┼
A	10 Å	26 Ø	42	58	74 ψ	90 ζ	106 ≠	122 ─
B	11 å	27 ø	43	59	75 κ	91 Φ	107 ≤	123 ├
C	12 Æ	28 Œ	44	60	76 λ	92 Λ	108 ≥	124 ┤
D	13 æ	29 œ	45	61	77 μ	93 Ψ	109 ©	125 ┴
E	14 ç	30 Ç	46	62	78 η	94 σ	110 ®	126 ┬
F	15 ß	31 ∞	47	63	79 Ω	95 Ξ	111 ≈	127 │

Appendix D: Utility Programs



These are some common operations that can be performed from the external interface:

- Taking measurements
- Binary waveform transfer into an array
- Storing and recalling front panel settings
- Handling SRQs (DSA service requests)
- String transfer to the DSA display

The following programs demonstrate these utilities on popular instrument controllers.

Setup

These applications are for use with the Tektronix PEP series of controllers, or IBM PC-compatible computers configured with a National Instruments GPIB-PC Interface Card. A compatible computer with a similar GPIB interface card can also be used. These programs are written in Microsoft QuickBASIC, Version 4.0.

We also show Hewlett-Packard 200 and 300 Series controller versions of these programs. These programs are written in HP BASIC, Versions 2.1 through 4.0.

DSA Interface Configuration

Set up the GPIB parameters of the DSA as follows:

DSA 601 and DSA 602 Interface Configuration

GPIB Function	Selection
Mode	TalkListen
Address	1
Terminator	EOI/LF
Debug	Off

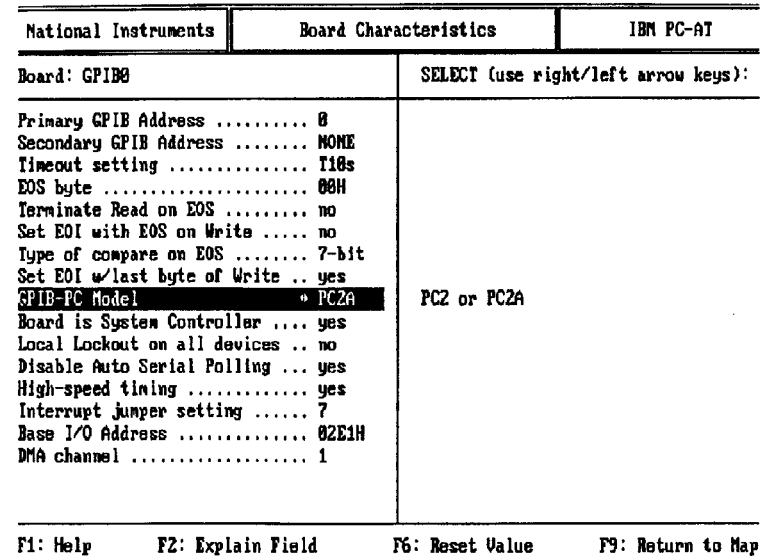
Computer Interface Configurations

The following information describes how to set up your GPIB driver system for using these programs.

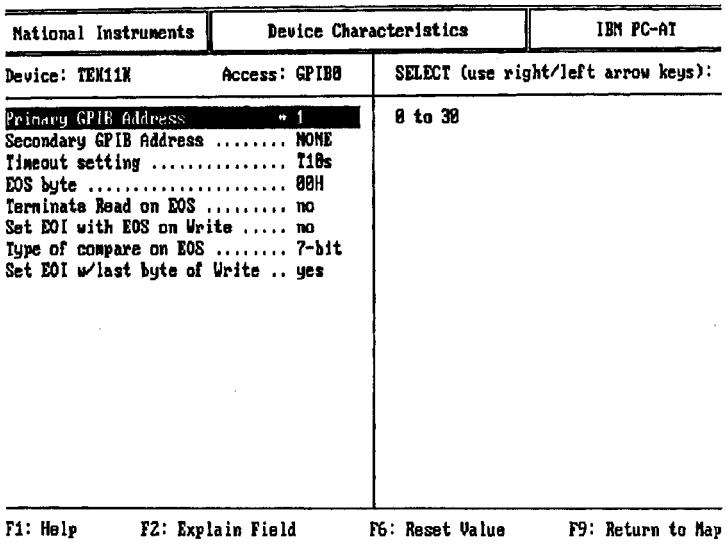
Tektronix PEP Series or IBM PC-Compatible Computers—require you to invoke the configuration program for your GPIB interface. For example, for the National Instruments GPIB-PC Interface Card, invoke the **ibconfig.exe** file and follow the instructions.

The following illustrations show the appropriate configuration for using these utility programs.

The first illustration shows how your GPIB driver board characteristics should be set, and the second illustration shows how your device (DSA) characteristics should be set.



GPIB Driver-Board (GPIB) Settings

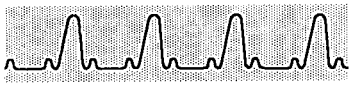


GPIB Driver-Device (Oscilloscope) Settings

Refer to your HP 200 or 300 Series controller programming manual for configuration details.

HP 200/300 Series Controllers—These programs require you to load the accompanying "I/O" file for your controller.

Note: *In these examples, it is assumed that the "@Br" and "BR%" variables identify the DSA assigned to the GPIB port of your controller.*



Tektronix PEP or IBM PC/XT/AT Controller

The following five program examples are for IBM controllers.

Taking Measurements

```
CALL IBFIND("tek11k", bd%)
CALL IBWRT(bd%, "MSLIST PER,FREQ,MAX,PP,RISE,
 FALL;MEAS?")
msg$ = SPACE$(200)
CALL IBRD(bd%, msg$)
PRINT msg$
END
```

Transferring a Binary Waveform into an Array

```
REM WFM I/O for the 11k scope using Microsoft
   QuickBASIC 4.0 & BC 6.0
CALL ibfind("tek11k", bd%)
CALL ibwrt(bd%, "LONGFORM ON;SELECT?")
msg$ = SPACE$(80): CALL ibrd(bd%, msg$)
CALL ibwrt(bd%, "ENCDG WAV:BIN;BYT.OR LSB;
 OUTPUT    " + MID$(msg$, 8, 6))

CALL ibwrt(bd%, "CURVE?")
CALL ilrd(bd%, msg$, 20)
 hbyte$ = " ": lbyte$ = " "
CALL ilrd(bd%, hbyte$, 1): CALL ilrd(bd%,
   lbyte$, 1)
bytes = ASC(hbyte$) * 256 + ASC(lbyte$)
nr.pt = (bytes - 1) / 2
DIM wfm%(nr.pt)
CALL ibrdi(bd%, wfm%(), bytes)
CALL ilrd(bd%, msg$, 1)

SCREEN 2:   WINDOW (0, -32767)-(nr.pt, 32767)
PSET (0, wfm%(0))
FOR i = 0 TO nr.pt - 1: PSET (i, wfm%(i)):
NEXT
END
```

Storing and Recalling Front Panel Settings

```
CALL ibfind("tek11k", bd%)
CALL ibwrt(bd%, "ENCDG SET:BINARY;SET?")
msg$ = SPACE$(5000)
CALL ibrd(bd%, msg$)
INPUT "Press Enter to send the setup back to
the   scope", A$
CALL ibwrt(bd%, msg$)
END
```

Handling SRQs

```
CALL ibfind("tek11k", bd%)
CALL ibwrt(bd%, "SRQMASK USER:ON;RQS ON")
PRINT "Press the RQS icon on the DSA (Esc to
  exit)"
WHILE INKEY$<>CHR$(27)
    GOSUB POLL
WEND
END

POLL:
 msg$ = SPACE$(80)
 stbyte%=0
 call ibrsp(bd%, stbyte%)
 IF stbyte%<>0 THEN
   CALL ibwrt(bd%, "EVENT?")
   CALL ibrd(bd%, msg$)
   PRINT "Status byte:";stbyte%
   PRINT msg$ : PRINT
 END IF
RETURN
```

Transferring a String to the DSA Display

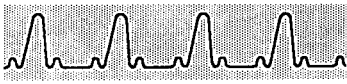
```
CALL ibfind("tek11k", bd%)
x = 5: REM x: {0 to 49}
y = 5: REM y: {0 to 31}
text$ = "'hello there world'"
msg$ = "text x:" + STR$(x) + ",y:" + STR$(y) +
  ",string:" + text$
CALL ibwrt(bd%, msg$)
END
```

HP 200 & 300 Series Controllers

The following five program examples are for HP controllers.

Taking Measurements

```
10 DIM Meas$[200]
20 ASSIGN @Br TO 701;EOL CHR$(10) END
30 OUTPUT @Br;"MSLIST PER,FRE,MAX,PP,RISE,FALL"
40 OUTPUT @Br;"MEAS?"
50 ENTER @Br;Meas$
60 PRINT Meas$
70 END
```

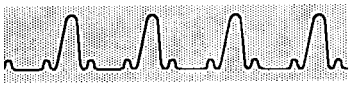


Transferring a Binary Waveform into an Array

```
10 ASSIGN @Br TO 701;EOL CHR$(10) END
20 ASSIGN @Brbin TO 701;FORMAT OFF
30 OUTPUT @Br;"LONGFORM ON"
40 OUTPUT @Br "SELECT?"
50 ENTER @Br;Trace$
60 OUTPUT @Br;"ENCDG WAVFRM:BIN;BYT.OR MSB;
    OUTPUT "&Trace$[8]
70 OUTPUT @Br;"CURVE?"
80 ENTER @Br USING "#,20A,W";Header$,Bytcnt
90 Nr_pt = (Bytcnt-1)/2
100 ALLOCATE INTEGER Curve(1:Nr_pt)
110 ENTER @Brbin;Curve(*)
120 ENTER @Br USING "B";Cksum
130 PRINT Curve(*)
140 DEALLOCATE Curve(*)
150 END
```

Storing and Recalling Front Panel Settings

```
10 DIM Setting$[5000]
20 ASSIGN @Br TO 701;EOL CHR$(10) END
30 OUTPUT @Br;"ENCDG SET:BINARY;SET?"
40 ENTER @Br USING "-K";Setting$
50 DISP "press CONTINUE to reset the front panel"
60 PAUSE
70 OUTPUT @Br;Setting$
80 END
```



Handling SRQs

```
 10 DIM Event$[100]
 20 ASSIGN @Br TO 701;EOL CHR$(10) END
 30 ON INTR 7 GOSUB Poll
 40 ENABLE INTR 7;2
 50 OUTPUT @Br;"SRQMASK USER:ON;RQS ON"
 60 DISP "press RQS icon on DSA"
 70 GOTO 70
 80 POLL: Stat = SPOLL(701)
 90  OUTPUT @Br;"EVENT?"
100  ENTER @Br;Event$
110  PRINT Stat,Event$
120  ENABLE INTR 7
130 RETURN
140 END
```

Transferring a String to the DSA Display

```
10 DIM Text$[100]
20 ASSIGN @Br TO 701;EOL CHR$(10) END
30 INPUT "TEXT: ",Text$,"LOCATION: ",X,Y
40 OUTPUT @Br;"TEXT X:";X;",Y:";Y;",
    STRING:'"&Text$&"'"
50 END
```

Appendix E: GPIB Interface Functions

GPIB Interface Functions Implemented

The following table lists the GPIB interface function and electrical function subsets supported by the DSA 601 and DSA 602 Digitizing Signal Analyzer, with a brief description of each.

GPIB Functions

Interface Function	Subset	Meaning
Acceptor Handshake	AH1	The DSA can receive multi-line messages across the GPIB from other devices.
Controller	C0	No Controller capability; the DSA cannot control other devices.
Device Clear	DC1	The DSA can respond both to the DCL (Device Clear) interface message, and to the Selected Device Clear (SDC) interface message when the DSA is listen-addressed.
Device Trigger	DT0	No Device Trigger capability; the DSA does not respond to the GET (Group Execute Trigger) interface message.
Electrical	E2	The DSA uses tri-state buffers, which are optimal for high-speed data transfer.
Listener	L4	The DSA becomes a listener when it detects its listen address being sent over the bus with the ATN line asserted. The DSA ceases to be a listener and becomes a talker when it detects its talk address being sent over the bus with the ATN line asserted.
Parallel Poll	PP0	No Parallel Poll capability; the DSA does not respond to PPC (Parallel Poll Configure), PPD (Parallel Poll Disable), PPE (Parallel Poll Enable), or PPU (Parallel Poll Unconfigure) interface messages, nor does it send out a status message when the ATN and EOI lines are asserted simultaneously.

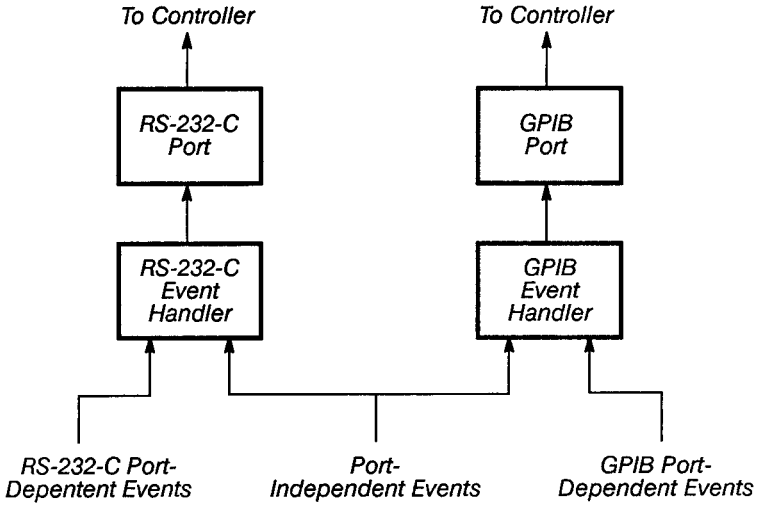
GPIB Functions (Cont.)

Interface Function	Subset	Meaning
Remote/ Local	RL1	The DSA can respond to both the GTL (Go To Local) and LLO (Local Lock Out) interface messages.
Service Request	SR1	The DSA can assert the SRQ line to notify the controller-in-charge that it requires service.
Source Handshake	SH1	The DSA can initiate multi-line messages to send across the GPIB to other devices.
Talker	T5	The DSA becomes a talker when it detects its talk address being sent over the bus with the ATN line asserted. The DSA ceases to be a talker and becomes a listener when it detects its listen address being sent over the bus with the ATN line asserted. The DSA also ceases to be a talker when it detects another device's talk address being sent over the data lines with ATN asserted.

Appendix F: System Event Handling

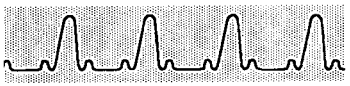
Status and Event Reporting System

Status bytes and event codes combine to represent common instrument system events. The following illustration shows the remote interface status and event reporting system of the DSA and summarizes its major elements. These elements will be discussed in the following pages.



Remote Interface Status Reporting System Block Diagram

The system events that are generated by the DSA are handled as either port-dependent or port-independent events.



Port-dependent Events

A port-dependent event is generated when one of the following system status conditions occurs:

- Command error
- Execution error
- Execution warning

Port-dependent events are returned only to the port responsible for the event. For example, if the DSA detects a command error in an RS-232-C–only command, the event associated with the error will be returned only to the RS-232-C port.

Port-independent Events

Port-independent events are always returned to both the GPIB and RS-232-C ports. A port-independent event is generated when one of the following system status conditions occurs:

- Internal error
- Internal warning
- Operation complete
- Power-on
- User request

System Event Handling Priorities

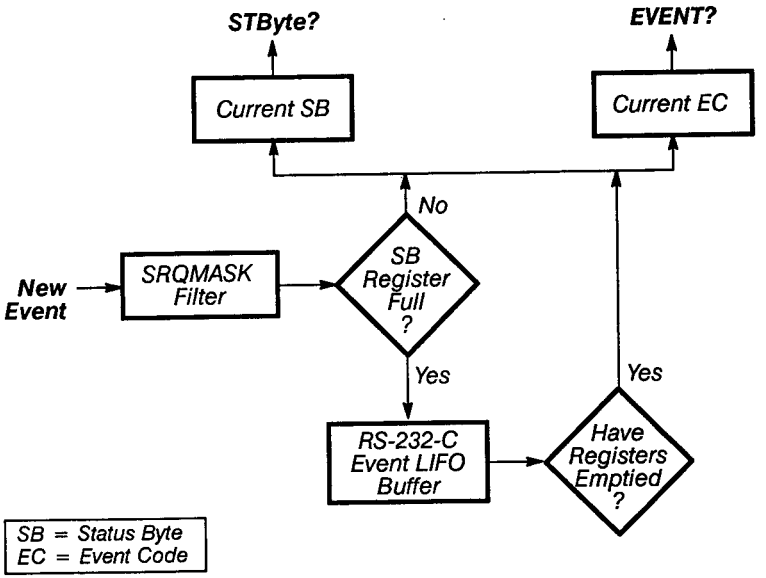
Since more than one event may occur before a GPIB or RS-232-C controller can respond to a service request, the DSA uses the following priorities to report events.

Event Priorities

Priority	Event Class
1	Power on
2	Command error Execution error Execution warning Internal error Internal warning Operation complete User request
3	No status to report

RS-232-C Event Handling

The following illustration is a block diagram of the RS-232-C event handler. The event handler consists of two software registers (SB and EC in the illustration) for the current status byte and current event code, and a LIFO (last-in first-out) buffer.



RS-232-C Event Handling

When a new event is passed to the event handler, the DSA checks the SRQMASK for that event. If the SRQMASK is off, the event is discarded. If the SRQMASK is on, the DSA checks to see if the current status byte register is empty (has "no status to report"). If it is empty, the event handler latches the new status byte and event code into the current status byte and event code registers. Once these registers contain data, all subsequent events are stacked in a 40-event LIFO buffer. Should a new event cause the LIFO buffer to overflow, the oldest event in the buffer is discarded.

Reading the RS-232-C Current Event Registers

An RS232 STBYTE? query returns the contents of the current status byte register. This is a nondestructive read.

An RS232 EVENT? query returns the contents of the current event code register and, assuming the LIFO buffer contains event(s), moves the top LIFO event into the current status byte and event code registers. If the buffer is empty, the current status byte is changed to "No Status To Report" and event code 400 is written to the current event-code register. In effect, EVENT? causes the RS-232-C event handler to update its software registers and make the next event (if any) available for subsequent STBYTE? or EVENT? queries.

GPIB Event Handling

The illustration on the following page is a block diagram of the GPIB event handler. This event handler consists of two software registers (Polled EC and Current EC in the illustration), a LIFO buffer, and the IEEE STD 488 serial poll register (a hardware register).

The RS-232-C current-status-byte software register is functionally equivalent to the serial poll hardware register diagram shown on the following page.

Operation of the GPIB event handler depends upon whether GPIB RQS is set to ON or OFF.

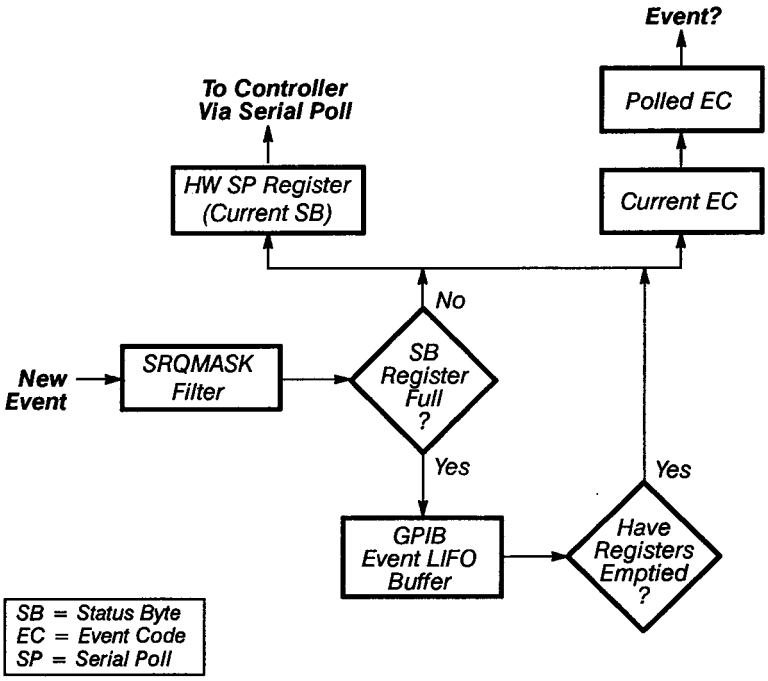
Event Reporting When GPIB RQS is Off

When GPIB RQS is off, the polled event code register is not used when a new event is passed to the event handler. If the SRQMASK for an event is off, then the event is discarded. However, if the SRQMASK for the event is on, the DSA checks to see if the current status byte register is empty or has "no status to report." If it is empty, the event handler latches the new status byte and event code into the hardware serial poll register and current event code register. Once this latched state is entered, all subsequent events are stacked in a 41-event LIFO buffer. Should a new event cause the LIFO buffer to overflow, the oldest event in the buffer is discarded.

Notice that when GPIB RQS is off, the GPIB event handler behaves virtually the same as the RS-232-C event handler, with the exception that the current status byte is stored in a hardware register and not in a software register.

Reading the GPIB Current Event Registers (RQS Off)

A GPIB controller uses an IEEE-STD-488 serial poll to read the contents of the hardware serial poll register, which is identical to the current status byte register. This is a nondestructive read. There is no DSA command provided to read the GPIB hardware serial poll register.



GPIB Event Handling

When RQS is off, only the EVENT? query updates the event handler's software and hardware registers. Repeated serial polls simply return the same status-byte value.

A GPIB EVENT? query command returns the contents of the current event code register and, assuming the LIFO buffer contains event(s), moves the top LIFO event into the current status byte and event code registers. If the buffer is empty, the current status byte is changed to "No Status To Report" and event code 400 is written to the current event code register. In effect, EVENT? causes the GPIB event handler to update its hardware and software registers, and make the next event (if any) available for subsequent serial polls or EVENT? queries.

Event Reporting When GPIB RQS Is On

When a new event is passed to the event handler, the same operations are executed as when GPIB RQS is off. The only difference is that bit 7 of the status byte of the new event is set, causing the DSA to assert SRQ after writing the status byte to the serial poll register.

Note that when GPIB RQS is on, the polled event code register is significant. At power-on or whenever RQS is turned on, this register is initialized with event code 0, which is referred to as the NULL event. The description string of the NULL event is:

```
"RQS is ON..status byte pending"
```

Reading the GPIB Current Event Registers (RQS On)

When GPIB RQS is on, it is the IEEE-STD-488 serial poll (not the EVENT?) that causes the event handler to update its event registers.

When the DSA asserts SRQ, an external controller must first serially poll the DSA to read the status byte of the system event that just occurred. The DSA responds to the serial poll by moving the current event code register contents into the polled event code register. The DSA next checks for pending events in the LIFO buffer. If found, the DSA moves the status byte of the top event into the hardware serial poll register, thus updating it and causing the DSA to generate another SRQ. At the same time, the event code for top event is moved into the current event code register, thus updating it. However, if no events are pending in the LIFO buffer, the DSA moves a status byte into the hardware serial poll register that indicates No Status To Report, and its corresponding event code 400 is moved into the current event code register. No SRQ is generated under these conditions.

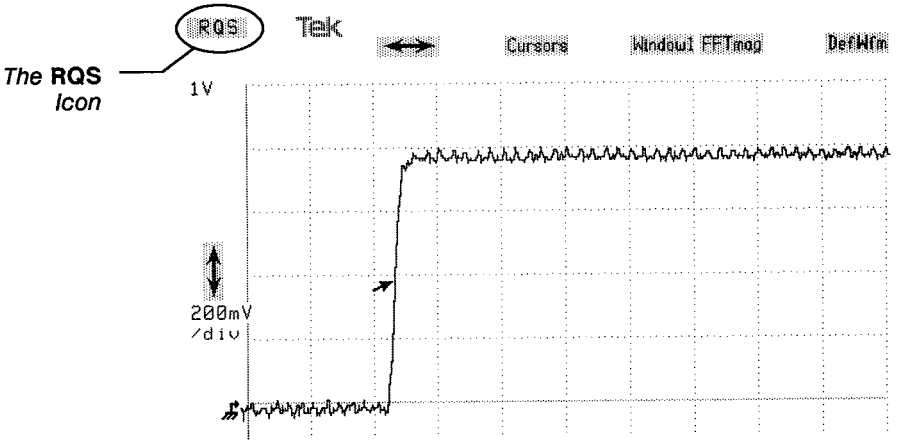
If a controller sends an EVENT? following the serial poll, the DSA returns the contents of the polled event code register and initializes it to the NULL event. Then, at the next serial poll, the DSA again moves the contents of the updated current event code register into the polled event code register. This operation ensures that the status byte and the polled event code.

Summary of Important Points When RQS is On

- The EVENT? query returns the contents of the polled event code register.
- The proper sequence for reading event registers is to first serial poll the DSA and then, if more information is desired, follow up with an EVENT? query.
- When EVENT? returns the NULL event, the DSA is signaling that a new event has occurred and its status byte must first be polled before its event code can be queried.
- If more than one event is pending and the DSA is serially polled twice with no intervening EVENT?, the event code associated with the first polled status byte is lost.

Turning On the RQS Icon with SRQMASK USER

The SRQMASK USER command allows you to make a Request for Service (RQS) from the front panel. When SRQMASK USER is on at either the GPIB or RS-232-C port, the DSA displays an RQS icon on its front panel. When initially displayed, the RQS icon is not highlighted and is not selected. When you touch the RQS icon, the icon is highlighted and an event 403 ("Front panel RQS icon selected") is reported to the ASCII port. When SRQMASK USER is off at both ports, the icon is not displayed. Since both USER masks are off by default at power-on, the RQS icon is not visible at that time.



The **RQS** *Icon*

RQS Icon on the Front Panel Display

The RQS icon changes from selected to not selected under any one of these circumstances:

- Both GPIB SRQMASK USER and RQS are on and a GPIB controller serially polls (and thereby clears) the status byte associated with event code 403.
- The GPIB SRQMASK USER is on and RQS is off and a GPIB controller uses EVENT? to read (and thereby clear) event code 403 from the GPIB event stack.
- The RS-232-C SRQMASK USER is on and an RS-232-C controller uses EVENT? to read event code 403.
- The GPIB SRQMASK USER is on and DCL (Device Clear) or SDC (Selected Device Clear) is received at the GPIB port. In this situation, all pending events (including event 403) are discarded. RS-232-C DCL has the same effect (assuming the RS-232-C SRQMASK USER is on).

- The GPIB SRQMASK USER is on and event code 403 is discarded from the GPIB stack. This situation arises when a GPIB controller does not query the GPIB event stack and subsequent DSA events cause the stack to overflow. When event code 403 is discarded, the message **Request for external service ignored** appears on the screen.

  If the RS-232-C SRQMASK USER is on and the above condition appears at the RS-232-C port, the DSA takes the same actions as it did for the GPIB interface.

Events Reported at Instrument Power-On

When the DSA is powered on, diagnostic tests automatically execute (unless bypassed with hardware straps). When diagnostics complete, nondiagnostic firmware in the DSA takes over and the remote interfaces are activated. The DSA then reports power-on status: event 401 (power on) if diagnostics passed or were bypassed, or event 394 if diagnostics failed. Specific information about diagnostic failure can be obtained with the DIAG? query-only command.

Following the power-on status report, the integrity of the DSA nonvolatile RAM (NVRAM) is checked and, if found to be unsatisfactory, one of the following events is reported:

- Event 657—NVRAM was completely initialized and all stored settings (if any) were discarded. This event is typically reported when the NVRAM battery fails.

- Event 658—This is the same as event 657, except that the following conditions are not initialized from the factory settings: mainframe link of the UID? command, the number of times the DSA has been powered on, and the length of time the DSA has been powered on.

Event 658 is typically reported when bad settings are passed to the DSA from an external interface, causing software crash. In this case, event 658 is reported when the DSA is subsequently powered off and then on.

Appendix G: Sampling Rates and Intervals

This appendix contains tables of the sampling rates in samples per second, followed by tables of sampling intervals in seconds, for all combinations of {TBMAIN | TBWIN} TIME and LENGTH.

Sampling Rate (Samples-Per-Second), for LENGTHs 512 to 5120

	LENGTH Values				
TIME	**512**	**1024**	**2048**	**4096**	**5120**
100.0 s	0.5 S	1.0 S	2.0 S	5.0 S	5.0 S
50.0 s	1.0 S	2.0 S	4.0 S	10.0 S	10.0 S
20.0 s	2.5 S	5.0 S	10.0 S	25.0 S	25.0 S
10.0 s	5.0 S	10.0 S	20.0 S	50.0 S	50.0 S
5.0 s	10.0 S	20.0 S	40.0 S	100.0 S	100.0 S
2.0 s	25.0 S	50.0 S	100.0 S	250.0 S	250.0 S
1.0 s	50.0 S	100.0 S	200.0 S	500.0 S	500.0 S
500.0 ms	100.0 S	200.0 S	400.0 S	1.0 kS	1.0 kS
200.0 ms	250.0 S	500.0 S	1.0 kS	2.5 kS	2.5 kS
100.0 ms	500.0 S	1.0 kS	2.0 kS	5.0 kS	5.0 kS
50.0 ms	1.0 kS	2.0 kS	4.0 kS	10.0 kS	10.0 kS
20.0 ms	2.5 kS	5.0 kS	10.0 kS	25.0 kS	25.0 kS
10.0 ms	5.0 kS	10.0 kS	20.0 kS	50.0 kS	50.0 kS
5.0 ms	10.0 kS	20.0 kS	40.0 kS	100.0 kS	100.0 kS
2.0 ms	25.0 kS	50.0 kS	100.0 kS	250.0 kS	250.0 kS
1.0 ms	50.0 kS	100.0 kS	200.0 kS	500.0 kS	500.0 kS
500.0 µs	100.0 kS	200.0 kS	400.0 kS	1.0MS	1.0MS
200.0 µs	250.0 kS	500.0 kS	1.0MS	2.5MS	2.5MS
100.0 µs	500.0 kS	1.0MS	2.0MS	5.0MS	5.0MS
50.0 µs	1.0MS	2.0MS	4.0MS	10.0MS	10.0MS
40.0 µs	–	–	–	–	–

Sampling Rate (Samples-Per-Second), for LENGTHs 512 to 5120

	LENGTH Values				
TIME	**512**	**1024**	**2048**	**4096**	**5120**
20.0 µs	2.5MS	5.0MS	10.0MS	25.0MS	25.0MS
10.0 µs	5.0MS	10.0MS	20.0MS	50.0MS	50.0MS
8.0 µs	-	-	-	-	-
5.0 µs	10.0MS	20.0MS	-	100.0MS	100.0MS
4.0 µs	-	-	50.0MS	-	-
2.5 µs	-	-	-	-	-
2.0 µs	25. 0MS	50.0MS	100.0MS	250.0MS	250.0MS
1.0 µs	50.0MS	100.0MS	-	500.0MS	500.0MS
800.0 ns	-	-	250.0MS	-	-
500.0 ns	1 00.0MS	-	-	1.0 GS	1.0 GS
400.0 ns	-	250.0MS	500.0MS	-	-
250.0 ns	-	-	-	2.0 GS	2.0 GS
200.0 ns	250.0MS	500.0MS	1.0 GS	-	-
100.0 ns	5 00.0MS	1.0GS	2.0 GS	5.0 GS	5.0 GS
50.0 ns	1.0 GS	2.0GS	4.0 GS	10.0 GS	10.0 GS
25.0 ns	2.0 GS	-	-	-	-
20.0 ns	-	5.0GS	10.0 GS	25.0 GS	25.0 GS
10.0 ns	5.0 GS	10.0GS	20.0 GS	50.0 GS	50.0 GS
5.0 ns	10.0 GS	20.0GS	40.0 GS	100.0 GS	100.0 GS
4.0 ns	-	-	-	-	-
2.0 ns	25.0 GS	50.0GS	100.0 GS	250.0 GS	250.0 GS
1.0 ns	50.0 GS	100.0GS	200.0 GS	500.0 GS	500.0 GS
500.0 ps	100.0 GS	200.0GS	-	1.0 TS	1.0 TS
400.0 ps	-	-	500.0 GS	-	-
200.0 ps	250.0 GS	500.0 GS	1.0 TS	-	-

Sample Rates (Samples-Per-Second), for LENGTHs 8192 to 32768

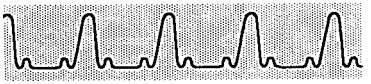
	LENGTH Values				
TIME	**8192**	**10240**	**16384**	**20464**	**32768**
100.0 s	10.0 S	10.0 S	20.0 S	20.0 S	50.0 S
50.0 s	20.0 S	20.0 S	40.0 S	40.0 S	100.0 S
20.0 s	50.0 S	50.0 S	100.0 S	100.0 S	250.0 S
10.0 s	100.0 S	100.0 S	200.0 S	200.0 S	500.0 S
5.0 s	200.0 S	200.0 S	400.0 S	400.0 S	1.0 kS
2.0 s	500.0 S	500.0 S	1.0 kS	1.0 kS	2.5 kS
1.0 s	1.0 kS	1.0 kS	2.0 kS	2.0 kS	5.0 kS
500.0 ms	2.0 kS	2.0 kS	4.0 kS	4.0 kS	10.0 kS
200.0 ms	5.0 kS	5.0 kS	10.0 kS	10.0 kS	25.0 kS
100.0 ms	10.0 kS	10.0 kS	20.0 kS	20.0 kS	50.0 kS
50.0 ms	20.0 kS	20.0 kS	40.0 kS	40.0 kS	100.0 kS
20.0 ms	50.0 kS	50.0 kS	100.0 kS	100.0 kS	250.0 kS
10.0 ms	100.0 kS	100.0 kS	200.0 kS	200.0 kS	500.0 kS
5.0 ms	200.0 kS	200.0 kS	400.0 kS	400.0 kS	1.0MS
2.0 ms	500.0 kS	500.0 kS	1.0MS	1.0MS	2.5MS
1.0 ms	1.0MS	1.0MS	2.0MS	2.0MS	5.0MS
500.0 µs	2.0MS	2.0MS	4.0MS	4.0MS	10.0MS
200.0 µs	5.0MS	5.0MS	10.0MS	10.0MS	25.0MS
100.0 µs	10.0MS	10.0MS	20.0MS	20.0MS	50.0MS
50.0 µs	20.0MS	20.0MS	-	-	100.0MS
40.0 µs	-	-	50.0MS	50.0MS	-
20.0 µs	50.0MS	50.0MS	100.0MS	100.0MS	250.0MS
10.0 µs	100.0MS	100.0MS	-	-	500.0MS
8.0 µs	-	-	250.0MS	250.0MS	-

Sample Rates (Samples-Per-Second), for LENGTHs 8192 to 32768

	LENGTH Values				
TIME	**8192**	**10240**	**16384**	**20464**	**32768**
5.0 µs	-	-	-	-	1.0 GS
4.0 µs	250.0MS	250.0MS	500.0MS	500.0MS	-
2.5 µs	-	-	-	-	2.0 GS
2.0 µs	500.0MS	500.0MS	1.0 GS	1.0 GS	-
1.0 µs	1.0 GS	1.0 GS	2.0 GS	2.0 GS	5.0 GS
800.0 ns	-	-	-	-	-
500.0 ns	2.0 GS	2.0 GS	4.0 GS	4.0 GS	10.0 GS
400.0 ns	-	-	-	-	-
250.0 ns	-	-	-	-	-
200.0 ns	5.0 GS	5.0 GS	10.0 GS	10.0 GS	25.0 GS
100.0 ns	10.0 GS	10.0 GS	20.0 GS	20.0 GS	50.0 GS
50.0 ns	20.0 GS	20.0 GS	40.0 GS	40.0 GS	100.0 GS
25.0 ns	-	-	-	-	-
20.0 ns	50.0 GS	50.0 GS	100.0 GS	100.0 GS	250.0 GS
10.0 ns	100.0 GS	100.0 GS	200.0 GS	200.0 GS	500.0 GS
5.0 ns	200.0 GS	200.0 GS	-	-	1.0 TS
4.0 ns	-	-	500.0 GS	500.0 GS	-
2.0 ns	500.0 GS	500.0 GS	1.0 TS	1.0 TS	-
1.0 ns	1.0 TS	1.0 TS	-	-	-
500.0 ps	-	-	-	-	-
400.0 ps	-	-	-	-	-
200.0 ps	-	-	-	-	-

Sampling Intervals, for LENGTHs 512 to 5120

	LENGTH Values				
TIME	**512**	**1024**	**2048**	**4096**	**5120**
100.0 s	2.0 s	1.0 s	500.0 ms	200.0 ms	200.0 ms
50.0 s	1.0 s	500.0 ms	250.0 ms	100.0 ms	100.0 ms
20.0 s	400.0 ms	200.0 ms	100.0 ms	40.0 ms	40.0 ms
10.0 s	200.0 ms	100.0 ms	50.0 ms	20.0 ms	20.0 ms
5.0 s	100.0 ms	50.0 ms	25.0 ms	10.0 ms	10.0 ms
2.0 s	40.0 ms	20.0 ms	10.0 ms	4.0 ms	4.0 ms
1.0 s	20.0 ms	10.0 ms	5.0 ms	2.0 ms	2.0 ms
500.0 ms	10.0 ms	5.0 ms	2.5ms	1.0 ms	1.0 ms
200.0 ms	4.0 ms	2.0 ms	1.0 ms	400.0 µs	400.0 µs
100.0 ms	2.0 ms	1.0 ms	500.0 µs	200.0 µs	200.0 µs
50.0 ms	1.0 ms	500.0 µs	250.0 µs	100.0 µs	100.0 µs
20.0 ms	400.0 µs	200.0 µs	100.0 µs	40.0 µs	40.0 µs
10.0 ms	200.0 µs	100.0 µs	50.0 µs	20.0 µs	20.0 µs
5.0 ms	100.0 µs	50.0 µs	25.0 µs	10.0 µs	10.0 µs
2.0 ms	40.0 µs	20.0 µs	10.0 µs	4.0 µs	4.0 µs
1.0 ms	20.0 µs	10.0 µs	5.0 µs	2.0 µs	2.0 µs
500.0 µs	10.0 µs	5.0 µs	2.5µs	1.0 µs	1.0 µs
200.0 µs	4.0 µs	2.0 µs	1.0 µs	400.0 ns	400.0 ns
100.0 µs	2.0 µs	1.0 µs	500.0 ns	200.0 ns	200.0 ns
50.0 µs	1.0 µs	500.0 ns	250.0 ns	100.0 ns	100.0 ns
40.0 µs	-	-	-	-	-
20.0 µs	400.0 ns	200.0 ns	100.0 ns	40.0 ns	40.0 ns
10.0 µs	200.0 ns	100.0 ns	50.0 ns	20.0 ns	20.0 ns
8.0 µs	-	-	-	-	-

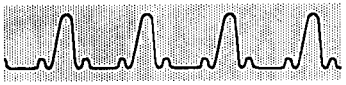


Sampling Intervals, for LENGTHs 512 to 5120

	LENGTH Values				
TIME	**512**	**1024**	**2048**	**4096**	**5120**
5.0 µs	100.0 ns	50.0 ns	–	10.0 ns	10.0 ns
4.0 µs	–	–	20.0 ns	–	–
2.5 µs	–	–	–	–	–
2.0 µs	40.0 ns	20.0 ns	10.0 ns	4.0 ns	4.0 ns
1.0 µs	20.0 ns	10.0 ns	–	2.0 ns	2.0 ns
800.0 ns	–	–	4.0 ns	–	–
500.0 ns	10.0 ns	–	–	1.0 ns	1.0 ns
400.0 ns	–	4.0 ns	2.0 ns	–	–
250.0 ns	–	–	–	500.0 ps	500.0 ps
200.0 ns	4.0 ns	2.0 ns	1.0 ns	–	–
100.0 ns	2.0 ns	1.0 ns	500.0 ps	200.0 ps	200.0 ps
50.0 ns	1.0 ns	500.0 ps	250.0 ps	100.0 ps	100.0 ps
25.0 ns	500.0 ps	–	–	–	–
20.0 ns	–	200.0 ps	100.0 ps	40.0 ps	40.0 ps
10.0 ns	200.0 ps	100.0 ps	50.0 ps	20.0 ps	20.0 ps
5.0 ns	100.0 ps	50.0 ps	25.0 ps	10.0 ps	10.0 ps
4.0 ns	–	–	–	–	–
2.0 ns	40.0 ps	20.0 ps	10.0 ps	4.0 ps	4.0 ps
1.0 ns	20.0 ps	10.0 ps	5.0 ps	2.0 ps	2.0 ps
500.0 ps	10.0 ps	5.0 ps	–	1.0 ps	1.0 ps
400.0 ps	–	–	2.0 ps	–	–
200.0 ps	4.0 ps	2.0 ps	1.0 ps	–	–

Sampling Intervals, for LENGTHs 8192 to 32768

	LENGTH Values				
TIME	**8192**	**10240**	**16384**	**20464**	**32768**
100.0 s	100.0 ms	100.0 ms	50.0 ms	50.0 ms	20.0 ms
50.0 s	50.0 ms	50.0 ms	25.0 ms	25.0 ms	10.0 ms
20.0 s	20.0 ms	20.0 ms	10.0 ms	10.0 ms	4.0 ms
10.0 s	10.0 ms	10.0 ms	5.0 ms	5.0 ms	2.0 ms
5.0 s	5.0 ms	5.0 ms	2.5ms	2.5ms	1.0 ms
2.0 s	2.0 ms	2.0 ms	1.0 ms	1.0 ms	400.0 µs
1.0 s	1.0 ms	1.0 ms	500.0 µs	500.0 µs	200.0 µs
500.0 ms	500.0 µs	500.0 µs	250.0 µs	250.0 µs	100.0 µs
200.0 ms	200.0 µs	200.0 µs	100.0 µs	100.0 µs	40.0 µs
100.0 ms	100.0 µs	100.0 µs	50.0 µs	50.0 µs	20.0 µs
50.0 ms	50.0 µs	50.0 µs	25.0 µs	25.0 µs	10.0 µs
20.0 ms	20.0 µs	20.0 µs	10.0 µs	10.0 µs	4.0 µs
10.0 ms	10.0 µs	10.0 µs	5.0 µs	5.0 µs	2.0 µs
5.0 ms	5.0 µs	5.0 µs	2.5 µs	2.5 µs	1.0 µs
2.0 ms	2.0 µs	2.0 µs	1.0 µs	1.0 µs	400.0 ns
1.0 ms	1.0 µs	1.0 µs	500.0 ns	500.0 ns	200.0 ns
500.0 µs	500.0 ns	500.0 ns	250.0 ns	250.0 ns	100.0 ns
200.0 µs	200.0 ns	200.0 ns	100.0 ns	100.0 ns	40.0 ns
100.0 µs	100.0 ns	100.0 ns	50.0 ns	50.0 ns	20,0 ns
50.0 µs	50.0 ns	50.0 ns	-	-	10.0 ns
40.0 µs	-	-	20.0 ns	20.0 ns	-
20.0 µs	20.0 ns	20.0 ns	10.0 ns	10.0 ns	4.0 ns
10.0 µs	10.0 ns	10.0 ns	-	-	2.0 ns



Sampling Intervals, for LENGTHs 8192 to 32768

	LENGTH Values				
TIME	**8192**	**10240**	**16384**	**20464**	**32768**
8.0 µs	-	-	4.0 ns	4.0 ns	-
5.0 µs	-	-	-	-	1.0 ns
4.0 µs	4.0 ns	4.0 ns	2.0 ns	2.0 ns	-
2.5 µs	-	-	-	-	500.0 ps
2.0 µs	2.0 ns	2.0 ns	1.0 ns	1.0 ns	-
1.0 µs	1.0 ns	1.0 ns	500.0 ps	500.0 ps	200.0 ps
800.0 ns	-	-	-	-	-
500.0 ns	500.0 ps	500.0 ps	250.0 ps	250.0 ps	100.0 ps
400.0 ns	-	-	-	-	-
250.0 ns	-	-	-	-	-
200.0 ns	200.0 ps	200.0 ps	100.0 ps	100.0 ps	40.0 ps
100.0 ns	100.0 ps	100.0 ps	50.0 ps	50.0 ps	20.0 ps
50.0 ns	50.0 ps	50.0 ps	25.0 ps	25.0 ps	10.0 ps
25.0 ns	-	-	-	-	-
20.0 ns	20.0 ps	20.0 ps	10.0 ps	10.0 ps	4.0 ps
10.0 ns	10.0 ps	10.0 ps	5.0 ps	5.0 ps	2.0 ps
5.0 ns	5.0 ps	5.0 ps	-	-	1.0 ps
4.0 ns	-	-	2.0 ps	2.0 ps	-
2.0 ns	2.0 ps	2.0 ps	1.0 ps	1.0 ps	-
1.0 ns	1.0 ps	1.0 ps	-	-	-
500.0 ps	-	-	-	-	-
400.0 ps	-	-	-	-	-
200.0 ps	-	-	-	-	-

Glossary

Acquisition
The process of repeatedly sampling the signals coming through input channels, and accumulating the samples into waveforms.

ASCII
Acronym for American Standard Code for Information Interchange. ASCII is a standard eight-bit code used by many computers and data terminals.

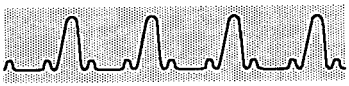
Asynchronous
Relating to data transmissions which are not synchronized through a system clock. Also, errors which are not synchronized with a command.

Autoset
A means of letting the DSA set itself to provide a stable and meaningful display of a given waveform.

Averaging
Displaying a waveform that is the combined result of several acquisitions, thereby reducing random noise.

Binary Block
Tektronix-specified format for binary data transmissions: %,<*byte count*><*data value*><*data value*>...<*data value*><*checksum*>.

BNF
Acronym for Backus-Naur Form, which is a formal language structure for syntax definition.



Channel
The electrical path from a plug-in amplifier input to the digitizer, i.e., an input corresponding to one of the BNC connectors on a plug-in amplifier. Also, the smallest component of a waveform expression.

Channel Number
An identifier, usually in the form <*slot*> <*ui*>, that distinguishes plug-in unit channels, e.g., "L4."

Checksum
Checksum comparison is a serial communication operation used to verify data accuracy by comparing the sum of data received against a previously computed sum (checksum).

Complex Waveform
A waveform with a waveform expression beyond a single channel specification. Any waveform using a numeric value, a function, a reference to a stored waveform, or an arithmetic operator is a complex waveform. However, using the average function does not make a waveform complex.

Concatenate
To link commands together.

Cursor
Any of four styles of paired markers that you position with the knobs or CURSOR commands. The DSA displays the positions of the cursors and the distance between them, in axis units.

DCE
Acronym for data communications equipment. The DSA is configured as a DCE device as defined in the EIA standard RS-232-C.

Debug Mode
Copies input data from either the GPIB or the RS-232-C interface to the front panel display for program troubleshooting.

Default Measurement Parameter
A value from the default set of measurement parameters. The operator can change the default values. Whenever a waveform is created, the measurement parameters are copied from the default set.

Device-Dependent Message
Messages initiated by a controller that can only be understood by a specific device. The entire command set are the primary device-dependent messages for the DSA.

Distal
The point farthest (most distant) from a reference point. As used in the DSA, the ending measurement point for timing measurements.

DMA
Acronym for direct memory access. DMA capability is a feature available in some controllers that transfers data directly into memory by bypassing the central processing unit (CPU). The DSA comes standard with a GPIB-compatible DMA.

DTE
Acronym for data terminal equipment which is a computer or terminal as defined in EIA standard RS-232-C.

DUT
Acronym for device under test.

EIA
Acronym for Electronics Industries Association.

Enveloping
Displaying a waveform that shows the extremes of variation of several acquisitions.

Escape Character Set
An alternate character set that is accessed by including an ASCII escape character (decimal 27) in front of the appropriate ASCII character.

FFT
Fast Fourier Transform. An algorithm used to convert time domain data to frequency domain data.

FFT Window
A means of modifying time domain data prior to conversion to frequency domain data to reduce frequency leakage, which is caused by discontinuities between the end and the beginning of the time domain data.

Floating Point Value
A type of numeric argument (<*NR2*> or <*NR3*>) that includes a decimal point and may include an exponent.

GPIB
Acronym for General Purpose Interface Bus. The GPIB interface is an eight-bit parallel bus that allows remote computer control of the DSA and other synchronous devices. GPIB characteristics are specified in IEEE STD 488 1978.

Histogram
A representation of the frequency of occurrence of voltage levels where one axis represents the range of voltages and the other axis represents the frequency of occurrence within a single waveform record. The DSA internally uses a histogram of the waveform to determine topline and baseline.

IEEE STD 488
The Institute of Electrical and Electronic Engineers specification for the GPIB interface.

Initialize
Setting the DSA to a completely known set of default conditions.

LIFO
Acronym for the last-in first-out method used to process I/O buffer contents.

LSB
Acronym for least significant bit or least significant byte.

Main
Refers to the primary time base used for acquiring data. See Window.

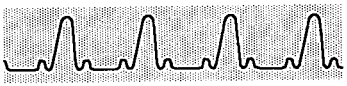
Measurement
An automated numeric readout that the DSA provides directly from the displayed waveform in real time, without operator intervention.

Measurement Parameter
One of several control/command parameters that the DSA operator can exercise over the automated measurement process.

Measurement Tracking
The process of automatically adjusting the measurement parameters to reflect changes in the waveform.

Mesial
The middle point of a range of points. As used in the DSA, the middle measurement point between proximal and distal points for timing measurements.

MSB
Acronym for most-significant bit or most-significant byte.



Nonvolatile RAM (NVRAM)
Random access memory (RAM) with a battery backup system to prevent the loss of data in case of power failure.

Parse
To decode or analyze data according to a syntax.

Point Accumulate Mode
A mode of operation where the DSA displays newly acquired waveform data points and keeps the previously acquired data points on the screen.

Proximal
The point nearest (in closest proximity) to a reference point. As used in the DSA, the beginning measurement point for timing measurements.

Quoted String
An element of DSA command syntax (<*qstring*>). A quoted string is required by some command arguments and returned as responses to specific queries. The <*qstring*> element is enclosed by quotes and can be any of the characters defined in the DSA character set.

Record Length
The number of samples (data points) that make up a waveform.

RS-232-C
An interface that allows remote operation of the DSA via a controller or terminal. Serial asynchronous data can be transmitted between the DSA and another device as defined in EIA standard RS-232-C.

Sample Interval
The time interval between successive samples in a waveform record.

Scalar
A specific quantity that has magnitude but not direction (a real number, not a vector).

Selected Waveform
The highlighted (brightest) waveform of a multi-waveform display. The selected waveform is the waveform that is acted on by the knobs, menu selectors and commands.

Setting
The state of the system at some given time.

Signed Integer Value
A type of numeric argument (<*NR1*>) which is an integer with a leading sign.

Stored Waveform
A waveform record with attributes that is saved in a dedicated area of memory.

Synchronous
Data transmission in which timing is provided by a clock in the sending unit.

Tektronix Codes and Formats
An shortform title for the Tektronix GPIB Codes, Formats, Conventions, and Features internal standard. The DSA syntax and commands comply with this internal Tektronix standard.

Time Base
The time-dependent specifications that control the acquisition of a waveform. The time base determines when and for how long to acquire and digitize signal data points.

Trace
The visible representation of an input signal or combination of signals. Identical to waveform.

Tracking
The process of automatically adjusting the measurement parameters to reflect changes in the waveform.

Trigger
An electrical event that initiates acquisition of a waveform record and to which time attributes and measurements are referenced.

Truncate
To delete less significant digits from a number. Truncation reduces precision.

Twos-Complement
A representation of negative numbers used by digital computer systems to facilitate arithmetic processing.

Uptime
The number of hours the instrument has been powered on.

Unsigned Integer
A type of numeric argument (<*ui*>) which is an integer without a leading sign (e.g., TRACE <*ui*> or TRACE3).

Waveform
The visible representation of an input signal or combination of signals. Identical to trace.

Waveform Expression
The definition of what the waveform displays. It can include one or more channels combined arithmetically and modified by functions.

Waveform Number
A number assigned by the DSA to identify a waveform. Display waveforms are numbered 1 through 8. A new waveform is always given the lowest available number.

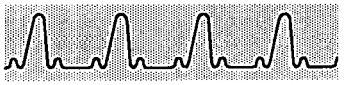
Waveform Preamble
A response returned from the WFMPRE? query that contains the scaling information for the waveform. A waveform preamble consists of the WFMPRE header followed by preamble arguments. All preamble data are ASCII encoded.

Window
Data acquired using a secondary time base with a higher sample rate (and therefore higher resolution) than the Main time base. (See Main.) A waveform that represents a horizontally expanded portion of another waveform.

XY Waveform
A waveform where both horizontal and vertical position of the data points reflect signal data.

YT Waveform
A waveform where the vertical position of the waveform data points reflects signal data, and the horizontal position of the waveform data points reflects time.

Index



A

B

C

D

E

F

G

H

I

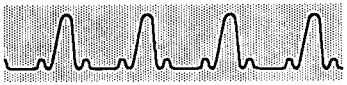
J

L

M

N

O



P

Q

R

S

T

U

V

W

MANUAL CHANGE INFORMATION

At Tektronix, we continually strive to keep up with latest electronic developments by adding circuit and component improvements to our instruments as soon as they are developed and tested.

Sometimes, due to printing and shipping requirements, we can't get these changes immediately into printed manuals. Hence, your manual may contain new change information on following pages.

A single change may affect several sections. Since the change information sheets are carried in the manual until all changes are permanently entered, some duplication may occur. If no such change pages appear following this page, your manual is correct as printed.

Tektronix
COMMITTED TO EXCELLENCE

MANUAL CHANGE INFORMATION

Date: 9/89 **Change Reference:** C2/0989

Product: DSA 601 & DSA 602 Programmer Reference **Manual Part No:** 070-7251-00

Product Group: 47

These changes are for instruments with firmware version 1.2 or above (including all new instruments). For earlier firmware versions, simply discard these insert pages.

TEXT CHANGES

Pull and replace the following pages with the pages attached to this insert:

3-1 through 3-4

3-11 through 3-24 (replace with insert pages 3-11 through 3-26)

5-13 through 5-20

B-1 through B-6

Glossary-5 through Glossary-6

Command Groups

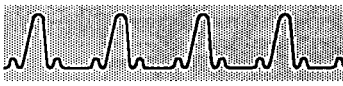
This section presents the command set arranged by function. The *DSA 601 and DSA 602 Command Reference* presents complete command information arranged alphabetically.

Functional Groups

The table below lists the groups and their function(s). The following two pages show all the commands grouped by function.

Functional Groups in the Command Set

Group	Functions Controlled
Acquisition	Acquisition (digitizing) of waveforms
Calibration/Enhanced Accuracy	Enhanced accuracy functions
Channel/Vertical	Plug-in amplifier vertical parameters
Cursor	Waveform cursor selection and positioning
Data Transfer	Transfer of waveform data and front panel settings to and from the DSA
Diagnostics	Self-tests diagnostics and extended diagnostics
Display and Color	Front panel display parameters and colors
External I/O	Printer parameters, debug functions, and RS-232-C parameters
Label and Text	Placement of user-defined labels and text
Measurement	DSA measurement functions
Miscellaneous/System	System and front panel functions
Status and Event	Instrument event reporting, hardware identification, and configuration information
Time Base/ Horizontal	Main and window record length and position
Triggering	Triggering parameters
Waveform and Settings	Waveform creation and modification, and front panel settings commands



Cursor

CURSOR
DOT1ABS
DOT2ABS
DOT1REL
DOT2REL
H1BAR
H2BAR
V1BAR
V2BAR

Data Transfer

ABBWFMPRE
BYT.OR
CURVE
ENCDG
INPUT
OUTPUT
SET?
WAVFRM?
WFMPRE

Diagnostics

DIAG?
TEST

Display and Color

COLOR
COLORMAP
DISPLAY

External I/O

ALTINK
BITMAP
COPY
DEBUG
HPGL
PIN8
PIN24
RS232
TEK4692
TEK4696
TEK4697

Status and Event

CONFIG?
EVENT?
ID?
IDPROBE?
PIVERSION?
RQS
SRQMASK
STBYTE?
UID?

Time Base / Horizontal

ADJTRACE<*ui*>
MAINPOS
TBMAIN
TBWIN
WIN1POS
WIN2POS

Triggering

TR?
TRLEVEL
TRMAIN
TRWIN
TSMAIN?
WTMODE

Waveform and Settings

ADJTRACE<*ui*>
CLEAR
DELETE
FPSLIST?
FPSNUM?
NVRAM?
PZMODE
RECALL
REMOVE
SCANSTOWFM
SELECT
SETSEQ
STOLIST?
STONUM?
STORE
TRACE<*ui*>
TRACE[<*ui*>]?
TRANUM?
WFMSCALING

DSA 601 and DSA 602 Command Groups (Cont.)

Acquisition

AUTOACQ
AUTOSET
AVG
CONDACQ
DELTA
DIGITIZER
ENV
FILTER
FFT
INCACQ
INTERLEAVE
NAVG
NENV
NREPTRIG

Calibration / Enhanced Accuracy

CALPROBE
CALSTATUS?
CALTEMPDELTA?
CCALCONSTANTS
CHSKEW?
LCALCONSTANTS
MCALCONSTANTS
RCALCONSTANTS
SELFCAL

Channel / Vertical

CH<*slot*><*ui*>
CH<*slot*>?
CH?

Label and Text

LABABS
LABEL
LABREL
TEXT

Measurement

BASELINE
COMPARE
DAINT
DISTAL
DLYTRACE
LMZONE
MEAS?
<*meas*>?
MESIAL
MLEVEL
MSCOUNT
MSLIST
MSLOPE
MS<*meas*>?
MSNUM?
MSTAT?
MSYS
MTIME
MTRACK
PROXIMAL
REFLEVEL
REFSET
REFTRACE
RMZONE
SNRATIO
STATISTICS
TOPLINE
TTAVERAGE
<*meas*> ::=
CROSS
DELAY
DUTY
FALLTIME
FREQ
GAIN
MAX
MEAN
MID
MIN
OVERSHOOT
PDELAY
PERIOD
PHASE
PP
RISETIME
RMS
SKEW
TTRIG
UNDERSHOOT
WIDTH
YTENERGY
YTMNS_AREA
YTPLS_AREA

Miscellaneous / System

ABSTOUCH
CALIBRATOR
DATE
DEF
DSYMENU?
DSYSTOFMT
DSYSTOTD
FEOI
FPANEL
FPUPDATE
HSBATT?
INIT
LONGFORM
OPTIONS?
PATH
POWERON?
PROBE
SCLOCKD
SPEAKER
TIME
UNDEF
UPTIME?
USERID

DSA 601 and DSA 602 Command Groups

Acquisition Commands

The Acquisition commands control waveform digitizing. The commands are presented in three groups: commands that control signal acquisition, such as DIGITIZER, commands that control waveform functions, such as AVG, and commands that affect acquisition parameters, such as NAVG.

Signal Acquisition Commands

Command	Meaning
AUTOACQ	Selects traces and controls memory wrapping for repetitive single trigger acquisition.
AUTOSET	Adjusts the waveform signal for optimal display.
CONDACQ	Controls the condition(s) on which the acquisition of waveforms stops.
DIGITIZER	Starts and stops waveform acquisition.

Waveform Function Commands

Command	Meaning
AVG	Turns waveform averaging on or off. (Averaging can also be defined in the waveform description; use the TRACE DESCRIPTION command in the Waveform and Settings group.)
DELTA	Compares an acquired waveform with an enveloped reference waveform and performs various actions under specified conditions.
ENV	Turns waveform enveloping on or off. (Enveloping can also be defined in the waveform description using the TRACE DESCRIPTION command.)

Sending Waveform Data to the DSA

The controller normally sends waveform data that came from a previous query.

In general, waveform data are sent back to the DSA in the following sequence:

1. If the waveform data are binary encoded, use BYT.OR to specify the transmission order of the returned data.
2. Use the INPUT command to select the stored waveform location where the data will be written.
3. Use the WFMPRE command to return the waveform preamble to the DSA.
4. Finally, use the CURVE command to return unscaled waveform data to the DSA.

Diagnostics Commands

The Diagnostics commands invoke self-tests diagnostics or extended diagnostics. The Diagnostics commands are shown in the following table:

Diagnostics Commands

Command	Meaning
DIAG?	Returns the result of the diagnostic tests.
TEST	Performs self-test or extended-test diagnostics.

Printer Commands (Cont.)

Command	Meaning
HPGL	Controls Tektronix HC100 plotters and other devices conforming to the HP-GL format.
PIN8	Controls standard Epson 8-pin bit image graphics printers, such as the Tektronix 4644.
PIN24	Controls extended Epson 24-pin dot graphics printers.
TEK4692	Controls Tektronix 4692 Color Graphics Copiers and Tektronix 4693D Color Image Printers in 4692 emulation mode.
TEK4696	Controls Tektronix 4696 and Tektronix 4695 Color Ink-Jet Printers.
TEK4697	Controls the Tektronix 4697 Color Ink-Jet Printer.

Label and Text Commands

Label and Text commands control how user text is placed on the display. Labels are text you create to identify waveforms and front panel settings. The Label and Text commands are shown in the following table:

Label and Text Commands

Command	Meaning
LABABS	Positions the label associated with the selected waveform to an absolute location.
LABEL	Defines and deletes labels, and controls whether they are displayed.
LABREL	Positions the label associated with the selected waveform to a location relative to its former absolute location.
TEXT	Defines and positions a text string on the display.

Display and Color Commands

Display commands control how waveforms appear on the display. Color commands determine the colors used on the display. The Display and Color commands are shown in the following table:

Display and Color Commands

Command	Meaning
COLOR	Determines the colors used in the display.
COLORMAP	Determines which color model is used for the display, and assigns waveform colors in the standard model.
DISPLAY	Controls display intensity, number of graticules, and whether waveforms are displayed as dots or vectors.

External I/O Commands

The External I/O commands produce hardcopy printout of the display, control the debug feature for both interfaces, and set the parameters of the RS-232-C interface. The External I/O commands are presented in two groups: interface commands and printer commands.

Interface Commands

Command	Meaning
DEBUG	Displays the ASCII commands on the front panel as they are executed via the specified interface.
RS232	Sets the parameters of the RS-232-C interface.

Printer Commands

Command	Meaning
ALTINKJET	Controls HP Thinkjet and HP LaserJet printers.
BITMAP	Controls screen capture by an external computer.
COPY	Produces a printout of the display.

Measurement Commands

The Measurement commands take waveform measurements and control the parameters with which these measurements are taken. In this discussion, measurement commands are defined first, followed by measurement parameter commands, and finally measurement execution commands.

Measurements and the Selected Waveform

All measurement commands and queries apply to the selected waveform. When no waveforms are defined, there is no selected waveform. Measurement parameters cannot be set or queried without a selected waveform.

By default, the most recently defined waveform is the selected waveform. Use the SELECT command to select a different waveform when more than one waveform appears on the display.

Each displayed waveform has a unique set of measurement parameters. Therefore, measurement queries and set commands access measurement data that apply only to the selected waveform. Altering the measurement parameters of the selected waveform has no effect on the measurement data of any other waveform.

Measurement (<*meas*>) Commands

No measurements can be taken on XY waveforms, on FFT waveforms, or on waveforms displayed in variable or infinite persistence modes.

Three types of measurements can be taken: timing, amplitude, and area/energy. The measurements are defined in the following tables. Measurements are taken on the selected waveform.

The symbol <*meas*> represents one or more measurements. A <*meas*> measurement can be individually queried or <*meas*> can represent one or more measurements in a measurement execution command.

Timing Measurements

<meas>	**Meaning**
CROSS	The time from the trigger point to a specified reference level crossing.
DELAY	The time between the first and last mesial crossing of a waveform within the measurement zone.
DUTY	The percentage of a period that a waveform spends above the mesial level.
FALLTIME	The transition time of a falling pulse edge, from the distal to proximal levels.
FREQ	Frequency (reciprocal of the period measurement).
PDELAY	Propagation delay between mesial crossings on two waveforms (used with the DLYTRACE command).
PERIOD	The time between the first and next mesial crossing of the same slope.
PHASE	The phase relationship of the reference waveform to the selected waveform, expressed as a value from -360 to +360 degrees.
RISETIME	The transition time of a rising pulse edge, from the proximal to distal levels.
SKEW	The propagation delay or time delay between mesial crossings on two different waveforms (used with the REFTRACE command).
TTRIG	The time between the main trigger point and the window trigger point.
WIDTH	The time between the first and next mesial crossing of the opposite slope.

Measurement Parameter Commands

Measurement parameters apply to the selected waveform. The following commands set the measurement parameters.

Measurement Parameter Commands

Command	Meaning
BASELINE	Sets the absolute value of the baseline when measurement tracking (MTRACK) is turned off.
DAINT	Sets the data interval (one waveform period or the entire measurement zone) for taking measurements.
DISTAL	Sets the distal (farthest from origin) reference level, typically 90% of the baseline-to-topline value.
DLYTRACE	Sets the reference (delayed) waveform used with the PDELAY measurement.
LMZONE	Sets the left limit of the measurement zone.
MESIAL	Sets the mesial (middle) reference level (the end point of the waveform period), typically 50% of the baseline-to-topline value.
MLEVEL	Determines whether certain measurement parameters are absolute voltage values or a percentages of the baseline-to-topline value.
MSCOUNT	Sets the number of samples used to compute measurement statistics.
MSLOPE	Sets the crossing slope for measurements.
MTIME	Determines whether the limits of the measurement zone are set in absolute units of the horizontal time base, or a percentage of the waveform record.
MTRACK	Controls measurement tracking (whether you or the DSA set the baseline and topline values).
PROXIMAL	Sets the proximal (nearest to origin) reference level, typically 10% of the baseline-to-topline value.
REFLEVEL	Sets a user-defined signal reference level.
REFSET	Sets the reference value(s) used in comparison mode (COMPARE is set to ON).

Amplitude Measurements

<meas>	**Meaning**
GAIN	Ratio of the peak-to-peak amplitude of the reference waveform to the peak-to-peak amplitude of the selected waveform.
MAX	Maximum amplitude (most positive peak voltage).
MEAN	Average amplitude (arithmetic mean voltage).
MID	Amplitude midpoint, halfway between the maximum amplitude and the minimum amplitude.
MIN	Minimum amplitude (most negative peak voltage).
OVERSHOOT	Difference between the maximum amplitude and the topline value, given as a percentage of the difference between the topline and baseline values.
PP	Peak-to-peak value; the voltage difference between the maximum amplitude and minimum amplitude.
RMS	True root-mean-square voltage.
UNDERSHOOT	Difference between the baseline value and the minimum amplitude, given as a percentage of the difference between the topline and baseline values.

Area/Energy Measurements

<meas>	**Meaning**
YTENERGY	The energy represented under the curve of a Yt waveform (this integral of the squared voltages can be divided by the resistance of the circuit to yield power measurements).
YTMNS_AREA	The difference between the area under a Yt curve above a specified reference level, and the area under the curve below that level.
YTPLS_AREA	The total, absolute value of all areas between a Yt waveform and a user-specified reference level.

Measurement Parameter Commands (Cont.)

Command	Meaning
REFTRACE	Sets the reference waveform used by the SKEW, GAIN and PHASE measurement.
RMZONE	Sets the right limit of the measurement zone.
SNRATIO	Sets the amplitude of a noise rejection band centered on the MESIAL level.
TOPLINE	Sets the absolute value of the topline when measurement tracking (MTRACK) is turned off.
TTAVERAGE	Sets the number of samples used by the TTRIG measurement.

Measurement parameter interactions—in the following table show how various parameters affect the measurement system.

Measurement Parameter Interactions

Parameter(s)	Effects
MTRACK	Determines whether BASELINE and TOPLINE are set dynamically by the DSA using measurement tracking (histograms) or are set by you.
BASELINE TOPLINE	Used to calculate range values for PROXIMAL, MESIAL, and DISTAL, when MTRACK is set to OFF and MLEVEL is RELATIVE.
MLEVEL	Determines whether range values for PROXIMAL, MESIAL, and DISTAL are interpreted as percentages or as absolute numbers.
DISTAL PROXIMAL	Sets the measurement end points (voltage levels) for RISETIME and FALLTIME <*meas*>.
MESIAL	Sets measurement endpoints for DELAY, DUTY, FREQ, PDELAY, PERIOD, PHASE, SKEW, and WIDTH <*meas*>; and when DAINT is set to SINGLE, for MEAN, RMS, YTENERGY, YTMNS_AREA and YTPLS_AREA <*meas*>.

Measurement Parameter Interactions (Cont.)

Parameter(s)	Effects
MTIME	Determines whether LMZONE and RMZONE ranges are in absolute horizontal units or a percentage of the waveform record.
LMZONE RMZONE	Establishes a measurement zone to isolate part of a waveform.
DAINT	Limits the measurement zone for MEAN, RMS, YTENERGY, YTMNS_AREA, and YTPLS_AREA.
REFLEVEL	Sets the crossing level for CROSS, and provides a comparison level for YTMNS_AREA <*meas*>.
MSLOPE	Sets the crossing slope for CROSS <*meas*>.
SNRATIO	Noise filter for DELAY, DUTY, FREQ, PDELAY, PERIOD, PHASE, SKEW, and WIDTH <*meas*>; and when DAINT is set to SINGLE, for MEAN, RMS, YTENERGY, YTMNS_AREA and YTPLS_AREA <*meas*>.
MSCOUNT	Sets the number of samples used to calculate statistics for MS<*meas*>? and MSTAT?.
DLYTRACE	Selects the delayed waveform used with the PDELAY <*meas*>.
REFTRACE	Selects the reference waveform used with GAIN, PHASE, and SKEW <*meas*>.

Miscellaneous/ System Commands

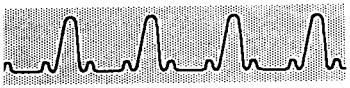
The Miscellaneous/System commands include front panel commands, system status queries, and other useful functions. The Miscellaneous/System commands are presented in two groups: front panel commands and system commands.

Front Panel Commands

Command	Meaning
ABSTOUCH	Mimics a touch to the front panel display area or the major menu buttons.
CALIBRATOR	Controls the front panel calibrator output.
DSYMENU?	Returns which major menu is currently displayed.
DSYSTOFMT	Controls whether date or hundredths of seconds is displayed in the stored waveform time and date strings.
DSYSTOTD	(Ignored in DSAs with firmware version 1.2 or above.)
FPANEL	Controls front panel lockout.
FPUPDATE	Controls when front panel readouts are updated.
PROBE	Determines the result of a probe button press.
SPEAKER	Controls whether the DSA beeps when the display is touched.

System Commands

Command	Meaning
DATE	Sets the date on the system calendar.
DEF	Defines logical names for command strings.
FEOI	Forces a message terminator in a command string.
HSBATT?	Returns the status of Option 3C, Acquisition Memory External Power Input.
INIT	Initializes the system.



Measurement Execution Commands

The following commands control the taking of measurements and the measurement execution modes.

Measurement Execution Commands

Command	Meaning
COMPARE	Controls comparison mode. When comparison mode is on, reference values set with the REFSET parameter are subtracted from measurement values; the difference is returned.
MEAS?	Executes and returns the values of the measurements in the measurement list (MSLIST).
<*meas*>?	Executes and returns the value of the specified measurement (<*meas*>).
MSLIST	Selects the measurements (<*meas*>) which are executed continuously in the Measure major menu, and are executed once at a MEAS? query.
MS<*meas*>?	Returns measurement statistics (min, max, mean, and standard deviation) for the specified <*meas*>.
MSNUM?	Returns the number of measurements in the measurement list (MSLIST).
MSTAT?	Returns measurement statistics for the measurements in MSLIST.
MSYS	Controls display of the Measure major menu.
STATISTICS	Controls whether statistics are calculated.

Event Commands

Command	Meaning
EVENT?	Returns event code information.
RQS	Sets whether the DSA asserts the SRQ line after an event occurs (GPIB only).
SRQMASK	Controls (masks) reporting of certain classes of event.
STBYTE?	Returns status byte information (RS-232-C only).

Time Base/ Horizontal Commands

The Time Base/Horizontal commands control horizontal position, establish acquisition scaling, and select a time base. The Time Base/Horizontal group contains these commands:

Time Base/Horizontal Commands

Command	Meaning
ADJTRACE <*ui*>	Controls the horizontal magnification and position of the selected waveform.
MAINPOS	Sets the horizontal position of the main waveform record with respect to the main trigger.
TBMAIN	Sets the main horizontal (time base) parameters.
TBWIN	Sets the window horizontal parameters.
WIN1POS WIN2POS	Sets the horizontal position of the window 1 or window 2 waveform record with respect to the window trigger.

Waveform Commands

Command	Meaning
STONUM?	Returns the number of stored waveforms.
STORE	Copies displayed waveforms to memory.
TRACE<*ui*>	Defines a waveform and its characteristics.
TRANUM?	Returns the number of displayed waveforms.
WFMSCALING	Sets a scaling flag to create new waveforms in fast (integer) or high-precision (floating-point) mode.

Front Panel Settings Commands

Command	Meaning
DELETE	Deletes stored waveforms or front panel settings.
FPSLIST?	Returns a list of stored front panel settings.
FPSNUM?	Returns the number of stored front panel settings.
NVRAM?	Returns the amount of available nonvolatile RAM.
RECALL	Recalls stored front panel settings from memory.
SETSEQ	Controls sequencing of front panel settings.
STORE	Stores front panel settings in nonvolatile RAM.

Triggering Commands

The Trigger commands select and define the operation of the triggering system. You can define triggers on main or window waveforms. The Trigger commands are:

Triggering Commands

Command	Meaning
TR?	Returns the same information as: TRMAIN?;TRWIN?
TRLEVEL	Controls trigger DC level mode.
TRMAIN	Sets the main trigger parameters.
TRWIN	Sets the window trigger parameters.
TSMAIN?	Returns the time from the trigger point to the 0 point, for real time single-shot acquisitions only.
WTMODE	Sets the window triggering mode.

Waveform and Settings Commands

Waveform and Setting commands select, store, remove, and specify waveform and front panel setting (FPS) characteristics. Waveform and Settings commands are presented in two groups: waveform commands and front panel settings commands.

Waveform Commands

Command	Meaning
ADJTRACE <*ui*>	Controls pan/zoom mode, vertical size and position, and window trace separation.
CLEAR	Discards acquired data for displayed waveforms.
DELETE	Deletes stored waveforms or front panel settings.
PZMODE	Controls multiple waveform pan/zoom mode.
REMOVE	Discards displayed waveforms and descriptions.
SCANSTOWFM	Controls scanning of stored waveforms.
SELECT	Designates the selected waveform.
STOLIST?	Returns a list of all stored waveforms.

Execution Errors

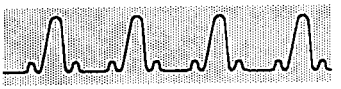
Execution errors are reported when a message is parsed but cannot be executed. Execution errors have event codes from 200 to 299. The SRQMASK for execution errors is SRQMASK EXERR. The status byte for an execution error returns **98** (decimal) with RQS set to ON, and **34** (decimal) with RQS set to OFF. All execution errors are listed on the following pages.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
222	**%O needed to support that function**	HSBATT?	Attempted to use an non-existent option.
223	**Illegal base label**	LABEL	Numerals specified as part of base label.
224	**Function not available in selected plug-in range**	CALPROBE	Attempted probe calibration on an 11A33 plug-in amplifier unit with CH<*slot*> <*ui*> PROTECT: ON.
225	**Cannot change label while current acquisition mode is running**	DELTA	Attempted to change stored waveform labels or base label with Act-on-Delta running.
		LABEL	Attempted to change stored waveform labels or base lable with Repetitive Single Trigger running.
226	**Trigger timer not available**	Trigger Source Expressions	Trigger description specifies use of a timer more than once.
227	**Not available with Extended Triggering**	TRMAIN, TRWIN, Trigger Source Expressions	Attempted to set MODE to AUTOLEVEL, SLOPE to MINUS, COUPLING to AC, ACHF, or ACNOISE, or WTMODE to EVHOLDOFF or TIHOLDOFF with extended triggering mode active (i.e., a WHILE, AND, OR, or XOR operator appearing in the Main trigger expression).
228	**Label not found**	CLEAR, DELETE, INPUT, OUTPUT, RECALL, REMOVE, SELECT, STORE, SCANSTOWFM	No matching label found with <*qstring*> syntax used.
229	**No stored waveforms**	LABEL? STO, SCANSTOWFM	No waveforms were stored when LABEL was queried or SCANSTOWFM FROM or SCANSTOWFM TO link was issued.
230	**Can't set front panel calibrator amplitude**	CALIBRATOR	Attempted to set AMPLITUDE when frequency (FREQ) is 1KHz or 1MHz.

Execution Errors

Event Code	Description	Commands that Generate Code	Explanation
203	**I/O buffers full**		Both input and output buffers are full. Output buffer is cleared.
205	**%A out of range – value ignored**	ABSTOUCH	Out-of-range ABSTOUCH argument.
		MCALCONSTANTS	Out-of-range <*ui*> values.
211	**Can't change AUTOACQ trace selection**	AUTOACQ	Attempted to turn off last trace, select more than four traces, or select XY trace for repetitive single trigger aquisition.
214	**That function is incompatible with %O**	DELTA, FFT	Attempted DELTA or FFT command with Option 3C–Acquisition Memory External Power Supply installed.
215	**Can't undo autoset**	AUTOSET	Attempted AUTOSET UNDO with no previous AUTOSET performed.
216	**Can't spool hardcopy**	COPY	Attempted COPY START when printer spooler is full.
217	**Can't keep scan waveform**	SCANSTOWFM	No current scan waveform exists to keep, or scanning was never started, or the template waveform has changed, or too many traces already exist to creat another.
218	**Can't start scanning**	SCANSTOWFM	There are no stored waveforms, or eight traces are already defined, or Repetitive Trigger or Delta is in effect.
219	**Record length of delta description test wfm cannot be greater than record length of test wfm**	DELTA	Attempted to enter new delta description or attempted to increase record length which conflicts with current delta description.
220	**Connect probe to calibrator and restart operation**	CALPROBE	Attempted to run probe calibration on a channel with no input signal.
221	**Illegal delta description**	DELTA	<*qstring*> argument not specified correctly (such as a non-envelope waveform).



Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
231	**Autoset - not functional with this waveform type**	AUTOSET	Attempted to autoset window waveform, which has no "parent" main waveform, when main waveform time base not triggered.
232	**That XY waveform has incompatible components**	TRACE<*ui*>	Attempted to create XY trace with horizontal and vertical components with incompatible scaling modes.
233	**Delayed trace must not be the selected trace**	DLYTRACE	Attempted to specify currently selected trace as PDELAY delayed trace.
234	**Unsupported printer function**	COPY	Format unsupported for currently selected printer.
235	**Duplicate label - label not changed**	LABEL, FPS<*ui*>, TRACE<*ui*>, STO<*ui*>, WFMPRE	Attempted to create a duplicate label.
236	**Illegal color number**	COLOR, HPGL, TEK4692, TEK4696	Out-of-range color index.
237	**No labels defined**	LABEL?	No labels defined for specified links.
238	**Label not defined**	LABEL?	No label is defined for FPS<*ui*>, STO<*ui*>, or TRACE<*ui*> links.
239	**Improper version number**	SET <*bblock*>	Version number of received binary settings block not the same as current firmware version number.
240	**Can't accumulate nonacquired waveform**	TRACE<*ui*>	Attempted to enable point accumulate with non-acquired trace.
241	**Too many acquisitions**	TRACE<*ui*>	Trace definition would cause the DSA to acquire more than 14 total acquisitions.
242	**ENHANCED ACCURACY available after %T**	SELFCAL	Attempted SELFCAL FORCE before 20-minute warmup elapsed.

Execution Errors (Cont.)

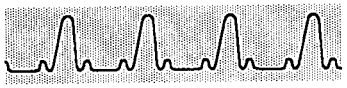
Event Code	Description	Commands that Generate Code	Explanation
243	**That function is disabled by a hardware strap**	CCALCONSTANTS, LCALCONSTANTS, RCALCONSTANTS	Attempted to set plug-in unit calibration constants with hardware strap disabled.
		MCALCONSTANTS	Attempted to set DSA calibration constants with hardware strap disabled.
		UID	Attempted to modify serial number with hardware strap disabled.
244	**%B plug-in channel(s) used differently in main and window sources**	Trigger Source Expressions	A channel was "chopped" between the main and window trigger sources, when WTMODE is set to EVHOLDOFF or TIHOLDOFF.
245	**Autoset - only functional with 11K plug-ins**	AUTOSET	Attempted to autoset waveform containing a 7000-series plug-in channel.
246	**Can't sequence settings**	RECALL	Attempted to sequence settings with SETSEQ OFF.
247	**No settings defined**	LABEL?	Attempted to query LABEL? FPS when no settings exist.
		PROBE	Attempted to assign probes to SETSEQ with no settings defined.
		RECALL	Attempted to sequence settings with no settings defined.
		SETSEQ	Attempted to turn SETSEQ on with no settings defined
248	**Misuse of AVG/ENV function**	AVG, ENV	Attempted to turn AVG or ENV on when selected trace is XY, or when selected trace is composed only of stored and scalar components. Or attempted to turn AVG or ENV off when selected waveform's vertical description not enclosed by the AVG or ENV function.
		TRACE<*ui*>	Trace description includes AVG or ENV function with a non-acquired argument, such as AVG (ST01) or ENV(1000).

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
252	**Illegal stored settings number**	DELETE, LABEL, RECALL, STORE	Out-of-range <*ui*> argument with one of these commands.
255	**Out of memory**	DELTA	Insufficient memory to create DELTA DESCRIPTION: <*qstring*>.
		DIGITIZER	Insufficient memory to save repetitive trigger or delta waveform.
		STORE	Insufficient memory to store a trace or insufficient NVRAM to store settings.
		SCANSTOWFM	KEEP link or MODE:SCAN link sent; insufficient memory to create another displayed trace.
		TRACE	Insufficient memory to create a new trace.
		Waveform Retrieval and Scaling (Data Transfer)	INPUT command references nonexistent stored waveform, insufficient memory to create stored waveform record.
		WFMPRE	Insufficient memory to create stored waveform record for preamble.
257	**Illegal stored waveform number**	DELETE, DELTA, INPUT, LABEL, OUTPUT, STORE, TRACE	Out-of-range STO <*ui*> argument for one of these commands.
		SCANSTOWFM	Argument of FROM or TO link not valid stored waveform.
263	**Illegal channel number**	CH <*slot*> <*ui*>	Attempted to set query parameters of plug-in channel that does not exist.
		TRACE <*ui*>	TRACE expression references illegal plug-in channel.
		Trigger Source Expressions	Trigger Source Expression references nonexistent 11000-series channel number.
264	**No further XY waveforms may be defined**	TRACE <*ui*>	Attempted to define more than the maximum permissible number of XY traces.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
249	**Illegal use of trace positioning function**	ADJTRACE <*ui*>	Attempted to modify HMAG, HPOSITION, HVPOSITION, HVSIZE, TRSEP, VPOSITION, or VSIZE values when modification is not permitted (for example, when PANZOOM is off).
250	**No traces defined**	ADJTRACE?, LABEL?, TRACE?	Query attempted with no traces displayed.
		LABEL	Attempted to label or delete a label on a trace when no traces are currently displayed.
		AVG, CURSOR, DOT1ABS, DOT2ABS, DOT1REL, DOT2REL, ENV, H1BAR, H2BAR, V1BAR, or V2BAR	Attempted to set or query one of these commands with no traces defined.
		BASELINE, DAINT, DISTAL, DLYTRACE, LMZONE, MESIAL, MSLOPE, MTIME,PROXIMAL, REFLEVEL, REFTRACE, RMZONE, SNRATIO, TOPLINE	Attempted to set or query one of these measurement commands with no selected trace.
251	**Illegal trace number**	ADJTRACE, AUTOACQ, CLEAR, COLOR, COLOR MAP, CURSOR, DELTA, LABEL, OUTPUT, REMOVE, SELECT, STORE, TRACE	Out-of-range <*ui*> argument with one of these commands.



Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
265	**Illegal DATE/TIME**	DATE	Illegal date value or syntax specified.
		TIME	Illegal time value or syntax specified.
		WFMPRE	Invalid date or time string entered. The date or time is set to the current clock value.
266	**DEF expansion overflow**	DEF	Expansion string overflowed internal expansion buffer.
267	**Illegal DEF string**	DEF	Illegal logical name specified.
268	**Illegal DEF recursion**	DEF	Unacceptable DEF recursion detected. Recursive logical names are acceptable only when recursion occurs to the right of an unquoted semicolon.
269	**No such trace**	ADJTRACE, AUTO-ACQ, CLEAR, CURVE, DELTA, LABEL, REMOVE, SELECT, STORE, TRACE, WAVFRM?, WFMPRE	Referenced, or attempted to set or query parameters of a nonexistent trace using one of these commands.
270	**No such stored waveform**	CURVE?	CURVE? query attempted, OUTPUT references nonexistent stored waveform.
		DELETE	Attempted to delete nonexistent stored waveform.
		DELTA	Attempted to reference a nonexistent stored waveform.
		LABEL	Attempted to label or query for a label of a nonexistent stored waveform.
		TRACE <*ui*>	TRACE expression referenced legal but undefined stored waveform.
		WAVFRM?	WAVFRM? query attempted, OUTPUT referenced nonexistent stored waveform.

Appendix B: Reserved Words

Reserved words represent the entire set of predefined command words for the DSA, including headers, links, and arguments.

In this section, reserved words appear in mixed case, with the required minimum substring in uppercase.

A

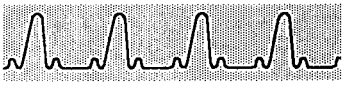
ABBwfmpre
ABOrt
ABORTIng
ABSOlute
ABStouch
AC
ACCumulate
ACHf
ACLf
ACNoise
ACState
ADJtrace
ALEvel
ALL
ALL_Wavfrm
ALTinkjet
ALWays
AMPLitude
AMPoffset
AMPS
ANBlevel
ANLevel
ARMed
ASCii
AUTO
AUTOAcq
AUTOLevel
AUTOSet
AVG

B

BASEDelta
BASELAbel
BASeline
BAUd
BINary
BINHex
BIT/nr
BITMap
BLAckman
BLHarris
BN.fmt
BOTh
BW
BWHi
BWLo
BYPassed
BYT/nr
BYT.or

C

CALDue
CALIbrator
CALJumper
CALProbe
CALStatus
CALTempdelta
CCAlconstants
CENter
CENTRonics
CH
CHIme
CHKsm0
CHSkew
CLEar
CMDerr
COLor
COLORMap
COMpare
CONDacq
CONFig
CONSecpts
CONTInuous
COPy
COUpling
CR
CRLf
CROss
CRVchk
CURRent



I

ID
IDLe
IDProbe
IMPedance
INCAcq
INErr
INFPersist
INIt
INPut
INTensity
INTERleave
INWam

J

JMPR

K

KEEp

L

LABAbs
LABel
LABRel
LCAlconstants
LEFt
LENgth
LEVel
LF
LFCr
LIGhtness
LINear
LMZone
LONgform
LOWer
LSB
LT

M

MAIn
MAINPos
MANual
MAX
MCAlconstants
MEAN
MEAS
MEMWrap
MESial
MID
MIN
MINUs
MLEvel
MNSCoupling
MNSOffset
MNSProbe
MODe
MSB
MSCount
MSCRoss
MSDElay
MSDUty
MFAIltime
MSFReq
MSGain
MSLIst
MSLOpe
MSMAx
MSMEan
MSMID
MSMIN
MSNum
MSOvershoot
MSPDelay
MSPEriod
MSPHase
MSPP
MSRisetime
MSRMs
MSSkew
MSTat
MSTtrig
MSUndershoot
MSWidth
MSYs
MSYTEnergy
MSYTMns_area
MSYTPls_area
MTIme
MTRack
MULTitrace

N

NAVg
NENHanced
NENv
NEVer
NEWconfig
NEXt
NEXTRep
NLEvel
NONe
NORmal
NOTInstalled
NOTrg
NR.pt
NREPCurve
NREptrig
NT
NTAuto
NULl
NVRam

CURSor
CURVe

D

DAInt
DATACompress
DATAFormat
DATE
DBM
DC
DCHf
DCNoise
DEBug
DEF
DEFAult
DEGrees
DELAy
DELete
DELTa
DEScription
DIAg
DIGitizer
DIGJumpers
DIRection
DISAble
DISPlay
DISTal
DIThered
DIVS
DLYtrace
DOT1Abs
DOT1Rel
DOT2Abs
DOT2Rel
DOTs
DRAft
DSYmenu
DSYSTOFmt
DSYStotd
DUAl
DUTy

E

ECHo
ECL
EDGe
EMPty
ENAble
ENCdg
ENHanced
ENV
EOL
EQ
EVEN
EVENT
EVENTS
EVHoldoff
EXErr
EXPDJumpers
EXPMJumpers
EXWarn

F

FAIled
FALltime
FASt
FEOi
FFT
FILl
FILTer
FLAgging
FORce
FORMat
FPAnel
FPNext
FPS
FPSList
FPSNum
FPUpdate
FREq
FROm
FULl

G

GAIn
GPIb
GRAticule
GRLocation
GRType
GT

H

H1Bar
H2Bar
HAMming
HANning
HARd
HBArs
HERtz
HIPrec
HIRes
HMAg
HORiz
HPGl
HPOsition
HSBatt
HUE
HUNdredths
HVPosition
HVSize

O

ODD
OFF
OFFSet
OHMs
ON
OPCmpl
OPTional
OPTIONS
ORIginal
OUTput
OVErshoot

P

PAIred
PANzoom
PARity
PASsed
PATh
PCTg
PDElay
PERiod
PERSistence
PHAse
PIN8
PIN24
PIVersion
PIVOt
PLOtter
PLSCoupling
PLSOffset
PLSProbe
PLUs
PORt
POWeron
PP
PREvious
PRInter
PRINTIng
PROBe
PROTect
PROXimal
PT.fmt
PULse
PZMode

R

RATe
RCAlconstants
REAdout
RECall
RECTangular
REDuced
REFErence
REFLevel
REFset
REFTrace
RELative
REMAining
REMove
REPCurve
REPeat
REPTrig
RI
RIGht
RISetime
RMS
RMZone
RQS
RS232
RUN

S

SATuration
SAVe
SAVEFactory
SCAn
SCANStowfm
SCLockd
SCReen
SEConds
SELect
SELFcal
SENsitivity
SEQuence
SET
SETPIPE
SETSeq
SHOrt
SINgle
SKEw
SLOpe
SNRatio
SOFt
SOUrce
SPEaker
SPLit
SPOoling
SRQ
SRQMask
STANdard
STARt
STATIstics
STAtus
STByte
STO
STOList
STONum
STOP
STOPBits
STORe
STORE_Recall
STRing
SYSMON
SYStem

T

TBMain
TBWin
TEK4692
TEK4696
TEK4697
TESt
TEXt
TIHoldoff
TIMe
TIMER1
TIMER2
TO
TOPDelta
TOPline
TOTalpts
TR
TRAce
TRANUm
TRG
TRIAngular
TRIgger
TRLevel
TRMain
TRSep
TRWin
TSMain
TTAverage
TTL
TTRig
TYPe

U

UID
UN
UNDEF
UNDershoot
UNDO
UNIts
UPPer
UPTime
USEr
USERId
USIng
UTIlity
UTILITY1
UTILITY2

V

V1Bar
V2Bar
VARPersist
VBArs
VC
VCOffset
VECtors
VERBose
VERt
VOLts
VPOsition
VSIze

W

WATts
WAVfrm
WFId
WFMCalc
WFMpre
WFMScaling
WFMSCAN
WHOle
WIDth
WIN1Pos
WIN2Pos
WINDow
WTMode

X

X
XCOord
XDIv
XINcr
XMULt
XQUal
XTNd
XUNit
XY

Y

Y
YCOord
YDIv
YMULt
YTEnergy
YTMns_area
YTPls_area
YUNit
YZEro

O

ODD
OFF
OFFSet
OHMs
ON
OPCmpl
OPTional
OPTIONS
ORIginal
OUTput
OVErshoot

P

PAIred
PANzoom
PARity
PASsed
PATh
PCTg
PDElay
PERiod
PERSistence
PHAse
PIN8
PIN24
PIVersion
PIVOt
PLOtter
PLSCoupling
PLSOffset
PLSProbe
PLUs
PORt
POWeron
PP
PREvious
PRInter
PRINTIng
PROBe
PROTect
PROXimal
PT.fmt
PULse
PZMode

R

RATe
RCAlconstants
REAdout
RECall
RECTangular
REDuced
REFErence
REFLevel
REFset
REFTrace
RELative
REMAining
REMove
REPCurve
REPeat
REPTrig
RI
RIGht
RISetime
RMS
RMZone
RQS
RS232
RUN

S

SATuration
SAVe
SAVEFactory
SCAn
SCANStowfm
SCLockd
SCReen
SEConds
SELect
SELFcal
SENsitivity
SEQuence
SET
SETPIPE
SETSeq
SHOrt
SINgle
SKEw
SLOpe
SNRatio
SOFt
SOUrce
SPEaker
SPLit
SPOoling
SRQ
SRQMask
STANdard
STARt
STATIstics
STAtus
STByte
STO
STOList
STONum
STOP
STOPBits
STORe
STORE_Recall
STRing
SYSMON
SYStem

Initialize
Setting the DSA to a completely known set of default conditions.

LIFO
Acronym for the last-in first-out method used to process I/O buffer contents.

LSB
Acronym for least significant bit or least significant byte.

Main
Refers to the primary time base used for acquiring data. See Window.

Measurement
An automated numeric readout that the DSA provides directly from the displayed waveform in real time, without operator intervention.

Measurement Parameter
One of several control/command parameters that the DSA operator can exercise over the automated measurement process.

Measurement Statistics
The accumulation of a history of individual measurement readouts, showing the mean and standard deviation of a selected number of samples.

Measurement Tracking
The process of automatically adjusting the measurement parameters to reflect changes in the waveform.

Mesial
The middle point of a range of points. As used in the DSA, the middle measurement point between proximal and distal points for timing measurements.

MSB
Acronym for most-significant bit or most-significant byte.

Nonvolatile RAM (NVRAM)
Random access memory (RAM) with a battery backup system to prevent the loss of data in case of power failure.

Parse
To decode or analyze data according to a syntax.

Point Accumulate Mode
A mode of operation where the DSA displays newly acquired waveform data points and keeps the previously acquired data points on the screen.

Proximal
The point nearest (in closest proximity) to a reference point. As used in the DSA, the beginning measurement point for timing measurements.

Quoted String
An element of DSA command syntax (<*qstring*>). A quoted string is required by some command arguments and returned as responses to specific queries. The <*qstring*> element is enclosed by quotes and can be any of the characters defined in the DSA character set.

Record Length
The number of samples (data points) that make up a waveform.

RS-232-C
An interface that allows remote operation of the DSA via a controller or terminal. Serial asynchronous data can be transmitted between the DSA and another device as defined in EIA standard RS-232-C.