Tektronix

RTD 710A
Instrument Interfacing Guide

070-7207-00

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

First Edition: October 1988

Printed in the U.S.A.

Tektronix, Inc., P.O. Box 1000, Wilsonville, OR 97070–1000

WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of one (1) year from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

TABLE OF CONTENTS

SECTION 1 INTRODUCTION TO THE RTD 710A

SECTION 2 COMMUNICATING WITH THE RTD 710A

SECTION 3 GPIB COMMAND REFERENCE

SECTION 4 ADVANCED GPIB INFORMATION

SECTION 5 SRQ AND EVENT CODES

SECTION 6 RTD 710A PROGRAM EXPLANATIONS/LISTINGS

APPENDIX A ASCII & IEEE (GPIB) CODE CHART

LIST OF ILLUSTRATIONS

INTRODUCTION TO THE RTD 710A

The purpose of this document is to:

- Familiarize new users with the RTD 710A vertical, timebase, triggering, and waveform memory management capabilities.
- Provide a catalogue of commands, SRQ codes, and event codes for programmers using the RTD 710A on the General Purpose Interface Bus, or GPIB (IEEE 488 Standard Bus).
- Explain the use of the RTD 710A on the GPIB, and supply example programs for the following computers:
- The Tektronix 4041 System Controller
- The Tektronix SPS BASIC system
- HP 200 Series controllers
- The IBM PC, using BASICA and GURU II GPIB software package.

The GPIB/RTD 710A program examples show coding for:

- Acquiring a waveform and scaling it into a voltage array
- Reading and decoding breakpoint locations and sample rates
- Handling a Service Request (SRQ) and reading the Event code from the RTD 710A
- Graphing an acquired waveform onto a monitor

Vertical Section

The RTD 710A uses a 10-bit, dual-stage, flash conversion process to convert the analog signal to digital form. The converter provides 1024 levels of resolution.

Vertical accuracy is improved by making better use of the available voltage conversion range. The input range selection offers finer increments than the 1-2-5 sequence usually found on oscilloscopes and digitizers. The RTD 710A uses instead a 1 - 1.25 - 1.6 - 2 - 2.5 - 3.2 - 4 - 5 - 6.2 - 8 - 10 sequence.

An Auto-calibration feature also improves the accuracy of the digitized data. This function automatically compensates for zero-volt, gain, and phase errors in the input amplifier and A/D converter. With this compensation, the RTD 710A delivers 0.4% gain accuracy for range gain and drift. Auto-calibration is invoked when:

a. The voltage range changes,

b. The operator pushes the AUTO CAL button, or

c. The AUTocal command is received via the GPIB.

The RTD 710A vertical section lets you add dc offset to the input signal. Dc offset lets the RTD 710A use a more sensitive range to measure the top or base of a waveform (for example, overshoot and undershoot). You can vary the dc offset by 199% of the selected voltage range in 1% increments. You can program the dc offset in percent of full scale or volts, from the front panel or over the GPIB.

Record Modes

The RTD 710A offers four types of Recording modes (also called 'Waveform Acquisition' modes): Normal, Average, Envelope, and Auto-advance.

Normal is the basic recording mode. In this mode, the digitizer acquires a full waveform record and, depending on the trigger mode, either freezes the memory or re-acquires a waveform on the next trigger event.

Average recording mode causes successive waveform records to be averaged from 2 to 16384 times. If the trigger mode is Auto or Normal, the RTD 710A averages continuously and updates the acquisition memory with the latest averaged data.

When the selected number of averages is reached, the RTD 710A resets the waveform record and restarts the operation.

If the digitizer is in Single trigger mode, it acquires the specified number of averages, then enters the Hold state. If the trigger mode is Compare, the RTD 710A completes the specified number of averages and then compares against a reference waveform.

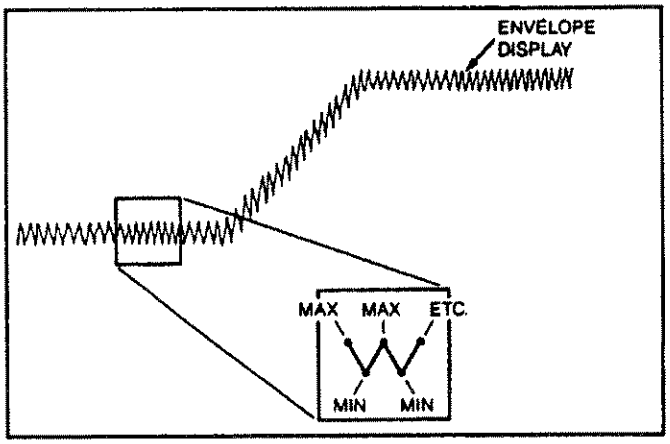


Fig. 1-1 Example of envelope waveform display.

Envelope mode continuously envelopes the waveform being digitized. Enveloping constructs a waveform from the minimum and maximum values of the sampled data points. Enveloping shows many waveform aberrations that might otherwise be missed, such as frequency and voltage drift.

If the trigger mode is Auto or Normal, the RTD 710A envelopes continuously and updates the acquisition memory with the latest envelope data. If the trigger mode is Single, the RTD 710A enters the Hold state after acquiring the specified number of envelopes. If the trigger mode is set to Compare, the number of envelopes is set to 1. Figure 1-1 shows an example envelope waveform display.

Auto-advance mode selects an acquisition mode that lets the RTD 710A automatically advance through successive waveform memory locations. This capability allows the digitizer to quickly capture many single-shot events along with their trigger information.

In Auto or Normal trigger mode, the RTD 710A will perform the Auto-advance recording sequence until you press the HOLD key. In Auto-advance mode, this configuration captures four single-shot events.

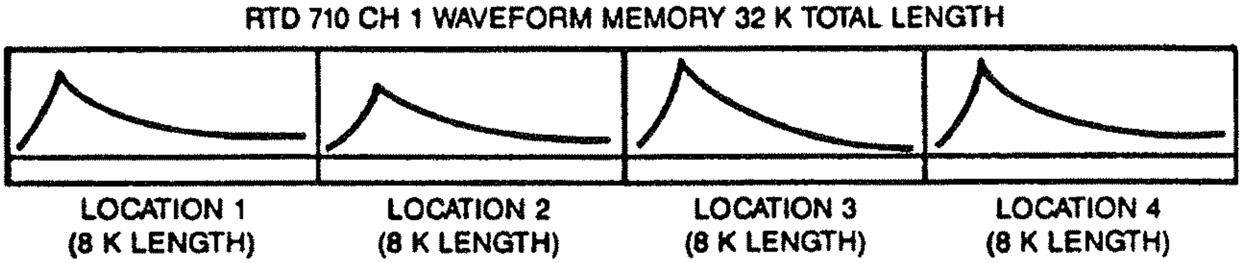


Fig. 1-2 Channel 1 memory segmented into four records.

Timebase Section

The timebase of the RTD 710A is very flexible to solve specialized needs. Some key features are:

- Up to 256, 1k-word acquisitions with short re-arm time
- External clocking from dc to 200 MHz
- Direct or divided timebase clock output for synchronizing with other equipment

You should consider the following four factors when setting up the RTD 710A timebase section:

- Record length
- Waveform locations
- Sample rate
- Sample rate switching (also called "breakpoints").

Record Length: The RTD 710A allocates 128K words for each channel. The waveform memory is very flexible; you can divide it into smaller record lengths (1K, 2K, 4K, 8K, 16K, 32K, 64K, 128K). The number of records available can vary from one record with 128K length to 128 records with 1K length each. In CH1 ONLY and High Speed modes, Channel 1 can be set to 256K word memory length.

Waveform Records and Locations: You can partition memory into smaller waveforms; up to 256 1K-word segments for CH1 ONLY vertical mode and High Speed sampling mode, or 128 1K-word segments in Dual Channel mode. Smaller records are better if the event to be captured does not require all memory available or if there are many sequential events to be captured prior to transferring waveforms.

Auto Advance. The RTD 710A has an Auto-advance mode that automatically advances through successive records of the waveform memory after each trigger. This mode allows the RTD 710A to quickly capture low (<5 ms rearm time, >200 events/sec) repetition rate events.

Sample Rate: The RTD 710A offers two Sample modes: Normal and High Speed. In Normal mode, the timebase has a maximum sample rate of 100 MHz for 10 ns resolution between points. In High Speed mode, the timebase's maximum rate is 200 MHz for 5 ns resolution between points. The sample rate can vary from the maximum rate of 5 ns down to 200 milliseconds. The RTD 710A can also be externally clocked from 200 MHz down to dc. Direct or divided timebase clock output is available on the RTD 710A. This feature is valuable for synchronizing with other equipment. The divided sample rate corresponds with the sample rate set on the front panel.

Sample Rate Switching (Breakpoints): The RTD 710A timebase can change the sample rate (the time between points) while acquiring a waveform. This feature lets you make the most efficient use of the available waveform memory by allowing the RTD 710A to switch sample rates (at breakpoints) from one speed to another. The cursors can be used to determine where to set the breakpoints.

Figure 1-3 shows one possible use of breakpoints. The objectives are to:

1) Acquire a waveform that has a 40 ns risetime, 500 ms pulse width, and a 100 ns falltime, and
2) Use a 1K word waveform location.

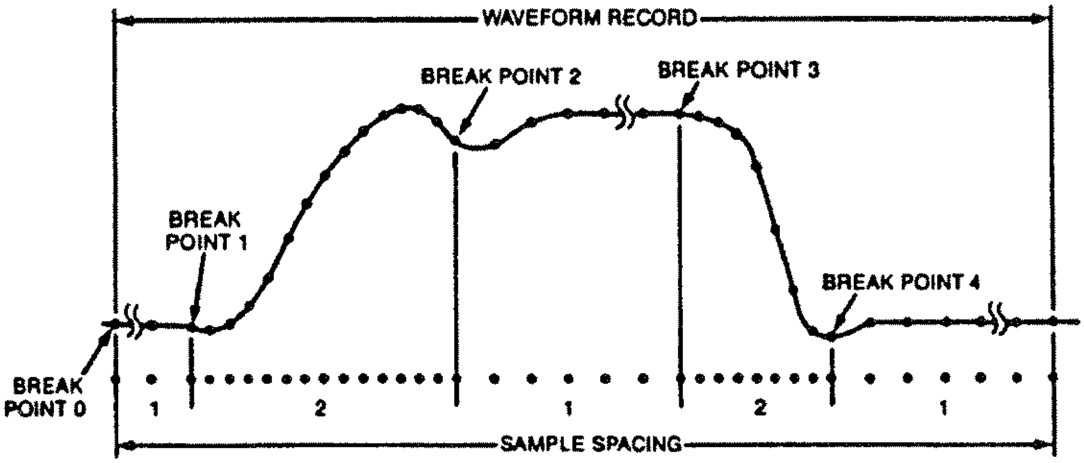


Fig. 1-3 Breakpoint waveform display.

Without breakpoints, this waveform could not be captured with adequate resolution of the rising and falling edges. Adequate resolution requires a record of about 50M words to capture the entire waveform at 10 ns/point. With sample rate switching, the sequence would be:

1) Start the sample rate at 10 ns/point until the rising edge is captured;
2) Slow the sample rate to 100 ms/point and then increase it to 10 ns/point to capture the falling edge.

The RTD 710A can perform these steps in a 1K-word waveform location without losing timing information such as pulse width. Use breakpoints only after the trigger event - no breakpoints before the trigger occurs. In Dual Channel mode, you can set a breakpoint 16 points after the trigger.In CH1 ONLY mode, you can set a breakpoint 32 points after the trigger.

Trigger Section

The RTD 710A has a very versatile trigger section. All the triggering modes that are standard on a digitizer or oscilloscope are offered. It also offers trigger modes that allow triggering on a signal that would be difficult (or impossible) to capture using the standard modes.

Trigger Delay: Pre-Trigger and Post Trigger

The RTD 710A's A/D converter continually converts the analog signal and fills memory. When a trigger event occurs, the RTD 710A can freeze information that occurred before the trigger event (pre-trigger) or it can accurately delay far past the trigger event (post-trigger).

The amount of pre- and post-triggering can be set using either TIME (time notation) or waveform data points (clock point notation, labeled PT on the RTD 710A front panel). You can set the trigger delay either remotely over the bus or by pressing the TRIG DELAY button and turning the knob to the desired pre- or post-trigger setting. Pre-trigger is designated with a minus (-) before the number.

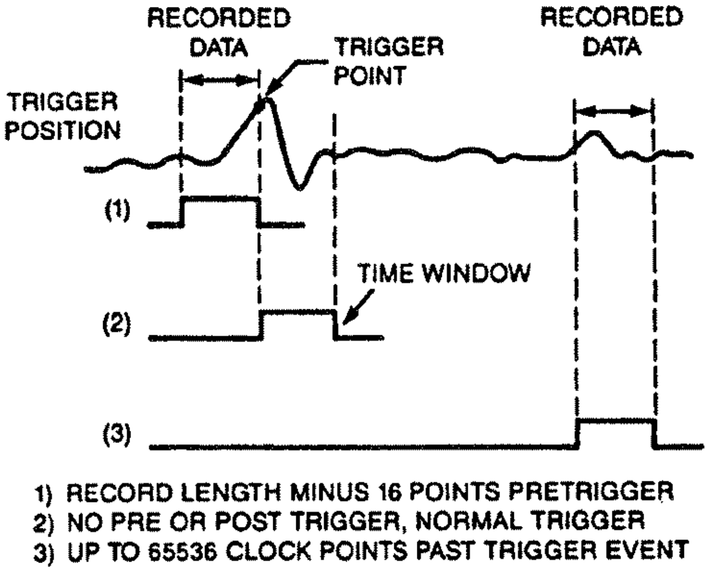

RECORDED DATA
TRIGGER POINT
RECORDED DATA
TRIGGER POSITION
(1)
(2)
(3)
TIME WINDOW
1) RECORD LENGTH MINUS 16 POINTS PRETRIGGER
2) NO PRE OR POST TRIGGER, NORMAL TRIGGER
3) UP TO 65536 CLOCK POINTS PAST TRIGGER EVENT

Fig. 1-4 Pre- and post-trigger illustration.

Pre-trigger: The pre-trigger range is the record length minus 16 data points in Dual Channel (normal sampling) mode. If CH1 ONLY acquisition is used, the number of pre-trigger data points is the record length minus 32.

When setting pre-trigger remotely, express the desired amount as a negative number in increments of 8 in Dual Channel mode or increments of 16 in CH1 ONLY mode. Determine time notation by multiplying the trigger delay by the sampling clock interval. For example, entering a -2000 with a sample rate of 5 ns for a Dual Channel acquisition with a record length of 4K words yields pre-trigger of 2000 times 5E-9 or 10us of pre-trigger. You can also directly enter pre-trigger in time notation. Figure 1-4 illustrates pre-trigger in relation to the trigger point.

Post-trigger: Post-trigger has a range of 0 to 256K clock points past the trigger event. You can enter post-trigger in time or clock point notation. Post-trigger is expressed as a positive number in increments of 8 for Dual Channel mode or 16 in CH1 Only mode.

For example, entering 10us of trigger delay yields 10us of post-trigger. Figure 1-4 shows post-trigger in relation to the trigger event.

Trigger Modes

Five trigger modes are available in the RTD 710A: Auto, Normal, Single, Compare-in, and Compare-out. The RTD 710A must be armed to recognize a trigger event.

In **Auto**, the recording function free runs, recording the base line in the absence of a triggering signal. The Auto triggering mode permits triggering on signals that have a 50 Hz or higher repetition rate.

In **Normal**, recording starts with the occurrence of a valid trigger signal. The RTD 710A waits until the trigger before storing waveform data.

In **Single**, one record is made after the trigger event occurs. After the recording ends, the RTD 710A enters the Hold state. In order to acquire a new waveform, you must press the HOLD/RESET key or reset the RTD 710A with the HOLD RESet command on the GPIB.

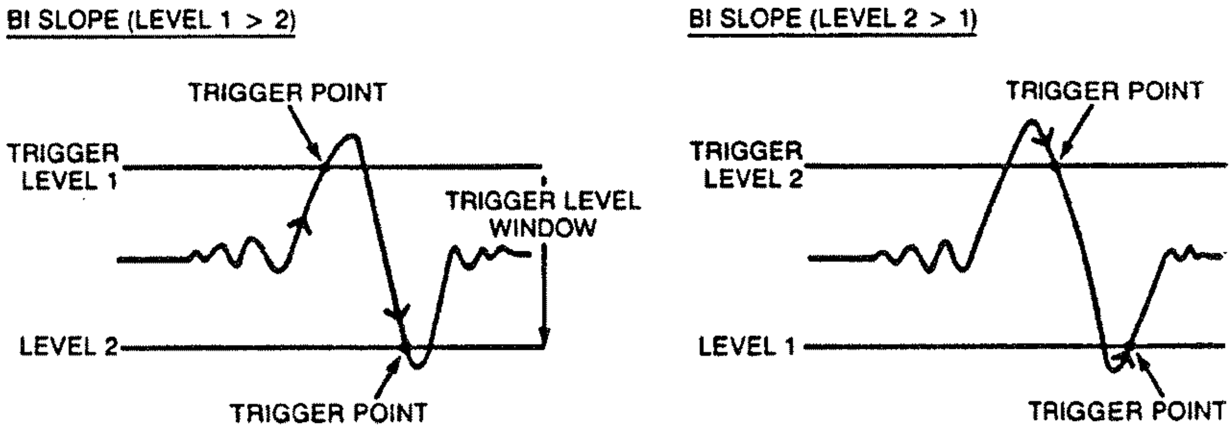
In **Compare mode** trigger, the RTD 710A continuously records in the Normal trigger mode until an acquired signal satisfies the compare condition, either compare-in or compare-out. In compare mode, the last memory location automatically becomes the reference memory location. The reference waveform data should be enveloped data. If the operation complete (OPC ON/OFF, Table 3-12) SRQ capability is turned on, the RTD 710A will issue an SRQ (status=66, event=451) when compare mode has been satisfied.

Compare-in: All the acquired waveform data points must be in the range determined by the reference waveform data. If the acquired waveform is entirely inside the reference, the RTD 710A enters the HOLD state and an SRQ is issued (if OPC ON).

Compare-out: If any part of the acquired waveform falls outside of the reference, the RTD 710A enters the HOLD state and an SRQ is issued (if OPC ON).

Here are some important points regarding Compare-in and Compare-out modes:

- The record length must be less than 64K words for Dual Channel and less than 128K words for CH1 Only acquisition.
- The RTD 710A uses the last location as the reference memory. This location cannot be assigned as an acquisition location in the Compare modes. For example, if the record length is set to 8K words in dual channel mode, record location 1 is the comparison waveform and record location 64 is the reference.
- Both Compare modes digitize a waveform after a trigger event and make a comparison if the acquired waveform meets the requirements defined by the reference memory.
- If the waveform is acquired using the Envelope mode, the number of envelopes must be one.
- If the acquired waveform passes the comparison, the RTD 710A re-arms the timebase for another acquisition. If the acquired waveform fails the comparison, the RTD 710A enters the HOLD state and sends a service request (SRQ status 66, event code 451 "Hold has occurred in compare trigger mode"). The instrument controller can wait for this SRQ and event code to control a test program.


BI SLOPE (LEVEL 1 > 2)

BI SLOPE (LEVEL 2 > 1)

Fig. 1-5 Bi-slope trigger example.

Trigger Slope

The RTD 710A offers extensive triggering capabilities, including several trigger slope and level combinations. Not only does it have Positive (+) and Negative (-) slope triggering, it also has Bi-slope and Hysteresis trigger levels. The Bi-slope and Hysteresis trigger capabilities qualify the trigger event further than ever before. These capabilities let you reduce false triggers caused by unusual signal conditions. The following paragraphs explain the various trigger slope and level functions.

Positive slope (+): This function sets the RTD 710A to trigger on the waveform's positive slope. The trigger circuit generates a trigger pulse when a positive-going trigger signal crosses the selected trigger level determined by Trigger Level 1.

The range is set with two threshold levels, Level-positive (Trigger Level 1) and Level-negative (Trigger Level 2) using the LEVEL 1 and 2 keys and the PARAMETER ENTRY knob. Level 1 recognizes a positive-going trigger signal determined by the LEVEL 1 setting. Level 2 recognizes a negative-going trigger signal determined by the LEVEL 2 setting. Figure 1-5 illustrates how Bi-slope can trigger on either polarity of a trigger signal.

Hysteresis: The Hysteresis trigger lets you select a window where the trigger signal must first pass a selected level in the proper polarity, then pass a second level using an opposite polarity trigger signal. This capability is useful because it reduces the chances of false triggering on a noisy signal.

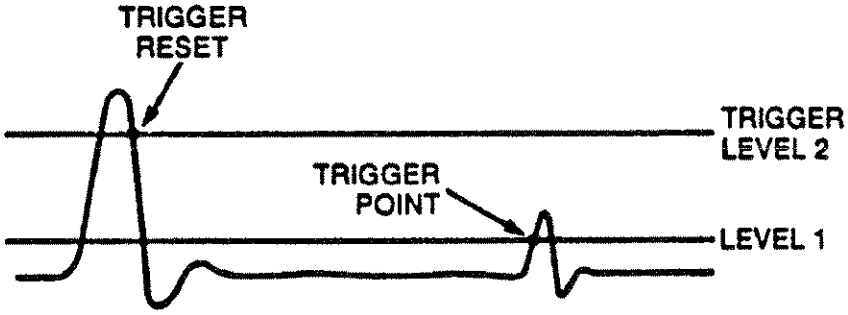


Fig. 1-6 Positive hysteresis trigger example.

Negative slope (-): This function sets the RTD 710A to trigger on the negative slope of the waveform. The trigger circuit generates a trigger pulse when a negative-going trigger signal crosses the selected trigger level determined by Trigger Level 1.

Bi-slope: This function triggers on the positive or negative slope, whichever occurs first; this is a powerful feature when the polarity of the input signal is unknown. The RTD 710A recognizes the trigger when the input signal goes out of a predetermined range.

+ Hysteresis: In this mode, the trigger circuit generates the trigger pulse when the positive-going trigger signal crosses the selected trigger level determined by Trigger Level 1. After the RTD 710A recognizes the first point determined by Level 1, a negative-going trigger signal must pass through the selected Hysteresis level determined by Trigger Level 2. Figure 1-6 shows a signal on which triggering is difficult using normal triggering methods but easy using the Hysteresis trigger.

- **Hysteresis:** In this mode, the trigger circuit generates the trigger pulse when the negative-going trigger signal crosses the selected trigger level determined by Trigger Level 2. After the RTD 710A recognizes the first point determined by Level 1, a positive-slope trigger signal must pass through the selected Hysteresis level determined by Trigger Level 1. Figure 1-7 shows two negative-hysteresis trigger examples, one with Level 1 greater than Level 2, and one with Level 2 greater than level 1.

porch) clamp circuitry to the channel 1 input amplifier.

The RTD 710A provides three TV Trigger coupling modes. These modes let you select either horizontal or vertical sync pulses to get 'horizontal line sync' or 'vertical sync pulse' triggering. This option permits triggering on a specific line number within a video field. This capability is fully controllable from the front panel and over the GPIB.

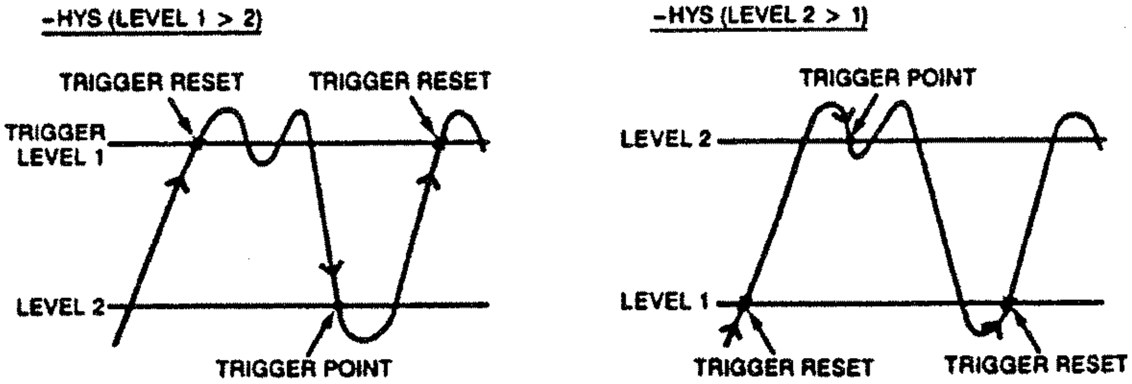


Fig. 1-7 Negative hysteresis examples.

Trigger Coupling

The Trigger Coupling switch lets you select the type of signal to be passed to the trigger circuits. The coupling can:

a) Pass all frequency components,

b) Reject either high or low frequency components, or

c) Trigger on a line, field 1, or field 2 using the TV trigger option.

DC: Couples all frequency components of the signal to the trigger circuit.

HF Reject: Attenuates triggering components above 50 KHz.

LF Reject: Capacitively couples signals, blocking dc components of the triggering signal. It attenuates the signal components below 50 KHz.

AC: capacitively couples signals, blocking dc components. It attenuates frequency components below 60 Hz.

The next three coupling choices are available only on the RTD 710A with the TV Trigger Option (Option 05). This option simplifies triggering and acquiring television signals. The option adds TV (back

Line: Encountering a specific video horizontal line sync pulse triggers the time base. The range of line numbers is 1 to 1280. The FIELD LINE # control selects a specific line number within a field.

FLD 1 (Field 1): In this mode, the time base triggers on the first field of the input TV signal. Use field 1 for video signals.

FLD 2 (Field 2): In this mode, the time base triggers on the second field of the input TV signal.

The Control Section

This section contains a variety of controls for display location selection, vertical and horizontal zooming, cursor measurements, plotter control, and more. Non-volatile settings storage can be used to easily recall a specific display location and zoom setups.

Non-Volatile Settings Storage

Up to 20 non-volatile instrument setups can be saved in the RTD 710A. This feature is very useful for storing commonly used settings. The user presses either the SAVE or RECALL button and uses the the PARAMETER ENTRY KNOB to select the setup number, then presses save or recall again to perform the task.

Display Locations

If the RTD 710A has an X-Y monitor attached (e.g., a Tektronix 620 monitor or a 2225 portable scope), channel 1 and channel 2 waveforms can be displayed, manipulated, and measured. Any of the possible 256 record locations can be displayed by selecting the display location. One location from each channel can be viewed. This is performed by pressing the DISPLAY LOCATION button and turning the PARAMETER ENTRY KNOB. Response time is quicker if the DOTS display mode is used instead of the LINES mode. The RTD 710A does not have to perform line interpolation before displaying the waveform in DOTS mode.

Vertical and Horizontal Zoom

Vertical and horizontal zoom expands or compresses the selected display channel's waveform to suit viewing needs, such as precision cursor measurements or viewing a 256K word waveform. The display always shows 2K words of data. If more than 2K words are to be displayed, the RTD 710A uses a technique to reduce the number of points without losing the major points of interest. The display will show the amount of reduction by putting a one (1) minus the reduction factor. For example, if the display shows; '1-4', it means dividing 8K words by 4 (yielding 2K display data).

The waveform can be vertically expanded or compressed using the following multiplication factors: 0.25 - 0.5 - 1 - 2 - 4 - 8 - 16 - 32 times. The LED display shows the zooming factor used.

You can also horizontally expand or compress the waveform. The zoom factor can vary from 1/32 to 16. The LED display again shows the zooming factor used.

Cursor Measurements

The RTD 710A lets you make manual measurements using cursors. You can measure volts, delta volts, time, delta time, 1/time, and 1/delta time. Table 1-1 summarizes the measurements available.

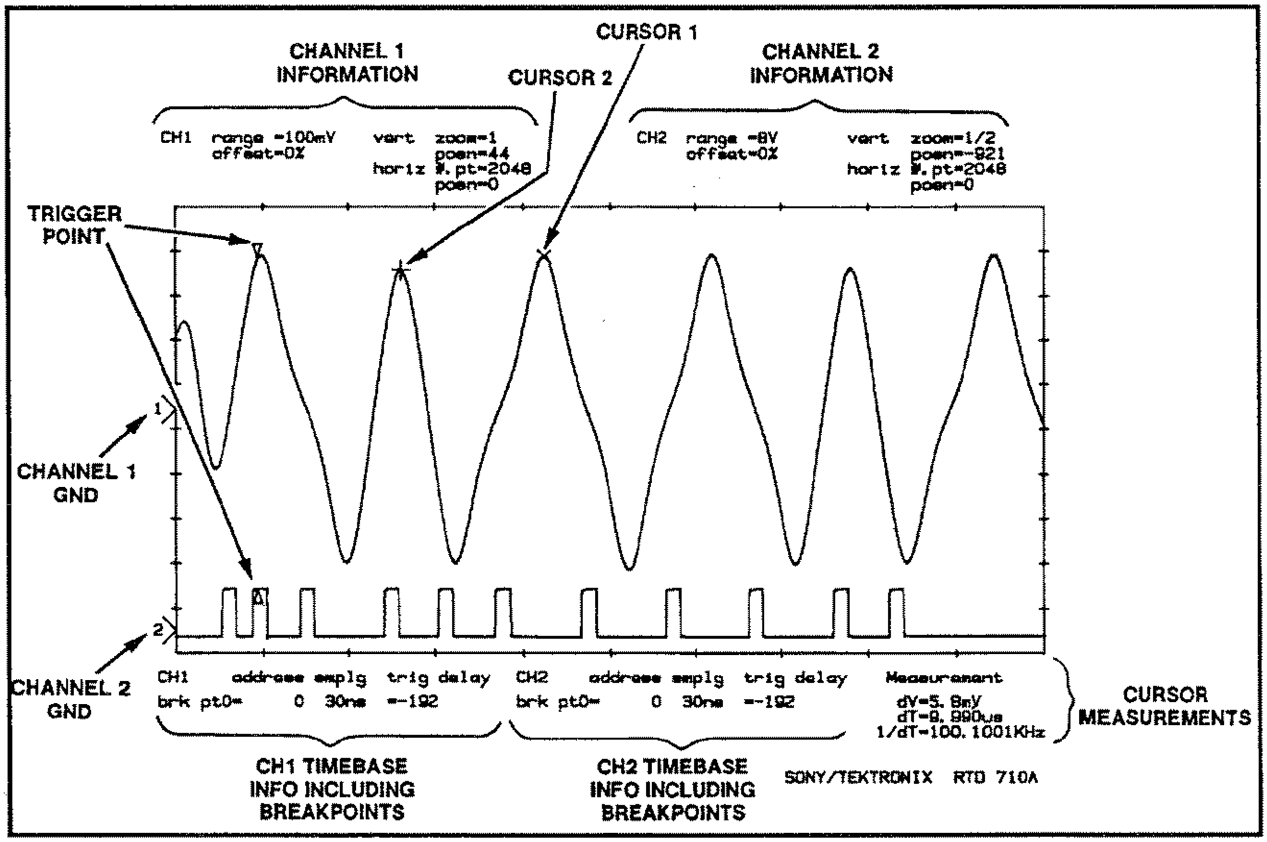
Table 1-1

RTD 710A CURSOR MEASUREMENTS

Measurement Cursor	Function	Calculation
1	V	Voltage difference from GND to Cursor 1.
	T	Time interval from trigger to Cursor 1.
	1/T	Reciprocal of time, yields frequency information.
1,2	Delta V	Cursor 1 - Cursor 2, voltage or vertical points units
	Delta T	Cursor 1 - Cursor 2, time units or horizontal points units.
	1/Delta T	Reciprocal of delta time, yields frequency information.

Plotter Control

An HPGL compatible plotter (e.g., Tektronix HC100) can be connected to the RTD 710A for display and measurement hardcopies. Whatever is displayed will be plotted. If cursor(s) are on, voltage, time, and frequency are listed. The plots are made in 3 colors for easy viewing. Refer to section 2, "GPIB Plotter Output" for information. An example plot is shown in Figure 1-8.



```
CH1  range =100mV     vert  zoom=1        CH2  range =8V        vert  zoom=1/2
     offset=0%              posn=44            offset=0%              posn=-921
                      horiz #.pt=2048                           horiz #.pt=2048
                            posn=0                                    posn=0

CH1      address smplg  trig delay   CH2      address smplg  trig delay    Measurement
brk pt0=       0  30ns  =-192        brk pt0=       0  30ns  =-192           dV=5.8mV
                                                                             dT=9.990us
                                                                        1/dT=100.1001KHz

                                                SONY/TEKTRONIX  RTD 710A
```

Fig.1-8 Plotter output example.

COMMUNICATING WITH THE RTD 710A

This section of the RTD 710A IIG gives an introduction to the GPIB, with descriptions of bus functions and command syntax.

Setting the RTD 710A for GPIB Operation

A group of binary switches on the back of the RTD 710A lets you configure the instrument for use on the GPIB. These switches offer three optional configuration settings:

- Talk-only (TON) mode
- Message terminator
- GPIB address

Talk Only (TON) Option

When the RTD 710A is used with an instrument controller, you must turn the TON option off (set it to logic 0), and thus make the RTD 710A a talk/listen instrument. The only time TON should be on (set to logic 1) is when you want to use an HPGL-compatible plotter connected to the GPIB without an instrument controller. Refer to the section titled "GPIB Plotter Output" later in this section for more details about the plotter output capability.

Message Terminator

You can set the message terminator for either EOI or EOI/LF. If your instrument controller supports EOI termination, set this option to logic 0. If the controller does not support "EOI Only" termination, select EOI/LF by setting the option to logic 1. The RTD 710A then sends and accepts commands using Line Feed (LF or EOI) as the terminator.

GPIB Address

You can set the RTD 710A's GPIB address at any number from 0 to 30. Five switches set the address, in binary. For example, to select address 17, set bit 16 (the 2^4 switch) and bit 1 (2^0) to logic 1, and all other address switches to logic 0. Note that only one instrument can reside at each address on the GPIB. Figure 2-1 shows a close-up of the switch and the locations of each settable option.

Determining the GPIB Address from the Front Panel

The address of the RTD 710A can be determined by pressing the RQS ID button on the front panel. The address is displayed for approximately one second on the Control (rightmost) LED display.

Introduction to the GPIB

GPIB controllers use high-level languages such as BASIC, C, and Pascal to define messages and transfer them to and from the digitizer. Statements in these languages usually contain three parts:

- The input/output keyword (such as PRINT or READ)
- The GPIB logical unit designator (such as '1' or 'DIG')
- A character string or string-variable designator (such as 'CH1 RANGE:2') that forms a specific instrument command or response

In this Guide, all program examples are in Tek 4041 BASIC unless otherwise noted. The RTD 710A is set to GPIB address 1.

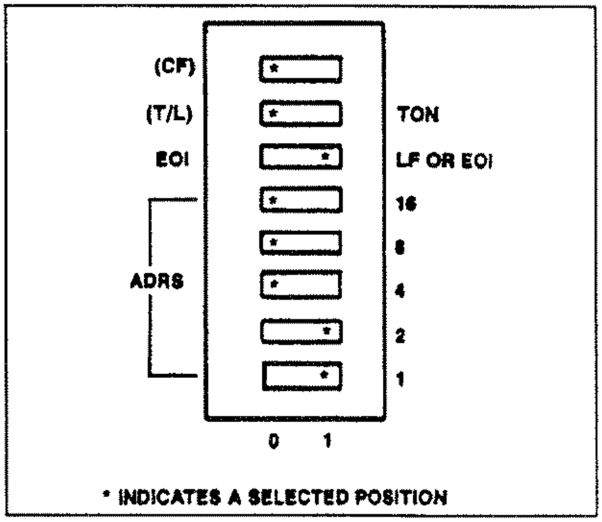


Fig. 2-1 Settable switch option locations.

Output Statements

The following examples show output statements in several controller languages. Any instrument command can take the place of "STRING" in the examples. Each statement assumes a prior configuration and declaration to the GPIB port and the device (the RTD 710A) on the bus.

```
Tek 4041 BASIC: PRINT #1:"STRING"
Tek SPS BASIC: PUT "STRING" INTO #1
[1]IBM  AT/XT/PC:WRT$="CH1 RANGE:20"
               CALL IBWRT
               (SCOPE%,WRT$)
```

Note[1] with Tek GURU II GPIB Interface and Microsoft BASICA; where WRT$ is an output string buffer.

```
[2]IBM AT/XT/PC:  IBWRT(scope,wrt,cnt);
```

Note[2] With National GPIB/Pascal Interface and Turbo Pascal; where wrt is an output string buffer and cnt is the number of bytes to be sent.

```
HP 9000 Series
200/300 BASIC: OUTPUT 701;"STRING"
Fluke 1720A
BASIC:         PRINT @1%:"STRING"
[3]NEC PC9801:    PRINT @SCOPE%;
               "STRING"@
```

Note[3] With NEC GPIB board and N88BASIC; where the @ character at the end of the line means "SET EOI WITH LAST BYTE."

Query Commands and Responses

The RTD 710A transmits measurement or status information after the controller sends a query command to the digitizer. Query commands are formed by immediately following a command header with a question mark. The query and input operations can be specified by separate statements. With some controllers, a prompting input statement can perform both functions. For example, the following Tek 4041 BASIC statement acquires the channel 1 settings. Set$ has been dimensioned as a 650-character string variable.

```
  160 Input #1 prompt "Ch1?":set$
```

The controller addresses the RTD 710A as a listener at primary address #1 and sends "Ch1?" over the bus. The controller then reassigns the instrument to be a talker and receives the characters into target variable set$. The set$ variable then contains the channel 1 information, for example:

```
  Set$="CH1 RANGE:2.0E+0,OFFSET:0"
```

If a query includes a command argument to specify a parameter, the digitizer shortens the response. For example, if the controller sends CH1? RANGE to the digitizer, the response could be:

```
  Set$="CH1 RANGE:2.0E+0"
```

The returned string set$ is in the proper format to be sent back to the RTD 710A. There are however cases where the settings string conflict with current instrument settings. An example is sending channel 2 information when in high speed mode. The RTD 710A will issue an execution error SRQ when settings conflict. There are a number of ways to handle the SRQ's and are explained in section 4, titled "Sending settings to the RTD 710A". The string can be saved to a file or retained in computer memory. The controller can then send the string to the RTD 710A at a later date, to set the conditions defined in the string.

Input Statements

The following examples show input statements in several controller languages. The instrument response occupies the SET$ or RD$ buffer in the examples. Each statement assumes a prior configuration and declaration of the GPIB port and the device (RTD 710A) on the bus. To identify the information desired, a query must precede every input:

```
Tek 4041 BASIC:INPUT #1:Set$
Tek SPS BASIC: GET RD$ FROM #1
[1]IBM AT/XT/PC:  IBRD(SCOPE%,RD$)
```

Note[1] With Tek GURU II GPIB Interface and Microsoft BASICA; where RD$ is an input string buffer.

```
[2]IBMAT/XT/PC:   IBRD(scope,rd,cnt);
```

Note[2] With National GPIB/Pascal Interface and Turbo Pascal; where rd is an input string buffer and cnt is the number of bytes to be read.

```
HP 9000 Series
200/300 BASIC: ENTER 701;Set$
Fluke 1720A
BASIC:         INPUT @1%:Set$
[3]NEC PC9801:    INPUT
               @SCOPE%;A,B,CLINE
               INPUT @SCOPE%;S$
```

Note[3] With NEC GPIB board and N88BASIC. The first input example reads in numeric variables, and the second example reads in a string variable.

Instrument Settings Transfer

You should copy initial setups from a manually operated instrument to the controller using the SET? query. The RTD 710A settings can be modified at a later time with a few commands.

The RTD 710A can transfer the current setup to a controller in response to a SET? query. The entire setup is described in ASCII characters, as defined in the command tables. The next section includes an example 4041 program that queries and reads settings from the RTD 710A.

Sending and Receiving ASCII Settings

To save an ASCII setup for future use, request a SET? string from the digitizer and store it in a string variable. To set the instrument up with the parameters, simply send the string back to the instrument. The stored parameters can be stored in a file and read at a later time to set the digitizer to a desired state. The following example listing illustrates how to read the settings in ASCII into a string variable SET$. There are instances where instrument settings can conflict. In these cases, the RTD 710A issues an execution error SRQ. There are a number of ways to handle this condition. Refer to section 4, titled "Sending settings to the RTD 710A" for more information.

```
390    ! Dimension string variable
400    Dim set$ to 650
405    ! Read front panel setup
410    Input #1 prompt "set?":set$
. .
. .
. .
740     ! Reset settings defined by SET$
750    Print #1: set$
. .
. .
```

Handling Service Request (SRQ) and Event Codes

The most recent RQS ON or RQS OFF command determines whether the digitizer generates the Service Request (SRQ) message on the GPIB when either an error or a change in status occurs. The disabling effect of the RQS command is reset to RQS ON at power-on. The SRQ indicator located in the lower right portion of the front panel lights when the digitizer asserts SRQ on the bus.

If the controller is configured and programmed appropriately, the SRQ interrupts the normal program flow. To service an interrupt, the controller performs a Serial Poll. In response, the digitizer returns a Status Byte (STB), which reveals the type of event that occurred. The instrument whose SRQ is handled stops asserting the SRQ line.

If another SRQ is pending, either from the same or another instrument on the GPIB, the SRQ line will continue to be asserted until all SRQ's are handled. If the controller does not respond to the SRQ message, the instrument continues to operate and communicate normally, even though the condition that caused the SRQ may invalidate an operation such as a measurement, setting, or acquisition.

After reading a status byte, the controller can seek more information about the event by sending the EVENT? query. The instrument returns a number that identifies the specific event. Section 5 defines the various status bytes, event codes, and errors. The example program location shown below shows how to:

- Handle an SRQ
- Determine the instrument status
- Display the status byte and associated event code on the controller screen

```
. .
. .
800     ! Define variables as integers
810     Integer stb,dev,event
820     ! Perform auto serial poll
830     Poll stb,dev
840     !Status=stb, Device address= dev.
850
860     !Get event code from dev
870     Input #dev prompt "event?": event
        ! Get event code from dev
880 !
890     print "SRQ from ";dev;", status=
        ";stb;", event= ";event
. .
. .
```

Sending the USER ON command to the RTD 710A enables the user SRQ interrupt; you can generate an SRQ from the front panel by pressing the RQS/ID button. The RTD 710A sends a status byte of 67 or 83 and an event code of 403 signifying a operator request. These numbers can be used to control program flow by waiting for the operator to press RQS/ID before taking such actions as acquiring a waveform or making a cursor measurement.

GPIB Plotter Output

Whatever is displayed on the RTD 710A monitor will be can be plotted to an HPGL-compatible plotter. Plotting can be done stand-alone in Talk-Only mode or initiated by a controller on the GPIB.

Plots without controller: The plotter should be set to "listen only" and the RTD 710A to "talk only." With the plotter and RTD 710A set correctly, the digitizer sends the proper commands to the HPGL-compatible plotter to make a hardcopy when you press the front panel PLOT button. The RTD 710A is made a "talk only" by setting the Talk ONly (TON) dip switch to logical one (1). Making the plotter a listener only requires looking the plotter owners manual. In many cases, setting all plotter address dipswitches to one (1) makes it a "listener only"

Plots with controller: The instrument controller can initiate a plot over the GPIB by setting up the RTD 710A & plotter and sending the proper commands. The example program below illustrates how to make a plot:

Initiating a HP-GL Plot from the Controller

The following is a sample program for making a copy using an HP-GL plotter.

```
100 !
110 !  A copy from RTD 710A to a HP-GL plotter
120 !  that's connected the GPIB
130 !  RTD 710A address 1, Plotter address 2
140 !
150    Init all
160    Open #1: "gpib(pri=1, eom=<0>):"
170    Print #1: "DEVICE SET-
       TINGS;ON,WAVFRM:ON, GRAT:ON"
180    Print #1:"PLOT"
190    Wbyte atn(unt,unl,34,65):Make plotter
       listener
200    On eoi then call alldone
210    Enable eoi
220    Wait
230    End
1000   Sub alldone
1010   Wbyte atn (unt,unl)
1020   Print "Plot is done."
1030   Resume
1040   End
```

Interface Messages

This section describes the effects of GPIB interface messages received by the digitizer from a controller. See ANSI/IEEE Standard 488-1978 for detailed descriptions of interface messages and resultant interface states. These interface messages may be explicitly generated by the GPIB interface software in the controller or they can be composed in hexadecimal format according to the IEEE Standard. Unlike instrument commands, they can not be sent as character strings.

My Listen Address (MLA) and My Talk Address (MTA)

MLA messages condition the instrument to receive commands. MTA messages respond to queries and serial polls. MLA is received when the Attention (ATN) line is true and the data on the GPIB is decimal 32 plus the address set on the dip switches on the back of the RTD 710A. MTA is received when Attention is true and the data on the GPIB is decimal 64 plus the RTD's address. For example, if the RTD 710A is set to address 1 on the dip switch, then the listen address is 32+1=33 and the talk address is 64+1=65.

Local Lockout (LLO)

This command locks out the front panel so the operator cannot change the front panel settings while under program control.

Whether it is addressed or not, the instrument may receive the Local Lockout message (LLO) at any time. After the instrument receives the LLO message, the My Listen Address (MLA) message locks out the front panel controls and the Go To Local (GTL) message enables the front panel to local control. The LLO message disables the front panel controls immediately if the instrument is addressed as a listener when the LLO is received.

If a program sends the LLO message, it should also send the GTL message to a listener-addressed device when the front panel is active. The LOCK indicator lights when front panel operation is suspended.

If the controller sets the Remote Enable (REN) line false or if power cycles off and on, the effect of the LLO message is cancelled and the instrument controls operate normally.

Remote Enable (REN)

When the Remote Enable (REN) line is true and the instrument receives its listen address (MLA), the digitizer can receive data from the bus, and the digitizer's Remote (REM) indicator will light.

If the REN line goes false, the instrument must receive MLA again before it can receive commands. If a command is in process when REN goes false, the command continues to execute.

Messages sent to the RTD 710A when REN is false generate an SRQ (status=97, event=201) if SRQ is not disabled, indicating "Command not executable in local mode."

Go to Local (GTL)

If GTL is received, the instrument must receive MLA again before it can receive commands. If a command is in process when GTL is received, it continues to execute.

GTL and a listen address (MLA) are necessary before the front panel is operable.

Unlisten (UNL) and Untalk (UNT)

The UNL message is equivalent to talk address decimal 31, so the address sent is 31+32=63. The UNT message is equivalent to listen address decimal 31, so the address sent is 31+64=95. The UNL message cancels the MLA message. The UNT message cancels the MTA message. The Untalk and Unlisten commands are called "universal" commands. All GPIB instruments stop talking and listening when the controller sends UNT and UNL messages.

Interface Clear (IFC)

IFC is a hardware line in the GPIB interface. When this line is true, the RTD 710A responds by Untalking (UNT) and Unlistening (UNL).

Device Clear (DCL)

The DCL message initializes communication between the instrument and the controller. In response to DCL, the instrument clears input and output messages as well as unexecuted control settings. Errors and events waiting to be reported are cleared, except the power-on event. The SRQ message is cleared unless SRQ is true from a power-on condition.

Selected Device Clear (SDC)

SDC performs the same function as DCL, but requires the instrument to have been listen-addressed. (More than one instrument may have been addressed.) This functions allows the controller to perform a device-clear on selected instruments.

Serial Poll Enable (SPE) and Serial Poll Disable (SPD)

The Serial Poll Enable (SPE) message causes the instrument to transmit its serial poll status byte when it is talk-addressed. The Serial Poll Disable (SPD) message returns the instrument to normal operation.

Parallel Poll

The RTD 710A does not support parallel polling commands.

Group Execute Trigger (GET)

The RTD 710A supports the Group Execute Trigger (GET) function. In the RTD 710A, the DT ON command, listed in Table 3-10, enables the RTD 710A to recognize the GET command. When the GET function is enabled, the instrument reads commands but doesn't execute them until the GET command is received. This capability allows many instruments to be synchronized by having them wait for the GET command to execute. A multiple digitizer system can use the GET command to acquire many channels at the same time.

Command Handler

A command handler performs these functions:

- Establishes communication between the controller and the digitizer
- Sends commands and queries to the RTD 710A
- Receives responses from the digitizer
- Processes the responses as required

The following outline indicates a general sequence of command handling functions. The commands used by a particular controller may vary.

1. Initialize the controller.
2. Disable the service-request handler until the program is ready to handle them.
3. Get the GPIB address of the RTD 710A.
4. Enable the service-request handler.
5. Send a command to the digitizer.
6. Check for a response from the digitizer, either an SRQ or a response to a query.
7. Process any response as desired.
8. Repeat steps 5 through 7 as desired.

Service Request Handler

A service request handler processes the interrupts generated by the SRQ message on the GPIB. For example, after placing cursors on points of interest on a waveform, you can press the RQS/ID button to assert the SRQ line and interrupt the controller. The controller can verify that the RQS/ID button was pressed (by checking to see if the status=67 or status=83 and the event code=403). After verifying the button status, the controller can read the cursor measurement from the RTD 710A.

Section 5 identifies all the SRQ and Event codes of the RTD 710A.

Some controllers can ignore service requests. Others require a programmed response to SRQ. Most controllers ignore service requests until SRQ interrupts are explicitly enabled.

An SRQ handler needs an interrupt enabling statement (for example, the ON SRQ statement) near the beginning of the program and a serial poll subroutine with that label. The ON SRQ statement directs program control to the serial poll subroutine whenever an SRQ interrupt occurs. The instrument maintains the identity of an event that generates an SRQ until the interrupt service subroutine executes a serial poll.

The following general steps handle service requests from the digitizer:

1. Perform a serial poll to determine which device on the bus is requesting service. The serial poll clears an SRQ generated by the digitizer, unless more than one event has been identified.
2. Send an EVENT? query to the digitizer requesting service.
3. If the EVENT? query response is not zero, perform the appropriate response to the event.
4. Return to the main program.

GPIB Commands

GPIB commands set instrument operating states, query the operating states, and query the results of measurements. Users specify these commands in mnemonics that are related to function names or front-panel control names. Commands follow the conventions established in the Tektronix Standard Codes and Formats guidelines.

All command messages consist of headers, arguments, separators, and messages terminators.

Headers

A command must have a header. A few commands are fully specified by a header alone. For example:

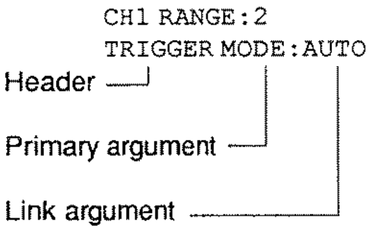
```
INIT
```

Arguments

Most commands require arguments after the headers. An argument must be separated from its header by a space. For example:

```
VMODE CH1
BWLIM ON
```

Some commands also have link arguments. The primary and link arguments must be separated by a colon:



```
CH1 RANGE:2
TRIGGER MODE:AUTO
```

Some headers allow multiple arguments, which must be separated by commas. A colon still separates primary and link arguments.

```
CH1 RANGE:20,OFFSET:20
SAMPLE MODE:NORM,CLOCK:INT,
INTERVAL: 20E-9
```

Command Separator

You can combine multiple commands in one message by separating the individual commands with semicolons:

```
CH1 RANGE:2;SAMPLE MODE:NORM
```

Queries

In a query, the question mark must immediately follow the header, with no space between them:

```
CH1? RANGE
SAMPLE? INTERVAL
```

Message Terminator

Messages can be terminated with either EOI or EOI/LF, depending on the system controller. You can set the RTD 710A's GPIB interface to accept either terminator, by using the dip switch on the rear panel.

With EOI selected, a data byte received with EOI asserted is recognized as the end of an input message. The instrument also asserts EOI concurrently with the last byte of an output message.

With the LF setting, either an LF character or any data byte received with EOI asserted is recognized as the end of an input message. With the LF selection, the instrument transmits a Carriage Return character followed by Line Feed (LF) with EOI asserted to terminate an output message.

If the controller supports EOI, use EOI only as a terminator. This approach eliminates any unwanted terminations if the binary waveform data contains the Carriage Return and Line Feed characters.

Abbreviations

Defined words in headers and arguments can be entered full length, or can be shortened to reduce typing and bus traffic. The command tables show the essential characters of headers and arguments in upper-case, and non-essential characters in lower-case. The instrument accepts either upper-case or lower-case characters. For example, the following commands are acceptable:

```
VMODE?
vmo?

vmode dual
VMO DUA
```

Numeric Arguments

Table 2-1 depicts the formats for numeric arguments in the GPIB command set. Both signed and unsigned numbers are accepted but unsigned numbers are interpreted as positive.

The symbol <NRx> indicates that any of the three formats is allowed. When only one format is permitted, it is represented by <NR1>, <NR2>, or <NR3>.

Table 2-1
NUMERIC ARGUMENT FORMAT OF GPIB COMMANDS

Numeric Argument Symbol		Number Format	Examples
<NRx>	<NR1>	Integers	+1, 2, -1, -10
	<NR2>	Explicit decimal points	-3.2, +5.1, 1.2
	<NR3>	Floating point in scientific notation	-1.0 E-2, +10.3E9

GPIB COMMAND REFERENCE

This section of the RTD 710A Instrument Interfacing Guide lists the commands used by the RTD 710A. It does not cover possible errors caused by arguments being out of range or other causes.

Some special conventions are observed in this section:

- <NRx>: Denotes that either an integer number (e.g., 1, 345), a short floating point number (e.g., 2.4, 456.7), or a number expressed in scientific notation (e.g., 2E-9) can be sent to the RTD 710A.

- The vertical bar symbol (|): Separates the exact choices available for a 'multiple-choice" argument. For example, when entering the record length (in the Timebase table), many choices are available, denoted as:

  1024|2048|4096|8192|16384|32768|65536|etc

- Double equal signs (==): Mean that the choices available must be exactly as shown in the explanation.

- Brackets ([]): Enclose an optional argument or arguments.

- 'Query only" in the function column of a table means that no corresponding settings command exists. 'Query only" commands can be sent to the RTD 710A, but no setting action will occur.

The command reference tables are broken into functional groups. Here is a summary of the tables contained in Section 3:

Table 3-1
VERTICAL SYSTEM CONTROL GROUP

Header	Argument	Link Argument	Description
VMOde	CH1 DUAl		Selects the vertical input mode. CH1 selects CH1 only. DUAl selects both CH1 and CH2. A record Length of 256 K is incompatible with DUAl mode. Selecting DUAl mode limits the record length to 128 K, and a warning SRQ is issued if EXW is ON and 256 K is attempted. DUAl mode and High-speed sample mode are incompatible. Selecting DUAl forces sample mode to NORmal, and the sample interval is set to 10 ns if 5 ns is attempted. Also, a warning SRQ is issued if EXW is on. Record location number or auto advance location is limited to 128 K if vertical mode is set to DUAl, and a warning SRQ is issued if EXW is on. CH1 only mode and XY display mode are mutually exclusive. Selecting CH1 only mode forces display mode to YT.
VMOde?			Queries the vertical mode selection. Example response: VMODE DUAL
BWLim	ON OFF		Enables the 20 MHz bandwidth limiter.
BWLim?			Queries the status of the 20 MHz bandwidth limiter. Example response: BWLIM ON

More Vertical System Control Commands on next page

Table 3-1 (Cont)
VERTICAL SYSTEM CONTROL GROUP

Header	Argument	Link Argument	Description
CH1 (or CH2)	RANge:	<NRx>	Sets the CH1 (or CH2) full-scale range to <NRx> volts. <NRx>::=0.1–500 V in a 28-step sequence of 1-1.25-1.6-2-2.5-3.2-4-5-6.2-8. Range is automatically adjusted to a probe attenuation change. Unit is volts. <NRx> is truncated and limited to the nearest legal setting, and warning SRQ is issued if EXW is on. When queried, the value is <NR3>.
	UNit:	PERcent VOLts	Selects the units of measure for the CH1 (or CH2) input offset.
	OFFset:	<NRx>	Sets the CH1 (or CH2) input offset value in % of full scale or <volts>. <NRx>::=–199% to +199% in 1% steps. <volts>::=(% value) * (full scale range). The offset automatically adjusts to probe attenuation changes. <NRx> is truncated and limited to the nearest legal setting, and a warning SRQ is issued if EXW is on. When queried, the value is <NR1> for PERcent or <NR3> for VOLts.
	COUpling:	AC GND DC TVClamp	Selects CH1 (or CH2) coupling. TVClamp cannot be selected for CH2. If TVClamp is selected and the TV Trigger option is not installed, an SRQ is issued (if EXR is on) and the command is ignored.
CH1? (or CH2)?	RANge UNit OFFset COUpling PRObe		Queries the selected CH1 (or CH2) vertical settings. RANge value is in the <NR3> format. OFFset value for % of full scale is in <NR1> format; for <volts> it is in <NR3> format. PRObe reports probe attenuation factor (X1 or X10).
CH1? (or CH2?)			Queries the CH1 (or CH2) vertical settings, except probe attenuation. Example response: CH1 RANGE:2.5E+0,UNIT:PERCENT, OFFSET:0,COUPLING: AC

Table 3-2
TIME BASE AND RECORDING GROUP

Header	Argument	Link Argument	Description
SAMple	MODe:	NORm HISpd	Selects the sampling mode. HISpd forces VMOde:CH1, TRIgger SOUrce:CH1, DISplay MODe:YT. If RECOrd MODe:ENV, HISpd forces it to RECOrd MODe:NORm. Trigger delay value and Break Point address may be truncated; if setting or value change occurs, an SRQ is issued if EXW is on. When Normal mode is set, sample interval is limited to 10 ns or more.
	CLOck:	INT EXT	Selects the sampling clock source. EXT forces TRIgger DUNit:POInt, BREakpoint UNit:POInt, MEAsure FREquency:OFF, TIMe:OFF.
	INTerval:	<NRx>	Sets the sampling rate. <NRx>::=5E–9 to 2E–1 seconds in steps of one for internal clock source. <NRx>::=1 to 4E+7 points in steps of one for external clock source in normal sample mode. <NRx>::=1 to 4E+7 points in a 1-2-4-6-8 sequence for external clock source in high-speed sample mode. <NRx> is truncated or limited to nearest legal setting, and a warning SRQ is issued if EXW is on. When queried, the value is <NR3>.
SAMple?	MODe CLOck INTerval		Queries the selected SAMple settings. INTerval value is in <NR3> format.
SAMple?			Queries for all SAMple settings. Example response: SAMPLE MODE:NORM,CLOCK:INT, INTERVAL:10.0E–9

More Time Base and Recording Group Commands on next page

Table 3-2 (Cont)
TIME BASE AND RECORDING GROUP

Header	Argument	Link Argument	Description
RECOrd	MODe:	NORm	Selects the recording mode.
		AVE	AVErage forces a record LENgth of 131072.
		ADV	ADV selects the auto-advance mode, and forces TRIgger MODe:NORm if it was INComp or OUTcomp.
		ENV	ENV selects the enveloping mode, which forces SAMple MODe:NORm, and TRIgger MODe: NORm if it was INComp or OUTcomp. In ENVelope mode, Record Length value is limited to 128 K. If 256 K is selected in this mode, a warning SRQ is issued if EXW is on.
	ENVelope:	<NRx>	Sets number of times to envelope. <NRx>::=1 to 16348 in 2^n steps. An <NRx> of 99999 causes continuous enveloping until HOLd ON is received. When <NRx> is 2 or more, the last waveform memory segment is used for enveloped waveform accumulation. In this case, record location is changed to the last location minus one, and a warning SRQ is issued if EXW is on. <NRx> is truncated and limited to a legal number, and a warning SRQ is issued if EXW is on. When queried, the value is <NR1>.
	AVErage:	<NRx>	Sets the number of times to average. <NRx>::=2 to 16348 in 2^n steps. If <NRx> exceeds the range, it is truncated and limited to a legal number, and a warning SRQ is issued if EXW is on. When queried, the value is <NR1>.
	LOCation:	<NRx>	Sets the waveform memory location number as the acquisition segment. <NRx>::=1 to 256 (VMOde CH1) <NRx>::=1 to 128 (VMOde DUAl) If RECOrd MODe:ENV and ENVelope is equal to or more than 2: <NRx>::=1 to 255 (VMOde CH1) <NRx>::=1 to 127 (VMOde DUAl) If <NRx> exceeds the range, it is truncated or limited to match a legal value, and a warning SRQ is issued if EXW is on. When queried, the value is <NR1>.

More Time Base and Recording Group Commands on next page

Table 3-2 (Cont)
TIME BASE AND RECORDING GROUP

Header	Argument	Link Argument	Description
RECOrd?	MODe ENVelope AVErage LOCation		Queries the selected RECOrd settings. ENVelope value is in <NR1> format. AVErage value is in <NR1> format. LOCation value is in <NR1> format.
RECOrd?			Queries for all RECOrd settings. Example response:RECORD MODE:NORM, AVERAGE:2,ENVELOPE:1,LOCATION:1
LENgth	NRx		Sets the acquisition record length. <NRx>::=1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072, or 262144. 262144 is for VMOde CH1 and SAMple MODe HISpd only. <NRx> is truncated or limited to the nearest legal setting, and a warning SRQ is issued if EXW is on. Record Length value affects the trigger delay value and record location number. If limitation occurs because of a record length value change, the values are limited to the nearest legal value, and a warning SRQ is issued if EXW is on. Record Length value also affects the Cursor position value, Horiz Zoom value, and Display Location number. These values are also limited to the nearest legal value or number. When queried, the value is <NR1>.
LENgth?			Queries for the record acquisition length. Response is in <NR1> format. Example response: LENGTH 2048
HOLd	ON NEXt RESet		Controls the acquisition (recording) start and stop for time base. ON immediately stops an acquisition operation. Some data may be invalid. NEXt allows the current acquisition operation to continue through its specified record length, then enter the hold state. RESet clears the hold state and initiates the start of a new acquisition.
HOLd?			Reports the acquisition status for both time base channels. Example response: HOLD RESET

More Time Base and Recording Group Commands on next page

Table 3-2 (Cont)
TIME BASE AND RECORDING GROUP

Header	Argument	Link Argument	Description
BREakpoint	CLEar:	<NRx>	Clears the specified breakpoint. <NRx>::=1-5. If requested number is illegal and EXR is on, an error SRQ is issued.
	UNIt:	POInt TIMe	Selects the units of measure for the breakpoint settings. If SAMple CLOck:EXT is selected, only POInt is available.
	SET:	<NRx>:<NRx>	Sets up to 5 breakpoints. First <NRx> in the argument assigns the break point location. Break point location limit is 10 to 524272 for Normal sampling mode, 10 to 524256 for High-Speed sampling mode. If limitation occurs, a warning SRQ is issued if EXW is on. Actual effective location is limited to the value shown below. First <NRx>::=<address>|<time> <address>::=+16 to (record length+trigger delay –16) in increments of 8 in NORmal SAMple MODe. <address>::=+32 to (record length+trigger delay –32) in increments of 16 in HISpd SAMple MODe. <time> is calculated by <address> * (sample interval in seconds). If the external clock source is selected, only <address> can be set. Second <NRx> sets the sampling interval for the breakpoint defined by the first <NRx>. See SAMple INTerval for allowable <NRx> values.
BREakpoint?	UNIt SET		Reports selected BREakpoint settings. For SET, the first <NRx> value is in <NR1> format for <points> and <NR3> format for <time>. The second <NRx> value is in <NR3> format.
BREakpoint?			Reports all BREakpoint settings. Example response: BREAKPOINT UNIT:POINT, SET:0:10.0E–9,SET:520:100.0E–9

More Time Base and Recording Group Commands on next page

Table 3-2 (Cont)
TIME BASE AND RECORDING GROUP

Header	Argument	Link Argument	Description
REPeat		<NRx>	Sets the number of acquisition cycles for the time base (not available via front panel). This command is the equivalent of having the instrument acquire <NRx> times and receiving the CURve? query. <NRx>::=1 to 65536 in increments of 1. Forces TRIgger MODe:SGL and RECOrd MODe:NORm.
NUMacq?			Reports the number of acquisitions. Response is in <NR1> format. For TRIgger MODe:INComp or :OUTcomp, the returned value indicates the number of compare operations that occurred before entering the HOLd state. For TRIgger MODe: AUTo, NORm, or SGL, the returned value indicates the number of acquisitions that occurred before entering the HOLd state. For RECOrd MODe:ADV, the response indicates the last record location that completed acquisition before entering the HOLd state. An intermediate number is returned if the HOLd state has not been entered. The controller should monitor the OPC SRQ to determine when the HOLd state is entered.

Table 3-3
TRIGGERING GROUP

Header	Argument	Link Argument	Description
TRIgger	MODe:	AUTo NORm SGL INComp OUTcomp	Selects the trigger operating mode. In compare mode (IN or OUT), 256 K record length is not permittted. If record length is 256 K and EXW is on, a warning SRQ is issued. More than two envelope times and compare mode are incompatible. If compare mode is set, envelope # times is set to 1.
	DUNit:	POInt TIMe	Selects the units of measure for the trigger delay value.
	DELay:	<NRx>	Sets trigger delay value. A negative number indicates pretrigger. <NRx>::=<points> or <time>. <NRx>::= –(LENgth:<NRx>–8) to 262136 in increments of 8 for SAMple MODe:NORm. <NRx>::= –(LENgth:<NRx>–16) to 262128 in increments of 16 for SAMple MODe:HISpd. <time>::=(points) * (sample interval). If <NRx> is not legal, it is truncated and limited to a legal hardware setting, and a warning SRQ is issued if EXW is on. When queried, value is <NR1> for Point unit or <NR3> for Time unit. Trigger delay value determines the trigger position where a negative number sets pretrigger and a positive number sets posttrigger. Trigger position is used as a time origin for the break point location, cursor position, PT. Off of waveform preamble, STArt of DATa command, and STArt of WINdow command. Change of trigger delay value may affect the above values, and limitation may occur for those values.

More Triggering Group Comands on next page

Table 3-3 (Cont)
TRIGGERING GROUP

Header	Argument	Link Argument	Description
	COUpling:	AC HFReg LFReg DC LINes FLD1 FLD2	Selects trigger coupling. HFReg attenuates signals above 50 kHz. LFReg attenuates signals below 50 kHz. AC blocks dc components and attenuates signals below 60 Hz. LINes, FLD1, and FLD2 are for TV Option only. For FLD1 and FLD2, TRIgger SOUrce:CH1 is forced and if TRIgger SLOpe is set to BI, PHYs, or NHYs, it is forced to NEGative. If Trig Source is CH2 and TV trigger coupling is requested, the Trig Source is changed to CH1, and SRQ is issued, if EXW is on. If LINE or FLD1 or FLD2 is requested and TV Trigger option is not installed, an SRQ is issued, if EXR is on, and the command ignored.
	SOUrce:	CH1 CH2 EXT	Selects the trigger source. Selection defaults to CH1 for TV coupling selections and for SAMple MODe:HISpd.
TRIgger	SLOpe	POSitive NEGative BISlope PHYs NHYs	Selects trigger slope. Selection defaults to NEGative if BISlope, PHYs, or NHYs is selected with TV coupling selections, and an SRQ is issued if EXR is on.
	LUNit:	PERcent VOLts	Selects the units of measure for trigger level 1 and level 2.
	LEV1:	<NRx>	Sets trigger level 1. <NRx>::= –99% to +99% in 1% steps. <NRx> in Volts::=(% value) * RANge. When external trigger source is selected, full scale is ±5v. If <NRx> is illegal, it is truncated or limited to a legal hardware setting, and a warning SRQ is issued if EXW is on. <NRx> in volt value takes Range value into account. If range value is changed or probe attenuation is changed and level unit is VOLts, the level value is automatically changed (see above formula). When queried, the value is <NR1> for Percent unit or <NR3> for Volts unit.

More Triggering Group Comands on next page

Table 3-3 (Cont)
TRIGGERING GROUP

Header	Argument	Link Argument	Description
	LEV2:	<NRx>	Sets trigger level 2. <NRx>::= −99% to +99% in 1% steps. <NRx> in Volts::=(% value) * RANge. If <NRx> is illegal, it is truncated or limited to a legal hardware setting, and a warning SRQ is issued if EXW is on. <NRx> in volt value takes Range value into account. If range value or probe attenuation factor is changed and level unit is VOLts, level value is automatically changed by formula above. When queried, the value is <NR1> for Percent unit or <NR3> for Volts unit.
	LINe:	<NRx>	TV Option. Selects which TV line number from the selected TRIgger COUpling field causes the trigger. <NRx>::=1 to 1280 If outside this range, a warning SRQ is issued if EXW is on.
	LCNStart:	PREfld ATFld	TV Option. Defines where the TV line count starts. PREfld line count starts 3 lines before the field-sync pulse (System-M). ATFld line count starts at the field-sync pulse (Nonsystem-M). This setting forces trigger coupling to FLD1. When queried, the value is <NR1>.

More Triggering Group Comands on next page

Table 3-3 (Cont)
TRIGGERING GROUP

Header	Argument	Link Argument	Description
TRIgger?	MODe DUNit DELayCOUpling SOUrce SLOpe LUNit LEV1 LEV2 LINe LCNStart		Reports the selected trigger settings. LEV1 (or LEV2) % value is in <NR1> format; voltage value is in <NR3> format. DELay <points> value is in <NR1> format; <time> value is in <NR3> format. LINe value is in <NR1> format.
TRIgger?			Reports all trigger settings. Example response: TRIGGER MODE:AUTO,DUNIT:POINT, DELAY:–400,COUPLING:DC,SOURCE:CH1, SLOPE:POSITIVE,LUNIT:PERCENT,LEV1:0, LEV2:0,LCN START:PREFLD
ARM	MODe:	INT EXT	Selects the arming signal source.
	DELay:	<NRx>	Sets the trigger arming delay value. <NRx>::=0, or 10ms to 10s in a 1-2-5 sequence. If <NRx> is illegal, it is truncated and limited to a legal hardware setting, and a warning SRQ is issued if EXW is on. When queried, the value is <NR3>.
ARM?	MODe DELay		Reports the selected trigger ARM settings. DELay value is in <NR3> format.
ARM?			Reports all trigger ARM settings. Example response: ARM MODE:INT,DELAY:0.0E+0

Table 3-4
CURSOR AND DISPLAY CONTROL GROUP

Header	Argument	Link Argument	Description
CURSor	ONE:	DISP1 DISP2 OFF	Enables cursor 1 on CH1 display. Enables cursor 1 on CH2 display. Disables cursor 1.
	TWO:	DISP1 DISP2 OFF	Enables cursor 2 on CH1 display. Enables cursor 2 on CH2 display. Disables cursor 2.
	SCRoll:	ALIgn INDep	Selects cursor scroll mode. ALIgn locks CH1 and CH2 displays together during scrolling using cursor 1. INDep allows scrolling on one display channel while the other remains locked.
	POS1:	<NRx>	Specifies the location of cursor 1 on the display. <NRx>::=(TRIgger DELay) to (end of memory segment). (end of memory segment)::= (TRIgger DELay)+(LENgth<NRx>). If <NRx> is illegal, the number is truncated or limited to nearest legal value, and a warning SRQ is issued if EXW is on. If the Trigger Delay value is changed, the Cursor Position value may be limited. When queried, the value is <NR1>.
	POS2:	<NRx>	Specifies the location of cursor 2 on the display. See POS1 above for <NRx> values and other limits.
CURSor?	ONE TWO SCRoll POS1 POS2		Reports the selected CURSor setting. POS1 and POS2 values are in <NR1> format.
CURSor?			Reports all CURSor settings. Example response: CURSOR ONE:OFF,TWO:OFF,SCROLL:ALIGN

More Cursor and Display Control Group Commands on next page

Table 3-4 (Cont)
CURSOR AND DISPLAY CONTROL GROUP

Header	Argument	Link Argument	Description
DISplay	CHAnnel:	CH1 CH2	Selects which display channel is programmed by DISplay LOCation, VZOom, and VPOsn.
	LOCation:	<NRx>	Selects the waveform memory record of the channel to be displayed. <NRx>::=1 to 256 (CH1 or HISpd mode) <NRx>::=1 to 128 (DUAl mode) If <NRx> does not meet the other function settings or is outside range, it is truncated and limited to a legal value, and a warning SRQ is issued if EXW is on. When queried, the value is <NR1>.
	MODe:	YT XY	Selects the display mode. Both channels must be displayed for XY mode. If the Vertical Mode is set to CH1 Only, XY mode is not allowed. If XY is selected in CH1 Only or HISpd modes, an SRQ is issued if EXR is on, and the command ignored.
DISplay	INTerpol	DOT LINe	Selects either dot or vector (line) display mode for the CRT monitor output signal. There are two algorithms for data thin-out method for horizontally compressed display, ranging X1/2 to X1/128. One is Even-Spaced thin-out and the other one is Envelope. The thin-out algorithm is automatically determined by the INTerpol setting. Dot mode uses the Even-Spaced algorithm. Line mode uses the Envelope algorithm. Dot mode offers faster display update.
DISplay?	CHAnnel LOCation MODe INTerpol		Reports the selected DISplay setting. LOCation is in <NR1> format.
DISplay?			Reports all DISplay settings. Example response: DISPLAY CHANNEL:CH1,LOCATION:1, MODE:YT,INTERPOL:LINE

More Cursor and Display Control Group Commands on next page

Table 3-4 (Cont)
CURSOR AND DISPLAY CONTROL GROUP

Header	Argument	Link Argument	Description
HZOom	X1/128 X1/64 X1/32 X1/16 X1/8 X1/4 X1/2 X1 X2 X4 X8 X16		Selects the horizontal display scaling factor. The display memory is 2048 points. When a fraction is used in HZOom, e.g. 1/128, the data is thinned out to view a longer length waveform in 2K points. The numerator is the length of the data to be displayed. Using the example 1/128, this takes a 256 K waveform and divides it down (using the thin-out algorithm) to the 2 K display memory.
HZOom?			Reports the horizontal scaling factor of the display. Example response: HZOOM X1
VZOom	X1/4 X1/2 X1 X2 X4 X8 X16 X32		Selects the vertical scaling factor for the display. A fraction compresses the display data. If an integer number (32) is used, the waveform is expanded by that number of times. The VPOsn command is useful for positioning the zoomed waveform.
VZOom?			Reports the vertical scaling factor of the display. Example response: VZOOM X1
VPOsn	<NRx>		Sets the vertical zero offset for the displayed channel. This is useful when using YZOom. <NRx>::= –2048 to 2047 in steps of 1. If <NRx> is not within the limitation determined by vertical zoom rate, <NRx> is truncated and limited to a legal value, and a warning SRQ is issued if EXW is on. When queried, the value is <NR1>.
VPOsn?			Reports the vertical zero offset value for the displayed channel, which is returned in <NR1> format. Example response: VPOSN 0

Table 3-5
CURSOR MEASUREMENT GROUP

Header	Argument	Link Argument	Description
MEASure	VOLt:	ON OFF	Enables voltage measurement mode.
	TIMe:	ON OFF	Enables time measurement mode.
	FREquency:	ON OFF	Enables the 1/T or 1/ΔT measurement mode. 1/T, 1/ΔT and external clock are incompatible. If requested, an error SRQ is issued if EXR is on. If one cursor is on (using the CURsor commands listed in Table 5-5), the measurements are made relative to zero volts and the trigger point. If two cursors are on, the measurements are made relative to the difference between cursor 1 and cursor 2.
MEASure?	VOLt TIMe FREquency		Reports status of the selected measurement mode.
MEASure?			Reports status of all the measurement modes. Example response:MEASURE VOLT:OFF, TIME:OFF,FREQUENCY:OFF
Value?			Reports the measured value, which is returned in <NR3> format. If the cursor function is off or a measurement cannot be made, ##### is returned. If requested measurement can't be made because time scale is different between CH1 and CH2, an error SRQ is issued if EXR is on.

Table 3-6
WAVEFORM TRANSFER GROUP

Header	Argument	Link Argument	Description
WFMpre	WFId:	"<ascii string>"	Query only; ignored if sent as setting command. Responds with the waveform source and location (e.g.,WFMPRE WFID:"CH1_LOCATION1").
	ENCdg:	BINary	Query only; ignored if sent as setting command. Responds with curve encoding, which is fixed at binary (e.g., WFMPRE ENCDG:BINARY).
	NR.Pt:	<NR1>	Query only; ignored if sent as setting command. Responds with the number of data points in the source waveform, which is set by the DATA COUNT command (e.g., WFMPRE NR.PT:<NR x>).
	PT.Fmt:	Y	Defines how to interpret the curve data. Y format means that x information is implicit and the data points set are the y value, and each point is sequential in time order. ENV format is used for enveloped waveforms. Data is sent in format y1max, y1min, y2max, y2min, . . ., and when received by the RTD 710A, it is treated as enveloped data.
	XINcr:	<NRx>	Sets the sample interval value for waveforms input by the CURVe command. If XUNit is SEC, <NRx> is from 5.0E–9 to 200E–3. If XUNit is EXT, <NRx> is from 4.0E+0 to 1.0E+7.
	YZEro:	<NRx>	Sets the input Offset for waveforms input by the CURVe command. If <NRx> is illegal, it is truncated and limited to match a legal sample interval, and a warning SRQ is issued if EXW is on. When queried, the value is <NR1>. <NRx>::=–199% to 199% of full scale in 1% steps.
	PT.Off:	<NRx>	Sets trigger delay value for waveforms input by CURVe command. A negative number indicates pretrigger.
	XUNit:	SEC EXT	Sets the unit of measure for the Sample interval value for waveforms input by the CURVe command. If XUNit is SEC, the XINcr value is in seconds. If XUNit is EXT, the XINcr value is the scaling multiplication factor for the external clock.
	YOFf:	512	Query only; ignored if sent as setting command. Responds with the ground level of the input with 0% Offset, which is fixed at the center of full scale range (i.e., 512).

More Waveform Transfer Group Commands on next page

Table 3-6 (Cont)
WAVEFORM TRANSFER GROUP

Header	Argument	Link Argument	Description
	YMUlt:	<NRx>	Sets the input Range value for waveform input by the CURVe command. <NRx>::=1.0E–1 to 5.0E+2. When an illegal value is received, the argument is truncated and limited to a legal value, and a warning SRQ is issued if EXW is on. When queried, the value is <NR3>. Volts per point is determined by: 2 * YMULT/1024.
	YUNit:	V	Query only; ignored if sent as setting command. Responds with the units of measure for the input Range value, which is fixed at Volts.
	BYT/nr:	2	Query only; ignored if sent as setting command. Responds with the width of the binary data field, which is fixed at two bytes per point.
	BN.Fmt:	RP	Query only; ignored if sent as setting command. Responds with the binary data format of the curve data, which is fixed as a right-justified, binary-positive integer.
	BIT/nr:	10	Query only; ignored if sent as setting command. Responds with the binary data precision, which is 10 bits.
	BKPt:	<NRx>:<NRx>	Sets the breakpoint location and sample interval for waveforms input by the CURVe command. The first <NRx> defines the breakpoint location with reference to the trigger point. The second <NRx> defines the sample interval associated with the breakpoint. If BKPt:0:0 is sent, it clears all previously set breakpoints. If an <NRx> is illegal, it is truncated or limited to a legal value, and a warning SRQ is issued if EXW is on.
WFMpre?	WFId	ENCdg NR.Pt PT.Fmt XINcr PT.Off XUNit YZEro YOFf YMUlt YUNit BYT/nr BN.FMT BIT/nr BKPt	Reports the selected waveform preamble item. ENCdg response fixed: BINary. NR.Pt is in <NR1> format. XINcr is in <NR3> format. PT.Off is in <NR1> format. YZEro is in <NR1> format. YOFf response fixed: 512 YMUlt is in <NR3> format. YUNit response fixed: V BYT/nr response fixed: 2 BN.Fmt response fixed: RP BIT/nr response fixed: 10 BKPt is in <NR1>:<NR3> format.

More Waveform Transfer Group Commands on next page

Table 3-6 (Cont)
WAVEFORM TRANSFER GROUP

Header	Argument	Link Argument	Description
WFMpre?			Reports all waveform preamble items. Example response:WFMPRE WFID:"CH1_LOCATION1", ENCDG:BINARY,NR.PT:2048,XUNIT:SEC, XINCR:10.0E–9,PT.FMT:Y,PT.OFF:–400, YZERO:0,YOFF:512,YMULT:2.5E+0,YUNIT:V, BYT/NR:2,N.FMT:RP,BIT/NR:10, BKPT:0:10.0E–9,BKPT:520:100.0E–9
CURVe	<waveform data>		Sends waveform data from the controller to the RTD 710A without the preamble. The format of the argument <waveform data> is defined by the DATa BFOrmat command. Target memory channel, memory segment, start address, etc., are also defined by using the DATa command.
CURVe?			Sends binary block waveform data from the RTD 710A to the controller without sending the preamble. The source memory channel, memory segment, start address, etc., are defined by using the DATa command. For details of <binary block data>, refer to the DATa BFOrmat command description. Repeated binary block transfer is acceptable when DATa BFOrmat is set to BINary, not acceptable with ARBitrary mode. For binary block data details, see DATa BFOrmat command description.
WAVfrm?			Performs the same function as sending the WFMpre? and CURVe? queries. The RTD 710A sends the waveform preamble and then the curve data for the waveform specified by the DATa command.

More Waveform Transfer Group Commands on next page

Table 3-6 (Cont)
WAVEFORM TRANSFER GROUP

Header	Argument	Link Argument	Description
DATa	CHAnnel:	CH1 CH2	Selects the memory channel to be transferred. If an illegal channel is requested, an error SRQ is issued, if EXR is on, and the command is ignored.
	LOCation:	<NRx>	Assigns the memory location number for the selected memory channel. <NRx>::=1 to 256 for VMOde CH1. <NRx>::=1 to 128 for VMOde DUAl. If <NRx> is illegal, an error SRQ is issued if EXR is on, and the command is ignored.
	STArt:	<NRx>	Designates the starting memory address at which waveform data is sent or received. <NRx>::=262136 to 524277. Allowable <NRx> range can be calculated as: (TRIgger DELay) +(record LENgth–1)–2. If <NRx> is illegal, an error SRQ is issued if EXR is on, and the command is ignored. Use WFMpre PT.Off to determine start from first point.
	COUnt:	<NRx>	Sets the number of data points to be transferred. <NRx>::=2 to 262144. If <NRx> does not match the other setting, an error SRQ is issued if EXR is on, and the command is ignored.
	BFOrmat:	BINary ARBitrary	Designates the block format of the curve data transmission. BINary is conventional binary block format defined in the Codes and Format standard. The format is as follows: Curve %ccbbb. . . .bbs [{,%ccbbb. . . . bbs }..,] Where c=byte-count; b=binary data, high byte first; and s=checksum. ARBitrary is the Definite Length Arbitrary Block format defined in IEEE Std 488.2. The format is as follows: Curve #xc. .cbbb. . . . . . . .bbs Where x=number of bytes in the byte-count number; c=byte-count; b=binary data, high byte first; and s=checksum.

More Waveform Transfer Group Commands on next page

Table 3-6 (Cont)
WAVEFORM TRANSFER GROUP

Header	Argument	Link Argument	Description
	BSIze:	<NRx>	Sets the block size of the binary data (2-byte data) string when using BFOrmat BINary. <NRx>::=1024, 2048, 4096, 8192, or 16384. If <NRx> exceeds the value above, an error SRQ is issued if EXR is on, and the command is ignored. This argument is used for BINary format transfer. It is not used for ARBitrary format transfer.
DATa?	CHAnnel LOCation STArt COUnt BSIze		Reports the selected DATa setting. Note: all responses except BFOrmat are in <NR1> format.
DATa?			Reports all DATa settings. Example response: DATA CHANNEL:CH1,LOCATION:1, START:–400,COUNT:2048,BFORMAT:BINARY, BSIZE:2048

Table 3-7
CALIBRATION GROUP

Header	Argument	Link Argument	Description
AUTocal			Initiates the auto-calibration function. If an error SRQ occurs, any command other than TESt causes the error to be ignored and normal operation to resume.
MONcal	STArt RESet		STArt connects the monitor calibration signal to the monitor output (X, Y, and Z). RESet disconnects the monitor calibration signal.

Table 3-8
SELF-TEST GROUP

Header	Argument	Link Argument	Description
TESt	MODe:	ON OFF	Enables the self-test mode. All test commands, including TESt, RUN, ERRor?, STEp, HALt, DEPosit, EXEcute, and FETch are acceptable when MODe is set to ON, but not when set to OFF. If an attempt is made to select a test command when MODe is off, an error SRQ is issued if EXR is on. TESt MODe is acceptable at any time.
	TYPe:	SELFDiag EXTDiag	Selects the level of self-test diagnostics to be run. SELFDiag selects the standard power on diagnostics (Level 1). EXTDiag selects extended diagnostics (Level 2). See TESt NUMber below. <NRx> indicates the test number within the TYPe (extended diagnostics) that is to be run. If the TYPe is SELFDiag, NUMber is reset to 0 when a RUN command is received. If <NRx> is illegal, an error SRQ is issued if EXR is on, and the command is ignored.
	NUMber:	<NRx>	Selects the extended diagnostics test number. <NRx>::=00000 to 99999; defaults 00000 to TESt TYPe:SELFDiag.
TESt	DATa	<NRx>	Sets data for the test defined by NUMber:<NRx>. When queried, the value is <NRx> for Range setting and <NR1> for other settings. If <NRx> is illegal, it is truncated to the nearest legal number.
	LOOp:	ONE FAIL PASs CONt	Causes the selected self-test to run in a loop until the link argument condition is satisfied, or a HALt command is received.
TESt?	MODe TYPe NUMber DATa LOOp		Reports the current TESt setting. Number is in <NR1> format. Data is in <NR3> format for range setting and <NR1> for other settings.

More Self-Test Group Commands on next page

Table 3-8 (Cont)
SELF-TEST GROUP

Header	Argument	Link Argument	Description
TESt?			Reports all TESt settings. Example response: TEST MODE:OFF,TYPE:EXTDIAG,NUMBER:0, LOOP:ONE,DATA:0
RUN			Starts the self-test function selected by NUMber:<NRx>. Any tests in this test hierarchy below the selected number, except those requiring operator intervention, are run. If OPC is ON, an operation completion SRQ is issued when the test is finished. If a test is unsuccessful, an SRQ is issued if INR is on. An ERRor? query may then be used to retrieve the error code.
STEp			Causes the current test to advance to the next step in the sequence and execute that step. If OPC is ON, an operation complete SRQ is issued when the step is finished.
ERRor?			Reports the string of error codes (up to 40) that occurred during the last RUN command. Zero is returned when there are no errors.
HALt			Stops any test being executed. Specifically used to stop a looping test.

Table 3-9
UTILITY COMMAND GROUP

Header	Argument	Link Argument	Description
ID?			Queries for the RTD 710A ID message which is returned as: ID SONY_TEK/RTD710A,V81.1,F<rom><patch> <rom>::=1-character ROM version <patch>::=2-character patch version Example response: ID SONY_TEK/RTD710A, V81.1, F1.00
SET?			Reports the current instrument control settings, which are returned in an ASCII string that can be sent to the RTD 710A for setup. Example response: CH1 RANGE:2.5E+0,UNIT:PERCENT,OFFSET:0,COUPLING:AC;CH2 RANGE:50.0E+0, UNIT:PERCENT,OFFSET:0,COUPLING:AC;VMODE DUAL;BWLIM OFF;ARM MODE:INT, DELAY:0.0E+0;SAMPLE MODE:NORM,CLOCK:INT,INTERVAL:10.0E–9; RECORD MODE:NORM,AVERAGE:2,ENVELOPE:1,LOCATION:1; BREAKPOINT UNIT:POINT,SET:0:10.0E–9,SET:520:100.0E–9;LENGTH 2048; TRIGGER MODE:AUTO,DUNIT:POINT,DELAY:–400,COUPLING:DC,SOURCE:CH1, SLOPE:POSITIVE,LUNIT:PERCENT,LEV1:0,LEV2:0,LCN START:PREFLD; DISPLAY CHANNEL:CH1,LOCATION:1,MODE:YT,INTERPOL:LINE;HZOOMX1; VZOOMX1;VPOSN 0;CURSOR ONE:OFF,TWO:OFF,SCROLL:ALIGN
HELp?			Note: response is a list of all valid command headers available to the user. Example response: HELP ARM,AUTOCAL,BASE,BREAKPOINT,BWLIM,CER,CH1,CH2,CURSOR,CURVE,DATA, DEVICE,DISPLAY,DT,ERROR,EVENT,EXR,EXW,HALT,HELP,HOLD,HZOOM,ID,INIT,INR, LENGTH,MAXIMUM,MEAN,MEASURE,MID,MINIMUM,MONCAL,NCROSS,NUMACQ,OPC, OVER,PCROSS,PEAKTOPEAK,PLOT,RECALL,RECORD,REPEAT,RQS,RUN,SAMPLE,SAVE, SET,SRQ,STEP,TEST,TOP,TRIGGER,USER,VALUE,VMODE,VPOSN,VZOOM,WAVFRM, WFMPRE,WINDOW,WRI

More Utility Group Commands on next page

Table 3-9 (Cont)
UTILITY COMMAND GROUP

Header	Argument	Link Argument	Description
SAVe	<NRx>		Stores the current instrument front panel setting into non-volatile memory at location <NRx>. <NRx>::=1 to 20. If <NRx> is illegal, it is truncated or limited to the nearest legal number and EXW is issued if on.
RECall	<NRx>		Recalls the instrument front panel settings from non-volatile memory location <NRx>. <NRx>::=1 to 20. If <NRx> is illegal, it is truncated or limited to the nearest legal number and EXW is issued if on. If requested recall location is empty, an error SRQ is issued if EXR is on, and the command is ignored.
DEVice	SETtings:	ON OFF	Determines whether instrument settings designated by the DISplay LOCation are plotted.
	GRAt:	ON OFF	Determines whether graticule is plotted.
	WAVfrm	ON OFF	Determines whether waveforms designated by the DISplay LOCation command are plotted.
PLOt			Command only. Causes the instrument to output a data string whose format is determined by the DEVice command. If the instrument is in acquisition when PLOt is received and is set to talker, the plotting waits until the acquisition is completed and enters the HOLd state. If display location is 0 (display off) when PLOt command is received, an execution error SRQ is issued if EXR is on. During PLOt, all front panel keys are locked except RQS/ID key. DCL or SDC aborts the plot.
DEVice	SETtings GRAt WAVfrm		Returns the device mode setting. Example response: DEVICE SETTINGS:OFF,GRAT:ON, WAVFRM:ON.

Table 3-10
DEVICE TRIGGER GROUP

Header	Argument	Link Argument	Description
DT	ON OFF		Enables the RTD 710A to respond to the GPIB GET command.
DT?			Reports the status of the DT setting. Example response: DT OFF

Table 3-11
INITIALIZATION GROUP

Header	Argument	Link Argument	Description
INIt	[ALL]		Executes all of the INIt command arguments (PANel, WAVfrm, and GPIb). The [ALL] argument is optional.
	PANel		Resets all front panel controls to default settings. See Section 3, in Operators Manual for default settings.
	WAVfrm		Clears the waveform memory to zero.
	GPIb		Clears the event buffer; sets DT:OFF, sets Service Request Group commands to: RQS:ON, OVER:OFF, WRI:OFF, CER:ON, EXR:ON, INR:ON, EXW:ON, OPC:OFF, USER:ON; and sets DATa, WINdow, and DEVice as shown below.
			DATa CHAnnel:CH1 LOCation:1 STArt:(Trigger Delay value of CH1, Location 1) COUnt:(Record Length value) BSIze:(Record Length value, not to exceed 16384) BFOrmat:BINary
			WINdow CHAnnel:CH1 LOCation:1 STArt:(Trigger Delay value of CH1, Location 1) STOp:(Trigger Delay value)+(Record Length value)–1 LEVel:512
			DEVice SETting:ON GRAt:ON WAVfrm:ON

Table 3-12
SERVICE REQUEST CONTROL AND EVENT QUERY GROUP

Header	Argument	Link Argument	Description
RQS	ON OFF		Enables all SRQ functions except power on and fatal error.
RQS?			Reports the RQS status.
OVEr	ON OFF		Enables the Overrange SRQ function. Issues a status 193 or 209, event codes 752 through 755, when on.
OVEr?			Reports the OVEr status.
WRI	ON OFF		Enables the Acquisition Complete SRQ function. When on, the RTD 710A will issue an OPC (status 208) and event code 750.
WRI?			Reports the WRI status.
CER	ON OFF		Enables the Command Error SRQ function. Issues a status 97 or 113 when on.
CER?			Reports the CER status.
EXR	ON OFF		Enables the Execution Error SRQ function. Issues a status 98 or 114 when on.
EXR?			Reports the EXR status.
INR	ON OFF		Enables the Internal Error SRQ function. Issues a status 99 or 115 when on.
INR?			Reports the INR status.
EXW	ON OFF		Enables the Execution Warning SRQ function. Issues a status 101 or 117 when on.
EXW?			Reports the EXW status.
OPC	ON OFF		Enables the Operation Complete SRQ function. Issues a status 66 or 82 when on.
OPC?			Reports the OPC status.
USEr	ON OFF		Enables the front-panel generated SRQ. Issues a status 67 or 83 when on..
USEr?			Reports the USEr status.
SRQ?			Reports the status of all Service Request Group commands. Example response:OVER OFF; USER ON;WRI ON;RQS ON;CER ON;EXR ON;EXW ON;INR ON;OPC OFF
EVEnt?			Queries for the most recent event code; returns 0 if none exists. It also clears the event buffer. See Section 5, SRQ & Event Codes for specific Event Codes.

Table 3-13
INTERNAL WAVEFORM ANALYSIS GROUP

Header	Argument	Link Argument	Description
WINdow	CHAnnel:	CH1 CH2	Selects the source waveform channel for subsequent waveform analysis commands. If VMOde CH1 is selected, CH2 cannot be selected. If CH2 is requested, an error SRQ is issued if EXR is on.
	LOCation:	<NRx>	Sets the waveform memory segment for subsequent waveform analysis commands. <NRx>::=1 to 256. If <NRx> does not match other settings, an error SRQ is issued if EXR is on, and the command is ignored.
	STArt:	<NRx>	Sets the start of the memory address for the actual measurement queries. <NRx>::=–262136 to 524279. <NRx> may be calculated as follows: (TRIgger DELay)+(record LENgth–1). <NRx> must be less than stop address. If they are not of the proper size relationship, they are swapped. If <NRx> is illegal, an error SRQ is issued if EXR is on, and the command is ignored.
	STOp:	<NRx>	Sets the end of the address for the waveform measurement queries. Refer to STArt for <NRx> values.
	LEVel:	<NRx>	Sets the vertical level for PCRoss and NCRoss. <NRx>::=1 to 1023.
WINdow?	CHAnnel LOCation STArt STOp LEVel		Queries for the selected window settings. LOCation is in <NR1> format. STArt is in <NR1> format. STOp is in <NR1> format. LEVel is in <NR1> format. Example response:WINDOW CHANNEL:CH1, LOCATION:1,START:–400,STOP:1647, LEVEL:512

More Internal Waveform Analysis Group Commands on next page

Table 3-13
INTERNAL WAVEFORM ANALYSIS GROUP

Header	Argument	Link Argument	Description
MAXimum?			Queries for the maximum value of the acquired waveform between the STArt and STOp addresses. The value returned (0-1023) is in <NR1> format.
MINimum?			Queries for the minimum value of the acquired waveform between the STArt and STOp addresses. The value returned (0-1023) is in <NR1> format.
TOP?			Queries for the top value of the acquired waveform between the STArt and STOp addresses. The value returned (0-1023), which is calculated by the histogram method, is in <NR1> format.
BASE?			Queries for the base value of the acquired waveform between the STArt and STOp addresses. The value returned (0-1023), which is calculated by the histogram method, is in <NR1> format.

More Internal Waveform Analysis Group Commands on next page

Table 3-13 (Cont)
INTERNAL WAVEFORM ANALYSIS GROUP

Header	Argument	Link Argument	Description
PCRoss?			Queries for the waveform point within the STArt/STOp window where the positive-going waveform first crosses the LEVel setting. The value returned, which is linearly interpolated, is in <NR1> or <NR2> format. If there is no positive-crossing point, the ASCII string ##### is returned.
NCRoss?			Queries for the waveform point within the STArt/STOp window where the negative-going waveform first crosses the LEVel setting. The value returned, which is linearly interpolated, is in <NR1> or <NR2> format. If there is no negative-crossing point, the ASCII string ##### is returned.
MID?			Queries for the vertical mid level value within the STArt/STOp window of the acquired waveform. The returned value (0-1023) is in <NR1> format, and is calculated as follows: MID=(max+min)/2.
MEAN?			Queries for the vertical algebraic mean value within the STArt/STOp window of the acquired waveform. The returned value is in <NR1> format and is calculated as follows: Mean= $\sum_{i=1}^{N}$ W(i) where W is waveform data.
PEAktopeak?			Reports the vertical peak-to-peak value within the STArt/STOp window of the acquired waveform. The returned value (0-1023) is in <NR1> format.

ADVANCED GPIB INFORMATION

This section contains helpful information about the following topics:

- Transferring waveforms
- Using the internal signal processing
- Transferring waveforms to the RTD 710A
- Sending settings
- Updating the waveform preamble

Query Commands to Transfer Binary Waveforms

You select from several options to read a waveform from the RTD 710A.

These include:

1) The source channel and record location
2) Is an entire waveform record or only a portion of a record to be acquired?
3) Is the waveform to be broken up into smaller portions (e.g., sending a 64K waveform in 16K portions using repeated binary block format)
4) Sending an arbitrary length waveform up to 256K in length

To ensure reliable waveform transfer, the following six commands must be sent to the RTD 710A:

CHA - Selects channel 1 or channel 2 as the waveform source.

LOC - Selects the record location to transfer from channel 1 or 2 defined above.

START - Defines the first point to be transferred. If acquiring an entire record, use the trigger delay value in the waveform preamble (PT.OFF) to define the starting point.

COUNT - Defines the number of points to be sent.

BSIZE - Defines the block size of the waveform to be transferred. If the waveform is greater than 16K words, the BSIZE is automatically set to 16K words. Most computers can read an 8K waveform into a single variable. If you want to transfer a waveform with a longer record length, set the BSIZE to the maximum length the computer can handle and read the waveform into a number of variables. The REPEATED BINARY BLOCK format is explained later in this section.

BFOrmat - Designates the block format of the curve data transmission, either binary block or arbitrary format.

Binary Waveform Formats

The RTD 710A can send binary waveform information using three formats:

BINARY BLOCK - Transfer the entire waveform in one block. The BSIZE command is the same size as the total number of points (specified by the COUNT command). Most computers can read 8K words in one block. The binary waveform date is transferred as shown in Figure 4-1.

Example: `CURVE %ccbbb..............bbs`

```
CURVE <space>%ccbbb.....bbs[{a%ccbbb.....bbs}..]
```

Where: c= byte-count;

b= binary data, high byte first; and

s= checksum

<space> = ASCII character 32

Fig. 4-1 Waveform data structure using repeat binary block

REPEATED BINARY BLOCK FORMAT - Breaks the data into smaller blocks so a computer can read a large waveform into many arrays. The BSIZE command determines the length of the blocks..

Example: `CURVE %ccbbbb.....bbs,%ccbbb.... bbs,%ccbbb....bbs`

ARBITRARY LENGTH FORMAT - It is the definite length arbitrary block format defined in the IEEE-488.2 standard. This format allows transfer of any number of waveform data words from one to the waveform record length (up to 256K words). This format is selected by sending the BFOrmat command (Table 3-6). The arbitrary length format is shown in Figure 4-2.

```
CURVE <space>#xc..cbbb.....bbs
```

Where: x= number of bytes in byte-count number;

c= byte-count b= binary data, high byte first; and

s= checksum

<space>=ASCII character 32

Fig. 4-2 Waveform data structure - Arbitrary length format

The Byte Count

The byte count reflects the number of bytes being sent, including the checksum. For each waveform data point, the RTD 710A sends two waveform data bytes. B = High byte and b = Low byte of the byte count. The following equation determines byte count:

Byte Count = (B * 256) + b.

Number of data points = byte count - 1

For example, if the waveform has 16384 points, the byte count reflects sending the high and low waveform data bytes so the byte count equals 16384 * 2 + 1 = 32769.

High and Low Bytes

Because the RTD 710A is a 10-bit digitizer, the waveform data must be sent in two bytes. It is sent High Byte (highbyte), Low Byte (lowbyte) for each data point. Using the following formula reduces the data to a single word:

Waveform data point= (highbyte * 256) + lowbyte

Waveform Acquisition Examples

Example 1: Acquiring an entire record from channel 1, record 20.

1) Specify the waveform to be acquired and query for the waveform preamble by sending the following RTD 710A commands in the program sequence listed below.

```
100    Dim wfmpre$(300)
110    Print #RTD:"DAT CHA:CH1,LOC:20;WFMPRE?"
130    Input #RTD:wfmpre$
```

The waveform preamble contains the number of points and the trigger delay, as well as other information used for scaling the waveform.

2) Parse out the number of points (NR.PT) and the trigger delay (PT.OFF). The example program lines below assumes a user-defined function called "ARG" that finds the search string (the second argument) and returns the value behind the search value.

```
150    numpts=arg(wfmpre$,"NR.PT")
160    trigdely=arg(wfmpre$,"PT.OFF")
```

There is example code for the 4041, HP 200 series, IBM PC using BASICA, and SPS BASIC in section 6 showing how to parse out the desired arguments.

3) Send the RTD 710A command string necessary for proper definition of the waveform to be transferred.

```
200    Print #RTD:"DAT CH:CH1,LOC:20,START:"&str$(trigdely)&",COUNT:"&str$
       (numpts)&",BSIZE:"&str$(numpts)&";CURVE?"
```

4) The RTD 710A is now ready to transfer the waveform to the controller. Use the commands specific for a particular computer to read in the waveform information. The example programs in section 6 show how to read in a waveform for supported computers.

Example 2: Acquiring a partial waveform defined by cursors 1 and 2 from channel 1, record 20.

1) Read the cursor positions from the RTD 710A using the following program sequence. This assumes the cursors are turned on and positioned.

```
100    Print #RTD:"CURS ONE:DISP1;CURS TWO:DISP1"
110    Input #RTD prompt "CURS? POS1":null,start
120    Input #RTD prompt "CURS? POS2":null,stoppt
```

The variable 'null' reads the first number sent in the response, which is the cursor number (e.g., CURSOR POS1:235; Null=1, Start=235)

2) Determine the difference between 'start' and 'stoppt' to get the number of points and store in variable 'numpts'. Numpts=stoppt-start+1

3) Send the RTD 710A command string necessary for proper definition of the waveform to be transferred.

```
Print #RTD:"DAT CHA:CH1,LOC:20,START: "&str$(start)&",COUNT:"&str$(numpts)
&",BSIZE:"&str$(numpts)&";CURVE?"
```

4) The RTD 710A is now ready to transfer the waveform to the controller.

The BSIZE Command

The BSIZe command defines the number of words of waveform data sent in each block when not using arbitrary format. Let's look at transferring an 8K word waveform with both methods. With the first, BSIZE = COUNT (8K words); with the second, BIZE = COUNT/2 (COUNT = 6K words, BSIZE = 8K words).

Example 1, Binary Block Format: BSIZE = COUNT = 8192. The following commands are sent to the RTD 710A to read in the 8K of data from channel 1, record location 2, and trigger delay of -250 points (or 250 points of pre-trigger).

```
Print #RTD:"DAT CHA:CH1,LOC:2,START:-250,COUNT:8192,BFOR:BIN,BSIZE:8192;
CURVE?"
```

Because COUNT and BSIZE are equal, the entire waveform is transferred in one block.

The waveform data can be read into a string variable as shown below:

```
100 ! 2 times the number of points because each data point is sent as two
      bytes, 9 bytes for 'CURVE %Bh' and 1 byte for the checksum.
110 !
120   Dim wfm$((2 * numpts)+10)
125   PRINT #RTD:"CURVE?"
130   Input #RTD:wfm$
```

Example 2, Repeated Binary Block Format: BSIZE = COUNT/2, COUNT = 16384, BSIZE = 8192. The following commands are sent to the RTD 710A to read in the 8K of data from channel 1, record location 2, and trigger delay of -250 points with BSIZE equal to 8192.

```
Print #RTD:"DAT CHA:CH1,LOC:2,START: -250,COUNT:16384,BFOR:BIN,BSIZE:8192;
CURVE?"
```

Because BSIZE is smaller than COUNT, the waveform data will be sent using repeated binary block format.

The waveform can be read into two string variables as follows:

```
100   Dim header$ to 6 ! Dimensioned to read the 'CURVE<space>'
110 !
120 !
```

The next dimension statement allocates 16384 bytes of waveform data (represents the first 8192 data points) plus 1 byte for the percent sign '%', 2 bytes for byte count, and 1 byte for checksum.

```
130 !
140   Dim wfm1$ to (16384+4)
150   Dim comma$ to 1 !
      Dimensioned to read the comma between the two blocks.
160 !
170 !
```

The next dimension statement allocates 16384 bytes of waveform data (represents the second 8192 data points) plus 1 byte for the percent sign '%', 2 bytes for byte count, and 1 byte for checksum.

```
180 !
190   Dim wfm2$ to (16384+4)
195   PRINT #RTD:"CURVE?"
200   Input #RTD:header$,wfm1$,comma$,wfm2$
```

The waveform strings 'wfm1$' and 'wfm2$' are now ready to be parsed into integer arrays and scaled.

Using the Internal Signal Processing Commands

There are several signal processing commands available over the GPIB. These commands can be used to segment a waveform for transfer or for getting answers without transferring a waveform to the computer. The signal processing commands are explained in Table 3-13 (Internal Waveform Analysis Group) of the command reference in Section 3. Listed below is a summary of these commands:

WINdow This command uses the arguments "channel," "record location," "start point," "stop point," and "level" to tell where to perform subsequent signal processing operations.

MAX?, MIN? Queries for the maximum and minimum inside the defined window.

TOP?, BASE? Queries for the top and base based on the histogram method. The histogram helps negate ringing on pulses to give more accurate pulse parameters.

ID? Queries for the mid point between the maximum and minimum inside the defined window.

MEAN? Queries for the vertical algebraic mean between the maximum and minimum inside the defined window.

PEAKtopeak? Queries for the vertical peak topeak inside the defined window.

PCROSS?, NCROSS? Pcross and Ncross find positive and negative crossing locations defined by the level argument in the window command.

The commands listed above return unscaled values either in clock points (for PCROSS and NCROSS), and digitizer units (0 to 1023; for all vertical queries). The "Vertical Value Scaling" example program illustrates how to make voltage measurements. A timing measurement example is shown in "Horizontal Value Scaling".

Vertical Value Scaling

To scale a vertical value, two pieces of information must be obtained; the input range and the dc offset.

Assume a waveform has been captured in channel 1, record 1, with pretrigger of 400 points and a record length of 2048 points. The measurements are to be made over the entire record. The following 4041 program example shows how to make max, min and peak-to-peak voltage measurements.

NOTE: If throughput is important, voltage measurements can be made without scaling the returned value.

```
10 !  Program example showing how to specify a target waveform to make 15
      measurements using the internal signal processing of the RTD 710A.
20 !
25 !  This example will make max, min, & peak-to-peak measurements and
30 !  scale them into voltage values.
35 !
40 !  This program assumes; num pts=2048, trig delay= -400 pts.
45 !
50    Integer rtd,cha,loc,strt,cnt,status,add
55    Rtd=1 ! RTD 710A at address 1
60    Cha=1 ! Channel 1
65    Loc=1 ! Location 1
70    Strt=-400 ! 400 points of pretrigger
75    Cnt=2048 ! 2K waveform
90 !
95 !  Set to single sweep, turn on WRI ON srq capability
```

```
100 !
105  Print #rtd:"TRIG MOD:SGL;WRI ON"
110 !
115 ! Define channel, loc, etc. to read input range and offset
120 !
125  Print #rtd:"DAT CHA:CH"&str$(cha)&",LOC:"&str$(loc)&
     ",STA:"&str$(strt)&",COU:"&str$(cnt)
130  !
135  !Define cha, loc, etc. for signal processing measurements.
140  !
145  Print #rtd:"WIND CHA:CH"&str$(cha)&",LOC:"&str$(loc)&
     ",STA:"&str$(strt)&",STOP:"&str$(cnt+strt-1)
150  !
155  !Read input range and dc offset for successive voltage measurements.
160  !
165  Input #rtd prompt "WFM? YMULT":ym$
170  Ym=val(ym$)
175  Input #rtd prompt "WFM? YZERO":yz$
180  Yz=val(yz$)
185  !
190  !Turn ch1 and ch2 displays off to increase throughput
195  !
200  Input #rtd prompt "DIS CHA:CH1;DIS? LOC":ch1dis$
205  Input #rtd prompt "DIS CHA:CH2;DIS? LOC":ch2dis$
210  Print #rtd:"DIS LOC:0;DIS CHA:CH1,LOC:0"
215 !
220  For loop=1 to 5 ! Make 5 peak-to-peak measurements
225  Print #rtd:"HOLD RESET" ! Acquire a new waveform
235  Peak=voltmeas("PEAK?",yz,ym)
240  Maxum=voltmeas("MAX?",yz,ym)
245  Minum=voltmeas("MIN?",yz,ym)
250  Print "Pk-pk volts:";peak;" volts, Max:";maxum;" volts, Min:";minum;"
     volts"
255  Next loop
260 !
265 ! Turn displays back on
270 !
275  Print #rtd:"DIS CHA:CH1;"&ch1dis$
280  Print #rtd:"DIS CHA:CH2;"&ch2dis$
285  End
300  Function voltmeas(query$,yz,ym) local ivolt,vpb,vshift,measur
305 !
310 ! Function to query the RTD 710A using the voltage signal processing
315 ! command stored in 'query$' and convert it into a voltage value.
320 !
325  Integer ivolt
330  Vshift=0
335  Input #rtd prompt query$:ivolt
340  Vpb=2*ym/1024 ! Determine volts per bit
345  If pos(query$,"PEAK",1) then goto measur
350  Vshift=512-(yz*5.12) ! Take dc offset into account
355  Measur: voltmeas=(ivolt-vshift)vpb
360  Return
365  End
```

Horizontal Value Scaling

The complexity of scaling a horizontal measurement depends on whether or not breakpoints are used. If breakpoints are not used, the XINCR value in the waveform preamble is the only value needed. If breakpoints are used, the example programs show how to parse the breakpoint information into numbers.

Sending 16K and longer waveforms to RTD 710A without arbitrary block format

When a 16K word or larger waveform is read in from the RTD 710A it is sent using Repeated Binary Block Format. The RTD 710A does not read waveforms using Repeated Binary Block Format. However, there is a way to send up to a 64K waveform. You change the starting location of the data and send up to 16K blocks of the waveform. Here is an example 4041 program illustrating how to send a waveform.

4041 Example program :

```
100 !
110 ! Get a waveform from CH1, location 1, start -400, count 16384, block
      size 8192
120 ! Then send the waveform back to the RTD 710A location 2.
130 !
140 ! RTD 710A settings:  Record length 16384, Trigger delay -400, GPIB
      address 1
150 !
160   Open #1:"GPIB(pri=1,eom<o>=):"
170 !
180 ! Get a waveform using Repeated Binary Block Format
190 !
200   Dim header$ to 6 ! Dimension a string variable to read  'CURVE <space>'
210 !
220 ! Dimension two strings to read two 8K Word blocks of waveform data
230 ! 2 bytes/word, 16K bytes total plus 3 bytes for
      '%<bytecount><bytecount>' plus 1 byte for checksum
240 !
250   Dim blk1$ to 16388, blk2$ to 16388
260   Dim comma$ to 1  ! Dimension a string to read the comma between the
      two blocks of waveform data
270   Print #1:"DAT CHA:CH1,LOC:1,STA:-400,COU:16384,BSI:8192"
280   Dim wfmpre$ to 256  ! Dimension string for waveform preamble
290   Input #1 prompt "WFMPRE?":wfmpre$   ! Read waveform preamble
300   Input #1 prompt "CURVE?":header$,blk1$,comma$,blk2$
310 !
320 ! Send the waveform back to the RTD 710A NOT using Repeated Binary
      Block Format
330 !
340   Print #1:"DISPLAY CHA:CH1,LOC:0;DISPLAY CHA:CH2,LOC:0"  ! Turn off
      both channels displays
350   Print #1:wfmpre$
360   Print #1:"DAT CHA:CH1,LOC:2,STA:-400,COU:8192,BSI:8192"
370   Wbyte atn(mta,33)   ! Make RTD 710A a listener
380   Wbyte header$,blk1$,eoi  ! Send first 8K block of waveform data,
      then terminate using EOI
390   Wbyte atn(unt,unl)  ! Untalk & Unlisten the RTD 710A
400   Print #1:"DAT STA:7792"  ! Define the start to be (8192-400)
410   Wbyte atn(mta,33)   ! Make RTD 710A a listener
420   Wbyte header$,blk2$,eoi  ! Send 2nd 8K block of waveform data, then
      terminate using EOI
430   Wbyte atn(unt,unl)  ! Untalk & Unlisten the RTD 710A
440   Print #1:"DISPLAY CHA:CH1,LOC:2"  ! Display channel 1, location 2
450   End
```

A couple of suggestions for reliable waveform transfer and display.

1. When you set the record length, have the RTD 710A trigger (AUTO trig) and fill the memory with garbage data (HOLD RESET works well). This allocates the memory so when you transfer a waveform, it fits perfectly. The reason for this is that the RTD 710A doesn't update waveform memory using the front panel settings until it acquires new data.
2. Turn all displays off until the entire waveform(s) are transferred. Sometimes the display memory is not updated correctly if the waveform is sent with the target location(s) turned on.

Sending settings to the RTD 710A

When settings are sent to the RTD 710A, it checks to see if any commands conflict with the present settings. If it "sees" a conflict, it tells the computer by issuing an execution error SRQ (status 98). Some common conflicts in settings come from being in CH1-ONLY or HIGH-SPEED mode and information concerning channel 2 are received.

Let's look at an example. Let's assume that the RTD 710A is in single channel mode (CH1-ONLY or High-speed mode). When a settings string is sent, the channel 2 vertical information (e.g., RANGE, OFFSET, etc.) comes before the command that sets DUAL or CH1-ONLY mode. When the RTD 710A sees the channel 2 information when in CH1-ONLY mode, it issues a EXECUTION ERROR SRQ {status 98} and specific EVENT code {EVENT 250, CH2 input can't be used in the high-speed sampling mode} stating that this information is in conflict with the present settings.

Four solutions are outlined here;

- Initialize the front panel before sending settings
- Turn off the error message SRQ capability during settings transfer
- Handle the SRQ's with an SRQ handler
- Download various settings and store in RTD 710A's non-volatile front panel memory.

Solution #1: Initialize the front panel before sending settings

```
100 !
110 ! Example program showing how to send settings without SRQ errors by
      initializing
120 ! front panel before sending settings.
130 !
140 ! RTD 710A address 1
150 !
160   Dim setting$ to 550  ! Dimension variable setting$ for 550 characters
170   Print "Set RTD 710A for DUAL CHANNEL mode, Press <return> when
      ready: ";
180   Input null$
190   Print #1:"SET?"  ! Query for settings from RTD 710A
200   Input #1: Setting$  ! Read settings into variable 'Setting$'
210   Print "Now set the RTD 710A for CH1-ONLY mode, Press <return> when
      ready: ";
220   Input null$
230 !
240 ! Send settings to RTD 710A
250 !
260   Print #1:"INIT PANEL"   ! Initialize front panel settings to factory
      default
270   Wait .3  ! Wait short amount of time (300 ms)

280   Print #1:Setting$ ! Send entire front panel settings stored in
      string 'Setting$'
290   .    .    .    .    .
890   .    .    .    .    .
900   End
```

Solution #2: Turn off the error message SRQ capability during settings transfer

```
100 !
110 ! Example program showing how send settings without SRQ errors by
      turning off the
120 ! Execution error SRQ (status 98) capability during the settings
      transfer.
130 !
140 ! RTD 710A address 1
150 !
160   Dim setting$ to 550 ! Dimension variable setting$ for 550 characters
170   Print  "Set  RTD 710A  for DUAL CHANNEL mode, Press <return> when
      ready: ";
180   Input null$
190   Print #1:"SET?"  ! Query for settings from RTD710A
200   Input #1: Setting$  ! Read settings into variable 'Setting$'
210   Print "Now set the RTD 710A for CH1-ONLY mode, Press <return> when
      ready: ";
220   Input null$
230   Print #1:"EXR OFF"   ! Turn off EXECUTION ERROR SRQ capability
240   Print #1:Setting$  ! Send entire front panel settings stored in
      string 'Setting$'
250   Print #1:"EXR ON"   ! Turn EXECUTION ERROR SRQ capability back on
260   .    .    .    .    .
270
890   .    .    .    .    .
900   End
```

Solution #3: Handle the SRQ's with an SRQ handler

```
100 !
110 ! Example program showing how to handle any EXECUTION ERROR SRQ's
120 ! by handling them.
130 !
140 ! RTD 710A address 1
150 !
160   On SRQ then call Srqhndl  ! Set up SRQ handler
170   Enable SRQ  ! In 4041 BASIC, SRQ must be enabled
180   Dim Setting$ to 550  ! Dimension variable setting$ for 550 characters
190   Print "Set RTD 710A for DUAL CHANNEL mode, Press <return> when ready:
      ";
200   Input null$
210   Print #1:"SET?"  ! Query for settings from RTD 710A
220   Input #1: Setting$  ! Read settings into variable 'Setting$'
230   Print "Now set the RTD 710A for CH1-ONLY mode, Press  <return> when
      ready: ";
240   Input null$
250   Print #1:Setting$  ! Send settings to RTD 710A, an SRQ should occur
260   .   .   .   .   .
350   .   .   .   .   .
360   End ! End of main program
500   Sub Srqhndl ! Start of subprogram SRQHNDL
510 !
520 ! Sub to handle service requests (SRQ's) and read the associated EVENT
      code from the

530 ! RTD 710A, Report either an execution warning or error and event code
      onto console device.
540 !
550   Poll status,address;1  ! Perform a serial poll on the RTD 710A
      (address 1).  Status byte in variable 'Status'
560   Print #address:"EVENT?"  ! Query for the EVENT code
570   Input #address: Event  ! Read in EVENT code and store in variable
      'Event'
580 !
590 ! If the either an execution error occurs, print message and ignore
600 !
610   If status=98 or status=114 then Print "Execution Error, Event ";Event
620   Resume
630   End  ! SRQ handler end
```

Solution #4: Download various settings and store in RTD 710A's non-volatile front panel memory.

```
100 !
110 ! Example program showing how send settings without SRQ errors by
      turning off the
120 ! SRQ capability during the settings transfer, storing the settings in
      non-volatile storage, and then
130 ! recalling them later (without any SRQ's).
140 !
150 ! RTD 710A address 1
160 !
170 ! Assume 5 settings are stored in string variables 'setting$(1) thru
      setting$(10)'
180 !
190   Integer i  ! The numeric index must be an integer
200   Dim setting$(5) to 550  ! Dimension variable setting$ as a 5
      dimensional string array with 550 characters in each element
210 !
220 ! Lines 250 to 310 learn 5 setups from the RTD 710A.  Different
      languages may not support a 5
230 ! dimensional string array.  In this case, use 5 different string
      variables and forget the FOR/NEXT loop.
240 !
250   For i=1 to 5  ! Learn 5 different setups
260   Print "Step ";i
270   Print "Set RTD 710A for different settings.  Alternate between DUAL,
      CH1-ONLY, & HI-SPEED modes, Press <return> when ready: ";
280   Input null$
290   Print #1:"SET?"  ! Query for settings from RTD 710A
300   Input #1: Setting$(i)  ! Read settings into variable 'Setting$(i)'
310   Next i
320   Print "Press <Return> when ready to send the 5 different setups: ";
330   Input null$
340 !
350 ! Lines 380 to 450 show how to send 5 setups to the RTD 710A
      non-volatile setup memory.
360 ! The EXECUTION ERROR SRQ is turned off during the settings transfer.
      Note: up to 10 setups
365 ! can be saved this way.
370 !
380   Print #1:"EXR OFF"   ! Turn off EXECUTION ERROR SRQ capability
390   For i=1 to 5
400   Print "Sending setup #";i;" to RTD 710A non-volatile setup storage
      #";i
410   Print #1:Setting$(i)   ! Send entire front panel settings stored in
      string 'Setting$(i)'
420   Print #1:"SAVE ";I  ! Store settings in non-volatile location
      defined by 'I'
430   Next i
440   Print #1:"EXR ON"   ! Turn EXECUTION ERROR SRQ capability back on
450   .   .   .   .   .
600   Print #1:"RECALL 2"  ! Recall settings stored in location #2, no SRQ
      when recalled internally
610   .   .   .   .   .
720   Print #1:"RECALL 5" ! Recall settings stored in location #5, no SRQ
      when recalled internally
730   .   .   .   .   .
900   End
```

Making sure the preamble reflects the waveform data

When front panel settings are changed and new waveform data is acquired, the RTD 710A does not automatically update the waveform preamble. The preamble is updated by sending the INIT GPIB command before asking for the preamble, as shown below:

```
100 !
110 ! Example program showing how to have waveform preamble updated using
      INIT GPIB command
120 !
140 ! RTD 710A address 1
150 !
160   Dim wfmpre$ to 300
170   Print #1:"INIT GPIB" ! Send INIT GPIB.  This will cause the preamble
      to be updated
180   Print #1:"DAT CHA:CH2,LOC:15;WFMPRE?"  ! Ask for waveform preamble
190   Input #1:wfmpre$ ! Read correct waveform preamble for ch1, loc1
200   .   .   .   .   .
210
890   .   .   .   .   .
900   End
```

Section 5
SRQ AND EVENT CODES

Status bytes and event codes are used to report operating status to the controller. They are helpful in program development and troubleshooting.

Responses to the controller divide into two classes: status bytes and event codes. Status bytes indicate instrument condition in general; event codes give a more specific indication of instrument condition. For example, if the RTD 710A asserts a command error SRQ (status 97), the EVENT? query can be sent to find out what exactly is wrong with the command. If the message "MODE CH11" (there is no channel 11) is sent, the RTD 710A responds with a command error SRQ and when queried for the event, it sends an EVENT 103; command argument error.

STATUS BYTES

There are two classes of status bytes: device-dependent and system status. Device-dependent bytes report status bytes unique to the RTD 710A. System status bytes report conditions common to other instruments on the GPIB that conform with the Tektronix Standard Codes and Formats. Table 5-1 lists the RTD 710A status bytes.

Table 5-1
RTD 710A STATUS BYTES

	BINARY		DECIMAL				
			RQS ON		RQS OFF		
TITLE	Bit ----- 7654	3210	Not Busy	Busy	Not Busy	Busy	Priority
System Status							
Command Error	0R1X	0001	97	113	33	49	2
Execution Error	0R1X	0010	98	114	34	50	2
Internal Error	0R1X	0011	99	115	35	51	2
Power Fail	0R1X	0100	100	116	36	52	2
Execution Warning	0R1X	0101	101	117	37	53	2
Power ON	010X	0001	65	81	65	81	1
Operation Complete	0R0X	0010	66	82	2	18	2
User Request	010X	0011	67	83	67	83	2
No Status To Report	000x	0000	0	16	0	16	3
Device Dependent Status							
Waveform Data Write (End For Time Base A)	1R0X	0000	192	208	128	144	3
Input Over/Under Range	1R0X	0010	194	210	130	146	3

Bit 7 = Device Dependent Bit
Bit 6 = RQS Bit
Bit 5 = Error Bit
Bit 4 = Busy Bit

R is set to 1 when the GPIB RQS mode is set to RQS:ON. If the RTD 710A is in the RQS:OFF mode and polled, the status byte is sent without DI06 asserted.

X is the busy bit and is set to 1 if the RTD 710A is busy when the status byte is read.

EVENT CODES

An event code is sent to the controller upon request, usually following the receipt of a status byte. Event codes identify instrument status more specifically than status bytes, thus further aiding the user in determining instrument status. To receive an event code, the controller must first send an event query to the RTD 710A.

This event is in the form: EVEnt?

The response is: EVENT <NR1>

The value <NR1> is an event code corresponding to instrument status. The event code is cleared when it is reported.

Events fall into the following classes:

Command Error:	A command error results when a message cannot be completely parsed. Typically wrong command syntax.
Execution Error:	An execution error exists when a message is parsed but cannot be executed. For example, out-of-range value or conflicting settings.
Internal Error:	This error results from a malfunction or fault condition within the RTD 710A.
System Event:	This class includes all events that are not errors or warnings.
Execution Warning:	The command received is being executed, but a potential problem may exist.

Table 5-2 through 5-6 list the RTD 710A event codes.

Table 5-2
RTD 710A COMMAND ERROR EVENT CODES

Event Code	Instrument Status
Command Error Events–CER ON; SRQ Status 97 or 113	
101	Command header error.
102	Header delimiter error.
103	Command argument error.
104	Argument delimiter error.
105	Non-numeric argument.
106	Missing argument.
107	Invalid message unit delimiter.
108	Checksum error.
109	Bytecount error.
151	Symbol or number too long. Command ignored.

Table 5-3
RTD 710A EXECUTION ERROR EVENT CODES

Event Code	Instrument Status
Execution Error Events–EXR ON; SRQ Status 98 or 114	
201	Command not executable in LOCAL
206	Group Execute Trigger ignored
250	CH2 input can't be used in the high-speed sampling mode
251	No TV trigger option is installed
252	Trigger source CH2, BIslope, +HYS, and -HYS, are not usable when TV trigger coupling (Lines, FLD1, or FLD2) is selected
254	Trigger source CH2 is not usable in high-speed sampling mode
255	Selected recall memory is empty.
256	XY display mode is not allowed because two waveforms are not displayed on the monitor or because the vertical mode is set to CH1 only
258	Target cursor waveform channel is not displayed
260	Delta time (ΔdT) or 1/ΔdT measurement is not allowed when using the an external clock source
262	Requested break point to be cleared does not exist
263	More than five breakpoints requested. Last breakpoint request was ignored.
266	Illegal DATA CHAnnel request in this setting - CH2 is not allowed in the CH1 only mode
267	DATA LOCation number requested is outside valid range
268	Requested DATA STArt number is not valid
269	Requested Data Count size number is not valid
270	Requested DATA BSIze number is not valid
271	Requested Window Channel is not valid - CH2 is not allowed when using the CH1 Only mode
272	Requested Window Location number is outside the valid range
273	Requested Window Start or Stop number is outside the valid range
274	Requested Window Level value for queries is not valid
275	Requested Waveform is not valid or not available
276	Illegal test number
277	Self Test group commands are not acceptable in TESt MODe: OFF
278	Command is not acceptable in TESt MODe: ON, need to turn TESt MODe: OFF
279	Plot is not allowed because no waveform is being displayed (both set to location zero).

Table 5-4
RTD 710A INTERNAL ERROR EVENT CODES

Event Code	Instrument Status
Internal Errors - INR ON; SRQ Status 99 or 115	
302	System error or DMA controller error
350	Auto-Calibration failed - Use ERR? query to determine failure code
351	NVM save or recall error (checksum error)
352	Current Level 2 test or Power-On Self Test error - Must use ERR? query to determine failure code

Table 5-5
RTD 710A SYSTEM EVENT CODES

Event Code	Instrument Status
System	
401	Power on
403 (USER ON)	User request
450 (OPC ON)	Hold has occurred after N times Average or Envelope operation
451 (OPC ON)	Hold has occurred in Compare Trigger mode
452 (OPC ON)	Current Level 2 test or Power-On Self Test has finished
453	Plot completed.

Table 5-6
RTD 710A EXECUTION WARNING EVENT CODES

Event Code	Instrument Status
Execution Warning -	EXW ON; SRQ Status 101 or 117
550	Requested Input Range value was rounded or limited
551	Requested Input Offset value was rounded or limited
552	Selecting the High-Speed sampling mode forced the vertical mode to CH1 only
553	Requested Trigger Level value was rounded or limited
554	Requested Record Length value was rounded or limited
555	Selecting the Envelope mode forced the record length to 128K or the Record Location number was modi-fied to an allowable number or the sample rate was changed to Normal
556	Selecting the Dual vertical mode forced one of the following: • The Record Length to 128K • The Record Location number was modified to an allowable maximum number • The sample rate was decreased from 5 nS to 10 nS • The Sample mode was changed to normal
557	Selecting the Compare Trigger mode forced the Record Length to 128K or the Record Location number was modified to the maximum allowable value
558	Selecting LINES, FLD1, or FLD2 forced the Trigger Source to CH1 or the Trigger slope was set to minus
559	Requested Arm Delay value was limited
560	Requested TV Line number was limited
561	Requested Record Location number or Auto-Advance Location was limited
562	Record Length change forced Record Location number to a maximum allowable number or the Trigger Delay value to a maximum allowable value
563	Requested Trigger Delay value was truncated or limited
564	Requested Sample Interval Value was limited-This event code is also used when the sampling speed is changed from High-Speed mode (5 nS) to Normal mode (10 nS)
565	Requested Break Point address value was truncated or limited
566	Requested Average number of times was truncated or limited
567	Requested Envelope number of times was truncated or limited
568	Requested Cursor 1 Position value was limited
569	Requested Cursor 2 Position value was limited
570	Requested Horizontal Zoom value was limited
571	Requested Vertical Zoom value was limited
572	Requested Vertical Position value was limited
573	Requested Display Location number was limited
574	Requested Save or Recall location number was rounded or limited.
575	More than five Break Points were requested.
576	Target display channel for Cursor 1 was changed.
577	Target display channel for Cursor 2 was changed.
578	XINCR value for WFMPRE was limited.
579	PT.OFF value for WFMPRE was limited.
580	YZEro value for WFMPRE was limited.
581	YMUlt value for WFMPRE was limited.
582	Break Point value for WFMPRE was limited.
583	Requested REPEAT count number was limited.
584	Waveform data STArt or STOp requested number for data queries was swapped because requested start position is larger than stop position or request stop position is smaller than the start position.

Table 5-7
RTD 710A DEVICE DEPENDENT EVENT CODES

Event Code	Instrument Status
Device Dependent Event	
750 (WRI ON)	Single acquisition sequence is completed and the waveform in memory is ready to be read.
752 (OVER ON)	CH1 input overrange has occurred.
753 (OVER ON)	CH1 input underrange has occurred.
754 (OVER ON)	CH2 input overrange has occurred.
755 (OVER ON)	CH2 input underrange has occurred.

RTD 710A PROGRAM EXPLANATIONS/LISTINGS

These program routines enhance the RTD 710A's capabilities and ease its implementation into an ATE or research system. Specifically, this section covers the following areas:

- Read and decode scaling information from the RTD 710A
- Acquiring and scaling a waveform into a voltage array.
- Performing a serial POLL on the RTD 710A and query for the EVENT code.
- Graphing the waveform onto a monitor.

There are complete program listings for the 4041, IBM PCs using the Tek GURU II GPIB PACKAGE, HP 200/300 Series computers, and SPS BASIC. Each program contains routines that accomplish the areas above.

Getting the Waveform Scaling Information

The RTD 710A waveform preamble contains all the information needed to convert the binary waveform sent by the digitizer (range 0 to 1023) into actual vertical and horizontal data. The preamble information includes:

- The number of points in the waveform (with NR.PT)
- The vertical and horizontal scaling per bit (with YMULT and XINCR)
- The DC offset (with YOFF)
- The trigger location (with PT.OFF)

The binary waveform array associated with this preamble can be scaled into a voltage array where pretrigger and DC offset are included in the calculation. A typical waveform preamble follows:

```
WFMPRE WFID:'CH1_LOCATION1",ENCDG:
BINARY,NR.PT:8192,PT.FMT:Y,XINCR:5.
0E-9,PT.OFF:0,XUNIT:SEC,YZERO:0,
YOFF:512,YMULT:100.0E-3,YUNIT:
V,BYT/NR:2,BN.FMT:RP,BIT/NR:
10,BKPT:0:5.0E-9
```

Refer to Table 3-6 (Waveform Transfer Group commands) for a complete definition of each element in the waveform preamble.

Binary Waveform Acquisition

The RTD 710A sends waveform data via the GPIB in binary. The RTD 710A is a 10 bit digitizer, so each data point must be sent using two bytes, high byte first. The GPIB controller must know how many data points are going to be sent so it can dimension the proper variable to accept data. The waveform acquisition routine inputs the number of points to be acquired. The number of points can be retrieved from the WAVEFORM SCALING routine below. Refer to this section for an explanation of determining the number of points.

Waveform Scaling Routine

This section explains how the binary waveform data is scaled into a voltage array.

Three pieces of information are needed to scale a binary waveform into a voltage array:

- The ymultiplier (variable ym)
- The yoffset (variable yo)
- The yzero (variable yz)

These items are defined in the waveform preamble and have been segmented into numeric variables.

An equation determines the total offset on the signal from either the RTD 710A dc offset function or the offset on the input signal. The dc offset added by the RTD 710A is defined in the variable 'Yzero'. A relationship called the 'vertical shift" (in this example, variable 'Vshift' is used) determines the total offset of the input signal. The following equation determines the total offset on the signal so the voltage waveform is properly scaled.

Equation:

$Vshift = yoffset — (yzero * 0.01 * ymult * 1024)/(2 * ymult)$

Canceling the input range 'Ymult" out of the numerator and denominator and multiplying (.01*1024) and dividing by 2 yields the quantity 5.12.

Therefore: $Vshift = yo — (yz * 5.12)$

The volts per bit must be determined to scale the waveform. The volts per bit (Vpb) is found by multiplying the ymultiplier (ymult) times 2 (because the input range is plus and minus the ymult) and dividing by 1024 (for a 10 bit digitizer). The equation below shows how volts per bit is determined.

Equation: $Vpb = 2 * ymult/1024$

The binary waveform can now be scaled into a voltage waveform using the volts per bit 'Vpb" and the vertical shift 'Vshift." The scaling equation is shown below (assuming the use of a for-next loop to scale each element in the array):

```
For i=1 to np
Wfm(i)=(iwfm(i)-vshift) * Vpb
Next i
```

Some controllers, for example the 4041 and HP 200 series (using the MAT command) have commands that support implied array operations. The same scalingdone in the for-next loop above can be accomplished in one statement.

Read Breakpoint Information Routine

The RTD 710A allows the sample rate to be switched (called breakpoints) to make better use of the available waveform memory. For example, the user can speed up the sample rate for the rising and falling edges of a pulse and slow the sample rate down for the slower portions (that is, the top and base) of a waveform.

This routine reads the breakpoints from the waveform preamble and decodes them for timing measurements across breakpoints.

Serial Poll and Event Query

The service request (SRQ) and event query capability of the RTD 710A is a powerful way to determine its status of normal and abnormal conditions. Some examples of helpful service requests/event codes are shown in table 6-1.

Table 6-1
EXAMPLES OF SERVICE REQUEST AND EVENT CODES

SRQ Code	Event Code	Explanation
97	103	Command error; Command argument is wrong
98	261	Execution error; selected non-volatile settings location is empty
67	403	Device dependent status byte; the operator has pressed the RQS button (USER button) on the RTD 710A front panel
66	450	Hold has occured after N times average or envelope operations

This table is not a complete listing of the SRQ and event capability of the RTD 710A, but a sample of some of the information that can be obtained while developing and executing programs. The RTD 710A can:

- Tell you when a command cannot be understood or executed
- Inform that a setting has been limited because of conflicts with other settings
- Signal when a waveform is ready to transfer
- Allow an operator to signal the controller from the front panel of the RTD 710A
- Signal that the input signal is over or under range

4041/RTD 710A Program Example

Assumptions:

- The RTD 710A is at address 1
- The waveform to be acquired is from channel 1, location 1
- The waveform length is less than 16384 points—If the waveform is greater than 16K, refer to the example in the RTD 710A manual using REPEATED BLOCK TRANSFER or ABRITRARY FORMAT

*

*** 4041 program listing ****

```
100  !
110  ! RTD 710/ 4041 program to acquire waveform, scale into a voltge
120  ! array, read and decode breakpoint information, and graph
130  ! waveform on 4000 series terminal
140  !
150  ! 4041 requires: 4041R01 & 4041R02
160  ! RTD 710 is set to GPIB address 1 connected to GPIB0:
170  !
180    Init all
190    On srq then call srqhandl
200    Call initprog(rtd,glu,cha,loc)
210    Call getscale(rtd,cha,loc,np,xi,pt,yz,yo,ym,wfmpre$)
220    If np<=16384 then goto go_on
230    Print "Channel ";cha;", Location ";loc;" has ";np;" points"
240    Print "This program can read a length up to 16384"
250    Print "Ending program...."
260    Goto fini
270    Go_on:call getwfm(rtd,cha,loc,pt,np,iwfm)
280    Call scalewfm(iwfm,np,ym,yo,yz,wfm)
290    Call getbkpt(wfmpre$,nbkpt,bkpt)
300    If nbkpt<2 then goto grphwfm
310    For i=1 to nbkpt
320    Print "BREAKPOINT AT POINT:";bkpt(i,1);", SAMPLE RATE:";bkpt(i,2)
330    Next i
340    Input prompt "Press <Return> to plot waveform ":a$
350    Grphwfm:call drawwfm(glu,cha,loc,xi,pt,wfm)
360    Fini:end
400    Sub initprog(var rtd,glu,cha,loc)
410    Integer rtd,glu,cha,loc
420    Cha=1 ! Acquire channel 1 info
430    Loc=1 ! Acquire location 1 info
440    Rtd=1
450    Glu=100
460    Open #rtd:"GPIB0(PRI="&str$(rtd)&",EOM=<0>,TRA=FAST):"
470    Open #glu:"COMM0:"
480    Print #rtd:"INIT GPIB" ! Initialize RTD 710's GPIB port
490    Return
500    End
600    Sub getscale(rtd,cha,loc var np,xi,pt,yz,yo,ym,wfmpre$) local fmt$
610  !
620  ! Get scale factors for waveform in channel CHA
630  !
640    Dim wfmpre$ to 400
650    Print "READING SCALE FACTORS"
660    Print #rtd:"DAT CHA:CH"&str$(cha)&",LOC:"&str$(loc)
670    Input #rtd prompt "WFM?":wfmpre$
680    Np=arg("NR.P",wfmpre$)
```

**** 4041 program listing (Cont.) ****

```
690    Xi=arg("XIN",wfmpre$)
700    Pt=arg("PT.O",wfmpre$)
710    Yz=arg("YZE",wfmpre$)
720    Yo=arg("YOF",wfmpre$)
730    Ym=arg("YMU",wfmpre$)
740    Return
750    End
800    Function arg(header$,string$)
810  !
820  ! Return the numeric argument associated with HEADER$ in STRING$
830  !
840    Arg=valc(string$,pos(string$,":",pos(string$,header$,1))+1)
850    Return
860    End
900    Sub getwfm(rtd,cha,loc,start,count var iwfm)
910  !
920  ! Get a waveform from specified channel, segment, start & count
930  ! and return it in the integer array, IWFM
940  !
950    Print "READING WAVEFORM"
960    Print #rtd:"DAT
       CHA:CH"&str$(cha)&",LOC:"&str$(loc)&",STA:"&str$(start)&",
       COU:"&str$(count)&",BSI:"&str$(count) !select proper RTD710 channel &
       location
970    Integer iwfm(np)
980    Print #rtd:"CURVE?"
990    GETARR rtd,iwfm,16
1000   Return
1010   End
1100   Sub scalewfm(iwfm,np,ym,yo,yz var wfm) local bpb,vshift
1110  !
1120  !Sub to scale the binary waveform 'iwfm' into a voltage array
1130  !'wfm'
1140  !
1150   Print "SCALING WAVEFORM"
1160   Dim wfm(np)
1170   Vpb=2*ym/1024 ! 2 * Full Scale Range Divided by 1024 Levels
1180   Vshift=yo-yz*5.12 ! Take offset voltage into account
1190   ARRSCL iwfm,vpb,vshift,wfm
1200   Return
1210   End
1300   Sub getbkpt(wfmpre$ var nbkpt,bkpt) local-
       fini,pareit,bkpt$,a,b,strt,tmp,tmp1
1310  !
1320  !Label the breakpoints on a waveform plot
1330  !
1340   Integer a,b,strt,nbkpt
1350   A=pos(wfmpre$,"BKPT:",1)
1360   Bkpt$=seg$(wfmpre$,a,len(wfmpre$)-a+1)
1370   Nbkpt=0
1380   Delete var bkpt
1390   Dim bkpt(5,2)
1400   For i=1 to 5
1410   Gosub parseit
1420   If i=6 then exit to fini
1430   Bkpt(i,1)=tmp
```

**** 4041 program listing (Cont.) ****

```
1440   Bkpt(i,2)=tmp1
1450   Nbkpt=nbkpt+1
1460   Next i
1470   Goto fini
1480   Parseit:a=pos(bkpt$,"BKPT:",strt)+5
1490   If a<6 then i=6 else goto readval
1500   Goto subdone
1510   Readval:tmp=valc(bkpt$,a)
1520   B=pos(bkpt$,":",a)
1530   Tmp1=valc(bkpt$,b+1)
1540   Strt=b
1550   Subdone:return
1560   Fini:dim bkpt(nbkpt,2)
1570   Return
1580   End
1600   Sub drawwfm(glu,cha,seg,xi,pt,wfm)
1610  !
1620  !Draw a waveform
1630  !
1640   Ec$=chr$(27)
1650   Print ec$&"%!0"&ec$&"NF3" !set 4105 to tek codes, use flagging
1660   Print ec$&chr$(12)&ec$&"LZ" !ERASE GRAPHICS AND DIALOG AREA
1670   GINIT glu,4105,1
1680   COLOR 1
1690   OFFSET pt*xi,xi
1700   YPLOT wfm
1710   TITLE "CHANNEL "&str$(cha)&", LOCATION "&str$(seg)&", NUM PTS:
       "&str$(np)
1720   YTITLE "Volts"
1730   XTITLE "Seconds"
1740   Input prompt "Press <Return> to continue ":a$
1750   Print ec$&"LLA>" !SET DIALOG LINES TO 30
1760   Return
1770   End
1800   Sub srqhandl
1810  !
1820  !Poll the RTD710A in response to SRQ
1830  !
1840   Integer status,add
1850   Poll status,add
1860   Input #add prompt "EVE?":event
1870  !Display the status & event codes
1880   Print "SRQ Asserted, Status = ";status;
1890   Print "Event Code = ";event
1900   Print "path= "&ask$("path",all)
1910   Resume
1920   End
```

IBM PC/RTD 710A Program Example

This IBM BASICA example program:

- Reads and decodes the scale factors
- Acquires a waveform (up to 8K long)
- Graphs the acquired waveform on the monitor
- Scales the acquired waveform into a voltage array

(Note: This routine is skipped using a 'GOTO" statement in this

example because it is not necessary to run the program. It is

included for individual needs of the user.)

- Reads and decodes breakpoint information

The IBM main program listing can be used if the following conditions are satisfied:

The user has Tek GURU II Ver 2.0 and the program 'DIGPULSE.BAS" will run properly.

- The initialization and the subroutines from 'DIGPULSE.BAS" are
- used by the main program provided. The procedure to put the main
- program together with the 'DIGPULSE" routines is as follows:

1. Load BASICA.

2. Enter the main program listing, lines 100 to 2820.

3. Type: Save 'RTD.MAIN",A ! Save as an ASCII file.

4. Type: Load 'digpulse.bas."

5. Type: Delete 1-9850 ! This removes everything but the 'DIGPULSE" utility subroutines.

6. Type: Save 'RTD.SUBS",A ! Save as an ASCII file.

7. Type: Load 'digpulse.bas."

8. Type: Delete 40-13220 ! This removes everything but the initialization code for the National GPIB card.

9. Type: Merge 'rtd.main" ! This will append the main program to the initialization code.

10. Type: Merge 'rtd.subs" ! This loads the DIGPULSE subroutines.

11. Type: Save 'rtd710.bas" ! Save the whole program away. The program should now be able to acquire and graph a waveform from CH1, LOCATION 1.

**** IBM Main Program Listing ****

```
Lines 1 thru 10: Insert first ten lines from your Digpulse program
here

11  '   for compiled version, remark lines 1-9 and
12  '   remove remark (') from the following line
13  '   COMMON IBSTA%,IBERR%,IBCNT%
14      COMPILE%=0 ' set to 0 for interpreted BASIC or -1 for compiled BASIC
35      '------------------------------------------------------------------
100     HELP%=0 ' ask for NO help on all routines
110     SCREEN 2    'hi resolution screen for both IBM and Hercules graphics
        cards
120     CLS:PRINT "RTD710A WAVEFORM ACQUIRE AND GRAPH"
130     PRINT:PRINT
140     CHA%=1 'SELECT CHANNEL 1 OF RTD710, CHANGE AS NEEDED
150     LOCAT%=1 'SELECT LOCATION 1 OF THE RTD710, CHANGE AS NEEDED
160     CHA$=MID$(STR$(CHA%),2,1)
170     LOCAT$=MID$(STR$(LOCAT%),2,LEN(STR$(LOCAT%))-1)
180  '
190  '  IT'S ASSUMED HERE THAT THE RTD710 IS SET TO ADDRESS 1.  THIS CODE
200  '  SELECTS THE RTD710 AS THE ACTIVE DEVICE BY SELECTING 'TEKDEV1'.
210  '
220     DEVICE$="TEKDEV1"
```

** IBM Main Program Listing (Cont.) **

```
230     FUNC%=1:GOSUB 10000 'SELECT RTD710 AT ADDRESS 1
240     GOSUB 1410 'SERIAL POLL
250   ' ---------------------------------------------------------------------
260   ' START OF GET SCALING INFORMATION SUBROUTINE
270   '
275     PRINT "PRESS <ENTER> TO ACQUIRE WAVEFORM FROM CHANNEL "+CHA$+", LOCA-
        TION "+LOCAT$;:INPUT A$
280     PRINT "READING AND DECODING WAVEFORM PREAMBLE"
290     WRT$="INIT GPIB;DAT CHA:CH"+CHA$+",LOC:"+LOCAT$+";WFMPRE?"
300     FUNC%=3:GOSUB 10000 'SEND MESSAGE
310     FUNC%=4:GOSUB 10000 'READ WAVEFORM PREAMBLE INTO RD$
320     WFMPRE$=RD$
330   '
340     SRCH$="NR.PT":GOSUB 1320:NP=TEMP 'SET NUMPTS INTO VAR NP
350     IF NP<=8192 GOTO 400
360     PRINT "The waveform length to acquire is"+STR$(NP)+" points."
370     PRINT "This program can only handle a 8192 point waveform"
380     PRINT "Ending program....."
390     GOTO 1310
400     SRCH$="XINCR":GOSUB 1320:XI=TEMP 'SET XINCREMENT TO VAR XI
410     SRCH$="PT.OFF":GOSUB 1320:PT=TEMP 'SET TRIG DELAY TO VAR PT
420     SRCH$="YZERO":GOSUB 1320:YZ=TEMP 'SET YZERO TO VAR YZ
430     SRCH$="YOFF":GOSUB 1320:YO=TEMP 'SET YOFFSET TO VAR YO
440     SRCH$="YMULT":GOSUB 1320:YM=TEMP 'SET YMULTIPLIER TO VAR YM
460     NP$=MID$(STR$(NP),2,LEN(STR$(NP))-1)
470   ' 480 'GOSUB 1400 'SERIAL POLL
490   '
500   ' END OF GET SCALING INFORMATION SUBROUTINE
510   ' ---------------------------------------------------------------------
520   ' START OF GET BINARY WAVEFORM SUBROUTINE
530   '
540     PRINT "READING BINARY WAVEFORM FROM RTD710"
550     WRT$="DAT CHA:CH"+CHA$
560     WRT$=WRT$+",LOC:"+LOCAT$+",START:"+STR$(PT)
570     WRT$=WRT$+",COUNT:"+NP$+",BSIZE:"+NP$+";CURVE?"
580   '
590   ' INITIALIZE ALL VARIABLES NEEDED TO ACQUIRE AND GRAPH WAVEFORM
600   '
610     LE%=NP
620     VS=128
630     VZ=511
640     EOI%=1
650     PRI%=1
660     SEC%=0
670     CONV%=0
680     MODE%=4
690     CP%(0)=2:CP%(1)=LE%-2
700     VU=YM/4
710     NA$="RTD 710 WFM, CHA "+CHA$+", LOC "+LOCAT$
720     OFS=VU*(VZ/VS+YZ)
730     DSI=XI
740     CUR1%=0
750     CUR2%=NP-1
760     FUNC%=3:GOSUB 10000 'SELECT CHANNEL, LOCATION, ETC & SEND 'CURVE?'
        QUERY
770     DIM WFM%(LE%)
```

**** IBM Main Program Listing (Cont.) ****

```
780    FUNC%=11:GOSUB 10000 'GET BINARY WAVEFORM
790    GOTO 950 'SKIP SCALING ROUTINE UNLESS NEEDED
800  '
810  ' END OF GET BINARY WAVEFORM ROUTINE
820  ' -----------------------------------------------------------------
830  ' START OF SCALE BINARY WAVEFORM INTO VOLTAGE WAVEFORM SUBROUTINE
840  ' RAW BINARY WAVEFORM DATA IS RETURNED IN ARRAY 'RDD%'.
850  '
860    PRINT "SCALING INTO VOLTAGE WAVEFORM"
870    VSHIFT=YO-(YZ*5.12)
880    VPB=(YM*2)/1024
890    FOR I=1 TO NP
900    WFM(I)=(WFM(I)-VSHIFT)*VPB
910    NEXT I
920  '
930  ' END OF SCALE BINARY WAVEFORM ROUTINE
940  ' -----------------------------------------------------------------
950  ' GRAPH WAVEFORM
960    KEYF=4
970    FUNC%=14:GOSUB 10000 'GRAPH WAVEFORM
980  '
990  ' END OF GRAPH WAVEFORM
1000  '-----------------------------------------------------------------
1010  'READ AND DECODE BREAKPOINT INFORMATION ROUTINE
1020  '
1030   POSIT%=INSTR(WFMPRE$,"BKPT:")
1040   RD$=MID$(WFMPRE$,POSIT%,LEN(WFMPRE$)-POSIT%+1)
1050   NBKPT%=0 'SET NUMBER OF BREAKPOINTS TO ZERO
1060   DIM BKPT%(5,2)
1070   BKPT(1,1)=0
1080   FOR I=1 TO 5
1090   FUNC%=18
1100   GOSUB 10000 'PARSE OUT LOCATION OF BREAKPOINT
1110   IF I=1 THEN GOTO 1150
1120   BKPT(I,1)=RESULT
1130   FUNC%=18
1140   GOSUB 10000 'PARSE OUT SAMPLE RATE
1150   BKPT(I,2)=RESULT
1160   NBKPT%=NBKPT%+1
1170   POSIT%=INSTR(RD$,"BKPT:")
1180   IF POSIT%<1 THEN I=6
1190   IF I=6 THEN GOTO 1210
1200   RD$=MID$(RD$,POSIT%)
1210   NEXT I
1220  '
1230  'This next section displays the breakpoint locations and sample rates
1240  'It is not necessary when using the breakpoint information
1250  '
1260   IF NBKPT%<2 THEN GOTO 1310
1270   PRINT "NUMBER OF BREAKPOINTS:"+STR$(NBKPT%)
1280   FOR I=1 TO NBKPT%
1290   PRINT "BREAKPOINT LOCATION:"+STR$(BKPT(I,1))+", SAMPLE
       RATE:"+STR$(BKPT(I,2))
1300   NEXT I
1310   END
```

** IBM Main Program Listing (Cont.) **

```
1320  '--------------------------------------------------------------------
1330  'PARSE DESIRED INFORMATION FROM WFMPRE$ AND PUT NUMBER IN
1340  'VARIABLE 'TEMP'.
1350  '
1360   POSIT%=INSTR(WFMPRE$,SRCH$)+LEN(SRCH$)+1
1370   TMP$=MID$(WFMPRE$,POSIT%,15)
1380   TEMP=VAL(TMP$)
1390   RETURN
1400  '--------------------------------------------------------------------
1410  'START OF SERIAL POLL SUBROUTINE
1420   FUNC%=2:GOSUB 10000
1430   WRT$="EVENT?"
1440   FUNC%=3:GOSUB 10000
1450   FUNC%=4:GOSUB 10000
1460   EVT%=VAL(RD$)
1470   PRINT "SERIAL POLL, STATUS:"+STR$(SPR%)+",EVENT:"+STR$(EVT%)
1480   IF SPR%>16 THEN 1420
1490   RETURN
1500  'END OF SERIAL POLL SUBROUTINE
1510  '--------------------------------------------------------------------
```

**** HP 200/300 Series/RTD 710A Example Program Listing ****

```
101  !
103  ! Waveform acquire, scale, read
104  ! breakpoints and decode, and graph
105  ! scaled waveform program
106  !
107  ! Written on 9826 w/ Ver 4.0 BASIC
108  ! using GPIB port 8
109  !
111  ! Requires loading the following binary
112  ! files before executing:
113  ! LOAD BIN <FILENAME> where <filename> =
114  ! o-<IO>Opt 4
115  ! o-<GRAPH>Opt 2
116  ! o-<MAT>Opt 7
117  !
119    GOSUB Initialise
124    ON INTR 8 CALL Srqhandl
125    ENABLE INTR 8;2
130    CALL Getscale(@Rtd,Cha,Loc,Np,Xi,Pt,Yz,Yo,Ym,Wfmpre$)
131    IF Np<=8192 THEN 137
132    PRINT "The number of points:";Np;"is too long"
133    PRINT "This program can only handle a 8192 wfm"
134    PRINT "Ending program"
135    GOTO Fini
137    ALLOCATE INTEGER Iwfm(1:Np)
138    ALLOCATE Wfm(1:Np)
140    CALL Getwfm(@Rtd,Cha,Loc,Pt,Np,Iwfm(*))
150    CALL Scalewfm(Np,Ym,Yo,Yz,Wfm(*),Iwfm(*))
160    CALL Getbkpt(@Rtd,Nbkpt,Bkpt(*))
161    PRINT "Num breakpoints:";Nbkpt
163    FOR I=1 TO Nbkpt
164    PRINT "Breakpoint loc:";Bkpt(I,1);", Sample rate:";Bkpt(I,2)
165    NEXT I
170    CALL Graphwfm(Wfm(*),Xi,Np)
171    GOTO Fini
180    Initialise:!
181    OPTION BASE 1
190    Addr=1 !RTD ADDRESS
202    DIM Wfmpre$[600],Wrt$[200]
203    Cha=1 !SELECT CHANNEL (1 or 2)
205    Loc=1 !SELECT LOCATION (1 to 256)
206    DIM Bkpt(5,2)
210    ABORT 8
220    REMOTE 800+Addr
230    CLEAR 800+Addr
240    ASSIGN @Rtd TO 800+Addr;EOL CHR$(13) END
250    RETURN
260    Fini:END
1000   SUB Getscale(@Rtd,Cha,Loc,Np,Xi,Pt,Yz,Yo,Ym,Wfmpre$)
1010   REM
1020   REM GET SCALE FACTORS FOR WAVEFORM IN CHANNEL CHA, LOCATION LOC
1030   REM
1040   Wrt$="INIT GPIB;DAT CHA:CH"&VAL$(Cha)&",LOC:"&VAL$(Loc)
1041   PRINT "READING SCALE FACTORS"
1043   OUTPUT @Rtd;Wrt$&";WFM?"
```

**** HP 200/300 Series/RTD 710A Example Program Listing (Cont) ****

```
1050    ENTER @Rtd;Wfmpre$
1060    Np=FNArg("NR.PT:",Wfmpre$) !  NUMBER OF POINTS IN WAVEFORM
1070    Xi=FNArg("XINCR:",Wfmpre$) !  TIME PER POINT
1080    Pt=FNArg("PT.OFF:",Wfmpre$) !  AMOUNT OF PRETRIGGER
1090    Yz=FNArg("YZERO:",Wfmpre$) !  GROUND REFERENCE
1100    Yo=FNArg("YOFF:",Wfmpre$) !  DC OFFSET
1110    Ym=FNArg("YMULT:",Wfmpre$) !  VOLTS PER BIT
1130    SUBEND
1220    SUB Getwfm(@Rtd,Cha,Loc,Start,Np,INTEGER Iwfm(*))
1230    REM
1240    REM GET A WAVEFORM FROM SPECIFIED CHANNEL, LOCATION, START & COUNT
1250    REM AND RETURN IT IN THE INTEGER ARRAY, IWFM
1251    DIM Wrt$[200]
1260    Wrt$="DAT CHA:CH"&VAL$(Cha)&",LOC:"&VAL$(Loc)&",STA:"&VAL$(Start)
1270    Wrt$=Wrt$&",COU:"&VAL$(Np)&",BSI:"&VAL$(Np)
1271    PRINT "READING";Np;"POINT BINARY WAVEFORM"
1280    OUTPUT @Rtd;Wrt$&";CURVE?"
1290    ENTER @Rtd USING "#,9A";Cur$
1291    ENTER @Rtd USING "%,W";Iwfm(*)
1292    ENTER @Rtd USING "B";Chk !CHECKSUM
1301    SEND 8;UNL UNT
1302    CLEAR 8
1310    SUBEND
1320    SUB Scalewfm(Np,Ym,Yo,Yz,Wfm(*),INTEGER Iwfm(*))
1330    REM
1340    REM SCALE BINARY WAVEFORM STORED IN IWFM INTO A VOLTAGE WAVEFORM
        STORED IN
1350    REM WFM
1370    Vpb=2*Ym/1024 !  2*FULLSCALE RANGE DIVIDED BY 1024 EQUALS THE VOLTS
        PER BIT
1380    REM
1390    REM THE NEXT SECTION DETERMINES THE DC OFFSET
1400    REM 'VSHIFT' USING YOFFSET,YZERO (ALWAYS 512)
1410    REM , AND YMULT.  VSHIFT EQUALS:
1420    REM VSHIFT=YOFFSET-(YZERO*.01*YMULT*1024)/(2*YMULT)
1430    REM VARIABLE  YO     YZ       YM          YM
1440    REM
1450    REM CANCELING THE YMULT OUT OF THE NUMERATOR
1460    REM AND DENOMINATOR AND MULTIPLYING .01*1024
1470    REM AND DIVIDING BY 2 YIELDS 5.12
1480    REM
1490    REM THEREFORE: VSHIFT=YO-(YZ*512)
1500    REM
1501    PRINT "SCALING INTO VOLTAGE WAVEFORM"
1510    Vshift=Yo-(Yz*5.12)
1520    Vpb=(Ym*2)/1024
1541    MAT Wfm= Iwfm-(Vshift)
1542    MAT Wfm= Wfm*(Vpb)
1580    SUBEND
1920    SUB Graphwfm(Wfm(*),Xi,Np)
1930    GINIT
1940    GCLEAR
1950    GRAPHICS ON
1951    Amax=MAX(Wfm(*))
1952    Amin=MIN(Wfm(*))
1953    Voffset=(Amax-Amin)/20
```

** HP 200/300 Series/RTD 710A Example Program Listing (Cont) **

```
1954   Hoffset=(Np)/20
1955   OUTPUT 2;""; ! Clears dialog area
1956  !Message between quotes is as follows:
1957  !"<CTRL-BACKSPACE>K"
1959   MOVE 35,90
1960   LABEL "RTD 710A WAVEFORM"
1961   VIEWPORT 20,110,20,80
1962   WINDOW 0,(Np)+Hoffset,Amin-Voffset,Amax+Voffset
1963   MOVE 0,Amin-Voffset
1964   DRAW Np+Hoffset,Amin-Voffset
1965   DRAW Np+Hoffset,Amax+Voffset
1966   DRAW 0,Amax+Voffset
1967   DRAW 0,Amin-Voffset
1968   MOVE 0+Hoffset/2,Wfm(1)+Voffset/2
1970   FOR I=2 TO Np
1980   DRAW I+Hoffset/2,Wfm(I)+Voffset/2
1990   NEXT I
2000   DISP "<ENTER> TO CONTINUE:";
2010   INPUT A$
2020   GCLEAR
2510   SUBEND
2520   DEF FNArg(Header$,String$)
2530   REM
2540   REM RETURN THE NUMERIC ARGUMENT ASSOCIATED WITH HEADER$ IN STRING$
2550   REM
2560   Ps=POS(String$,Header$)+LEN(Header$)
2570   Tp$=String$[Ps;15]
2580   RETURN VAL(Tp$)
2590   FNEND
2970   SUB Getbkpt(@Rtd,Nbkpt,Bkpt(*))
2980   REM
2990   REM READ AND DECODE BREAKPOINT INFORMATION
3000   REM RETURNING THE NUMBER OF BREAKPOINTS IN
3010   REM VARIABLE 'NBKPT' AND THE BREAKPOINT
3020   REM INFORMATION IN VARIABLE 'BKPT'
3030   REM
3060   Nbkpt=0
3070   DIM Bkpt$[100],Tmp$[200]
3080   OUTPUT @Rtd;"BRE UNI:POINT;BRE? SET",END
3090   ENTER @Rtd;Bkpt$
3100   FOR I=1 TO 5
3110   GOSUB Parseit
3113   IF I=6 THEN GOTO 3120
3114   Bkpt(I,1)=Tmp
3115   Bkpt(I,2)=Tmp1
3117   Nbkpt=Nbkpt+1
3120   NEXT I
3121   GOTO 3290
3130   Parseit: A=POS(Bkpt$,"SET:")
3140   IF A<1 THEN Nomore
3152   Tmp$=Bkpt$[A+4;LEN(Bkpt$)]
3170   Tmp=VAL(Tmp$)
3180   A=POS(Tmp$,":")
3185   Tmp$=Tmp$[A+1;LEN(Tmp$)]
3200   Tmp1=VAL(Tmp$)
3240   Bkpt$=Tmp$
```

** HP 200/300 Series/RTD 710A Example Program Listing (Cont) **

```
3250    GOTO Done
3260    Nomore: I=6
3270    Done: RETURN
3290    SUBEND
3300    SUB Srqhandl
3310    Sb=SPOLL(801)
3313    IF Sb<16 THEN 3351
3320    BEEP
3330    OUTPUT 801;"EVENT?"
3340    ENTER 801;Event$
3350    PRINT "SRQ...Status=";Sb;", ";Event$
3351    ENABLE INTR 8;2
3360    SUBEND
```

** SPS BASIC/RTD 710A Example Program Listing **

```
10      REM ---RTD 710A / SPS BASIC VER 3.0 WAVEFORM ACQUIRE, SCALE,
20      REM     AND GRAPH.  THIS ROUTINE ALSO READS AND DECODES
30      REM     BREAKPOINT INFORMATION.
40      REM
50      PA=1\REM -- RTD 710 ADDRESS SET TO 1
60      IB=0\REM -- INTERFACE BUS 0
70      CH=1\REM -- CHANNEL 1
80      LO=1\REM -- LOCATION 1
90      LOAD "GPI"
100     SIFTO @IB,2000
110     SIFCOM @IB,"DCL"
120     WHEN @IB HAS "SRQ" GOSUB 1720
140     GOSUB 320\REM -- GET SCALING INFORMATION
150     IF NP<=8192 THEN 200
160     PRINT "THE NUMBER OF POINTS:";NP;" IS TOO LONG"
170     PRINT "THIS PROGRAM CAN ONLY HANDLE A 8192 WAVEFORM"
180     PRINT "ENDING PROGRAM"
190     GOTO 310
200     GOSUB 670\REM -- ACQUIRE WAVEFORM
210     GOSUB 1080\REM -- SCALE WAVEFORM INTO VOLTAGE ARRAY
220     GOSUB 1290\REM -- READ AND DECODE BREAKPOINT INFO
230     WAVEFORM GW IS W1,XI,HA$,VA$
240     HA$="SEC"\VA$="VOLTS"
250     EC$=CHR(27)\REM ESCAPE CHARACTER
260     PRINT EC$&"LI100"&EC$&"LZ"&EC$&"KA0"\REM SET UP TERMINAL FOR GRAPH
270     GRAPH GW\REM -- GRAPH ACQUIRED WAVEFORM
280     SMOVE 200,750
290     PRINT "RTD 710A WAVEFORM, CHA:";CH;", LOCATION:";LO;", NUMPTS:";NP
300     PRINT EC$&"KA1"\REM DAENABLE YES
310     END
320     REM --- SUBROUTINE TO ACQUIRE RTD 710 SCALE FACTOR DATA
330     REM
340     REM --- VARIABLE DEFINITIONS:
350     REM
360     REM IB - INTERFACE BUS NUMBER (0,1,2, OR 3)
370     REM PA - PRIMARY ADDRESS OF THE RTD 710 (0 - 30)
380     REM CH - CHANNEL TO ACQUIRE WAVEFORM DATA FROM (1 OR 2)
390     REM LO - SELECT LOCATION OF WAVEFORM CHANNEL (1 TO 256)
400     REM P,P1 - TEMPORARY POINTERS AND VALUE HOLDER
410     REM A$ - TEMPORARY STRING FOR OUTPUT TO DIGITIZER
420     REM WP$ - STRING CONTAINING THE WAVEFORM PREAMBLE
430     REM S$ - CONTAINS STRING TO BE LOCATED IN B$
440     REM
450     REM --- SUBROUTINE STARTS HERE
460     REM
470     A$="INIT BPIB;DAT CHA:CH"&STR(CH)&",LOC:"&STR(LO)
480     PRINT "READING WAVEFORM PREAMBLE FOR CH";CH;", LOCATION";LO
490     PUT A$ INTO @IB,PA+32
500     REM --- GET WAVEFORM SCALING INFORMATION
510     PUT "WFM?" INTO @IB,PA+32
520     GET WP$ FROM @IB,PA+64
530     S$="NR.PT"\GOSUB 620\NP=P\REM NP = NUMBER OF POINTS
540     S$="XINCR"\GOSUB 620\XI=P\REM XI = TIME PER POINT
550     S$="PT.OFF"\GOSUB 620\PT=P\REM PT = AMOUNT OF PRETRIGGER
560     S$="YZERO"\GOSUB 620\YZ=P\REM YZ = GROUND REFERENCE
```

** SPS BASIC/RTD 710A Example Program Listing **

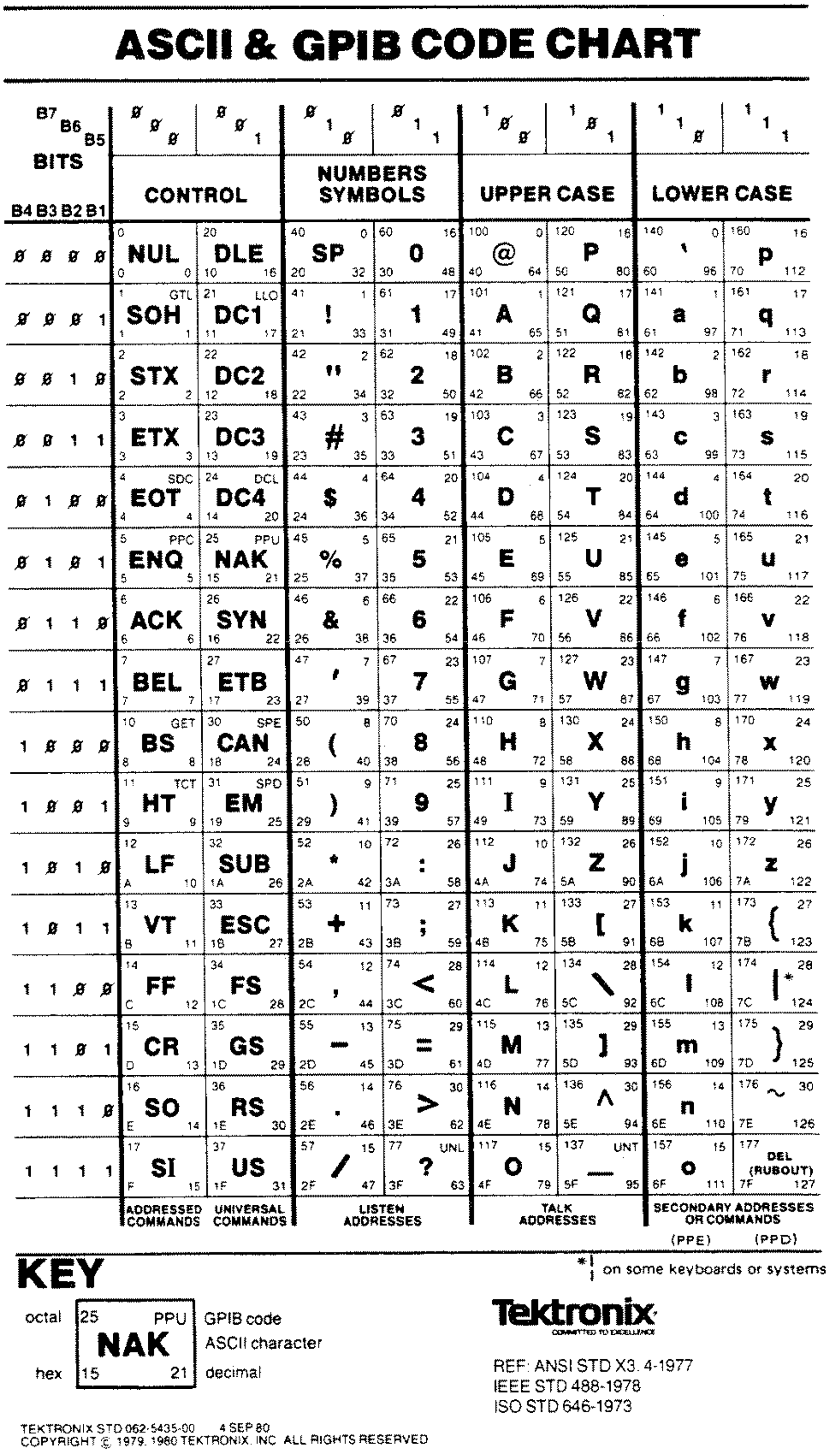
```
570    S$="YOFF"\GOSUB 620\YO=P\REM YO = Y OR VERTICAL OFFSET (ALWAYS 512)
580    S$="YMULT"\GOSUB 620\YM=P\REM YM = Y OR VERTICAL MULTIPLIER
590    DELETE A$,S$
600    RETURN
610    REM --- SUBROUTINE TO EXTRACT NUMERIC VALUE FROM PREAMBLE STRING
620    P=POS(WP$,S$,1)\REM FIND KEY STRING
630    P=POS(WP$,":",P)+1\REM  FIND BEGINNING OF VALUE TO CONVERT
640    P1=POS(WP$,",",P)-1\REM  FIND END OF VALUE TO CONVERT
650    P=VAL(SEG(WP$,P,P1))\REM  CONVERT VALUE
660    RETURN
670    REM --- SUBROUTINE TO ACQUINRE RTD 710 BINARY WAVEFORM DATA
680    REM
690    REM NOTE: THIS ROUTINE WILL READ AN 8K WORD WAVEFORM OR LESS
700    REM
710    REM --- VARIABLE DEFINITIONS:
720    REM
730    REM IB - INTERFACE BUS NUMBER (0,1,2, OR 3)
740    REM PA - PRIMARY ADDRESS OF THE RTD 710 (0 TO 30)
750    REM CH - CHANNEL TO ACQUIRE WAVEFORM DATA FROM (1 OR 2)
760    REM LO - SELECT LOCATION OF WAVEFORM CHANNEL  (1 TO 256)
770    REM ST - STARTING LOCATION OF THE WAVEFORM DATA TO SEND
780    REM  RANGE:  -65535 TO 131069
790    REM NP - NUMBER OF DATA POINTS TO BE SENT AND RECEIVED
800    REM  MAXIMUM OF 8192 ELEMENTS (8K WORD WAVEFORM)
810    REM WF - WAVEFORM ARRAY TO RECEIVE WAVEFORM DATA
820    REM A$ - TEMPORARY STRING TO HOLD SETUP COMMANDS
830    REM TP - TEMPORARY ARRAY TO HOLD "CURVE" PART OF WAVEFORM
840    REM  DATA STREAM
850    REM BC - BYTE COUNT
860    REM TM - TERMINATION CHARACTER
870    REM
880    REM --- SUBROUTINE BEGINS HERE
890    REM --- BUILD SETUP STRING
900    A$="DAT CHA:CH"&STR(CH)&",LOC:"&STR(LO)&",STA:"&STR(PT) 910
       A$=A$&",COU:"&STR(NP)&",BSIZ:"&STR(NP)
920    PRINT "READING";NP;" POINT WAVEFORM"
930    REM --- SEND SETUP STRING
940    PUT A$ INTO @IB,PA+32
950    PUT "CURVE?" INTO @IB,PA+32
960    REM --- GET DATA
970    DELETE TP\INTEGER TP(6)
980    IFDTM @IB,"UNP"
990    GET TP FROM @IB,PA+64\REM GET "CURVE" FROM DIGITIZER
1000   IFDTM @IB,"PAK","HBF"
1010   DELETE WF,A$,TP
1020   GET BC FROM @IB,PA+64
1030   INTEGER WF((BC-1)/2-1)\REM SIZE ARRAY TO RECEIVE WAVEFORM DATA
1040   GET WF FROM @IB,PA+64
1050   GET TM FROM @IB,PA+64
1060   IFDTM @IB,"UNP"
1070   RETURN
1080   REM --- SUBROUTINE TO SCALE RTD 710 WAVEFORM DATA INTO A
1090   REM VOLTAGE ARRAY
1100   REM
1110   REM --- VARIABLE DEFINITIONS:
1120   REM
```

** SPS BASIC/RTD 710A Example Program Listing **

```
1130   REM NP - NUMBER OF POINTS IN WAVEFORM
1140   REM YZ - GROUND REFERENCE OR DC OFFSET
1150   REM YO - VERTICAL OFFSET (ALWAYS 512)
1160   REM YM - VERTICAL MULTIPLIER
1170   REM
1180   REM ---   SUBROUTINE STARTS HERE
1190   DELETE W1
1200   PRINT "SCALING WAVEFORM"
1210   DIM W1(NP-1)\REM MAKE A FLOATING POINT ARRAY TO HOLD SCALED DATA
1220   REM  CALCULATE THE SCALE OF VOLTS PER VERTICAL BIT
1230   VB=2*YM/1024
1240   REM  CALCULATE THE VERTICAL SHIFT TO BE SUBTRACTED
1250   VS=YO-(YZ*1024/2)
1260   REM SUBTRACT OFFSET FROM ARRAY AND MULTIPLY BY VERTICAL SCALE
1270   W1=(WF-VS)*VB
1280   RETURN
1290   REM --- SUBROUTINE TO READ AND DECODE BREAKPOINTS
1300   REM
1310   REM --- VARIABLE DEFINITIONS:
1320   REM
1330   REM  IB - INTERFACE BUS NUMBER (0,1,2, OR 30
1340   REM  PA - PRIMARY ADDRESS OF THE RTD 710 (0 - 30)
1350   REM  P,P1 - TEMPORARY POINTERS FOR SEGMENTING B$
1360   REM  B$ - TEMPORARY STRING FOR INPUT FROM DIGITIZER
1370   REM  NB - NUMBER OF BREAKPOINTS
1380   REM  BP - BREAKPOINT DATA ARRAY (5 X 2 ARRAY)
1390   REM  (X,1) = CONTAINS BREAKPOINT LOCATION
1400   REM  (X,2) = CONTAINS BREAKPOINT SAMPLE RATE
1410   REM X RANGE: 1 TO 5
1420   REM
1430   DELETE B$,BP,NB
1440   DIM BP(5,2)
1450   BP=0
1460   NB=0
1470   P1=POS(WP$,"BKPT:",1)
1480   B$=SEG(WP$,P1,LEN(WP$))
1490   P=1\REM  STARTING LOCATION OF SEARCH
1500   FOR I=1 TO 5\REM  I = BREAKPOINT NUMBER
1510   GOSUB 1620
1520   IF P1=LEN(B$) THEN I=6\REM  FORCE LOOP TO END
1530   NB=NB+1
1540   NEXT I
1550   PRINT CHR(13)
1560   FOR I=1 TO NB
1570   PRINT "BREAKPOINT AT POINT: ";BP(I,1);", SAMPLE RATE: ";BP(I,2)
1580   NEXT I
1590   WAIT 2000
1600   RETURN
1610   REM --- SUBROUTINE TO FIND AND EXTRACT BREAKPOINT INFO
1620   P=POS(B$,"BKPT",P)+5\REM  FIND "BKPT" AND LOCATION INFO
1630   IF P=0 THEN 1700\REM  NO MORE DATA
1640   P1=POS(B$,":",P)-1\REM  GET END OF BREAKPOINT LOCATION
1650   BP(I,1)=VAL(SEG(B$,P,P1))
1660   P=P1+2\REM  START POSITION OF BREAKPOINT DATA VALUE
1670   P1=POS(B$,",",P)-1\REM END OF BREAKPOINT DATA VALUE
1680   IF P1=-1 THEN P1=LEN(B$)
```

**** SPS BASIC/RTD 710A Example Program Listing ****

```
1690    BP(I,2)=VAL(SEG(B$,P,P1))
1700    RETURN
1710    REM POLL ROUTINE
1720    POLL @IB,SB,AD,SA;PA+64
1730    AD=AD-64
1740    PUT "EVENT?" INTO @IB,AD+32
1750    GET EV$ FROM @IB,AD+64
1760    PRINT "SRQ AT ADDRESS "&STR(AD)&", STATUS= "&STR(SB)&", "&EV$
1770    RETURN
```



ASCII & GPIB CODE CHART

B7 B6 B5 BITS	Ø Ø Ø	Ø Ø 1	Ø 1 Ø	Ø 1 1	1 Ø Ø	1 Ø 1	1 1 Ø	1 1 1
B4 B3 B2 B1	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
Ø Ø Ø Ø	0 NUL 0 0	20 DLE 10 16	40 0 SP 20 32	60 16 0 30 48	100 0 @ 40 64	120 16 P 50 80	140 0 ` 60 96	160 16 p 70 112
Ø Ø Ø 1	1 GTL SOH 1 1	21 LLO DC1 11 17	41 1 ! 21 33	61 17 1 31 49	101 1 A 41 65	121 17 Q 51 81	141 1 a 61 97	161 17 q 71 113
Ø Ø 1 Ø	2 STX 2 2	22 DC2 12 18	42 2 " 22 34	62 18 2 32 50	102 2 B 42 66	122 18 R 52 82	142 2 b 62 98	162 18 r 72 114
Ø Ø 1 1	3 ETX 3 3	23 DC3 13 19	43 3 # 23 35	63 19 3 33 51	103 3 C 43 67	123 19 S 53 83	143 3 c 63 99	163 19 s 73 115
Ø 1 Ø Ø	4 SDC EOT 4 4	24 DCL DC4 14 20	44 4 $ 24 36	64 20 4 34 52	104 4 D 44 68	124 20 T 54 84	144 4 d 64 100	164 20 t 74 116
Ø 1 Ø 1	5 PPC ENQ 5 5	25 PPU NAK 15 21	45 5 % 25 37	65 21 5 35 53	105 5 E 45 69	125 21 U 55 85	145 5 e 65 101	165 21 u 75 117
Ø 1 1 Ø	6 ACK 6 6	26 SYN 16 22	46 6 & 26 38	66 22 6 36 54	106 6 F 46 70	126 22 V 56 86	146 6 f 66 102	166 22 v 76 118
Ø 1 1 1	7 BEL 7 7	27 ETB 17 23	47 7 ' 27 39	67 23 7 37 55	107 7 G 47 71	127 23 W 57 87	147 7 g 67 103	167 23 w 77 119
1 Ø Ø Ø	10 GET BS 8 8	30 SPE CAN 18 24	50 8 (28 40	70 24 8 38 56	110 8 H 48 72	130 24 X 58 88	150 8 h 68 104	170 24 x 78 120
1 Ø Ø 1	11 TCT HT 9 9	31 SPD EM 19 25	51 9) 29 41	71 25 9 39 57	111 9 I 49 73	131 25 Y 59 89	151 9 i 69 105	171 25 y 79 121
1 Ø 1 Ø	12 LF A 10	32 SUB 1A 26	52 10 * 2A 42	72 26 : 3A 58	112 10 J 4A 74	132 26 Z 5A 90	152 10 j 6A 106	172 26 z 7A 122
1 Ø 1 1	13 VT B 11	33 ESC 1B 27	53 11 + 2B 43	73 27 ; 3B 59	113 11 K 4B 75	133 27 [5B 91	153 11 k 6B 107	173 27 { 7B 123
1 1 Ø Ø	14 FF C 12	34 FS 1C 28	54 12 , 2C 44	74 28 < 3C 60	114 12 L 4C 76	134 28 \ 5C 92	154 12 l 6C 108	174 28 |* 7C 124
1 1 Ø 1	15 CR D 13	35 GS 1D 29	55 13 – 2D 45	75 29 = 3D 61	115 13 M 4D 77	135 29] 5D 93	155 13 m 6D 109	175 29 } 7D 125
1 1 1 Ø	16 SO E 14	36 RS 1E 30	56 14 . 2E 46	76 30 > 3E 62	116 14 N 4E 78	136 30 ∧ 5E 94	156 14 n 6E 110	176 30 ~ 7E 126
1 1 1 1	17 SI F 15	37 US 1F 31	57 15 / 2F 47	77 UNL ? 3F 63	117 15 O 4F 79	137 UNT — 5F 95	157 15 o 6F 111	177 DEL (RUBOUT) 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	
							(PPE)	(PPD)

*| on some keyboards or systems

KEY

octal	25 PPU	GPIB code
	NAK	ASCII character
hex	15 21	decimal

Tektronix
COMMITTED TO EXCELLENCE

REF: ANSI STD X3. 4-1977
IEEE STD 488-1978
ISO STD 646-1973

TEKTRONIX STD 062-5435-00 4 SEP 80

Fig. A-1. ASCII and IEEE (GPIB) Code Chart.