Programmer Manual

Tektronix

CSA 803 & 11801A

070-7738-01

Please check for change information at the rear of this manual.

Second Printing: May 1990

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
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, IL for Israel, etc.).

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

Printed in U.S.A.

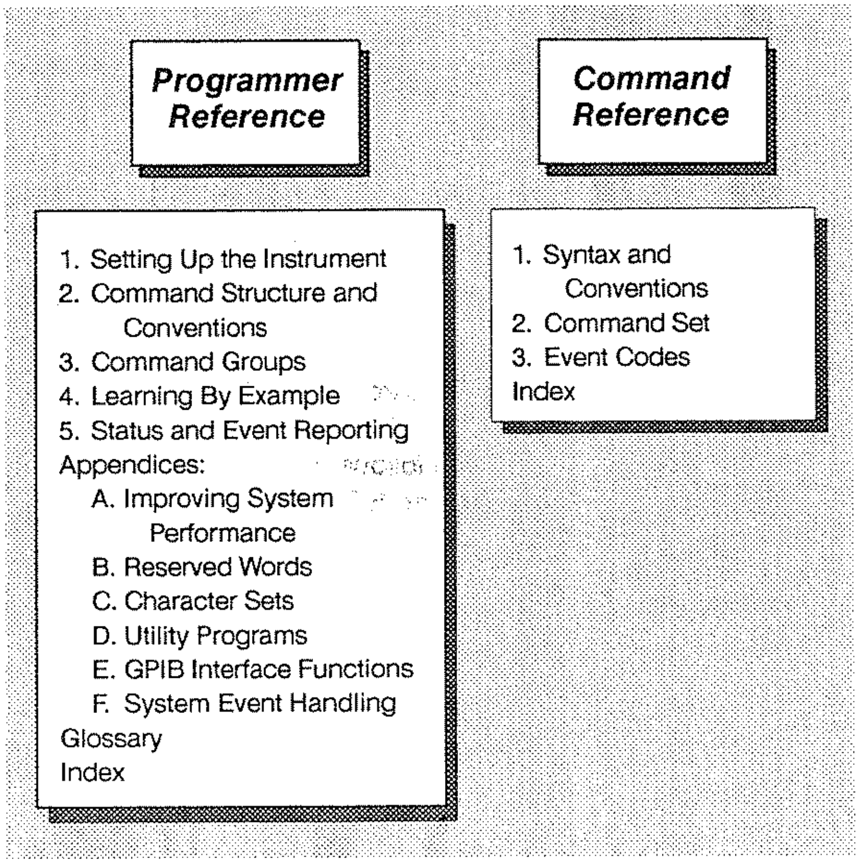
About This Manual

This document, along with the *CSA 803 and 11801A Command Reference*, provides comprehensive, easily accessible information to aid you in operating your instrument 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.

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.



Programmer Reference

1. Setting Up the Instrument
2. Command Structure and Conventions
3. Command Groups
4. Learning By Example
5. Status and Event Reporting

Appendices:

A. Improving System Performance
B. Reserved Words
C. Character Sets
D. Utility Programs
E. GPIB Interface Functions
F. System Event Handling

Glossary

Index

Command Reference

1. Syntax and Conventions
2. Command Set
3. Event Codes

Index

CSA 803 and 11801A Programmer Documentation

Related Manuals

You may want to refer to the other manuals that complete the documentation set for your CSA 803 or 11801A:

- The *CSA 803 Tutorial* (Tektronix part number 070-7718-01) or the *11801A Tutorial* (Tektronix part number 070-8025-00) for step-by-step instructions that demonstrate basic operation of the instrument.

- The *CSA 803 User Reference* (Tektronix part number 070-7719-01) or the *11801A User Reference* (Tektronix part number 070-8021-00) 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 *CSA 803 and 11801A Command Reference* (Tektronix part number 070-7720-01) describes the commands used to program the CSA 803 and 11801A.

- The *CSA 803 Service Reference* (Tektronix part number 070-7721-00) provides information to maintain and service the CSA 803, and provides a complete board-level description of CSA 803 operation. The *11801A Service Reference* (Tektronix part number 070-8024-00) provides the equivalent information for the 11801A.

- The *Installation/User* and *Service Reference* for the sampling heads you are using with your CSA 803 or 11801A.

Table of Contents

Setting Up the Instrument

Setting Up the Instrument

This section describes the implementation of each interface on the CSA 803 and 11801A, shows how to connect your instrument to other instruments that have either a GPIB or an RS-232-C interface, and explains how to set up the instrument's front panel for remote operation.

Connecting the Instrument 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:

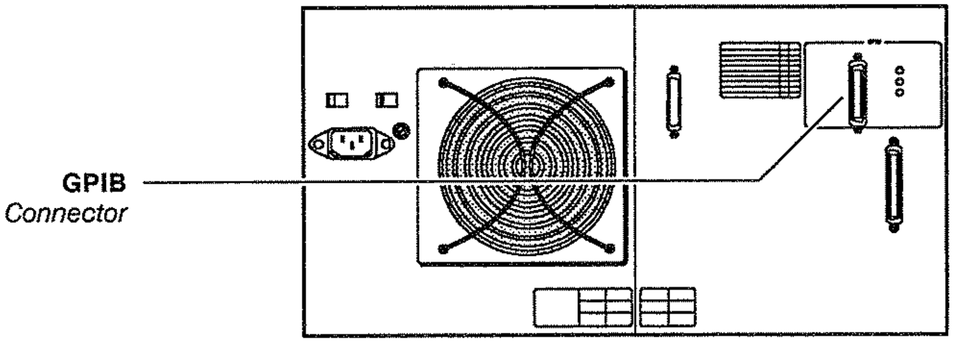
- No more than 15 devices (including the controller) can be included on a single bus.
- In order to maintain bus electrical characteristics, one device load must be connected every two meters (six feet) of cable length.. Generally, each instrument represents one device load on the bus.
- The total cumulative 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

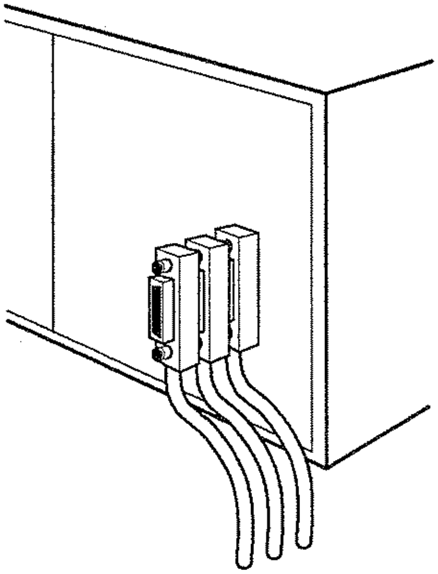
An IEEE STD 488 GPIB cable (available from Tektronix) is required to connect two GPIB devices.

GPIB 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.



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 **GPIB/RS232C** selector in the major menu area. The **GPIB/RS232C Parameters** pop-up menu appears in the display area, with the following GPIB selectors:

- **Mode** sets the instrument to be a talker/listener, to be a talker-only, or to be off the bus. Set the instrument 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.

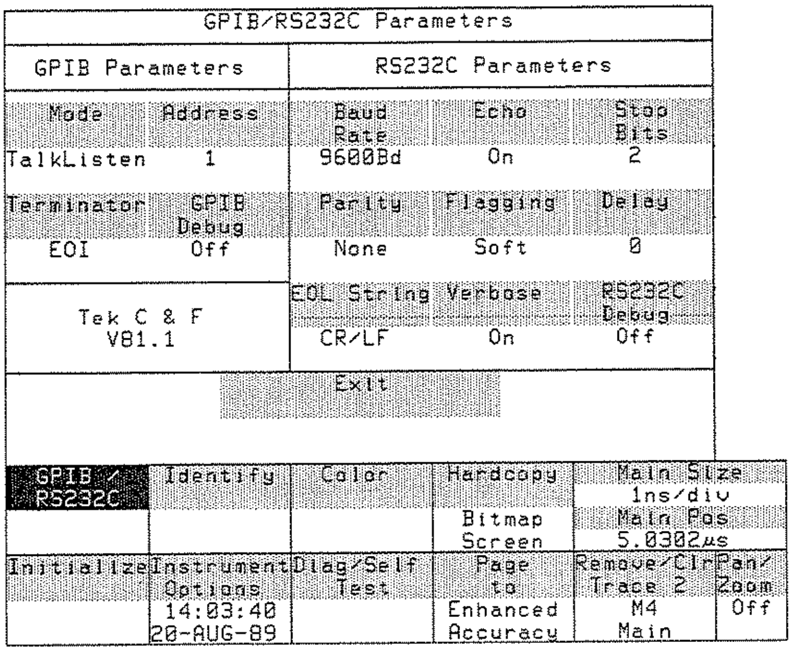
- **Address** sets a unique primary communication address for the instrument. The address range is 0 to 30.
- **Terminator** sets the method of indicating the end of device-dependent messages sent between the controller and the instrument. The choices are EOI (assert EOI line with transmission of last 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 (instrument commands) appear at the top of the instrument 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.



GPIB/RS232C Parameters

GPIB Parameters		RS232C Parameters		
Mode	Address	Baud Rate	Echo	Stop Bits
TalkListen	1	9600Bd	On	2
Terminator	GPIB Debug	Parity	Flagging	Delay
EOI	Off	None	Soft	0
Tek C & F V81.1		EOL String	Verbose	RS232C Debug
		CR/LF	On	Off

Exit

GPIB / RS232C	Identify	Color	Hardcopy Bitmap Screen	Main Size 1ns/div Main Pos 5.0302µs	
Initialize	Instrument Options 14:03:40 20-AUG-89	Diag/Self Test	Page to Enhanced Accuracy	Remove/Clr Trace 2 M4 Main	Pan/Zoom Off

Typical GPIB Settings on the **GPIB/RS232C 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/RS232C Parameters** pop-up menu in the *User Reference* for your instrument.

Connecting the instrument 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 instrument. The remainder of this section tells how to connect and set up the instrument 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 instrument 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 for 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 instrument to another device in any of three common configurations:

- instrument to a 25-pin DTE device (most terminals)
- instrument to a 25-pin DCE device (for example, an IBM PC or compatible with a 25-pin COM port)
- instrument 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 will allow you to connect the devices in these configurations.

RS-232-C Pin Mappings

CSA 803/ 11801A	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 nonstandard 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 an output line of one DCE to an output line of the other. *Disregarding this restriction may damage one or both devices.*
- Ensure that the signal ground of the instrument is connected to the signal ground of the external device.
- Ensure that the chassis ground of the instrument is connected to the chassis ground of the external device.

CSA 803 and 11801A 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.

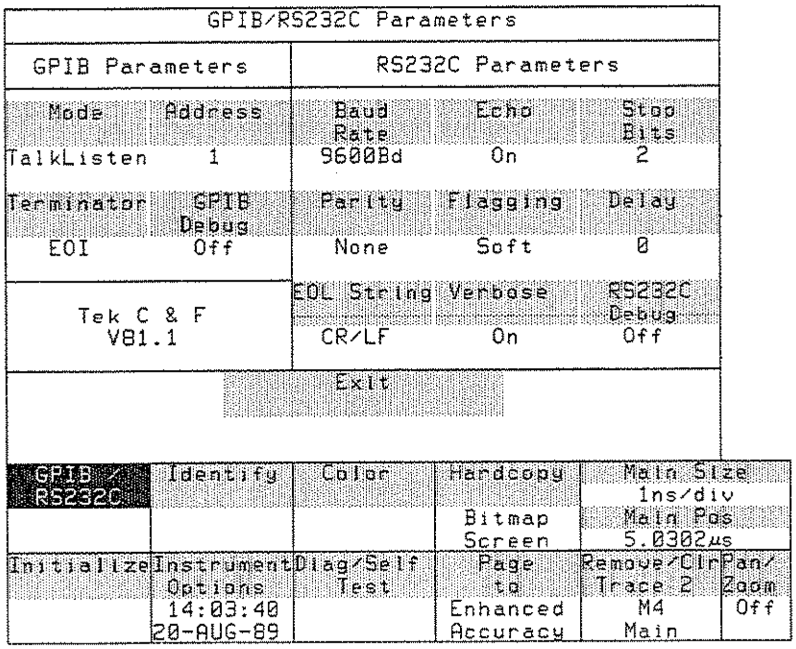
Setting Up RS-232-C Parameters

You can set the parameters of the RS-232-C interface 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.

Use the following steps to set up the RS-232-C parameters at the instrument 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 **GPIB/RS232C** selector in the major menu area. The **GPIB/RS232C Parameters** pop-up menu is now displayed.



GPIB/RS232C Parameters

GPIB Parameters		RS232C Parameters		
Mode TalkListen	Address 1	Baud Rate 9600Bd	Echo On	Stop Bits 2
Terminator EOI	GPIB Debug Off	Parity None	Flagging Soft	Delay 0
Tek C & F V81.1		EOL String CR/LF	Verbose On	RS232C Debug Off
		Exit		

GPIB / RS232C	Identify	Color	Hardcopy Bitmap Screen	Main Size 1ns/div Main Pos 5.0302µs	
Initialize	Instrument Options 14:03:40 20-AUG-89	Diag/Self Test	Page to Enhanced Accuracy	Remove/Clr Trace 2 M4 Main	Pan/Zoom Off

The **GPIB/RS232C Parameters** *Pop-Up Menu*

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.

□ 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:

- **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 instrument 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 instrument 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 instrument (for example, when a BASIC program on a small computer is being used to control the instrument 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 instrument 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 instrument does not use or recognize any flagging.

When flagging is set to hard, the instrument uses the DTR (Data Terminal Ready) and CTS (Clear To Send) lines to control data transmission. On output, the instrument transmits data only when the DTR line is asserted. When the DTR line is not asserted, the instrument stops transmitting data. On input, the instrument 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.

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

When flagging is set to soft, on output the instrument stops transmitting data when it receives an XOFF (DC3) character, and begins transmitting again when it receives an XON (DC1) character. On input, the instrument 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 instrument to respond to a query. The delay range is 0 to 60 seconds, in multiples of 20 milliseconds.

- **Verbose** displays status and event messages on the front panel as commands are executed. When verbose is turned on, each command sent to the instrument 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 instrument to receive the message.

  Turn verbose off when a computer program is transmitting data to the instrument (for example, when a BASIC program on a small computer is controlling the instrument with the RS-232-C interface). The first command your program sends the instrument 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 on, input/output processing is slowed.

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

Command Structure and Conventions

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 *CSA 803 and 11801A Command Reference*.

A large set of commands can be used to control the operations and functions of the instrument 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 instrument using an enhanced American Standard Code for Information Interchange (ASCII) character encoding. The CSA 803 and 11801A support 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 instrument to take a specific action. Queries ask the instrument to return information about its state.

Commands are composed of four syntactic elements:

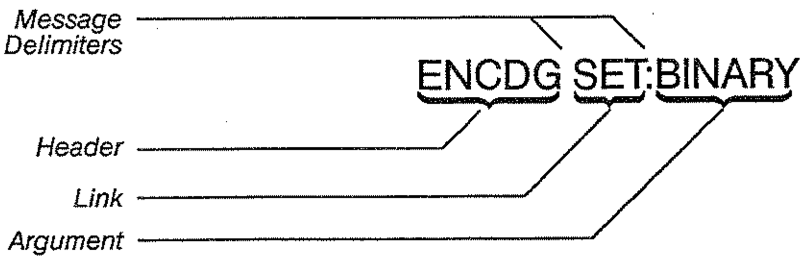
<header> ::= The command name; if it ends with a question mark, the command is a query.

<delimiter> ::= A space, colon (:), comma (,), or semicolon (;) which breaks the message into segments for the instrument 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 instrument 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.

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 instrument to return a measurement, trace data, or a status condition (for example, a current setting or mode). The instrument 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.

```
AVG?
AVG ON
```

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

- 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 trace, channel, or color. These commands are:

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

ADJTRACE[<*ui*>]?

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

COLOR<*ui*>?

TRACE[<*ui*>]?

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

<*ui*> represents a channel number.

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

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

  This form returns the specified links and arguments for the specified channel (<*ui*>) only.

- CH<*alpha*><*ui*>?

  This form returns all links and arguments for the specified channel.

- CH?

  This form returns all links and arguments for all channels.

The ADJTRACE<*ui*>, COLOR<*ui*>, and TRACE<*ui*> commands are similar, in that you can specify the trace number (<*ui*>) or query all traces 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 instrument as a valid command. This makes it easy to save a response from the instrument in a string variable, then send the contents of the string variable back to the instrument as a command when you want to return the instrument 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 instrument as a set command string. Any such attempt always results in a syntax error.

For example, if you sent the following query to the instrument 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 instrument as a set command without generating a syntax error. In such cases, the instrument simply ignores the attempted modification of the query-only link(s).

For example, the response from the CHM1? 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 instrument 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 instrument provides a command (DEFINE) that enables you to create new command names. That is, you can rename an existing 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:

```
DEFINE “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?”
```

The DEF? query command will return a list of all defined strings and their associated command definitions. To query for a list of defined command strings, give this command:

```
DEF?
DEFINE "DATIME?","DATE?;TIME?";DEFINE "E","rs232
echo:on";DEFINE "V","rs232 verbose:on"
```

Command Processing Conventions

Command processing conventions are rules that specify how the instrument 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 instrument, getting "long-form" or "short-form" responses from the instrument, 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 instrument 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

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

The LONGFORM command determines whether the instrument 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"
```

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

Here are the same examples with LONGFORM set to OFF:

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

```
EVENT?
EVENT 250
```

Using Upper and Lower Case

The CSA 803 and 11801A accept both upper and lower case alphabetic input data. Thus, the following two commands are recognized as identical:

```
RQS ON
```

```
rqs on
```

The instrument 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 CSA 803 and 11801A allow 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 **GPIB/RS232C** pop-up menu from the Utility 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 instrument 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/RS232C** pop-up menu from the Utility 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 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 instrument 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 instrument has an opportunity to process the contents, the instrument 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 instrument 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 instrument unconditionally clears its GPIB output buffer (no error is reported). This means that the GPIB output buffer of the instrument 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 instrument unconditionally clears the GPIB output buffer. An execution error is also reported (event code 203, "I/O buffers full").

If the GPIB buffers are empty—and the instrument is 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 instrument over the RS-232-C interface.

This condition may occur when executing a program that expects input from the instrument'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 instrument 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 instrument.

DCL and SDC interface messages do not change instrument 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— the RS-232-C port, note the following points:

- Do not transmit binary block data to the instrument 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 **GPIB/RS232C** from the Utility 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 instrument (for example, when a BASIC program is being used to control the instrument 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 instrument 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 instrument solicits command input with a "&gt;" 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 instrument to return a response for each command sent. When VERBOSE is set to OFF, only valid queries return a response from the instrument.

You can turn verbose on or off from the front panel by selecting **GPIB/RS232C** from the Utility major menu and touching the RS232C Parameters **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 instrument 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 instrument 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 response behavior with VERBOSE mode set to ON.

Examples of Responses with VERBOSE ON

Input Command(s)	**CSA 803/11801A 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 instrument to return responses to the RS-232-C. Valid set commands, invalid set commands, and invalid queries elicit no response event messages from the instrument. 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 response behavior with VERBOSE set to OFF.

Examples of Responses with VERBOSE OFF

Input Command(s)	**CSA 803/11801A 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 instrument is turned on and completes its power-on cycle, the instrument 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 instrument 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 instrument 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 instrument 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 instrument's RS-232-C message processing. BREAK signals do not change instrument 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 instrument 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 CSA 803/11801A RS-232-C interface takes the following actions:

During these I/O error recovery steps (when ECHO is set to OFF), the instrument 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

Command Groups

This section presents the command set arranged by function. The *CSA 803 and 11801A 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 (sampling) of traces
Calibration	Self-calibration functions
Channel	Sampling head vertical parameters
Cursor	Trace cursor selection and positioning
Data Transfer	Transfer of trace data and front panel settings to and from the instrument
Diagnostics	Self-tests and extended diagnostics
Display and Color	Display, Histogram, Mask testing, and color parameters
External I/O	Printer parameters, debug functions, and RS-232-C parameters
Label and Text	Placement of user-defined labels and text
Measurement	Measurement functions
Miscellaneous	System and front panel functions
Status and Event	Instrument event reporting, hardware identification, and configuration information
Time Base	Main and window record length and position
Trace and Settings	Creation and removal of traces and associated front panel settings
Triggering	Triggering parameters

Acquisition

ACQNUM
ACQUISITION
AUTOSET
AVG
CONDACQ
ENV
NAVG
NENV
NGRADED
NHIST.PT
NWAVFRM

Calibration

CALIBRATE<*alpha*><*ui*>
DAFILTERING
DAMEASREF
DCOMP
DIV2
TBCALMODE

Channel

CH<*alpha*><*ui*>
CH?
RHOPOS
CDElay:<*NRx*>

Cursor

CURSOR
DOT1ABS
DOT2ABS
DOT1REL
DOT2REL
H1BAR
H2BAR
RHOZERO
V1BAR
V2BAR

Label and Text

LABABS
LABEL
LABREL
TEXT

Measurement

AMPLITUDE
COMPARE
CROSS
DUTY
EXTINCTION
FALLTIME
FREQ
JITTER
MAX
MEAN
<*meas*>?
MEAS?
MID
MIN
MPARAM
MSLIST
MSNUM?
MSYS
NOISE
OVERSHOOT
PDELAY
REFSET
PERIOD
PHASE
RISETIME
PP
RMS
STATHIST
STATISTICS
TOPBASE
UNDERSHOOT
WIDTH
YTENERGY
YTMNS_AREA
YTPLS_AREA

Miscellaneous/ System

ABSTOUCH
BCOCORRECTION
DATE
DEFINE
DSYMENU?
FPANEL
FPUPDATE
INIT
LONGFORM
POWERON?
SPEAKER
TIME
UNDEF
UPTIME?

Command Groups

Data Transfer

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

Diagnostics

DIAG?
TEST

Display and Color

COLOR
DISPLAY
DSYS?
GRATICULE
HISTOGRAM
MASK
MASKSTAT

External I/O

ALTINKJET
BITMAP
COPY
DEBUG
FEOI
HPGL
PIN8
PIN24
RS232
TEK4692
TEK4696
TEK4697

Status and Event

EVENT?
ID?
RQS
SAMID?
SRQMASK
STBYTE?
UID?

Time Base / Horizontal

MAINPOS
TBMAIN
TBWIN

Trace and Settings

ADJTRACE<*ui*>
CLEAR
DELETE
FPSLIST?
FPSNUM?
HREFPT
MAXTRANUM
NVRAM?
RECALL
REMOVE
SELECT
SETSEQ
STD? *(11801A only)*
STOLIST?
STONUM?
STORE
TRACE<*ui*>
TRACE[<*ui*>]?
TRANUM?
WFMSCALING
WIN
WINLIST
WINNUM

Triggering

TRIGGER

Command Groups (Cont.)

Acquisition Commands

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

Signal Acquisition Commands

Command	Meaning
ACQUISITION	Starts and stops unconditional trace acquisition.
AUTOSET	Adjusts the trace signal for optimal display.
CONDACQ	Controls the condition(s) on which the acquisition of traces stops.

Trace Function Commands

Command	Meaning
AVG	Turns trace averaging on or off. (Averaging can also be defined in the trace description; use the TRACE DESCRIPTION command in the Trace and Settings group.)
AVG.ENV	Turns both trace averaging and enveloping on or off. (Can also be defined in the trace description; use the TRACE DESCRIPTION command in the Trace and Settings group.)
ENV	Turns trace enveloping on or off. (Enveloping can also be defined in the trace description using the TRACE DESCRIPTION command.)

Acquisition Parameter Commands

Command	Meaning
ACQNUM?	Returns the number of acquisition systems.
NAVG	Sets the number of acquisitions to be used in trace averaging.
NENV	Sets the number of acquisitions to be used in trace enveloping.
NGRADED	Sets the number of bins that must overflow on a color graded display to halt conditional acquisition.
NHIST.PT	Sets the number of data points that must be included in a histogram to halt conditional acquisition.
NWAVFRM	Sets the number of trace records that must be processed into histogram, color graded, or masks to halt conditional acquisition.

Calibration Commands

The Calibration commands initiate instrument self-calibration features and report on their condition. Calibration refers to the state of accuracy of the instrument.

System Calibration and Status Commands

Command	Meaning
CALIBRATE	Performs manual or automatic calibration of a specified sampling head channel.
DAFILTERING	Delay Adjust Filtering sets the hardware filtering constant used by the delay adjustment command, CALIBRATE DADJ.
DAMERASREF	Sets the inter-head delay to mid-range for the specified sampling head channel.
DCOMP	Sets continuous strobe delay calibration on or off.
DIV2	Halves the internal calibrator frequency.
TBCALMODE	Sets the time base calibration mode.

Channel/ Vertical Commands

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

Channel/Vertical Commands

Command	Meaning
CH<*alpha*><*ui*>	Sets the vertical parameters for the specified channel.
CH?	Returns the vertical parameters for all channels.
RHOPOS<*ui*>	Sets the rho calconstant on the specified trace.
CDElay:<*NRx*>	Specifies the delay between channels.

Cursor Commands

Cursor commands control the creation and placement of cursors on traces. 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.

Cursor Commands (Cont.)

Command	Meaning
H1BAR H2BAR	Positions the first and second horizontal bar cursors to specified absolute locations.
RHOZERO	Calibrates the rho scale for the selected trace.
V1BAR V2BAR	Positions the first and second horizontal bar cursors to specified absolute locations.

Cursors and the Selected Trace

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

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

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

Cursor Positioning Methods

Dot cursors may be horizontally positioned by graticule divisions, percentage of the trace record length, or horizontal units of the selected trace. 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 trace.

When cursors are positioned or queried by graticule divisions, use the illustration shown in the DOT1ABS command entry in the *CSA 803 and 11801A Command Reference* to interpret the dimensions of single and dual graticules.

Data Transfer Commands

Data transfer commands transfer trace information and front panel settings to and from the instrument 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 trace data. Scaling information is included in the trace preamble.
SET?	Returns the front panel settings to the controller.
VPCURVE?	Transfers all trace records used in a variable persistence trace.
WAVFRM?	Returns the trace preamble and trace data points for the trace specified by OUTPUT. WAVFRM? is equivalent to entering WFMPRE?;CURVE?
WFMPRE	Contains the links of the trace 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 trace transferred to the instrument.
OUTPUT	Selects the stored or displayed trace to be transferred from the instrument.

Retrieving Trace Data From the Instrument

In general, the controller retrieves trace data from the instrument in the following sequence:

1. Use the ENCDG command to select trace 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 traces that are to be retrieved.

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

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

Sending Trace Data to the Instrument

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

In general, trace data are sent back to the instrument as a stored trace in the following sequence:

1. If the trace 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 trace location where the data will be written.

3. Use the WFMPRE command to return the trace preamble to the instrument.

4. Finally, use the CURVE command to return unscaled trace data to the instrument.

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.

TEST destroys all stored traces and user-defined expansion strings created with the DEFINE command, resets the TEXT X:, Y: coordinates to 0,0, and removes user-entered text from the display.

Display and Color Commands

Display commands control how traces appear on the display, Histogram analysis, and Mask testing. 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 for traces and other display features.
DISPLAY	Controls display intensity, number of graticules, dot/vector mode, and transfers of color graded pixel data.
DSYS	Selects the Display Modes major menu to enable Histograms and Mask testing.
GRATICULE	Sets X and Y axis units and scaling.
HISTOGRAM	Controls vertical and horizontal histogram displays for active traces.
MASK	Creates mask polygons for trace testing.
MASKSTAT	Starts Mask test of traces and returns Mask statistics.
MASKSTAT	Starts Mask test of traces and returns Mask statistics.

External I/O Commands

The External I/O commands produce hardcopy printout of the display, set the parameters for the RS232 interface, and control the debug feature for the GPIB and RS-232-C interfaces. 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.
FEOI	Forces a message terminator in a command string.
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.
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 traces 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 trace to an absolute location.
LABEL	Defines and deletes labels, and controls whether they are displayed.
LABREL	Positions the label associated with the selected trace 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 trace measurements and control the parameters with which these measurements are taken. In this discussion, the measurement system features are briefly presented, then the measurement commands are defined, followed by measurement parameter commands, and finally measurement execution commands. For a detailed discussion of the measurement system, refer to the *User Reference* for your instrument.

Selecting Measurements

Measurements initiated from the GPIB/RS-232-C interface operate differently than front panel measurements. From the ASCII interface, measurements can be specified for any trace or for a range of traces.

Each displayed trace has a set of user selected measurements and a set of user controlled measurement parameters that define how the measurements will be taken. Select measurements for a trace with the MSLIST command. The selected measurements are taken continuously while the Measure major menu is selected (MSYS command). Use the MEAS<*ui*> command to return the measurement results for all trace measurements. Or, on the selected trace only, you can query individual measurements with *<meas>?* command (e.g., RISE?, CROSS?). Individual measurement queries are allowed regardless of the major menu selected and they need not be specified with MSLIST.

Measurement Modes: Hardware, Software, and Software Statistics

The CSA 803 and 11801A support software, hardware, and software statistical modes for taking measurements. The MPARAM MMODE command selects the mode.

The software mode (SW) calculates measurements from the trace record data. All measurements can be performed in the software mode. This is the default measurement mode.

The hardware mode (HW) uses special hardware timer circuits to perform timing measurements on samples before they are digitized and assembled into a trace record. The HW mode provides measurement results much quicker than the SW mode. Only timing measurements may be taken in the hardware mode.

The measurements CROSS, FALLTIME, FREQ, PDELAY, PERIOD, RISETIME, and WIDTH can be taken in the hardware mode. The other measurements may be taken in the software mode only.

The Software Statistical Mode (STAT) calculates trace measurement parameters using histograms.

Measurement Statistics

The CSA 803 and 11801A can collect statistics on a specified measurement taken on the selected trace. Select a measurement and the sample size (number of measurement samples to include) with the STATISTICS command. Use the STAT? command to return the statistical mean, standard deviation, and the number of samples used. Measurement statistics are available for software and hardware measurements only.

Comparing Measurement With References

The COMPARE command lets you compare a measurement to a reference value. The REFSET command sets the reference value to either the current value for each active measurement or to some absolute value for a single measurement <*meas*>.

Measurement (<*meas*>) Commands

No measurements can be taken on XY traces.

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

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**
AMPLITUDE	The amplitude (topline - baseline) of the selected trace. SW or STAT.
CROSS	The time from the trigger point to a specified reference level crossing. HW or SW.
DUTY	Percentage of a period that a trace spends above the MESIAL level. SW or STAT.
EXTINCTION	The extinction ratio (topline ÷ baseline) of the selected trace. SW or STAT.
FALLTIME	The transition time of a falling pulse edge, from the distal to proximal levels. HW, SW, or STAT.
FREQ	Frequency (reciprocal of the period measurement). HW, SW, or STAT.
JITTER	The jitter of the trace, calculated at the location (mesial or eye cross) that you specify with the MPARAM JITTLO command. STAT only.
PDELAY	Propagation delay between mesial crossings on two traces (used with REFTRACE). HW, SW, or STAT.
PERIOD	The time between the first and next mesial crossing of the same slope. HW, SW, or STAT.
PHASE	The phase relationship of the reference trace to the selected trace, expressed as a value from -360 to +360 degrees. SW or STAT.
RISETIME	The transition time of a rising pulse edge, from the proximal to distal levels. HW, SW, or STAT.
WIDTH	The time between the first and next mesial crossing of the opposite slope. HW, SW, or STAT.

Amplitude Measurements

<meas>	**Meaning**
MAX	Maximum amplitude (most positive peak voltage). SW or STAT.
MEAN	Average amplitude (arithmetic mean voltage). SW or STAT.
MID	Amplitude midpoint, halfway between the maximum amplitude and the minimum amplitude. SW or STAT.
MIN	Minimum amplitude (most negative peak voltage). SW or STAT.
NOISE	The noise of the trace, calculated halfway between the left and right crossings, or halfway between the left and right measurement zones. STAT only.
OVERSHOOT	Difference between the maximum amplitude and the topline value, given as a percentage of the difference between the topline and baseline values. SW or STAT.
PP	Peak-to-peak value; the voltage difference between the maximum amplitude and minimum amplitude. SW or STAT.
RMS	True root-mean-square voltage. SW or STAT.
UNDERSHOOT	Difference between the baseline value and the minimum amplitude, given as a percentage of the difference between the topline and baseline values. SW or STAT.

Area/Energy Measurements

<meas>	**Meaning**
YTENERGY	The energy represented under the curve of a YT trace (this integral of the squared voltages can be divided by the resistance of the circuit to yield power measurements). SW or STAT.
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. SW or STAT.
YTPLS_AREA	The total, absolute value of all areas between a YT trace and a user-specified reference level. SW or STAT.

Measurement Parameter Group

Measurement parameters are set with the MPARAM command and the appropriate link for a specified trace. Some measurement parameters apply just to software measurements and some apply to hardware measurements while others apply to both measurement modes.The following MPARAM command links set the measurement parameters.

Measurement Parameter (MPARAM) Links

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.
DISPERSION	Sets whether the results of the queries JITTER? and NOISE? are displayed as peak-to-peak or RMS deviation (Standard Deviation) values.
DISTAL	Sets the distal (farthest from origin) reference level, typically 90% of the baseline-to-topline value.

Measurement Parameter (MPARAM) Links (Cont.)

Command	Meaning
JITTLEVEL?	Valid only in MMODE STAT. Returns the jitter measurement level. Updates only if JITTER is in the trace's measurement list and the trace is selected or a MEAS? query is sent.
JITTLOCATION	Sets jitter measurement location on the trace. On an eye diagram JITTLOCATION can be measured at the Eye cross (CROSS) or at the mesial (MESIAL).
JITT.HISTPT?	Valid only in MMODE STAT. Returns the number of histogram points used to calculate the jitter on the trace. Updates only if JITTER is in the trace's measurement list and the trace is selected or a MEAS? query is sent.
LMZONE	Sets the left limit of the measurement zone.
MESIAL	Sets the mesial (middle) reference level (the end point of the trace period), typically 50% of the baseline-to-topline value.
MFILTERING	Sets the filtering constant for hardware measurements.
MLEVELMODE	Determines whether certain measurement parameters are absolute voltage values or percentages of the baseline-to-topline value.
MMODE	Selects the hardware, software, or statistical software measurement mode.
MSLOPE	Sets the crossing slope for measurements.
MTRACK	Controls measurement tracking (whether you or the instrument set the baseline and topline values).
MTRANS	Sets the transition number on which hardware measurements begin.
NOISLOCATION	Sets whether the noise is measured around the trace's topline or baseline.

Measurement Parameter (MPARAM) Links (Cont.)

Command	Meaning
NOIS.HISTPT?	Valid only in MMODE STAT. Returns the number of histogram points used to calculate the noise on the trace. Updates only if NOISE is in the trace's measurement list and the trace is selected or a MEAS? qeuery is sent.
NOISLOCATION	Sets whether noise is measured around the trace's topline or baseline.
PROXIMAL	Sets the proximal (nearest to origin) reference level, typically 10% of the baseline-to-topline value.
REFBASELINE	Sets the absolute value of the baseline on the reference trace when measurement tracking (MTRACK) is off.
REFFILTERING	Sets the filtering constant on the reference trace for hardware measurements.
REFLEVEL	Sets the reference level used in CROSS and other measurements.
REFSET	Sets the reference value(s) used in comparison mode (COMPARE is set to ON).
REFLMZONE	Sets the left limit of the measurement zone on the reference trace.
REFMESIAL	Sets the mesial (middle) reference level on the reference trace.
REFRMZONE	Sets the right limit of the measurement zone on the reference trace.
REFTRACE	Sets the reference trace used by the PDELAY and PHASE measurement.
REFSLOPE	Sets the crossing slope on the reference trace.
REFSNRATIO	Sets the amplitude of a noise rejection centered on the MESIAL level of the reference trace.
REFTOPLINE	Sets the absolute value of topline on the reference trace when Tracking (MTRACK) is off.

Measurement Parameter (MPARAM) Links (Cont.)

Command	Meaning
REFTRACE	Selects the reference trace which provides the endpoint for some measurements.
REFXSITION	Sets the transition number on the reference trace that will end hardware measurements.
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 off.

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

Measurement Parameter Interactions (Cont.)

Parameter(s)	Effects
BASELINE TOPLINE	Used to calculate range values for PROXIMAL, MESIAL, and DISTAL, when MTRACK is set to OFF and MLEVELMODE is RELATIVE.
DAINT	Limits the measurement zone for MEAN, RMS, YTENERGY, YTMNS_AREA, and YTPLS_AREA.
DISPERSION	PP or RMSDEV set whether the results of JITTER? and NOISE? queries are displayed as peak-to-peak or RMS deviation (Standard Deviation) values.
DISTAL PROXIMAL	Sets the measurement end points (voltage levels) for RISETIME and FALLTIME <*meas*>.
JITTLOCATION	CROSS or MESIAL set jitter measurement location on the trace. On an eye diagram JITTLOCATION can be measured at the Eye cross or at the mesial.
LMZONE RMZONE	Establishes a measurement zone to isolate part of a trace.
MESIAL	Sets measurement endpoints for DUTY, FREQ, PDELAY, PERIOD, PHASE, and WIDTH <*meas*>; and when DAINT is set to SINGLE, for MEAN, RMS, YTENERGY, YTMNS_AREA and YTPLS_AREA.
MLEVELMODE	Determines whether range values for PROXIMAL, MESIAL, DISTAL, REFMESIAL, and REFLEVEL are interpreted as percentages or as absolute numbers.
MMODE	Selects hardware, software, or software statistical measurement modes. Only timing measurements can be taken in the hardware mode.

Measurement Parameter Interactions

Parameter(s)	Effects
MTRACK	Determines whether BASELINE and TOPLINE are set dynamically by the instrument using measurement tracking or are set by you.
MTRANS, REFXSITION	Selects the signal transition number on which to begin and end the PDELAY measurement. The transitions are defined by slope, mesial level, and filtering constant.
NOISLOCATION	TOPLINE or BASELINE set whether the noise is measured around the trace's topline or baseline.
REFLEVEL	Sets the crossing level for CROSS and a comparison level for YTMNS_AREA <*meas*>. MMODE and MLEVELMODE affect the range.
REFTRACE	Selects the reference trace used with PDELAY, and PHASE <*meas*>. The reference trace has a set of parameters that control the endpoint of some timing measurements (e.g., REFBASELINE, REFFILTERING, REFLEVEL, REFSLOPE, etc.).
SNRATIO	Noise filter when MMODE set to SW for DUTY, FREQ, PDELAY, PERIOD, PHASE, and WIDTH ; and when DAINT is set to SINGLE, for MEAN, RMS, YTENERGY, YTMNS_AREA and YTPLS_AREA.

Measurement Execution Commands

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

Measurement Execution Commands

Command	Meaning
AMPLITUDE?	When MMODE is set to SOFT or STAT, return the amplitude of the trace (topline - baseline).
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.
EXTINCTION?	When MMODE is set to SOFT or STAT, returns the extinction ratio of the trace (topline ÷ baseline).
JITTER?	When MMODE is set to STAT, returns the jitter calculated at the location JITTLOCATION on the trace.
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*>) taken on the selected trace.
MSLIST	Selects the measurements (<*meas*>) for a trace 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 which must be selected to collect measurement statistics.
NOISE?	When MMODE is set to STAT, returns the noise calculated half way between the left and right crossings, or half way between the left and right measurement zones.

Measurement Execution Commands (Cont.)

Command	Meaning
REFSET	Sets reference values on the specified trace for comparison measurements with the COMPARE command.
STAT?	Returns measurement statistics (mean and standard deviation) for the <*meas*> specified with STATISTICS.
STATHist	Returns statistical information created by the HISTOGRAM function.
STATISTICS	Selects measurements for statistics calculations.
TOPBASE	Selects the method of determining topline and baseline.

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.
DSYMENU?	Returns which major menu is currently displayed.
FPANEL	Controls front panel lockout.
FPUPDATE	Controls when front panel readouts are updated.
SPEAKER	Controls whether the instrument beeps when the display is touched.

System Commands

Command	Meaning
DATE	Sets the date on the system calendar.
DEFINE	Defines logical names for command strings.
INIT	Initializes the system.
LONGFORM	Controls whether the instrument returns full or abbreviated query responses and event information.
POWERON?	Returns the number of times the instrument has been powered on.
TIME	Sets the time on the system clock.
UNDEF	Deletes logical names previously defined with DEF.
UPTIME?	Returns the number of hours the instrument has been powered on.

Status Queries and Event Commands

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

Status Queries

Command	Meaning
ID?	Returns version numbers of system firmware.
UID?	Returns serial numbers of the instrument and its sampling heads.

Event Commands

Command	Meaning
EVENT?	Returns event code information.
NMASK	Sets the number of mask hits that must be acquired to stop conditional acquisition.
RQS	Sets whether the instrument asserts the SRQ line after an event occurs (GPIB only).
SAMID?	Returns the model number of sampling heads.
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
MAINPOS	Sets the horizontal position of the main trace record with respect to the main trigger.
TBMAIN	Sets the main horizontal (time base) parameters.
TBWIN	Sets the window horizontal (time base) parameters.

Trace and Settings Commands

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

Trace Commands

Command	Meaning
ADJTRACE<*ui*>	Controls pan/zoom mode, vertical size and position, and window trace separation.
CLEAR	Discards acquired data for displayed traces.
DELETE	Deletes stored traces or front panel settings.
DISPNUM?	Returns the number of traces displayed.
HREFPT	Sets the horizontal pivot point about which a trace is expanded.
MAXTRANUM?	Returns the highest trace number possible with the current configuration.
REMOVE	Discards displayed traces and their descriptions.
SELECT	Designates the selected trace.
STD[<*ui*>]?	*11801A only.* Returns the saved trace description specified by <*ui*>. When <*ui*> is omitted, returns all saved trace descriptions.
STOLIST?	Returns a list of all stored traces.
STONUM?	Returns the number of stored traces.
STORE	Copies displayed traces to memory.
TRACE<*ui*>	Defines a trace and its characteristics.
TRANUM?	Returns the number of displayed traces.
TRALIST?	Returns a list of all displayed traces.
WFMSCALING	Sets a scaling flag to create new traces in fast (integer) or high-precision (floating-point) mode.
WIN	Sets a variety of window trace parameters including window position and autowindow placement.

Trace Commands

Command	Meaning
WINLIST?	Returns the numbers of displayed window traces.
WINNUM?	Returns the total number of defined window traces.

Front Panel Settings Commands

Command	Meaning
DELETE	Deletes stored traces 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. The Trigger commands are:

Triggering Commands

Command	Meaning
TRIGGER	Sets trigger parameters such as source, level, mode, and slope and returns its current status.

Learning by Example

Learning By Example

The nine examples in this section demonstrate how to perform typical CSA 803 and 11801A functions from the programmable interfaces.

The example programs are divided into sections similar to the front panel operation examples in the *Tutorial* for your instrument.

These examples show how to control basic functions remotely

1—Displaying a Trace

2—Managing Multiple Traces

3—Defining Complex Traces

4—Using Signal Processing & Transferring a Trace

These examples use the automated measurement system

5—Taking Automated Measurements

6—Taking Delay Measurements

7—Taking ΔDelay Measurements From a Reference Trace

These examples introduce some advanced features

8—Comparing Displayed Traces to Stored Traces

9—Taking TDR Measurements

Introduction to the Examples

These examples show you how to use the BASIC programming language and the CSA 803 and 11801A command set to perform a variety of remote operations over the RS-232-C interface. The examples require you to have a basic knowledge of your instrument from the front panel, of test and measurement systems, and of programming in the BASIC language.

The Learning by Example programs will familiarize you with the process of controlling the instrument remotely, and enable you to extend this knowledge to your applications.

For instance, in Example 1: Displaying a Trace, the entire program is explained, line by line, so you can see how the CSA 803/11801A commands combine with the controller's BASIC programming language to perform tasks within an application program.

Organization of the Examples

Each example begins with a brief explanation of its purpose and the setup of necessary equipment and accessories (e.g., signal generator and cable configuration). Then each program is listed and explained. You are expected to read the program descriptions as you use the example programs from the disk that is supplied.

Since this is an interactive tutorial, each step of an example prompts you to take an action that performs some task. In most cases, the action is implemented when you press the 'Enter' key on your computer keyboard.

Software and Computer Hardware Needed

The tutorial examples are written for an IBM PC/XT/AT, Tektronix PEP301, or other IBM PC-compatible computer configured with an RS-232-C interface.

The example programs are supplied on a single, IBM-formatted, $5^1/_4$ inch floppy disk at the front of this manual. They run under most common BASIC language implementations, including:

- IBM BASICA.COM
- IBM Compiled BASIC, Ver. 1.0 and 2.0
- Microsoft QuickBASIC, Ver. 1.0 to 4.0 or Compiled 6.0
- IBM BASIC.COM

NOTE: *Example 4 will not work properly with IBM BASIC.COM because it does not support the WINDOW function used to plot transferred traces. Most other BASIC implementations support the WINDOW plotting function.*

Equipment Needed

For these examples you will need a CSA 803 or 11801A with a two channel sampling head installed. An SD-24 is required for Example 9: Taking TDR Measurements. Each example begins with an equipment setup diagram that shows the cable connections for that example.

Additional equipment needed includes a signal generator (1 to 50 MHz sine wave, 0.5 V to 1 V output) with a separate trigger output, a power divider, four SMA cables (a 2 ns and a 5 ns cable are recommended), and adapters to connect the SMA cables to the signal generator. A signal-splitting T adaptor may be substituted for the power divider.

Installing the Example Software

To complete the following installation, you will need either a hard disk or a formatted floppy diskette to make a working copy of the example programs. Whenever an example command is shown, press the 'Enter' key to execute the command.

Boot up your computer (using its DOS system disk), carefully install a two channel sampling head in the leftmost *sampling head* compartment. Then power-on the CSA 803/11801A.

Configure the RS-232-C interfaces on the CSA 803/11801A and on your computer.

- To configure the CSA 803 or 11801A, press the UTILITY major menu button, then touch the GPIB/RS232C Parameters menu selector. Set the RS-232-C parameters as follows:

Baud	9600	Echo	Off
Stop Bits	1	Parity	NONE
Flagging	HARD	Delay	0
EOL String	CR/LF	Verbose	Off

- Consult the manual for your computer for information on configuring its RS-232-C interface.

Hard disk Installation

If your computer has a hard disk, proceed as follows:

- Create a new directory to receive the example programs, using the MKDIR command from MS-DOS.

```
mkdir examples
```

- Make the new directory the current directory, using the CD command.

```
cd examples
```

- To copy the contents of the supplied examples disk to the new directory, place the disk into drive A of your computer. Then type:

```
copy a:*.*
```

This completes the software installation.

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:.

- Type the following command:

```
diskcopy a: b:
```

If your computer has a single floppy disk drive, proceed as follows:

- Type the following command:

```
diskcopy a: b:
```

- Follow the instructions that appear on the screen in response to the DISKCOPY command, being very careful to insert the supplied tutorial disk when the SOURCE diskette is requested, and to insert a formatted floppy disk when the DESTINATION diskette is requested.

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 the name of your BASIC program file).

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

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 and example programs reside, for example,

```
cd examples
```

Then, enter:

```
basica menu.bas /C:5000
```

or use QB, or the correct invocation for your BASIC application.

Menu.bas is the menu program used to select the examples.The /C: 5000 argument sets up a COM buffer of 5000 bytes for use in

Example 4: Using Signal Processing & Transferring a Waveform. As an alternative, you can invoke the MENU program with a batch file such as the HELP batch file on the Learning By Example disk. Entering

```
help
```

will automatically invoke the MENU program using BASICA MENU.BAS /C:5000. You can modify this batch file with your text editor to use the name of your BASIC application.

The MENU program displays a list of example 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. Then, type the name of the example program, followed by *<ENTER>*, for example,

```
basica single.bas
```

or QB SINGLE.BAS, or the correct invocation for your BASIC application.

Exiting the Learning by Example Software

When an example 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.

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>.

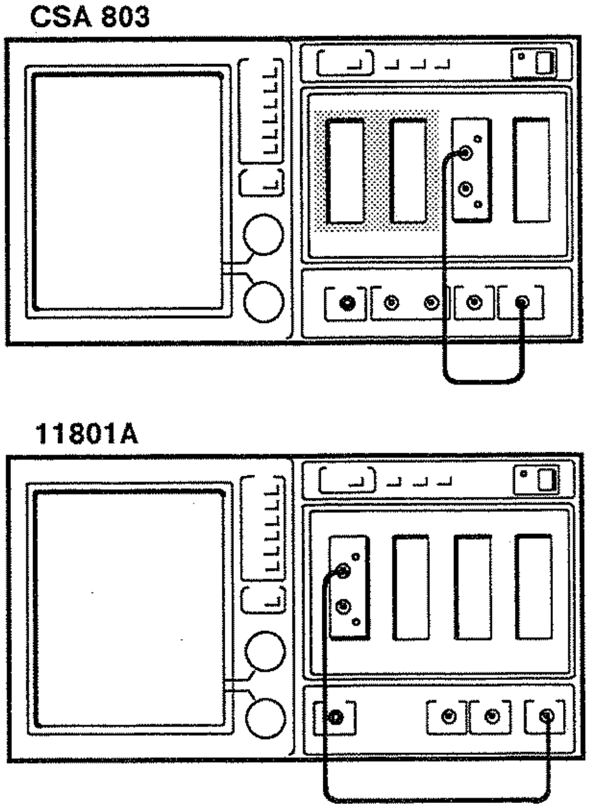
MENU.BAS Program Listing

```
100 CLS
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 & Transferring a
Waveform"
180 PRINT "   5) Taking Automated Measurements"
190 PRINT "   6) Taking Delay Measurements"
195 PRINT "   7) Taking Delta-Delay Measurements From a
Reference Trace"
200 PRINT "   8) Comparing Displayed Traces to Stored
Waveforms"
207 PRINT "   9) Taking TDR Measurements"
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>9 THEN GOTO 220
300 ON PROGNUM GOTO 310,320,330,340,350,360,370,380,390
310 LOAD "EX1.BAS",R
320 LOAD "EX2.BAS",R
330 LOAD "EX3.BAS",R
340 LOAD "EX4.BAS",R
350 LOAD "EX5.BAS",R
360 LOAD "EX6.BAS",R
370 LOAD "EX7.BAS",R
380 LOAD "EX8.BAS",R
390 LOAD "EX9.BAS",R
```

Example 1: Displaying a Trace

This example shows how you can quickly display a meaningful trace. You will also become familiar with basic oscilloscope control.

For this example you will need a CSA 803 or 11801A with at least one sampling head installed and one SMA connecting cable.



Connections for Example 1

To start the program, select example 1 from the Examples Menu. Refer to the earlier discussion Running the Learning by Example Software for information on using the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7

102 PRINT "Example 1: Displaying a Trace"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope´s parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$

160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180    INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190    IF A$="" THEN A$="COM1"
200 WEND

210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1

230 INPUT "Press Enter to ask for the scope´s ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$

270 INPUT "Press Enter after connecting the CALIBRATOR to channel 1
(head 1)",A$

280 INPUT "Press Enter to initialize scope.",A$
290 PRINT #1,"INIT"

300 A$=""
310 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
320 INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
330 WEND

340 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"
```

(cont.)

Program Description

Line 100 defines your computer cursor type and locations. In this case, the first two commas mean that the *x* and *y* coordinate arguments are not specified, 1 means the cursor is turned on, 0 places the cursor at the top of the cell to start, and 7 places the cursor at the bottom of the cell to finish.

Lines 102 through 150 identify the example and remind you to set up the CSA 803/11801A RS-232-C communication parameters appropriately.

Lines 160 through 200 define your communication path to be an RS-232-C communication port (COM1 or COM2).

Lines 210 and 220 alert you that your computer is being set up to use the RS-232-C communication parameters: 9600 baud, no parity, 8-bit word length, and one stop bit.

> **Note:** If this program crashes here, you should check that your RS-232-C cable is wired correctly, or that you have selected the correct RS-232-C port on your computer (preferably COM1).

Line 230 prompts you to initiate a identification query and line 240 actually sends the ID? command to the instrument.

Lines 250 and 260 read the ID? response message and print it on your computer screen.

Line 270 reminds you to make the necessary hardware setup for this example.

Line 280 prompts you to initialize the instrument and line 290 actually sends the INIT command to the instrument.

Line 300 initializes your forthcoming debug response to null.

Lines 310 through 330 waits for you to type Y or N to enable or disable DEBUG mode. Turning DEBUG on enables you to view the incoming commands in the top two lines of the front-panel display.

Line 340 sends the command to turn DEBUG on, if you respond with an upper or lower case **Y** character.

(cont.)

Program Listing (cont.)

```
350 INPUT "Press Enter to setup trace1 from sampling head 1, channel 1.",A$
360 PRINT #1,"TRACE1 DESCRIPTION:´M1´"

370 INPUT "Press Enter to set the trigger to INTERNAL.",A$
380 PRINT #1,"TRIGGER SOURCE:INTERNAL"

390 INPUT "Press Enter to manually set the size and position of the trace.",A$
400 PRINT #1,"TBMAIN TIME:10E-9;MAINPOS 55E-9;CHM1 SENSI:.1,OFFSET:0"

410 INPUT "Press Enter to send ´AUTOSET´ to the scope",A$
420 PRINT #1,"AUTOSET START"

430 PRINT "Example Completed."
435 CLOSE 1
440 LOAD "MENU.BAS",R
```

Program Description (cont.)

Line 350 waits for you to press 'Enter' to initiate the definition of Trace1 and line 360 sends the command to define Trace1 to be from Channel 1 of the installed sampling head.

Line 370 prompts you to select internal triggering for this example and line 380 sends the command to set the trigger source to internal.

Line 390 prompts you to set the exact size and position of Trace1 using time base and channel commands. Line 400 sends the commands to set the horizontal time per division of the main time base, the position of the main trace, and the vertical sensitivity and position of Channel 1.

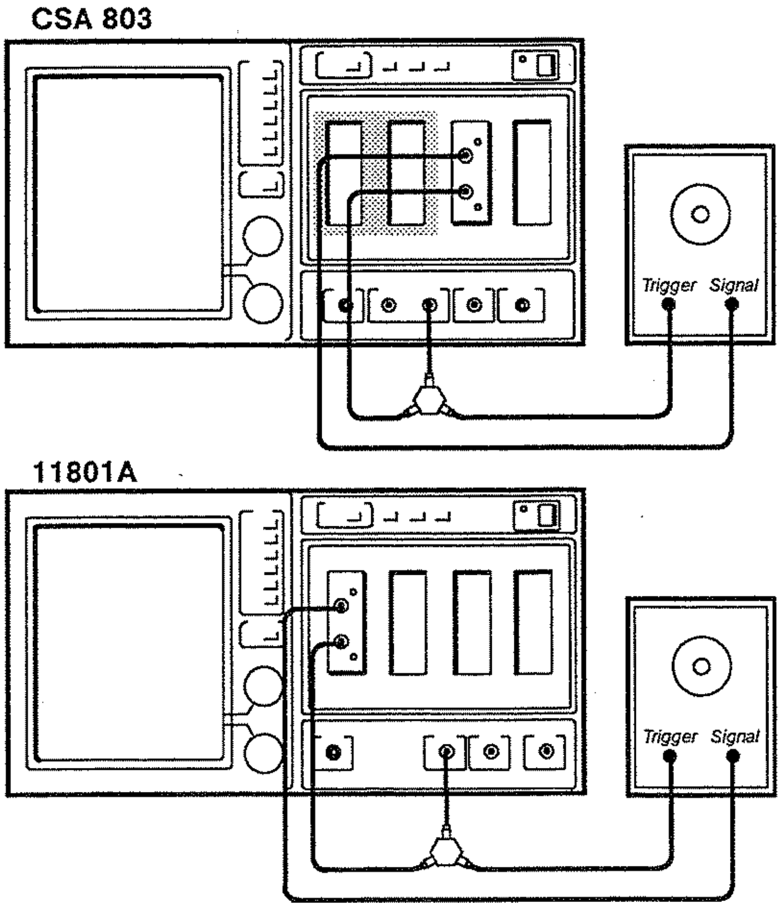
Line 410 waits for you to press 'Enter' to initiate AUTOSET of the selected trace. This automatically sets the size and position parameters. Note that if triggering were set to external source, AUTOSET would also adjust the triggering parameters to produce a stable trace display.
Line 420 sends the command to set the vertical and horizontal parameters automatically.

Lines 430 and 435 prompt you that this example is finished and formally close the program. Line 440 returns you to the Examples Menu.

Example 2: Managing Multiple Traces

This example demonstrates multiple traces and graticules on the display. You will also become familiar with trace selection and management.

For this example you will need a CSA 803 or 11801A with at least one sampling head installed, three SMA connecting cables, a signal generator that provides a separate trigger output, a power divider, and adapters to fit the SMA cables to the signal generator. A signal-splitting T adaptor may be substituted for the power divider.



Connections for Example 2

To start the program, select example 2 from the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 2: Managing Multiple Traces"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope´s parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$
160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180    INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190    IF A$="" THEN A$="COM1"
200 WEND
210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1
230 INPUT "Press Enter to ask for the scope´s ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$

270 PRINT "Press Enter after making the following connections :"
280 PRINT " Connect a 500kHz to 5MHz signal (.5V to 1V PP) to channel 1
(head 1)."
290 PRINT " Connect trigger out of your signal source to the TRIGGER INPUT of
the scope."
300 PRINT " Use a power splitter or a power ´T´ connecter to connect your
trigger out"
310 INPUT "  to channel 2 (head 1)",A$

320 INPUT "Press Enter to initialize scope.",A$
330 PRINT #1,"INIT"

340 A$=""
350 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
360    INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
370 WEND
380 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"

390 INPUT "Press Enter to setup trace1 from sampling head 1, channel 1.",A$
400 PRINT #1,"TRACE1 DESCRIPTION:´M1´"

410 INPUT "Press Enter to set the trigger to EXTERNAL",A$
420 PRINT #1,"TRIGGER SOURCE:EXTERNAL"
```

(cont.)

Program Description

Lines 100 through 260 of Example 2 describe the same type of preparatory procedure as described in Example 1, program lines 100 through 260.

Lines 270 through 310 of this example remind you to make the necessary hardware setup for this example.

Lines 320 prompts you to initialize the instrument and line 330 sends the INIT command.

Lines 340 through 370 wait for you to type Y or N to enable or disable DEBUG mode.

Line 380 displays the selected DEBUG mode.

Line 390 waits for you to press 'Enter' to define Trace1 and line 400 sends the command to define Trace1 as Channel 1 of the sampling head.

Line 410 prompts you to select external triggering for this example and line 420 sends the command to set the triggering source to external.

(cont.)

Program Listing (cont.)

```
430 INPUT "Press Enter to manually set the size and position of the trace.",A$
440 PRINT #1,"TBMAIN TIME:2E-6;MAINPOS 55E-9;CHM1 SENSI:.2,OFFSET:-1"

450 INPUT "Press Enter to send ´AUTOSET´ to the trace in period mode.",A$
460 PRINT #1,"AUTOSET MODE:PERIOD,START"

470 INPUT "Press Enter to send ´AUTOSET´ in the edge mode",A$
480 PRINT #1,"AUTOSET MODE:EDGE,START"

490 INPUT "Press Enter to setup trace2 from sampling head 1, channel 2.",A$
500 PRINT #1,"TRACE2 DESCRIPTION:´M2´;AUTOSET HORIZ:OFF,START,HORIZ:ON"

510 INPUT "Press Enter to select trace1 as the highlighted trace.",A$
520 PRINT #1,"SELECT TRACE1"

530 PRINT "Now we will ask for the trace count and their numbers."
540 INPUT "Also, we will ask for their descriptions. (Press Enter)",A$
550 PRINT #1,"TRANUM?;TRALIST?"
560 LINE INPUT #1,RESPONSE$
570 PRINT RESPONSE$
580 PRINT #1,"TRA1? DESCR;TRA2? DESCR"
590 LINE INPUT #1,RESPONSE$
600 PRINT RESPONSE$

610 INPUT "Press Enter to create the second graticule",A$
620 PRINT #1,"DISP GRAT:DUAL"

630 INPUT "Press Enter to move trace1 to the upper graticule",A$
640 PRINT #1,"ADJTRACE1 GRLOC:UPPER"

650 INPUT "Press Enter to reduce back to the single graticule",A$
660 PRINT #1,"DISP GRAT:SINGLE"

670 INPUT "Press Enter to remove TRACE1 and TRACE2.",A$
680 PRINT #1,"REMOVE TRACE1,TRACE2"

690 PRINT "Example Completed."
695 CLOSE 1
700 LOAD "MENU.BAS",R
```

Program Description (cont.)

Line 430 prompts you to set the exact size and position of Trace1 (set values are in line 440) using timebase and channel commands.

Line 440 sends the commands to set the horizontal time per division of the main time base, the position of the main trace, and the vertical sensitivity and position of Channel 1. You can compare the results of this setting with the AUTOSET results from lines 450 and 460.

Line 450 prompts you to AUTOSET the selected trace in period mode. This is the common mode to automatically size, position, and trigger a trace when you're interested in its periodic characteristics. Note that since external trigger sourcing is selected, AUTOSET includes adjusting trigger parameters.

Line 460 sends the command to automatically set the trace size, position, and trigger parameters using PERIOD mode of the AUTOSET function.

Line 470 prompts you to AUTOSET the selected trace in EDGE mode. This is the common mode to automatically size, position, and trigger a trace when you're interested in the characteristics of the rising or falling edge.

Line 480 sends the command to automatically set the size, position, and trigger parameters using EDGE mode of the AUTOSET function.

Line 490 waits for you to press 'Enter' to define Trace2. Line 500 sends the command to define Trace2 to be from Channel 2 of the sampling head. It also sends the commands necessary to autoset the selected trace, (Trace2), and to keep Trace1 and Trace2 at the same horizontal size and position settings.

Line 510 prompts you to select Trace1, thus highlighting it and line 520 sends the command to select Trace1.

Lines 530 and 600 identify the number of defined traces, their assigned trace numbers (this may include previously stored traces) and their trace source descriptions.

Line 610 prompts you to create a second graticule and line 620 sends the command to create it.

Line 630 prompts you to move Trace1 to the upper graticule and line 640 sends the command to do this.

Line 650 prompts you to go back to a single graticule and line 660 sends the command to do this.

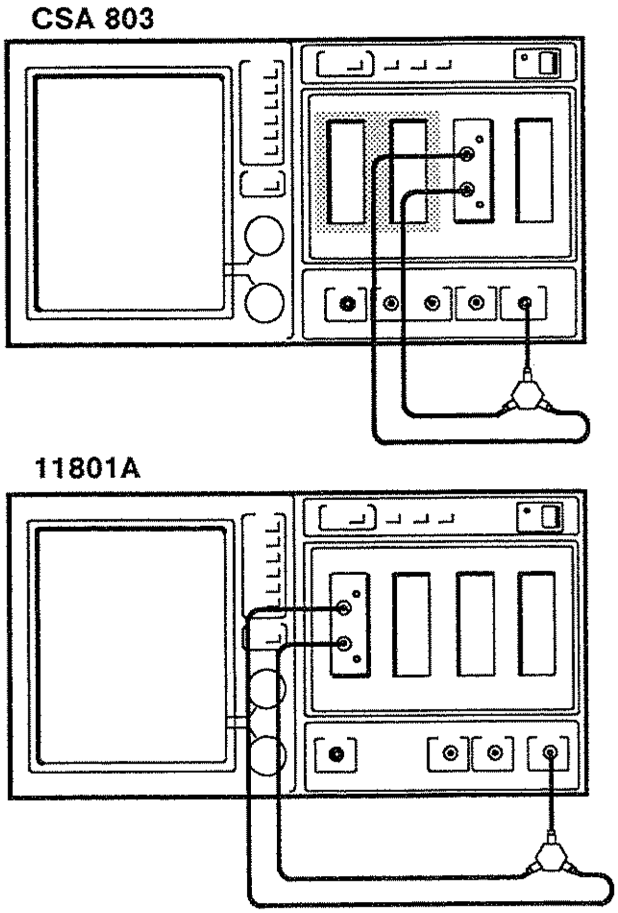
Line 670 prompts you to remove Trace1 and Trace2 from the display and line 680 sends the command to remove both traces from the display.

Lines 690 and 695 prompt you that this example is finished and formally close the program. Line 700 returns you to the Examples Menu.

Example 3: Defining Complex Traces

This example shows how you can create traces that combine signals from more than one channel.

For this example you will need a CSA 803 or 11801A with at least one dual-channel sampling head installed. Also, a power divider and two SMA cables of different lengths will be used (2 ns and 5 ns cables are recommended). A signal-splitting T adaptor may be substituted for the power divider.



Connections for Example 3

To start the program, select example 3 from the Examples Menu. Refer to the earlier discussion Running the Learning by Example Software for information on using the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 3: Defining Complex Traces"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope's parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$
160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180    INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190    IF A$="" THEN A$="COM1"
200 WEND
210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1
230 INPUT "Press Enter to ask for the scope's ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$

270 PRINT "Press Enter after making the following connections :"
280 PRINT "  Use a power splitter or a power 'T' connecter to connect a 2ns and
a 5ns"
290 PRINT "  cable to channel 1 and 2 of sampling head 1."
300 INPUT "  Connect the 'T' or splitter to the CALIBRATOR",A$
310 INPUT "Press Enter to initialize scope.",A$
320 PRINT #1,"INIT"

330 A$=""
340 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
350    INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
360 WEND
370 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"

380 INPUT "Press Enter to setup 2 traces from channel 1 & 2.",A$
390 PRINT #1,"TRACE1 DESCRIPTION:'M1';TRACE2 DESC:'M2'"

400 INPUT "Press Enter to set the trigger to INTERNAL.",A$
410 PRINT #1,"TRIGGER SOURCE:INTERNAL"

420 INPUT "Press Enter to set the main timebase to 10ns/div.",A$
430 PRINT #1,"TBMAIN TIME:10E-9;CHM1 SENSITIVITY:100E-3;CHM2 SENSITIV-
ITY:100E-3"
```

(cont.)

Program Description

Lines 100 through 260 of Example 3 describe the same type of preparatory procedure as described in Example 1, program lines 100 through 260.

Lines 270 through 300 of this example remind you to make the necessary hardware setup for this example.

Line 310 prompts you to initialize the instrument and line 320 sends the INIT command.

Lines 330 through 360 wait for you to type Y or N to enable or disable DEBUG mode.

Line 370 displays the selected DEBUG mode.

Lines 380 through 390 wait for you to press 'Enter' to initiate the definition of Trace1 and Trace 2 from Channel 1 and 2 of the sampling head.

Line 400 prompts you to select internal triggering for this example and line 410 sends the command to set the triggering source.

Line 420 prompts you to set the main time base and line 430 sends the main time base command (10 ns/div setting) and the commands to set the sensitivity (i.e., vertical size) of Channel 1 and Channel 2 to 100 mV (this is to give Trace1 and Trace2 common scaling).

(cont.)

Program Listing (cont.)

```
440 INPUT "Press Enter to create trace3 as the difference of channel 1 & 2",A$
450 PRINT #1,"TRACE3 DESCRIPTION:´M1-M2´"

460 INPUT "Press Enter to remove traces 1 and 2.",A$
470 PRINT #1,"REMOVE TRACE1,TRACE2"

480 INPUT "Press Enter to make a window on trace 3.",A$
490 PRINT #1,"TRACE4 DESCRIPTION:´M1-M2 ON WIN´"

500 INPUT "Press Enter to change the window time base and position.",A$
510 PRINT #1,"TBWIN TIME:1E-9;WIN4 POS:73E-9"

520 INPUT "Press Enter to make a second window on trace 3.",A$
530 PRINT #1,"TRACE5 DESCRIPTION:´M1-M2 ON WIN´"
540 INPUT "Press Enter to change the window position on trace 5.",A$
550 PRINT #1,"WIN5 POS:86E-9"
560 INPUT "Press Enter to add trace separation to Trace 4 and 5.",A$
570 PRINT #1,"ADJTRACE4 TRSEP:-1;ADJTRACE5 TRSEP:1"

580 PRINT "Example Completed."
585 CLOSE 1
590 LOAD "MENU.BAS",R
```

Program Description (cont.)

Line 440 prompts you to create a third trace, the difference between Trace1 and Trace2 and line 450 sends the command to create Trace3 by subtracting the signal from Channel 2 from the signal from Channel 1.

Line 460 prompts you to remove Trace1 and Trace2 from the display and line 470 sends the command to remove these traces, leaving only Trace3.

Line 480 prompts you to create a window trace on Trace3 and line 490 sends the command to create window Trace4 by using the window time base and subtracting the signal of Channel 2 from the signal of Channel 1.

Line 500 prompts you to change the horizontal size and position of the window time base and line 510 sends the command to set the time per division of the window time base to 1 ns per division and the horizontal position of window Trace4 to 73 ns.

Lines 520 through 570 prompt you to create a second window trace and position it by adjusting the trace separation of Trace4 and Trace5.

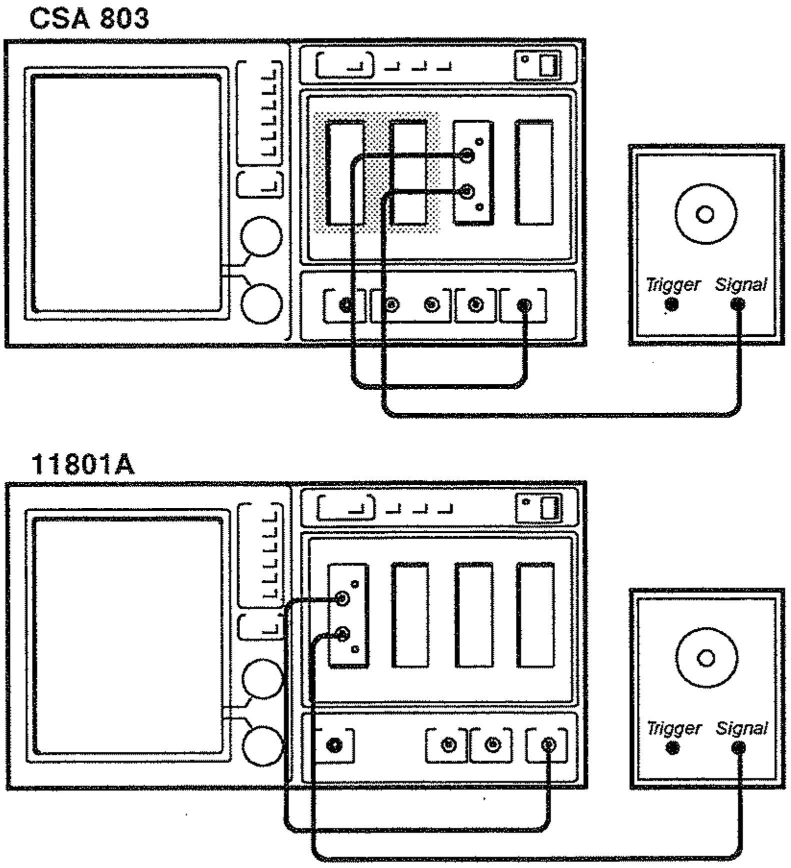
Lines 580 and 585 prompt you that this example is finished and formally close the program. Line 590 returns you to the Examples Menu.

Example 4: Using Signal Processing & Transferring a Waveform

This example shows how the CSA 803 and 11801A can process your signals to extract more information and how to transfer traces to your computer. Trace averaging is started on a trace then used to halt conditional acquisition. Three other display types are invoked; *variable persistence*, *infinite persistence*, and *color graded*. Also, traces are transferred from the CSA 803 or 11801A to your computer.

You will simulate a noisy signal by adding the calibrator to a very small (50 mVp-p), high frequency (50 MHz) sine wave that is not synchronized with the trigger. If you have a broad spectrum noise source, you should substitute it for the signal generator in this example.

For this example you will need a CSA 803 or 11801A with at least one dual-channel sampling head installed, a signal generator, and two SMA cables.



Connections for Example 4

To start the program, select example 4 from the Examples Menu. Refer to the earlier discussion Running the Learning by Example Software for information on using the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 4: Using Signal Processing & Transferring a Waveform"
103 PRINT
110 PRINT "* * * You must have invoked BASIC with a COM buffer of 5000 bytes"
111 PRINT "* * * to run this example."
120 PRINT "* * * Sample invocation: BASIC /C:5000 or BASICA /C:5000 or QB
      /C:5000"
122 PRINT "* * * If you have not invoked BASIC in this way, this example WILL
      NOT RUN;"
123 PRINT "* * * get out of BASIC and re-invoke."
124 PRINT
130 PRINT "Ensure that the cable is connected and the scope's parameters are"
140 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
150 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
160 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
170 INPUT "  Press Enter when ready",A$
180 A$=""
190 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
200     INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
210     IF A$="" THEN A$="COM1"
220 WEND
230 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
240 OPEN A$+":9600,N,8,1" AS #1
250 INPUT "Press Enter to ask for the scope's ID",A$
260 PRINT #1,"ID?"
270 LINE INPUT #1,RESPONSE$
280 PRINT RESPONSE$
```

(cont.)

Program Description

Lines 100 through 280 of Example 4 describe the same type of preparatory procedure as described in Example 1, program lines 100 through 260. One exception is a warning that you must have started Basic with a argument that would set up a 5000 byte buffer used in the waveform transfer.

Program Listing (cont.)

```
290 PRINT "Press Enter after making the following connections :"
300 PRINT " Connect a 50 MHz sine wave, 50 mV p-p, to channel 2."
310 PRINT " Connect the Calibrator signal to channel 1."
320 INPUT "Press Enter to initialize scope.",A$
330 PRINT #1,"INIT"

340 A$=""
350 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
360     INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
370 WEND
380 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"

390 INPUT "Press Enter to set the trigger to INTERNAL.",A$
400 PRINT #1,"TRIGGER SOURCE:INTERNAL"
```

(cont.)

Program Description (cont.)

Lines 290 through 310 of this example remind you to make the necessary hardware setup for this example.

Line 320 prompts you to initialize the instrument and line 330 sends the INIT command.

Lines 340 through 370 wait for you to type Y or N to enable or disable DEBUG mode.

Line 380 displays the selected DEBUG mode.

Lines 380 through 390 wait for you to press ‘Enter’.to initiate the definition of Trace1 Trace 2 from Channel 1 and 2 of the sampling head.

Line 390 prompts you to select internal triggering for this example and line 400 sends the command to set the triggering source.

(cont.)

Program Listing (cont.)

```
410 INPUT "Press Enter to define an averaged and a normal summation-trace with
M1 and M2",A$
420 PRINT #1,"TRACE1 DESCR:´M1+M2´;TRACE2 DESCR:´AVG(M1+M2)´"
430 INPUT "Press Enter to set the time base and vertical size of trace 2",A$
440 PRINT #1,"TBMAIN TIME:10E-9;CHM1 SENSI:.2;CHM2 SENSI:.2"

450 INPUT "Press Enter to set the number of averages to 12.",A$
460 PRINT #1,"NAVG 12"
470 INPUT "Press Enter to begin a conditional acquisition.",A$
480 PRINT "The average countdown will be complete when ´n´ = zero."
490 PRINT #1,"CONDACQ TYPE:AVG"
500 REMAINING=999
510 WHILE REMAINING>0
520     PRINT #1,"CONDACQ? REMAINING"
530     LINE INPUT #1,RESPONSE$
540     REMAINING=VAL(MID$(RESPONSE$,INSTR(RESPONSE$,":")+1))
550     PRINT REMAINING;
560 WEND
570 PRINT

580 INPUT "Press Enter to send the 512 point curve to the PC.",A$
590 GOSUB 805

600 INPUT "Press Enter to turn the average function off for TRACE2.",A$
610 PRINT #1,"AVG OFF;CONDACQ TYPE:CONTINUOUS"

620 INPUT "Press Enter to set the display for infinite persistence.",A$
630 PRINT #1,"DISPLAY TYPE:INFINITE"
640 INPUT "Press Enter to set the display to variable persistence.",A$
650 PRINT #1,"DISPLAY TYPE:VARIABLE"
652 INPUT "Press Enter to set the display to color graded.",A$
654 PRINT #1,"DISPLAY TYPE:GRADED"
656 INPUT "Press Enter to set the display back to normal.",A$
658 PRINT #1,"DISPLAY TYPE:NORMAL"

660 PRINT "Press Enter to smooth channel 1."
670 INPUT " (note: Turning on smoothing for CHM1 also smooths CHM2)",A$
680 PRINT #1,"CHM1 SMOOTHING:ON;CHM2? SMOOTHING"
690 LINE INPUT #1,RESPONSE$
700 PRINT RESPONSE$

710 INPUT "Press Enter to turn off the smoothing.",A$
720 PRINT #1,"CHM1 SMOOTHING:OFF;CHM2? SMOOTHING"
730 LINE INPUT #1,RESPONSE$
740 PRINT RESPONSE$
```

(cont).

Program Description (cont.)

Line 410 waits for you to press 'Enter' to define two traces and line 420 sends the command that defines Trace1 to be the summation of Channels 1 and 2, and it sends the command that defines Trace2 to be the average of the summation of Channels 1 and 2.

Line 430 prompts you to set the horizontal and vertical size of Trace1 and Trace2. Line 440 sends the commands that set the main time base to 10 ns per division and the sensitivities of Channel 1 and 2 to 200 mV per division. Note that both channels must be set since they are summed.

Lines 450 through 570 prompts you to set 12 averages (i.e., average 12 records) for a conditional trace acquisition and to initiate conditional acquisition which stops acquisition after 12 acquisitions.

Line 580 prompts you to send the acquired trace/waveform to your computer. Line 590 sends the program to a subroutine (lines 805 through 1020) that transfers the data points of your waveform to your computer. And, if you have graphics, it draws the waveform on your computer screen. Then it returns you to line 600 to continue the example.

Line 600 prompts you to turn off the average function for Trace2, and line 610 sends the command to turn off averaging and reset conditional acquire to the continuous mode.

Lines 620 and 630 prompt you to send the command that sets the display type to Infinite Persistence.

Lines 640 and 650 prompt you to send the command that sets the display type to Variable Persistence.

Lines 652 and 654 prompt you to send the command that sets the display type to Color Graded.

Lines 656 and 658 prompt you to send the command that sets the display type back to Normal.

Lines 660 and 670 prompt you to select trace smoothing for Channel 1 which also starts smoothing for Channel 2.

Line 680 sends the command to turn smoothing on for Channel 1 and sends a command to query Channel 2 to see if smoothing was really turned on. Lines 690 and 700 read the query response and print it on the computer screen.

Lines 710 and 740 prompt you to turn smoothing off, and then send commands to turn smoothing off for Channel 1, and to query if Channel 2 smoothing was turned off.

(cont.)

Program Listing (cont.)

```
750 INPUT "Press Enter to set the main record to 2048 and the window to
5120",A$
760 PRINT #1,"TBMAIN LENGTH:2048;TBWIN LENGTH:5120"
770 INPUT "Press Enter to send the 2048 point curve to the PC.",A$
780 GOSUB 810

790 PRINT "Example Completed."
795 CLOSE 1
800 LOAD "MENU.BAS",R

805 REM The following line sets the output waveform to be TRACE2; the waveform
806 REM will be sent in 16 bit binary format and each word will be sent as low
807 REM order byte followed by high order byte; lastly, ask for the data.
810 PRINT #1,"OUTPUT TRACE2;ENCDG WAV:BIN;BYT.OR LSB;CURVE?
815 REM read one byte off the bus
820 A$=INPUT$(1,#1)
825 REM loop until we see the percent sign
830 WHILE A$<>"%" :   A$=INPUT$(1,#1) :WEND
835 REM next, read in the data length count (always in MSB,LSB word format)
840 HB$=INPUT$(1,#1) : LB$=INPUT$(1,#1)
845 REM compute the byte count
850 BYTE.CNT%=ASC(HB$)*256+ASC(LB$)
855 REM convert the byte count into waveform point count
860 NR.PT%=(BYTE.CNT%-1)/2 : DIM WFM%(NR.PT%)
865 REM read in the waveform in one word at a time
870 FOR I%=1 TO NR.PT% :   WFM%(I%)=CVI(INPUT$(2,#1)) : PRINT ".";: NEXT
875 REM lastly, read in the checksum (non-zero only for stored waveforms)
880 LINE INPUT #1,CHKSUM$
890 PRINT
900 INPUT "Waveform copy complete. Press Enter to graph.",A$
905 REM check for graphics adapter
910 DEF SEG = (&H40)
920 CRTTYPE = PEEK(&H49)
930 DEF SEG = 0
940 IF CRTTYPE =7 THEN PRINT "Graphics not available." : ERASE WFM% : RETURN
945 REM switch to 640x200 (try SCREEN 9 for EGA if you use QuickBASIC)
950 SCREEN 2
960 REM set the screen to be zero +/- 32k  by  record-length (e.g. 1024)
970 WINDOW(1,-32767)-(nr.pt%,32767)
980 REM plot each point as a pixel
990 FOR i%=1 TO nr.pt% : PSET(i%,wfm%(i%)) : NEXT
1000 INPUT "Press Enter to clear the screen.",A$
1005 REM remove the waveform array and set the screen back to text
1010 ERASE WFM% : SCREEN 0
1020 RETURN
```

Program Description (cont.)

Line 750 prompts you to set the main record length to 2048 data points and the window record length to 5120 data points. Line 760 sends the commands to do this.

Line 770 prompts you to send the new 2048 point curve (i.e., trace or waveform) to your computer.

Line 780 sends the program to the curve transfer and screen draw subroutine (lines 805 though 1020) and returns you to line 790 when its finished.

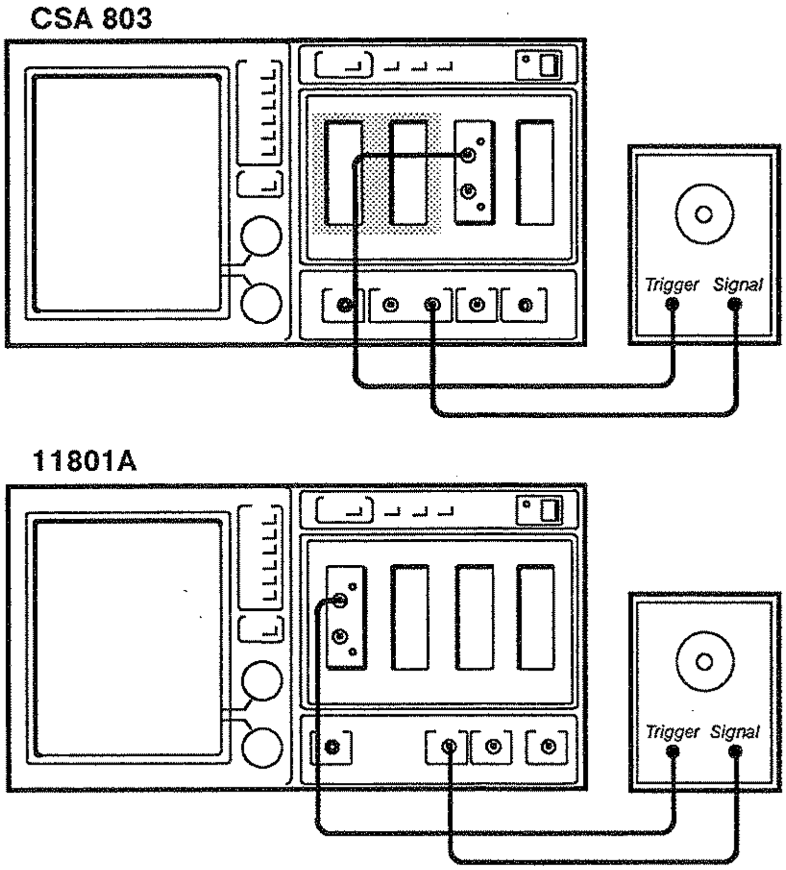
Lines 790 and 795 prompt you that this example is finished and formally closes the program. Line 800 returns you to the Examples Menu.

Lines 805 through 1020 compose a subroutine that transfers the data points of a trace record to your computer. And, if you have graphics capability, it draws the waveform on your computer screen. It then returns to line following the calling GOSUB command to continue the example.

Example 5: Taking Automated Measurements

This example demonstrates how quickly you can take a dynamic measurement from a displayed trace.

For this example you will need a CSA 803 or 11801A with a least one sampling head installed, a signal generator, and two SMA connecting cables.



Connections for Example 5

To start the program, select example 5 from the Examples Menu. Refer to the earlier discussion Running the Learning by Example Software for information on using the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 5: Taking Automated Measurements"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope´s parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,     EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$
160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180     INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190     IF A$="" THEN A$="COM1"
200 WEND
210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1
230 INPUT "Press Enter to ask for the scope´s ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$

270 PRINT "Press Enter after making the following connections :"
280 PRINT " Connect a 1 MHz sine wave, .5 to 1V P-P to channel 1."
290 PRINT " Connect the generator trigger out to the scope TRIGGER INPUT."
300 INPUT "Press Enter to initialize scope.",A$
310 PRINT #1,"INIT"

320 A$=""
330 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
340     INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
350 WEND
360 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"

370 INPUT "Press Enter to setup trace1 from sampling head 1, channel 1.",A$
380 PRINT #1,"TRACE1 DESCRIPTION:´M1´"

390 INPUT "Press Enter to set the trigger to EXTERNAL.",A$
400 PRINT #1,"TRIGGER SOURCE:EXTERNAL"

410 INPUT "Press Enter to send ´AUTOSET´ to the scope",A$
420 PRINT #1,"AUTOSET MODE:PERIOD,START"
```

(cont.)

Program Description

Lines 100 through 260 of Example 5 describe the same type of preparatory procedure as described in Example 1, program lines 100 through 260.

Lines 270 through 310 of this example remind you to make the necessary hardware setup and to initialize the instrument for this example.

Lines 320 through 360 wait for you to type Y or N to enable or disable DEBUG mode then display the selected DEBUG mode.

Line 370 waits for you to press 'Enter' to define Trace1 and line 380 sends the command to define Trace1 as Channel 1 of the sampling head.

Line 390 prompts you to select external triggering for this example and line 400 sends the command to do this.

Line 410 prompts you to AUTOSET the selected trace and line 420 sends the command to AUTOSET in period mode.

(cont.)

Program Listing (cont.)

```
430 INPUT "Press Enter to calculate RMS and FREQUENCY on the SELECTed
TRACE.",A$
440 PRINT #1,"RMS?;FREQ?"
450 LINE INPUT #1,RESPONSE$
460 PRINT "(a histogram was calculated for each measurement query)"
470 PRINT RESPONSE$

480 INPUT "Press Enter to put RMS and FREQUENCY into the measure list for
TRACE1.",A$
490 PRINT #1,"MSLIST1 RMS,FREQ"
500 PRINT "Press Enter to get the query results"
510 INPUT " (only one histogram is made for all measurements)",A$
520 PRINT #1,"MEAS1?"
530 LINE INPUT #1,RESPONSE$
540 PRINT RESPONSE$

550 INPUT "Press Enter to setup the statistical data.",A$
560 PRINT #1,"STATISTICS MEAS:RMS,N:24;MSYS ON"
570 INPUT "Press Enter to fetch the statistical data.",A$
580 PRINT #1,"STAT?"
590 LINE INPUT #1,RESPONSE$
600 PRINT RESPONSE$

610 INPUT "Press Enter to change the measurement zones on the trace.",A$
620 PRINT #1,"MPARAM1 LMZONE:15,RMZONE:85"

630 INPUT "Press Enter to measure CHM1 freq with s/w (using new zones).",A$
640 PRINT #1,"MPARAM1 MMODE:SW;MSLIST1 FREQ;MEAS1?"
650 LINE INPUT #1,RESPONSE$
660 PRINT RESPONSE$

670 INPUT "Press Enter to measure CHM1 freq with h/w (zones not used).",A$
680 PRINT #1,"MPARAM1 MMODE:HW;MEAS1?"
690 LINE INPUT #1,RESPONSE$
700 PRINT RESPONSE$

710 INPUT "Press Enter to remove TRACE 1",A$
720 PRINT #1,"REMOVE TRACE1"

770 PRINT "Example Completed."
775 CLOSE 1
780 LOAD "MENU.BAS",R
```

Program Description (cont.)

Line 430 prompts you to measure the RMS and frequency of the selected trace (i.e., Trace1).

Line 440 sends the commands to query RMS and frequency and return their values.

> **Note:** Beginning with these independent RMS and frequency measurements, there are four more times that these measurements are taken. Each time, a different measurement method is used to demonstrate the various methods that are available to you with the CSA 803 or 11801A. The subsequent measurements are made in lines 520 (demonstrating measurement list measurements), 640 (demonstrating software system measurements), and 680 (demonstrating hardware system measurements).

Lines 450 through 470 read the query responses and print the RMS and frequency values on your computer screen.

Line 480 prompts you to add RMS and frequency to the measurement list (MSLIST) for Trace1 and line 490 sends the command to do this.

Line 500 through 540 query the new values for the RMS and frequency measurements just added to the measurement list for Trace1.

Lines 550 and 560 set up the instrument to take statistical data on the RMS measurement, using 24 samples, and to turn the measurement system on. Also, note that MSYS must be ON to use the statistics function.

> **Note:** If you press 'Enter' too soon in line 570, line 560 will not have had time to take all 24 samples.

Line 570 prompts you to get the statistical data and lines 580 through 600 query the statistical data.

Line 610 prompts you to change the reference zones for taking measurements and line 620 sends the command changing the left and right measurement zones for Trace1. It sets them to 15 percent and 85 percent, respectively.

Lines 630 through 660 measure Trace1 frequency in software mode, using the new measurement zones.

Line 670 through 700 measure Trace1 frequency in hardware mode, without the measurement zones.

Line 710 prompts you to remove Trace1 from the display and line 720 sends the command to do this.

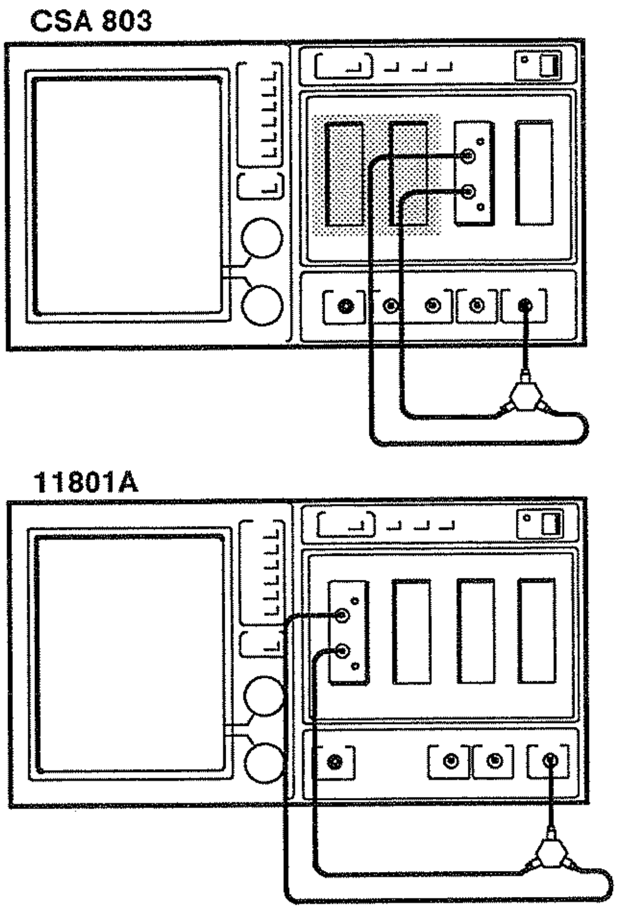
Lines 770 and 775 prompt you that this example is finished and formally close the program. Line 780 returns you to the Examples Menu.

Example 6: Taking Delay Measurements

This example shows another way to measure trace parameters, such as delay, using cursors. When a measurement you want to make is not included in the list of automated measurements, cursors can be used.

You can use the cursors to take two common measurements, trace amplitude and delay between traces. The CSA 803 and 11801A can do both of these as automated measurements, so you can compare the method of using automated measurements to using cursors.

For this example you will need a CSA 803 or 11801A with at least one dual-channel sampling head installed. Also, a power divider and two SMA cables of different length will be used (2 ns and 5 ns cables are recommended). A signal splitting T adaptor may be substituted for the power divider.



Connections for Example 6

To start the program, select example 6 from the Examples Menu. Refer to the earlier discussion Running the Learning by Example Software for information on using the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 6: Taking Delay Measurements"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope's parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$
160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180    INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190    IF A$="" THEN A$="COM1"
200 WEND
210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1
230 INPUT "Press Enter to ask for the scope's ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$
```

(cont.)

Program Description

Lines 100 through 260 of Example 6 describe the same type of preparatory procedure as described in Example 1, program lines 100 through 260.

(cont.)

Program Listing (cont.)

```
270 PRINT "Press Enter after making the following connections :"
280 PRINT " Use a power splitter or a power ´T´ connecter to connect a 2ns and
a 5ns"
290 PRINT " cable to channel 1 and 2 of sampling head 1."
300 INPUT " Connect the ´T´ or splitter to the CALIBRATOR of the 11800",A$
310 INPUT "Press Enter to initialize scope.",A$
320 PRINT #1,"INIT"
330 A$=""

340 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
350     INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
360 WEND
370 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"

380 INPUT "Press Enter to setup trace1 from sampling head 1, channel 1.",A$
390 PRINT #1,"TRACE1 DESCRIPTION:´m1´"
400 INPUT "Press Enter to set the trigger to the INTERNAL.",A$
410 PRINT #1,"TRIGGER SOURCE:INTERNAL"

420 INPUT "Press Enter to set to the time base to 10ns/div",A$
430 PRINT #1,"TBMAIN TIME:10E-9"
```

(cont.)

Program Description (cont.)

Lines 270 through 330 of this example remind you to make the necessary hardware setup and to initialize the instrument for this example.

Lines 340 through 370 wait for you to type Y or N to enable or disable DEBUG mode then display the selected DEBUG mode.

Line 380 waits for you to press 'Enter' to define Trace1 and line 390 sends the command to define Trace1 as Channel 1 of the sampling head.

Line 400 prompts you to select internal triggering for this example and line 410 sends the command to do this.

Line 420 prompts you to set the time per division for the main time base and line 430 sends the command to do this.

(cont.)

Program Listing (cont.)

```
440 INPUT "Press Enter to turn on the cursor readout.",A$
450 PRINT #1,"CURSOR READOUT:ON"


460 INPUT "Press Enter to select horizontal bars.",A$
470 PRINT #1,"CURSOR TYPE:HBARS"

480 INPUT "Press Enter to move the bars near the middle of the CRT.",A$
490 PRINT #1,"H1BAR YDIV:1;H2BAR YDIV:-1"

500 INPUT "Press Enter to go to paired cursors on trace 1.",A$
510 PRINT #1,"CURSOR TYPE:PAIRED"
520 INPUT "Press Enter to position the cursors on divisions 1 and 9.",A$
530 PRINT #1,"DOT1ABS XDIV:-4;DOT2ABS XDIV:4"

540 PRINT #1,"DOT1ABS? YCOORD;DOT2ABS? YCOORD;PP?"
550 LINE INPUT #1,RESPONSE$
560 PRINT "Here are the vertical values from each cursor:"
570 PRINT RESPONSE$

580 M$=MID$(RESPONSE$,INSTR(RESPONSE$,":")+1)
590 PP=ABS(VAL(M$)-VAL(MID$(M$,INSTR(M$,":")+1)))
600 PRINT "Here is the Peak to Peak value from the cursors:";PP

610 M$=MID$(RESPONSE$,INSTR(RESPONSE$,"PP")+1)
620 PP=VAL(MID$(M$,INSTR(M$," ")+1))
630 PRINT "Here is the Peak to Peak value from the measurement system:";PP

640 INPUT "Press Enter to create a second trace on the lower graticule.",A$
650 PRINT #1,"TRACE2 DESC:'M2';DISPLAY GRAT:DUAL;ADJTRA1 GRLOC:UPPER;ADJTRA2
GRLOC:LOWER"

660 PRINT #1,"SELECT TRACE1;CURSOR TYPE:SPLIT,REFERENCE:TRACE2"
670 PRINT "You may use the cursors to make a timing measurement, but we will"
680 PRINT " show you how to make accurate automatic measurements with the "
690 PRINT " measurement system."

700 PRINT "Example Completed."
705 CLOSE 1
710 LOAD "MENU.BAS",R
```

Program Description (cont.)

Line 440 prompts you to turn on the cursor readout and line 450 sends the command to do this.

Line 460 prompts you to select horizontal bar cursors and line 470 sends the command to set the cursor type to horizontal bars.

Line 480 prompts you to place the cursors near the middle of the display and line 490 sends the command to place horizontal bar 1 at plus one division on the Y-axis and to place horizontal bar 2 at minus one division on the Y-axis.

Line 500 through 530 select dot-paired cursors and repositions them, setting dot 1 at an absolute value of minus four divisions on the X-axis and setting dot 2 at an absolute value of plus four divisions on the X-axis.

Lines 540 through 570 query the Y-coordinate values at dot 1 and dot 2 and the peak-to-peak measurement and displays these vertical values and their accuracy qualifiers to your computer screen

Line 580 removes the header, link, and colon from the beginning of the returned string.

Line 590 and 600 calculates the peak-to-peak value based on cursor values extracted in line 580.

Lines 610 through 630 finds the peak-to-peak character string returned from lines 540 and 550 (i.e., made with the automated measurement system) and extracts the peak-to-peak value, which is printed to your computer screen.

Lines 640 and 650 create a second trace (from Channel 2) and place it on a second lower graticule. Trace1 remains on the upper graticule.

Line 660 sends the commands to select Trace1, select split-dot cursors, and to use Trace2 as a reference.

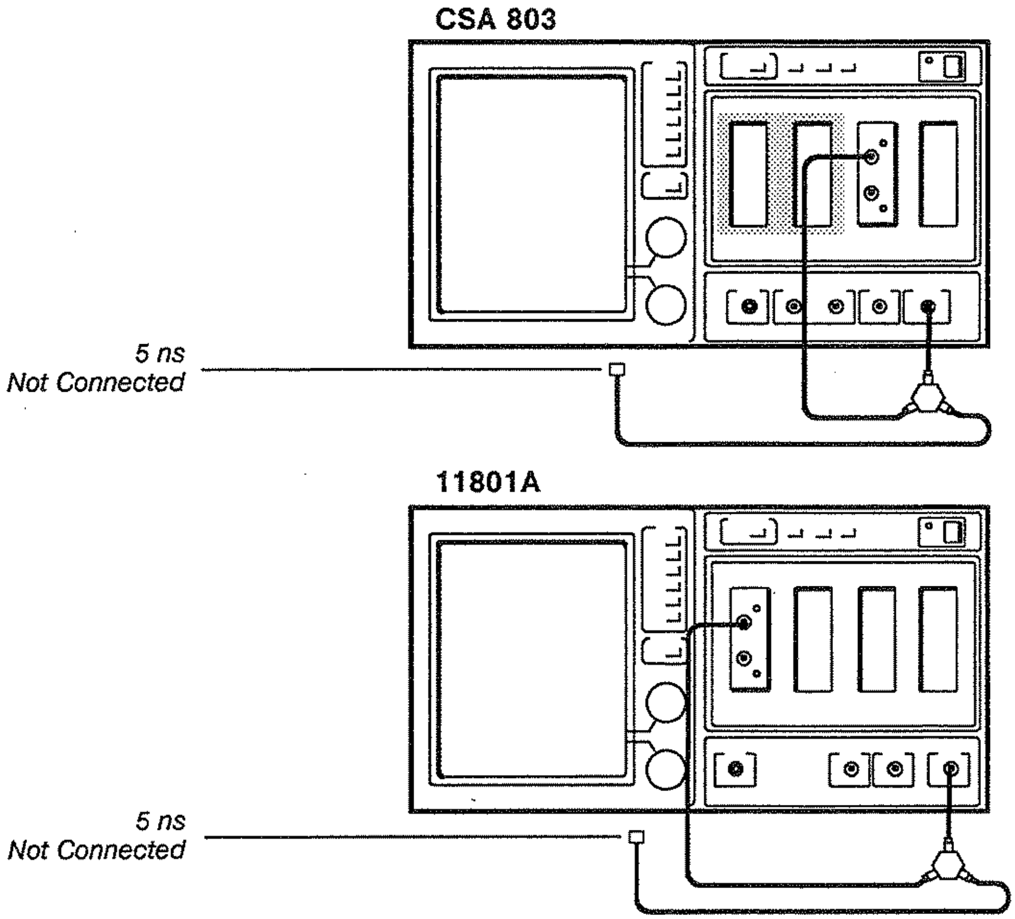
Lines 670 through 690 inform you that you may use the cursors to make delay timing measurements, but recommend that you use the automatic measurement system, instead. In fact, a part of Example 9 demonstrates a method of taking delay measurements with the automatic measurement system (i.e., the PDELAY or propagation delay measurement command).

Lines 700 and 705 prompt you that this example is finished and formally close the program. Line 710 returns you to the Examples Menu.

Example 7: Taking ΔDelay Measurements

This example demonstrates the compare and reference features of the automated measurement system. This is particularly useful when you need to check a series of components or circuits to see if a particular measurement falls within an acceptable range.

For this example you need a CSA 803 or 11801A with at least one sampling head installed. Also, two SMA cables of different length will be used (2 ns and 5 ns cables are recommended). Use the short cable first, then later, when prompted, use the long cable.



Connections for Example 7

To start the program, select example 7 from the Examples Menu. Refer to the earlier discussion Running the Learning by Example Software for information on using the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 7: Taking Delta-Delay Measurements"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope´s parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$
160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180    INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190    IF A$="" THEN A$="COM1"
200 WEND
210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1
230 INPUT "Press Enter to ask for the scope´s ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$
```

(cont.)

Program Description

Lines 100 through 260 of Example 7 describe the same type of preparatory procedure as described in Example 1, program lines 100 through 260.

(cont.)

Program Listing

```
270 PRINT "Press Enter after making the following connections :"
280 PRINT " Use a power splitter or a power ´T´ connecter to connect a 2ns and
a 5ns"
290 PRINT " cable to channel 1 of sampling head 1 (connect the 2ns cable
first)."
300 INPUT " Connect the ´T´ or splitter to the CALIBRATOR of the 11800",A$
310 INPUT "Press Enter to initialize scope.",A$
320 PRINT #1,"INIT"

330 A$=""
340 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
350     INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
360 WEND
370 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"

380 INPUT "Press Enter to setup trace1 from sampling head 1, channel 1.",A$
390 PRINT #1,"TRACE1 DESCRIPTION:´M1´"

400 INPUT "Press Enter to set the trigger to the INTERNAL.",A$
410 PRINT #1,"TRIGGER SOURCE:INTERNAL"

420 INPUT "Press Enter to set to the time base to 10ns/div",A$
430 PRINT #1,"TBMAIN TIME:10E-9"
```

(cont.)

Program Description

Lines 270 through 320 of this example remind you to make the necessary hardware setup and to initialize the instrument for this example.

Lines 330 through 370 wait for you to type Y or N to enable or disable DEBUG mode then display the selected DEBUG mode.

Line 380 waits for you to press 'Enter' to define Trace1 and line 390 sends the command to define Trace1 as Channel 1 of the sampling head.

Line 400 prompts you to select internal triggering for this example and line 410 sends the command to do this.

Line 420 prompts you to set the time per division for the main time base and line 430 sends the command to set the main time base to 10 ns/division.

(cont.)

Program Listing (cont.)

```
440 INPUT "Press Enter to change the measurement to h/w mode.",A$
450 PRINT #1,"MPARAM1 MMODE:HW"

460 INPUT "Press Enter to set the MSLIST to CROSS and save the value in
memory.",A$
470 PRINT #1,"MSLIST1 CROSS;REFSET1 CURRENT:CROSS"

480 INPUT "Press Enter to turn COMPARE ON and bring up the measure menu.",A$
490 PRINT #1,"COMPARE ON;MSYS ON"

500 INPUT "Press Enter after removing the 2ns cable and connecting the 5ns
cable.",A$

510 INPUT "Press Enter to ask the scope for the time difference between the
cables.",A$
520 PRINT #1,"CROSS?"
530 LINE INPUT #1,RESPONSE$
540 PRINT RESPONSE$;" (in seconds)"

550 INPUT "Press Enter to change the Prop-Velocity to FEET.",A$
560 PRINT " (the default velocity is .7 * (speed of light)"
570 PRINT #1,"GRATICULE XUNIT:FEET"

580 INPUT "Press Enter after reconnecting the 2ns cable to CH1.",A$
590 INPUT "Press Enter to save the CROSS value in memory (in feet).",A$
600 PRINT #1,"REFSET1 CURRENT:CROSS"

610 INPUT "Press Enter after removing the 2ns cable and connecting the 5ns
cable.",A$

620 INPUT "Press Enter to ask the scope for the feet difference between the
cables.",A$
630 PRINT #1,"CROSS?"
640 LINE INPUT #1,RESPONSE$
650 PRINT RESPONSE$;" (in feet)"

660 PRINT "Example Completed."
665 CLOSE 1
670 LOAD "MENU.BAS",R
```

Program Description (cont.)

Lines 440 and 450 select the hardware measurement mode for Trace1.

Lines 460 and 470 place the CROSS measurement into the measurement list for Trace1 and save its current measurement value as a reference.

Lines 480 and 490 prompt you to turn on the compare measurements mode and then select the Measure menu.

Lines 500 through 540 prompt you to reconfigure your cable connection and send a CROSS measurement query to the instrument to determine the difference between crossing points. This effectively measures the time difference (in seconds) between the two cables.

Lines 550 through 570 change the X-axis graticule X-axis scaling to feet which will allow the propagation delay to be measured in feet.

Line 580 prompts you to reconfigure your cable connection in preparation for measuring the length difference between the two cables.

Line 590 prompts you to save the current crossing value as the reference.

Line 600 sends the command to use the current crossing value for Trace1 as the reference measurement.

Line 610 prompts you to reconfigure your cable connection.

Lines 620 through 650 determine the difference between your new crossing value and the reference value.

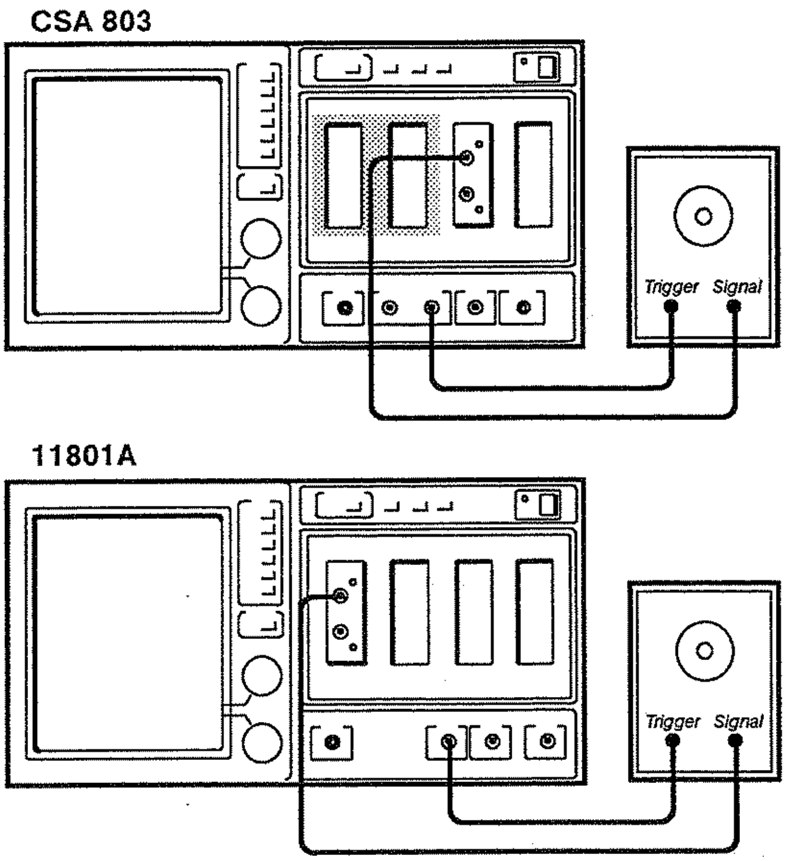
Lines 660 and 670 prompt you that this example is finished and ends the program.

Lines 660 and 665 prompt you that this example is finished and formally close the program. Line 670 returns you to the Examples Menu.

Example 8: Comparing Traces

This example demonstrates storing a trace that represents a "snapshot" of a particular moment, and using that trace as a basis for comparison with other traces. This is similar to Example 7, where a reference measurement was used as a basis of comparison. This time, the entire trace will be saved as a reference.

For this example you will need a CSA 803 or 11801A with at least one sampling head installed, two SMA connecting cables, and a signal generator.



Connections for Example 8

To start the program, select example 8 from the Examples Menu. Refer to the earlier discussion Running the Learning by Example Software for information on using the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 8: Comparing Traces"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope's parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$
160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180    INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190    IF A$="" THEN A$="COM1"
200 WEND
210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1
230 INPUT "Press Enter to ask for the scope's ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$270 PRINT "Press Enter after making the following connec-
tions :"
280 PRINT " Connect a 500kHz to 5MHz sinewave signal (.5V to 1V PP) to channel
1."
290 INPUT " Connect trigger out of your signal source to the TRIGGER INPUT of
the scope.",A$
300 INPUT "Press Enter to initialize scope.",A$
310 PRINT #1,"INIT"320 A$=""
330 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
340    INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
350 WEND
360 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"
370 INPUT "Press Enter to setup trace1 from sampling head 1, channel 1.",A$
380 PRINT #1,"TRACE1 DESCRIPTION:'M1'"
```

(cont.)

Program Description

Lines 100 through 380 of Example 8 describe the same type of preparatory procedure (including the necessary hardware setup) as described in Example 1, program lines 100 through 360.

(cont.)

Program Listing (cont.)

```
390 INPUT "Press Enter to set the trigger to EXTERNAL",A$
400 PRINT #1,"TRIGGER SOURCE:EXTERNAL"

410 INPUT "Press Enter to send ´AUTOSET´ to the scope",A$
420 PRINT #1,"AUTOSET MODE:PERIOD,START"
```

(cont.)

Program Description (cont.)

Lines 100 through 380 of Example 8 describe the same type of preparatory procedure (including the necessary hardware setup) as described in Example 1, program lines 100 through 360.

Line 390 prompts you to select external triggering for this example and line 400 sends the command to do this.

Line 410 prompts you to AUTOSET the selected trace and line 420 sends the command to begin autoset using period mode.

(cont.)

Program Listing (cont.)

```
430 INPUT "Press Enter to store trace1 to sto1.",A$
440 PRINT #1,"STORE TRA1:STO1"

450 INPUT "Press Enter to remove trace 1.",A$
460 PRINT #1,"REMOVE TRACE1"

470 INPUT "Press Enter to recall sto1.",A$
480 PRINT #1,"TRACE1 DESCR:´STO1´"

490 INPUT "Press Enter to create a difference trace showing only the noise.",A$
500 PRINT #1,"TRACE1 DESC:´STO1-M1´"
510 INPUT "Press Enter to find the RMS value of the noise over the WHOLE
record.",A$
520 PRINT #1,"MPARAM1 DAINT:WHOLE;MSLIST1 RMS;MEAS1?"
530 LINE INPUT #1,RESPONSE$
540 PRINT RESPONSE$

550 PRINT "Example Completed."
555 CLOSE 1
560 LOAD "MENU.BAS",R
```

Program Description (cont.)

Lines 430 and 440 store Trace1 as Stored Trace1.

Lines 450 and 460 remove Trace1 from the display.

Lines 470 and 480 recall Stored Trace1 to the display as Trace1.

Lines 490 and 500 create a trace that is the difference between Stored Trace1 and Channel 1.

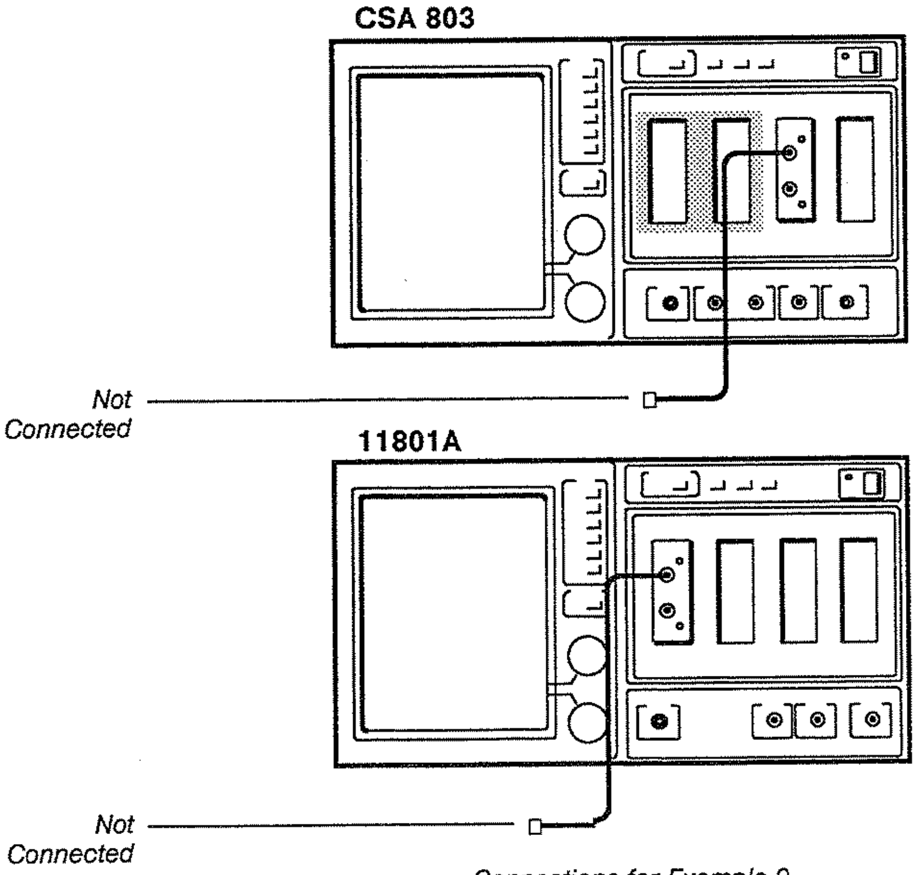
Lines 510 through 540 measure the RMS value of Trace1. The measurement parameter data interval is set to WHOLE for Trace1 and the RMS measurement is included in the measurement list for Trace1, then all defined measurements for Trace1 (i.e., RMS) are queried.

Lines 550 and 555 prompt you that this example is finished and formally close the program. Line 560 returns you to the Examples Menu.

Example 9: Taking TDR Measurements

This example demonstrates the TDR (time domain reflectometry) feature of the SD-24 sampling heads when they are used in combination with the CSA 803. TDR is a method of examining and measuring a network or transmission line by sending a pulse into the network and monitoring the reflections.

For this example you will need a CSA 803 or 11801A with at least one SD-24 Sampling head installed. Also, you will need one SMA cable, preferably of 5 ns length.



Connections for Example 9

To start the program, select example 9 from the Examples Menu.

Program Listing

```
100 LOCATE ,,1,0,7
102 PRINT "Example 9: Taking TDR Measurements"
103 PRINT
110 PRINT "Ensure that the cable is connected and the scope´s parameters are"
120 PRINT "set accordingly : (major menu UTILITY, minor menu RS232 Parameters)"
130 PRINT "  Baud 9600,  Echo OFF,  Stop Bits 1,  Parity NONE,  Flagging HARD"
140 PRINT "  Delay 0,    EOL String CR/LF,        Verbose OFF"
150 INPUT "  Press Enter when ready",A$
160 A$=""
170 WHILE A$<>"COM1" AND A$<>"COM2" AND A$<>"com2"
180    INPUT "Press Enter for using COM1 (default), else type COM2. ",A$
190    IF A$="" THEN A$="COM1"
200 WEND
210 PRINT "We are opening ";A$;" at 9600 baud, no parity, and 1 stop bit."
220 OPEN A$+":9600,N,8,1" AS #1
230 INPUT "Press Enter to ask for the scope´s ID",A$
240 PRINT #1,"ID?"
250 LINE INPUT #1,RESPONSE$
260 PRINT RESPONSE$
270 PRINT "Press Enter after connecting your cables to the scope.  Require-
ments:"
280 INPUT "   Connect a 5ns SMA cable to CHM1 (leave it open ended)",A$
290 INPUT "Press Enter to initialize scope.",A$
300 PRINT #1,"INIT"
310 A$=""
320 WHILE A$<>"y" AND A$<>"Y" AND A$<>"n" AND A$<>"N"
330    INPUT "Do you want to watch the commands in DEBUG mode ? (y/n)",A$
340 WEND
350 IF A$= "y" OR  A$= "Y" THEN PRINT #1,"DEBUG RS232:ON" ELSE PRINT #1,"DEBUG
RS232:OFF"
360 INPUT "Press Enter to setup trace1 from sampling head 1, channel 1.",A$
370 PRINT #1,"TRACE1 DESCRIPTION:´M1´"

380 INPUT "Press Enter to set the trigger to INTERNAL",A$
390 PRINT #1,"TRIGGER SOURCE:INTERNAL"

400 INPUT "Press Enter to turn on the TDR pulse",A$
410 PRINT #1,"CHM1 TDRSTATE:ON"

420 INPUT "Press Enter to change the x to FEET",A$
430 PRINT #1,"GRATICULE XUNIT:FEET"
```

(cont.)

Program Description

Lines 100 through 370 of Example 9 describe the same type of preparatory procedure (including the necessary hardware setup) as described in Example 1, program lines 100 through 360.

Line 380 prompts you to select internal triggering for this example and line 390 sends the command to do this.

Line 400 prompts you to to turn on the time delay reflectometry pulse and line 410 sends the command to turn on the TDR pulse for Channel 1.

Lines 420 and 430 change the graticule X-axis scaling to feet.

(cont.)

Program Listing (cont.)

```
440 INPUT "Press Enter to calibrate CHM1 for TDR amplitude",A$
450 PRINT #1,"CALM1 TDRAMPLITUDE:350E-3"

460 INPUT "Press Enter to send AUTOSET",A$
470 PRINT #1,"AUTOSET START"

480 INPUT "Press Enter to put CROSS in the measurement list",A$
490 PRINT #1,"MSLIST1 CROSS"

500 INPUT "Press Enter to set the measurement parameters",A$
510 PRINT #1,"MPARAM1 MLEVELMODE:ABSOLUTE,REFLEVEL:-300E-3"

520 INPUT "Press Enter to save one timing value to memory",A$
530 PRINT #1,"REFSET1 CURRENT:CROSS"

540 INPUT "Press Enter to turn COMPARE ON",A$
550 PRINT #1,"COMPARE ON"

560 INPUT "Press Enter to adjust the reference level",A$
570 PRINT #1,"MPARAM1 REFLEVEL:-50E-3"

580 INPUT "Press Enter to ask the scope for the measurement",A$
590 PRINT #1,"MEAS1?;GRAT? XUNIT"
600 LINE INPUT #1,RESPONSE$
610 PRINT RESPONSE$

620 PRINT "Example Completed."
625 CLOSE 1
630 LOAD "MENU.BAS",R
```

Program Description (cont.)

Lines 440 and 450 calibrate TDR pulse amplitude for Channel 1 by sending the command to set TDR amplitude for Channel 1 to 60% (i.e., obtain a 250 mV TDR pulse amplitude).

Lines 460 and 470 AUTOSET Trace1.

Lines 480 and 490 add the CROSS measurement to the measurement list for Trace1.

Lines 500 and 510 select the measurement parameters for Trace1 by sending commands to set the measurement level mode for Trace1 to absolute, and the measurement reference level to minus 300 mV.

Lines 520 and 530 save the Trace1 current crossing value as the reference value.

Lines 540 and 550 turn on the measurement comparison mode.

Lines 560 and 570 set the reference level for Trace1 to minus 50 mv.

Lines 580 through 610 query the instrument for the current crossing measurement value and the X-axis graticule scaling for Trace1. This response, in feet, represents the distance the TDR pulse must travel— out the connector, through the cable, and back again. To find only the actual length of the cable, locate the start of the cable. Then measure the distance to the rising edge of the second step. Finally, take this value and divide by two.

Line 620 prompts you that this example is finished.

Line 630 formally ends the program.

Lines 620 and 625 prompt you that this example is finished and formally close the program. Line 630 returns you to the Examples Menu.

Status and Event Reporting

Status & Event Reporting

The CSA 803 and 11801A provide 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 instrument.

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 for 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	System status bits. The state of these four bits varies with the type of event that is reported.
2	
3	
4	
5	Busy bit. Asserted only when 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 instrument 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 instrument.

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 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 instrument 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 instrument to assert the SRQ signal line whenever a new event occurs and RQS is set to on. A GPIB controller may then interrogate the instrument with an IEEE STD 488 serial poll and obtain a status byte that describes the event that occurred.

When RQS is set to off, the only new event that will cause the instrument 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 instrument with an IEEE STD 488 serial poll to read the most recent status byte from the serial poll register of the instrument.

RQS for RS-232-C service requests—is always set to 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 instrument to determine the latest status condition that has occurred in the instrument.

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 event will cause the instrument to request service.

However, in addition to polling the instrument 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 GPIB/RS232C pop-up menu in the front panel Utility major menu, or the or the RS232 VERBOSE command.

When VERBOSE is set to ON, each command sent to the instrument 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 four normal conditions defined:

- **No Status To Report** reports when there is no event or device dependent status to report.
- **Power On** reports when the instrument 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.

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 instrument malfunctions.
- **Execution Warning** reports when the instrument is operating, but you should be aware of possibly inaccurate results.
- **Internal Warning** reports when the instrument 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		DECIMAL	
	Status Bits		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
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 instrument 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	Channel number
%A	Argument name
%b	Mainframe indicator: M
%B	Mainframe verbose indicator
%C	Calibration request string: “Calibration due”
%O	Option description string (e.g., “Prescaler deleted”
%t	Six words of data describing a mainframe malfunction intended for use by field and factory service.

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*> CURVE <*bblock*>	Binary block byte count of binary settings returned to the instrument 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.
		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, "M1 +", or attempted to create non-acquired trace component (for example, STO <*ui*>, <*NRx*>, or combinations) on WIN time base.
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.
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.

Command Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
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.
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.

Execution Errors

Event Code	Description	Commands that Generate Code	Explanation
200	**No masks defined**	MASK, MASKSTAT	No masks defined for mask command
201	**Display type not color graded**	DISPLAY	User tried to query display data when not in Color graded.
202	**DC coupling not allowed**	TRIGGER	User tried to set trigger coupling. Not supported on CSA 803.
203	**I/O buffers full**		Both input and output buffers are full. Output buffer is cleared.
204	**No selected measurement**	MEAS	No measurement in the MSLIST.
205	**%A out of range – value ignored**	ABSTOUCH	Out-of-range ABSTOUCH argument.
206	**No mask point near enough to delete**		Front Panel only command.
207	**No such mask**	MASK	Mask<ui> is not defined.
208	**Illegal number of mask vertices**	MASK	Input incorrect number of mask vertices. Values not paired for XY point (ie. nrx1, nry1,nrx2, need an nry2), or 0 points or > 50 points.
209	**Mask point not on screen**	MASK	MASK point in input array is off screen.
210	**Illegal mask number**	MASK<ui>	User referred to mask<ui> where ui was less than 1 or greater than 10.
211	**Setting cursor 1 to zero not permitted on XY trace**	CURSOR SETZERO	Tried to set cur1 zero on an XY trace.
212	**Waveform not scaled in Rho units**	CURSOR	User specified split cursors between Rho and non-Rho wfms.
213	**Set zero Rho allowed only with Rho units**	RHOZERO	Waveform not scaled in Rho.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
214	**Invalid filter argument**	TRACE	Filtering argument in trace command incorrect.
215	**Illegal color number**	COLOR, HPGL, TEK4692, TEK4696	Out-of-range color index.
216	**Unsupported printer function**	COPY	Format unsupported for currently selected printer.
217	**Not enough memory for alternate topline-baseline calculations**	TOPBASE	Out of memory for calculating topline–baseline in Alternate method
218	**No logical names defined**	DEFINE?	No logical names currently defined.
219	**Logical name not defined**	DEFINE?, UNDEF	User asked for a logical name that was not defined.
220	**WFMPRE links not allowed with multiple trace output**	WFMPRE, WAVEFRM	User had OUTPUT set to TRA<ui>TO-TRA<ui> or similar and asked for WFMPRE.
221	**That XY waveform has incompatible components**	TRACE	Waveform has calculated vs stored or other incompatible type.
222	**Record length too long for Non-Normal display type**	DISPLAY, HISTOGRAM, MASKSTAT	User changed display mode (infinite, variable, or graded) with record length > 512.
223	**Can not select Rho units**	GRATICULE	Tried to change to Rho units when either External Prescaler trigger or TDR not turned on.
224	**Selecting volts–baseline correction disabled**		
225	**Baseline calibration failed to find –1 rho point**		Baseline correction couldn't find –1 rho this calibration pass.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
226	**Rho scaling failed to fine −1 rho point-Can not calibrate rho scales**	AUTOSET	Could not scale waveform in Rho accurately.
228	**Waveform expression too long**	TRACE, ENV, AVG	Trace description exceeded 55 characters.
229	**No windows are defined**	WIN, WINNUM, WINLIST	No window waveforms are defined.
230	**Illegal window number**	WIN	WIN<ui>; ui out of range.
231	**No such window**	WIN<ui>	No such WIN<ui> trace is displayed.
232	**Software only measurements not allowed in current measurement mode**	<meas>, MPARAM	Measurement not allowed in Hardware mode
233	**Measurement not found in measurement list**	<meas>	Measurement not in measurement list (MSLIST)
234	**Calculated or stored traces cannot be measured in hardware mode**	<meas>, MPARAM, TRACE	Only actively acquired traces can be measured in the hardware mode.
235	**Waveform cannot be measured in software mode**	MSLIST	Some <meas> in list not allowed in software measurement mode
236	**No reference channel selected**	CURSOR, MPARAM	No reference channel selected for operation.
237	**Waveform is not a non-tracking autowindow**	WIN	Tried to do a locate or other auto window function.
238	**Window is not an autowindow**	WIN	Tried to do a locate or other auto window function.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
239	**Improper version number**	SET <*bblock*>	Version number of received binary settings block not the same as current firmware version number.
241	**Too many acquisitions**	TRACE<*ui*>	Trace definition would cause the instrument to acquire more than eight traces.
242	**Requested display type not allowed when calculating histograms, masking or statistical measurement**	DISPLAY TYPE	Cannot have Variable or Normal display type when acquiring histograms or masks. Cannot have any display type other than color graded when performing statistical measurements.
243	**That function is disabled by a hardware strap**	UID	Attempted to modify serial number.
244	**Histogram/mask system not active**	HISTOGRAM, MASK, MASK-STAT,STATHIST	Tried to manipulate Histograms or masks without DSYS being on.
245	**No stored waveforms**	LABEL? STO	No waveforms were stored when LABEL was queried or no free memory for stored waveforms
246	**Can't sequence settings**	RECALL	Attempted to sequence settings with SETSEQ OFF.
247	**No settings defined**	LABEL?,RECALL, SETSEQ	Attempted one of these commands when no stored settings exist.
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.

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, 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.
		MPRAM	Attempted to set or query a measurement parameter with no selected trace.
251	**Illegal trace number**	ADJTRACE, CLEAR, CURSOR, 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	Setting <*ui*> does not exist.
253	**Stored setting does not exist**	DELETE, LABEL, RECALL, STORE	Out-of-range <*ui*> argument with one of these commands.
254	**Histograms and masks are not allowed on calculated traces**	HISTOGRAM, MASKSTAT, TRACE	Histogram/mask turned on for calculated or stored trace.
255	**Out of memory**	STORE	Insufficient memory to store a trace or insufficient NVRAM to store settings.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
		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.
256	**Label not found**	CLEAR, DELETE, INPUT, OUTPUT, RECALL, REMOVE, SELECT, STORE	No matching label found with <*qstring*> syntax used.
257	**Illegal stored waveform number**	DELETE, INPUT, LABEL, OUTPUT, STORE, TRACE	Out-of-range STO <*ui*> argument for one of these commands.
258	**Duplicate label**	LABEL	Label specified already used.
259	**No labels defined**	LABEL	No labels defined for specified links.
260	**Label not defined**		No label is defined for front panel settings, stored, or acquired traces.
261	**Trace is not a window**	WIN	Trace referred to is not a window.
262	**Waveform must have acquired components**	VPCURVE, DISPLAY	The waveform is not active so Infinite, Variable, and Color Graded have no meaning.
263	**Illegal channel number**	CH <*alpha*> <*ui*>	Attempted to set parameters of sampling head channel that was out of range.
264	**No further XY waveforms may be defined**	TRACE<*ui*>	Attempted to define more than the maximum permissible number of XY traces.
265	**Illegal DATE/TIME**	DATE	Illegal date value or syntax specified.
		TIME	Illegal time value or syntax specified.
266	**DEF expansion overflow**	DEFINE	Expansion string overflowed internal expansion buffer.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
267	**Illegal DEF string**	DEFINE	Illegal logical name specified.
268	**Illegal DEF recursion**	DEFINE	Unacceptable DEFINE 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, 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.
		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.
		WFMPRE	WFMPRE? query attempted, OUTPUT referenced nonexistent stored waveform.
271	**No such DEF**	UNDEF, DEFINE	Argument not defined in current list of logical names.
272	**Channel %b%a is not capable of this function**	CH, CALIBRATE	Tried to adjust a function not supported by that sampling head channel.
273	**No such FPS**	DELETE	Attempted to delete undefined stored settings number.
		LABEL	Attempted to label or query undefined stored settings number.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
		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	**Automatic window mode not allowed for calculated traces**	WIN	Auto windows not allowed on calculated traces.
275	**%B not installed**		
276	**No variable persistence traces defined**	VPCURVE	Asked for Variable Persistence data when not in Variable Persistence display mode.
277	**No histogram selected for trace**	STATHIST	No histogram is defined on the selected waveform.
278	**Measurement system not active**	STATISTICS, <*meas*>?	A measurement query was sent when the Measure major menu was not selected. Use the MSYS ON command.
279	**Bad trace description**	TRACE<*ui*>	Trace description was contained unrecongizable arguments.
280	**Invalid smoothing argument**	TRACE<*ui*>	TRACE expression contains out-of-range SMOOTH argument.
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.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
283	**Can't clear nonacquired waveform**	CLEAR	Attempted to clear trace that has only stored trace components (for example, TRACE1 DESCRIPTION:"STO3") or is scalar (LOG(5)).
284	**Can't change setting for a tracking/ auto-window**	MPARAM	Tried to set top/base lines for auto/tracking window.
285	**Can not perform measurements on XY waveforms**	<meas>, MSYS, MPARAM	No automatic measurements are allowed on XY traces.
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.
288	**Inappropriate trigger level units**	TRIGGER	Improper LEVEL 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: ▪ Selected waveform is XY (regardless of measurement). ▪ Reference measurement specified as DUTY, FREQ, or PERIOD; no period can be found within specified measurement zone.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
290	**Current reference measurement failed (cont)**	REFSET	■ 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. ■ 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 PHASE and no period can be found within the reference waveform measurement zone. ■ Reference measurement specified is PHASE and valid mesial crossings cannot be found within the measurement zones of the reference and selected waveforms.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
290	**Current reference measurement failed (cont)**	REFSET	▪ Reference measurement specified is PDELAY and valid mesial crossings cannot be found within the measurement zones of the delayed and selected waveforms.
291	**TEXT not permitted when acquired XY trace is active**	TEXT	Attempted to place text on display when acquired XY trace is active.
292	**No sampling heads are installed in %B**	UID	Attempted to query the serial number of a sampling head channel that is nonexistent.
293	**No sampling heads are installed**	UID	Attempted to query the serial number of a sampling head channel that is nonexistent.
294	**Dual graticules not permitted with XY trace**	GRATICULE	XY traces allowed only on single graticule display.
295	**%O option not installed**	TRIGGER	Tried to set trigger source to Prescaler without prescaler installed.
296	**Nonacquired XY waveforms are not allowed in Non-Normal display type**	TRACE, DISPLAY	Non-acquired XY waveforms only allowed in Normal display type.
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.

Execution Errors (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
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 AVG.ENV, 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.
2000	**Statistical Measurements allowed only in Color-Graded mode**	<*meas*>, MPARAM	Attempted to query/set statistical measurements when display type is other than Color Graded.
2001	**Only one waveform per axis can be measured in Statistics mode**	TRACE DESCRIPTION, MSLIST, DISPLAY GRATICULE	Can measure only one waveform per axis when measurement mode is set to Statistical.
2002	**%0 not available**	TRIGGER, STD, RECALL, STORE, CALIBRATE, TRACE DESCRIPTION	Attempted to set/query an item that is not available/installed in the instrument.
2003	**Smoothing is not permitted when either channel in a head is in random data mode**	CH <*alpha*> <*ui*> SMOOTHING	Attempted to turn smoothing on for a channel when Random Data mode is turned on for **either** channel in the head. Smoothing and Random Data mode are incompatible.

Internal Errors

Internal errors are reported if the instrument 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
386	**Minor time base calibration problem: %a**		Time base couldn't calibrate to optimum *this* calibration pass.
387	**Time base calibration failed: %a**		Time base couldn't calibrate to optimum *this* calibration pass.
388	**Time base calibration failed at power-up: %a**		Time base couldn't calibrate to optimum *this* calibration pass.
390	**Time base system error: %t**		Time base internal error.
391	**Time base processor interrupt: %t**		Time base internal error.
393	**Acquisition memory fault: %a**		Acquisition internal error.
394	**Test completed and failed**	TEST	Self-tests or extended diagnostics completed and failed.
395	**Error detected in acquisition system #%a: %t**		Acquisition system detected an internal error.

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.

System Events

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	
441	Store constants complete on selected channel	USER	67	3	
442	Calibrate ONCE complete	USER	67	3	
443	Blowby calibration complete	USER	67	3	
444	Loop gain calibration complete	USER	67	3	
445	Delay adjust calibration complete	USER	67	3	
446	Delay adjust measurement on reference channel complete	USER	67	3	
447	Offset null calibration complete	USER	67	3	
448	TDR amplitude calibration complete	USER	67	3	
449	All calibration complete on selected channel	USER	67	3	
450	Conditional acquire completed	OPCMPL	66	2	CONDACQ
451	Front panel setting recall complete	OPCMPL	66	2	RECALL
452	Initialization complete	OPCMPL	66	2	INIT
460	Test completed and passed	OPCMPL	66	2	TEST
461	Calibration completed and passed	OPCMPL	66	2	TEST
462	Hardcopy complete	OPCMPL	66	2	COPY
463	Measurements complete	OPCMPL	66	2	
464	Autoset complete	OPCMPL	66	2	AUTOSET

System Events (Cont.)

Event Code	Description	SRQMASK	Status Bytes		Commands that Generate Code
465	**Locate complete**		70	6	WIN
466	**Color Graded traces cannot be output on an HPGL device**		70	6	COPY
467	**Offset Calibration Complete**				CALIBRATE *<alpha> <ui>*

Execution Warnings

Execution warnings are reported when the instrument 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.

Execution Warnings

Event Code	Description	Commands that Generate Code	Explanation
550	**%A out of range – limit set**	Any requiring numeric values	A value was included with a command or link argument that was out of range. The limit value nearest the requested value is used.
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	The window trigger source is set equal to main source.
554	**No trace defined – Autoset terminated**	AUTOSET	AUTOSET initiated with no traces defined and no signal source can be found (for example, no sampling heads are installed), 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 instrument and the checksum comparison did not fail.
556	**No active acquisitions – acquisition remains stopped**	ACQUISITION	Attempted to start acquisition when no traces are defined, or when no defined traces contain "active" components (as opposed to scalar and stored components).

Execution Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
557	**Hardcopy aborted**	COPY	COPY operation aborted.
558	**Blowby Calibration failed - can't find reference step**	CALIBRATE	
559	**Calibration failed - unable to achieve requested value on channel %b%a**	CALIBRATE	
560	**Front panel setting incomplete: Out of memory or change in configuration**	RECALL	
561	**Offset calibration failed - check channel terminations**	CALIBRATE	
562	**Delay adjust calibration failed - can't find transition**	CALIBRATE	
563	**TDR amplitude calibration failed - can't find TDR pulse**	CALIBRATE	
564	**Autoset failed, bad trigger level**	AUTOSET	
565	**Autoset vertical failed**	AUTOSET	Vertical AUTOSET algorithm detects signal whose DC component is larger than offset range of least-sensitive gain setting of channel.
566	**Horizontal size set to default**	AUTOSET	Couldn't accurately autoscale horizontal size.
567	**Retrace defaulted to selected trace**		
568	**Reference trace invalid**		

Execution Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
569	**Loop gain calibration failed – can't find reference step**		
570	**Delay adjust measurement on reference channel failed – Can't find transition**		
571	**Nothing to abort**	COPY	Attempted to abort with no copy in progress.
572	**Waveform record length(s) reduced to 512 for Non-Normal display type**	DISPLAY, HISTOGRAM, MASKSTAT	Record lengths greater than 512 allowed only in normal display type.
573	**Persistence time reduced due to lack of memory**	STORE	User stored trace or FPS causing memory to be used.
574	**All front panel settings currently defined – setting not saved**	STORE <*qstring*>	All 10 FPS used, so user has to either delete or STORE FPS<ui>.
575	**Incompatible firmware versions**	ID?	Subsystem firmware versions installed are incompatible.
576	**Smoothing has been turned off for both channels in the head; smoothing is incompatible with random data mode**	CH<*alpha*> <*ui*> DATATYPE	Setting data type to random data causes smoothing to be turned off for both channels in head.

Internal Warnings

Internal warnings are reported when a problem has been detected. The instrument 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.

Internal Warnings

Event Code	Description	Commands that Generate Code	Explanation
653	RS-232 input parity error		
654	RS-232 input framing error		
655	RS-232 input buffer overrun		
656	Internal table search failed		
657	Nonvolatile RAM completely reset – probable battery failure	*<Powerup>*	
658	Nonvolatile RAM front panel settings lost – instrument ID data retained	*<Powerup>*	
661	Channel %b%a powered up during mainframe operation – Cycle power to utilize		
662	Channel %b%a powered down during mainframe operation – Cycle power to continue		
663	Change in channel %b%a configuration		
664	Channel %b%a was not maintained at desired calibration delay value		

Internal Warnings (Cont.)

Event Code	Description	Commands that Generate Code	Explanation
665	**Teksecure Erase Memory Status: Erased; Instrument ID, on-time, and number of power-ups retained**	*<Powerup>*	
666	**Nonvolatile RAM front panel settings and saved trace descriptions last; Instrument ID data retained**	*<Powerup>*	

Appendices

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.

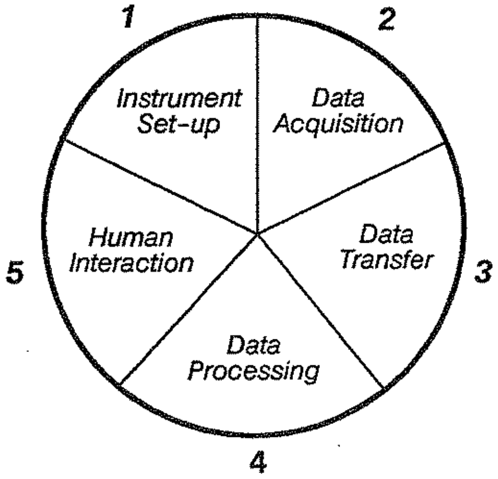
First, you must be familiar with your instrument controller, measurement instruments, data recorders, and with your chosen software (operating system, programming language, device drivers, etc.). When you know the capabilities of your system, you are better prepared to write efficient application programs.

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 instrument 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 instrument to check whether any settings associated with the command function have changed prior to the command. If any associated settings have changed, the instrument must load the new settings into its hardware.

The second part of the setup time is the time it takes the instrument to settle to the specified setting. For example, when vertical size is set automatically, the instrument 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 instrument 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 they are settling, 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.

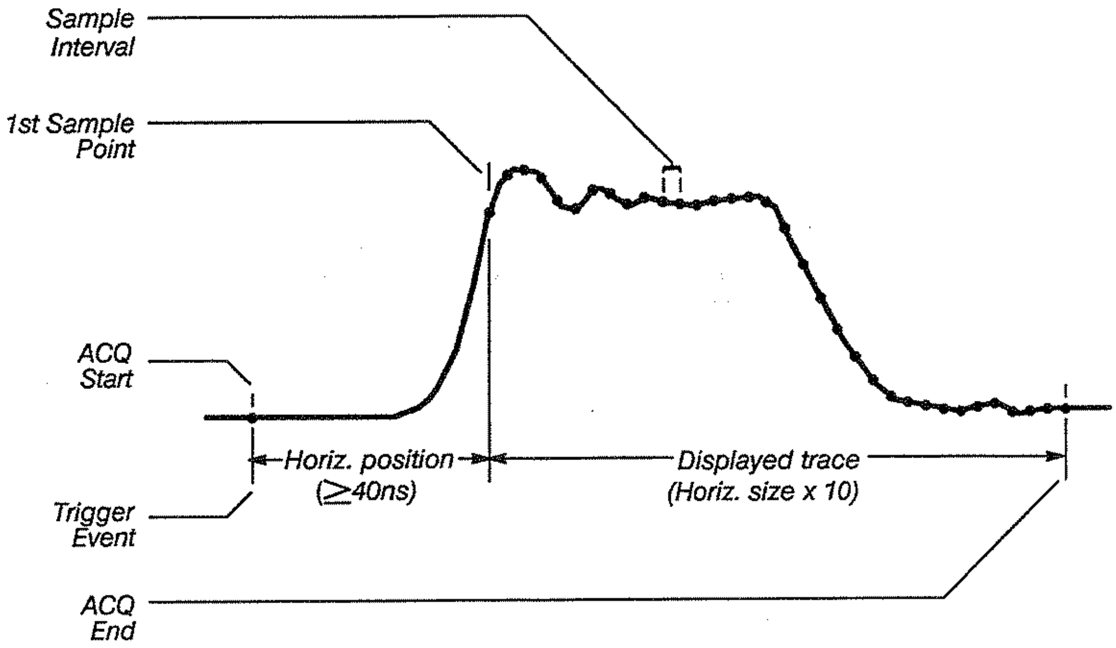
Data Acquisition Time

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.

For sequential mode digitizing, the significant factors that affect the digitizing time are the frequency of the trigger events, and the horizontal size and record length selection.

The CSA 803 and 11801A sample the incoming waveform every five microseconds (minimum sampling interval between each data point), or every valid trigger (equal to the trigger period).

The digitizing time is equal to the number of samples taken (equal to the record length) times the sample interval or the right-most dot/sample horizontal position (i.e, acquisition end).



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 trigger, if possible.
- 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 instrument 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 instrument 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 (instrument 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 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 instrument to manipulate the data, plus the time required by the controller to further process the data. The instrument can deliver raw, semi-processed data, or completely processed data, depending on the requirements of the application. The processing speed of the instrument 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 takes 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 instrument 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 the 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 instrument 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 CSA 803 and 11801A predefined command words, including headers, links, and arguments.

In this section, reserved words appear in mixed case, with the required minimum leading substring in uppercase. Any leading substring of the full reserved word will be accepted as input, so long as the minimum string is given.

A

A
ABBwfmpre
ABOrt
ABSLevel
ABSolute
ABSTouch
AC
ACQNum
ACQuisition
ACTive
ADJtrace
ALL
ALLFps
ALLSTD
ALLSTO
ALLTrace
ALL_wavfrm
ALTErnate
ALTinkjet
ASCii
ATTenuation
AUTO
AUTOSet
AVG
AVG.env

B

B
BASeline
BAUd
BCOrrection
BINary
BINHex
BIT/nr
BITMap
BLOwby
BN.fmt
BOTh
BYPassed
BYT/nr
BYT.or

C

C
CALibrate
CENter
CH
CHKsm0
CLEar
CMDerr
COLor
COMpare
CONDacq
CONFig
CONTinuous
COPy
COUnt
CR
CRLf
CROss
CRVChk
CRVId
CSTore
CUR1
CURRent
CURSor
CURVe
C.WINBottom
C.WINLeft
C.WINRight
C.WINTop

D

D
DADj
DAFiltering
DAInt
DAMeasref
DATACompress

DATE
DATRdy
DC
DCOmp
DEBug
DEF
DEFAult
DELAy
DELete
DEScription
DIAg
DIRection
DISPlay
DISTal
DIThered
DIV2
DIVS
DOT1Abs
DOT1Rel
DOT2Abs
DOT2Rel
DOTs
DRAft
DREcall
DSYmenu
DUAl
D.WINBottom
D.WINLeft
D.WINRight
D.WINTop

E

ECHo
EDGe
EMPty
ENCdg
ENH_accuracy
ENV
EOL
EQ
ER
EVEN
EVENT
EXErr
EXTCoupling
EXTernal
EXWarn

F

FAIled
FALltime
FASt
FEEt
FEOi
FILtering
FLAgging
FORCe
FORMat
FPAnel
FPNext
FPS
FPSList
FPSNum
FPUpdate
FREq

G

GPIb
GRADed
GRAticule
GRLocation
GT

H

H1Bar
H2Bar
HARd
HBArs
HIPrec
HIRes
HISTogram
HIST.pt
HISTScaling
HMAg
HORiz
HPGl
HPOsition
HREfpt
HW

I

ID
IEEE
INActive
INChes
INErr
INIt
INPut
INTENsity
INFinite
INTernal
INWarn

K

KILl

L

LABel
LABRel
LEFt
LENgth
LEVel
LF
LFCr
LINear
LMOde
LMZone
LOCate
LOG10
LONgform
LOOpgain
LOWer
LSB
LT

M

MAIn
MAINPos
MANual
MASK
MASKStat
MAX
MAXTranum
MEAN
MEAS
MESial
METers
MFIltering
MID
MIN
MINUs
MLEvelmode
MMOde
MODe
MPAram
MSB
MSLIst
MSLOpe
MSNum
MSYs
MTRack
MTRANs

N

N
NAVg
NCOunt
NCUrrent
NENv
NF
NGRAded
NHISt.pt
NONe
NORmal
NOTrg
NR.pt
NT
NULl
NVRam
NWFm
NWFMS

O

ODD
OFF
OFFSet
OHMS
ON
ONCE
ONUll
OPCmpl
OPTional
OQUal
OR
OUTput
OVErshoot

P

PAIred
PANzoom
PARity
PASsed
PCTg
PDElay
PERiod
PERSistence
PHAse
PIN8
PIN24
PLUs
PORt
POS
POWeron
PP
PREscaler
PRInter
PROPvelocity
PROXimal
PT.fmt

R

READout
RECall
RECOrd

REDuced
REFAmplitude
REFBaseline
REFErence
REFFiltering
REFLEvel
REFLMzone
REFMesial
REFRmzone
REFSEt
REFSLope
REFSNratio
REFTOpline
REFTRace
REFXsition
RELative
REMAining
REMove
RESET
RHO
RHOFactor
RHOPos
RHOZero
RI
RIGht
RISetime
RMS
RMSDiv
RMZone
RQS
RS232
RUN

S

SAMid
SCReen
SEConds
SELect
SENsitivity
SET
SETSeq
SETZero
SIGMA1
SIGMA2
SIGMA3
SINgle
SLOpe
SMOothing
SNRatio
SOFt
SOUrce
SPEaker
SPLit
SRQmask
STARt
STAT
STATIstics
STATUs
STByte
STDDev
STO
STOList
STONum
STOP
STOPBits
STORe
STORE_recall
STRing
SW
SYSmon

T

TBCalmode
TBMain
TBWin
TDRAmplitude
TDRDelay
TDRPolarity
TDRState
TEK4692
TEK4696
TESt
TEXt
TF
TIMe
TO
TOFfset
TOPBase
TOPline
TOTal
TRAce
TRACK
TRAList
TRANS
TRANUm
TRG
TRIgger
TRSep
TYPe

U

UID
UN
UNDEF
UNDershoot
UNDO
UPPer
UPTime
URECall
USEr
UTIlity

V

V1Bar
V2Bar
VARiable
VBArs
VECtors
VERBose
VERt
VOLts
VPCurve
VPOsition
VSIze

W

WAVfrm
WFId
WFMCalc
WFMPre
WFMScaling
WHOle
WIDth
WIN
WINList
WINNum

X

X
X1
X10
X2
X5
XCOord
XDIv
XINcr
XMUlt
XQUal
XTNd
XUNit
XY
XZEro

Y

Y
YCOord
YDIv
YMUlt
YQUal
YTEnergy
YTMns_area
YTPls_area
YUNit
YZEro

Z

ZEROPoint

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 instrument 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 *CSA 803 and 11801A Command Reference*.

The character-set tables begin on the following page.

ASCII Character Set

	0	1	2	3	4	5	6	7
0	0 NU	16 DL	32	48 0	64 @	80 P	96 `	112 p
1	1 SH	17 D1	33 !	49 1	65 A	81 Q	97 a	113 q
2	2 SX	18 D2	34 "	50 2	66 B	82 R	98 b	114 r
3	3 EX	19 D3	35 #	51 3	67 C	83 S	99 c	115 s
4	4 ET	20 D4	36 $	52 4	68 D	84 T	100 d	116 t
5	5 EQ	21 NK	37 %	53 5	69 E	85 U	101 e	117 u
6	6 AK	22 SY	38 &	54 6	70 F	86 V	102 f	118 v
7	7 BL	23 EB	39 '	55 7	71 G	87 W	103 g	119 w
8	8 BS	24 CN	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 SB	42 *	58 :	74 J	90 Z	106 j	122 z
B	11 VT	27 EC	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 ▒

Escape Character Set

	0	1	2	3	4	5	6	7
0	0 Ă	16 Ñ	*Color 1 32	48	64 Π	80 π	96 ↓	112 ---
1	1 ă	17 ñ	*Color 2 33	49	65 α	81 θ	97 ↑	113 Ä
2	2 Ö	18 ¿	*Color 3 34	50	66 γ	82 ρ	98 →	114 🄴
3	3 ö	19 ¡	*Color 4 35	51	67 δ	83 Σ	99 ←	115 🅁
4	4 Ü	20 Ã	*Color 5 36	52	68 Δ	84 τ	100 ∫	116 🅃
5	5 ü	21 ã	*Color 6 37	53	69 ϵ	85 υ	101 ÷	117 ┘
6	6 à	22 À	*Color 7 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 X̄	79 Ω	95 Ξ	111 ≈	127 │

*These color indices can be used to produce colored text on the display with the TEXT command. The color indices correspond to the display colors defined with the COLOR<ui> command. See the Color Indexes table in the COLOR<ui> discussion in the *CSA 803 and 11801A Command Reference*.

To produce colored text, preceed text with one of these escape characters. The TEXT command allows multiple color changes per line of text. Refer to the *CSA 803 and 11801A Command Reference* for command syntax information.

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 (instrument service requests)
- String transfer to the instrument 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.

Interface Configuration

Set up the GPIB parameters of the instrument as follows:

CSA 803 or 11801A Interface Configuration

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

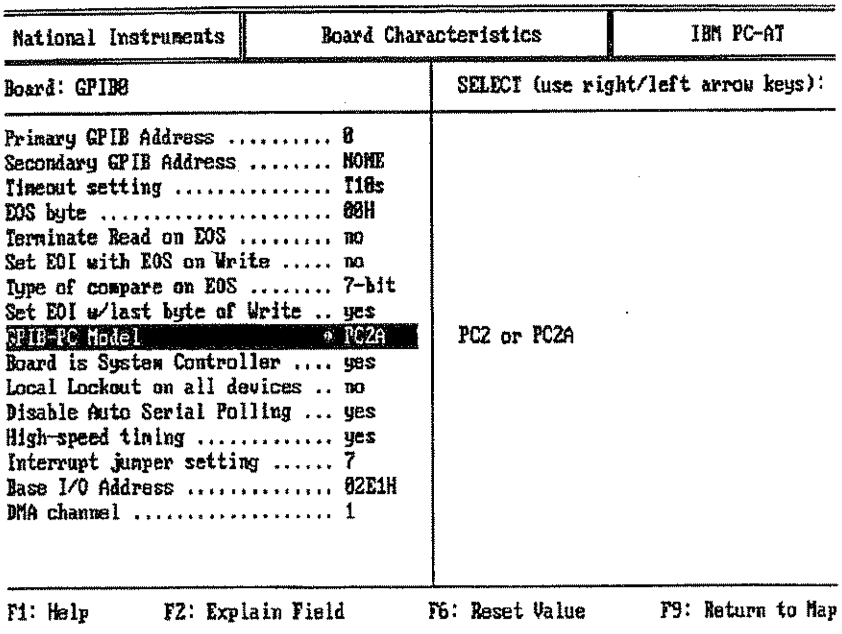
Computer Interface Configurations

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 (instrument) characteristics should be set.



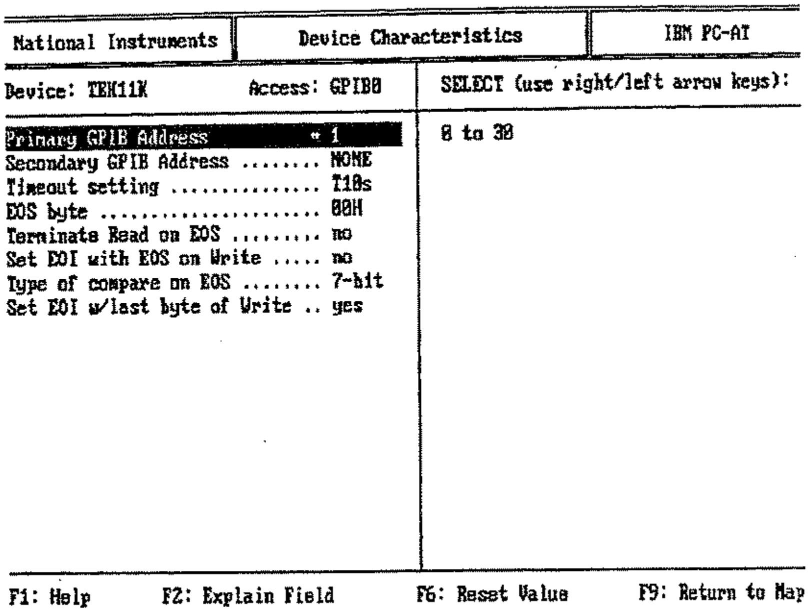
```
National Instruments    Board Characteristics                IBM PC-AT

Board: GPIB0                            SELECT (use right/left arrow keys):

Primary GPIB Address .......... 0
Secondary GPIB Address ........ NONE
Timeout setting ............... T10s
EOS byte ...................... 00H
Terminate Read on EOS ......... no
Set EOI with EOS on Write ..... no
Type of compare on EOS ........ 7-bit
Set EOI w/last byte of Write .. yes
GPIB-PC Model                 * PC2A    PC2 or PC2A
Board is System Controller .... yes
Local Lockout on all devices .. no
Disable Auto Serial Polling ... yes
High-speed timing ............. yes
Interrupt jumper setting ...... 7
Base I/O Address .............. 02E1H
DMA channel ................... 1

F1: Help        F2: Explain Field        F6: Reset Value        F9: Return to Map
```

GPIB Driver-Board (GPIB) Settings



```
National Instruments   Device Characteristics          IBM PC-AT

Device: TEK11K       Access: GPIB0   SELECT (use right/left arrow keys):

Primary GPIB Address          = 1    0 to 30
Secondary GPIB Address ........ NONE
Timeout setting ............... T10s
EOS byte ...................... 00H
Terminate Read on EOS ......... no
Set EOI with EOS on Write ..... no
Type of compare on EOS ........ 7-bit
Set EOI w/last byte of Write .. yes

F1: Help      F2: Explain Field      F6: Reset Value      F9: Return to Map
```

GPIB Driver-Device (CSA 803 or 11801A) 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 instrument 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%, "MSLIST1 PER,FREQ,MAX,PP,RISE,
 FALL;MEAS1?")
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 CSA803/11801A
(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 CSA 803/11801A 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;"MSLIST1 PER,FRE,MAX,PP,RISE,FALL"
40 OUTPUT @Br;"MEAS1?"
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 CSA803/11801A"
 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 CSA 803/11801A 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 CSA 803 and 11801A with a brief description of each.

GPIB Functions

Interface Function	Subset	Meaning
Acceptor Handshake	AH1	The instrument can receive multi-line messages across the GPIB from other devices.
Controller	C0	No Controller capability; the instrument cannot control other devices.
Device Clear	DC1	The instrument can respond both to the DCL (Device Clear) interface message, and to the Selected Device Clear (SDC) interface message when the instrument is listen-addressed.
Device Trigger	DT0	No Device Trigger capability; the instrument does not respond to the GET (Group Execute Trigger) interface message.
Electrical	E2	The instrument uses tri-state buffers, which are optimal for high-speed data transfer.
Listener	L4	The instrument becomes a listener when it detects its listen address being sent over the bus with the ATN line asserted. The instrument 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 instrument 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.

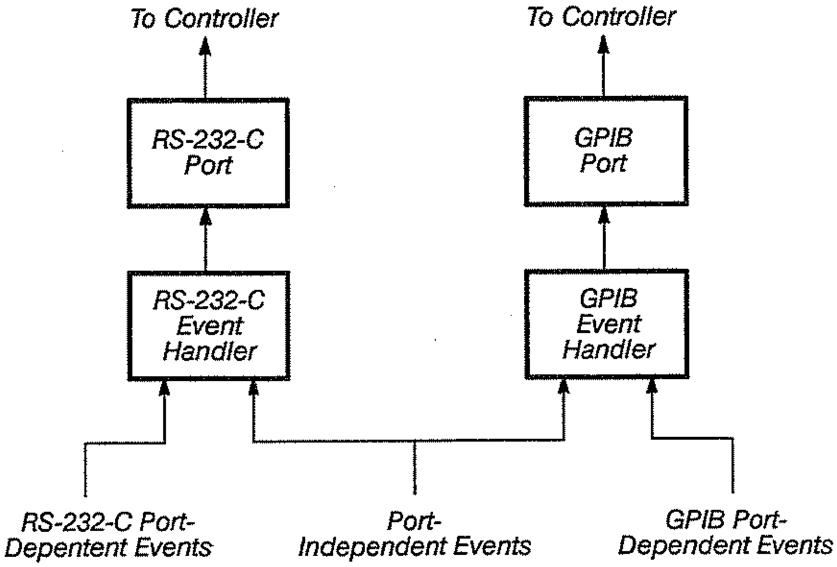
GPIB Functions (Cont.)

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

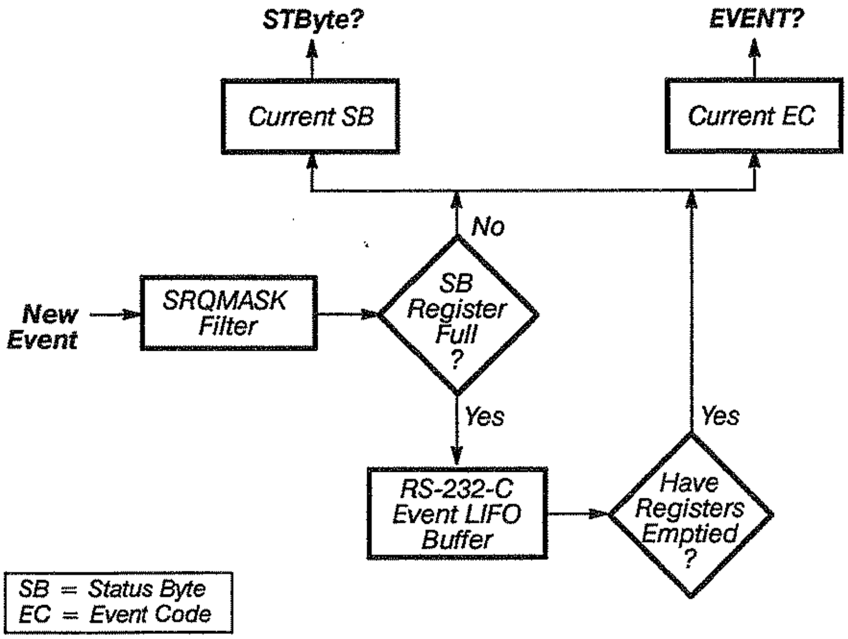
Since more than one event may occur before a GPIB or RS-232-C controller can respond to a service request, the instrument uses the following priorities to report events.

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 instrument checks the SRQMASK for that event. If the SRQMASK is off, the event is discarded. If the SRQMASK is on, the instrument 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).

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

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

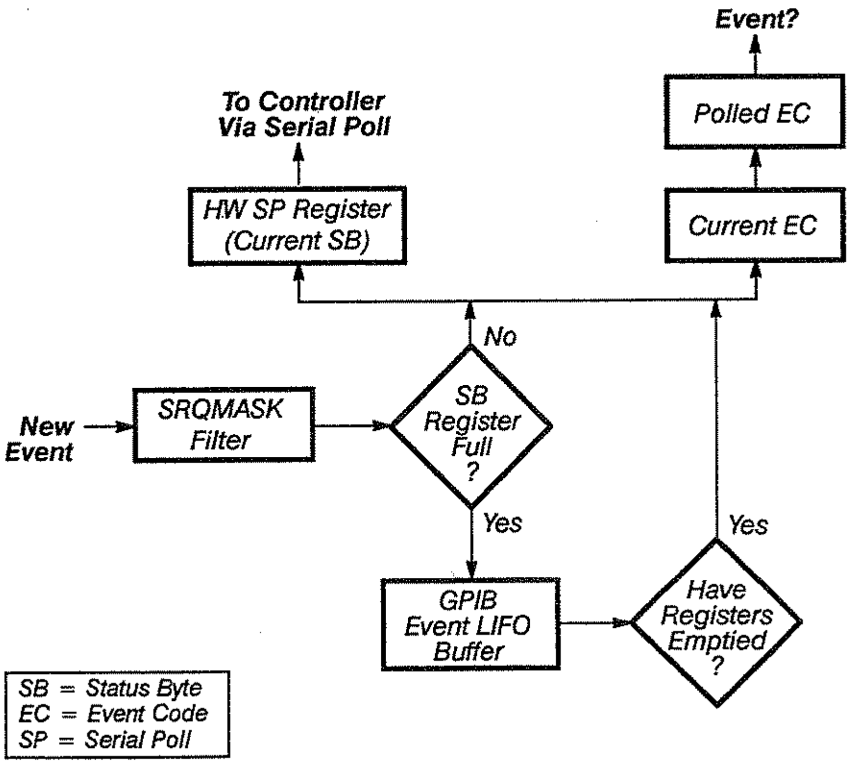
Event Reporting When GPIB RQS is Off

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 instrument 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 instrument 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 instrument 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 instrument asserts SRQ, an external controller must first serially poll the instrument to read the status byte of the system event that just occurred. The instrument responds to the serial poll by moving the current event code register contents into the polled event code register. The instrument next checks for pending events in the LIFO buffer. If found, the instrument moves the status byte of the top event into the hardware serial poll register, thus updating it and causing the instrument 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 instrument 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.

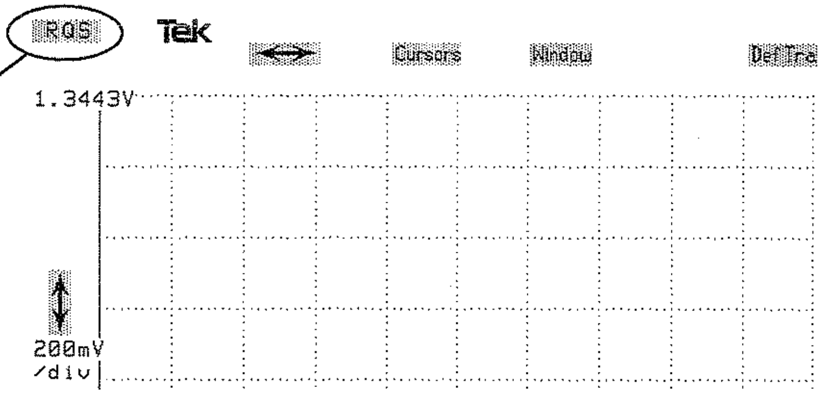
If a controller sends an EVENT? following the serial poll, the instrument returns the contents of the polled event code register and initializes it to the NULL event. Then, at the next serial poll, the instrument again moves the contents of the updated current event code register into the polled event 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 instrument and then, if more information is desired, follow up with an EVENT? query.
- When EVENT? returns the NULL event, the instrument 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 instrument 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 instrument 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 instrument 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 instrument takes the same actions as it did for the GPIB interface.

Events Reported at Instrument Power-On

When the instrument is powered on, diagnostic tests automatically execute (unless bypassed with hardware straps). When diagnostics complete, nondiagnostic firmware in the instrument takes over and the remote interfaces are activated. The instrument 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 instrument 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 instrument has been powered on, and the length of time the instrument has been powered on.

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

Glossary

Glossary

Acquisition

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

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 instrument set itself to provide a stable and meaningful display of a given trace.

Averaging

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

Baseline Correction

The process of maintaining the displayed vertical placement of a trace, correcting for changes in the signal levels that would ordinarily move the trace up or down.

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

A place to connect a signal or attach a network or transmission line to sampling heads. Also, the smallest component of a trace expression.

Channel Number

The number assigned to a specific signal input connector. The top channel of the left-most sampling head compartment of the mainframe is always mainframe channel 1, regardless of any repositioning or omission of sampling heads.

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).

Color Graded

A display mode in which the instrument displays regions of a trace in different colors according to the densities of displayed points (based on multiple acquisitions of the trace) in that region.

Complex trace

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

Concatenate

To link commands together.

Cursor

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

DCE

Acronym for data communications equipment. The instrument 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 trouble-shooting.

Default Measurement Parameter

A value from the default set of measurement parameters. The operator can change the default values. Whenever a trace 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 instrument.

Distal

The point farthest (most distant) from a reference point. As used in the instrument, the ending measurement point for some 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 instrument 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 trace 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.

Floating Point Value

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

GPIB (General Purpose Interface Bus)

The GPIB interface is an eight-bit parallel bus that allows remote computer control of the instrument and other synchronous devices. GPIB characteristics are specified in IEEE STD 488 1978.

Hardware Measurement

An automated measurement that is captured by special circuitry that monitors signals directly, as opposed to software measurements that are derived from acquired trace samples.

Histogram (Display)

A statistical function, selected from the Display Modes major menu, that displays a graph of the statistical distribution of acquired sample points. The histogram graph may be calculated for either the horizontal values or vertical values of the sample data. The histogram is performed on the selected trace within a window defined by the user with the the Histogram Limits.

Histogram (Measurements)

The instrument performs a histogram algorithm on a trace to determine some measurement parameters, such as topline and baseline. This measurement histogram is not directly controllable by the user.

Horizontal Reference Point

The point about which the trace is expanded or contracted when horizontal size adjustments are made. The horizontal reference point remains anchored as the rest of the trace grows or shrinks around it.

IEEE STD 488

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

Infinite Persistence Mode

A mode of operation where the instrument displays newly acquired trace data points and keeps the previously acquired data points on the screen.

Initialize

Setting the instrument to a completely known set of default conditions.

Internal Clock

A trigger source that is synchronized with the Calibrator signal.

LIFO

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

Loop Gain

Adjusts the capability of the sampling head to respond to large changes between samples. Excessive loop gain causes the sampling head to overshoot large transitions, while insufficient loop gain causes the sampling head to undershoot transitions.

LSB

Acronym for least significant bit or least significant byte.

Main

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

Major Menu

The menu that is displayed at the bottom of the screen alongside the Knob menu. One of the several major menus is always displayed.

Mask

A displayed, user defined polygon with up to 50 vertices used for Mask Testing.

Mask Testing

Testing of the selected trace against one to ten user defined masks. Sample points falling within the masks are added to a count which may be used to control acquisition.

Measurement

An automated numeric readout that the instrument provides directly from the displayed trace in real time, without operator intervention.

Measurement Parameter

One of several control/command parameters that the instrument 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 trace.

Mesial

The middle point of a range of points. As used in the instrument, 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.

Offset Nulling

Adjusts the DC voltage accuracy of a sampling head by nulling offset errors in the sampling head so that the DC output voltage equals the DC input voltage.

Parse

To decode or analyze data according to a syntax.

Persistence

The amount of time a data point remains displayed. There are four persistence modes available in the CSA 803 and 11801A: Normal, Variable, Infinite and color Grading.

Pixel

A visible point on the display. The display is 551 pixels wide and 704 pixels high. Each pixel may be set to one of eight predefined colors.

Proximal

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

Quoted String

An element of instrument 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 instrument character set.

Record Length

The number of samples (data points) that make up a trace.

RS-232-C

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

Sample Interval

The time interval between successive samples in a trace record.

Sampling Head

A high-performance amplifier unit that captures the incoming signal and reports the sampled data to the mainframe. Sampling head units can be selected for specifications that best suit your application.

Scalar

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

Selected Trace

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

Setting
The state of the front panel and system at some given time.

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

Smoothing
Processing applied by the sampling head prior to the digitization of a trace, to reduce apparent noise. With smoothing, the sampling head samples the signal 8 times instead of once, and the average of the samples is then used by hardware measurements and the digitizing circuitry.

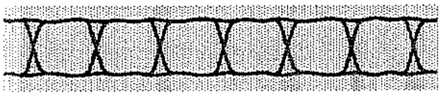
Software Measurement
An automated measurement that is derived from acquired trace samples, as opposed to hardware measurements that are captured by special circuitry that monitors the signals directly.

Stored trace
A trace 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.

TDR
Acronym for time-domain reflectometry. TDR is a method of characterizing a transmission line or network by sending a signal from one end and monitoring the electrical reflections. The CSA 803 and 11801A are compatible with TDR technology when used with a TDR-compatible sampling head.

TDR Amplitude
A sampling head adjustment for the amplitude of the TDR pulse.



Tektronix Codes and Formats

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

Time Base

The time-dependent specifications that control the acquisition of a trace. 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.

Trace Expression

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

Trace Number

A number assigned by the instrument to identify a trace. Display traces are numbered 1 through 8. A new trace is always given the lowest available number.

Tracking

The process of automatically adjusting the measurement parameters or window position to reflect changes in the trace.

Trigger

An electrical event that initiates acquisition of a trace 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).

Variable Persistence Mode

A mode of operation where the instrument displays newly acquired trace data points and keeps the previously acquired data points on the screen for a specified duration.

Waveform

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

Waveform Preamble

A response returned from the WFMPRE? query that contains the scaling information for the trace. A trace 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 trace that represents a horizontally expanded portion of another trace.

XY trace

A trace where both horizontal and vertical position of the data points reflect signal data.

Yt trace

A trace where the vertical position of the trace data points reflects signal data, and the horizontal position of the trace data points reflects time.

Index

Index