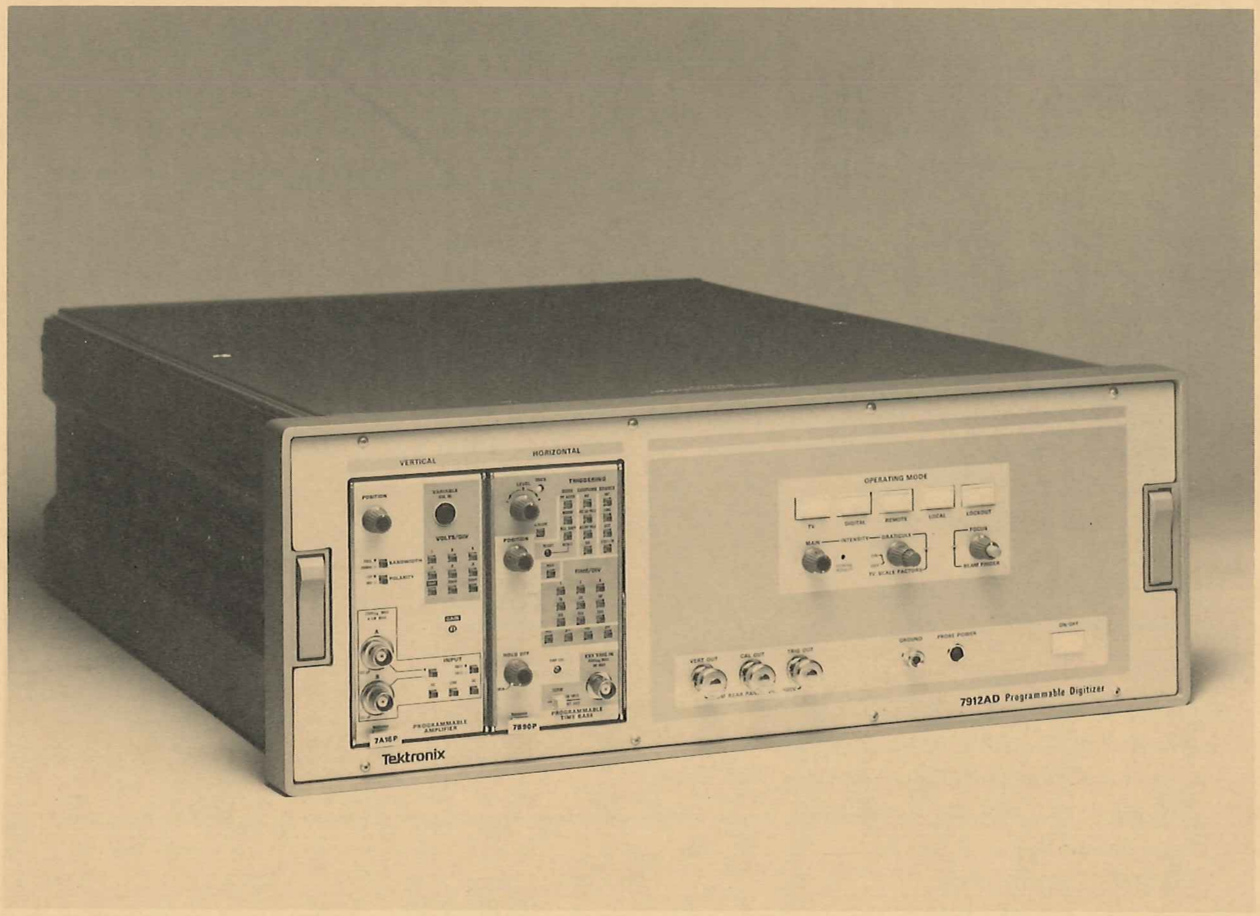
TEK INTERFACING INFORMATION

GPIB MADE EASY

7912AD Instrument Interfacing Guide



This interfacing guide is designed to help you get started using the Tektronix 7912AD Programmable Digitizer on the GPIB as quickly and easily as possible. It provides information on setting up the 7912AD for GPIB operation as well as some sample programs that illustrate a few simple operations with the 7912AD on the GPIB. This guide is not intended to take the place of the 7912AD Operators Manual or the manuals for your controller. These manuals will help you become familiar with the controller and the 7912AD.

For additional copies, order:
070-5077-00
Product Group 45
First Printing JUL 1984
Revised FEB 1986

Tektronix®
COMMITTED TO EXCELLENCE

TABLE OF CONTENTS

SETTING UP THE 7912AD

Before you can communicate with your 7912AD on the GPIB, you must:

- Select primary and secondary addresses
- Select the appropriate message terminator
- Connect the GPIB cables

Of course, if the primary address, secondary address, and the message terminator are already set the way you want them, you don't have to change anything.

The switches and jumpers that set the primary and secondary bus addresses and message terminator are located inside the 7912AD. Since hazardous voltages are exposed by removing the 7912AD cover, the address and message terminator selection must be performed by qualified personnel only.

WARNING

Hazardous voltages exist inside the 7912AD. The PRINCIPAL POWER switch on the rear panel must be turned off and the power cord must be disconnected before opening the instrument cover. Internal settings should only be made by qualified personnel.

Address Selection

When you are choosing a bus address for the instrument, keep a few things in mind. First, the address must be unique—no other devices on the same bus may have the same address. Second, remember that some controllers reserve an address for themselves. For instance, the Tektronix 4050-Series Desktop Computers reserve address zero for themselves, and the Tektronix 4041 uses address 30 by default (which may be changed). If your controller reserves an address, the instrument cannot be set to that address.

Finally, setting any device to address 31 effectively removes it from the bus—the device cannot be addressed.

Factory Address Settings

Factory address settings vary. If your 7912AD was purchased as a stand-alone instrument, the primary address is probably set to 1 and the secondary address is probably set to 0. If your 7912AD was purchased as part of a system, the addresses are set as appropriate for the system. The following discussion tells you how to find the current address settings.

What are the Current Address Settings?

There are two ways to find the current address settings. You can press the front-panel REMOTE but-

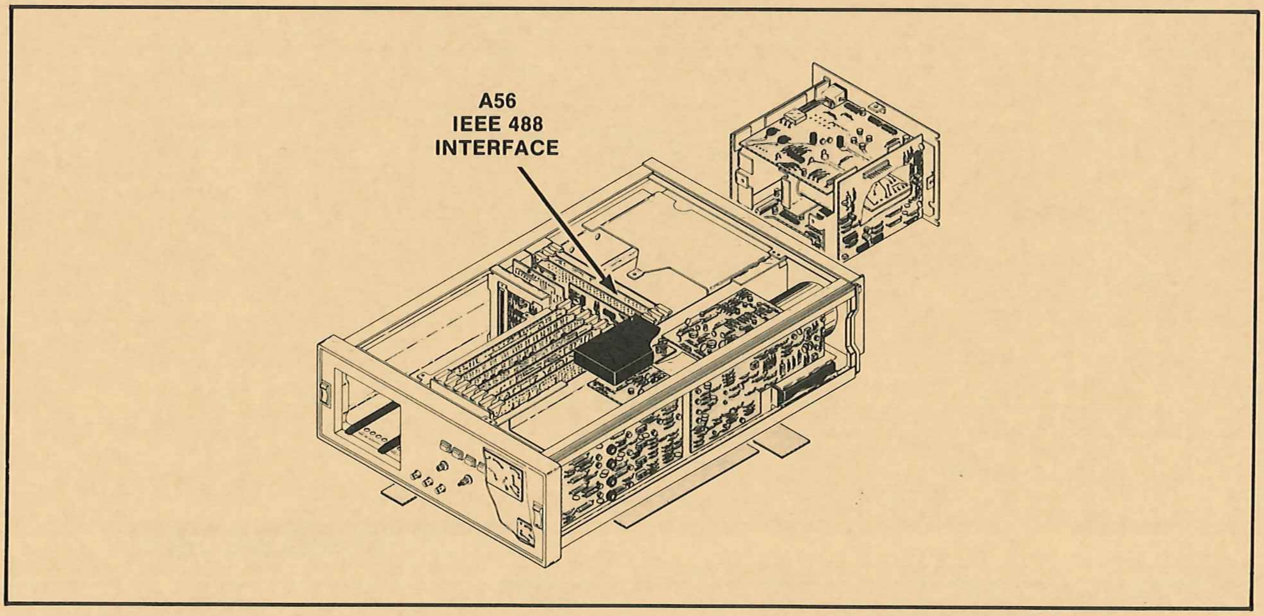


Fig. 1. *Location of the IEEE 488 Interface Card. It contains the DIP switches to set the primary and secondary addresses for the 7912AD.*

ton while the 7912AD is in TV mode (the TV SCALE FACTORS switch must be ON); the primary (PA) and secondary (SA) addresses are displayed on the TV monitor. The other way to determine the address settings is to to remove the top cover and look at the address DIP switches, located on the IEEE 488 Interface card. Figure 1 shows the location of the IEEE 488 Interface card. Figure 2 shows the locations of the DIP switches on the card. (Again, there are hazardous voltages present inside the 7912AD; this operation should be performed only by qualified personnel.)

Setting the Primary Bus Address

The IEEE 488 interface board contains two sets of switches near the top of the board (Fig. 2). The left switch (S400) sets the primary address in the range of 0-31. Each individual switch sets one bit of a binary number that represents the address. A switch set in the OPEN position (down) represents a "1" in that bit. For example, to set the primary address to 10, the switches are set as shown in Fig. 2.

Setting the Secondary Bus Address

The 7912AD uses a secondary address to allow the controller to select the mainframe or one of the programmable plug-ins installed in the mainframe. The secondary address switch (S402) is just to the right of the one that controls the primary address. The secondary address is set in the same way as the primary address. The switch sets the mainframe secondary address. The plug-ins automatically assume the next two consecutive secondary addresses as follows:

Plug-in Compartment	Secondary Address
Channel A plug-in	Mainframe secondary address + 1
Channel B plug-in	Mainframe secondary address + 2

Figure 2 shows how the switches would be set to select a secondary address of 3 for the mainframe. This setting means that the left (Channel A) plug-in would be set to secondary address 4, and the right (Channel B) plug-in would be set to secondary address 5.

Secondary addresses do not have to be unique—several instruments can use the same secondary address as long as they have unique primary addresses. However, if you are using programmable plug-ins, do not set the mainframe secondary address higher than 28 (decimal) since you must leave room for the plug-ins to assume the next two higher addresses (29 and 30). The highest valid secondary address is 30.

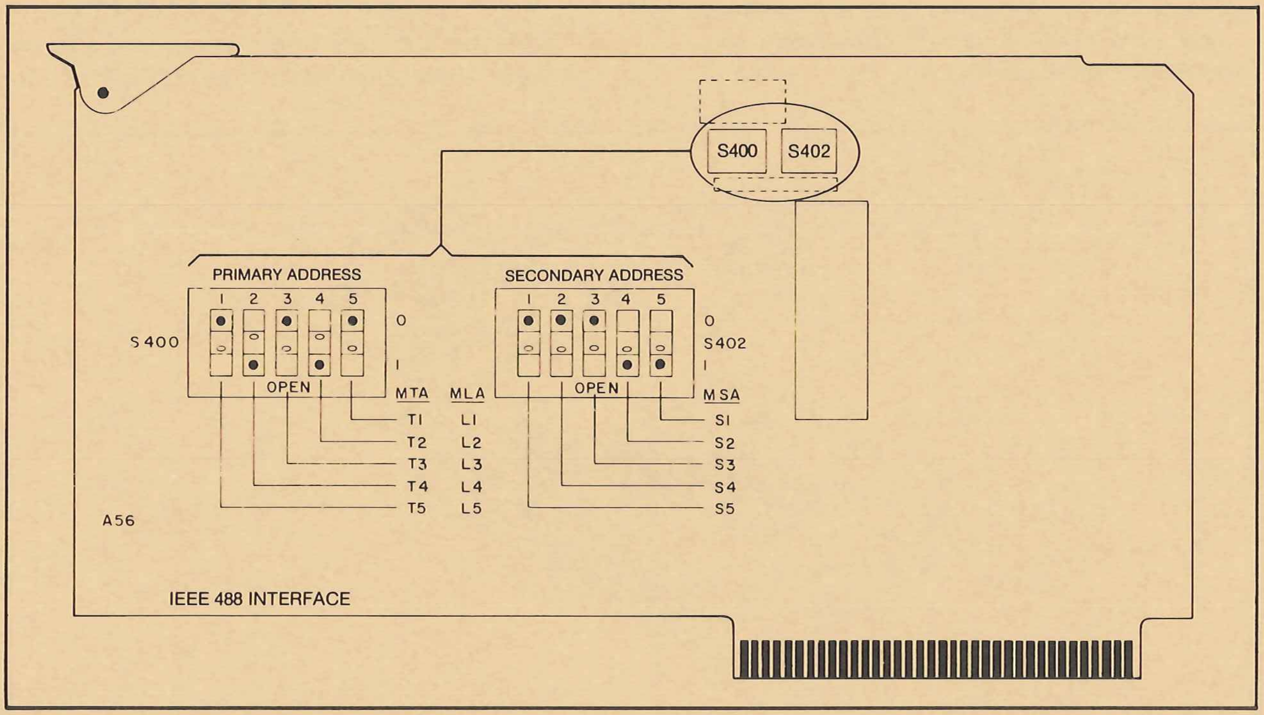


Fig. 2. *Location of the DIP switches that control the primary and secondary GPIB addresses of the 7912AD and its plug-ins. The left DIP switch (S400) controls the primary address and is set to 10 in this example. The right switch controls the secondary address and is set to 3 in this example.*

Setting the Message Terminator

The 7912AD is factory set to use the EOI (End Or Identify) bus line as the message terminator. Alternatively, you can set the message terminator to be line feed. A jumper on the Translator board (A22) selects the message terminator. Figure 3 shows the location of the Translator board. Figure 4 shows the location of the jumper (P410). Placing the jumper on the two lower pins selects the EOI message terminator; placing it on the two upper pins selects the line feed message terminator. When the 7912AD is sending data with the line feed delimiter, it adds a carriage return and line feed to the end of the message and asserts EOI with the line feed.

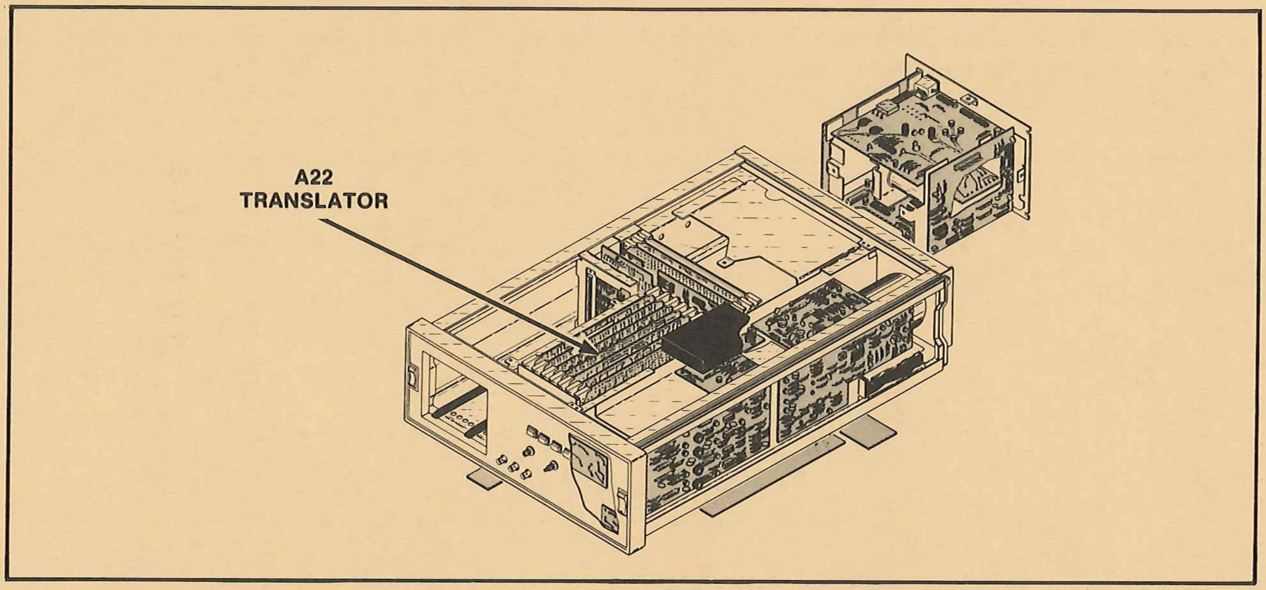


Fig. 3. *Location of the Translator board containing the jumper that selects the message terminator.*

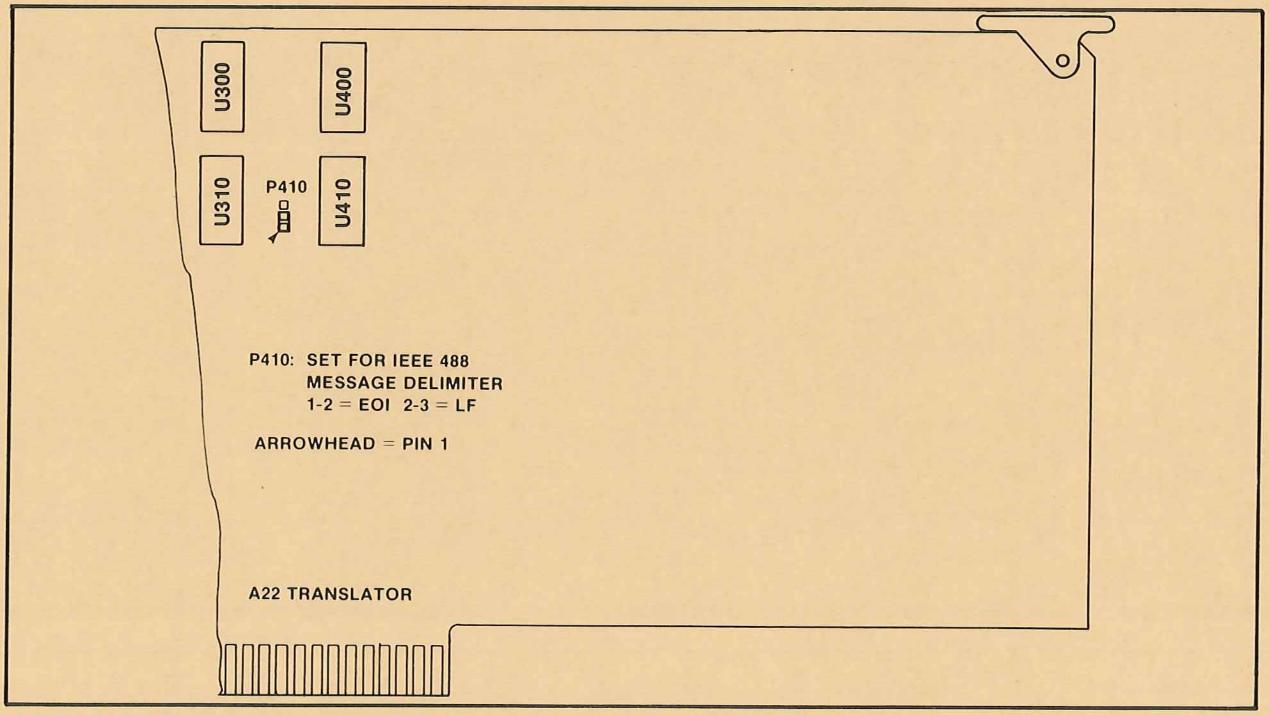


Fig. 4. *Location of the jumper that selects the message terminator. The jumper is set to the EOI message terminator in this example.*

The line feed terminator position should only be used with controllers that require line feed as a message terminator. Many controllers, including the Tektronix 4041 and 4050-series, have commands that allow you to change the terminator. Check your controller manuals for more information.

Cabling the GPIB System

Attach the 7912AD to the GPIB using a standard GPIB cable. The GPIB system may be cabled in two general configurations: star or linear (Fig. 5). While the star is the recommended configuration, these configurations may be mixed as long as the total cable length does not exceed 20 meters and the instruments are distributed on the bus according to a few rules.

First, no more than 15 total devices (including controller) can be included on a single bus. In addition, to maintain the bus electrical characteristics, one device load must be connected for every two meters of cable (generally each instrument represents one device load to the bus). The 7912AD represents one device load.

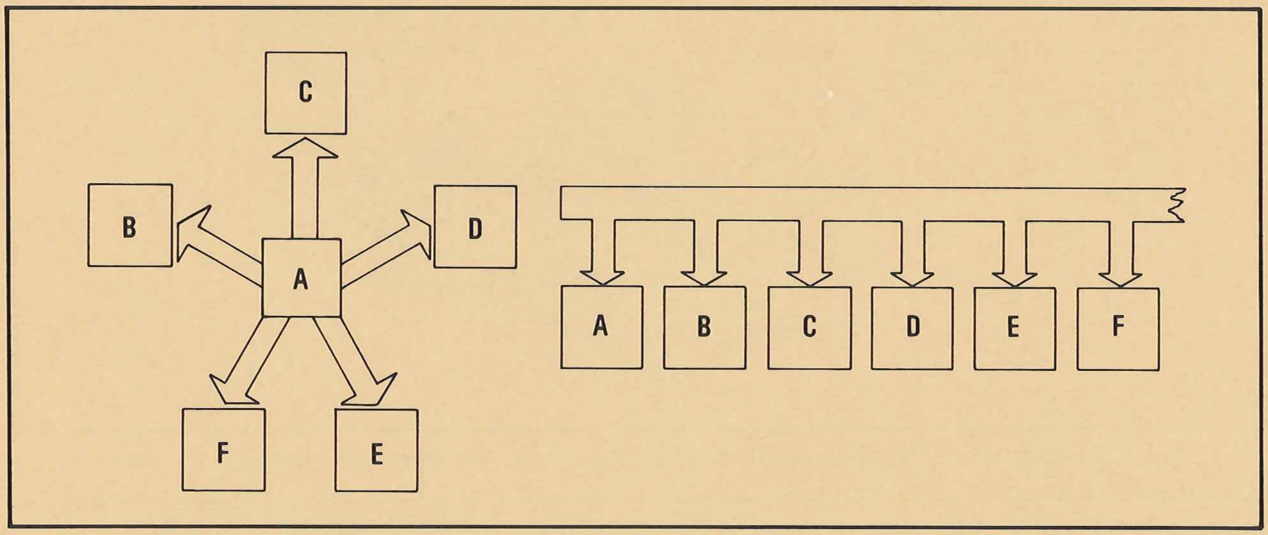


Fig. 5. *The two basic GPIB cable configurations are star and linear, but the two can also be mixed as long as a few restrictions are observed.*

PROGRAMMING THE 7912AD

Power Up

With the 7912AD cabled and with addresses and message terminator set, you're ready to power-up the system. Perform the power-up according to the instructions in your instrument manual. If the 7912AD passes the power-up tests, it asserts SRQ and sets the status byte to 65 (normal power-up). If you are using programmable plug-ins, they assert SRQ, too.

Usually, you'll want to clear the power-up SRQ by doing a serial poll. However, if your controller's SRQ interrupt is disabled, the power-up SRQ can be ignored.

Power-Up Default Settings

When the 7912AD powers up, all instrument settings are set to default values. Table 1 shows the default power-up settings for the 7912AD, the 7A16P Programmable Amplifier, and the 7B90P Programmable Time Base.

TABLE 1
POWER-UP DEFAULT SETTINGS

7912AD		
	MODE	TV
	TW	100
	RT	2
	DT	OFF
	GRAT	OFF
	XYZ	OFF
	REM	OFF
	OPC	OFF
	DEF	OFF
7A16P		
	BANDWIDTH	FULL
	POLARITY	+UP
	VOLTS/DIV	5
	INPUT	A
	INPUT IMPEDANCE	1 MΩ
	INPUT COUPLING	DC
7B90P		
	TRIGGERING MODE	PP AUTO
	TRIGGERING COUPLING	AC
	TRIGGERING SOURCE	INT
	TRIGGERING SLOPE	+
	TIME/DIV	1 μs
	MAGNIFICATION	1X

GPIB Messages

All messages transferred over the GPIB can be divided into two general classes:

- INTERFACE MESSAGES
- DEVICE-DEPENDENT MESSAGES

INTERFACE MESSAGES are commands that control interface functions. The IEEE 488 standard specifies these messages so that they are standard for all devices. The controller asserts the ATN line on the GPIB to distinguish these messages from the device-dependent messages. The commands used to send these messages depend entirely upon the particular controller used.

7912AD Response to Interface Control Messages

The 7912AD does not respond to the following interface control messages:

- PPC-Parallel Poll Configure
- PPU-Parallel Poll Unconfigure
- TCT-Take Control

The 7912AD does respond to the other interface control messages as follows:

GTL Go To Local. This causes the instrument to go to the local state. In this state, device-dependent messages continue to be accepted by the instrument, but only commands that do not affect the local operating controls or data memory will be executed.

GET Group Execute Trigger. If the 7912AD is in the DT ON state and a DIG command is waiting, the GET interface message causes the 7912AD to begin acquiring data.

LLO Local Lockout. This will lock out all front-panel controls of the 7912AD if the instrument is in the remote state. If the 7912AD is in the local state, LLO causes it to enter the local with lockout state. In remote with lockout state, only the BEAMFINDER, REMOTE, and ON/OFF controls are functional.

SDC,DCL Selected Device Clear and Device CLear. Either of these messages will reset the instrument input and output buffers used for transfer of device-dependent messages, and if in the waveform transfer mode, will halt all transfers. Also resets the status byte.

SPE,SPD Serial Poll Enable and Serial Poll Disable. The 7912AD has full serial poll capability.

IFC Interface Clear. This will reset the interface only and does not affect any instrument function.

DEVICE-DEPENDENT MESSAGES consist of the commands or data that control instrument functions. For example, a command to set the 7912AD digitizing rate is a device-dependent message. These types of messages are sent with the GPIB ATN line in the unasserted state. The message content and format is not specified in the IEEE 488 standard, and therefore is left to the instrument designer.

These device-dependent messages are the basic vocabulary of the instrument specifically provided to control the various functions of that instrument. This vocabulary is called the command set, and is unique to each instrument.

Device-dependent messages are further divided into two classes:

- SET COMMANDS-commands that tell the instrument to perform a function
- QUERY COMMANDS-commands that ask the instrument a question

The 7912AD command set is listed in the appendix of this guide. The list shows the commands and the valid arguments. Notice that many of the commands can be used as a set command (a command that sets a parameter or initiates an operation) or, by adding a question mark to the end of the command, they can be used as a query (a command that returns a current setting or value). For example, the REC command sets the number and length of records, while the REC? command returns the current number and length of records.

Multiple commands can be sent in a single message with the commands separated by semicolons. But there can only be one command, per message, that requires output from the 7912AD, and it must be at the end of the message. If the message is longer than 256 characters, the 7912AD will stop accepting characters long enough to process the first 256 characters and execute the commands. Then it will accept the remainder of the message.

Sending Commands to the 7912AD

Most GPIB controllers provide a set of simple statements that allow you to transfer device-dependent messages to and from GPIB instruments. In the 4050-series and the 4041 system controllers, the PRINT statement accomplishes this function. For example, the following statement sends a string of commands to the 7912AD from a 4052A. In this example, the 7912AD is set for primary address 10 and the mainframe secondary address is zero.

```
PRINT @10,0:"DIG DAT;READ PTR,VER"
```

For the 4041, the same message can be sent with the following statements:

```
OPEN #100:"GPIB(PRI=10,SEC=0):"
PRINT #100:"DIG DAT;READ PTR,VER"
```

Notice from the above examples that the syntax of the 7912AD message in the quotes is the same. All that changes is the syntax of the controller's output statement. Generally, there are three components in any GPIB output statement: The keyword (such as PRINT, WRITE, etc.), the GPIB address, and the message itself. Once you understand the syntax of the controller's output statement, it's just a matter of plugging in the 7912AD commands you need.

Getting the 7912AD Response

When you send a query or data transfer command, such as SET? or READ ATC, the 7912AD expects to be addressed to talk so that it can send the response. Like the simple PRINT statements illustrated above, most controllers provide some type of INPUT or READ statement to get data from a GPIB device. Consider, for example, the following statements in 4050 BASIC.

```
PRINT @10,0:"VS1?"
INPUT @10,0:VS1$
```

The first statement sends a command to the 7912AD that tells it to transmit its vertical scale factor for channel 1 of the vertical plug-in. In response to the VS1? query, the 7912AD returns a message of the form:

```
VS1 +500.E-03;
```

The second statement above reads this response and stores it in the string variable VS1$.

4041 BASIC simplifies the process of getting a query response with an INPUT PROMPT. This statement allows you to send the 7912AD a command string with a query and get the response, all in one line. The following 4041 statement duplicates the two 4052A statements above.

```
INPUT PROMPT "VS1?" #100:VS1$
```

Sometimes it's helpful to have the numeric part of query responses stored in numeric variables instead of string variables. This allows a program to operate on the values directly instead of extracting the values from the string. All you have to do to get the numeric values is substitute numeric variables for the string variable in the INPUT statement like this for the 4041:

```
INPUT PROMPT "VS1?" #100:DUMMY,VS1
```

The 4041 sends the query command as before, but this time it only stores the numeric parts of the response. Since there are two numeric values (VS1 +500.E-03;) in the response, each value is stored in a separate variable. The first value—the 1 in VS1—is not needed and is stored in DUMMY, and the second value—the vertical scale factor—is stored in VS1.

The same technique can be used in the 4052A with these statements:

```
PRINT @10,0:"VS1?"
INPUT @10,0:DUMMY,VS1
```

Getting Waveform Data from the 7912AD

Waveform data from the 7912AD is a little different than query responses. The 7912AD sends its waveform data in binary instead of ASCII like the query responses. Binary data transfer requires fewer bytes than an equivalent ASCII data transfer, so the transfer is faster. But receiving binary data usually requires using some different commands than receiving ASCII data.

The 7912AD sends its binary data in a binary block format specified by the Tektronix Codes and Formats Standard. The format of the binary block is:

%<byte count> <data value>...<checksum>;

Where:

% is the ASCII percent character.

BYTE COUNT is a 16-bit binary number sent in two bytes, most significant byte first. The value indicates the number of bytes that remain to be transmitted in the block, including the checksum, but not including the message unit delimiter (semicolon).

DATA VALUE is a 16-bit binary number sent in two bytes, most significant byte first. If the data value is less than 16 bits in length, it is sent with the unused high-order bits set to 0. For example, the decimal number 511 would be sent in two bytes: 00000001 and 11111111.

CHECKSUM is an eight-bit, two's-complement binary number that is the modulo-256 sum of all preceding bytes in the block excluding the % character.

; is the ASCII semicolon character

Reading binary block data. To read binary block data with the 4052A, you can use low-level RBYTE and WBYTE commands or you can use GPIB ROM pack commands if you have the 4052R14 GPIB Enhancement ROM Pack. If you are using simple RBYTE commands to read binary data from the bus, the percent sign and both bytes of the byte count must be read into individual variables with an RBYTE statement. Then the waveform data can be read into an array. Finally, the checksum and semicolon must be read. For example, the 4052A (without the 4052R14 ROM Pack) routine to read a binary block from the 7912AD would look something like this:

```
100 DIM WAVEFORM(512)
110 PRINT @1,0:"READ ATC"
120 WBYTE @65,96:
130 RBYTE HEADER,BCNT_HI,BCNT_LO
140 FOR I=1 TO 512
150    RBYTE HIBYTE,LOBYTE
160    WAVEFORM(I)=HIBYTE*256+LOBYTE
170 NEXT I
180 RBYTE CHECKSUM,SEMICOLON
190 WBYTE @95:
```

The preceding routine is memory efficient but relatively slow. You can trade memory efficiency for speed with the following routine:

```
100 DIM WAVEFORM(512),WAVETMP(512,2)
110 DIM CONSTANT(2)
120 CONSTANT(1)=256
130 CONSTANT(2)=1
140 PRINT @1,0:"READ ATC"
150 WBYTE @65,96:
160 RBYTE HEADER,BCNT_HI,BCNT_LO,WAVETMP
170 RBYTE CHECKSUM, SEMICOLON
180 WBYTE @95:
190 WAVEFORM=WAVETMP MPY CONSTANT
```

This routine is about three times faster but takes about three times as much memory. Fortunately, most

Tektronix controllers have a command to automatically read binary blocks including all the formatting and error-checking bytes. These commands are usually both memory efficient and fast. With a 4052R14 ROM pack, the program shown above reduces to:

```
100 DIM WAVEFORM(512)
110 PRINT @1,0:"READ ATC"
120 CALL"BININ","PACK",WAVEFORM,ERR;1,0
```

With the 4041, you can do the same thing with the following program segment:

```
100 INTEGER WAVEFORM(512)
110 OPEN #100:"GPIB(PRI=1,SEC=0,TIM=4):"
120 PRINT #100:"READ ATC"
130 INPUT #100 USING "+16%":WAVEFORM
```

SRQs and Instrument Status

The 7912AD asserts SRQ for the following conditions:

- POWER ON — Every time the 7912AD powers up, it asserts SRQ. (The 7A16P and the 7B90P do too.)

- ERROR — If the 7912AD detects invalid command syntax, an execution error, or an internal error, it asserts SRQ.

- REMOTE REQUEST — If REMOTE REQUEST is set the 7912AD asserts SRQ when the REMOTE button is pushed.

- POWER FAIL — The 7912AD asserts SRQ when it detects an imminent power failure. It can still respond to a serial poll for 10 milliseconds after asserting SRQ.

To clear the SRQ, you must either poll the device or send a device clear message (DCL or SDC). The 7912AD reports a status byte when it is polled by the controller. The status byte contains the internal status of the instrument. Two main types of status bytes may be reported:

- SYSTEM STATUS which indicates conditions that are common among all instruments that conform to the Tektronix Standard Codes and Formats (e.g., Command Error).

- DEVICE STATUS which indicates conditions that are unique to that instrument type only. The 7912AD has only one device dependent condition: REMOTE REQUEST. All others are system status conditions.

The status byte reports errors in general terms. Use the ERR? query to more specifically define an error.

7912AD Status Bytes

	BIT #								DECIMAL CODE	
STATUS	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**NOT BUSY**	**BUSY**
NORMAL CONDITIONS (SYSTEM STATUS)										
No Condition	0	0	0	X	0	0	0	0	0	16
Power On	0	1	0	X	0	0	0	1	65	81
Operation Complete	0	X	0	X	0	0	1	0	66(2)	82(18)
ABNORMAL CONDITIONS (SYSTEM STATUS)										
Command Error	0	1	1	X	0	0	0	1	97	113
Execution Error	0	1	1	X	0	0	1	0	98	114
Internal Error	0	1	1	X	0	0	1	1	99	115
Power Fail Error	0	1	1	X	0	1	0	0	100	116
DEVICE DEPENDENT STATUS										
Remote Request	1	X	0	X	0	0	0	1	192(129)	209(145)

Fig. 6. *7912AD Status bytes. The X in column 5 is 1 if the 7912AD is busy and 0 if not. There's an X in column 7 where the 7912AD could set the status byte without asserting SRQ. The numbers in parentheses are the values of the status bytes if bit 7 is not set.*

The status byte. The status byte read from the 7912AD during a serial poll contains the following information:

Bit 8	-	Device Status=1; System Status=0
7	-	Service Requested (SRQ)
6	-	Abnormal Condition=1; Normal=0
5	-	Busy
4	-	Device/System status
3	-	Device/System status
2	-	Device/System status
1	-	Device/System status

Figure 6 shows the bit settings of the eight status byte values that the 7912AD uses.

The following 4052A program shows how you can process SRQ's:

```
100 DIM DIGITIZER(3,2)
110 CALL "CONFIGURE",ERRCODE;DIGITIZER  ! Initialize device array
120 ON SRQ THEN 2000

130 WAIT
140 GOTO 130

1999 STOP
2000 POLL DEVICE,STATUS;DIGITIZER
2010 PRINT "SRQ FROM INSTRUMENT ";
2020 IF STATUS>=128 THEN 4000  !Is device-dependent bit set?
2030 INDEX=(INT(STATUS/32) MOD 2)*2+(STATUS MOD 16)+1
2040 GOSUB INDEX OF 2080,3000,3020,3040,3060,3080,4000,4020,4040
2050 PRINT "** NO CONDITION **"
2060 RETURN
2070 PRINT "** POWER ON **"
2080 RETURN
2090 PRINT "** OPERATION COMPLETE **"
3000 RETURN
3010 PRINT "** COMMAND ERROR **"
3020 RETURN
3030 PRINT "** EXECUTION ERROR **"
3040 RETURN
3050 PRINT "** INTERNAL ERROR **"
3060 RETURN
3070 PRINT "** POWER FAIL **"
3080 RETURN
3090 PRINT "** REMOTE REQUEST **"
4000 RETURN
```

Lines 130 and 140 cause the program to wait for an SRQ interrupt repeatedly. Normally you would replace them with a program of your own to do the processing you want to do. This program is for the Tektronix 4052A. Similar techniques can be used with other controllers.

APPENDIX

7912AD Command Set

Header	Argument	Description	Notes
MOD[E]	TV	Set instrument to TV mode	3,8
	DIG	Set instrument to digital mode	8
DIG	DAT[A]	Digitize data	1,4,8
	GRA[T]	Digitize graticule only	1,4,8
	SSW	Digitize on single sweep trigger	1,4,8
	DEF,<NR1>	Digitize only defects n times	1,4,8
	SA,<NR1>	Digitize and signal average 1 to 64 times	1,4,8
DT	ON	Wait for GET interface message to digitize	
	OFF	Do not wait for GET interface message to digitize	3
GRAT	ON	Write only the graticule on the target	8
	OFF	Reset graticule-only mode	3,8
XYZ	ON	Enable XYZ outputs to display raw data	4,8
	OFF	Disable XYZ outputs	3,4,8
	RAW	Same as ON argument	4,8
	ATC	Enable XYZ outputs to display ATC data	4,8
	SA	Enable XYZ outputs to display signal-averaged data	4,8
	EDG[E]	Enable XYZ outputs to display edge-determined data	4,8
	DEF	Enable XYZ outputs to display defects data	4,8
MAI	<NR1>	Set main intensity from 0 to 1023	8
GRI	<NR1>	Set graticule intensity from 0 to 255	8
FOC	<NR1>	Set focus from 0 to 63	8
SSW	ARM	Arm single-sweep trigger	8
	DIS	In single-sweep mode, but disarmed	2
	NSS	Not in single-sweep mode	2
TV	ON	Turn on TV display of scale factors	8
	OFF	Turn off TV display of scale factors	8
REM	ON	Assert SRQ when REMOTE pressed	
	OFF	Do not assert SRQ when REMOTE pressed	3
OPC	ON	Assert SRQ when operation complete	
	OFF	Do not assert SRQ when operation complete	3
DEF	ON	Flag defects in raw vertical data	4,8
	OFF	Reset defect flags in raw vertical data	3,4,8
LOA[D]	<BINARY BLOCK>	Load defects array from IEEE 488 bus	1,4,6,8
ATC		Perform simple ATC on raw vertical data	1,4,8
INT	<NR1> or NONE	Max. no. of consecutive interpolated data points	2
EDG[E]		Determine edges of raw waveform	1,4,8
TW	<NR1>	Set max. trace width for EDGE from zero to 512	4
RT	<NR1>	Set max. ratio of trace widths for EDGE from 1 to 32767	5

Header	Argument	Description	Notes
SET	<MESSAGE UNITS>	Settings of programmable functions (header is omitted)	2
TES[T]		Self-test data memory	1,4,8
REA[D]	VER	Transmit vertical data array	1,4,7
	PTR	Transmit pointers data array	1,4,7
	SC1	Transmit channel 1 scale factors	1,7
	SC2	Transmit channel 2 scale factors	1,7
	ATC	Transmit average-to-center data	1,4,7
	SA	Transmit signal-averaged data	1,4,7
	EDG[E]	Transmit edge-determined data	1,4,7
	DEF	Transmit defect data	1,4,7
REP	<NR1>	Repeat DIG DAT/READ PTR,VER sequence 1 or more times	1,4,8
DUM[P]	RAW	Dump raw data memory area	1,4,7
	PR	Dump processed data memory area	1,4,7
VS1	<NR3> or NONE	Scale factor for vertical channel 1	2
VS2	<NR3> or NONE	Scale factor for vertical channel 2	2
HS1	<NR3> or NONE	Scale factor for horiz. channel 1	2
HS2	<NR3> or NONE	Scale factor for horiz. channel 2	2
VU1	<CHARACTERS>	Units for vertical channel 1	2
VU2	<CHARACTERS>	Units for vertical channel 2	2
HU1	<CHARACTERS>	Units for horizontal channel 1	2
HU2	<CHARACTERS>	Units for horizontal channel 2	2
ERR	<NR1> or NONE	Code for error indicated in last status byte reported	2
SRQ	NULL	Service request code (7912AD provides no other response)	2
ID	<CHARACTERS>	Identity of instrument	2

Notes:

1 — Can only be used as set commend.
2 — Can only be used as query commend.
3 — Power-up condition.
4 — Memory control operation.
5 — Divided by 32 when received by 7912AD.
6 — BINARY BLOCK is defined under Waveform Data I/O in your 7912AD Operators Manual.
7 — More than one argument allowed, arguments delimited by commas.
8 — Not executed in local state as set command (does not apply to query).

TEKTRONIX INSTRUMENTATION SOFTWARE LIBRARY

Instrument Utility Software for the 7912AD

Instrument Utility Software is available from Tektronix for the 7912AD Programmable Digitizer. This software consists of a set of subroutines and subprograms that perform common instrument functions over the GPIB such as data acquisition, front-panel set-up, etc. These routines are designed to be easily integrated into your application programs. And since they are small and well documented, the routines are easy to modify to suit your particular applications. Refer to the current Tektronix Instrumentation Software Library Catalog for instrument options, ROM packs, and other required equipment.

The following Instrument Utility Software was available when this Instrument Interfacing Guide was printed. Other software may be available; contact your local Tektronix Field Office for further information.

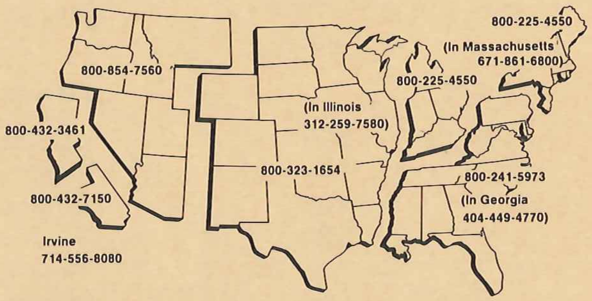
Description	Tektronix Part No.
7912AD/4041 Instrument Utility Software (DC-100 tape)	062-7787-00

Ordering Instrument Utility Software (U.S. Only)

Your local Tektronix Field Office has the current prices for software available from the Tektronix Instrumentation Software Library.

You may order Tektronix Instrumentation Software Library programs from Tektronix Central Parts Ordering by using the toll-free number serving your area. The following map identifies the geographical regions in the U.S. and the toll-free number serving each region.

Call the toll-free number serving your area and give the Customer Service Representative the Tektronix nine-digit part number and name of the software package you want to order. If you have any questions about the software, call your local Tektronix Field Office.



Ordering Instrument Utility Software (Outside the U.S.)

Outside of the U.S., order Tektronix Instrumentation Software Library programs through your local Tektronix sales office or from the Tektronix Instrumentation Software Library order point serving your area. Refer to the following list for the applicable library order point.

Africa, Europe, Middle East

Contact local Tektronix sales office.

Australia

Tektronix Instrumentation Software Library
Tektronix Australia Pty. Limited
Sydney
80 Waterloo Road
North Ryde, N.S.W. 2113

Canada

Tektronix Instrumentation Software Library
Tektronix Canada Ltd.
P.O. Box 6500
Barrie, Ontario
Canada L4M 4V3

Caribbean, Latin America, and Far East (except Japan)

Tektronix Instrumentation Software Library
Export Marketing
Tektronix, Inc.
P.O. Box 500
Beaverton, OR 97077
U.S.A.

Japan

Tektronix Instrumentation Software Library
Sony/Tektronix Corporation
9-31 Kitashinagawa-5
Tokyo 141 Japan

Program Library

The Tektronix Instrumentation Software Library includes over 200 software programs for a variety of Tektronix programmable instruments and controllers. The Library Catalog provides abstracts of the available software. Programs are available as ready-to-load media or as listings (see Catalog). For a copy of the latest catalog, contact your local Tektronix Field Office or representative and ask for Tektronix Instrumentation Software Library Catalog #45W-5570.

Program Contributions

If you have a program which you would like to submit to the Tektronix Instrumentation Software Library, we will send you, in exchange, one software package of your choice from the User-Exchange Software portion of the Program Library (see current library catalog). Submitted programs must use Tektronix programmable instruments and must meet certain coding and documentation standards.

To contribute a program, submit a copy of the program on media along with a listing and a Tektronix Instrument Software Library release form (see current library catalog). If the program was created as part of your employment, the release must be signed by an authorized representative of your employer. Acceptance of the program is subject to review of the Tektronix Instrumentation Software Library staff.

For further information on submitting a program or for information about coding and documentation standards, contact:

Tektronix Instrumentation Software Library
Tektronix, Inc.
Group 157, 54-016
P.O. Box 500
Beaverton, OR 97077

784

Tek Lit No. 45W-5542