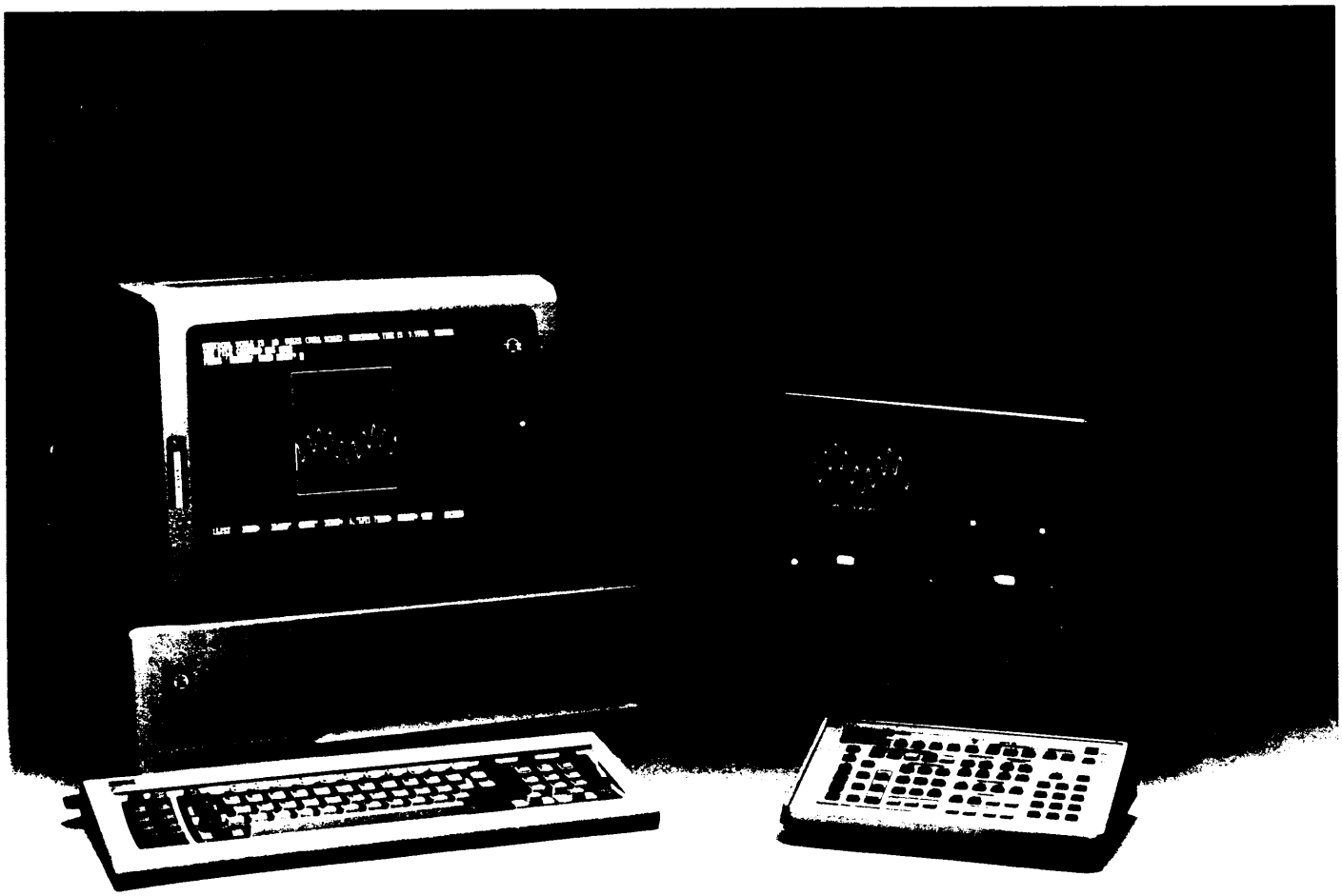
BASIC SOFTWARE PROGRAMS FOR COMMUNICATING BETWEEN THE 7854 AND IBM PC



Tektronix
COMMITTED TO EXCELLENCE

TABLE OF CONTENTS

FIGURE

INTRODUCTION

In the past few years, the personal computer has found its way into many new places. Today's engineering bench often includes one. A new challenge is using the engineering test equipment with the personal computer. Many instruments, like the Tektronix 7854 Oscilloscope, can be interfaced to a personal computer. For example, with the popular IBM PC and an IEEE-488 interface card from National Instruments (GPIB-PC), you can store or retrieve 7854 application programs on diskettes. You can also transfer waveforms, text, and commands between the 7854 and IBM PC.

This application note describes the installation and configuration of the National Instruments' IEEE-488 interface card, Model GPIB-PC2 and BASIC programs to make these transfers. It also contains programs written in IBM BASIC to do the following:

- Store and retrieve 7854 programs.
- Transfer and store waveforms from the 7854.
- Send any specific 7854 command.
- Send text to the 7854.
- Send waveforms stored on diskette back to the 7854.
- Graph waveforms stored on diskette.
- Send numeric data to the 7854.
- Transfer 7854 results to the IBM-PC.

NATIONAL GPIB CARD INSTALLATION

General Information. Adding circuit cards to the IBM PC and IBM XT is not difficult. Adding cards to the IBM portable PC is more difficult and is not discussed in detail here. Even though no changes were made to the factory settings, a quick review of the settings follows:

- The I/O port address is set by the switches at position U14. The factory setting is Hex 2B8. Figure 3-3 in this document and page 3-3 in the National Instruments' manual show this address setting.
- The DMA channel is selected by jumpers on a row of pins (P1). The factory setting is DMA channel 1. The National Instruments manual, the IBM PC manual, and manuals for other installed cards should be consulted before any changes are made to the selection of DMA channel 1.
- Interrupts are not implemented by the supporting software package. The jumper is left in the storage position.

Board Installation. The National Instruments' interface manual lists a procedure to install the card in the IBM PC, but here is a review:

- If your computer is an IBM PC XT, use the 'park' command to protect the disk.
- Turn off the PC and remove power cord.
- Remove the 5 screws from rear panel using a ¼" socket or straight blade screwdriver (see figure 2-1).
- Slide off the plastic housing.
- Check the switch settings on the National card (see figure 2-2).
- Remove the metal cover with a ⅛" socket or a straight blade screwdriver and install the National card (see figure 2-3).
- Slide the plastic cover on and reinstall the 5 screws.

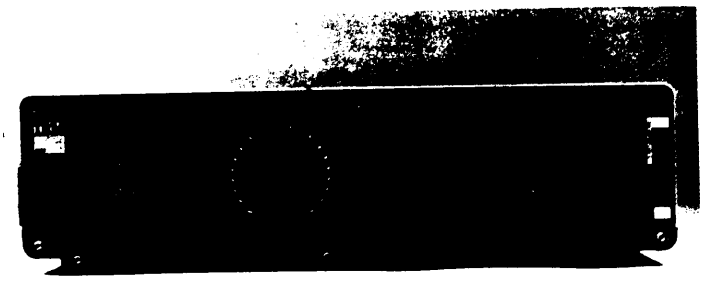


Fig. 2-1. IBM PC Rear Panel

Fig. 2-2. Switch Settings

Fig. 2-3. Installation of National Card

Software and Board Check

This section describes the board and software check out of the National Instruments' interface card. Programs are provided for the hardware, software, and interface settings. These programs are user interactive, requiring correct inputs from the keyboard. The software described in the following runs on IBM Personal Computer DOS Version 2.0 and up. If errors occur while running the installation checkout software, the following steps can be taken:

1. Check the card installation (another card could be using DMA channel 1).
2. Re-run the software (pay strict attention to the user prompt).

If the error still occurs, a call to National Instruments is in order. The number is listed in the manual for the interface.

Hardware Check.
IBDIAG.EXE confirms that the hardware circuits are working properly. To run the program, install the diskette in the A disk drive and enter "ibdiag." The program prompts you for inputs and displays any errors or failures (see figure 2-4).

Fig. 2-4. IBDIAG Results

```
A>ibdiag
National Instruments Diagnostic
GPIB-PC Customer Stand Alone Diagnostic
Copyright © 1984 National Instruments, Inc.

This diagnostic is provided to verify that you have correctly installed your GPIB-PC hardware. It tests the programmed I/O, DMA and functional capability of the GPIB-PC to the extent possible without a tester connected on the GPIB bus.

NOTE—This diagnostic must be run separately for each GPIB-PC card installed in your computer.

To run the diagnostic, you need to know which I/O address and DMA channel the GPIB-PC is set for. If you have not changed the settings on the card from what was set at the factory, you may use the default values listed in this program. Refer to section 3 of the GPIB-PC manual if you need further explanation of hardware configuration.

Do you wish to continue? (y/n) : y
   Enter board address in hex (default = 2B8) : 2B8
Is the GPIB-PC configured to use DMA? (y/n) : y
   Enter DMA channel number (1-3) (default = 1) : 1
   DISCONNECT ALL GPIB CABLES FROM THE GPIB-PC
   Press return to begin tests :
   Tests running . . . . . .
   GPIB-PC diagnostic tests successfully completed.
   Type e to exit : e
```

GPIB-PC is a Trademark of National Instruments Corporation. IBM PC and IBM PC-XT are Trademarks of International Business Machines.

Software Check. This is a two part program. The first part does not require a device connected to the GPIB. The second requires a GPIB monitor/analyzer. Only the first half of the program is discussed here. For more information about the second part of the program, see the National Instruments' interface manual. IBTEST.EXE verifies the correct software installation and the first test is generally sufficient. To run IBTEST, disconnect all instruments from the GPIB and enter "ibtest." An example of the IBM PC screen is shown in figure 2-5.

Interface Configuration. This program controls the instruments used on the GPIB. You must understand the requirements of each instrument. Is it a listen only device? Is it a talk/listen device? What terminating character is required? The software lets you make these choices, but you must understand the requirements of the instruments with which you want to communicate. To run the software, enter "ibconf." Figure 2-6 shows the screens of the software as it is being run. The 7854 uses the default parameters; an <ENTER> will leave the parameters unchanged.

Fig. 2-5. IBTEST Results

```
A>ibtest

A>ibtsta
Software Installation Test.

This test verifies the proper installation and operation of
the software in a stand alone or isolated environment
and requires no interaction with the user. Disconnect any
instruments that are connected to the GPIB-PC, then
press return to begin.

Did you turn off DMA capability for board GPIB when you
configured your handler software? (y/n)
n

Test in progress...

Test complete.

If no errors occurred the software is properly installed and
operational.

A>ibtstb
Software Installation Test.

This test requires some user interaction plus the use of a
GPIB bus tester or analyzer that is able to monitor and
display the state of the GPIB lines, accept data from the
board, and send data to the board. The National Instru-
ments GPIB-400 Bus Tester or equivalent device from
another manufacturer can be used for this purpose.
Attach the tester to the board and press return to begin. If
no tester is available, exit this test by entering <control> c.

Terminate batch job (y/n)? y
A>
```

Fig. 2-6. IBCONF Results

```
A>ibconf

National Instruments GPIB-PC Software Configuration
Utility, Rev B.2 Copyright © 1984 National Instruments,
Inc.

This program allows you to reconfigure certain param-
eters defined in the file GPIB.COM and automatically
edits the file CONFIG.SYS as needed. If you have not
already done so, copy this program and the file
GPIB.COM to the root directory of the boot disk drive and
restart IBCONF from there.

  • The binary file will NOT be altered when the DEFINE
    command is used to enter new default values.
  • The binary file WILL be altered if the SAVE command
    is used to write out new default values.
  • At any time before issuing the SAVE command, you
    may quit this program by typing <CTRL-C> in
    response to the arrow (>) prompt, and the binary file
    will remain unchanged.

IT IS STRONGLY RECOMMENDED THAT YOU MAKE
A BACKUP COPY OF THE FILE TO BE CONFIGURED
BEFORE PROCEEDING WITH THIS PROGRAM.

Press return to begin.
>>>>

Scanning file spib.com...reading data...completed.
2 Board(s)   16 Device(s)
```

Fig. 2-6. (cont)

DEFINE SYSTEM PARAMETERS
(effective for all boards)

PARAMETER	DEFAULT	NEW VALUE
Reference Clock Frequency? (decimal)	[8 MHz]. . >	
Disable Automatic Serial Polling?	[no]. . >	

You may now proceed with board and device level configurations. If you need help, type HELP.

cmd> H

COMMAND		EXPLANATION
Sh[ow] or Sh[ow]	<unit name> -ALL	SHOW currently defined values for a board or device, or show all board names and the names of the devices they access.
D[efine] or D[efine]	<unit name> -ALL	DEFINE new default values for an existing board or device, or define new values for all units from a specified source file.
Sa[ve]		SAVE the currently defined values for all boards and devices to the file on disk.
H[elp]		HELP! Display this screen.
E[xit]		EXIT this program.

NOTE 1—New devices cannot be defined, but existing devices can be redefined by changing their name field (see DEFINE).

NOTE 2—If <unit name> is omitted from SHOW or DEFINE, then the last board or device referenced will be used.

Fig. 2-6. (cont)

cmd> Sh GPIBO

NAME: GPIBO **NUMBER: B O**

(1) Primary GPIB Address 00H
(2) Secondary GPIB Address . 0:none
(3) Timeout Setting 13
(4) EOS Byte 00H
(5) EOS Modes . . . . . none
(6) Board Options . . . . . (SC EOT)
(7) Port Address 02B8H
(8) DMA Channel 1
(9) High-speed Timing . . . . . OFF

Footnotes:

CS	Board is system controller.
EOT	Send EOI w/last byte of write.
REOS	Terminate read on EOS.
XEOS	Send EOI with EOS byte.
BIN	Use 8-bit compare on EOS.
nnH	Hexadecimal number.

cmd> sh -all

FILE: gpib.com **BOARDS: 2** **DEVICES: 16**

Board GPIBO
>>>accesses>>> DEV 0 DEV 1 DEV2 DEV3
DEV4 DEV5 DEV6 DEV7
DEV8 DEV9 DEV10 DEV11
DEV12 DEV13 DEV14 DEV15

Board GPIB1 >>>access>>>

Fig. 2-6. (cont)

cmd> d GPIBO

NAME: GPIBO **NUMBER: B O**

PARAMETER	DEFAULT	NEW VALUE
Primary GPIB Address? (hex)	[00H]	>
Secondary GPIB Address? (hex)	[0:none]	>
Timeout settings? (decimal)	[13]	>
EOS byte? (hex or '<char>')	[00H]	>
EOS modes (REOS XEOS BIN)		
Terminate read on EOS?	[no]	>
Send EOI with EOS byte?	[no]	>
Use 8-bit compare on EOS?	[no]	>
Board Options (SC EOT)		
Board is System Controller?	[yes]	>
Timeout setting? (decimal)	[13]	>
EOS byte? (hex or '<char>')	[00H]	>
EOS modes (REOS XEOS BIN)		
Terminate read on EOS?	[no]	>
Send EOI with EOS byte?	[no]	>
Use 8-bit compare on EOS?	[no]	>
Board Options (SC EOT)		
Board is System Controller?	[yes]	>
Send EOI w/last byte of write?	[yes]	>
Port Address? (hex)	[02B8H]	>
DMA channel? (decimal)	[1]	>
Use high-speed (3-state) Timing?	[no]	>

cmd> sa
CONFIRM: save all boards and devices? (y/n)
>>>> y
cmd >
cmd> e
CONFIRM: exit ibconf? (y/n)
>>>> y
A>

User Trap. To change the configuration file, you cannot just make the change and run the program. The file containing the configuration information (config. sys) is read only when the system is booted up. Therefore, if you make changes to the configuration, re-boot the system. Pressing the CTRL, ALT and DEL keys simultaneously will re-boot the system and read the new configuration.

PROGRAMMING DETAILS

General Information

Loading the Interface Drivers. The software you receive with the National interface card contains a sample program (DECL.BAS). This program sets up a memory location to call a specific interface command. It can be loaded into memory and used as the first few lines in any program. Figure 3-1 shows a printout of the program. Each basic program that uses the IEE-488 interface must be loaded. Figure 4-1 is an example.

Software Organization. This software package is command line/menu driven. A menu is printed on screen and the menu item is selected by pressing the corresponding function key. By using the INKEY$ function in BASIC, the software waits for you to select the next software routine. There are ten subroutines, eight dealing with transfering information to and from the 7854. The ninth subroutine graphs waveforms that have been saved on diskette, and the last subroutine exits from BASICA back to DOS.

Programming the 7854

Using the 7854 Service Request. Many of the commands sent to the 7854 need time to complete. The 7854 has an interrupt scheme that is based on the serial poll system which tells the controller when it has finished the last task. You can use the software to ask the 7854 which state it is in, especially before asking the 7854 to do something else. Even though a simple, single command, such as "AQR", is sent to the 7854, the software must wait until the acquisition is complete before issuing further commands (see Figure 3-2).

Fig. 3-1. Loading of Drivers

```
5 REM !!! LINES 10 TO 60 SET UP THE GPIB
DRIVER CALLS!!!
10  CLEAR  ,60000!          ' GPIB-PC Rev. B.1
20  IBINIT1 = 60000!        ' BASICA Declaration File
30  IBINIT2 = IBINIT1 + 3 '
40  BLOAD "bib.m",IBINIT1
50  CALL IBINIT1 (IBFIND, IBTRG, IBCLR, IBPCT, IBSIC,
    IBLOC, IBPPC, IBBNA, IBONL, IBRSC, IBSRE,
    IBRSV, IBPAD, IBSAD, IBIST, IBDMA, IBEOS, IBTMO,
    IBEOT)
60  CALL IBINIT2 (IBGTS, IBCAC, IBWAIT, IBPOKE,
    IBWRT, IBWRTA, IBCMD, IBCMDA, IBRD, IBRDA,
    IBSTOP, IBRPP, IBRSP, IBDIAG, IBXTRC, IBSTA%,
    IBERR%, IBCNT%)
```

Fig. 3-2. Wait for Completion of Task

```
WRT$ = "0 WFM AQR"                ! STRING TELLS
                                    7854 TO
                                    ACQUIRE
                                    WAVEFORM
CALL IBWRT(DSO%,WRT$)             ! SEND THE
                                    STRING
IF IBSTA%<0 THEN GOSUB 190        ! CHECK THE
                                    GPIB STATUS
CALL IBRSP(DSO%,STA%): IF STA%<>2 AND
STA%<>66 THEN 5140
  ! IF THE 7854 HAS NOT COMPLETED THE
    ACQUIRE THEN WAIT at line 5140
```

Fig. 3-3. 7854 Switch Settings

Programming 'Overhead'. Writing programs in BASIC means that many of the lines of code have little to do with the task you are trying to do. These are programming 'overhead'. The subroutines listed on the pages that follow illustrate this. Each routine has 10 to 15 lines of code that do the following:

1. Show you which routine you are in.
2. Let you get back to the menu if this was not the routine you wanted.
3. Set up any 'on error' conditions.
4. Print a list of file names already stored on disk.

Since the average subroutine length is about 30 lines, 1/3 of the code deals with the interfacing. About 1/2 of the rest of the code deals with the task you are trying to accomplish. The rest of the code does the general housekeeping necessary to the job.

7854 GPIB Facts. The 7854 transfers ASCII bytes over the GPIB. This includes waveforms, text, or results. The rear panel of the 7854 contains the address switches and GPIB bus terminating characters. Figure 3-3 shows the 7854 settings for this software. By pressing the 7854 ID button, the current 7854 settings are printed on screen. To run this software the 7854 is set to TALK LISTEN EOI/LF with a primary bus address of 1.

Changing Points Per Waveform. Although this is explained in the 7854 Operators Manual, a common 7854 programming error occurs when sending numbers to the 7854. For example, to change the points per waveform, the 7854 expects to see a space between each number in the string it receives. To change the 7854 to 1024 points per waveform, the following string is sent "1 0 2 4 >P/W".

SOFTWARE PACKAGE DETAILS

7854 Setup

Turn the 7854 power switch on. Set the rear panel switches (figure 3-3), and connect the GPIB cable.

Program Details

The remaining sections of this note will describe the BASIC program. The main line program will be described first, followed by each of the subroutines. The main line program contains the call routine for the drivers, the setting of the active keys, and the available menu of subroutines. Figure 4-1 is a print out of the main line program.

Fig. 4-1. Main Line Program

```
100 KEY OFF:SCREEN 0:LOCATE 1,1,0
110       CLEAR   ,60000!          ' GPIB-PC Rev. B.1
120       IBINIT1 = 60000!         ' BASICA Declaration File
130       IBINIT2 = IBINIT1 + 3    ' through line 170
140       BLOAD "bib.m",IBINIT1    ' supplied by National Instruments
150 DEF SEG=0:POKE 1047,64:DEF SEG
160       CALL IBINIT1(IBFIND,IBTRG,IBCLR,IBPCT,IBSIC,IBLOC,IBPPC,IBBNA,
            IBONL,IBRSC,IBSRE,IBRSV,IBPAD,IBSAD,IBIST,IBDMA,IBEOS,IBTMO,IBEOT)
170       CALL IBINIT2(IBGTS,IBCAC,IBWAIT,IBPOKE,IBWRT,IBWRTA,IBCMD,
            IBCMDA,IBRD,IBRDA,IBSTOP,IBRPP,IBRSP,IBDIAG,IBXTRC,IBSTA%,IBERR%,
            IBCNT%)
180 CLS:DEV$="DEV0"              ' Looks up 'DEV0' from the Configuration
190 CALL IBFIND (DEV$,DS0%)        ' file when the interface was set up.
200 IF IBSTA%<0 THEN GOTO 710      ' Checks for GPIB error. This is done after
                                 each read or write throughout this software.
210 CALL IBCLR(DS0%)               ' Sends 'clear' to 7854
220 WRT$="STORED"                  ' The 7854 is put
230 CALL IBWRT(DS0%,WRT$)          ' into the 'STORED' display
240 IF IBSTA%<0 THEN GOSUB 710     ' mode.
250 REM THIS IS TO THE MENU--
260 CLS:COLOR 7,0:GOSUB 620
265 LOCATE X-2,Y+18:PRINT " M E N U "                                          "
270 LOCATE X-1,Y:PRINT "                                                       "
280 LOCATE X-1,Y:PRINT "<S>AVE PROGRAM.....save 7854 program on disk           "
290 LOCATE X,Y:PRINT "                                                         "
300 LOCATE X+1,Y:PRINT "<L>OAD PROGRAM.....Load 7854 program from disk         "
310 LOCATE X+2,Y:PRINT "                                                       "
320 LOCATE X+3,Y:PRINT "<R>ECEIVE WAVEFORM.....Saves 7854 waveform             "
330 LOCATE X+4,Y:PRINT "                                                       "
340 LOCATE X+5,Y:PRINT "<T>RANSMIT WAVEFORM.....Send waveform to 7854          "
350 LOCATE X+6,Y:PRINT "                                                       "
360 LOCATE X+7,Y:PRINT "<D>ISPLAY WAVEFORM.....Display saved waveform          "
370 LOCATE X+8,Y:PRINT "                                                       "
380 LOCATE X+9,Y:PRINT "<W>RITE .....Sends text to 7854                        "
390 LOCATE X+10,Y:PRINT "                                                      "
400 LOCATE X+11,Y:PRINT "<C>OMMANDS STRINGS..Sends command string to 7854      "
410 LOCATE X+12,Y:PRINT "                                                      "
420 LOCATE X+13,Y:PRINT "<G>ET X REGISTER...Receive X register information     "
430 LOCATE X+14,Y:PRINT "                                                      "
440 LOCATE X+15,Y:PRINT "<O>UTPUT TO X REGISTER.....7854 X register            "
450 LOCATE X+16,Y:PRINT "                                                      "
460 LOCATE X+17,Y:PRINT "<E>XIT.....Exits to basic                             "
470 LOCATE X+18,Y:PRINT "
480 GOSUB 610      ' This subroutine does a 'poke' to lock on capital letters.
                     Only caps are defined in the 'INKEY$' statements.
490 Q$=INKEY$:IF Q$="" THEN 480
500 IF Q$="S" THEN GOSUB 1000:GOTO 100
510 IF Q$="L" THEN GOSUB 2000:GOTO 100
520 IF Q$="R" THEN GOSUB 3000:GOTO 100
530 IF Q$="T" THEN GOSUB 4000:GOTO 100
540 IF Q$="W" THEN GOSUB 5000:GOTO 250
550 IF Q$="D" THEN GOSUB 6000:GOTO 100
560 IF Q$="C" THEN GOSUB 7000:GOTO 100
570 IF Q$="G" THEN GOSUB 7500:GOTO 100
580 IF Q$="O" THEN GOSUB 8000:GOTO 100
590 IF Q$="E" THEN GOSUB 8500
600 BEEP:GOTO 490
610 DEF SEG=0:POKE 1047,64:DEF SEG:RETURN 'Turn on caps lock
```

Sending 7854 Programs. The two subroutines listed in Figures 4-2 and 4-3 will transfer programs written in the internal 7854 RPN language. The programs are read in the IBM-PC a line at a time. A carriage return (character 13) is added to each line. This is stored in the open file and the next line is brought in. This continues until the entire program is read in. To send the program back to the 7854, the file is opened and each line is read in and sent until the end of the file is found.

Fig. 4-1 (cont)

```
620 X=4:Y=16      ' lines 620 to 700 draws the box for the main menu
630 LOCATE X-2,Y-3:PRINT STRING$(53,205);CHR$(187)
640 FOR I=1 TO 20: LOCATE X-1,Y+50:PRINT CHR$(186):X=X+1:NEXT I
650 LOCATE X-1,Y- STRING$(53,205);CHR$(188)
660 LOCATE X-1,Y-3:PRINT CHR$(200)
670 FOR I=20 TO 1 STEP -1:LOCATE X-2,Y-3:PRINT CHR$(186):X=X-1:NEXT I
680 LOCATE X-2,Y-3:PRINT
700 RETURN
710 S$=HEX$(IBSTA%):PRINT S$:PRINT IBERR%:PRINT IBCNT%    ' Any GPIB error
                                                          is printed on screen
720 IF S$="8100" THEN 740            ' If the 7854 does not respond to the
                    first call, line 740 is executed and the program is ended.
730 RETURN
740 CLS:LOCATE 25,1:PRINT " The 7854 is NOT on line. Check the 7854 rear panel
settings or the GPIB cable."
750 END
```

Fig. 4-2. Store 7854 Programs

```
1000 REM ! ! 7854 programs are transferred and stored on disk
1010 ' lines 1010 to 1090 prints title of routine, lists programs on disk, and
      gives user an option of returning to main menu.
1020 CLS:LOCATE X+8,1:PRINT STRING$(80,205):LOCATE X-1,1:COLOR 0,7:PRINT
STRING$(240,32)
1030 LOCATE X,Y:PRINT "TRANSFER 7854 PROGRAM TO THE DISK":COLOR 7,0
1040 LOCATE X+10,Y:ON ERROR GOTO 9510:FILES "*.PRO":LOCATE X+16,1:PRINT "
 THESE FILES ALREADY EXIST"
1050 LOCATE X+8,Y+10:PRINT " 7854 P R O G R A M FILES "
1060 LOCATE 25,1:PRINT "COMMANDS:  <C>ONTINUE    <M>AIN MENU ";
1070 Q$=INKEY$:IF Q$="" THEN 1070
1080 IF Q$="C" THEN 1110
1090 IF Q$="M" THEN RETURN
1100 GOTO 1070
1110 LOCATE 25,1:PRINT STRING$(79,32);:LOCATE X+18,1:INPUT " INPUT FILE NAME
(without extension):",FILE$
1120 FILE$=FILE$+".PRO"
1130 OPEN FILE$ FOR OUTPUT AS #1
1140 WRT$="EXECUTE 0 GOTO PROGRAM SAVE"        ' sets 7854 to program mode,
1150 CALL IBWRT(DSO%,WRT$)                     ' sets program marker to line
0,
1160 IF IBSTA%<0 THEN GOSUB 710                ' and prepares to send program.
1170 CALL IBRSP(DSO%,STA%):IF STA%<>144 AND STA%<>208 THEN 1170
1180 V%=&HD:CALL IBEOS(DSO%,V%)       'sets terminating character to EOI only.
1190 F$=SPACE$(95)
1200 CALL IBRD(DSO%,F$)                ' read line from 7854
1210 IF IBSTA%<0 THEN GOSUB 710
1220 C$=CHR$(13)                       ' add carrage return to each line
1230 PRINT #1,F$+C$                    ' store line on disk
1240 X$=HEX$(IBSTA%)
1250 IF X$="2900" THEN 1260 ELSE  ' check GPIB status, if done then 1250
1260 CLOSE #1
1270 V%=&H40D:CALL IBEOS(DSO%,V%)      ' change terminator back to original
1280 WRT$="EXECUTE"                    ' take 7854 out of program mode
1290 CALL IBWRT(DSO%,WRT$)
1300 IF IBSTA%<0 THEN GOSUB 710
1310 CALL IBLOC(DSO%)
1320 Q$=""
1330 RETURN
```

Fig. 4-3. Send 7854 Programs

```
2000 REM ! ! 7854 programs are transfer from disk to the 7854
2010 V%=&HD:CALL IBEOS(DSO%,V%)    ' change terminator mode to EOI only
2020 ' lines 2020 to 2100 prints title of routine, lists programs on disk, and
gives user an option of returning to main menu.
2030 CLS:LOCATE X+8,1:PRINT STRING$(80,205):LOCATE X-1,1:COLOR 0,7:PRINT
STRING$(240,32)
2040 LOCATE X,Y:PRINT "TRANSFER 7854 PROGRAM FROM THE DISK":COLOR 7,0
2050 LOCATE X+10,Y:ON ERROR GOTO 9510:FILES "*.PRO":LOCATE X+16,1:PRINT "
THESE FILES HAVE BEEN STORED"
2060 LOCATE X+8,Y+10:PRINT " 7854 P R O G R A M FILES "
2070 LOCATE 25,1:PRINT "COMMANDS:  <C>ONTINUE    <M>AIN MENU ";
2080 Q$=INKEY$:IF Q$="" THEN 2080
2090 IF Q$="C" THEN 2120
2100 IF Q$="M" THEN RETURN
2110 GOTO 2080
2120 WRT$="PROGRAM CLP NEXT"       ' clears any program stored in 7854 memory
2130 CALL IBWRT(DSO%,WRT$)
2140 IF IBSTA%<0 THEN GOSUB 710
2150 LOCATE 25,1:PRINT STRING$(79,32);:LOCATE X+18,1:INPUT " INPUT FILE NAME
(without extension):",FILE$
2160 FILE$=FILE$+".PRO"
2170 OPEN FILE$ FOR INPUT AS #1
2180 INPUT #1,T$                    ' reads in line from disk
2190 IF EOF(1) THEN 2230            ' checks for end of file
2200 CALL IBWRT(DSO%,T$)            ' writes line to 7854
2210 IF IBSTA%<0 THEN GOSUB 710
2220 GOTO 2180                      'gets another line
2230 CLOSE #1
2240 V%=&H40D                       'resets terminator
2250 CALL IBEOS(DSO%,V%)
2260 IF IBSTA%<0 THEN GOSUB 710
2270 WRT$="EXECUTE"                 ' changes 7854 out of program mode
2280 CALL IBWRT(DSO%,WRT$)
2290 IF IBSTA%<0 THEN GOSUB 710
2300 CALL IBCLR(DSO%)
2310 CALL IBLOC(DSO%)
2320 Q$=""
2330 RETURN
```

Waveform Transfers. The two subroutines listed in Figures 4-4 and 4-5 will transfer waveforms to and from the 7854. The 7854 sends both preamble and curve (waveform) data. These are stored in two separate files (a .pre and a .wfm file). Once the transfer of curve data is started, the transfer will continue until the 7854 sends a SRQ saying that it is done. When sending the waveform back to the 7854, the end of the file is used to terminate the transfer.

Fig. 4-4. Storing Waveforms Subroutine

```
3000 REM ! ! Gets waveforms from 7854 and stores them on disk
3010 ' lines 3010 to 3100 prints title of routine, lists programs on disk, and
gives user an option of returning to main menu.
3020 CALL IBCLR(DSO%)
3030 CLS:LOCATE X+8,1:PRINT STRING$(80,205):LOCATE X-1,1:COLOR 0,7:PRINT
STRING$(240,32)
3040 LOCATE X,Y:PRINT "TRANSFER 7854 WAVEFORM TO THE DISK":COLOR 7,0
3050 LOCATE X+10,Y:ON ERROR GOTO 9510:FILES "*.WFM":LOCATE X+16,1:PRINT "
 THESE WAVEFORMS EXIST ON DISK"
3060 LOCATE X+8,Y+10:PRINT " 7854 W A V E F O R M FILES "
3070 LOCATE 25,1:PRINT "COMMANDS:  <C>ONTINUE    <M>AIN MENU ";
3080 Q$=INKEY$:IF Q$="" THEN 3080
3090 IF Q$="C" THEN 3120
3100 IF Q$="M" THEN RETURN
3110 GOTO 3080
3120 LOCATE 25,1:PRINT STRING$(79,32);:LOCATE X+18,1:INPUT " INPUT FILE NAME
(without extension):",FILE$
3130 CLS:LOCATE 15,10 :PRINT "TRANSFERRING WAVEFORM . . . . . "
3140 OPEN FILE$+".PRE" FOR OUTPUT AS #1
3150 OPEN FILE$+".WFM" FOR OUTPUT AS #2
3160 STA%=0
3170 I=1
3180 WRT$="0 WFM SENDX"              ' 7854 is asked to send 0 waveform
3190 CALL IBWRT(DSO%,WRT$)
3200 IF IBSTA%<0 THEN GOSUB 710
3210 RD$=SPACE$(95)
3220 CALL IBRD(DSO%,RD$)             ' the preamble is read first
3230 IF IBSTA%<0 THEN GOSUB 710
3240 PRINT #1,RD$;                   ' preamble is stored on disk (.pre file)
3250 CALL IBRD(DSO%,RD$)             ' waveform is then read in
3260 IF IBSTA%<0 THEN GOSUB 710
3270 PRINT #2,RD$;                   ' waveform is stored on disk (.wfm file)
3280 CALL IBRSP(DSO%,STA%):IF STA%=2 THEN 3320
3290 GOTO 3250                       ' if 7854 is not done then 3240
3300 ' user is given the option of getting another waveform or returning to
main menu
3310 CLS:LOCATE 10,10:PRINT "TRANSFER COMPLETE"
3320 LOCATE 25,1:PRINT "COMMANDS: <R>EPEAT  <M>ENU ";
3330 GOSUB 3380
3340 Q$=INKEY$:IF Q$="" THEN 3330
3350 IF Q$="R" THEN GOTO 3390
3360 IF Q$="M" THEN GOTO 3400
3370 GOTO 3340
3380 DEF SEG=0:POKE 1047,64:DEF SEG:RETURN
3390 CLOSE #1,#2:Q$="":GOTO 3000
3400 CLOSE #1,#2:Q$="":CALL IBLOC(DSO%):RETURN
```

Fig. 4-5. Sending 7854 Waveforms Subroutine

```
4000 REM ! ! Gets waveform from disk and transfers them to 7854
4010 ' lines 4010 to 4100 prints title of routine, lists programs on disk, and
gives user an option of returning to main menu.
4020 CLS
4030 LOCATE X+8,1:PRINT STRING$(80,205):LOCATE X-1,1:COLOR 0,7:PRINT
STRING$(240,32)
4040 LOCATE X,Y:PRINT "TRANSFER WAVEFORM FROM DISK TO 7854":COLOR 7,0
4050 LOCATE X+10,Y:ON ERROR GOTO 9510:FILES "*.WFM":LOCATE X+16,1:PRINT "
 THESE WAVEFORMS EXIST ON DISK"
```

Fig. 4-5 (cont)

```
4060 LOCATE X+8,Y+10:PRINT " 7854 W A V E F O R M FILES "
4070 LOCATE 25,1:PRINT "COMMANDS:  <   <M>AIN MENU ";
4080 Q$=INKEY$:IF Q$="" THEN 4080
4090 IF Q$="C" THEN 4120
4100 IF Q$="M" THEN RETURN
4110 GOTO 4080
4120 LOCATE 25,1:PRINT STRING$(79,32);:LOCATE X+18,1:INPUT " INPUT FILE NAME
(without extension):",FILE$
4130 CLS:LOCATE 16,10:PRINT "LOADING WAVEFORM FROM DISK . . . . . ."
4140 OPEN FILE$+".PRE" FOR INPUT AS #1
4150 LINE INPUT #1,PRE$          ' read in preamble from disk
4160 CLOSE #1
4170 L=VAL(MID$(PRE$,INSTR(1,PRE$,"NR.PT:")+6,9))  ' find out number of points
4180 OPEN FILE$+".WFM" FOR INPUT AS #2
4190 DIM W$(L):I=0               ' dimension string to number of points
4200 I=I+1      'lines 4210 to 4260 put waveform into a string array
4210 IF EOF(2) THEN 4320
4220 INPUT #2,WFM$
4230 IF I=L THEN 4260
4240 W$(I)=WFM$+CHR$(44)
4250 GOTO 4200
4260 L$=STR$(L)              ' changes number of points to a string
4270 A=LEN(L$)               '
4280 FOR J=1 TO A            ' puts a space between each value in the string
4290 LL$=LL$+LEFT$(L$,1)+CHR$(32):L$=RIGHT$(L$,A-J)
4300 NEXT J
4310 LL$=LL$+" >P/W":CALL IBWRT(DSO%,LL$):IF IBSTA%<0 THEN GOSUB 700  ' the
7854 is set to the proper points per waveform
4320 CLS:LOCATE 15,10:PRINT "TRANSFERRING WAVEFORM TO 7854 . . . . . ."
4330 W$(I)=WFM$:WRT$="0 WFM READX"  'tell 7854 to read waveform into 0
waveform memory
4340 CALL IBWRT(DSO%,WRT$)
4350 CALL IBRSP(DSO%,STA%):IF STA%<>147 AND STA%<>211 THEN 4350 'is 7854 ready
to read waveform?
4360 IF IBSTA%<0 THEN GOSUB 710
4370 V%=0:CALL IBEOT(DSO%,V%)      ' dont terminate on EOI
4380 CALL IBWRT(DSO%,PRE$)         ' write preamble to 7854
4390 V%=1
4400 FOR A=1 TO L
4410 IF A=I  THEN CALL IBEOT(DSO%,V%) ' if done then terminate transfer
4420 CALL IBWRT(DSO%,W$(A))        ' write waveform to 7854
4430 IF IBSTA%<0 THEN GOSUB 710
4440 NEXT A
4450 CLS:LOCATE 10,10:PRINT "TRANSFER COMPLETE"
4460 'user has option of sending another waveform to 7854
4470 LOCATE 25,1:PRINT STRING$(79,32);
4480 LOCATE 25,1:PRINT " COMMAND:   <R>EPEAT   <M>ENU ";
4490 GOSUB 4540
4500 Q$=INKEY$:IF Q$="" THEN 4490
4510 IF Q$="R" THEN GOTO 4550
4520 IF Q$="M" THEN GOTO 4560
4530 GOTO 4500
4540 DEF SEG=0:POKE 1047,64:DEF SEG:RETURN
4550 CLOSE #1,#2:ERASE W$:GOTO 4000
4560 CLOSE #1,#2:V%=&H40D:CALL IBEOS(DSO%,V%)
4570 CALL IBLOC(DSO%)
4580 Q$="":ERASE W$
4590 RETURN
```

Writing Text to 7854. This subroutine will transfer text to the 7854 to be displayed on the 7854 CRT. Figure 4-6 is the program listing for this subroutine. A box about 40 characters wide is generated on the screen. This is the number of characters the 7854 will print on each line. The 7854 will accept 12 lines of text. The software numbers each line for your information.

Sending Commands to 7854. This subroutine will allow you to send commands to the 7854. These commands are the usual 7854 commands found in the 7854 operators manual. It is up to you to provide any signal required by the 7854. However, you must use caution in choosing the commands to send. Figure 4-5 lists this subroutine.

Fig. 4-6. Send Text Subroutine

```
5000 REM ! ! Input text on IBM to be transfered to 7854
5010 'user can transfer text into lines 2 through 14 of the 7854
5020 CLS:GOSUB 620:LOCATE X-2,Y+10:PRINT " TEXT T R A N S F E R ":LOCATE X,Y+1
5030 V%=&HD:CALL IBEOS(DSO%,V%)
5040 PRINT "ENTER THE TEXT TO BE TRANSFERRED, EACH LINE"
5050 LOCATE X+1,Y+1:PRINT "NUMBER WILL BE PRINTED. ENTER 'XMT' TO TRANSFER"
5060 LOCATE X+2,Y+1:PRINT "TEXT. A <RETURN> WILL ENTER A BLANK LINE."
5070 C$=CHR$(13)
5080 FOR I=1 TO 12
5090 LOCATE X+(I+4),Y-1:PRINT I:LOCATE X+(I+4),Y+2:LINE INPUT L$
5100 IF L$="XMT" THEN 5140
5110 T$=T$+L$
5120 T$=T$+C$
5130 NEXT I
5140 WRT$="EXECUTE  >TEXT"
5150 CALL IBWRT(DSO%,WRT$)
5160 IF IBSTA%<0 THEN GOSUB 710
5170 CALL IBRSP(DSO%,STA%):IF STA%<>149 AND STA%<>213 THEN 5170
5180 CALL IBWRT(DSO%,T$)
5190 IF IBSTA%<0 THEN GOSUB 710
5200 WRT$="EXECUTE":CALL IBWRT(DSO%,WRT$)
5210 IF IBSTA%<0 THEN GOSUB 710
5220 CALL IBLOC(DSO%)
5230 Q$="":RETURN
```

Fig. 4-7. Send Commands Subroutine

```
7000 REM ! ! 7854 commands are inputed & sent to 7854
7010 ' lines 7010 to 7080 prints title of routine, lists programs on disk, and
gives user an option of returning to main menu.
7020 CLS:LOCATE X-3,1:COLOR 0,7:PRINT STRING$(240,32)
7030 LOCATE X-2,Y:PRINT "SENDING COMMANDS TO THE 7854    ":COLOR 7,0
7040 LOCATE 25,1:PRINT STRING$(79,32);:LOCATE 25,1
7050 PRINT " COMMANDS:   <C>ONTINUE  <M>AIN MENU ";
7060 Q$=INKEY$:IF Q$="" THEN 7060
7070 IF Q$="C" THEN 7100
7080 IF Q$="M" THEN 7140
7090 GOTO 7060
7100 CLS:LOCATE 10,10:INPUT "INPUT COMMAND TO BE SENT:",WRT$
7110 CALL IBWRT(DSO%,WRT$)
7120 LOCATE 15,10:INPUT "DO YOU WISH TO SEND MORE COMMANDS";ANS$
7130 IF ANS$="Y" THEN GOTO 7100
7140 CALL IBLOC(DSO%)
7150 Q$="":RETURN
```

X Register Transfers. Figures 4-8 and 4-9 list subroutines to read and write to the 7854 X Register. Because of the waveform processing power of the 7854, transferring the answers generated by the 7854 is important. These subroutines allow you to transfer the results from the X Register and print them out on the screen and to transfer values to the X Register.

Fig. 4-8. Getting X Register Contents

```
7500 REM !!! The X register is read and transfered to IBM PC
7510 ' lines 8010 to 8090 prints title of routine, lists programs on disk, and
gives user an option of returning to main menu.
7520 V%=&HD:CALL IBEOS(DSO%,V%)
7530 CLS:LOCATE X-3,1:COLOR 0,7:PRINT STRING$(240,32)
7540 LOCATE X-2,Y:PRINT "GET X REGISTER FROM THE 7854    ":COLOR 7,0
7550 LOCATE 25,1
7560 PRINT " COMMANDS:   <C>ONTINUE  <M>ENU ";
7570 Q$=INKEY$:IF Q$="" THEN 7570
7580 IF Q$="C" THEN 7610
7590 IF Q$="M" THEN 7780
7600 GOTO 7570
7610 Q$="":CLS:LOCATE 10,10:PRINT "THIS WILL RECEIVE THE CONTENTS OF THE X
REGISTER"
7620 CALL IBLOC(DSO%)
7630 LOCATE 12,10:PRINT "YOU MAY SETUP THE X REGISTER AS NEEDED"
7640 LOCATE 14,10:INPUT "HIT 'ENTER' TO CONTINUE",ANS$
7650 S$=SPACE$(40)
7660 WRT$="SENDX"
7670 CALL IBWRT(DSO%,WRT$)
7680 IF IBSTA%<0 THEN GOSUB 710
7690 CALL IBRSP(DSO%,STA%):IF STA%<>146 AND STA%<>210 THEN 7690
7700 CALL IBRD(DSO%,S$)
7710 IF IBSTA%<1 THEN GOSUB 710
7720 LOCATE 20,10:PRINT S$
7730 LOCATE 25,1:PRINT "COMMAND: <R>EPEAT   <M>ENU                    ";
7740 Q$=INKEY$:IF Q$="" THEN 7740
7750 IF Q$="R" THEN 7610
7760 IF Q$="M" THEN 7780
7770 GOTO 7740
7780 Q$="":CALL IBLOC(DSO%)
7790 V%=&H40D
7800 CALL IBEOS(DSO%,V%)
7810 RETURN
```

Graphing Waveforms from Diskette. This subroutine will graph waveforms previously transferred from the 7854. The program checks to see if the graphics board is installed. If the graphics board is not installed, an error message is generated and the main menu is displayed. Figure 4-10 lists this subroutine.

Fig. 4-9. Send Value to X Register

```
8000 REM !!! Numeric data is transfered to the X register
8010 Q$="":CLS:LOCATE X-2,1:COLOR 0,7:PRINT STRING$(240,32)
8020 LOCATE X-1,Y:PRINT "SEND INFORMATION TO X REGISTER    ":COLOR 7,0
8030 LOCATE 25,1
8040 PRINT " COMMANDS:   <C>ONTINUE  <M>ENU ";
8050 Q$=INKEY$:IF Q$="" THEN 8050
8060 IF Q$="C" THEN 8090
8070 IF Q$="M" THEN 8220
8080 GOTO 8050
8090 SS$=SPACE$(40):STA%=0
8100 Q$="":CLS:LOCATE 10,10:INPUT "INPUT THE VALUE TO SEND TO THE X
REGISTER",S$:SS$=S$+" ENTER"
8110 WRT$="READX"
8120 CALL IBWRT(DSO%,WRT$)
8130 IF IBSTA%<1 THEN GOSUB 710
8140 CALL IBRSP(DSO%,STA%):IF STA%<>147 AND STA%<>211 THEN 8130
8150 CALL IBWRT(DSO%,SS$)
8160 IF IBSTA%<1 THEN GOSUB 710
8170 LOCATE 25,1:PRINT "COMMAND: <S>END   <M>ENU                    ";
8180 Q$=INKEY$:IF Q$="" THEN 8180
8190 IF Q$="S" THEN CALL IBLOC(DSO%):GOTO 8090
8200 IF Q$="M" THEN 8220
8210 GOTO 8180
8220 Q$="":CALL IBLOC(DSO%)
8230 RETURN
```

Fig. 4-10. Graphing Stored Waveforms

```
6000 REM ! ! This routine gets a waveform from disk and graphs it on the
IBM PC
6010 ' lines 6010 to 6040 check for graphics capabilities
6020 DEF SEG=0
6030 IF (PEEK(&H410) AND &H30) <>&H30 THEN GMA=1
6040 DEF SEG:I=0
6050 IF GMA=1 THEN 6080
6060 PRINT "GRAPHIC MONITOR NOT INSTALLED; ABORTING COMMAND":FOR C=1 TO
1000:NEXT C:GOTO 6620
6070 ' lines 6060 to 6140 prints title of routine, lists programs on disk,
and gives user an option of returning to main menu.
6080 CLS:LOCATE X+8,1:PRINT STRING$(80,205):LOCATE X-1,1:COLOR 0,7:PRINT
STRING$(240,32):I=0
6090 LOCATE X,Y+5:PRINT "GRAPH 7854 WAVEFORM FROM THE DISK":COLOR 7,0
6100 LOCATE X+10,Y:ON ERROR GOTO 9510:FILES "*.WFM":LOCATE X+16,1:PRINT "
 THESE WAVEFORMS ARE STORED ON DISK"
6110 LOCATE X+8,Y+8:PRINT " S E L E C T  W A V E F O R M "
6120 LOCATE 25,1:PRINT "COMMANDS:  <C>ONTINUE    <M>AIN MENU ";
6130 Q$=INKEY$:IF Q$="" THEN 6130
6140 IF Q$="C" THEN 6170
6150 IF Q$="M" THEN RETURN
6160 GOTO 6130
6170 LOCATE 25,1:PRINT STRING$(79,32);:LOCATE X+18,1:INPUT "SELECT WAVFORM
FILE FOR GRAPH (without extension):",FILE$
6180 IF LEN (FILE$)<1 THEN BEEP:GOTO 6170
6190 OPEN FILE$+".PRE" FOR INPUT AS #1
6200 OPEN FILE$+".WFM" FOR INPUT AS #2
6210 CLS:LOCATE X+5,10:PRINT "LOADING WAVEFORM . . . . . "
6220 LINE INPUT #1,PRE$        ' input preamble from disk
```

Fig. 4-10 (cont)

```
6230 PW$="NR.PT:"
6240 Y=INSTR(1,PRE$,PW$)+6
6250 X=VAL(MID$(PRE$,Y,9))      ' find out the number of points
6260 DIM W(X)                   ' dimension w to number of points
6270 V=VAL(MID$(PRE$,INSTR(30,PRE$,"XINCR:")+6,9)):V=(V*X)/10  ' find time
per point
6280 C=VAL(MID$(PRE$,INSTR(30,PRE$,"YZERO:")+6,8))   ' find any vertical
offset
6290 Y=VAL(MID$(PRE$,INSTR(50,PRE$,"YMULT:")+6,9))   ' find vertical scale
factor
6300 IF EOF(2) THEN 6360          ' check for end of file
6310 I=I+1
6320 INPUT #2,WFM$                ' input a waveform point
6330 IF I=X THEN 6360
6340 W(I)=(VAL(WFM$)+C)*10        ' add offset, adjust value for 10 div,
store in an array
6350 GOTO 6300
6360 SCALE=3
6370 B=512/I
6380 COLOR 7,0:GOSUB 6880         ' generate graticule
6390 DRAW "S=SCALE;BM40,20;XTIC$;XTTIC$;"  'draw then graticule
6400 FOR J=1 TO X
6410 YPNT=75-W(J)*2*(SCALE/4):HPNT=40+J*(SCALE/4.16)*B
6420 DRAW "S=SCALE;M=HPNT;,=YPNT;"    ' plot the waveform
6430 NEXT J
6440 ' next three lines set t/div to approprate settings
6450 IF V<.000001 THEN V=V/1E-09:TDIV$="nSec/DIV"
6460 IF V<.001 THEN V=V/.000001:TDIV$="uSec/DIV"
6470 IF V<1 THEN V=V/.001:TDIV$="Sec/DIV"
6480 LOCATE 20,15:PRINT USING "HORIZONTAL SCALE= ###";V:LOCATE 20,40:PRINT
TDIV$
6490 LOCATE 10,55:PRINT "VERTICAL SCALE"
6500 LOCATE 11,55:PRINT "=";Y;"VOLTS/DIV"
6510 LOCATE 1,20:PRINT "Waveform name: ";FILE$;
6520 ERASE W
6530 ' user has option of plotting new waveform, printing this waveform to
a line printer or returning to main menu.
6540 LOCATE 25,1:PRINT "COMMANDS:  <P>RINT   <N>EW WAVEFORM    <M>ENU";
6550 GOSUB 6610
6560 Q$=INKEY$:IF Q$="" THEN 6550
6570 IF Q$="N" THEN GOTO 6620
6580 IF Q$="P" THEN GOTO 6670
6590 IF Q$="M" THEN GOTO 6630
6600 GOTO 6540
6610 DEF SEG=0:POKE 1047,64:DEF SEG:RETURN
6620 X=5:Y=16:CLOSE #1,#2:SCREEN 0,0,0:GOTO 6080
6630 Q$="":CLOSE #1,#2
6640 SCREEN 0,0,0
6650 X=5:Y=16:RETURN
6660 ' this routine will do a screen dump to a line printer
6670 LOCATE 1,1:PRINT DATE$:LOCATE 1,65:PRINT TIME$
6680 LOCATE 25,1:PRINT "Position paper to top of printer page. Printer
ready (Y/N) ?";
6690 Q$=INKEY$:IF Q$="" THEN 6690
6700 IF Q$="y" OR Q$="Y" THEN 6770
6710 GOTO 6690
6720 LOCATE 25,1:PRINT "ANOTHER PRINT-OUT (Y/N) ?";
6730 Q$=INKEY$:IF Q$="" THEN 6730
6740 IF Q$="Y" OR Q$="y" THEN 6770
6750 IF Q$="N" OR Q$="n" THEN 6540
6760 GOTO 6730
```

Ending the Program. This subroutine ends the program and returns to BASIC. All loose ends are taken care of and the interface is closed. Figure 4-11 lists this subroutine.

SUMMARY

By using the National interface card for the IBM PC, you can 'talk' to the 7854. You can transfer programs and waveforms to and from the 7854, write text on the 7854 CRT, and graph the stored waveforms, all with your IBM PC.

Roger Ensrud
Applications Engineer
Lab Instrument Division

Additional Information. Tektronix offers a software package to perform all the functions described in this document. For a fee, you can receive a diskette containing the 7854/IBM PC programs.

For further information or the address and phone number of the Tektronix sales office nearest you, call:
(800) 547-1512
In Oregon: (800) 452-1877

Fig. 4-10 (cont)

```
6770 LOCATE 25,1:PRINT STRING$(79,32);:SCREEN 2
6780 DEFINT A-Z:OPTION BASE 1:P=0:I=0:J=0:DIM ARRAY(3)
6790 DATA &H55          :REM 55H PUSH BP
6800 DATA &HCD, &H05    :REM CD05H INT 5
6810 DATA &H5D          :REM 5DH POP BP
6820 DATA &HCB          :REM 90H NOP
6830 P=VARPTR(ARRAY(1)):FOR I=0 TO 4:READ J:POKE(P+I),J:NEXT I
6840 DEF USR0=VARPTR (ARRAY(1))
6850 A=USR0(X)
6860 LPRINT CHR$(12)
6870 ERASE ARRAY:GOTO 6720
6880 REM GRATICULE IS DEFINED HERE
6890 CLS:SCREEN 2,,0,0
6900 HTIC$="D20;R5 L5;D20;R5 L5;D20;R5 L5;D20;R50 L50;D20;R5 L5;D20;R5
L5;D20;R5 L5;D20;"
6910 VTIC$="R50;U5 D5;R50;U5 D5;R50;U5 D5;R50;U5 D5;R50;U50 D50;R50;U5
D5;R50;U5 D5;R50;U5 D5;R50;U5 D5;R50;"
6920 TIC$=HTIC$+VTIC$
6930 HHTIC$="U20;L5 R5;U20;L5 R5;U20;L5 R5;U20;L50 R50;U20;L5 R5;U20;L5
R5;U20;L5 R5;U20;"
6940 VVTIC$="L50;D5 U5;L50;D5 U5;L50;D5 U5;L50;D5 U5;L50;D50 U50;L50;D5
U5;L50;D5 U5;L50;D5 U5;L50;D5 U5;L50;"
6950 TTIC$=HHTIC$+VVTIC$
6960 RETURN
```

Fig. 4-11. End Program

```
8500 REM ! ! EXIT ROUTINE
8510 CALL IBLOC(DSO%)
8520 V%=0:CALL IBONL(DEV$,V%)
8530 CLS:CALL IBLOC(DSO%):KEY ON:END
9000 REM ERROR HANDLING ROUTINE
9010 GOSUB 710
9020 CALL IBLOC(DSO%)
9030 RESUME 250
9500 REM !!! MORE ERROR CHECKING ROUTINES
9510 IF ERR=53 THEN PRINT "NO FILES EXIST WITH THIS EXTENSION":RESUME NEXT
9520 IF ERR=52 THEN CLOSE #1,#2:LOCATE X+16,1:PRINT "INVALID FILE NAME
":RESUME 4030
9530 IF ERR=62 THEN CLOSE #1,#2:ERASE W$:LOCATE X+16,1:PRINT "UNABLE TO READ
THIS WAVEFORM FILE ":RESUME 4030
9540 PRINT ERR
9550 PRINT STA%
```

For further information, contact:

U.S.A., Asia, Australia, Central & South America, Japan
Tektronix, Inc.
P.O. Box 1700
Beaverton, Oregon 97075
For additional literature, or the address and phone number of the Tektronix Sales Office nearest you, contact:
Phone: (800) 547-1512
Oregon only: (800) 452-1877
TWX: (910) 467-8708
TLX: 151754
Cable: TEKWSGT

Europe, Africa, Middle East
Tektronix Europe B.V.
European Headquarters
Postbox 827
1180 AV Amstelveen
The Netherlands
Phone: (20) 471146
Telex: 18312 - 18328

Canada
Tektronix Canada Inc.
P.O. Box 6500
Barrie, Ontario L4M 4V3
Phone: (705) 737-2700

Tektronix sales and service offices around the world:
Albania, Algeria, Argentina, Australia, Austria, Bangladesh, Belgium, Bolivia, Brazil, Bulgaria, Canada, Peoples Republic of China, Chile, Colombia, Costa Rica, Czechoslovakia, Denmark, East Africa, Ecuador, Egypt, Federal Republic of Germany, Fiji AWA New Zealand, Finland, France, Greece, Hong Kong, Hungary, Iceland, India, Indonesia, Ireland, Israel, Italy, Japan, Jordan, Korea, Kuwait, Lebanon, Malaysia, Mexico, The Netherlands, New Zealand, Nigeria, Norway, Pakistan, Panama, Peru, Philippines, Poland, Portugal, Qatar, Republic of South Africa, Romania, Saudi Arabia, Singapore, Spain, Sri Lanka, Sudan, Sweden, Switzerland, Syria, Taiwan, Thailand, Turkey, Tunisia, United Arab Emirates, United Kingdom, Uruguay, USSR, Venezuela, Yugoslavia, Zambia, Zimbabwe.

Tektronix®
COMMITTED TO EXCELLENCE

4/85 10M

42W-5802