Applied Industrial Hygiene

ISSN: 0882-8032 (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaph20

A Computer Program for Use with the Holaday HI-3320 or Metrosonics dl-332 Data Logger

Richard M. Edwards , Clinton Cox & Barbara A. Grajewski

To cite this article: Richard M. Edwards , Clinton Cox & Barbara A. Grajewski (1989) A Computer Program for Use with the Holaday HI-3320 or Metrosonics dl-332 Data Logger, Applied Industrial Hygiene, 4:11, 291-306, DOI: 10.1080/08828032.1989.10390655

To link to this article: https://doi.org/10.1080/08828032.1989.10390655

A Computer Program for Use with the Holaday HI-3320 or Metrosonics dl-332 Data Logger

Richard M. Edwards, Clinton Cox, and Barbara A. Grajewski
National Institute for Occupational Safety and Health, Division of Biomedical and Behavioral Science, Physical Agents Effects Branch, Radiation Section, 4676 Columbia Parkway, Cincinnati, Ohio 45226-1998

A computer program has been written that can transfer data from a Holaday HI-3320 or the equivalent Metrosonics dl-332 data logger to an MS-DOS™ computer and present that data in numeric or graphic format. A radiofrequency (RF) field survey instrument was connected to a Holaday Model HI-3320 Data Logger which recorded continuous data. After data were stored, the data logger was interfaced with an MS-DOS-compatible computer, and the software was used to upload the data to a computer. Measurement data could be saved and viewed in a variety of report forms, plotted onscreen, printed, or transferred in an ASCII file format to a separate statistics program for more detailed analysis. This software expedites determination of compliance with occupational exposure standards and guidelines or comparison with other exposure data. **Edwards, R.M.; Cox, C.; Grajewski, B.A.: A Computer Program for Use with the Holiday HI-3320 or Metrosonics dl-332 Data Logger. Appl. Ind. Hyg. 4:291–306; 1989.**

Introduction and Background

For continuous measurement of radiofrequency (RF) radiation in the workplace, standard field survey instruments must be connected to a continuous recording device such as a data logger. The data logger used for this application must meet specialized operational requirements: adequate electromagnetic interference (EMI) shielding; short sampling periods; and computer interfacing for retrieval, uploading (transfer), and statistical analysis of the measurement data.

A National Institute for Occupational Safety and Health (NIOSH) epidemiologic study of RF heater operators required the comprehensive and accurate measurement of exposures to RF radiation. Although no data logger examined met our requirements completely, the Holaday Industries HI-3320 (Holaday Industries, Inc., Eden Prairie, Minnesota) data logger was selected for use.[1] A Metrosonics dl-332 (Metrosonics, Inc., Rochester, New York) data logger was evaluated in the laboratory. Except for the different brand name, it appears to be identical to the Holaday HI-3320. However, the Metrosnics dl-332 was not evaluated in the field portion of the study.

Commercially available software (Metrosonics, Inc., Rochester, New York) is relatively expensive and does not fully meet our application and graphics needs. Thus, a QuickBASIC 4.0 (Microsoft® Corporation, Redmond, Washington) program was developed for use with our data logger files.

The Holaday HI-3320 is capable of storing and providing an output of raw or summary data in a variety of report forms. Using this program, the HI-3320 data logger can load these data as standard text (ASCII) files to an MS-DOS™-compatible computer. The software allows the data files to be saved, viewed, plotted onscreen, printed, or exported in an ASCII format for use with a separate statistics program when more detailed analysis is required. These options expedite determination of compliance with occupational exposure standards and guidelines or comparison with other exposure data.

Equipment and Methodology

Software Requirements

Originally, a BASICA version of this program, which will run on most MS-DOS-compatible computers, was written for use on a Compaq (Compaq Computer Corporation, Houston, Texas) portable computer. The program is limited by BASICA to 64 kbytes of memory which restricts the size of files available for viewing or graphing to about 34 kbytes. In addition, insufficient memory for string and stack space prevents repeated access to subroutines for viewing and graphing data. Also, the execution speed is far too slow for practical use of the BASICA interpreter.

Next, a QuickBASIC 3.0 version of the program was developed and then compiled to increase program speed. The compiler also provides a program compatible with other MS-DOS computers. However, it did not solve the problems of limited file size and insufficient string and stack space. To solve these problems, the program was rewritten, using subprograms instead of subroutines, and compiled in QuickBASIC 4.0. The inclusion of subprograms eliminated the stack space errors and allowed more efficient use of available string space increasing the maximum allowable size of data files. Recompiling provided an executable

The use of company or trade names is for identification only and does not imply indorsement by the National Institute for Occupational Safety and Health.

program called DATALOG.EXE for use with MS-DOS-compatible computers. DATALOG.EXE was tested on a Compaq DeskPro and a Compaq portable computer with a color graphics adapter (CGA) monitor. For the program to operate, the computer must have a minimum of 640 kbytes of memory.

DATALOG.EXE must have access to COMMAND.COM and SORT.EXE. These files either must be copied to the program disk or made available to the program through the path statement. The source code for the QuickBASIC 4.0 program is listed in the Appendix.

Hardware Configuration

The data logger must be preprogrammed to meet specifications required by DATALOG.EXE. The program communicates at 1200 baud through the data logger serial port that is connected to the computer RS-232 serial port. The program requires the presence of a tag number, a 1-second time history period length, and all three time history statistics within a time history compressed report in order to graph the data. The tag numbers are required to separate data logger recording events. The 1-second time history period length, including all three time history statistics, will provide a minimum, an average, and a maximum value for each second of a recorded event. Other parameters, e.g., test starting date and time, sample rate, and calibration points, create more informative reports but are not required for the computer–data logger upload.[2]

A cable that connects the HI-3320 to the computer is available from Metrosonics or can be fabricated from commercially available parts to meet RS-232C serial port (communication) requirements. In the data logger's digital I/O port, there are five pins of which only three are used to interface the data logger with the computer. Data are transmitted through pin 2 and received through pine 3; pin 1 serves as a relative ground between the devices.[2] The computer's RS-232 serial port requires a null-modem cable with no handshaking capabilities.[3]

Program Features

Key Functions

DATALOG.EXE is primarily menu driven, since ease of use was a major consideration in its design. User input is required only to name new data files. The program minimizes the number of keystrokes through extensive use of the function keys (F1–F10) and the cursor pad (Up, Down, Pg Up, Pg Dn, Home, and End keys). After a selection has been made from the MAIN MENU (Figure 1), a "help line," located at the bottom of the screen, guides the user through the program and provides assistance if an error condition occurs. The arrow keys are used to highlight an individual file name in subprograms requiring the selection of data files. The return key is pressed to retrieve the highlighted file for use within that subprogram. The F10 key is used to return to the MAIN MENU.

Main Menu Function Keys

(F1) Help—The help file provides a brief description of the various options found within the MAIN MENU and describes the procedures used to successfully complete activities selected from the MAIN MENU. The onscreen "help line" normally provides sufficient information to use the program.

(F2) Default Drive—This option allows the user to change the default drive for data storage and retrieval.

(F3) Data File Directory—The directory provides a list of all data logger files on disk.

F1.) HELP
F2.) DEFAULT DRIVE
F3.) DATA FILE DIRECTORY
F4.) CREATE A NEW DATA FILE
F5.) VIEW AN EXISTING FILE
F6.) PRINT AN EXISTING FILE.
F7.) PLOT A GRAPHICS FILE
F8.) EXIT TO MS-DOS

FIGURE 1. MAIN MENU of HI-3320 data logger program.

(F4) Create a New Data File—This option allows the user to upload data stored in the HI-3320 data logger to a default disk. Once this option is selected, the user will be asked to name the data file. The program will provide the proper file name extension based upon the type of report selected.

After entering the file name, a sub-menu of "REMOTE REQUESTS" and "CHARACTERS TO SEND TO THE HI-3320" is displayed. The report option is selected with function keys (refer to the *HOLADAY HI-3320 Data Logger Operator's Manual*[2] for a description of the various reports). Reports may contain all or portions of the recorded data, e.g., "amplitude" (frequency) distributions, summary statistics, and time histories. In addition, each type of report is available in either a compressed or formatted fashion. The compressed reports contain only data and are suitable for further analyses with minimal editing. The formatted reports add descriptive headings and graphics information to the data report.

(F5) View an Existing File—This option allows the user to view the data file by using the "Home," "End," "Pg Up," "Pg Dn," and arrow keys to scroll through the file.

(F6) Print an Existing File—This option prints out the data file. All printouts originate from disk files, not from the data logger.

(F7) Plot a Graphics File—This selection allows the user to plot an existing compressed time history (*.LCT) file on the screen.

(F8) Exit to MS-DOS—This selection ends the program and returns the computer to MS-DOS control.

Prompts and help instructions make additional documentation unnecessary for most users. If a more detailed statistical analysis is required, the compressed report form can be edited and exported as an ASCII text or a delimited ASCII text file. The ASCII file can then be uploaded into SAS (SAS Institute, Inc., Cary, North Carolina) or many other statistical, spreadsheet, or database programs.

Availability

The program's QuickBASIC 4.0 source code is listed completely in the Appendix. This source code listing must be compiled with QuickBASIC 4.0 to obtain a program that can be executed in the MS-DOS environment. The compiled QuickBASIC 4.0 version is a public domain program. Users should report to the authors any problems encountered or any modifications which improve the program's utility. The American Conference of Governmental Industrial Hygienists (ACGIH, 6500 Glenway Ave., Bldg. D-7, Cincinnati, OH 45211-4439) is offering the QuickBASIC 4.0 source code with its compiled version on a 360 kbyte MS-DOS disk for a fee of $10.00. The charges are for copy and distribution services and not for the program.

Conclusions and Recommendations

The introduction of EMI-shielded data loggers has made characterization of RF radiation exposure in the workplace easier and more accurate. The software described in this article provides an inexpensive and thorough way to implement the RF radiation measurement data logging process. Data files generated by the data logger provide the necessary information to assess compliance with occupational exposure standards and guidelines (ACGIH, American National Standards Institute, and Occupational Safety and Health Administration); the raw data can also be further analyzed for research studies.

The software is adapted for the particular requirements of RF radiation measurement. For example, subprograms to upload or display real-time data on the computer were not written since stray RF emissions preclude field use of a computer with the data logger. Addition of zoom graphics and a raw data file cleaning procedure are desirable features which could be added to the source code. The authors encourage the use of this software for RF radiation measurements and other applications and welcome comments or suggestions for its improvement.

References

1. Cox, C.; Grajewski, B.A.; Edwards, R.M.; et al: Two Systems for Collection, Storage, and Analysis of Measurements Made with RF Field Survey Instruments. Appl. Ind. Hyg. 4(11):286 (1989).
2. *Anonymous:* Operator's Manual; HI-3320 Data Logger. Holaday Industries, Inc. Eden Prairie, MN (1984).
3. Seyer, M.D.: RS-232 Made Easy. Prentice-Hall, Inc., Englewood Cliffs, NJ (1984).

Received 12/9/88; review decision 3/1/89; revision 6/12/89; accepted 6/15/89

Appendix

DATALOG.BAS
HOLADAY HI-3320 DATA LOGGER UPLOAD PROGRAM
Written by Richard M. Edwards, 5/27/88
Compiled with Microsoft BC /ah/e/o/v (QuickBASIC Version 4.0b)
QuickBASIC BC command options:

```
'/ah - allows dynamic arrays to occupy all of available memory.
'/e - indicates the presence of ON ERROR RESUME statements.
'/o - substitutes BCOM41.LIB run-time library for BRUN41.LIB
'     thereby creating a 'stand alone' executable file.
'/v - enables event trapping for communications.
'
'(*) - sets off and describes related portions of main or subprorgams.
'(=) - marks the begining of subprograms.
'(') - describes function of individual program lines.
'**********************SET-UP SUBPROGRAMS********************************
DECLARE SUB CLEARSCREEN (FG%, BG%, BOR%)
DECLARE SUB CLRKYBD ()
DECLARE SUB CREATE ()
DECLARE SUB DELAY (WAIT$)
DECLARE SUB DOS (F8$, F10$)
DECLARE SUB DISKDRIVE ()
DECLARE SUB FILE ()
DECLARE SUB GRAPH ()
DECLARE SUB HELP ()
DECLARE SUB INKEYS (V$)
DECLARE SUB MAINMENU ()
DECLARE SUB PAUSE (SEC)
DECLARE SUB PRINTMESSAGE ()
DECLARE SUB PRNT ()
DECLARE SUB SELCTFILE (FILENAME%)
DECLARE SUB VIEWER (OPENFIL$)
DECLARE SUB VIEWFILE ()
'************************DECLARE GLOBAL VARIABLES************************
COMMON SHARED FG%, BG%, BOR%
COMMON SHARED DRIVE$, DSKFIL$, GRAFIL$, MENU$, WAIT$
COMMON SHARED CLEARMESSAGE$, MESSAGE$, HEADER$
COMMON SHARED TOP$, SIDE$, BMID$, BOT$
COMMON SHARED HOME$, UpARROW$, PgUp$, END$, DnARROW$, PgDn$
COMMON SHARED F1$, F2$, F3$, F4$, F5$, F6$, F7$, F8$, F9$, F10$
'******************LISTING OF SELECTED VARIABLES**********************
'ADJUST% - Exponent for adjusting calibration data during plot.
'BG% - Background screen color.
'BMID$ - Screen border characters.
'BOR% - Border screen color.
'BOT$ - Screen border characters.
'CALIB - Numeric value of calibration point(CALIB$) two.
'CLEARMESSAGE$ - Clears portions of screen (e.g. 'help line).
'COMFIL$ - Communication port location and parameters.
```

```
'DAT - Individual tag data points.
'DL$ - Datalogger input string.
'DRIVE$ - Stores current working disk drive.
'DSKFIL$ - holds the working filename for the various subprograms.
'DT - String data array for view subprogram.
'EOD - End of data.
'ERROR$ - Error message for 'help line'.
'FG% - Foreground screen color.
'GRAFIL$ - Working file for plotting subprogram.
'HEADER$ - Displays current subprogram choice.
'&H17 - Location to check for Caps lock, Num lock, etc.
'&H40 - Location to check communication port buffer size.
'MAXTAG - Stores the value of the highest data logger tag number.
'MESSAGE$ - Stores help messages for 'help line'.
'MENU$ - Determines which onscreen menu will be provided.
'OPENFIL$ - Stores filename for use within various subprograms.
'PAD - Pads message strings with spaces to a uniform size.
'POSITION - Screen cursor position for diskdrive subprogram.
'SIDE$ - Screen border characters.
'SPEED$ - Baud rate for serial comunication.
'ST$ - Throwaway string variable.
'STDATE$ - Starting date (data logger upload).
'STTIME$ - Starting time (data logger upload)
'TAG$ - Event marker from data logger.
'TL - Event counter.
'TOP$ - Screen border characters.
'WAIT$ - Signals main program when data logger upload is complete.
'*********SET ASIDE FAR HEAP (NON BASIC) MEMORY FOR DATA ARRAYS********
' $DYNAMIC
A = 9: B = 1200
DIM SHARED DT(A, B) AS STRING * 16       'GRAPHICS DATA ARRAY
ERASE DT
A = 1600
DIM SHARED VT(A) AS STRING * 80          'VIEW STRING DATA ARRAY
ERASE VT
DIM SHARED DL$(9000)
ERASE DL$
'***********SET ASIDE DG GROUP (BASIC) MEMORY FOR VARIABLES************
' $STATIC
DIM SHARED DAT(1, 1200)          'INDIVIDUAL TAG DATA POINTS
DIM SHARED ST$(20)               'DATA FILE HEADER INFORMATION
DIM TAG$(10)                     '# OF DATA LOGGER TAGS ALLOWED
DIM STTIME$(10)                          'INDIVIDUAL TAG STARTING TIMES
DIM STDATE$(10)                          'INDIVIDUAL TAG STARTING DATE
CLS
'*****************TURN OFF AND DISABLE FUNCTION KEYS******************
KEY OFF: FOR I = 1 TO 10: KEY I, "": NEXT I
'**********************SET UP SCREEN PARAMETERS**********************
SCREEN 0, 1: WIDTH 80
FG% = 7: BG% = 1: BOR% = 9: COLOR FG%, BG%, BOR%: CLS
'**********************INITIALIZE VARIABLES*****************************
'*BORDER CHARACTERS*
TOP$ = CHR$(201) + STRING$(78, 205) + CHR$(187)
SIDE$ = CHR$(186) + STRING$(78, 32) + CHR$(186)
BMID$ = CHR$(204) + STRING$(78, 205) + CHR$(185)
BOT$ = CHR$(200) + STRING$(78, 205) + CHR$(188)
CLEARMESSAGE$ = STRING$(78, 32)
'
'*FUNCTION KEY CODES*
F1$ = CHR$(0) + CHR$(59)
F2$ = CHR$(0) + CHR$(60)
F3$ = CHR$(0) + CHR$(61)
F4$ = CHR$(0) + CHR$(62)
F5$ = CHR$(0) + CHR$(63)
F6$ = CHR$(0) + CHR$(64)
F7$ = CHR$(0) + CHR$(65)
F8$ = CHR$(0) + CHR$(66)
F9$ = CHR$(0) + CHR$(67)
F10$ = CHR$(0) + CHR$(68)
'
'*CURSOR CONTROL KEY CODES*
HOME$ = CHR$(0) + CHR$(71)
UpARROW$ = CHR$(0) + CHR$(72)
PgUp$ = CHR$(0) + CHR$(73)
END$ = CHR$(0) + CHR$(79)
```

```
        DnARROW$ = CHR$(0) + CHR$(80)
        PgDn$ = CHR$(0) + CHR$(81)
        '
        DRIVE$ = "A:\": CLOSE #2
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        '*************************PRINT TITLE SCREEN*************************
        LOCATE 5, 28: PRINT "DATALOGGER UPLOAD PROGRAM"
        LOCATE 8, 33: PRINT "04 MARCH 1988"
        LOCATE 10, 25: PRINT "WRITTEN BY  RICHARD M. EDWARDS"
        LOCATE 12, 31: PRINT "RADIATION SECTION"
        LOCATE 14, 25: PRINT "PHYSICAL AGENTS EFFECTS BRANCH"
        LOCATE 16, 18: PRINT "DIVISION OF BIOMEDICAL AND BEHAVIORAL SCIENCE"
        LOCATE 18, 14
        PRINT "NATIONAL INSTITUTE FOR OCCUPATIONAL SAFETY AND HEALTH"
        COLOR BG%, FG%, BOR%
        MESSAGE$ = "PRESS [F10] TO CONTINUE. ": CALL PRINTMESSAGE
        COLOR FG%, BG%, BOR%
        '*****************WAIT FOR [F10] TO BEGIN MAIN PROGRAM*****************
GETKEY1:
        CALL INKEYS(V$)
        IF V$ = F10$ THEN
            GOTO Start
        ELSE GOTO GETKEY1
        END IF
        '*************************BEGIN MAIN PROGRAM*************************
Start:
        ON ERROR GOTO ERRTRAP
        CALL MAINMENU
        FOR I = 1 TO 10: KEY I, "": NEXT I
        '*********************CHECK FOR FUNCTION KEY INPUT*********************
GETKEY2:
        CALL INKEYS(V$)
        IF V$ = F1$ THEN
            REDIM VT(1600) AS STRING * 80
            CALL HELP
        ELSEIF V$ = F2$ THEN
            CALL DISKDRIVE
        ELSEIF V$ = F3$ THEN
            CALL FILE
        ELSEIF V$ = F4$ THEN
            REDIM DL$(9000)
            CALL CREATE
        ELSEIF V$ = F5$ THEN
            REDIM VT(1600) AS STRING * 80
            CALL VIEWFILE
        ELSEIF V$ = F6$ THEN
            CALL PRNT
        ELSEIF V$ = F7$ THEN
            REDIM DT(9, 1200) AS STRING * 16
            CALL GRAPH
        ELSEIF V$ = F8$ THEN
            CALL DOS(F8$, F10$)
        ELSE GOTO GETKEY2
        END IF
        '
        CALL CLRKYBD
        GOTO Start
        '************************ERROR TRAPPING ROUTINE************************
ERRTRAP:
        ERASE DT, VT, DL$
        CLOSE : KEY OFF
        FOR J = 1 TO 10: KEY(J) OFF: NEXT J
        '
        IF ERR = 5 THEN
            ERROR$ = "ILLEGAL FUNCTION CALL.  "
        ELSEIF ERR = 24 THEN
            ERROR$ = "DEVICE TIMEOUT.  "
        ELSEIF ERR = 25 THEN
            ERROR$ = "DEVICE FAULT.  "
        ELSEIF ERR = 52 THEN
            ERROR$ = "BAD FILE NUMBER.  "
        ELSEIF ERR = 53 THEN
            ERROR$ = "FILE NOT FOUND.  "
        ELSEIF ERR = 57 THEN
            ERROR$ = "DEVICE I/O ERROR.  "
```

```
        ELSEIF ERR = 64 THEN
            ERROR$ = "BAD FILE NAME.  "
        ELSEIF ERR = 71 THEN
            ERROR$ = "DISK NOT READY.  "
        ELSEIF ERR = 72 THEN
            ERROR$ = "DISK MEDIA ERROR.  "
        ELSEIF ERR = 76 THEN
            ERROR$ = "PATH NOT FOUND.  "
        ELSE ERROR$ = "ERROR = " + STR$(ERR) + " IN LINE " + STR$(ERL) + ".  "
        END IF
        '
        MESSAGE$ = ERROR$ + "PRESS [RETURN] TO CONTINUE. "
        FG% = 12: BG% = 16
        CALL PRINTMESSAGE
ENDERRTRAP:
        S$ = INKEY$
        IF S$ <> CHR$(13) THEN GOTO ENDERRTRAP
        RESUME Start
        END
        '*************************END OF MAIN PROGRAM*************************

        '-----------------------------------------------------------------------
        SUB CLEARSCREEN (FG%, BG%, BOR%)
        '******************CLEARS SCREEN AND PRINTS BORDER*******************
        COLOR FG% - 7, BG%, BOR%
        LOCATE 1, 1: PRINT TOP$
        FOR I% = 2 TO 22: PRINT SIDE$; : NEXT I%
        PRINT BMID$; : PRINT SIDE$;
        LOCATE 25, 1: PRINT BOT$;
        COLOR FG%, BG%, BOR%
        END SUB

        '-----------------------------------------------------------------------
        SUB CLRKYBD
        '************************CLEARS KEYBOARD BUFFER************************
CLRBUFFR:
        V$ = INKEY$: IF V$ <> "" THEN GOTO CLRBUFFR
        END SUB

        '-----------------------------------------------------------------------
        SUB CREATE
        '******************UPLOADS ASCII TEXT FROM DATA LOGGER****************
        CLOSE : SPEED$ = "1200"          'CLOSE FILES & SETS BAUD TO 1200 bps
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        COMFIL$ = "COM1:" + SPEED$ + ",N,8,1"    'DEFINES COMMUNICATION
        ' PARAMETERS TO DATA LOGGER - NO PARITY, 8 DATA BITS AND 1 STOP BIT
        LOCATE 2, 2
        PRINT "THE FOLLOWING IS A LIST OF EXISTING DATA FILES ON DRIVE:"
        LOCATE 3, 2: FILES DRIVE$ + "*.L*"
        MESSAGE$ = "ENTER a Filename or press [F10] to return to the MAIN MENU."
        CALL PRINTMESSAGE
        KEY 10, "MENU" + CHR$(13)
        '********************GET AND VALIDATE USER INPUT**********************
GETFILENAME:
        DSKFIL$ = ""
        CALL CLRKYBD
        LOCATE 22, 2: PRINT CLEARMESSAGE$
        LOCATE 22, 2: INPUT ; DSKFIL$
        IF DSKFIL$ = "MENU" THEN GOTO ENDCREATE
        IF DSKFIL$ = "" THEN GOTO GETFILENAME
        '
        IF INSTR(DSKFIL$, ".") > 0 THEN
            MESSAGE$ = "The program will provide the proper extension."
            MESSAGE$ = MESSAGE$ + "  [F10]-MAIN MENU."
            CALL PRINTMESSAGE
            GOTO GETFILENAME
        ELSEIF LEN(DSKFIL$) > 8 THEN
            MESSAGE$ = "Please ENTER 8 characters or less.  [10]-MAIN MENU."
            CALL PRINTMESSAGE
            GOTO GETFILENAME
        ELSE GOTO STARTCREATE
        END IF
STARTCREATE:
        KEY OFF: FOR I = 1 TO 10: KEY I, "": NEXT I
        T = 0: TL = 0
```

```
        '**********************OPEN COMMUNICATION PORT*************************
        OPEN COMFIL$ FOR RANDOM AS #1 LEN = 256
        '*****************SEND CONTROL CHARACTERS TO DATA LOGGER****************
        PRINT #1, CHR$(13);
        CALL PAUSE(1!)
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        '******************PRINT DATA LOGGER REPORT - OPTIONS MENU**************
        LOCATE 2, 17
        PRINT "DATA LOGGER                                    CHARACTERS SENT "
        LOCATE 3, 17
        PRINT "REPORT OPTIONS                                 TO THE HI-3320"
        COLOR FG% - 7, BG%, BOR%: LOCATE 3, 2: PRINT BMID$
        COLOR FG%, BG%, BOR%
        LOCATE 6, 5
        PRINT "   [F1]     (C)OMPRESSED (A)LL REPORT                          ";
        COLOR BG%, FG%: PRINT " CA ": COLOR FG%, BG%
        LOCATE 8, 5
        PRINT "   [F2]     (F)ORMATTED (A)LL REPORT                           ";
        COLOR BG%, FG%: PRINT " FA ": COLOR FG%, BG%
        LOCATE 10, 5
        PRINT "   [F3]     (C)OMPRESSED AMPLITUDE (D)ISTRIBUTION REPORT       ";
        COLOR BG%, FG%: PRINT " CD ": COLOR FG%, BG%
        LOCATE 12, 5
        PRINT "   [F4]     (F)ORMATTED AMPLITUDE (D)ISTRIBUTION REPORT        ";
        COLOR BG%, FG%: PRINT " FD ": COLOR FG%, BG%
        LOCATE 14, 5
        PRINT "   [F5]     (C)OMPRESSED (O)VERALL STATISTICS REPORT           ";
        COLOR BG%, FG%: PRINT " CO ": COLOR FG%, BG%
        LOCATE 16, 5
        PRINT "   [F6]     (F)ORMATTED (O)VERALL STATISTICS REPORT            ";
        COLOR BG%, FG%: PRINT " FO ": COLOR FG%, BG%
        LOCATE 18, 5
        PRINT "   [F7]     (C)OMPRESSED (T)IME HISTORY REPORT                 ";
        COLOR BG%, FG%: PRINT " CT ": COLOR FG%, BG%
        LOCATE 20, 5
        PRINT "   [F8]     (F)ORMATTED (T)IME HISTORY REPORT                  ";
        COLOR BG%, FG%: PRINT " FT ": COLOR FG%, BG%
        MESSAGE$ = "PLEASE WAIT."
        CALL PRINTMESSAGE
        CALL PAUSE(1!)
        MESSAGE$ = "Press a function key to select a report.  [F10] - MAIN MENU. "
        CALL PRINTMESSAGE
        CALL CLRKYBD
        '**WAIT FOR REPORT OPTION SELECTION AND ASSIGN APPROPRIATE EXTENSION**
GETKEY3:
        CALL INKEYS(V$)
        IF V$ = F1$ THEN
            B$ = "CA"
            DSKFIL$ = DSKFIL$ + ".LCA"    '(L)OGGED (C)OMPRESSED (A)LL
        ELSEIF V$ = F2$ THEN
            B$ = "FA"
            DSKFIL$ = DSKFIL$ + ".LFA"    '(L)OGGED (F)ORMATTED (A)LL
        ELSEIF V$ = F3$ THEN
            B$ = "CD"
            DSKFIL$ = DSKFIL$ + ".LCD"    '(L)OGGED (C)OMPRESSED (D)ISTRIBUTION
        ELSEIF V$ = F4$ THEN
            B$ = "FD"
            DSKFIL$ = DSKFIL$ + ".LFD"    '(L)OGGED (F)ORMATTED (D)ISTRIBUTION
        ELSEIF V$ = F5$ THEN
            B$ = "CO"
            DSKFIL$ = DSKFIL$ + ".LCO"    '(L)OGGED (C)OMPRESSED (O)VERALL STAT.
        ELSEIF V$ = F6$ THEN
            B$ = "FO"
            DSKFIL$ = DSKFIL$ + ".LFO"    '(L)OGGED (F)ORMATTED (O)VERALL STAT.
        ELSEIF V$ = F7$ THEN
            B$ = "CT"
            DSKFIL$ = DSKFIL$ + ".LCT"    '(L)OGGED (C)OMPRESSED (T)IME HISTORY
        ELSEIF V$ = F8$ THEN
            B$ = "FT"
            DSKFIL$ = DSKFIL$ + ".LFT"    '(L)OGGED (F)ORMATTED (T)IME HISTORY
        ELSEIF V$ = F10$ THEN
            GOTO ENDCREATE
        ELSE GOTO GETKEY3
        END IF
        MESSAGE$ = "CREATING " + DRIVE$ + DSKFIL$ + " - PLEASE WAIT. "
```

```
        CALL PRINTMESSAGE
        '! CLS
CHKBUFFER:
        PRINT #1, CHR$(27);
        '
        IF NOT EOF(1) THEN
            DL$(0) = INPUT$(LOC(1), #1)
            OPEN DRIVE$ + DSKFIL$ FOR OUTPUT AS #2   'OPEN SELECTED FILE
        END IF
        FOR I = 1 TO 10: KEY I, "": KEY(I) OFF: NEXT I
        TRIES = 0
        IF RIGHT$(DL$(0), 1) = CHR$(27) THEN GOTO NO.REPLY
        '
        IF DL$(0) = CHR$(33) THEN
            PRINT #2, DL$(0)
            GOTO SENDREQUEST
        ELSEIF DL$(0) <> CHR$(33) THEN
            GOTO CHKBUFFER
        END IF
        '*************REQUEST SELECTED REPORT FROM DATA LOGGER*****************
SENDREQUEST:
        PRINT #1, B$: CALL PAUSE(1!)
        DL$(1) = INPUT$(LOC(1), #1)
        TRIES = TRIES + 1: IF TRIES = 5 THEN GOTO NO.REPLY
        IF DL$(1) = "" THEN GOTO SENDREQUEST
        PRINT DL$(1); : PRINT #2, DL$(1);
        '
        I = 1
        DO
        IF NOT EOF(1) THEN
            DL$ = INPUT$(LOC(1), #1): PRINT DL$; : PRINT #2, DL$;
            I = I + 1
        ELSEIF EOF(1) THEN
            CALL DELAY(WAIT$)
            IF WAIT$ = "FINISHED" THEN EXIT DO
        END IF
        LOOP
        CLOSE #2                                    'CLOSES ALL FILES
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        MESSAGE$ = "DOWNLOAD COMPLETE.  PLEASE WAIT.  UPDATING DSKFILS.LST "
        CALL PRINTMESSAGE
        SHELL "DIR " + DRIVE$ + "*.L* | SORT>" + DRIVE$ + "DSKFILS.LST"
        MESSAGE$ = "DOWNLOAD COMPLETE.  PLEASE WAIT.  UPDATING GRAFILS.LST "
        CALL PRINTMESSAGE
        SHELL "DIR " + DRIVE$ + "*.LCT | SORT>" + DRIVE$ + "GRAFILS.LST"
        GOTO ENDCREATE
NO.REPLY:
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        MESSAGE$ = "NO REPLY FROM DATA LOGGER.  [F10] - MAIN MENU. "
        CALL PRINTMESSAGE
GETKEY4:
        CALL INKEYS(V$)
        IF V$ = F10$ THEN GOTO ENDCREATE: IF V$ <> F10$ THEN GOTO GETKEY4
ENDCREATE:
        IF DSKFIL$ <> "MENU" THEN PRINT #1, CHR$(4)
        ERASE DL$
        KEY OFF: FOR I = 1 TO 10: KEY I, "": NEXT I
        MESSAGE$ = "PLEASE WAIT.": CALL PRINTMESSAGE
        CLOSE
        END SUB

        '=============================================================================
        SUB DELAY (WAIT$)
        '*****************WAITS 4 SECONDS FOR END OF FILE*********************
        Start = TIMER: WAIT$ = "": S = 0
        IF DL$(1) = "" THEN
            GOTO END.DELAY
        ELSEIF DL$(1) <> "" AND EOF(1) THEN
TIME:
            IF NOT EOF(1) THEN GOTO END.DELAY
            S = TIMER - Start
            IF S > 4 THEN WAIT$ = "FINISHED"
            IF WAIT$ = "FINISHED" THEN GOTO END.DELAY
            GOTO TIME
        END IF
```

```
END.DELAY:
       END SUB

       ═══════════════════════════════════════════════════════════════════

       SUB DISKDRIVE
       '*****************CHANGES DISK DRIVE FOR DATA FILE STORAGE***************
       FOR I = 1 TO 10: KEY(I) OFF: KEY I, "": NEXT I
       KEY 1, "A:\" + CHR$(13)
       KEY 3, "B:\" + CHR$(13)
       KEY 5, "C:\" + CHR$(13)
       KEY 7, "C:\LOG\" + CHR$(13)
       KEY 10, DRIVE$ + CHR$(13)
       MESSAGE$ = "SELECT DRIVE => [F1] - A:\ [F3] - B:\ [F5] - C:\ "
       MESSAGE$ = MESSAGE$ + "[F7] - C:\LOG\  [F10] - MENU "
       CALL PRINTMESSAGE
       '*****************POSITION CURSOR ON MAIN MENU SCREEN******************
       POSITION = 40
       IF LEN(DRIVE$) > 3 THEN POSITION = 38 - DRVLEN
       IF LEN(DRIVE$) > 8 THEN POSITION = 37 - DRVLEN
       IF LEN(DRIVE$) > 10 THEN POSITION = 36 - DRVLEN
       LOCATE 7, POSITION, 0
       COLOR BG%, FG%, BOR%
       '
       CALL CLRKYBD
       INPUT "", DRIVE$: DRIVE$ = UCASE$(DRIVE$)
       IF LEN(DRIVE$) = 1 THEN DRIVE$ = DRIVE$ + ":\"
       IF LEN(DRIVE$) = 2 THEN DRIVE$ = DRIVE$ + "\"
       IF INSTR(DRIVE$, "\") > 0 THEN
           IF MID$(DRIVE$, LEN(DRIVE$), 1) <> "\" THEN DRIVE$ = DRIVE$ + "\"
       END IF
       COLOR FG%, BG%, BOR%
GETDRVINPUT:
       MESSAGE$ = " Do you want to update the directory files on drive "
       MESSAGE$ = MESSAGE$ + DRIVE$ + " (Y or N)": CALL PRINTMESSAGE
GETDRVINKEY:
       R$ = INKEY$: IF R$ = "" THEN GOTO GETDRVINKEY
       IF UCASE$(R$) = "N" THEN GOTO ENDDRVSUB
       IF UCASE$(R$) <> "Y" THEN GOTO GETDRVINPUT
       MESSAGE$ = "DRIVE SELECTED.  PLEASE WAIT.  UPDATING "
       MESSAGE$ = MESSAGE$ + DRIVE$ + "DSKFILS.LST "
       CALL PRINTMESSAGE
       SHELL "DIR " + DRIVE$ + "*.L* | SORT>" + DRIVE$ + "DSKFILS.LST"
       MESSAGE$ = "DRIVE SELECTED.  PLEASE WAIT.  UPDATING " + DRIVE$
       MESSAGFE$ = MESSAGE$ + "GRAFILS.LST "
       CALL PRINTMESSAGE
       SHELL "DIR " + DRIVE$ + "*.LCT | SORT>" + DRIVE$ + "GRAFILS.LST"
ENDDRVSUB:
       END SUB

       '═══════════════════════════════════════════════════════════════════
       SUB DOS (F8$, F10$)
       '*****************************EXIT TO DOS*******************************
       V$ = ""
       MESSAGE$ = "PRESS [F8] TO EXIT TO DOS OR [F10] TO RETURN TO THE MENU."
       CALL PRINTMESSAGE
CLBUFER:
       V$ = INKEY$: IF V$ = "" THEN GOTO CLBUFER
       IF V$ = F8$ THEN
           COLOR 7, 0, 0: CLS : SYSTEM
       ELSEIF V$ = F10$ THEN
           GOTO ENDDOS
       ELSE GOTO CLBUFER
       END IF
ENDDOS:
       END SUB

       '═══════════════════════════════════════════════════════════════════
       SUB FILE
       '***PROVIDES A LIST OF DATA FILES (UPDATE OPTIONAL) ON DEFAULT DRIVE***
       CLOSE : MENU$ = "DSKFILS.LST"
GETINPUT:
       MESSAGE$ = "DIRECTORY SELECTED.  Do you want"
       MESSAGE$ = MESSAGE$ + " to update the directory files? (Y or N)"
       CALL PRINTMESSAGE
```

```
GETINKEY:
       R$ = INKEY$: IF R$ = "" THEN GOTO GETINKEY
       IF UCASE$(R$) = "N" THEN GOTO CONTINUE
       IF UCASE$(R$) <> "Y" THEN GOTO GETINPUT
       MESSAGE$ = "DIRECTORY UPDATE REQUESTED - PLEASE WAIT.  "
       MESSAGE$ = MESSAGE$ + "UPDATING DSKFILS.LST "
       CALL PRINTMESSAGE
       SHELL "DIR " + DRIVE$ + "*.L* | SORT>" + DRIVE$ + "DSKFILS.LST"
       MESSAGE$ = "DIRECTORY UPDATE REQUESTED - PLEASE WAIT.  "
       MESSAGE$ = MESSAGE$ + "UPDATING GRAFILS.LST "
       CALL PRINTMESSAGE
       SHELL "DIR " + DRIVE$ + "*.LCT | SORT>" + DRIVE$ + "GRAFILS.LST"
CONTINUE:
       FOR I = 1 TO 10: KEY(I) OFF: NEXT I
       FOR I = 1 TO 9: KEY I, "": NEXT I
       MESSAGE$ = " to scroll through the files.  [RETURN] or"
       HEADER$ = " DATA FILE DIRECTORY "
       CALL SELCTFILE(FILENAME%)
       END SUB

       '-------------------------------------------------------------------
       SUB GRAPH
       '****GRAPHS THE MAXIMA OF A (C)OMPRESSED (T)IME HISTORY (*.LCT) FILE***
BEGINGRAPH:
       CLOSE                                        'CLOSES FILES AND COM PORTS
       MENU$ = "GRAFILS.LST"
       MESSAGE$ = "PLOT A GRAPHICS FILE SELECTED.  PLEASE WAIT."
       CALL PRINTMESSAGE
       CALIBPT.1$ = "": CALIBPT.2$ = "": CALIB$ = ""
       REDIM STDATE$(10), STTIME$(10), TAG$(10), MAXTAG(10)
       FOR K = 1 TO 9: STDATE$(K) = "": TAG$(K) = "": NEXT K: KEY OFF
       FOR I = 1 TO 10: KEY I, "": NEXT I
       MESSAGE$ = " to select a file.  [RETURN] to confirm."
       HEADER$ = " PLOT A GRAPHICS FILE "
       CALL SELCTFILE(FILENAME%)
       IF MESSAGE$ = "RETURN TO MENU" THEN GOTO RETURNTOMENU
       MESSAGE$ = "PLEASE WAIT - LOADING " + GRAFIL$ + "     "
       BG% = BG% + 16
       CALL PRINTMESSAGE
       OPEN DRIVE$ + GRAFIL$ FOR INPUT AS #2
       '*************INPUTS DATA LOGGING PARAMETERS FROM DATA FILE************
DATAINPUT:
       INPUT #2, A$
       IF INSTR(A$, "!") = 0 THEN
           GOTO DATAINPUT
       ELSE GOTO INCRLOOP
       END IF
INCRLOOP:
       ST$(1) = A$: FOR I = 2 TO 14: INPUT #2, A$: ST$(I) = A$: NEXT I
       CALIBPT.1$ = ST$(5): CALIBPT.2$ = ST$(6)
       V = INSTR(ST$(6), "=")
       W = INSTR(V, ST$(6), ".")
       X = INSTR(W, ST$(6), " ")
       CALIB$ = MID$(ST$(6), V + 1, X - V + 2)
       IF ASC(RIGHT$(CALIB$, 1)) > 47 THEN GOTO EXPONENT ELSE GOTO NO.EXPONENT
EXPONENT:
       U = INSTR(CALIB$, ".")
       V = INSTR(U, CALIB$, " ")
       CALIBLEFT$ = LEFT$(CALIB$, V - 1)
       CALIBRIGHT$ = RIGHT$(CALIB$, LEN(CALIB$) - (V + 1))
       CALIB$ = CALIBLEFT$ + "E" + CALIBRIGHT$
NO.EXPONENT:
       Z = 1
       C = INSTR(1, ST$(14), " ")
       C = INSTR(C + 1, ST$(14), " ")
       C = INSTR(C + 1, ST$(14), " ")
       D = INSTR(ST$(14), "  ")
       STDATE$(Z) = LEFT$(ST$(14), C)
       STTIME$(Z) = MID$(ST$(14), C + 1, D - C)
       TAG$(Z) = RIGHT$(ST$(14), 2)
       LINE INPUT #2, A$       'INPUTS DATA FROM FILE TO LOCATE MAXIMUM VALUES
       J = 1
       V = INSTR(A$, ".")
       V = INSTR(V + 1, A$, ".")
       V = INSTR(V + 1, A$, ".")
```

```
        W = INSTR(V + 1, A$, " ")
        IF W = 0 THEN X = LEN(A$) - V
        IF V < W AND W <= LEN(A$) THEN X = W - V
        Y = V + X - 8
        DT(Z, J) = MID$(A$, Y, 8)
EOGRAFCHK:
        IF EOF(2) THEN GOTO PLOTDATA
        LINE INPUT #2, A$: IF INSTR(A$, ".") = 0 THEN GOTO NO.DATAPTS
        J = J + 1
        DT(Z, J) = MID$(A$, Y, 8)
        MAXTAG(Z) = J
        GOTO EOGRAFCHK
NO.DATAPTS:
        IF INSTR(A$, "I") > 0 THEN GOTO INCOMPLETE
        IF INSTR(A$, STDATE$(1)) > 0 THEN GOTO NEWTAGCHK
        GOTO EOGRAFCHK
NEWTAGCHK:
        IF RIGHT$(A$, 2) <> TAG$(Z) THEN GOTO GETNEWTAG ELSE GOTO EOGRAFCHK
GETNEWTAG:
        Z = Z + 1: J = 0
        STDATE$(Z) = LEFT$(A$, C)
        STTIME$(Z) = MID$(A$, C + 1, D - C)
        TAG$(Z) = RIGHT$(A$, 2)
        GOTO EOGRAFCHK
INCOMPLETE:
        J = J + 1
        DT(Z, J) = "0"
        GOTO EOGRAFCHK
        '******************PLOT MAXIMA FROM SELECTED DATA FILE******************
PLOTDATA:
        KEY OFF: KEY 10, ""
        SCREEN 2: CLS : LOCATE 20, 10
        PRINT "Y-AXIS:  CALIBRATION                          X-AXIS: TIME ";
        LOCATE 3, 2: PRINT "TEST STARTING DATE "; STDATE$(1);
        PRINT "              TEST STARTING TIME "; STTIME$(1)
        LOCATE 21, 10: PRINT CALIBPT.2$;
        LOCATE 21, 48: PRINT "        ";
        LOCATE 22, 10: PRINT CALIBPT.1$;
        LOCATE 22, 53: PRINT "SECONDS";
        VIEW (8, 30)-(600, 140), , 7
        TAG$(10) = "END"
        FOR J = 1 TO 10
            KEY(J) OFF
            KEY J, " " + TAG$(J) + "      " + CHR$(13)
            IF TAG$(J) = "" THEN KEY J, " EMPTY" + CHR$(13)
        NEXT J
        KEY ON
        V = LEN(DSKFIL$): V = 71 - V
        LOCATE 1, V: PRINT DSKFIL$
GETPLOTAG:
        LOCATE 1, 2: INPUT "SELECT A FUNCTION KEY TO PLOT A TAG#", TG$
        IF TG$ = "END" THEN GOTO ENDGRAPH
        I = VAL(TG$): CLS
        EOD = MAXTAG(I)
        CALIB = VAL(CALIB$)
        ADJUST% = 0
ADJUSTCALIB:
        IF CALIB < 100 THEN
            CALIB = CALIB * 10
            ADJUST% = ADJUST% + 1
            GOTO ADJUSTCALIB
        END IF
        FOR J = 1 TO EOD
            DAT(1, J) = VAL(DT(I, J))
            DAT(1, J) = DAT(1, J) * 10 ^ (ADJUST% + VAL(CALIBRIGHT$))
        NEXT J
        LOCATE 21, 54: PRINT "       ";
        LOCATE 21, 54: PRINT EOD;
        IF L = 0 THEN L = L + 1
        WINDOW (1, 1)-(EOD, CALIB)
        FOR N = 1 TO EOD
            X = N: Y = DAT(1, N): PSET (X, Y)
            IF X = 1 THEN GOTO INCRN
            LINE (X - 1, DAT(1, N - 1))-(X, Y)
            Z = Y
```

```
INCRN:
        NEXT N
        GOTO GETPLOTAG
ENDGRAPH:
        KEY OFF: SCREEN 0, 1: CLOSE
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        GOTO BEGINGRAPH
RETURNTOMENU:
        ERASE DT: KEY OFF: SCREEN 0, 1
        COLOR FG%, BG%, BOR%: CLOSE
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        END SUB
        '-------------------------------------------------------------------------
        SUB HELP
        '************PROVIDES A SCREEN DISPLAY OF A HELP TEXT FILE*************
        MESSAGE$ = "HELP SELECTED.  PLEASE WAIT.  LOADING HELPFILE.LXT."
        BG% = BG% + 16: CALL PRINTMESSAGE: COLOR FG%, BG%, BOR%
        DSKFIL$ = "HELPFILE.LXT"
        CALL VIEWER(OPENFIL$)
        COLOR FG%, BG%, BOR%: CLS
        END SUB

        '-------------------------------------------------------------------------
        SUB INKEYS (V$)
        '******************WAITS FOR KEYBOARD INPUT FROM USER*******************
INKEY:
        V$ = INKEY$: IF V$ = "" THEN GOTO INKEY
        END SUB

        '-------------------------------------------------------------------------
        SUB MAINMENU
        '************************PRINTS THE MAIN MENU*****************************
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        MESSAGE$ = "PLEASE WAIT.": CALL PRINTMESSAGE
        '
        DRVLEN = LEN(DRIVE$): PAD = 26 - DRVLEN
        PADLEFT = (PAD \ 2) - 1: PADRIGHT = (PAD \ 2) + (PAD MOD 2) + 1
        MLEFT$ = STRING$(PADLEFT - 3, 32): MRIGHT$ = STRING$(PADRIGHT - 3, 32)
        CLOSE #2
        '
        COLOR BG%, FG%, BOR%
        LOCATE 3, 22: PRINT "*             MAIN MENU             *"
        LOCATE 5, 23: PRINT "[F1]";
        COLOR FG%, BG%, BOR%: PRINT "          HELP            ";
        COLOR BG%, FG%, BOR%: PRINT "[F1]"
        LOCATE 7, 23: PRINT "[F2]";
        COLOR FG%, BG%, BOR%: PRINT MLEFT$; "DRIVE "; DRIVE$; MRIGHT$;
        COLOR BG%, FG%, BOR%: PRINT "[F2]"
        LOCATE 9, 23: PRINT "[F3]";
        COLOR FG%, BG%, BOR%: PRINT "   DATA FILE DIRECTORY    ";
        COLOR BG%, FG%, BOR%: PRINT "[F3]"
        LOCATE 11, 23: PRINT "[F4]";
        COLOR FG%, BG%, BOR%: PRINT "  CREATE A NEW DATA FILE  ";
        COLOR BG%, FG%, BOR%: PRINT "[F4]"
        LOCATE 13, 23: PRINT "[F5]";
        COLOR FG%, BG%, BOR%: PRINT "  VIEW AN EXISTING FILE   ";
        COLOR BG%, FG%, BOR%: PRINT "[F5]"
        LOCATE 15, 23: PRINT "[F6]";
        COLOR FG%, BG%, BOR%: PRINT "  PRINT AN EXISTING FILE  ";
        COLOR BG%, FG%, BOR%: PRINT "[F6]"
        LOCATE 17, 23: PRINT "[F7]";
        COLOR FG%, BG%, BOR%: PRINT "   PLOT A GRAPHICS FILE   ";
        COLOR BG%, FG%, BOR%: PRINT "[F7]"
        LOCATE 19, 23: PRINT "[F8]";
        COLOR FG%, BG%, BOR%: PRINT "      EXIT TO MS-DOS      ";
        COLOR BG%, FG%, BOR%: PRINT "[F8]"
        MESSAGE$ = "PRESS A FUNCTION KEY TO SELECT AN ACTIVITY. "
        CALL PRINTMESSAGE
        END SUB

        '-------------------------------------------------------------------------
        SUB PAUSE (SEC) STATIC
        '***************PAUSES PROGRAM FOR A SPECIFIED TIME PERIOD************
        CONST SecInDay = 24& * 60& * 60&
        LoopFinish = TIMER + SEC
```

```
        IF LoopFinish > SecInDay THEN
            LoopFinish = LoopFinish - SecInDay
            DO WHILE TIMER > LoopFinish
            LOOP
        END IF
        DO WHILE TIMER < LoopFinish
        LOOP
        END SUB

        '=============================================================================
        SUB PRINTMESSAGE
        '*************PROVIDES APPROPRIATE ONSCREEN HELP MESSAGES**************
        MESSAGELEFT$ = "": MESSAGERIGHT$ = ""
        MESSAGE = LEN(MESSAGE$)
        PAD = 78 - MESSAGE
        PADLEFT = (PAD \ 2): PADRIGHT = (PAD \ 2) + (PAD MOD 2)
        MESSAGELEFT$ = STRING$(PADLEFT, CHR$(32))
        MESSAGELEFT$ = MESSAGELEFT$ + MESSAGE$
        MESSAGERIGHT$ = STRING$(PADRIGHT, CHR$(32))
        MESSAGE$ = MESSAGELEFT$ + MESSAGERIGHT$
                '
        COLOR BG%, FG%, BOR%: LOCATE 24, 2, 0: PRINT CLEARMESSAGE$;
        LOCATE 24, 2: PRINT MESSAGE$;
        FG% = 7: BG% = 1: BOR% = 9: COLOR FG%, BG%, BOR%
        LOCATE 1, 2, 0
        END SUB

        '=============================================================================
        SUB PRNT
        '***************PROVIDES A PRINTOUT OF THE SELECTED FILE***************
BEGINPRINT:
        CLOSE : MENU$ = "DSKFILS.LST"
        MESSAGE$ = " PRINT AN EXISTING FILE SELECTED.  PLEASE WAIT."
        CALL PRINTMESSAGE
        HEADER$ = " PRINT AN EXISTING FILE "
        MESSAGE$ = " to select a file.  [RETURN] to confirm. "
        CALL SELCTFILE(FILENAME%)
        IF MESSAGE$ = "RETURN TO MENU" GOTO ENDPRNT
        MESSAGE$ = " PRINTING " + DSKFIL$ + ".  PLEASE WAIT. "
        CALL PRINTMESSAGE
        OPEN DRIVE$ + DSKFIL$ FOR INPUT AS #2
        LPRINT DSKFIL$
ENDOFFILECHK:
        IF EOF(2) THEN GOTO ENOFFILE
        A$ = INPUT$(1, #2): LPRINT A$;
        GOTO ENDOFFILECHK
ENOFFILE:
        CLOSE #2
        GOTO BEGINPRINT
ENDPRNT:
        END SUB

        '=============================================================================
        SUB SELCTFILE (FILENAME%)
        '*****************PROVIDES A MENU FOR DATA FILE SELECTION**************
        DIM FILENAME%(150): DIM FILENAME$(150)
        FILENUM% = 1: ROW% = 4
        FILENAME%(11) = FG%: FILENAME%(1) = BG%
        FOR I = 2 TO 10: FILENAME%(I) = FG%: FILENAME%(I + 10) = BG%: NEXT I
        FOR J% = 1 TO 150: FILENAME$(J%) = "": NEXT J%
        COLOR FG%, BG%, BOR%
        OPEN "I", #2, DRIVE$ + MENU$
        CALL CLEARSCREEN(FG%, BG%, BOR%)
        LOCATE 3, 27: COLOR BG%, FG%, BOR%: PRINT HEADER$: COLOR FG%, BG%, BOR%
        '***********READ IN FILENAMES FROM GRAFILS.LST OR DSKFILS.LST**********
EOFCHK:
        IF EOF(2) THEN GOTO ENDOFLE
        INPUT #2, A$
        IF MID$(A$, 9, 2) = " L" THEN
            FILENAME$(FILENUM%) = A$
            FILENUM% = FILENUM% + 1
            GOTO EOFCHK
        ELSE LOCATE ROW%, 24
            PRINT A$
            ROW% = ROW% + 1
```

```
            GOTO EOFCHK
        END IF
ENDOFLE:
        I% = 1
        COLOR FG%, BG%, BOR%
        MESSAGE$ = " Press " + CHR$(25) + " or " + CHR$(24) + MESSAGE$
        MESSAGE$ = MESSAGE$ + " [F10] - MAIN MENU ": CALL PRINTMESSAGE
REPRNT:
        FOR J% = 1 TO 10
            LOCATE J% + 10, 20
            COLOR FILENAME%(J%), FILENAME%(J% + 10)
            PRINT FILENAME$(J%)
        NEXT J%
        KEY OFF: FOR I = 1 TO 10: KEY 1, "": NEXT I
GETKEY5:
        CALL INKEYS(V$)
        IF V$ = F10$ THEN
            MESSAGE$ = "RETURN TO MENU"
            GOTO ENDSELCTFILE
        ELSEIF V$ = UpARROW$ THEN
            GOTO SCRLUP
        ELSEIF V$ = DnARROW$ THEN
            GOTO SCRLDWN
        ELSEIF V$ = CHR$(13) THEN
            GOTO SELCTFILE
        ELSE GOTO GETKEY5
        END IF
        STOP
SELCTFILE:
        DSKFIL$ = LEFT$(FILENAME$(I%), 8)
        V% = INSTR(DSKFIL$, CHR$(32)): IF V% = 0 THEN V% = V% + 9
        DSKFIL$ = LEFT$(DSKFIL$, V% - 1) + "." + MID$(FILENAME$(I%), 10, 3)
        GRAFIL$ = DSKFIL$: CLOSE #2
        COLOR FG%, BG%, BOR%
        GOTO ENDSELCTFILE
SCRLUP:
        I% = I% - 1
        IF I% < 1 THEN
            I% = I% + 1
            GOTO REPRNT
        ELSEIF I% < 10 THEN
            FILENAME%(I%) = BG%: FILENAME%(I% + 1) = FG%
            FILENAME%(I% + 10) = FG%: FILENAME%((I% + 10) + 1) = BG%
            GOTO REPRNT
        ELSE X% = I% - 10
            FILENAME%(I%) = BG%
            FILENAME%(I% + 10) = FG%
            FOR J% = I% TO (I% - 9) STEP -1
                LOCATE J% - X% + 10, 20
                COLOR FILENAME%(J%), FILENAME%(J% + 10): PRINT FILENAME$(J%)
            NEXT J%
            GOTO GETKEY5
        END IF
SCRLDWN:
        I% = I% + 1
        IF I% > FILENUM% - 1 THEN
            I% = I% - 1
            GOTO GETKEY5
        ELSEIF I% > 10 THEN
            X% = I% - 10
            FILENAME%(I% - 1) = BG%: FILENAME%((I% - 1) - 1) = FG%
            FILENAME%((I% - 1) + 10) = FG%: FILENAME%((I% - 1 + 10) - 1) = BG%
            FOR J% = I% TO (I% - 9) STEP -1
                LOCATE J% - X% + 10, 20
                COLOR FILENAME%(J% - 1), FILENAME%((J% - 1) + 10)
                PRINT FILENAME$(J%)
            NEXT J%
            GOTO GETKEY5
        ELSE FILENAME%(I%) = BG%
            FILENAME%(I% - 1) = FG%
            FILENAME%(I% + 10) = FG%
            FILENAME%((I% + 10) - 1) = BG%
            GOTO REPRNT
        END IF
```

```
ENDSELCTFILE:
       CLOSE
       END SUB

       '=============================================================================
       SUB VIEWER (OPENFILE$)
       '*************OUTPUTS TEXT FILES TO SCREEN FOR USER VIEWING************
       DEF SEG = &H40
       STATUSBYTE = PEEK(&H17)
       NUM% = 1: ROW% = 1: FILENUM% = 1
       FOR J% = 1 TO 1600: VT(J%) = "": NEXT J%
       KEY 10, CHR$(27)
       OPEN "I", #2, DRIVE$ + DSKFIL$
Diskinput:
       ST$ = ""
       IF EOF(2) THEN GOTO Startprint
       LINE INPUT #2, ST$: IF INSTR(ST$, CHR$(12)) > 0 THEN GOTO Diskinput
Linestring:
       VT(FILENUM%) = RTRIM$(ST$)
       FILENUM% = FILENUM% + 1
       ST$ = ""
       GOTO Diskinput
Startprint:
       CLS : I% = 1
       FOR J% = 1 TO 22: LOCATE J%, 1: PRINT VT(J%): NEXT J%
       MESSAGE$ = " PRESS " + CHR$(25) + " " + CHR$(24) + " [Home] "
       MESSAGE$ = MESSAGE$ + "[End] [PgUp] or [PgDn] keys to VIEW file. "
       MESSAGE$ = MESSAGE$ + " [F10] - END. "
       CALL PRINTMESSAGE
STATUSBYTE:
INKEYS:
       STATUSBYTE = PEEK(&H17)
       V$ = INKEY$: IF V$ = "" THEN GOTO INKEYS
       POKE &H17, 0
       IF V$ = HOME$ THEN
           I% = 0
       ELSEIF V$ = UpARROW$ THEN
           I% = I% - 1
           IF I% < 0 THEN I% = 0
       ELSEIF V$ = PgUp$ THEN
           I% = I% - 21
           IF I% < 0 THEN I% = 0
       ELSEIF V$ = END$ THEN
           I% = FILENUM% - 21
           IF I% < 0 THEN I% = 0
       ELSEIF V$ = DnARROW$ THEN
           I% = I% + 1
           IF I% > FILENUM% - 21 THEN I% = FILENUM% - 21
           IF I% < 0 THEN I% = 0
       ELSEIF V$ = PgDn$ THEN
           I% = I% + 21
           IF I% > FILENUM% - 21 THEN I% = FILENUM% - 21
           IF I% < 0 THEN I% = 0
       ELSEIF V$ = CHR$(27) THEN
           POKE &H17, STATUSBYTE
           CLS
           GOTO ENDVIEWER
       ELSE GOTO STATUSBYTE
       END IF
       '
       MESSAGE$ = " PRESS " + CHR$(25) + " " + CHR$(24) + " [Home] "
       MESSAGE$ = MESSAGE$ + "[End] [PgUp] or [PgDn] keys to VIEW file. "
       MESSAGE$ = MESSAGE$ + " [F10] - END. "
       CALL PRINTMESSAGE
       FOR J% = 1 TO 22: LOCATE J%, 1: PRINT VT(J% + I%): NEXT J%
       GOTO STATUSBYTE
ENDVIEWER:
       COLOR FG%, BG%, BOR%: CLS : CALL CLEARSCREEN(FG%, BG%, BOR%)
       KEY 10, ""
       END SUB
```

```
        '_____________________________________________________________________
        SUB VIEWFILE
        '******************OPENS SELECTED FILE FOR VIEW OPTION*****************
BEGINVIEWFILE:
        CLOSE : MENU$ = "DSKFILS.LST"
        MESSAGE$ = " VIEW AN EXISTING FILE SELECTED.  PLEASE WAIT."
        CALL PRINTMESSAGE
        HEADER$ = " VIEW AN EXISTING FILE "
        MESSAGE$ = " to select a file.  [RETURN] to confirm."
        CALL SELCTFILE(FILENAME%)
        IF MESSAGE$ = "RETURN TO MENU" GOTO ENDVIEWFILE
        MESSAGE$ = "PLEASE WAIT - LOADING " + DSKFIL$ + "    "
        BG% = BG% + 16
        CALL PRINTMESSAGE
        OPENFIL$ = DRIVE$ + DSKFIL$
        CALL VIEWER(OPENFIL$)
        GOTO BEGINVIEWFILE
ENDVIEWFILE:
        COLOR FG%, BG%, BOR%: CLS
        ERASE VT
        END SUB
```