HEWLETT
PACKARD

HP 75000 Model D20 Programming Note

Running D20 SCPI files using PC-DOS
- **HP IBASIC**
- **HP ITG II**
- **Microsoft QuickBASIC**
- **Microsoft QuickC**

Introduction

The HP 75000 Model D20 test system is designed to provide digital capability for mixed signal functional testing applications. The process of developing the digital content of the test is very straightforward. First, the test is created and debugged for correct operation on the Model D20 hardware. The test is then migrated to the final test system, which is often located on the factory floor.

Task Overview

The most efficient method of getting the final test up and running implements the HP E1496A digital test development software. This application software allows one to develop the digital test and create a command file for the D20 hardware. This command file will contain the SCPI (Standard Commands for Programmable Instruments) commands for the Model D20 hardware which can be easily integrated into the final test system. With this type of development structure, the type of computer used in the final test stand is not an issue because the SCPI data file can easily be moved to other computers as an ASCII file. This programming note documents the process of moving the D20 SCPI file when the final test system contains a PC-DOS computer. This note contains examples for downloading and running the SCPI file using HP's IBASIC for Windows, HP's Interactive Test Generator II, Microsoft's QuickBASIC and Microsoft's QuickC.

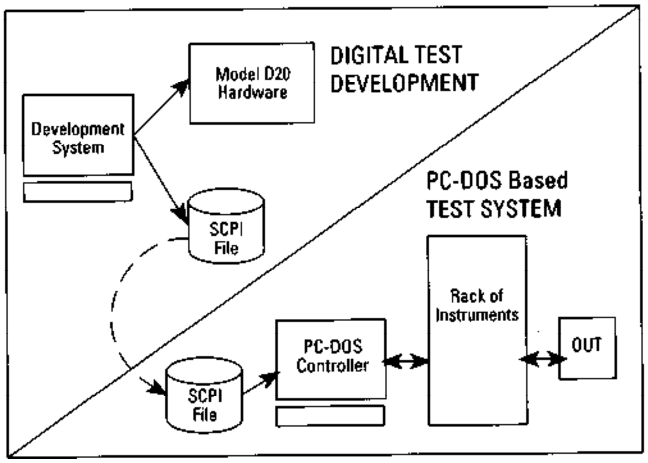


Figure 1. D20 Test Development Process

Step One: Creating the Test

The HP E1496A digital test development software provides a graphical, menu-driven environment for fast test development. This aids in the creation of the digital portion of the test program. The development software is used with or without the Model D20 hardware to develop digital tests. The Model D20 hardware controlled by the development software allows for on-line debugging of the digital test system.

The HP E1496A development software, which requires an HP 9000 Series 300/400 HP-UX workstation, eliminates the time required to learn and understand the commands, syntax and structure of the D20 programming language. The graphical programming environment allows you to program the D20 components based on the operation of the device to be tested. Tests can be developed and debugged for use with the D20 hardware. Debugging requires the D20 hardware (except for rules check) and provides capabilities such as stepping through the test one vector at a time, direct input or output control through groups of pins, etc. This process is documented in the HP 75000 Model D20 manuals. This note will not repeat this process, but rather assumes the user has created the tests using the process outlined in the system documentation.

Step Two: Creating the SCPI file

After a test or tests are created, the HP E1496A software can produce an output file consisting of the SCPI commands required for those tests. These commands can be executed later without the development software actually being present on the computer system.

Within the development software, use the File pulldown menu to select an operation for controlling the flow of information between the development system computer and the HP E1496A development software. The SCPI... menu selection allows the user to save the current D20 test as a SCPI command file.

Migrating the SCPI file from the HP-UX workstation to the PC-DOS environment is the next task. Save the complete D20 test as an ASCII SCPI file. ASCII files are stored with a .asc extension, a format which is handled more easily by PC application software. For example, if the file name is "testfile", the complete SCPI test file would be "testfile.asc".

Once the SCPI command file is saved on the development system, the file is now an HP-UX ASCII file. Standard HP-UX ASCII files and DOS ASCII files differ in several respects, but the main differences are in the area of end-of-line and end-of-file specifiers. HP-UX revision 7.0 and greater provides several DOS utilities to aid in the file conversion process (note that the HP E1496A development software requires HP-UX 7.0 or greater). The HP-UX user command for converting a file from HP-UX format to DOS format is **ux2dos**. For example, to convert the SCPI command file "testfile.asc" to DOS format, you would enter the command:

```
ux2dos testfile.asc > testfile.dat
```

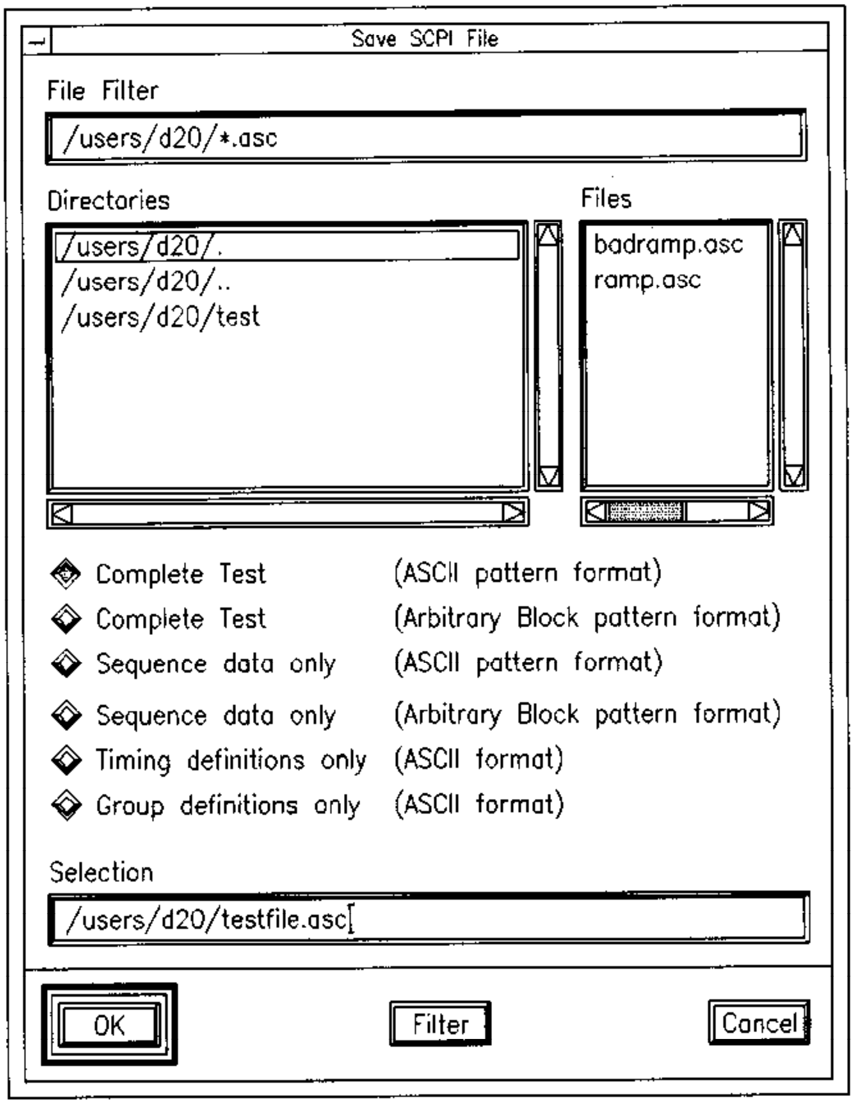
Executing this command converts the file "testfile.asc" to DOS format and saves it as "testfile.dat" without altering its contents.

Now that the file is in DOS format, it must be moved to the PC-DOS computer. The file can be transferred easily if both the HP-UX development system and the PC-DOS computer are networked together via LAN. In this case, you would use one of a variety of communication tools such as ftp (file transfer protocol) to copy the file from the HP-UX development system to the PC.

Another method involves using a floppy disk to transfer the file. In this case, both the HP-UX development computer and the PC-DOS computer must have the same type of floppy disk drives available (e.g., a 3.5" floppy drive). Since the file must be copied onto the DOS computer, the floppy disk must be a DOS formatted disk. The HP-UX user command **doscp** is used to copy the DOS file to the DOS formatted floppy disk using a command such as:

```
doscp testfile.dat /dev/dsk/fdisk:/testfile.dat
```

This command copies the file "testfile.dat" to the same name on the floppy disk that has the device driver of /dev/dsk/fdisk. The device driver is a special file HP-UX uses for communication with peripheral devices. Refer to the *HP-UX System Administration Manual* or consult your system administrator for more information.



/users/d20/*.asc

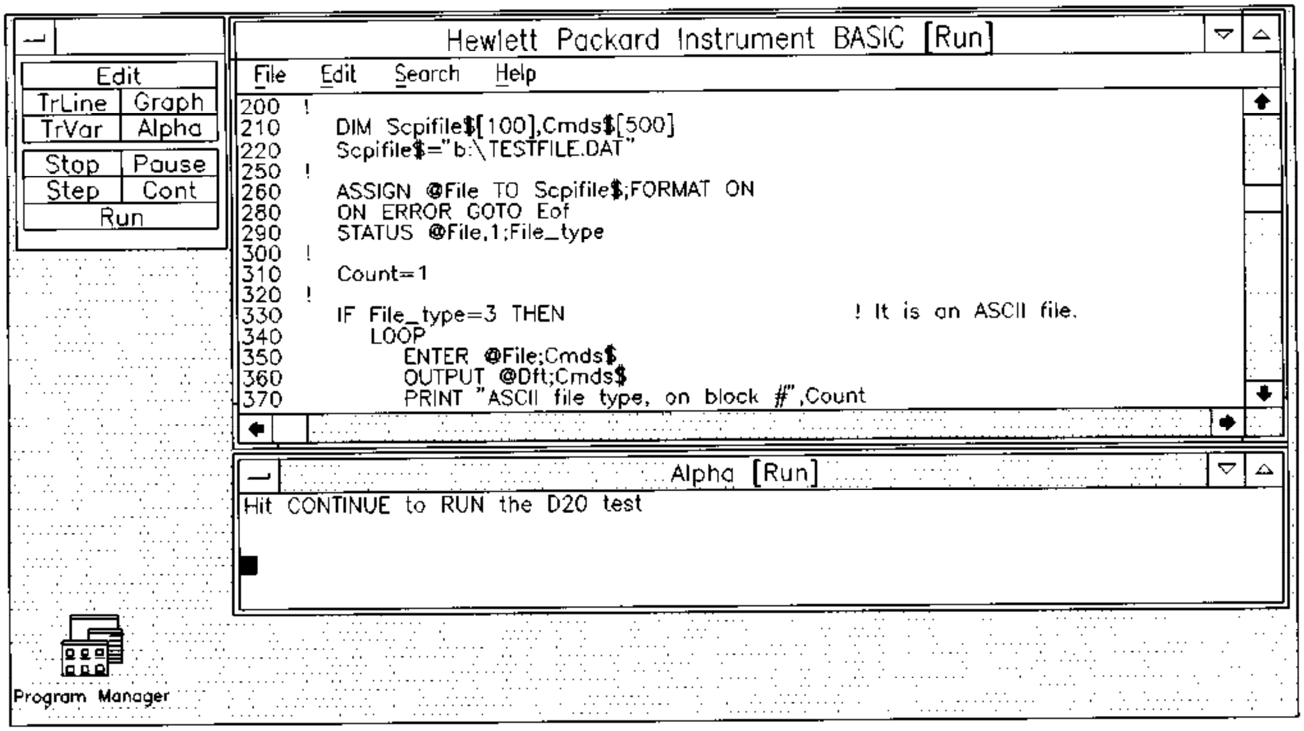
/users/d20/.
/users/d20/..
/users/d20/test

badramp.asc
ramp.asc

◆ Complete Test (ASCII pattern format)
◇ Complete Test (Arbitrary Block pattern format)
◇ Sequence data only (ASCII pattern format)
◇ Sequence data only (Arbitrary Block pattern format)
◇ Timing definitions only (ASCII format)
◇ Group definitions only (ASCII format)

/users/d20/testfile.asc

Figure 2. SCPI File Dialog Box



```
200  !
210     DIM Scpifile$[100],Cmds$[500]
220     Scpifile$="b:\TESTFILE.DAT"
250  !
260     ASSIGN @File TO Scpifile$;FORMAT ON
280     ON ERROR GOTO Eof
290     STATUS @File,1;File_type
300  !
310     Count=1
320  !
330     IF File_type=3 THEN                        ! It is an ASCII file.
340        LOOP
350           ENTER @File;Cmds$
360           OUTPUT @Dft;Cmds$
370           PRINT "ASCII file type, on block #",Count
```

```
Hit CONTINUE to RUN the D20 test
```

Figure 3. HP Instrument BASIC for Windows Development Environment

Once the file has been copied onto the floppy disk, it is now moved to the PC-DOS computer. The file is then extracted from the floppy disk or copied to the hard disk for use with a PC software application.

Step Three: Connecting the Model D20 to the PC

Next the Model D20 system must be connected to the PC. The VXI system should be composed of the HP Model D20 modules and an HP E1405 Command Module. The HP Command Module provides the HP-IB (IEEE-488) to VXI interface for the system as well as the ability to use the SCPI driver for the Model D20 register based devices. Additional details on setting up and configuring the VXI system can be found in the *HP C-size VXIbus Systems Installation and Getting Started Guide* and the Model D20 Hardware Installation Guide.

On the PC side, an HP 82335A HP-IB interface card connects the PC to the HP Command Module via an HP-IB cable. The HP 82335A interface card is bundled with software which allows the card to be programmed from several PC applications.

Step Four: Downloading and Running the test from the PC

The PC-DOS computer has many different programming alternatives and applications available for the test system. All of the following PC applications used the HP 82335A HP-IB interface and command library as the HP-IB interface to the D20 test system. Complete example programs are located in the tables at the end of this programming note.

HP IBASIC for Windows

One option for downloading the D20 SCPI file is to use HP's Instrument BASIC (IBASIC) for Windows. This Windows application provides an editor window for creating and modifying the program used for communicating with instruments. Using HP IBASIC for Windows, the user can write a series of programs for the final test system or incorporate many functions into one program.

A good procedure to follow when downloading D20 SCPI files is to first place the Model D20 hardware in a known state. This is accomplished using a series of commands directed to the D20 modules as shown in Table 1.

```
50  ASSIGN @Dft TO 70917.                    !D20 HP-IB Address

150 OUTPUT @Dft;"*RST"                       !Reset Hardware
160 OUTPUT @Dft;"DIG:GRO:DEL:ALL"            !Delete all pin groups
170 OUTPUT @Dft;"DIG:SEQ:DEL:ALL"            !Delete all sequences
180 OUTPUT @Dft;"DIG:TIM:CYCL:DEL:ALL"       !Delete timing cycles
190 OUTPUT @Dft;"FORM:DATA ASC"              !Response data in ASCII
```

Table 1. D20 Setup Routine in HP IBASIC

Next, the D20 SCPI file is brought into the HP IBASIC program and downloaded to the Model D20 hardware. Accomplish this using a file ASSIGN statement which brings in the SCPI file line by line. As illustrated in Table 2, the HP IBASIC program determines the format of the D20 SCPI file and uses the appropriate routine to ENTER the SCPI file and OUTPUT it to the Model D20 hardware. Using this procedure, a D20 SCPI file saved in either ASCII or Arbitrary Block Format (ABF) can be downloaded. ABF files require less disk space and download faster. The speed of executing the download function is also increased if the D20 SCPI file resides on the hard disk of the PC.

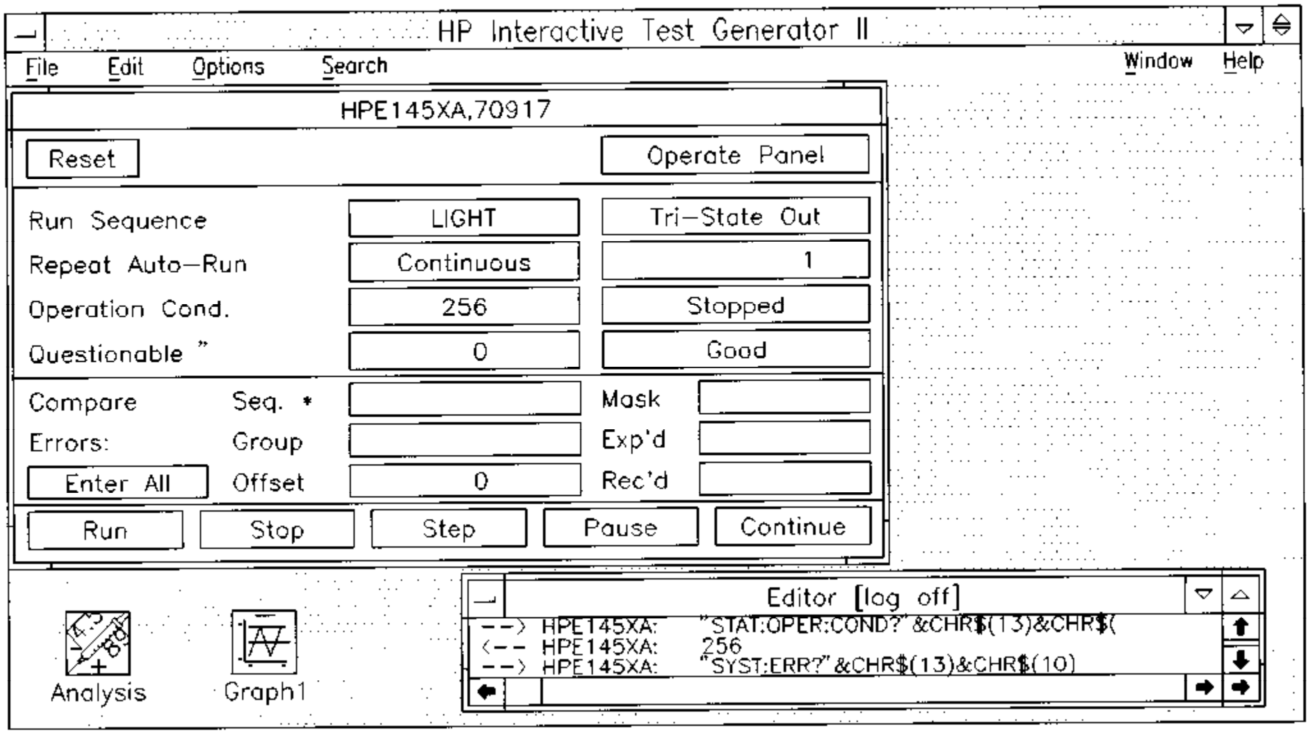
Once the D20 SCPI file has been downloaded, the "RUN" command is issued to the D20 modules and the D20 instrument enters the RUNning state. The example HP IBASIC program found in Table 5 will download and run the D20 digital test using an interrupt to alert the PC computer when the test is completed. During the time the test is running, the computer can control other instrumentation or perform other functions until the interrupt occurs from the D20.

HP Interactive Test Generator II

HP's Interactive Test Generator for DOS provides another tool for DOS test program development. HP Interactive Test Generator II contains an instrument control driver for downloading and running D20 SCPI files already created. This driver also contains debugging features such as single stepping though the test vectors and obtaining information on the status of the digital test.

```
210   DIM Scpifile$[100],Cmds$[500]
220   Scpifile$="b:\TESTFILE.DAT"          !Notice File location
230   !
240   ASSIGN @File TO Scpifile$;FORMAT ON
250   ON ERROR GOTO Eof
260    STATUS @File,1;File_type
270   !
280   Count=1
290   !
300   IF File_type=3 THEN                  ! It is an ASCII file.
310   LOOP
320   ENTER @File;Cmds$
330   OUTPUT @Dft;Cmds$
340   PRINT "ASCII file type, on block #",Count
350   Count=Count+1
360   END LOOP
370   ELSE ! It is an ABF file.
380   LOOP
390   ENTER @File USING "#,-K";Cmds$
400   OUTPUT @Dft USING "#,K";Cmds$
410   PRINT "ABF file type, on block #",Count
420   Count=Count+1
430   END LOOP
440   END IF
```

Table 2. SCPI file Download Routine in HP IBASIC



Run Sequence	LIGHT	Tri–State Out
Repeat Auto–Run	Continuous	1
Operation Cond.	256	Stopped
Questionable "	0	Good

Compare	Seq. *		Mask	
Errors:	Group		Exp'd	
	Offset	0	Rec'd	

```
-->  HPE145XA:    "STAT:OPER:COND?"&CHR$(13)&CHR$(
<--  HPE145XA:    256
-->  HPE145XA:    "SYST:ERR?"&CHR$(13)&CHR$(10)
```

Figure 4. HP Interactive Test Generator II with D20 instrument driver

The Interactive Test Generator II instrument driver for the Model D20 also has the ability to download both ASCII and Arbitrary Block Format (ABF) files. This driver can also log commands required to operate the D20 hardware. These logged commands can be used in other programming environments like Microsoft's QuickBASIC.

Microsoft QuickBASIC

A popular PC-DOS programming language is Microsoft's QuickBASIC. Again, the two major functions involve setting up the D20 hardware in a known state then retrieving and downloading the D20 SCPI file. The setup and download procedures are shown in the QuickBASIC code found in Table 3.

As in previous examples, downloading the D20 SCPI file is probably only a part of the overall test that will be executed on the PC. The download procedure can be a separate program run as a setup program before the test program or it can be incorporated into a larger test program. After the D20 SCPI file has been downloaded, an interrupt procedure can be used to allow the PC to perform other tasks while the D20 test is running.

The QuickBASIC example program found in the appendix at the end of this programming note uses the capabilities of the Model D20 to interrupt the computer when a compare failure occurs in the test.

Microsoft QuickC

The final test development language example uses Microsoft's QuickC to download the D20 SCPI file. As in other programming environments, the user develops additional procedures along with downloading the D20 SCPI file. These procedures build up a test suite for the testing to be performed in the final test system.

```
PRINT "Model D20: Reset and Deleting all information"
        CALL SENDCMD(d20&, "*RST")
        CALL SENDCMD(d20&, "GRO:DEL:ALL")
        CALL SENDCMD(d20&, "SEQ:DEL:ALL")
        CALL SENDCMD(d20&, "TIM:CYCL:DEL:ALL")
PRINT

'Open the SCPI file for transfer to system
OPEN "b:\TESTFILE.DAT" FOR INPUT AS #1

'Send each line of the file to the VXI system
DO UNTIL EOF(1)
        LINE INPUT #1, D$
        CALL SENDCMD(d20&, D$)
LOOP
CLOSE #1
```

Table 3. Setup and Download procedures using QuickBASIC

The practice of setting up the D20 hardware in a known state is again performed within a QuickC program. The D20 commands are the same as those previously found with the other languages. Table 7 contains the entire QuickC program for your reference. Downloading the SCPI file in QuickC is different than previous examples due the nature of file I/O and data manipulation from the C environment. The following section of the QuickC program presents one method of downloading a D20 SCPI file from QuickC.

```
dft_download ()

{
        int     i;
        char    temp[150];
        FILE    *fp;

printf ("Downloading D20 SCPI file\n\n");
        if ( fp = fopen( "B:\\TESTFILE.DAT", "rt") )
                {
                        while (fgets (temp, 250, fp ) != 0)
                                {
                                i = strlen( temp );
                                error = IOOUTPUTS ( dft, temp, i);
                                error_handler (error, "IOOUTPUTS");
                                }
                        fclose( fp );
                }
        else
                printf( "Error in opening file\n");
}
```

Table 4. Retrieving and Downloading D20 SCPI file in QuickC

This routine downloads the SCPI file or prints an error message if the file cannot be opened. As in other applications, additional procedures for interrupts, determining compare failures or retrieving D20 pattern data are implemented to build a test program. Such a program usually becomes part of an overall test system program for managing the entire final test process.

Thus, the procedure of migrating a D20 SCPI file from the development environment to a PC is a simple four step process. The ease of this task is due to the fact that the HP E1496A development software creates a D20 command file in an ASCII format. The file is then quickly transferred to a DOS format for downloading the SCPI file from the PC.

Four test development environments are presented within this programming note and similar routines are possible from other PC applications for downloading D20 SCPI files. Complete program listings for the three language based PC applications can be found on the following pages. Slightly different D20 functions are used in each program and can be adapted to whatever programming environment is used.

The HP IBASIC program contains a procedure for interrupting the computer when the D20 test has stopped. The Microsoft QuickBASIC program uses interrupts in a similar fashion but determines if the interrupt is due to a compare failure or not (this program assumes the D20 test uses the compare function). Lastly, the Microsoft QuickC program shows how you can run a test and acquire the response data from the first four vectors after the D20 test is completed (this program assumes response pins have been specified).

```
10   !      RE-SAVE "DOWNLOAD.TXT"
20   !      HP IBASIC for Windows Program
30   !        for downloading D20 SCPI files
40   !
50   ASSIGN @Dft TO 70917.       !D20 HP-IB Address
60   ON INTR 7 GOSUB Err_chk  !Command Error Interrupt
                              Handler
70   ENABLE INTR 7;2
80   !
90   CLEAR @Dft
100  !
110  OUTPUT @Dft;"*CLS"         !Clears all status event registers
120  OUTPUT @Dft;"*SRE 32"      !Enables D20 to SRQ on standard
                                event
130  OUTPUT @Dft;"*ESE 60"      !Enables error to generate std.
                                events
140  !
150  OUTPUT @Dft;"*RST"         !Reset Hardware
160  OUTPUT @Dft;"DIG:GRO:DEL:ALL"      !Delete all pin groups
170  OUTPUT @Dft;"DIG:SEQ:DEL:ALL"      !Delete all sequences
180  OUTPUT @Dft;"DIG:TIM:CYCL:DEL:ALL" !Delete timing cycles
190  OUTPUT @Dft;"FORM:DATA ASC"        !Response data in ASCII
200  !
210  DIM Scpifile$[100],Cmds$[500]
220  Scpifile$="b:\TESTFILE.DAT"  !Location of D20 SCPI file
230  !
240  ASSIGN @File TO Scpifile$;FORMAT ON
250  ON ERROR GOTO Eof
260  STATUS @File,1;File_type
270  !
280  Count=1
290  !
300  IF File_type=3 THEN          ! File is an ASCII file.
310      LOOP
320          ENTER @File;Cmds$
330          OUTPUT @Dft;Cmds$
340          PRINT "ASCII file type, on block #",Count
350          Count=Count+1
360      END LOOP
370  ELSE        ! File is an ABF file.
380      LOOP
390          ENTER @File USING "#,-K";Cmds$
400          OUTPUT @Dft USING "#,K";Cmds$
410          PRINT "ABF file type, on block #",Count
420          Count=Count+1
430      END LOOP
440  END IF
450  !
460  Eof:                         ! The end of the SCPI file is
                                  reached
470  PRINT ERRM$                  !Error should be: End of file or
                                  buffer found
480  ASSIGN @File TO *
490  PRINT "End of file reached. Download complete."
500  PRINT "Hit CONTINUE to RUN the D20 test"
510  PAUSE
520  !
530  ON INTR 7 GOSUB Test_stopped     !Test Completed
                                      Interrupt Handler
540  ENABLE INTR 7;2
550  !
560  OUTPUT @Dft;"STAT:OPER:ENAB 256"
                                !Enable D20 stopped condition
570  OUTPUT @Dft;"*SRE 128"     !Enable D20 Event for Interrupt
                                (SRQ)
580  !
590  OUTPUT @Dft;"RUN"          !Run the test
600  !
610  WHILE 1                !Endless while loop until test completes
620         !Other tasks could be executed here
630       DISP "DIGITAL TEST RUNNING"
640  END WHILE
650  !
660  STOP
670  !*****************************************************
680  Err_chk:                       ! Subroutine for Error checking
690       CALL Err_sub(@Dft)
700       ENABLE INTR 7;2
710       RETURN
720  !
730  Test_stopped:!
740       PRINT "D20 Test Completed"
750  !
760       END
770  !*****************************************************
780       SUB Err_sub(@Dft) !Subprogram for Error Identification
790           DIM Errmsg$[256]
800           Errnum=-1
810           WHILE Errnum<>0
820               OUTPUT @Dft;"SYST:ERR?"
830               ENTER @Dft;Errnum,Errmsg$
840               PRINT Errnum,Errmsg$
850           END WHILE
860       SUBEND
```

Table 5. HP IBASIC Program for Downloading a D20 SCPI file

```
DECLARE SUB INITIALIZE (ISC&)
DECLARE SUB SENDCMD (DEVICE&, CODES$)
DECLARE SUB GETDATAS (DEVICE&)
DECLARE SUB GETDATA (DEVICE&)
DECLARE SUB WAITFORSRQ (ISC&, DEVICE&)
'
'Microsoft QuickBASIC 4.5 program to send a D20 SCPI file down to the
' D20 hardware using Microsoft QuickBASIC.
'
' REM $INCLUDE: 'c:\hpib\qbsetup'

CLS                                                'Clear the PC screen
ISC& = 7                                           'Interface Select Code of HP 82335A
Dft& = 70917                                       'Full address of D20 modules

CALL INITIALIZE(ISC&)

PRINT "Model D20: Reset and Deleting all information"
        CALL SENDCMD(Dft&, "*RST;*CLS")            'Reset and Clear D20 Status
        CALL SENDCMD(Dft&, "GRO:DEL:ALL")          'Delete pin groups
        CALL SENDCMD(Dft&, "SEQ:DEL:ALL")          'Delete sequences
        CALL SENDCMD(Dft&, "TIM:CYCL:DEL:ALL")     'Delete timing cycles
PRINT

'Open the SCPI file for transfer to system
        OPEN "b:\TESTFILE.DAT" FOR INPUT AS #1     'Notice location of SCPI file
'Send each line of the file to the VXI system
        DO UNTIL EOF(1)
                LINE INPUT #1, D$
                CALL SENDCMD(Dft&, D$)
        LOOP
    CLOSE #1

'Check for any errors in the download
        CALL SENDCMD(Dft&, "SYST:ERR?")
        CALL GETDATAS(Dft&)
                IF LEFT$(INFO$, 2) <> "+0" THEN
                        DO
                        PRINT "ERROR: ", INFO$
                        LOOP WHILE LEFT$(INFO$, 2) <> "+0"
                        PRINT "Test not running - Errors encountered"
                END IF
'Set up D20 Interrupt Conditions
        CALL SENDCMD(Dft&, "STAT:OPER:ENAB 256") 'Enable Stopped Event
        CALL SENDCMD(Dft&, "STAT:QUES:ENAB 512") 'Enable Compare Fail Event
        CALL SENDCMD(Dft&, "*SRE 136")             'Unmask Questionable Data Register
                                                   ' Operation Status Register

        CALL SENDCMD(Dft&, "RUN")                  'Run the D20 test

        CALL WAITFORSRQ(ISC&, Dft&)                'Wait for D20 to Interrupt

        CALL SENDCMD(Dft&, "*STB?")                'Query Status Byte Register

        CALL GETDATA(Dft&)

                IF (VALUE AND 8) = 8 THEN          'Interrupt due to Compare Error
                        PRINT "Compare Error when executing test"
                        CALL SENDCMD(Dft&, "RESP:COMP:ERR?")
                        CALL GETDATAS(Dft&)
                        PRINT " Error Info:"; INFO$
                        PRINT " Error string contents:"
                        PRINT "SEQUENCE name, GROUP name, VECTOR number, "
                        PRINT " compare MASK, EXPECTED pattern, and RECEIVED pattern"
                ELSE                               'Interrupt due to test stopping only
                    PRINT "Test Completed with NO Errors"
                END IF
END
```

Table 6. Microsoft QuickBASIC Program for Downloading a D20 SCIPI File

```
SUB GETDATA (DEVICE&) 'Subprogram for entering numeric data from device
        SHARED PCIB.ERR, PCIB.BASERR, NOERR, VALUE

        MAX.LENGTH% = 100: ACT.LENGTH% = 0
CALL IOENTER(DEVICE&, VALUE)
        IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
END SUB

SUB GETDATAS (DEVICE&) 'Subprogram for entering string data from device
        SHARED PCIB.ERR, PCIB.BASERR, NOERR, INFO$

        MAX.LENGTH% = 100: ACT.LENGTH% = 0
        INFO$ = SPACE$(MAX.LENGTH%)
    CALL IOENTERS(DEVICE&, INFO$, MAX.LENGTH%, ACT.LENGTH%)
        INFO$ = LEFT$(INFO$, ACT.LENGTH%)
        IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
END SUB

SUB INITIALIZE (ISC&) STATIC        'Subprogram for initializing HP-IB interface
        SHARED PCIB.ERR, PCIB.BASERR, 'NOERR

        PRINT "Initializing HP-IB Interface"
        CALL IORESET(ISC&)
                IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
        CALL IOTIMEOUT(ISC&, 5!)
                IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
        CALL IOCLEAR(ISC&)
                IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
        PRINT

END SUB

SUB SENDCMD (DEVICE&, CODES$) 'Subprogram for sending commands to device
        SHARED PCIB.ERR, PCIB.BASERR, NOERR

        CALL IOOUTPUTS(DEVICE&, CODES$, LEN(CODES$))
        IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
END SUB

SUB WAITFORSRQ (ISC&, Dft&) STATIC
        SHARED PCIB.ERR, PCIB.BASERR, NOERR
                SRQ% = 1
CHECKSTAT:      CALL IOSTATUS(ISC&, SRQ%, STATUS%)
                IF PCIB.ERR <> NOERR THEN ERROR PCIB.BASERR
                PRINT "checking status"; STATUS%      'Computer could perform other tasks at this point
                IF STATUS% = 0 THEN GOTO CHECKSTAT
END SUB
```

Table 6. Continued

```
/*****************************************************************************************************
Microsoft QuickC 2.5 program for downloading                *
*   an HP 75000 Model D20 SCPI file        *
*****************************************************************************************************/
#include <stdio.h>
#include <string.h>
#include <cfunc.h>
#include <chpib.h>

short  error;

#define  isc  7              /*Interface select code of HP 82335A*/

#define  dft  70917          /*HP-IB Address of Model D20 */

#define      SRQLINE 1

main ()
{
initialize ();
dft_reset ();
dft_download ();
dft_int_setup ();
dft_run ();
wait_for_srq ();
dft_resp_data ();
}

/***Error Handling function*****************************************************************************/
error_handler (error, routine)

int          error;
char       *routine;
{
          char *estring;
          char ch;

          if (error != NOERR)
                {
                printf ("Error in call to %s \n", routine);
                printf ("Error = %d : %s \n", error, errstr(error));
                printf ("Press <RETURN> to continue: ");
                scanf ("%c", &ch);
                }
}

/***Initialize HP-IB Interface**************************************************************************/
initialize ()
{
  printf ("Initializing HP-IB interface. \n");
  error = IORESET (isc);
  error_handler (error, "IORESET");
  error = IOTIMEOUT (isc, 5.0);
  error_handler (error, "IOTIMEOUT");
  error = IOCLEAR (isc);
  error_handler (error, "IOCLEAR");
}

/***Reset D20 and delete previous tests*****************************************************************/
dft_reset ()
{
  printf ("D20: Reset and Deleting all information\n\n");
  error = IOOUTPUTS ( dft, "*RST;*CLS",9);
  error_handler (error, "IOOUTPUTS");
  error = IOOUTPUTS ( dft, "GRO:DEL:ALL", 11);
  error_handler (error, "IOOUTPUTS");
  error = IOOUTPUTS ( dft, "SEQ:DEL:ALL", 11);
  error_handler (error, "IOOUTPUTS");
  error = IOOUTPUTS ( dft, "TIM:CYCL:DEL:ALL", 16);
  error_handler (error, "IOOUTPUTS");
}
```

Table 7. Microsoft QuickC Program for Downloading a D20 SCPI file

```
/***Retrieve and download the D20 SCPI file*********************************************************************************/
dft_download ()
{
  int     i;
  char    temp[150];
  FILE    *fp;

  printf ("Downloading D20 SCPI file\n\n");
  if( fp = fopen( "B:\\TESTFILE.DAT", "rt") )  /*Location of SCPI file*/
           {
           while(fgets (temp, 250, fp ) != 0)
                 {
                 i = strlen( temp );
                 error = IOOUTPUTS ( dft, temp, i);
                 error_handler (error, "IOOUTPUTS");
                 }
           fclose( fp );
           }
  else
  printf( "Error in opening file\n");
}

/***Setup D20 Interrupt Condition**********************************************************************************************/
dft_int_setup ()
{
  error = IOOUTPUTS ( dft, "STAT:OPER:ENAB 256", 18);
  error_handler (error, "IOOUTPUTS");
  error = IOOUTPUTS ( dft, "*SRE 128", 8);
  error_handler (error, "IOOUTPUTS");
}

/***Run the D20 test******************************************************************************************************************/
dft_run ()
{
  error = IOOUTPUTS ( dft, "RUN", 3);
  error_handler (error, "IOOUTPUTS");
  printf ("D20 test running \n");
}

/***Wait for D20 to IComputer*************************************************************************************************/
wait_for_srq ()
{
  int response;

  do


           {
           error = IOSTATUS ( isc, SRQLINE, &response);
           error_handler (error, "IOSTATUS");
           printf("SRQ Loop\n");    /*Loop until SRQ*/
           }
  while (response == 0);
}

/***Enter D20 Pattern
Data******************************************************************************************************************************/
dft_resp_data ()
{
  int length;
  char info[100];

  error = IOOUTPUTS ( dft, "RESP:PATT:SEQ:PART? 0,4", 23);
  error_handler (error, "IOOUTPUTS");
  error = IOENTERS ( dft, info, &length);
  error_handler (error, "IOENTERS");
  printf("\n Response DATA: %s",info);
}
```

Table 7. Continued

Microsoft and MS-DOS are U.S. registered trademarks of Microsoft Corporation. Intel and Intel 286/386 are U.S. registered trademarks of Intel Corporation.

HEWLETT
PACKARD

For more information, call your local HP sales office listed in your telephone directory or an HP regional office listed at right for the location of your nearest sales office.United States:

United States
Hewlett-Packard Company
4 Choke Cherry Road
Rockville, MD 20850
(301) 670 4300

Hewlett-Packard Company
5201 Tollview Drive
Rolling Meadows, IL 60008
(708) 255 9800

Hewlett-Packard Company
5651 W. Manchester
Los Angeles, CA 90045
(213) 337 8035

Hewlett-Packard Company
2000 South Park Place
Atlanta, GA 30339
(404) 980 7351

Canada:
Hewlett-Packard Ltd.
6877 Goreway Drive
Mississauga, Ontario L4V 1M8
(416) 678 9430

Japan:
Yokogawa-Hewlett-Packard Ltd.
15-7, Nishi Shinjuku 4 Chome
Shinjuku-ku
Tokyo 160, Japan
(03) 5371 1351

Mexico:
Hewlett-Packard
Latin American Region Headquarters
Monte Pelvoux No. 111
Lomas de Chapultepec
11000 Mexico, D.F.
(525) 202 0155

Brazil:
(55 11) 709 - 1444

For other Latin American cities consult local telephone directory or call:
(525) 202 0155

Australia/New Zealand:
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
Australia (A.C.N. 004 394 763)
(03) 895 2895

Far East:
Hewlett-Packard Asia Ltd.
22/F Bond Centre, West Tower
89 Queensway
Central, Hong Kong
(852) 848 7777

Korea:
(82 2) 769 - 0114

Taiwan:
(886 2) 712 - 0404

Peoples' Republic of China:
(86 1) 505 - 3888

Singapore:
(65) 271 - 9444

Malaysia:
(60 3) 298 - 6555

For other Asian cities consult local telephone directory or call:
(852) 848 7777

In Europe, please call your local HP sales office or representative:

Austria:
(0222) 2500-0

Central Europe, USSR, and Yugoslavia:
Vienna – Austria
(0222) 2500-0

Belgium and Luxembourg:
(02) 761 31 11

Denmark:
(42) 81 66 40

Finland:
(90) 88 721

France:
(1) 69 82 65 00

Germany:
(06172) 16 0

Greece:
(01) 68 28 811

Iceland:
(91) 67 10 00

Ireland:
(01) 88 33 99

Israel:
Computation and
Measurement Systems (CMS) Ltd.
(03) 5380 333

Italy:
(02) 95 300 134

Netherlands:
(020) 547 6669

Norway:
(02) 87 97 00

Portugal:
(11) 301 73 30

Spain:
900 123 123

Sweden:
(08) 750 20 00

Switzerland:
(057) 31 21 11 (Headoffice)
(022) 780 41 11 (Suisse Romande)
(046) 05 15 05 (Customer Information Center)

South Africa:
HiPerformance Systems
(011) 802 5111

Turkey:
175 29 70

U.K.:
(0344) 369 369

Europe/Africa/Middle East:
Hewlett-Packard S.A.
Marcom Operations Europe
P.O. Box 529
1180 AM Amstelveen
The Netherlands

Printed in USA 2/92

5091-3810E