NASA Contractor Report 178339

HALOE TEST AND EVALUATION SOFTWARE

W. Edmonds
S. Natarajan

ST Systems Corporation (STX)
28 Research Drive
Hampton, VA 23666

Contract NAS1-18022

June 1987

(NASA-CR-178339) HALOE TEST AND EVALUATION SOFTWARE (ST Systems Corp.) 142 p Avail: NTIS HC A07/MF A01 CSCL 09B

N87-26575

Unclas
G3/61 0090709

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

TABLE OF CONTENTS

	Page
Abstract	iii
SECTION 1 - INTRODUCTION	1-1
SECTION 2 - HALOE	2-1
SECTION 3 - SOFTWARE DEVELOPMENT	3-1
SECTION 4 - DATA ACQUISITION	4-1
SECTION 5 - SYSTEM DESIGN	5-1
SECTION 6 - DATA REDUCTION & ANALYSIS	6-1
APPENDIX A - HARP	A-1
APPENDIX B - HPLOT	B-1
APPENDIX C - SPECRES	C-1
APPENDIX D - SPECPLT	D-1
APPENDIX E - MONITOR	E-1

Abstract

Computer programming, system development and analysis efforts during this contract were carried out in support of the Halogen Occultation Experiment (HALOE) at NASA/Langley. Support in the major areas of data acquisition and monitoring, data reduction and system development are described along with a brief explanation of the HALOE project. Documented listings of major software are located in the appendix.

SECTION 1 - INTRODUCTION

Support of the Halogen Occultation Experiment (HALOE) during this level-of-effort contract consisted of computer programming, system design, data acquisition, data reduction and data analysis efforts.

HALOE is briefly described in Section 2 of this final report. Section 3 covers computer programming developments. Section 4 describes data acquisition support. System design is reviewed in Section 5, and Section 6 covers data reduction and data analysis support. Listings of programs are in the appendix.

SECTION 2 - HALOE

The objective of the Halogen Occultation Experiment is to measure trace constituents of the upper atmosphere to determine the mechanism of ozone depletion. The HALOE instrument was designed to measure these gases using a solar occultation technique. Utilizing four gas correlation and four bolometer channels, the HALOE instrument will view the sun during orbital sunrise and sunset events to measure the spectral occultation caused by ozone, water vapor, nitrogen dioxide, carbon dioxide, hydrogen fluoride, hydrogen chloride, methane and nitric oxide. Knowledge of the distribution of these gases on a global level over a long period of time should provide the means to better understand the mechanism of ozone depletion. HALOE will be one of Ten instruments on UARS (Upper Atmosphere Research Satellite) currently scheduled for launch aboard the space shuttle from KSC in 1991.

SECTION 3 - SOFTWARE DEVELOPMENT

A number of computer programs were developed under this contract to support the testing and characterization of the HALOE instrument. A variety of computer systems and languages were used to accomplish these tasks. Computer hardware included HP-1000, IBM-XT and CDC Cyber computers. Computer languages utilized were FORTRAN, PASCAL, FORTH and IBM assembler.

The HALOE black body life test was supported with the development of a program called "HPLOT" on the CDC NOS facility. "HPLOT" (written in FORTRAN 5) plots the various black body parameters against the PRT (platinum resistance thermometer) and tabulates daily averages of all the parameters (see appendix for program listing and sample output).

"HARP" was developed on the HP1000 in FORTRAN to aid in the analysis of HALOE test data tapes. HARP will process data directly from tape or from disc files previously derived from test tapes. Data windowing features allow the user to select time segments for processing and/or archival to disc. Annotate records can be searched in a forward or reverse direction to locate significant events for processing. Plot files containing selected parameters can be created for another program "UPLOT" to plot on the HP pen plotter, or on the CRT. A statistics option allows the user to select parameters for statistical analysis and tabulation.

Using Turbo Pascal on an IBM-XT fitted with a Lab Master card, software was developed to acquire data from the HALOE GCETS

(Gas Correlation Electronic Test Set).

Several versions of this software were created to acquire data for IFOV, balance-linearity, spectral response and NO noise tests. Data acquired by these programs was written to disc files. Plotter programs were developed to generate plots of the data on an HP pen plotter connected to an IEEE-488 card in the IBM-XT. LaRCNET was used to transfer some of these data files to NOS for analysis by the HALOE science team.

During this contract, work was begun on software which will monitor the HALOE data stream on a real time basis. Data will be transferred from the HP1000 to the IBM-XT over an IEEE-488 bus (HPIB) and displayed on a color monitor in color coded form. Red or yellow will indicate out-of-limit conditions, while green or white will indicate acceptable values. The computer language "FORTH" was used to develop the communications between the HP1000 and the IBM, and Turbo Pascal was used to write the display software for the IBM. Listings and sample output from some of the significant pieces of software are contained in the appendix to this report.

SECTION 4 - DATA ACQUISITION

Data acquisition support activities were performed under this contract for the following specialized tests of the HALOE instrument: IFOV, balance-linearity, spectral response and NO noise testing.

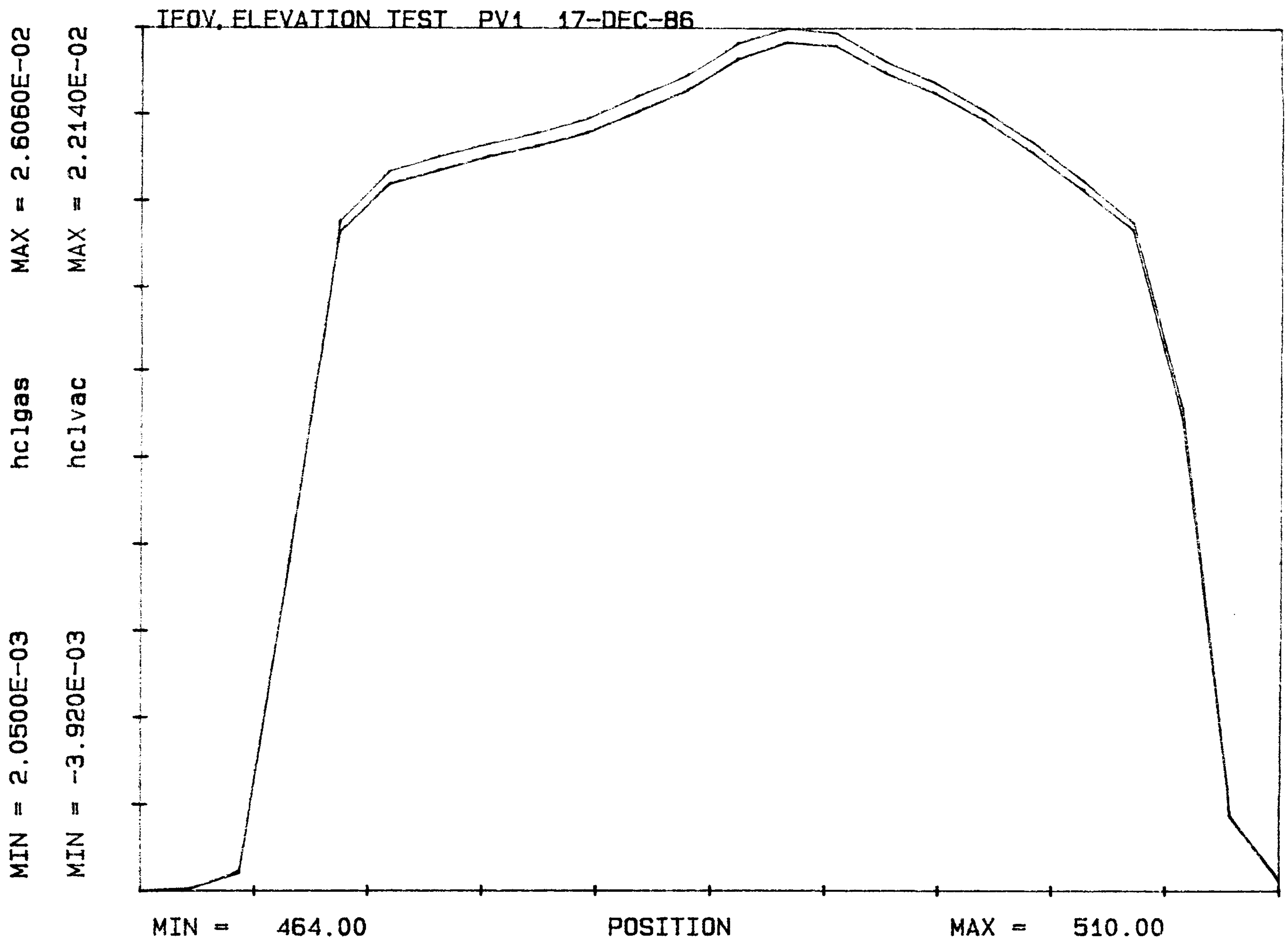
For the IFOV tests, measurements were made in azimuth and elevation for the gas correlation channels: HCl, HF, CH_4, NO (both gas and vacuum) and for the bolometer channels: H_2O, CO_2, NO_2, O_3. Results were tabulated and plotted immediately following each elevation or azimuth test (see sample plot).

Balance-linearity test data were acquired in a similar manner. To determine the linearity of each channel, correlation coefficients were calculated and printed out immediately following each test. Test data were also sent to the CDC NOS facility for further evaluation. Data was acquired for these tests using software developed under this contract (described elsewhere in this document) on an IBM-XT fitted with a Tecmar Lab Master data acquisition card.

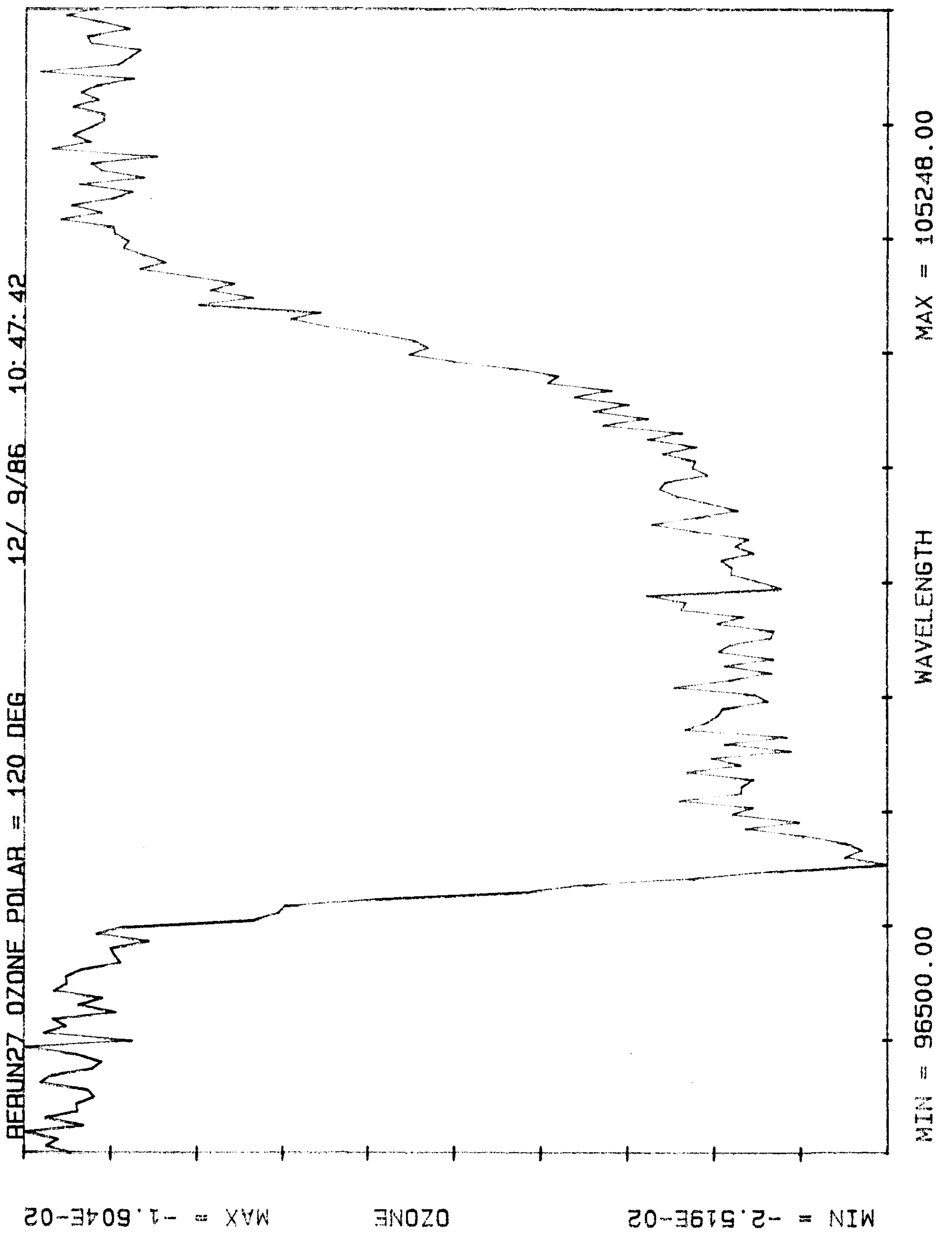
NO noise testing was accomplished by monitoring the NO channels (vac. & gas) during a series of manipulations of the instrument and associated equipment in the clean room.

Data acquisition efforts for the spectral response tests involved the use of additional software and hardware. In addition to the Lab Master software and hardware for data acquisition from the GCETS, the IBM-XT needed to communicate with the CD2A compudrive. This RS232 communications allowed the IBM-

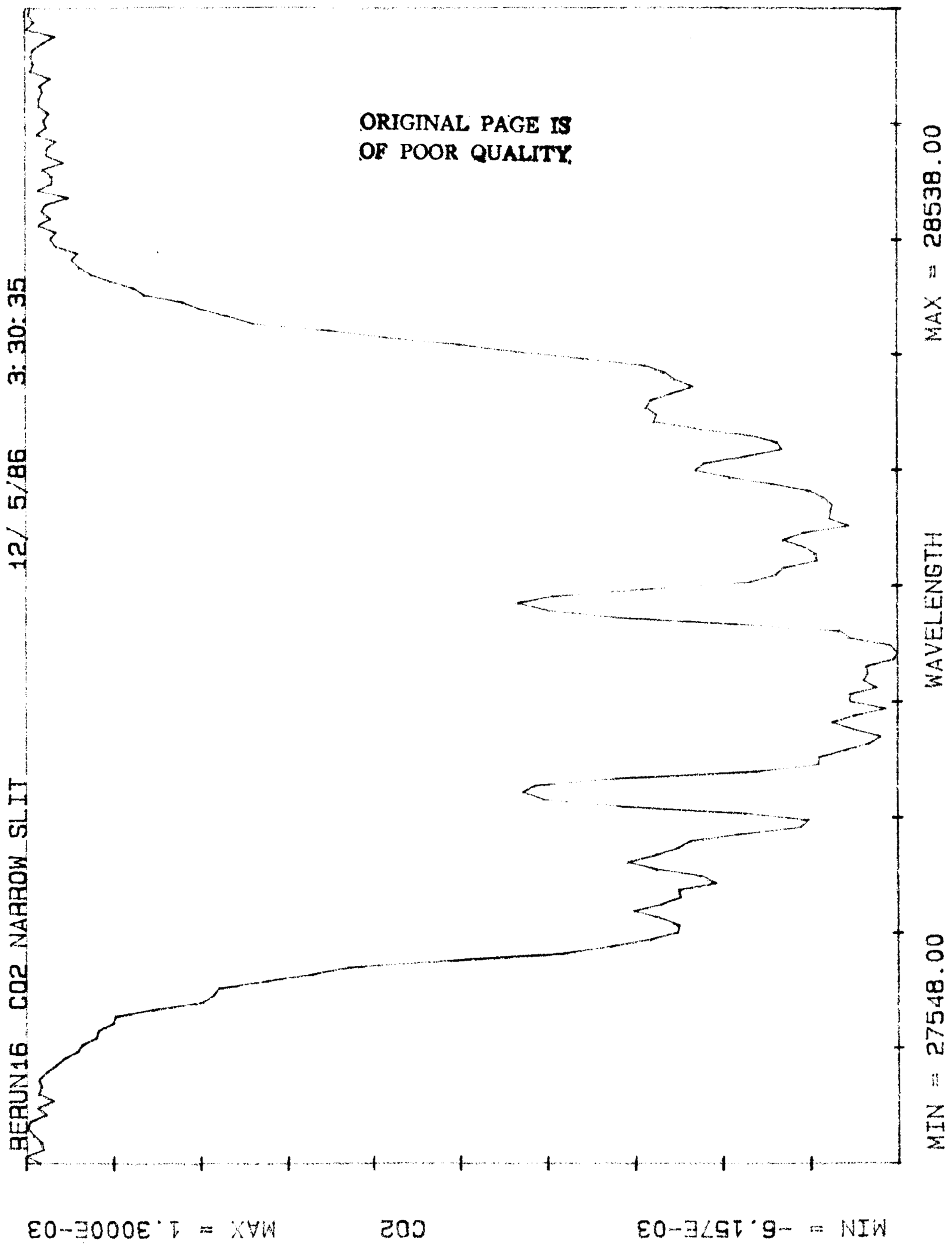
XT to detect when the spectrometer changed wavelength. Each step in wavelength was then used to trigger the acquisition of data from the GCETS. Data, including the wavelength, was then saved to disc for immediate processing after each spectral test. Plots were generated with the IBM and an HP pen plotter. The data was also sent to ACD using LaRCnet for further study by the science team (see sample spectral response plot and the data acquisition block diagram which follow).


IFOV, ELEVATION TEST PV1 17-DEC-86
MAX = 2.6060E-02
MAX = 2.2140E-02
hclgas
hclvac
MIN = 2.0500E-03
MIN = -3.920E-03
MIN = 464.00
POSITION
MAX = 510.00

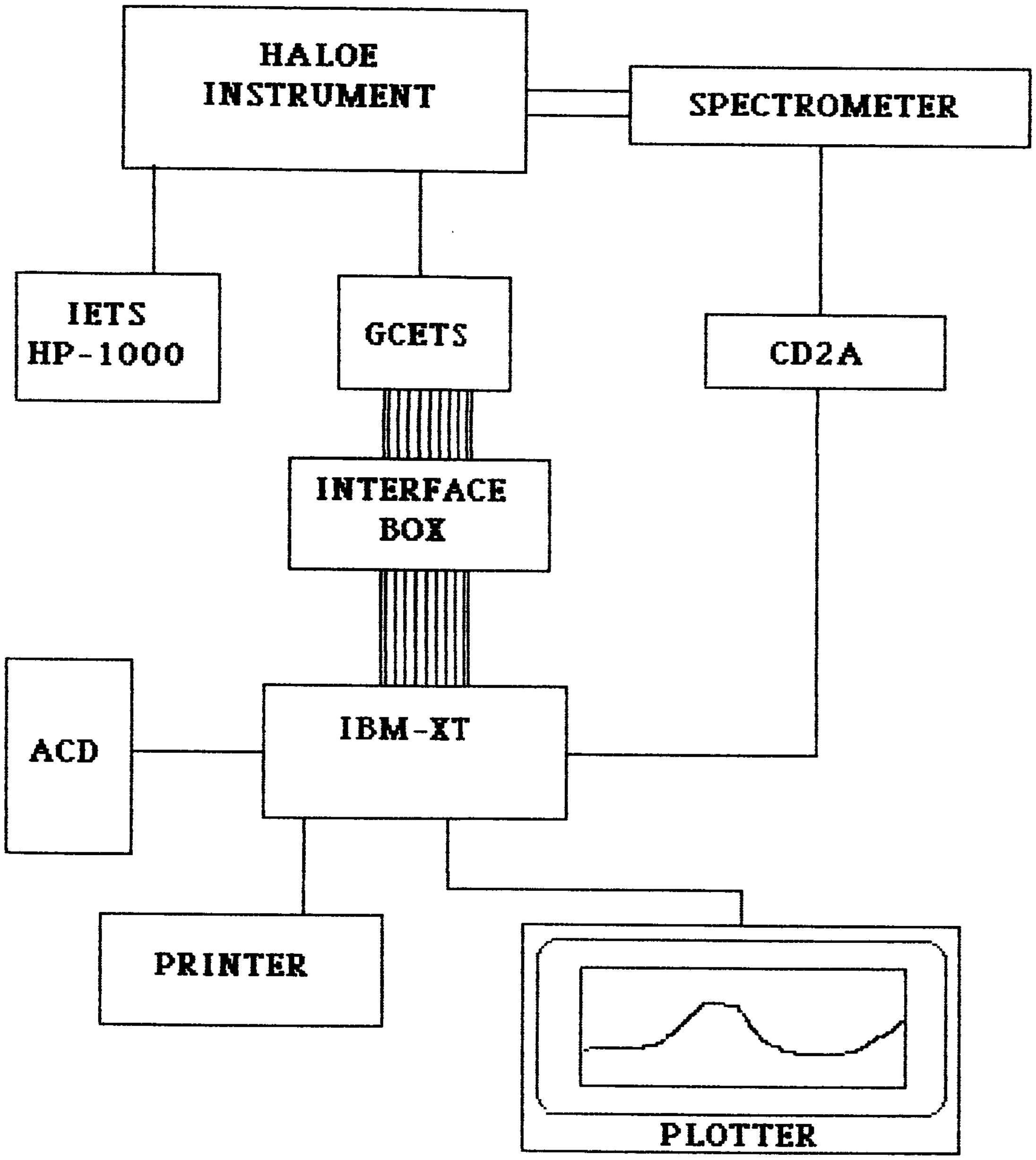
ORIGINAL PAGE IS
OF POOR QUALITY



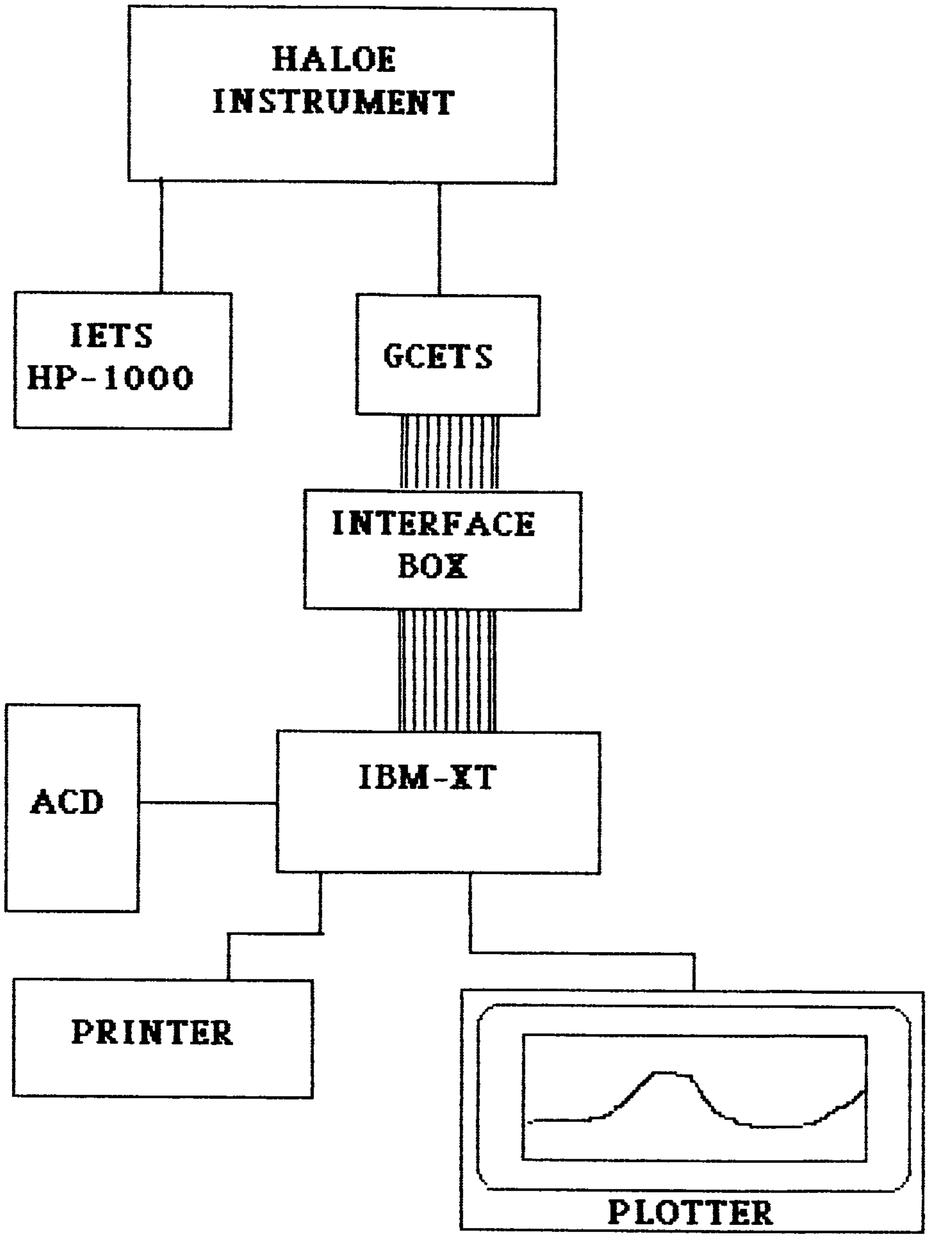
ORIGINAL PAGE IS
OF POOR QUALITY



HALOE SPECTRAL RESPONSE DATA ACQUISITION SET-UP



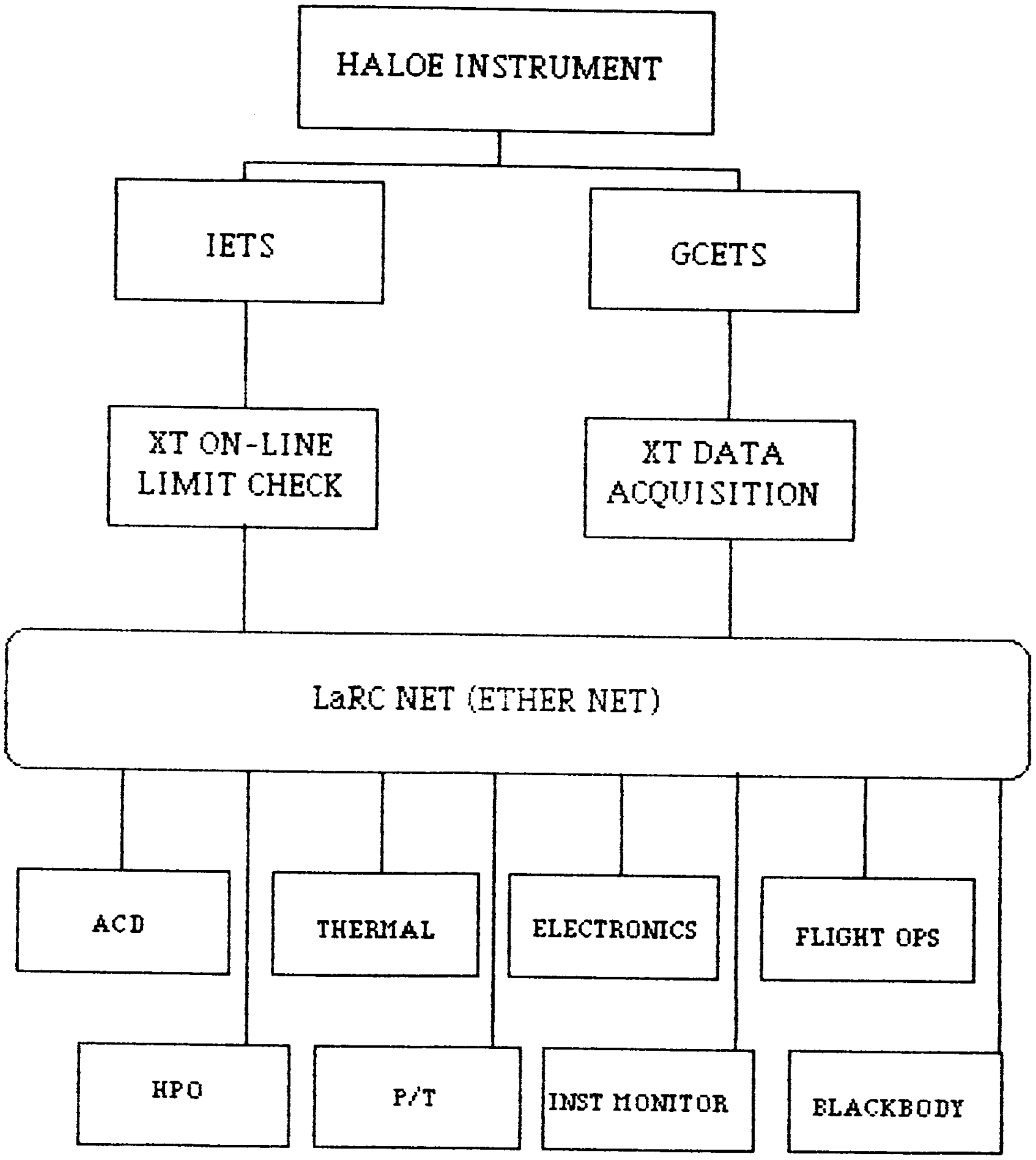
HALOE IFOV & BALANCE-LINEARITY DATA ACQUISITION SET-UP



SECTION 5 - SYSTEM DESIGN

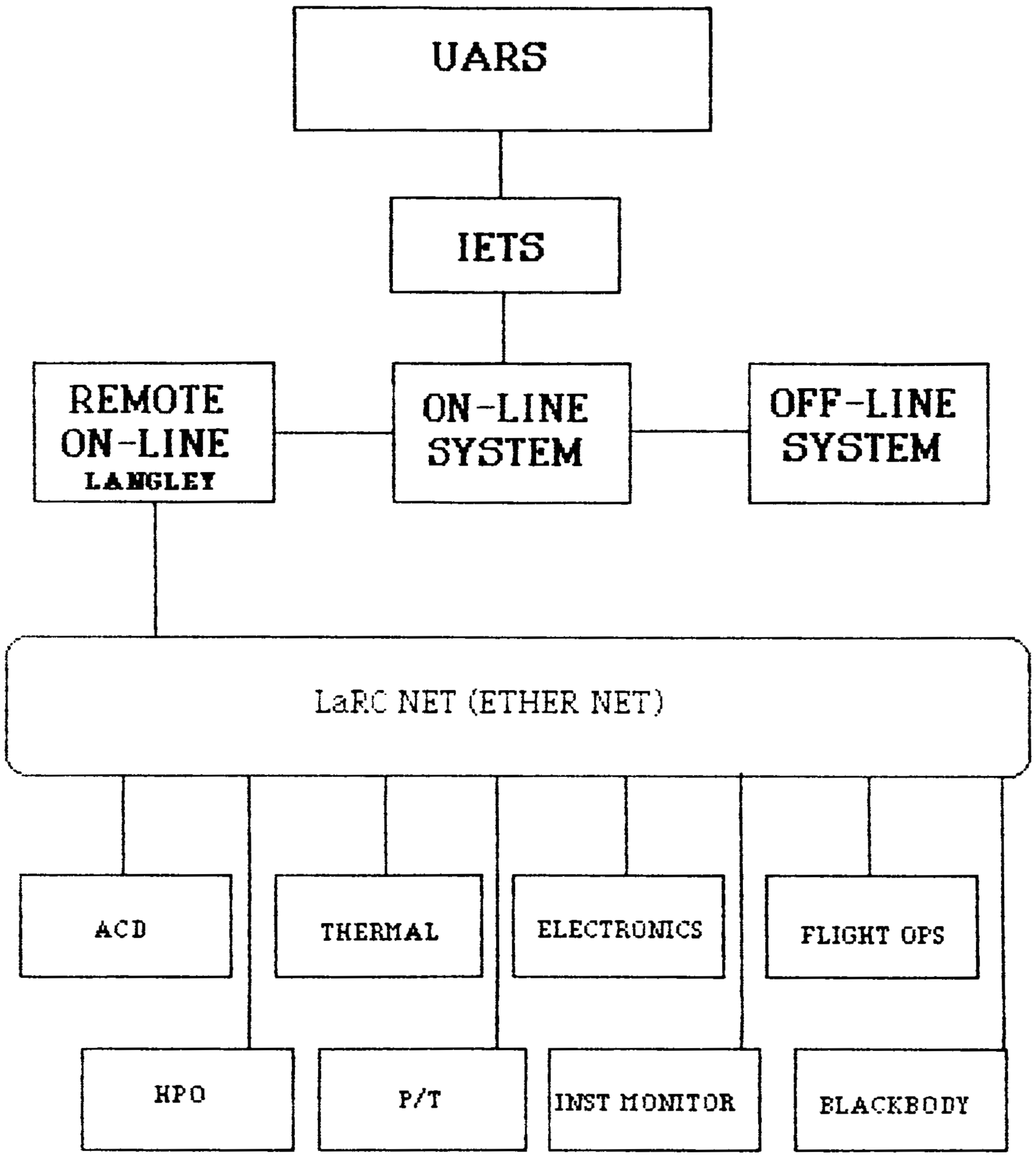
Considerable effort was made during this contract to design and implement a system for quick-look data reduction during the remaining testing at Langley and during satellite integration and testing when HALOE is installed on UARS (Upper Atmosphere Research Satellite). The attached block diagrams show the hardware configuration which was proposed and which will be assembled, tested and utilized under a subsequent contract. Some of the software requirements for this system were partially completed during this contract and will be finished early in the new contract period. Other system development work was done in the evaluation of an automated test control system. Although insufficient time and resources were available to fully design and implement such a system, a useful subset was designed and implemented on the HP1000 IETS. This system involved the use of FORTH (a computer language). FORTH facilitated the construction of commands and combinations of commands which could be issued to the HALOE instrument during tests. (These efforts were done under a separate STX contract and were accomplished by Milton Fabert).

HALOE QUICK-LOOK DATA SYSTEM
LANGLEY TEST SET-UP (PRE I&T)

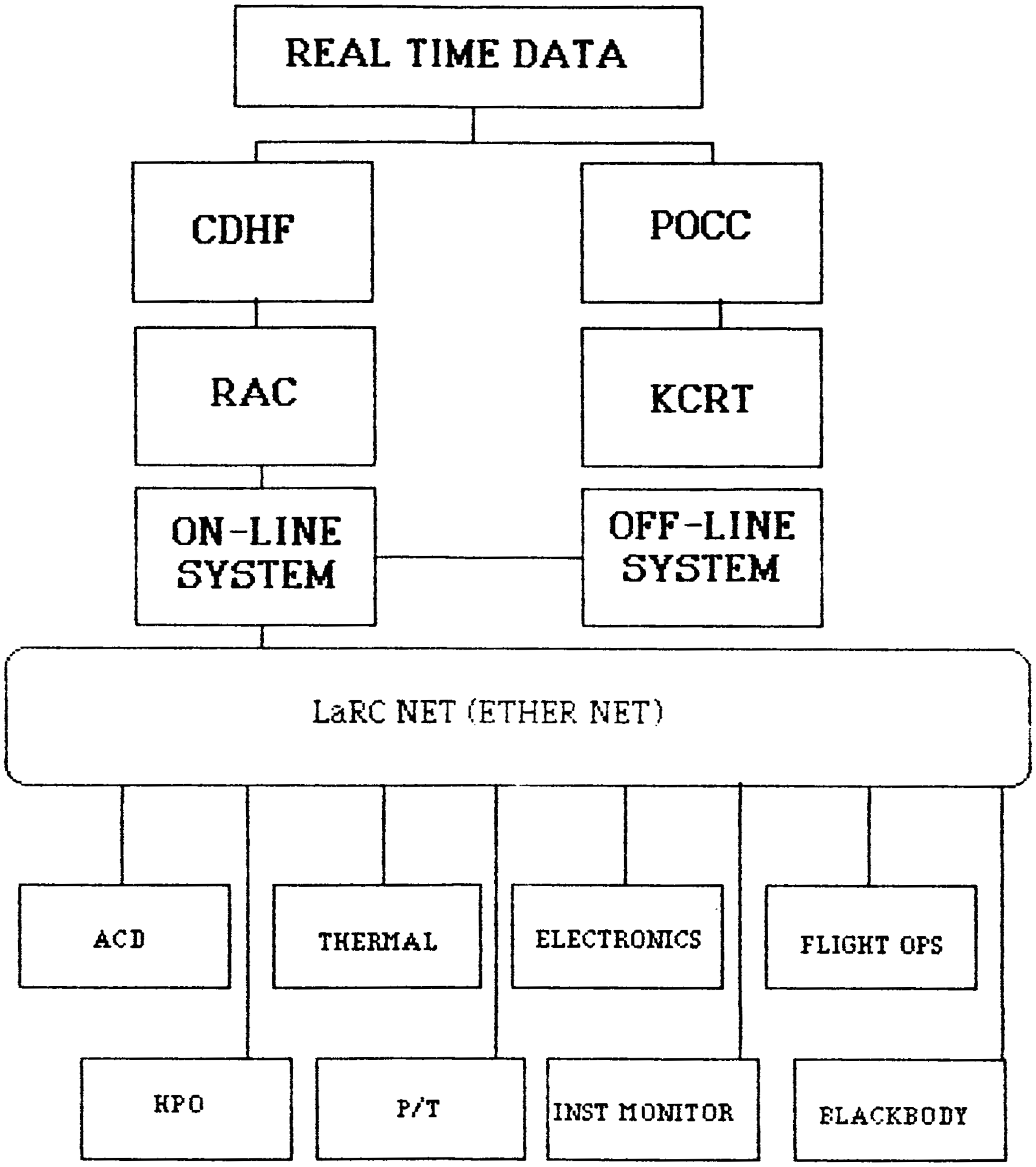


HALOE QUICK-LOOK DATA SYSTEM

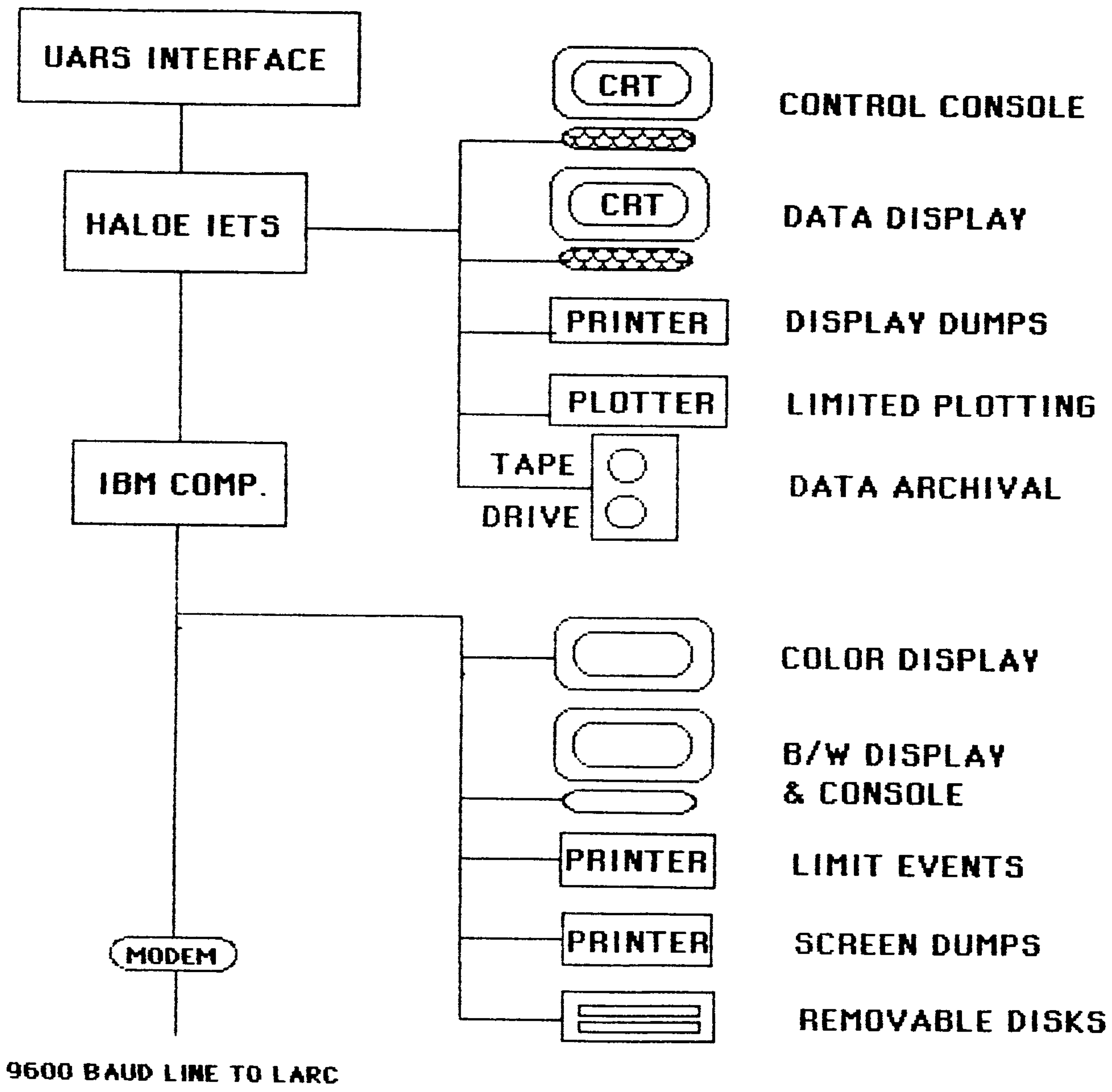
UARS I & T



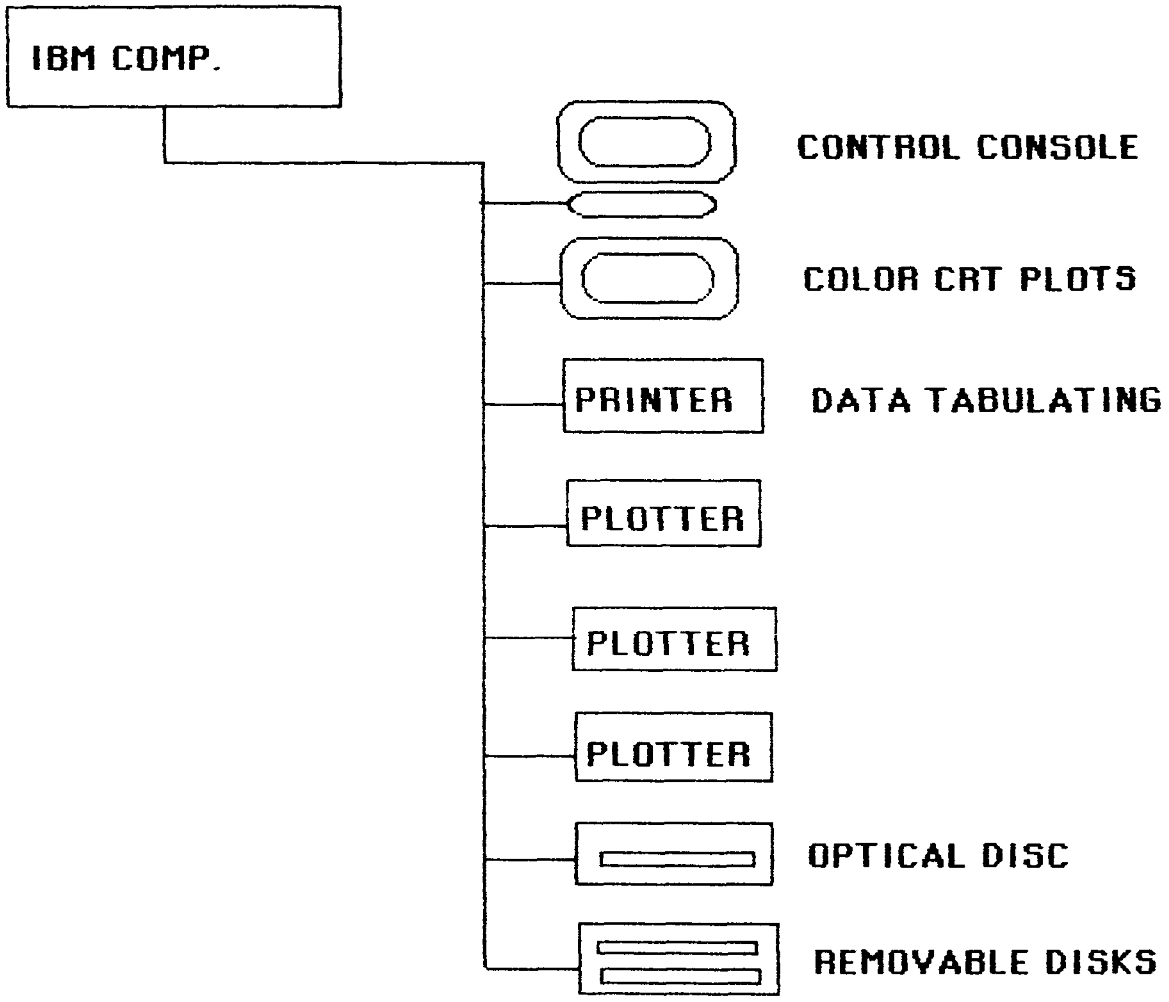
HALOE QUICK-LOOK DATA SYSTEM
POST LAUNCH



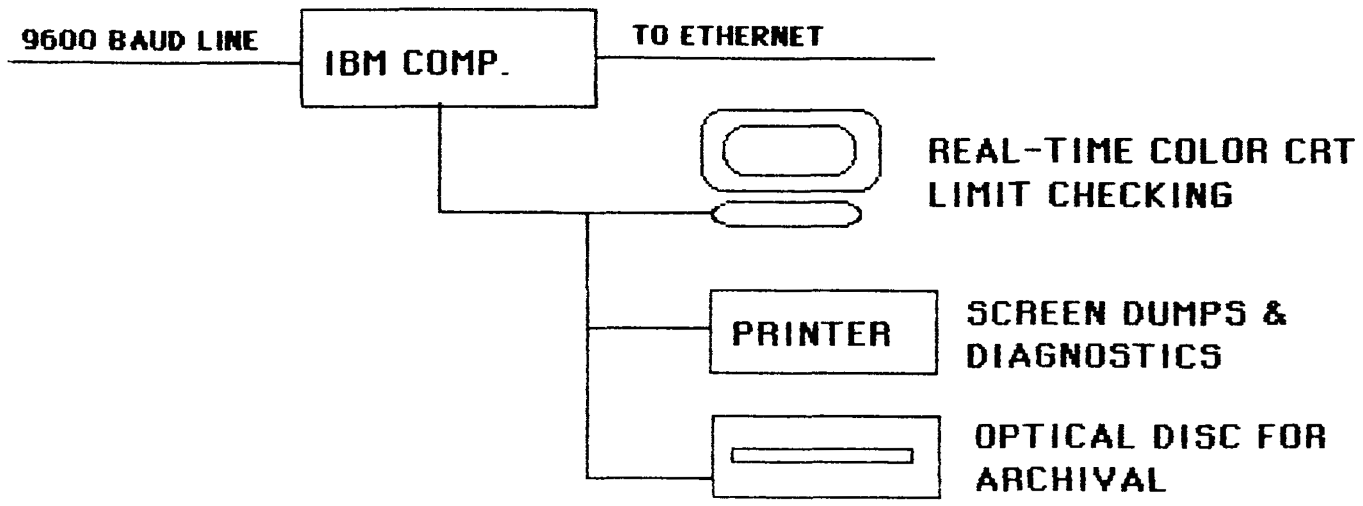
HALOE/UARS ON-LINE SYSTEM



HALOE/UARS OFF-LINE SYSTEM



HALOE - LANGLEY REMOTE ON-LINE DISPLAY



SECTION 6 - DATA REDUCTION & ANALYSIS

Data reduction and analysis efforts under this contract were largely concerned with the HALOE blackbody life tests. The HPLOT program described elsewhere in this report (and documented in the appendix) was utilized to evaluate, primarily through plot generation, a considerable quantity of HALOE blackbody test data.

HALOE instrument test data tapes were processed using the CDC NOS facility. Utilizing software developed by STX personnel under other contracts, a large number of tapes were converted into data files which were then used to generate a wide variety of plots. These plots were instrumental in the timely evaluation of HALOE EMI and thermal vacuum test data.

APPENDIX A - HARP

Program Name:	HARP (HALOE Analysis and Reduction Program)
Function:	HARP is designed to facilitate the processing of HALOE test data tapes for performance verification and characterization of the HALOE instrument.
Description:	HARP is a segmented program written in Fortran on an HP-1000 computer. At various stages of development and usage, HARP has had segments which were used to plot parameters on different output devices, to do Fourier analysis and to calculate statistical values such as mean and standard deviation for data taken at different "cal-wheel" positions.
Use:	HARP is invoked on an HP-1000 by typing HARP. The program is menu driven and will offer the user flexibility in determining input and output files and plotter devices. The windowing technique offered by HARP greatly facilitates the selection and processing of parameters of interest from the HALOE data stream during times of interest.

```
$EMA(XYZ,0)
$FILES(3,3)
      PROGRAM HARP(),HALOE ANALYSIS AND REDUCTION PROGRAM <870519.1240>
C      PROGRAM NAME:    HARP
C
C      WRITTEN BY WILLIAM L EDMONDS
C      STX CORPORATION
C                 NASA EXT 3761
C                 STX  865 0214
C
C
C
C          HARP (HALOE ANALYSIS AND REDUCTION PROGRAM ) IS THE BASE
C     SEGMENT OF A SYSTEM OF SOFTWARE DESIGNED TO ANALYZE AND REDUCE
C     HALOE TEST DATA TAPES. THIS BASE SEGMENT ( REFERRED TO AS HARP)
C     IS EXECUTED ONLY ONCE. IT CALLS THE MAIN SEGMENT (HARP0) TO
C     DISPLAY THE OPTION MENU AND PROCESS WHATEVER TASKS THE USER
C     SELECTS. SEE THE LISTING FOR HARP0 FOR A BRIEF DESCRIPTION
C     OF ITS FEATURES.
C
C
C
C
      INTEGER HARP0(3)
      COMMON/XYZ/ IVDT(7,200), NIBD(500),IVDTN(6),MNE(4,200),X(16384),
     *NPT(16),IDCNT,IST(6),IET(6),MON(4,16),
     *IDN(16),ITYP(16),IFREQ(16),XMIN(16),XMAX(16),NPTS
     *,SUMX(16),SUMX2(16)
      COMMON LUT,LULOG,LUIN,LUWIN,NTAP,INBUF(10),LBUF(1510),LUPR
C
C
C
CN     STRUCTURE OF VARIABLE DEFINITION TABLE (VDT)
CN       IVDT(I,ID) I=1 TO 7  ID = ID OF ASSOCIATED PARAMETER
CN            IVDT(1,I) = NIBBLE TABLE POINTER
CN            IVDT(2,I) = LIMIT TABLE INDEX
CN            IVDT(3,I) = DESCRIPTION IXDEX
CN            IVDT(4,I) = NUMBER OF OCCURANCES/ MAJOR FRAME
CN            IVDT(5,I) = START BIT WITHIN NIBBLE
CN            IVDT(6,I) = LENGTH (BITS)
CN            IVDT(7,I) = CONVERSION EQUATION #
CA
CA        NIBD(IVDT(1,ID)) - POINTS TO FIRST OCCURANCE  OF PARAMETER ID
CA          NIBD(IVDT(1,ID)+1) TO NIBD(IVDT(1,ID)+IVDT(4,ID)-1) POINT
CA                               TO SUCCESIVE OCCURANCES OF SAME
CA
CA        MNE(1,ID) - MNE(4,ID) CONTAINS NAME OF PARAMETER ID
CA
C*******************************************************************
      COMMON /ENG/ IENG
      COMMON /IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
     *,ITYPE
      COMMON/LLAGC/LAGC(16)
      LOGICAL IEOF,LAGC
      DOUBLE PRECISION*8 XMEAN,VAR,SD,SUMX,SUMX2,DIFF
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          LOGICAL LAGC
          DATA HARPO/'HARPO '/
    C
    C
    C          GET INPUT STRING IF ANY
    C
    C
          CALL GETST(INBUF,10,ILOG)
          IVDTN(1)=2HVA
          IVDTN(2)=2HRD
          IVDTN(3)=2HEF
          IVDTN(4)=2H
          IVDTN(5)=2H
          IVDTN(6)=2H
          CALL LGBUF(LBUF,1510)
          LUPR=6         ! DEFAULT OUTPUT IS TO PRINTER
          DO 100 I=1,16
      100 IFREQ(I)=0
          CALL SEGLD(HARPO,IRTN)
    C     LOAD MENU SEGMENT HARPO
          END

FTN4X COMPILER: HP92834 REV.2130 <810716>

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:   624        COMMON:  1526
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
PAGE   3   FTN.    OPTS:   LYI          12:41 PM  TUE., 19  MAY , 1987

           BLOCK DATA DLA
           COMMON LUT,LULOG,LUIN,LUWIN,NTAP,INBUF(10),LBUF(1510),LUPR
           COMMON/ENG/IENG
           COMMON /IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
          *,ITYPE
           COMMON/DISP/ IDD(100),IDDS(10),IDDNM(6,6)
           COMMON/LLAGC/LAGC(16)
           DATA IENG/2/
           END

FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON:  1526

BLOCK COMMON LLAGC    SIZE:     16

BLOCK COMMON DISP     SIZE:    146

BLOCK COMMON IDAT     SIZE:    302

BLOCK COMMON ENG      SIZE:      1
```

```
$EMA(XYZ,0)
$FILES(3,4)
      PROGRAM HARP0(5,99), MENU SEGMENT FOR HARP <851010.0905>
      COMMON /ENG/ IENG
      COMMON LUT,LULOG,LUIN,LUWIN,NTAP,INBUF(10),LBUF(1510),LUPR
      COMMON /XYZ/DAT(16384   ),NPT(16),IDCNT,IST(6),IET(6),MON( 4,16),
     *IDN(16),ITYP(16),IFREQ(16),XMIN(16),XMAX(16),NPTS
     *,SUMX(16),SUMX2(16)
C
      COMMON/VDT/IVDT(7,200),NIBD(500),IVDTN(6),MNE(4,200)
      COMMON/IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
     *,ITYPE
C
C     LU 41 = COMMAND FILE (IF ANY)   (LUT)
C     LU 8 = TAPE UNIT  (LUIN)
C     LU 40 = DISK FILE (IF ANY)  (LUIN)
C     LU 42 = WINDOW FILE    (LUWIN)
C
C
      LOGICAL IEOF
      DOUBLE PRECISION*8 XMEAN,VAR,SD,SUMX,SUMX2,DIFF
C
      DIMENSION NAM(6),ISTAR(4),IEND(4),ISTM(6),IETM(6),ITBUF( 6),
     *ITIME(13),ITIMS(13),ITBU2(6),ISCALS(6),IC(16)
      DIMENSION JTIMS(7),NEMO(4),IDESC(10)
      INTEGER HARP1(3),HARP2(3)
      EQUIVALENCE( JTIMS(1),ITIMS(1))
      INTEGER CKTM
      DIMENSION MPTS(16),INOTE(38)
      DATA NAM/'WINDOW::22'/
      DATA  HARP1/'HARP1 '/
      DATA HARP2/'HARP2 '/
      DATA ISCALS/'SCALES     '/

      DATA ISTAR/'STARTING'/
      DATA IEND/'ENDING  '/
C
C     HARP0 IS THE MENU SEGMENT OF HARP: GENERAL ANALYSIS PROGRAM
C     FOR HALOE. WHEN PROGRAM HARP IS RUN, THE FIRST SEGMENT LOADED
C     WILL BE HARP0. VARIOUS MENU ITEMS CAN THEN BE EXECUTED TO
C     SELECT THE INPUT DATA FILE, SELECT A TIME WINDOW, SELECT
C     PARAMETERS TO PROCESS AND DETERMINE WHAT CALCULATIONS AND PLOTS
C     ARE DESIRED.
C
C
      LUT=LOGLU(IDUM)      ! GET LU OF TERMINAL
      LUPR=6
      NPTS=16384        ! SET DEFAULT NUMBER PTS PER PARAMETER
      IDCNT=0
      OPEN(UNIT=20,FILE=IVDTN,IOSTAT=IOS,ERR=5)
      CALL RVDT(20)
      CLOSE(20)
C
C     READ IN SCALE FACTORS
C
```

```
C
      OPEN(UNIT=20,FILE=ISCALS,ERR=6,IOSTAT=IOS)
      CALL RWSCL(20,-1)
      CLOSE(20)
C
C
C
      GOTO10
5     WRITE(LULOG,2100)IOS,IVDTN
      STOP
6     WRITE(LULOG,2100)IOS,ISCALS
      STOP
10    CONTINUE
      IF(ILOG.NE.0)THEN       ! SEE IF WE'RE USING A DISK COMMAND FILE
      LUT=41                 ! YES...USE LU 41 (ARBITRARY #)
      OPEN(LUT,IOSTAT=IOS,ERR=1999,FILE=INBUF)
      ENDIF
      LULOG=LOGLU(IDUM)       ! SET OUTPUT TO TERMINAL
      WRITE(LULOG,2009)       ! DO YOU WANT TO SELECT INPUT FILE?
2009  FORMAT(" DO YOU WANT TO SELECT AN INPUT FILE? Y/N")
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HY)GOTO100    ! IF YES, GO TO FILE SELECTION ROUTINE
C
C
C     DISPLAY MENU AND INPUT SELECTION
C
C
1     WRITE(LULOG,2000)
2000  FORMAT(// " 1  =  SELECT NEW INPUT FILE NAME OR UNIT "/
     *             " 2  = SELECT TIME WINDOW "/
     *             " 3  = SELECT PARAMETERS    "/
     *             " 4  = GENERATE PLOTS "/
     *             " 5  = TIME SERIES ANALYSIS "/
     *             " 6  = SEARCH ANNOTATE RECORDS  "/
     *             " 7  = PRINT SELECTED PARAMETERS "/
     *             " 8  = STATISTICS             "/
     *             " 9  = PROCESS BY PARAMETER VALUE"/
     *             " 10 = EXECUTE A COMMAND FILE",/,
     *                " 11 = SPECIFY OUTPUT LOG DEVICE LU",/,
     *                " 12 = MAKE TREND SNAP-SHOT ",/,
     *                " 13 = QUIT ")
14    READ(LUT,*,END=15)IANS
      GOTO(100,200,300,400,500,600,700,800,900,1000,1100,1200,
     *1300)IANS
15    CLOSE(LUT)
      LUT=LOGLU(IDUM)
      GOTO14
C
C
C     SELECT INPUT FILE NAME OR UNIT
C
C
100   WRITE(LULOG,2010)          ! CHOOSE DISK OR TAPE INPUT
2010  FORMAT(" ENTER T FOR TAPE OR D FOR DISK INPUT FILE ")
      READ(LUT,2001)IANS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 2001   FORMAT(A1)
        DO 101 I=1,6
 101    IBTIM(I)=0
        IF(IANS.NE.1HT.AND.IANS.NE.1HD)THEN
        WRITE(LULOG,2002)
        GOTO1
        ENDIF
 2002   FORMAT(" INCORRECT RESPONSE ")
        CLOSE(LUIN)    ! CLOSE WHATEVER WAS OPEN IF ANYTHING
        CLOSE(LUWIN)     ! CLOSE WHATEVER WINDOW FILE WAS OPEN
        IF(IANS.EQ.1HT) THEN
        LUIN=8       ! INPUT WILL COME FROM TAPE UNIT
        WRITE(LULOG,2005)
 2005   FORMAT(" DO YOU WANT TO USE THE ALTERNATE TAPE DRIVE? (Y/N)")
        READ(LUT,2001)IANS
        IF(IANS.EQ.1HY)LUIN=9
        NTAP=5         ! SET FLAG TO FORCE READ BY REDAT ON 1ST CALL
        OPEN(LUIN,IOSTAT=IOS,ERR=1998)
        LUWIN = LUIN     ! DEFAULT WINDOW FILE IS THE INPUT FILE
        ELSE
C
C       GET NAME OF INPUT DISK FILE
C
        WRITE(LULOG,2003)
 2003   FORMAT(" ENTER NAME OF INPUT FILE (6A2) ")
        READ(LUT,2004)NAM
 2004   FORMAT(6A2)
        LUIN=40           ! ARBITRARY UNIT NUMBER
        OPEN(LUIN,IOSTAT=IOS,ERR=1997,FILE=NAM)
        LUWIN=LUIN          ! DEFAULT WINDOW FILE IS INPUT FILE
        ENDIF
        GOTO1               ! END OF OPTION 1
C
C
C
C------------------------------------------------------------------
C
C
C       SELECT TIME WINDOW AND CREATE WINDOW FILE
C
C
C
 200    CONTINUE
        WRITE(LULOG,2019)
 2019   FORMAT(" REWIND THE INPUT FILE? Y/N")
        READ(LUT,2001)IANS
        IF(IANS.EQ.1HY)REWIND(LUIN)
 201    WRITE(LULOG,2020)
 2020   FORMAT(" DO YOU WANT TO SPECIFY START & STOP TIMES (Y/N)")
        READ(LUT,2001)IANS
        IF(IANS.EQ.1HN)GOTO250        ! PROCESS FROM CURRENT TIME
 204    CONTINUE
        CALL GETIM(LUT,LULOG,ISTAR,ISTM,IER)
        IF(IER.EQ.0)GOTO205
 203    WRITE(LULOG,2021)
```

```
2021  FORMAT(" DO YOU WANT TO RE-ENTER (Y/N)?")
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HY)GOTO204
      GOTO1                          ! ABORT THIS OPTION
205   CONTINUE
      CALL GETIM(LUT,LULOG,IEND,IETM,IER)
C
C     NOW PUT START AND STOP TIMES INTO EMA COMMON ARRAYS IST & IET
C
      DO 206 I=1,6
      IST(I)=ISTM(I)
206   IET(I)=IETM(I)
      CALL REDAT(IEOF,1)          ! READ FIRST RECORD
      IF(IEOF)THEN
      WRITE(LULOG,2032)
      GOTO1
      ENDIF
      IF(IER.EQ.0)GOTO280
      WRITE(LULOG,2021)
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HY)GOTO205
      GOTO1                        ! ABORT
250   WRITE(LULOG,2025)
2025  FORMAT(" DO YOU WANT TO EXTRACT DATA STARTING AT ",/,
     *" CURRENT POSITION OF INPUT FILE? (Y/N)")
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HN)GOTO1      ! ABORT
      DO 252 I=1,6
252   ISTM(I)=IBTIM(I)
      WRITE(LULOG,2026)
2026  FORMAT(" ENTER NUMBER OF HOURS,MINUTES & SECS TO PROCESS",/,
     *" IN THE FORM HH,MM,SS (THREE INTEGERS SEPERATED BY COMMAS)")
C
      READ(LUT,*)IHR,MN,ISEC
      CALL REDAT(IEOF,1)       ! READ FIRST RECORD
      IF(IEOF)THEN
      WRITE(LULOG,2032)
2032  FORMAT(" INPUT FILE AT EOF, ABORTING ")
      GOTO1
      ENDIF
      DO 260 I=1,6
260   ISTM(I)=IBTIM(I)
      SEC=ISEC
      CALL ADTIM(ISTM,IHR,MN, SEC,IETM)     ! CALCULATE ENDING TIME
C
      WRITE(LULOG,2029)
2029    FORMAT(" START, STOP TIMES : ",//)
C
      CALL CNVTM(ISTM,ITIME)
      WRITE(LULOG,2036)ITIME
      CALL CNVTM(IETM,ITIME)
      WRITE(LULOG,2036)ITIME
2036  FORMAT(2X,13A2)
280   CONTINUE
285   WRITE(LULOG,2030)
```

```
2030  FORMAT(" DO YOU WANT TO SPECIFY NAME OF WINDOW FILE(Y/N)")
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HN)GOTO288
      WRITE(LULOG,2031)
2031  FORMAT(" ENTER WINDOW FILE NAME (6A2)")
      READ(LUT,2004)NAM
      LUWIN=0
288   IF(LUWIN.EQ.42)THEN
      WRITE(LULOG,2037)
2037  FORMAT(" APPEND TO WINDOW FILE IN USE? Y/N ")
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HY)GOTO289
      CLOSE(LUWIN)
      ELSE
      LUWIN=42    ! IN ANY EVENT, A NEW WINDOW FILE IS LU 42
      OPEN(LUWIN,IOSTAT=IOS,ERR=299,FILE=NAM,STATUS='UNKNOWN')
      ENDIF
C
289   CALL SEEK(ISTM,IERR)
      IF(IERR.GT.0)GOTO299
286   CALL REDAT(IEOF,0)       ! ZERO INDICATES ALL RECORD TYPES
      IF(IEOF)GOTO295
      IF(CKTM(IBTIM,IETM))287,287,295
287   WRITE(LUWIN,ERR=299)ITYPE,IPWR,IBTIM,IBUF,IDUM,IANHK,ISTAT
      WRITE(LULOG,2049)
2049   FORMAT(" STORING DATA IN WINDOW FILE")
      GOTO286
295   REWIND(LUWIN)
      GOTO1
299   WRITE(LULOG,2035)IERR,LUWIN
2035  FORMAT(" ERROR# ",I5," ON LU# ",I5)
      GOTO1
C
C-------------------------------------------------------------
300   CONTINUE
C
C     SELECT PARAMETERS TO PROCESS
C
      MAXP=16
      CALL PRAMS(MAXP,IER)
      CALL XTRAC(8)          ! EXTRACT SELECTED VALUES
      IF(IER.NE.0)GOTO1
C
C     INSERT DISPLAY OF PARAMETERS CHOSEN HERE..
C
      GOTO1
C
C
C---------------------------------------------------------------
C
C
400   CONTINUE
C
C     PLOT SELECTED PARAMETERS
C
```

```
C
        CALL SEGLD( HARP1,IERR)
        GOTO1
  500   CONTINUE
        CALL SEGLD(HARP2,IERR)
        IF(IERR.NE.0)WRITE(LULOG,501)IERR
  501   FORMAT(" ERROR SCHEDULING HARP2 SEGMENT, ERR#= ",I5)
        GOTO1
  600   CONTINUE
        WRITE(LULOG,6001)
  6001  FORMAT(" FORWARD OR REVERSE SEARCH? (F/R)")
        READ(LUT,2001)IANS
        IF(IANS.EQ.1HR)GOTO6500
        IF(IANS.NE.1HF)THEN
        WRITE(LULOG,6002)
  6002  FORMAT(" INVALID RESPONSE!")
        GOTO1
        ENDIF
  601   READ(LUIN,END=6099,ERR=6098)ITYPE,(INBUF(I),I=1,4),IBTIM
        CALL CNVTM(IBTIM,ITIME)
        WRITE(LULOG,6003)ITIME
        IF(IFBRK(KK))1,602,1
  602   IF(ITYPE.NE.3)GOTO601
        BACKSPACE(LUIN)
        READ(LUIN)ITYPE,(INBUF(I),I=1,4),IBTIM,INOTE
        CALL CNVTM(IBTIM,ITIME)
        WRITE(LULOG,6003)ITIME,INOTE
        GOTO601
  6099  WRITE(LULOG,'(" END OF INPUT FILE")')
        GOTO1
  6098  WRITE(LULOG,'(" ERROR ON INPUT FILE")')
        GOTO1
  6500  BACKSPACE(LUIN)
        BACKSPACE(LUIN)
  6501  READ(LUIN,END=6099,ERR=6098)ITYPE,(INBUF(I),I=1,4),IBTIM,INOTE
        IF(IFBRK(KK))1,6502,1
  6502  IF(ITYPE.NE.3)GOTO6500
        CALL CNVTM(IBTIM,ITIME)
        WRITE(LULOG,6003)ITIME,INOTE
        GOTO6500
  6003  FORMAT(1X,13A2,2X,38A2)
  700   CONTINUE
        ISEC=0
        WRITE(LULOG,7010)
  7010   FORMAT(//," 1 = SELECT PRINT FREQUENCY ",/,
       *" 2 = PRINT SELECTED PARAMETERS ",/,
       *" 3 = PRINT IN SELECTED DISPLAY FORMAT",/,
       *" 4 = RETURN TO MAIN MENU")
  701   READ(LUT,*    )IANS
        IF(ICHK(IANS,1,4))701,702,701
  702   GOTO(7100,7200,7300,1)IANS
  7100  WRITE(LULOG,7011)
  7011  FORMAT(" ENTER PRINT FREQUENCY ",/,
       *" 1 = EVERY SECOND",/,
       *" 2 = EVERY 2 SECONDS...ETC.")
```

```
      READ(LUT,*)ITDEL
      GOTO700
 7200  CONTINUE
      MAXFRQ=1
      MAXP=16
      CALL PRAMS(MAXP,IER)
      CALL XTRAC(MAXFRQ)
      DO 706 KK=1,IDCNT
 706  IC(KK)=1
      DO 703 KK=1,6
      ITBUF(KK)=IST (KK)
 703  ITBU2(KK)=IET(KK)
      CALL CNVTM(ITBUF,ITIMS)
      CALL CNVTM(ITBU2,ITIME)
 704  WRITE(LUPR,7000) ITIMS,
     *((MON(KK,LL      ),KK=1,4),LL=1,IDCNT)
      ILINE=0
 705  ILINE=ILINE+1
      IF(ILINE.GT.50)GOTO0704
      IHR=0
      MN=0
      SEC=FLOAT(ISEC)*1.024
      CALL ADTIM(ITBUF,IHR,MN, SEC,ITBU2)
      CALL CNVTM(ITBU2,ITIMS)
      WRITE(LUPR,7001)JTIMS,(DAT(IND(IC(NP),NP)),NP=1,IDCNT)
 7000 FORMAT(1H1,//,27X,13A2,//,14X,16(2X,4A2))
 7001 FORMAT(1X,7A2,1X,16E10.4)
      DO 710 KK=1,IDCNT
      IC(KK)=IC(KK)+ITDEL*MAXFRQ
      IF(IC(KK).GT.NPT(KK))GOTO1
 710  CONTINUE
      ISEC=ISEC+ITDEL
      IF(IFBRK(KL))1,705,1
 7300  CONTINUE
      CALL RDISP
 7301   CONTINUE
      CALL PRDS(IEOF)
      IF(IEOF)GOTO1
      IF(ITDEL.GT.1)CALL SKIPY(LUIN,ITDEL,IEOF,LULOG,NTAP)
      IF(IEOF)GOTO1
      IF(IFBRK(KL))1,7301,1
C
C
C
 800  CONTINUE
      JFIR=0             ! SET FLAG TO ACQUIRE BEGIN TIME
C     CALCULATE VARIOUS STATISTICAL VALUES
      WRITE(LULOG,8000)
 8000 FORMAT(//"  1 = STATS ON ALL SCIENCE DATA ",/,
     *           "  2 = STATS ON SELECTED PARAMETERS ",/,
     *           "  3 = RETURN TO MAIN MENU")
      READ(LUT,*)IANS
      GOTO(8100,8200,1)IANS
 8100 CONTINUE                  ! STATS ON ALL SCIENCE DATA
      IDCNT=12
      IDN(1)= IDGET(8HNOV     )
      IDN(2)= IDGET(8HNODV    )
      IDN(3)= IDGET(8HHCLV    )
      IDN(4)= IDGET(8HHCLDV   )
      IDN(5)= IDGET(8HHFV     )
      IDN(6)= IDGET(8HHFDV    )
      IDN(7)= IDGET(8HCH4V    )
      IDN(8)= IDGET(8HCH4DV   )
      IDN(9)= IDGET(8HO3V     )
      IDN(10)=IDGET(8HCO2V    )
      IDN(11)=IDGET(8HNO2V    )
      IDN(12)=IDGET(8HH2OV    )
      DO 807 I=1,12
      CALL IDMOV(I)
      IFREQ(I)=8
      MPTS(I)=0
807   CONTINUE
      GOTO808
8200  CALL PRAMS(16,IER)
      IF(IER.NE.0)GOTO1
808    CONTINUE
      DO 809 I=1,IDCNT
      SUMX(I)=0.
      SUMX (I)=0.0
      XMIN(I)=1.0E20
      XMAX(I)=-1.E20
      MPTS(I)=0
809   CONTINUE
810   CALL REDAT(IEOF,1)
      IF(IEOF)GOTO820
      IF(JFIR.EQ.0)THEN
      JFIR=1
      DO 817 K=1,6
817   ISTM(K)=IBTIM(K)
      ENDIF                     ! ACQUIRE BEGINNING TIME
      DO 815 K=1,IDCNT
      DO 816 L=1,IFREQ(K)
      ID=IDN(K)
      ICNTR=0
      IDAT=IGET(ID,L,ICNTR,V)
      SUMX(K)=SUMX(K)+V
      SUMX2(K)=SUMX2(K)+V*V
      IF(V.LT.XMIN(K))XMIN(K)=V
      IF(V.GT.XMAX(K))XMAX(K)=V
816   CONTINUE
      MPTS(K)=MPTS(K)+IFREQ(K)
815   CONTINUE
C     NPTS=NPTS+8   !  NUMBER OF POINTS SUMMED SO FAR
      IF(IFBRK(KK))820,810,820
820   CONTINUE
      IF(NPTS.EQ.0)THEN
      WRITE(LULOG,8005)
8005   FORMAT(" NO DATA OR EOF ENCOUNTERED IN INPUT FILE")
      GOTO1
      ENDIF
      DO 821 K=1,6
821   IETM(K)=IBTIM(K)
      CALL CNVTM(ISTM,ITIMS)
      CALL CNVTM(IETM,ITIME)
       WRITE(LULOG,8001)ITIMS,ITIME
8001   FORMAT(//" START : ",13A2,5X," STOP : ",13A2,/)
      IF(LUPR.NE.0)THEN
      WRITE(LUPR,'(1H1)')
      WRITE(LUPR,8001)ITIMS,ITIME
      WRITE(LUPR,8002)
      ENDIF
      WRITE(LULOG,8002)
8002  FORMAT(     //"  NAME        MINIMUM    MAXIMUM     MAX-MIN",
     *"       MEAN       VARIANCE     STD DEV    #PTS    "//)
      DO 830 I=1,IDCNT
      PTS=FLOAT(MPTS(I))
      XMEAN=SUMX(I)/PTS
      VAR=(PTS*SUMX2(I)-SUMX(I)*SUMX(I))/((PTS-1.D0)*PTS)
      IF(VAR.GT.0.0)SD=DSQRT(VAR)
      DIFF=XMAX(I)-XMIN(I)
      IF(DIFF.EQ.0.0)THEN
      VAR=0.0
      SD=0.0
      ENDIF
      WRITE(LULOG,8003)(MON(JJ,I),JJ=1,4),XMIN(I),XMAX(I),DIFF,XMEAN
     * ,VAR,SD,MPTS(I)
       IF(LUPR.NE.0)WRITE(LUPR,8003)(MON(JJ,I),JJ=1,4),XMIN(I),XMAX(I),
     *DIFF,XMEAN,VAR,SD,MPTS(I)
8003   FORMAT(1X,4A2,5(E10.6 ,1X),E10.6,I6)
830   CONTINUE
      GOTO1
898   CONTINUE
      WRITE(LULOG,8004)IOS
8004   FORMAT(" ERROR # ",I5," ON WINDOW FILE ")
      GOTO1
900   CONTINUE
      WRITE(LULOG,9001)
9001  FORMAT(" DO YOU WANT TO PROCESS BY PARAMETER VALUE?",
     */," (FOR CAL-WHEEL, IFOV, SPECTRAL RESPONSE ETC.)Y/N?")
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HN)GOTO1
901   CALL GETIM(LUT,LULOG,ISTAR,ISTM,IER)
      IF(IER.NE.0)THEN
      WRITE(LULOG,2021)
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HN)GOTO1
      GOTO901
      ENDIF
      CALL REDAT(IEOF,1)
      IF(IEOF)THEN
      WRITE(LULOG,2032)
      GOTO1
      ENDIF
      CALL SEEK(ISTM,IERR)
      IF(IERR.NE.0)GOTO9099
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      MAXP=23
      CALL PRAMS(MAXP,IER)
      IF(IER.NE.0)GOTO1
      WRITE(LULOG,9003)
9003   FORMAT(" ENTER NAME OF PARAMETER FOR STUDY",/,
     *"CS3  FOR CAL WHEEL; STATUS2 FOR IFOV,SPECTRAL RESPONSE")
      READ(LUT,2004)NEMO
      IDNUM=IDGET(NEMO)
      IF(IDNUM)910,910,920
910   WRITE(LULOG,9004)
9004   FORMAT(" NOT WHAT I WAS LOOKING FIR...")
      GOTO1
920   DO 930 I=1,IDCNT
      NPAR=I
      IF(IDNUM.EQ.IDN(I))GOTO950
930   CONTINUE
      IDCNT=IDCNT+1
      IDN(IDCNT)=IDNUM
      DO 932 I=1,4
      MON(K,IDCNT)=NEMO(I)
932   CONTINUE
      NPAR=IDCNT
950    CONTINUE
      WRITE(LULOG,9010)
9010  FORMAT(" ENTER SHORT DESCRIPTIVE NAME FOR PARAMETER",/,
     *" SUCH AS: SLIT POSITION OR WAVENUMBER OR CAL POSITION ETC.")
      READ(LUT,9011)IDESC
9011   FORMAT(10A2)
      WRITE(LULOG,9012)
9012  FORMAT(" ENTER # OF SECONDS (MAJOR FRAMES) OF DATA TO ",/,
     *" PROCESS AT EACH LEVEL OF THE PARAMETER")
      READ(LUT,*)NFRAM
      WRITE(LULOG,9013)
9013  FORMAT(" ENTER MINIMUM # SECONDS ACCEPTIBLE AT EACH LEVEL")
      READ(LUT,*)MINF
      WRITE(LULOG,9014)
9014   FORMAT(" ENTER MAXIMUM # LEVELS TO PROCESS")
      READ(LUT,*)MVAL
      CALL PMET(NFRAM,MINF,NPAR,MVAL,LUPR,IDESC,ISTM)
      GOTO1
9099  WRITE(LULOG,9002)IERR
      GOTO1
9002  FORMAT(" ERROR #",I5)
1000  CONTINUE
      WRITE(LULOG,1001)
1001  FORMAT(" DO YOU WANT TO EXECUTE A COMMAND FILE? (Y/N)")
      READ(LUT,2001)IANS
      IF(IANS.EQ.1HN)GOTO1
      WRITE(LULOG,1002)
1002  FORMAT(" ENTER NAME OF COMMAND FILE")
      READ(LUT,2004)NAM
      CLOSE(LUT)
      LUT=41
      OPEN(LUT,IOSTAT=IOS,ERR=1999,FILE=NAM)
      GOTO1
```

```
1100  CONTINUE
C     WRITE(LULOG,1101)
C1101 FORMAT(" DO YOU WANT TO CHANGE THE LIST LU? (Y/N)")
C     READ(LUT,2001)IANS
C     IF(IANS.EQ.1HN)GOTO1
C     WRITE(LULOG,1102)
C1102 FORMAT(" ENTER LU (6=PRINTER, 1 OR 12 = SCREEN, 0 = NONE")
C     READ(LUT,*)LULOG
C     GOTO1
1200   CONTINUE               ! TREND SNAP-SHOT
      WRITE(LULOG,1201)
1201  FORMAT(" DO YOU WANT TO SAVE A SNAP-SHOT? Y/N")
      READ(LUT,2001)IANS
      IF(IANS.NE.1HY)GOTO1
      WRITE(LULOG,1202)
1202  FORMAT("ENTER TREND FILE NAME")
      READ(LUT,2004)NAM
      OPEN(UNIT=20,IOSTAT=IOS,ERR=1299,FILE=NAM)
1203  READ(20,ERR=1299,END=1204)
      GOTO1203
1204  WRITE(20,ERR=1299)ITYPE,IPWR,IBTIM,IBUF,IDUM,IANHK,ISTAT
      CLOSE(20)
      GOTO1
1299  WRITE(LULOG,1298)IOS,NAM
1298  FORMAT(" ERROR # ",I5," ON FILE ",6A2)
      CLOSE(20)
      GOTO1
1300   STOP
1997  LUT=LOGLU(IDUM)            ! RESET LUT TO TERMINAL
      WRITE(LULOG,2100)IOS,NAM
      GOTO1
1998    LUT=LOGLU(IDUM)
      WRITE(LULOG,2101)IOS
      GOTO1
1999   LUT=LOGLU(IDUM)
      WRITE(LULOG,2102)IOS
      GOTO1
2100   FORMAT(" ERROR # ",I5,2X," FILE NAME :",6A2)
2101  FORMAT(" ERROR # ",I5,2X," WITH MAG TAPE ")
2102  FORMAT(" ERROR # ",I5,"   WITH COMMAND FILE ")
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)


 **  NO WARNINGS **  NO ERRORS **   PROGRAM:  4881       COMMON:  1526
```

```
$EMA(XYZ,0)
      SUBROUTINE PMET(NFRAM,MINF,NPAR,MVAL,IPRT,IDESC,ISTM)
      COMMON/IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
      COMMON LUT,LULOG,LUIN,LUWIN,NTAP,INBUF(10),LBUF(1510),LUPR
      COMMON /XYZ/DAT(16384  ),NPT(16),IDCNT,IST(6),IET(6),MON( 4,16),
     *IDN(16),ITYP(16),IFREQ(16),SUMX2(16),NPTS
     *,SUMX(16),XMEAN(16)
      DIMENSION IDESC(10),ISTM(6),ITBUF(13),PMEAN(16)
      LOGICAL IEOF
      DOUBLE PRECISION*8 XMEAN,VAR,SD,SUMX,SUMX2,DIFF
C
      DIMENSION MPTS(24),NAMP(6)
C
C
C     THIS ROUTINE PROCESSES DATA AT TIMES WHEN SOME VALUE SUCH
C     AS CAL WHEEL POSITION IS CONSTANT. SLIT POSITION OR SPECTRAL
C     WAVELENGTH ARE TWO OTHER TYPES OF PARAMETERS WHICH CAN BE
C     PROCESSED WITH THIS ROUTINE.
C
C     NFRAM = DESIRED NUMBER OF FRAMES OF DATA AT EACH LEVEL OR
C             VALUE OF PARAMETER (CAL WHEEL POSITION ETC.)
C     MINF =  MINIMUM NUMBER OF FRAMES ACCEPTIBLE AT EACH LEVEL
C     NPAR =  ID NUMBER OF PARAMETER BEING STUDIED
C     MVAL =  MAXIMUM NUMBER OF LEVELS TO STUDY
C    IPRT =  PRINT FLAG (0= NO PRINTOUT, OTHERWISE PRINT)
C
C

      CALL CNVTM(ISTM,ITBUF)
      WRITE(IPRT,1102)ITBUF,IDESC
 1102 FORMAT(1H1,15X,13A2,10X,10A2)
      WRITE(LULOG,1103)
 1103 FORMAT(" DO YOU WANT TO CREATE A PLOT FILE? Y/N")
      READ(LUT,1104)IANS
 1104 FORMAT(A1)
      IF(IANS.EQ.1HY)THEN
      WRITE(LULOG,1105)
 1105 FORMAT(" ENTER NAME OF PLOT FILE")
      READ(LUT,1106)NAMP
 1106 FORMAT(6A2)
      IPFLAG=1
      OPEN(20,FILE=NAMP,ERR=1120)
      BTIM= ISTM(2)+ISTM(3)*60.+ISTM(4)*3600.
      WRITE(20,1121)IDCNT,ISTM(6),ISTM(5),BTIM,((MON(I,J),I=1,4),J=1
     *,IDCNT)
 1121 FORMAT(I3,2I5,F10.3,64A2)
      ELSE
      IPFLAG=0
      ENDIF
      WRITE(IPRT,1100)((MON(KK,I),KK=1,4),I=1,IDCNT)
  1100 FORMAT(     //,4X,7(4A2,10X))
      NVAL=0
      DO 5 KL=1,6
      IST(KL)=IBTIM(KL)
5     CONTINUE
```

```
1       IF(NVAL.EQ.MVAL) GOTO230
        NVAL = NVAL +1
        IFRAM=0       ! LOCAL COUNTER FOR # FRAMES AT CURRENT LEVEL
        DO 10 I=1,IDCNT
        SUMX(I)=0.             ! INITIALIZE SUM TO ZERO
        SUMX2(I)=0.            ! INITIALIZE SUM X SQUARED TO 0.
        XMEAN(I) =0.           ! INIT SUM OF SQUARES
C       XMIN(I)=1.E20          ! INIT MIN VALUES
C       XMAX(I) =-1.E20        ! INIT MAX VALUES
        MPTS(I)=0              ! INIT NUMBER OF PTS FOR EACH ID
10      CONTINUE
        ICNTR=0
         IPAR=IDN(NPAR)
        IDAT=IGET(IPAR,1,ICNTR,V)
        VAL=V
        DAT(IND(NVAL,NPAR))=V       ! GET NPAR PARAMETER
20      DO 100 K=1,IDCNT
C       IF(IDN(K).EQ.IPAR)GOTO100
        DO 90 L=1,IFREQ(K)
        ID=IDN(K)
        ICNTR=0
        IDAT=IGET(ID,L,ICNTR,V)
        SUMX(K)=SUMX(K)+V
        SUMX2(K)=SUMX2(K)+V*V
C       IF(V.LT.XMIN(K))XMIN(K)=V
C       IF(V.GT.XMAX(K))XMAX(K)=V
90      CONTINUE
        MPTS(K)=MPTS(K)+IFREQ(K)
100     CONTINUE
        IFRAM=IFRAM+1
        IF(IFRAM.EQ.NFRAM)GOTO200
30      CALL REDAT(IEOF,1)
        IF(IEOF)THEN
        WRITE(LULOG,1000)
1000    FORMAT(" EOF ENCOUNTERED IN INPUT FILE")
        GOTO230
        ENDIF
        ICNTR=0
        IDAT=IGET(IPAR,1,ICNTR,V)
        IF(V.EQ.VAL)GOTO20
        IF(IFRAM.GT.MINF)GOTO200          ! FINISHED THIS LEVEL
        WRITE(LULOG,1001)IFRAM,VAL      ! NOT ENOUGH POINTS
1001    FORMAT(" FOUND ONLY ",I5," FRAMES AT LEVEL =",E12.4)
        NVAL =NVAL-1
        GOTO1
200         CONTINUE
        DO 210 KL=1,6
        IET(KL)=IBTIM(KL)
210     CONTINUE                        ! GET ENDING TIME
        DO 220 I= 1,IDCNT
C       IF(IDN(I).EQ.IPAR)GOTO220
        PTS=FLOAT(MPTS(I))
        XMEAN(I)=SUMX(I)/PTS
        PMEAN(I)=XMEAN(I)
        VAR=(PTS*SUMX2(I)-SUMX(I)*SUMX(I))/((PTS-1.D0)*PTS)
```

```
      IF(VAR.GT.0.D0)THEN
      SD=DSQRT(VAR)
      ELSE
      SD=0.0
C     DIFF=XMAX(I)-XMIN(I)
C     IF(DIFF.EQ.0.)THEN
C     VAR=0.0
C     SD=0.0
      ENDIF
      SUMX(I)=SD
220    CONTINUE
      IF(IPFLAG.NE.0)
     *WRITE(20)BTIM,(PMEAN(K),K=1,IDCNT)
      IF(LUPR.NE.0)
     *WRITE(LUPR,1010)((XMEAN(K),SUMX(K)),K=1,IDCNT)
1010  FORMAT(1X,14(F9.4))
225    CONTINUE
      CALL REDAT(IEOF,1)
      IF(IEOF)GOTO230
      ICNTR=0
      IDAT=IGET(IPAR,1,ICNTR,V)
      IF(V.EQ.VAL)GOTO225
      VAL=V
      GOTO1
230   CLOSE(20)
      RETURN
1120   WRITE(LULOG,1119)
1119  FORMAT("ERROR OPENING PLOT FILE")
      RETURN
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

**  NO WARNINGS **  NO ERRORS **   PROGRAM:  1220       COMMON:  1526
```

```
      SUBROUTINE ADTIM(ISTM,IHR,MN, SEC,IETM)
      DIMENSION ISTM(6),IETM(6)
      ISEC=SEC                        ! TRUNCATE VALUE OF SECONDS
      RSEC=SEC-FLOAT(ISEC)
      JSEC=RSEC*100
      IETM(1) = ISTM(1)+JSEC        ! SET ENDING .01 SECS TO STARTING VAL
      ICARY=IETM(1)/100
      IETM(1)=MOD(IETM(1),100)
      IETM(2) = ISTM(2)+ISEC+ICARY !ADD SECONDS TO STARTING SECS
      ICARY = IETM(2)/60            ! CALCULATE CARRY FOR MINUTES
      IETM(2) = MOD(IETM(2),60)     ! MOD MINUTES TO INSURE < 60
      IETM(3) = ISTM(3) + MN + ICARY  ! CALCULATE MINUTES
      ICARY = IETM(3)/60            ! CALCULATE CARRY FOR HOURS
      IETM(3) = MOD(IETM(3),60)     ! ADJUST MINUTES < 60
      IETM(4) = ISTM(4) + IHR + ICARY ! CALCULATE HOURS
      ICARY = IETM(4)/24            ! CALCULATE CARRY FOR DAYS
      IETM(4) = MOD(IETM(4),24)     ! INSURE THAT HOURS<24
      IETM(5) = ISTM(5) + ICARY ! CALCULATE ENDING DAY
      IYMOD = 365                   ! SET # DAYS IN YEAR
      IF(MOD(ISTM(6),4).EQ.0) IYMOD= 366    ! CHECK FOR LEAP YEAR
      ICARY = IETM(5)/IYMOD         ! CALCULATE YEAR CARRY
      IETM(6) = ISTM(6) + ICARY     ! ENDING YEAR
      RETURN
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:   178       COMMON: (NONE)
```

```
$EMA(XYZ,0)
      SUBROUTINE IDMOV(ID)
      COMMON /XYZ/DAT(16384   ),NPT(16),IDCNT,IST (6),IET (6),MON( 4,16),
     *IDN(16),ITYP(16),IFREQ(16),XMIN(16),XMAX(16),NPTS
     *,SUMX(16),SUMX2(16)
      COMMON/VDT/IVDT(7,200),NIBD(500),IVDTN(6),MNE(4,200)
      DO 10 I=1,4
10    MON(I,ID)=MNE(I,IDN(ID))
      RETURN
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    50       COMMON: (NONE)
```

```
           SUBROUTINE JULIN(IDAY,IYR,IM,IDA)
           DIMENSION IMS(12),IDY(13)
           INTEGER*4 IM,IMS
           DATA IMS /'JAN FEB MAR APR MAY JUNEJULYAUG SEPTOCT NOV DEC '/
           DATA IDY /0,31,59,90,120,151,181,212,243,273,304,334,365/
           IAD = 0
           IF(IDAY.LT.60)GO TO 5
           IADD = MOD(IYR,4)
           IF(IADD.EQ.0)IAD = 1
     5     DO 10 I=2,13
           IDC = IDY(I) + IAD
           IF(IDAY.LE.IDC)GO TO 20
     10    CONTINUE
     20    IMN=I-1
           IDA = IDAY - IDY(IMN)
           IF(IDY(IMN).GT.31)IDA = IDA - IAD
           IM = IMS(IMN)
           RETURN
           END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    127       COMMON: (NONE)
```

```
          SUBROUTINE SKIPY(LUIN,ITDEL,IEOF,LULOG,NTAP)
    C
    C     SKIP RECORDS IN THE INPUT FILE
          LOGICAL IEOF
          ITER=ITDEL-1
          DO 10 I=1,ITER
          READ(LUIN,END=20,ERR=30,IOSTAT=IERR)
    10    CONTINUE
          NTAP=5
          CALL REDAT(IEOF,1)
          IEOF=.FALSE.
          RETURN
    20    IEOF=.TRUE.
          RETURN
    30    WRITE(LULOG,1000)IERR
    1000  FORMAT(" ERROR # ",I5," ON INPUT FILE ")
          RETURN
          END
```

```
FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    84       COMMON: (NONE)
```

```
PAGE  19   FTN.    OPTS:   LYI          8:50 AM  WED., 20  MAY , 1987

              SUBROUTINE PRDS
        C     DUMMY SUBROUTINE
              RETURN
              END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:      5       COMMON: (NONE)
```

```
      BLOCK DATA HADAT
      COMMON/IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
     *,ITYPE
      COMMON/ENG/IENG
      COMMON/DISP/ IDD(100),IDDS(10),IDDNM(6,6)
      DATA IENG/2/
      END

FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM: (NONE)      COMMON: (NONE)

BLOCK COMMON DISP    SIZE:   146

BLOCK COMMON ENG     SIZE:     1

BLOCK COMMON IDAT    SIZE:   302
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          SUBROUTINE RDISP, READ DISPLAY FORMAT FILE  (WLE)
          DIMENSION IFILE(6)
C         COMMON/MONTR/ ITCLS,ITLEN,ISBUF(920)
          COMMON/VDT/ IVDT(7,200),NIBD(500),IVDTN(6),MNE(4,200)
          COMMON/DISP/ IDD(100),IDDS(10),IDDNM(6,6)
          COMMON LUT,LULOG,LUIN,LUWIN,NTAP,INBUF(10),LBUF(1510),LUPR
C         COMMON/MSK/ MASK(16)
C         DATA IFILE/'      :DS:22'/
          MXIDD = 100
          LUBD = 20
          LUDIR = 21
11        CALL FMTDR(LULOG,LUDIR,IDDNM)
1         WRITE(LULOG,'(" ENTER DISPLAY FORMAT #: _")')
          READ(LUT,*,ERR=1)IDN
          IF(ICHK(IDN,1,7))1,2,1
2         CONTINUE
          IF(IDN.NE.7)GO TO 22
            CLOSE(LUDIR)
            WRITE(LULOG,'(" ENTER NAME OF FILE: _")')
            READ(LUT,'(3A2)')(IFILE(I),I=1,3)
          GO TO 33
22        DO 3 I=1,6
3            IFILE(I)=IDDNM(I,IDN)
33        CONTINUE
          OPEN(UNIT=LUBD,FILE=IFILE,IOSTAT=ISTAT,ERR=990)
          REWIND LUBD
4         DO 5 I=1,MXIDD
5          READ(LUBD,*,END=6)IDD(I)
          CLOSE(LUBD)
          CLOSE(LUDIR)
6         RETURN
990       WRITE(LULOG,'(" ERROR OPENING ",6A2)')(IDDNM(I,IDN),I=1,6)
          END


FTN4X COMPILER: HP92834 REV.2130 (810716)


 **  NO WARNINGS **  NO ERRORS **   PROGRAM:   298        COMMON:  1526
```

```
          SUBROUTINE FMTDR(LULOG,LUDIR,IDIR),DISPLAY TLM FORMAT DIRECTORY
    C     DIMENSION IDIR(6,6),INAM(3)
    C     DATA INAM/6HFMDIR /
    C     OPEN(UNIT=LUDIR,FILE=INAM,IOSTAT=ISTAT,ERR=990)
    C     REWIND LUDIR
    C      READ(LUDIR)((IDIR(J,K),J=1,6),K=1,6)
    C     DO 10 K=1,6
    C      WRITE(LULOG,'(I4,1X,6A2)')K,(IDIR(J,K),J=1,6)
    C10   CONTINUE
    C     WRITE(LULOG,'("    7 ENTER DISPLAY FILE NAME")')
    C     RETURN
    990   CONTINUE
          WRITE(LULOG,'(" ERROR OPENING FMT DIR")')
          RETURN
          END


FTN4X COMPILER: HP92834 REV.2130 (810716)


 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    35       COMMON: (NONE)
```

```
$EMA(XYZ,0)
      SUBROUTINE PRAMS(MAXP,IER)
      DIMENSION NEMO(4),IHELP(2)
      COMMON /XYZ/DAT(16384   ),NPT(16),IDCNT,IST(6),IET(6),MON(4,16),
     *IDN(16),ITYP(16),IFREQ(16),XMIN(16),XMAX(16),NPTS
     *,SUMX(16),SUMX2(16)
      COMMON LUT,LULOG,LUIN,LUWIN,NTAP,INBUF(10),LBUF(1510),LUPR
      COMMON/IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
     *,ITYPE
      COMMON/VDT/IVDT(7,200),NIBD(500),IVDTN(6),MNE(4,200)
      DATA IHELP/'HELP'/
      IER=0
      IF(IDCNT.NE.0)THEN            ! DISPLAY PARAMS ALREADY CHOSEN
      DO 100 KL=1,IDCNT
      WRITE(LULOG,1001)(MON(K,KL),K=1,4)
100   CONTINUE
1001  FORMAT(1X,4A2)
      NPTS=16384/IDCNT
      WRITE(LULOG,1002)
1002  FORMAT(" THESE ARE THE CURRENT PARAMETERS, DO YOU ",/,
     *" WISH TO ENTER A NEW SET? (Y/N)")
      READ(LUT,'(A1)')IANS
      IF(IANS.EQ.1HN)RETURN
      ENDIF
300   IDCNT=0
301   IDCNT=IDCNT+1
      IF(IDCNT.GT.MAXP)GOTO350
305   WRITE(LULOG,3000)
3000  FORMAT(" ENTER PARAMETER NAME,HELP OR STOP")
      READ(LUT,2004)NEMO
2004  FORMAT(6A2)
      IF(NEMO(1).EQ.2HST.AND.NEMO(2).EQ.2HOP)GOTO350
      IF(NEMO(1).NE.2HHE.OR.NEMO(2).NE.2HLP)GOTO302
C
C     DISPLAY MNEMONICS HERE....
C
      WRITE(LULOG,3005)MNE
3005  FORMAT(/,(9(4A2)    ))
      GOTO305
302   IDN(IDCNT)=IDGET(NEMO)
      IF(IDN(IDCNT))303,350,310
303   WRITE(LULOG,3002)
      GOTO305
310   DO 312 K=1,4
312   MON(K,IDCNT)=NEMO(K)
313   WRITE(LULOG,3001)
3001  FORMAT(" ENTER TYPE (1=HEX,2=ENG,3=TEMP)_")
      READ(LUT,*,ERR=320)ITY
      IF(ICHK(ITY,1,3))320,325,320
320   WRITE(LULOG,3002)
3002  FORMAT(" INVALID ")
      GOTO313
325   ITYP(IDCNT)=ITY
      IFREQ(IDCNT)=IVDT(4,IDN(IDCNT))          ! GET THE FREQ
      GOTO301
```

```
350    IDCNT=IDCNT-1
       DO 355 KL=1,IDCNT
       WRITE(LULOG,3030)(MON(K,KL),K=1,4),
      *IDN(KL),ITYP(KL),NPT(KL)
355    CONTINUE
3030   FORMAT(1X,4A2,3I5)
       WRITE(LULOG,3031)
3031   FORMAT(" ARE THESE PARAMETERS CORRECT? Y/N ")
       READ(LUT,'(A1)')IANS
       IF(IANS.EQ.1HN)GOTO300
360    NPTS=16384/IDCNT
C      CALL XTRAC              ! EXTRACT THE DESIRED VARIABLES
C      CALL TO XTRACT WAS PLACED IN MAIN PROGRAM.
       RETURN
       END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:   683       COMMON:  1526
```

```
      SUBROUTINE XTRAC(MAXFRQ)
      RETURN
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:      6        COMMON: (NONE)
```

```
      SUBROUTINE GETIM(LUT,LULOG,ISTRG,ITIM,IER),PROMPT USER FOR TIME
C
C
C     GETIM PROMPTS THE USERS FOR TIME INPUT.
C     FIRST IT ASKS FOR MONTH/DAY/YEAR AND THEN
C     IT ASKS FOR HOURS/MIN/SEC. IF NO ERRORS ARE DETECTED
C     IT WILL RETURN A VALUE OF ZERO FOR IER. LUT IS THE
C     INPUT LOGICAL UNIT, LULOG IS THE LOGICAL UNIT FOR
C     DIAGNOSTIC OUTPUT. ISTRG IS A STRING (EITHER "BEGINNING"
C     OR "ENDING" USED IN PROMPTING INPUT. ON OUTPUT, ITIM WILL
C     CONTAIN:
C     ITIM(6) = YEAR (TWO DIGITS E.G. 85)
C     ITIM(5) = DAY NUMBER (DAY OF YEAR)
C     ITIM(4) = MILITARY HOUR NUMBER ( 0 TO 23)
C     ITIM(3) = MINUTES ( 0 TO 59)
C     ITIM(2) = SECONDS ( 0 TO 59)
C     ITIM(1) = .01 SECONDS (SET TO ZERO IN THIS ROUTINE)
C
      DIMENSION ITIM(6)
      DIMENSION          IDAY(12),ISTRG(4),IMO(12)
      DATA IDAY/31,28,31,30,31,30,31,31,30,31,30,31/
      DATA IMO/0,31,59,90,120,151,181,212,243,273,304,334/
C
C
C
      IER = 1    ! SET ERROR FLAG TO INDICATE ERROR
      WRITE(LULOG,2200)ISTRG        ! PROMPT USER FOR MN/DA/YR
2200  FORMAT(" ENTER ",4A2," TIME: MN/DA/YR ")
1     READ(LUT,*    ,ERR=1)MN,IDA,IYR
2201  FORMAT(I2,1X,I2,1X,I2)
      IF(MN.GT.0.AND.MN.LT.13)GOTO205
      WRITE(LULOG,2202)
2202  FORMAT(" WRONG!")
      RETURN
205   IF(IDA.GT.0.AND.IDA.LE.IDAY(MN))GOTO210
      IF(MN.EQ.2.AND.AMOD(FLOAT(IYR),4.).EQ.0..AND.IDAY.EQ.29)GOTO210
      WRITE(LULOG,2203)
2203  FORMAT(" INCORRECT DAY # ")
      RETURN
210   IF(IYR.GT.83.AND.IYR.LT.99)GOTO215
      WRITE(LULOG,2204)
2204  FORMAT(" I DON'T THINK THE YEAR IS CORRECT!")
      RETURN
215   WRITE(LULOG,2205)
2205  FORMAT(" ENTER HRS:MIN:SECS    E.G. 14:15:00 ( = 2:15 PM)")
2     READ(LUT,*    ,ERR=215)IHR,MIN,ISEC
      IF(IHR.GE.0.AND.IHR.LT.24)GOTO220
      WRITE(LULOG,2206)
2206  FORMAT(" INVALID ENTRY")
      RETURN
220   IF(MIN.GE.0.AND.MIN.LT.60)GOTO225
      WRITE(LULOG,2206)
      RETURN
225   IF(ISEC.GE.0.AND.ISEC.LT.60)GOTO230
      WRITE(LULOG,2206)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          RETURN
    230   IER=0     ! SET ERROR FLAG TO NO ERROR STATUS
          ITIM(1)=0
          ITIM(2)=ISEC
          ITIM(3) = MIN
          ITIM(4) = IHR
          ITIM(5) = IDA+IMO(MN)
          IF(MN.GT.2.AND.AMOD(FLOAT(IYR),4.).EQ.0)ITIM(5)=ITIM(5)+1
          ITIM(6) = IYR+1900
          RETURN
          END
```

```
FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:   443       COMMON: (NONE)
```

```
      SUBROUTINE SEEK(ISTM,IERR),SEEK TIME ON INPUT FILE
C
C
C
      SUBROUTINE SEEK LOOKS FOR A REQUESTED TIME IN THE INPUT FILE
C
C
      DIMENSION ISTM(6),ITIME(13)
      COMMON/IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
     *,ITYPE
      COMMON LUT,LULOG,LUIN,LUWIN,NTAP,INBUF(10),LBUF(1510),LUPR
      LOGICAL IEOF
      INTEGER CKTM
      IERR=1       ! INITIALIZE FLAG TO ERROR
      CALL CNVTM(IBTIM,ITIME)
      WRITE(LULOG,1000)ITIME
      CALL CNVTM(ISTM,ITIME)
      WRITE(LULOG,1001)ITIME
 1001 FORMAT(" SEEKING :",13A2)
      IF(CKTM(IBTIM,ISTM))100,300,300      ! SEE IF WE'RE ALREADY THERE
 100  READ(LUIN,END=103,ERR=900)ITYPE,IPWR,IBTIM
      IF(ITYPE.EQ.1)GOTO104
      GOTO100
 103  WRITE(LULOG,1003)
 1003 FORMAT(" EOF ON INPUT FILE, CONTINUE? Y/N ")
      READ(LUT,1004)IANS
 1004 FORMAT(A1)
      IF(IANS.NE.1HY)RETURN
      GOTO100
 104  CALL CNVTM(IBTIM,ITIME)
      WRITE(LULOG,1000)ITIME
 1000 FORMAT(" TIME = ",13A2)
      IF(IFBRK(KK))900,101,900
 101  IF(ITIME(1).EQ.2HIN)GOTO100
      IF(CKTM(IBTIM,ISTM))100,300,250
C     IF NOT THERE YET, GO BACK TO 100 AND CONTINUE
C     IF EXACTLY THERE, GOTO 300 AND RETURN
C     IF TIME NOW IS GREATER THAN REQUESTED, ADJUST TIME AND RETURN
 250  BACKSPACE(LUIN)
      NTAP=5
      CALL REDAT(IEOF,1)
      DO 260 I=1,6
 260  ISTM(I)=IBTIM(I)
 300  IERR=0
 900  RETURN
      END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:    231       COMMON:   1526
```

```
      SUBROUTINE REDAT(IEOF,ITYP),READ NEXT ITYP RECORD
C
C
C     REDAT READS MAJOR FRAMES OF HALOE DATA FROM THE INPUT FILE.
C     IEOF IS A FLAG PASSED BACK TO MAIN PROGRAM INDICATING END-OF-FILE
C     STATUS (= TRUE IF EOF)
C
      COMMON/IDAT/IBUF(256),IFLAG,IBTIM(6),ISTAT(10),IANHK(24),IPWR(4)
     *,ITYPE
      COMMON LUT,LULOG,LUIN,LUWIN,N,INBUF(10),LBUF(1510),LUPR
C
C     N= 5 WHEN PROCESSING MAG TAPE FIRST TIME, OR WHEN SEEKING NEW
C     TIME ON MAG TAPE. NOT USED IN DISK FILE MANIPULATION.
C
      DIMENSION IBUFF(1510)
      LOGICAL IEOF
      IF(LUIN.EQ.40)THEN
1     READ(LUIN,END=900,ERR=6  ,IOSTAT=IOS)ITYPE,
     *IPWR,IBTIM,IBUF,IDUM,IANHK,ISTAT
C     WRITE(LULOG,1001)ITYPE,IBTIM
C1001 FORMAT(" RECORD TYPE, TIME ", 7I5)
      IF(ITYP.EQ.0)GOTO5
      IF(ITYPE.NE.ITYP)GOTO1
5     IEOF=.FALSE.
      RETURN
6     WRITE(LULOG,1002)
1002  FORMAT(" END OF FILE ENCOUNTERED,REWIND,CONTINUE OR STOP?"
     *" (R/C/S)")
      READ(LUT,1003)IANS
1003  FORMAT(A1)
      IF(IANS.EQ.1HC)GOTO1
      IF(IANS.EQ.1HR)THEN
      REWIND(LUIN)
      GOTO1
      ENDIF
      GOTO900
100   IF(IOS.EQ.496)GOTO1      ! ERROR WAS DUE TO SMALLER RECORD TYPE
      WRITE(LULOG,1000)IOS
1000  FORMAT(" ERROR # ",I5," IN REDAT ROUTINE")
      STOP
      ELSE
3     N=N+1
      IF(N.GE.5)THEN
      N=0
2     READ(LUIN,END= 7  ,ERR=1800,IOSTAT=IOS)IBUFF
      ENDIF
      NN=N*302
      ITYPE=IBUFF(NN+1)
      IF(ITYP.EQ.0)GOTO4
      IF(ITYPE.NE.ITYP)GOTO3
4     IEOF=.FALSE.
      CALL MVARY(IBUFF(NN+6),IBTIM(1),6)
      CALL MVARY(IBUFF(NN+12),IBUF(1),256)
      CALL MVARY(IBUFF(NN+293),ISTAT(1),10)
      CALL MVARY(IBUFF(NN+2),IPWR(1),4)
```

```
           CALL MVARY(IBUFF(NN+269),IANHK(1),24)
     C     WRITE(LULOG,1001)ITYPE,IBTIM
           RETURN
     7     WRITE(LULOG,1002)
           READ(LUT,1003)IANS
           IF(IANS.EQ.1HC)GOTO2
           IF(IANS.EQ.1HR)THEN
           REWIND(LUIN)
           GOTO2
           ENDIF
     900   IEOF=.TRUE.
           RETURN
     1800  IF(IOS.EQ.496)GOTO2
           ENDIF
           RETURN
           END


FTN4X COMPILER: HP92834 REV.2130 (810716)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM:  1828       COMMON:  1526
```

APPENDIX B - HPLOT

Program Name: HPLOT

Function: HPLOT is used to plot HALOE Blackbody data.

Description: HPLOT is a Fortran V program written on the ACD NOS facility.

Use: HPLOT can be executed using the procedure listed below. The plots will be routed to the Calcomp plotters automatically.

```
.PROC,HPLOTPR,TAPENO.
GET,HPLOT.
FTN5,I=HPLOT,L=LF.
ATTACH,LARCGOS/UN=LIBRARY,NA.
COMMENT.PROCESSING DONE FOR TAPENO DATA.
GET,TAPE1=TAPENO.
LDSET,LIB=LARCGOS,PRESETA=NGINF.
LGO.
.NOTE,(/IF YOU WANT A PRINTED OUTPUT OF DAILY AND WEEKLY
.NOTE,AVERAGE ROUTE THE TAPE4 TO LINE PRINTER AS FOLLOWS
.NOTE,ROUTE,TAPE4,DC=LP/)
REVERT.
```

```
     PROGRAM HPLOT        74/860  OPT=1,ROUND= A/ S/ M/-D,-DS     FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.

      PROGRAM HPLOT
C
C    *THIS SOFTWARE TESTS  THE PERFORMANCE OF
C    *THE BLACKBODY OF THE HALOE INSTRUMENT.
C    *THE MAIN OBJECTIVE FOR DEVELOPING THIS PROGRAM IS TO
C    *FIND THE CO-RELATION BETWEEN PRT(PLATINUM RESISTANCE
C    *THERMOMETER) AND OTHER RELATED MEASUREMENTS SUCH AS
C    *CURRENT, POWER SUPPLY VOLTAGE,ENVIRONMENTAL TEMPERATURE
C    *CHANGESSUCH AS VACUUM CHAMBER WALL TEMPERATURE,BLACKBODY
C    *CASE TEMPERATURE,BB ISOLATOR MOUNT TEMPERATURE..ETC.
C
C
C
C
      COMMON/SCALE/XSCALE
      COMMON/TOP/ANS,ISTRIN
      COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
      COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
      COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
      COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
     *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
      INTEGER BHR,EHR,BDAY,SDAY
      DIMENSION X(5000),Y1(5000),Y2(5000),Y3(5000),Y4(5000),
     *Y5(5000),Y6(5000),Y7(5000),Y8(5000),Y9(5000),Y10(5000),
     *Y11(5000),Y12(5000),Y14(5000),Y15(5000),Y16(5000),JDAY(5000)
      DIMENSION DPRT(100),DBBP(100),DBBR(100),DCKT(100),DISOL(100),
     *DCHBR(100),DCASE(100),DRAD(100),DVPS(100),DPRESS(100),DDAY(100)
      DIMENSION WPRT(15),WBBP(15),WBBR(15),WCKT(15),WISOL(15),WCHBR(15),
     *WCASE(15),WRAD(15),WVPS(15),WPRESS(15),WWK(15)
      CHARACTER ANS*1,ISTRIN*18
C
C
C
C       *****************************************************
C       *                                                   *
C       * THIS SOFTWARE PLOTS THE VARIOUS PARAMETERS        *
C       * ASSOCIATED WITH THE TESTING OF THE BLACK BODY     *
C       *                                                   *
C       * THIS PROGRAM CAN PLOT THE FOLLOWING :             *
C       * 1.PLOT PRT WITH BBP AND BBR                       *
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
C          * 2.PLOT PRT WITH BBP AND BBR USING BBH             *
C          * 3.PLOT PRT WITH TCKT AND TCHBR                    *
C          * 4.PLOT PRT WITH TISOL BBP AND TCASE               *
C          * 5.PLOT PRT WITH VPS AND R R                       *
C          * 6.PLOT BBP BBR WITH TCKT AND TCHBR                *
C          * 7.PLOT PRT WITH PRESSURE                          *
C          * 8.GET THE DAILY AVERAGE OF ALL THE PARAMETERS    **
C          * 9.QUIT.
C          *                                                   *
C          *****************************************************
C
C
C     *INITIALISE THE LARCGOS GRAPHICS PACKAGE
C
       WRITE(4,75)
75     FORMAT('DAILY AVERAGES')
       WRITE(4,76)
76     FORMAT(3X,'DAY',8X,'PRT',8X,'BBP',8X,'BBR',8X,'CKT',8X,'ISOL',
      *7X,'CHBR',7X,'CASE',7X,'RADMTR',5X,'VPS',8X,'PRESHR')
       CALL PSEUDO
C
C
C     *PRINT THE MENU OF OPTIONS ON THE SCREEN
C     *
1      PRINT *,'THIS PROGRAM CAN PLOT ANY OF THE FOLLOWING '
        PRINT *,'1.PLOT PRT WITH BBP AND BBR USING BBV'
       PRINT *,'2.PLOT PRT WITH BBP AND BBR USING BBH'
       PRINT *, '3.PLOT PRT WITH TCKT AND TCHBR'
       PRINT *, '4.PLOT PRT WITH TISOL,BBP AND TCASE '
        PRINT *, '5.PLOT PRT WITH VPS AND R R'
       PRINT *, '6.PLOT BBP AND BBR WITH TCKT AND TCHBR'
       PRINT *,'7.PLOT PRT WITH PRESSURE'
       PRINT *, '8.PLOT THE DAILY AVERAGES OF ALL THE PARAMETERS'
       PRINT *, '9.PLOT THE WEEKLY AVERAGES OF ALL THE PARAMETERS'
       PRINT *,'10.QUIT '
       PRINT *,'PLEASE SELECT THE OPTION BY TYPING THE APPROPRIATE INDEX'
       NUM=0
       M=0
       IF(EOF(5).NE.0)GO TO 999
C
C     *INPUT THE PLOT OPTION
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 39     READ *,ICAP
        IF(ICAP.EQ.10)GO TO 999
        PRINT *,'ENTER THE YEAR (YYYY)'
        READ *,IYEAR
 10     PRINT *,'ENTER THE STARTING DAY (DDD)'
        READ *,IDAYS
        PRINT *,'ENTER THE STARTING TIME (HH,MM)'
        READ *,IHR,IMIN
        PRINT *,'ENTER THE ENDING DAY (DDD)'
        READ *,IDAYE
        PRINT *,'ENTER THE ENDING TIME (HH,MM)'
        READ *,IHRE,IMINE
C
C*****THE CALIBRATION FACTOR FOR COMPUTING PRT IS .0954
C*****THE MULTIPLIER FOR THE CURRENT IS 2
C*****THE CALIBRATION FACTOR FOR THE VACCUM SYSTEM IS 10-7
C
        PRINT *,'ENTER THE CALIBRATION FACTOR TO COMPUTE PRT'
        READ *,CALFAC
        PRINT *,'ENTER MULTIPLIER FOR CURRENT'
        READ *,CURMUL
        PRINT *,'ENTER CALIBRATION FOR THE VACUUM SYSTEM'
        READ *,CALVAC
C
C     *CONVERT THE STARTING TIME TO HOURS
C
C
        STIME=FLOAT(IHR)+FLOAT(IMIN)/60.
C
C     *CONVERT THE ENDING TIME TO HOURS
C
        ETIME=FLOAT(IHRE)+FLOAT(IMINE)/60.+((IDAYE-IDAYS)*24)
        BHR=STIME
        EHR=ETIME
        WRITE(3,51)ETIME
 51     FORMAT(2X,'ETIME',2X,F8.4)
        REWIND 1
C
C
C****READ THE DATA FILE
C
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          6     READ(1,900,END=18)NDAY,NHR,NMIN,BBV,BBI,TCKT,TISOL,
               *TCHBR,PRT,TCASE,VREF,PRESS,RADIO,VPS,BBH
          900   FORMAT(3I4,12F8.3)
          36    WRITE(3,47)NDAY,NHR,NMIN
          47    FORMAT(2X,'DAY',2X,3(I6))
          C
          C    *CONVERT THE TIME TO HOURS
          C
                DTIME=FLOAT(NHR)+FLOAT(NMIN)/60.+FLOAT((NDAY-IDAYS)*24)
                WRITE(3,52)DTIME
          52    FORMAT(2X,'DTIME',2X,F8.4)
          C
          C    *IF THE ENDING TIME IS REACHED GO TO STATEMENT NO.19
          C
                IF(DTIME.GT.ETIME)THEN
1                GO TO 19
1               ENDIF
                IF(NDAY.LT.IDAYS)THEN
1                  GO TO 6
1                  ELSEIF(NDAY.GT.IDAYS .AND. NUM.EQ.0)THEN
1                 PRINT *,IDAYS,NDAY
1         101     FORMAT(2X,I6,2X,I6)
1                       PRINT 100,NDAY,NHR,NMIN
1         5            PRINT *,'DO YOU WANT TO ENTER THE DAY AND TIME AGAIN?Y/N'
1                       READ (*,'(A1)')ANS
1                       IF (ANS .EQ. 'Y')THEN
2                           GO TO 10
2                           ELSEIF(ANS.EQ.'N')THEN
2                             GO TO 1000
2                       ENDIF
1                       PPINT *,'ERROR IN INPUT '
1                       GO TO 5
1               ENDIF
                DTIME=FLOAT(NHR)+FLOAT(NMIN)/60.+FLOAT((NDAY-IDAYS)*24)
          C     IF(DTIME .GT. STIME .AND. NUM .EQ. 0)THEN
          C       PRINT *,'STARTING TIME NOT IN THE TAPE '
          C       GO TO 1000
          C     ENDIF
                IF(DTIME.LT.STIME)THEN
1                  GO TO 6
1                  ELSEIF(DTIME.LE.ETIME)THEN
1                   IF(NUM.GT.5000)THEN
```

```
 2               PRINT *,'DATA POINTS MORE THAN 5000'
 2               PRINT *,'PLEASE START ALL OVER AGAIN'
 2               GO TO 1000
 2             ELSE
 2   C
 2   C    *START ACCUMULATING THE DATA POINTS TO PLOT
 2   C
 2                  NUM=NUM+1
 2                  JDAY(NUM)=NDAY
 2                  X(NUM)=DTIME
 2              WRITE(3,28)X(NUM)
 2   28         FORMAT(2X,'TIME',2X,F7.2)
 2                  Y1(NUM)=1000.0 - PRT / CALFAC
 2                WRITE(3,12)Y1(NUM)
 2   12         FORMAT(2X,'PRT',2X,F7.2)
 2                  Y2(NUM)=BBV
 2                  BBI=BBI * CURMUL
 2                WRITE(3,14)BBI
 2   14           FORMAT(2X,'BBI',2X,F7.2)
 2                  Y3(NUM)=BBI
 2                  Y4(NUM)=TCKT
 2                  Y5(NUM)=TISOL
 2                  Y6(NUM)=TCHBR
 2                  Y7(NUM)=TCASE
 2                  Y8(NUM)=RADIO
 2                  Y9(NUM)=BBH
 2                  Y10(NUM)=VPS
 2                  Y11(NUM)=BBV*BBI
 2                WRITE(3,16)Y11(NUM)
 2   16           FORMAT(2X,'BBP',2X,F7.2)
 2                   Y12(NUM)=BBV/BBI
 2              WRITE(3,17)Y12(NUM)
 2   17        FORMAT(2X,'BBR',2X,F7.2)
 2                 Y14(NUM)=BBH*BBI
 2               WRITE(3,25)Y14(NUM)
 2   25          FORMAT(2X,'BBP USING BBH',2X,F7.2)
 2                 Y15(NUM)=BBH/BBI
 2               WRITE(3,26)Y15(NUM)
 2   26           FORMAT(2X,'BBR USING BBH',2X,F7.2)
 2                 Y16(NUM)=PRESS*CALVAC
 2                  GO TO 6
 2                ENDIF
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
 1                    ENDIF
 1            C     *GET THE DAILY AVERAGE
              19      SPRT=0.
                      SBBP=0.
                      SBBR=0.
                      SCKT=0.
                      SISOL=0.
                      SCHBR=0.
                      SCASE=0.
                      SRAD=0.
                      SPRESS=0.
                      SVPS=0.
              C
              C
                      KDAY=JDAY(1)
                      M=0
                      NPTS=0
                      DO 575 I=1,NUM
              C
              C
                      IF (JDAY(I) .GT. KDAY .OR. I .EQ. NUM)THEN
 1                       M=M+1
 1                       DPRT(M) = SPRT / NPTS
 1                       DBBP(M) = SBBP / NPTS
 1                       DBBR(M) = SBBR / NPTS
 1                       DCKT(M) = SCKT / NPTS
 1                       DISOL(M) = SISOL / NPTS
 1                       DCHBR(M) = SCHBR / NPTS
 1                       DCASE(M) = SCASE / NPTS
 1                       DRAD(M) = SRAD / NPTS
 1                       DVPS(M) = SVPS / NPTS
 1                       DPRESS(M) = SPRESS / NPTS
 1                       DDAY(M) = FLOAT(KDAY)
 1                       WRITE(4,27)DDAY(M),DPRT(M),DBBP(M),DBBR(M),DCKT(M),DISOL(M),
 1                  *    DCHBR(M),DCASE(M),DRAD(M),DVPS(M),DPRESS(M)
 1            27         FORMAT(11(1X,F9.3))
 1            C
 1            C
 1                       NPTS=1
 1                       SPRT=Y1(I)
 1                       SBBP=Y11(I)
 1                       SBBR=Y12(I)
```

```
1                   SCKT=Y4(I)
1                   SISOL=Y5(I)
1                   SCHBR=Y6(I)
1                   SCASE=Y7(I)
1                   SRAD=Y8(I)
1                   SPRESS=Y16(I)
1                   SVPS=Y10(I)
1                   KDAY=JDAY(I)
1                   ELSE
1                     SPRT = SPRT + Y1(I)
1                     SBBP = SBBP + Y11(I)
1                     SBBR = SBBR + Y12(I)
1                     SCKT = SCKT + Y4(I)
1                     SISOL = SISOL + Y5(I)
1                     SCHBR = SCHBR + Y6(I)
1                     SCASE = SCASE + Y7(I)
1                     SRAD = SRAD + Y8(I)
1                     SPRESS = SPRESS + Y16(I)
1                     SVPS = SVPS + Y10(I)
1                     NPTS = NPTS + 1
1                ENDIF
           575   CONTINUE
           C
           C    *GET THE WEEKLY AVERAGE
           C
                 WRITE(4,143)
           143   FORMAT(2X,'WEEKLY AVERAGES')
                 DO 1050 IN=1,15
                 WPRT(IN)=0.
                 WBBP(IN)=0.
                 WBBR(IN)=0.
                 WCKT(IN)=0.
                 WISOL(IN)=0.
                 WCHBR(IN)=0.
                 WCASE(IN)=0.
                 WRAD(IN)=0.
                 WVPS(IN)=0.
                 WPRESS(IN)=0.
           1050  CONTINUE
                 NW=M/7
                 DO 590 LW=1,NW
                 ID=(LW-1)*7 + 1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IDE=ID+6
      DO 580 KD=ID,IDE
      WPRT(LW)=WPRT(LW)+DPRT(KD)
      WBBP(LW)=WBBP(LW)+DBBP(KD)
      WBBR(LW)=WBBR(LW)+DBBR(KD)
      WCKT(LW)=WCKT(LW)+DCKT(KD)
      WISOL(LW)=WISOL(LW)+DISOL(KD)
      WCHBR(LW)=WCHBR(LW)+DCHBR(KD)
      WCASE(LW)=WCASE(LW)+DCASE(KD)
      WRAD(LW)=WRAD(LW)+DRAD(KD)
      WVPS(LW)=WVPS(LW)+DVPS(KD)
      WPRESS(LW)=WPRESS(LW)+DPRESS(KD)
580   CONTINUE
      WPRT(LW)=WPRT(LW)/7.
      WBBP(LW)=WBBP(LW)/7.
      WBBR(LW)=WBBR(LW)/7.
      WCKT(LW)=WCKT(LW)/7.
      WISOL(LW)=WISOL(LW)/7.
      WCHBR(LW)=WCHBR(LW)/7.
      WCASE(LW)=WCASE(LW)/7.
      WRAD(LW)=WRAD(LW)/7.
      WVPS(LW)=WVPS(LW)/7.
      WPRESS(LW)=WPRESS(LW)/7.
      WWK(LW)=LW
      WRITE(4,165)WWK(LW),WPRT(LW),WBBP(LW),WBBR(LW),WCKT(LW),
     *WISOL(LW),WCHBR(LW),WCASE(LW),WRAD(LW),WPRESS(LW)
165   FORMAT(11(1X,F9.3))
590   CONTINUE
18    CONTINUE
      WRITE(3,30)DPRT(1),DBBP(1),DBBR(1),M
30    FORMAT(1X,3F7.1,1X,I3)
100   FORMAT(2X,'YOU HAVE ASKED FOR DAY BEFORE THE DATA PERIOD'/
     *'THE STARTING TIME ON THE TAPE IS  ',2X,I4,2X,I2,2X,I2)
      DO 400 I=2,NUM
      X(I)=X(I)-X(1)
400   CONTINUE
      X(1)=0.
      FYEAR=FLOAT(IYEAR)
      FHR=FLOAT(IHR)
      FMIN=FLOAT(IMIN)
      FHRE=FLOAT(IHRE)
      FMINE=FLOAT(IMINE)
      BDAY=JDAY(1)
      SDAY=JDAY(NUM)
C
C
C    *SUBROUINE DMDATE IS CALLED TO CONVERT THE DAY NUMBER
C     TO MONTH AND DATE
C
C
      CALL DMDATE(BDAY,SDAY)
      XSCALE=(EHR - BHR) / 8.
C
C
C    *BRANCHES TO RESPECTIVE SUBROUTINES DEPENDING UPON THE
C    *SELECTION OF THE OPTION
C
C
      IF(ICAP .EQ.1)THEN
       CALL SUB1(Y1,Y11,Y12,Y14,Y15,X)
       ELSEIF(ICAP.EQ.2)THEN
        CALL SUB2(Y1,Y14,Y15,X)
        ELSEIF(ICAP .EQ.3)THEN
         CALL SUB3(Y1,Y4,Y6,X,Y11,Y12)
         ELSEIF(ICAP .EQ. 4)THEN
          CALL SUB4(Y1,Y5,Y7,Y11,X)
          ELSEIF(ICAP .EQ.5)THEN
           CALL SUB5(Y1,Y10,Y8,X)
           ELSEIF(ICAP .EQ.6)THEN
            CALL SUB6(Y11,Y12,Y4,Y6,X)
            ELSEIF(ICAP.EQ.7)THEN
             CALL SUB7(Y1,Y16,X)
             ELSEIF(ICAP.EQ.8)THEN
                 CALL SUB8(DDAY,DPRT,DBBP,DBBR,DCKT,DISOL,
     *           DCHBR,DCASE,DRAD,DVPS,DPRESS,M,IDAYS,IDAYE)
                 ELSEIF(ICAP.EQ.9)THEN
                    CALL SUB9(WWK,WPRT,WBBP,WBBR,WCKT,WISOL,
     *             WCHBR,WCASE,WRAD,WVPS,WPRESS,NW)
                    ELSEIF(ICAP.EQ.10)THEN
                   GO TO 999
      ENDIF
      REWIND 1
      GO TO 1
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C     *TERMINATE THE GRAPHICS PACKAGE
      C:
      999   CALL CALPLT(0.,0.,999)
      1000  STOP
            END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES

  ANS          0B  /TOP/                      CHAR*1                  FMIN         2B  /TIM/
  BBH     252353B                             REAL                    FMINE        4B  /TIM/
  BBI     252341B                             REAL                    FMNTH        0B  /DATE/
  BBPMIN       2B  /KEY/                      REAL                    FYEAR        0B  /TIM/
  BBPSF        3B  /KEY/                      REAL                    I       252371B
  BBRMIN       4B  /KEY/                      REAL                    ICAP    252320B
  BBRSF        5B  /KEY/                      REAL                    ID      252377B
  BBV     252340B                             REAL                    IDAYE   252325B
  BDAY      1724B                             INTEGER                 IDAYS   252322B
  BHR       1722B                             INTEGER                 IDE     252400B
  CALFAC  252330B                             REAL                    IHR     252323B
  CALVAC  252332B                             REAL                    IHRE    252326B
  CURMUL  252331B                             REAL                    IMIN    252324B
  DBBP    250102B                             REAL          100       IMINE   252327B
  DBBR    250246B                             REAL          100       IN      252373B
  DCASE   251066B                             REAL          100       ISTRIN       0B  /TOP/
  DCHBR   250722B                             REAL          100       IYEAR   252321B
  DCKT    250412B                             REAL          100       JDAY    236126B
  DDAY    251706B                             REAL          100       KD      252401B
  DISOL   250556B                             REAL          100       KDAY    252367B
  DPRESS  251542B                             REAL          100       LW      252375B
  DPRT    247736B                             REAL          100       M       252317B
  DRAD    251232B                             REAL          100       NDAY    252335B
  DTIME   252354B                             REAL                    NHR     252336B
  DVPS    251376B                             REAL          100       NMIN    252337B
  EDAY         3B  /DATE/                     REAL                    NPTS    252370B
  EHR       1723B                             INTEGER                 NUM          6B  /KEY1/
  EMNTH        2B  /DATE/                     REAL                    NW      252374B
  ETIME   252334B                             REAL                    PREMIN      15B  /KEY1/
  FDAY         1B  /DATE/                     REAL                    PRESF       16B  /KEY1/
  FHR          1B  /TIM/                      REAL                    PRESS   252350B
  FHRE         3B  /TIM/                      REAL                    PRT     252345B
```

```
SUBROUTINE DMDATE      74/860  OPT=1,ROUND= A/ S/ M/-D,-DS     FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.

      C
      C
      C     *SUBROUTINE DMDATE TO GET THE MONTH AND DATE
      C
      C
            SUBROUTINE DMDATE(BDAY,SDAY)
            COMMON /DATE/FMNTH,FDAY,EMNTH,EDAY
            INTEGER BDAY,SDAY
            DIMENSION MNTH(12)
      C
      C
            DATA MNTH /1,32,60,91,121,152,182,213,244,274,305,335/
      C
      C
            DO 500 I=1,11
              IF(BDAY.LT.MNTH(I+1))THEN
1                FMNTH=FLOAT(I)
1                FDAY=FLOAT((BDAY-MNTH(I))+1)
1                GO TO 600
1             ENDIF
1     C
      500   CONTINUE
      C
      550   FMNTH=12
            FDAY=FLOAT((BDAY-MNTH(12))+1)
      C
      600   CONTINUE
      C
            DO 800 I=1,11
               IF(SDAY.LT.MNTH(I+1))THEN
1                EMNTH=FLOAT(I)
1                EDAY=FLOAT((SDAY-MNTH(I))+1)
1                GO TO 900
1              ENDIF
      800   CONTINUE
      C
      850   EMNTH=12
            EDAY=FLOAT((SDAY-MNTH(12))+1)
      C
      900   CONTINUE
```

```
   41                   RETURN
   42                   END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES

  BDAY          1   DUMMY-ARG                 INTEGER               FMNTH         0B  /DATE/
  EDAY          3B  /DATE/                    REAL                  I            65B
  EMNTH         2B  /DATE/                    REAL                  MNTH         51B
  FDAY          1B  /DATE/                    REAL                  SDAY          2   DUMMY-ARG


--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----

  FLOAT     REAL            1      INTRINSIC


--STATEMENT LABELS--(LO=A)
 -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF

    500  INACTIVE   DO-TERM       22           800  INACTIVE   DO-TERM       35
    550 *NO REFS*                 24           850 *NO REFS*                 37
    600      25B                  27           900      44B                  40


--ENTRY POINTS--(LO=A)
 -NAME---ADDRESS--ARGS---

  DMDATE      3B     2


--STATISTICS--

  PROGRAM-UNIT LENGTH             72B =  58
  CM LABELLED COMMON LENGTH        4B =  4
  CM STORAGE USED              61400B =  25344
  COMPILE TIME                 0.384 SECONDS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C
C
      SUBROUTINE SUB1(Y1,Y11,Y12,Y14,Y15,X)
C
C*****THE FOLLOWING SUBROUTINE SUB1 PLOTS PRT,BBP,BBR***
C*****WITH TIME ON THE X-AXIS*********
C*****THIS SUBROUTINE USES BBV TO COMPUTE THE POWER AND**
C*****RESISTANCE******
      COMMON/SCALE/XSCALE
      COMMON/TOP/ANS,ISTRIN
      COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
      COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
      COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
      COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
     *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
      DIMENSION Y1(NUM+2),Y11(NUM+2),Y12(NUM+2),X(NUM+2),
     *Y14(NUM+2),Y15(NUM+2)
      CHARACTER * 1 ANS
      CHARACTER ISTRIN*18
      DATA ISTRIN/'ELAPSED TIME(HRS)'/
      PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
      PRINT *,'PRT MIN ....980K'
      PRINT *,'PRT SF ....10 '
      PRINT *,'BBP MIN ...2 WATTS'
      PRINT *,'BBP SF ...2 '
      PRINT *,'BBR MIN ...2.5 OHMS'
      PRINT *,'BBR SF ......(.5)'
5     PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
      READ (*,'(A1)')ANS
      IF(ANS.EQ.'Y')THEN
        PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
        READ *,PRTMIN,PRTSF
      PRINT *,'INPUT THE NEW BBP MIN AND SF'
      READ *,BBPMIN,BBPSF
      PRINT *,'INPUT THE NEW BBR MIN AND SF'
      READ *,BBRMIN,BBRSF
        ELSEIF(ANS.EQ.'N')THEN
           GO TO 100
           ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
1                     PRINT *,'WRONG INPUT'
1                     GO TO 5
1                 ENDIF
         100      Y1(NUM+1)=PRTMIN
                  Y1(NUM+2)=PRTSF
                  Y11(NUM+1)=BBPMIN
                  Y11(NUM+2)=BBPSF
                  Y12(NUM+1)=BBRMIN
                  Y12(NUM+2)=BBRSF
                  CALL LEROY
                  CALL NEWPEN(1)
                  CALL CALPLT(2.,1.,-3)
                  X(NUM+1)=0.
                  X(NUM+2)=XSCALE
                  CALL AXES(0.,0.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,ISTRIN,.14,-18)
                  CALL AXES(0.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,'PRT',.14,3)
                  CALL AXES(0.,5.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,' ',0.0,1)
                  CALL AXES(9.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,' ',0.0,-1)
                  CALL NEWPEN (1)
                  CALL LINPLT(X,Y1,NUM,1,0,0,1,1)
                  CALL NEWPEN (2)
                  CALL CALPLT(-1.,0.,-3)
                  CALL AXES(0.,0.,90.,5.,Y11(NUM+1),Y11(NUM+2),1.,10.,'BBP',
                 *.14,3)
                  CALL CALPLT(1.,0.,-3)
                  CALL NEWPEN(2)
                  CALL LINPLT(X,Y11,NUM,1,0,0,1,2)
                  CALL NEWPEN (3)
                  CALL CALPLT(10.,0.,-3)
                  CALL AXES(0.,0.,90.,5.,Y12(NUM+1),Y12(NUM+2),1.,10.,'BBR',
                 *.14,3)
                  CALL CALPLT(-10.,0.,-3)
                  CALL NEWPEN(3)
                  CALL LINPLT(X,Y12,NUM,1,0,0,1,3)
                  CALL NEWPEN(1)
                  CALL HEADR
                  CALL NFRAME
                  RETURN
                  END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
SUBROUTINE SUB2        74/860  OPT=1,ROUND= A/ S/ M/-D,-DS      FTN 5.1+642           87/04/30. 09.48.(
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.

      C
            SUBROUTINE SUB2(Y1,Y14,Y15,X)
      C
      C*****THE FOLLOWING SUBROUTINE SUB2 PLOTS PRT,BBP,BBR**
      C*****THIS USES BBH TO COMPUTE BBPAND BBR***
      C
            COMMON/TOP/ANS,ISTRIN
            COMMON/SCALE/XSCALE
            COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
            COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
            COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
            COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
           *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
            CHARACTER ANS*1,ISTRIN*18
            DIMENSION Y1(NUM+2),Y14(NUM+2),Y15(NUM+2),X(NUM+2)
            DATA ISTRIN/'ELAPSED TIME(HRS)'/
            PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS :'
            PRINT *,'PRT MIN .....980K'
            PRINT *,'PRT SF .....10'
            PRINT *,'BBP MIN .....2WATTS'
            PRINT *,'BBP SF .....2'
            PRINT *,'BBR MIN ....3.5 OHMS'
            PRINT *,'BBR SF ....(.5)'
      5     PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
            READ (*,'(A1)')ANS
            IF(ANS.EQ.'Y')THEN
1            PRINT *,'INPUT THE NEW PRT MIN AND SF'
1            READ *,PRTMIN,PRTSF
1            PRINT *,'INPUT THE NEW BBP MIN AND SF'
1            READ *,BBPMIN,BBPSF
1            PRINT *, 'INPUT THE NEW BBR MIN AND SF'
1            READ *,BBRMIN,BBRSF
1            ELSEIF(ANS.EQ.'N')THEN
1            GO TO 100
1            ELSEIF(ANS.NE.'Y'.OR.ANS.NE.'N')THEN
1             PRINT *,'WRONG INPUT'
1             GO TO 5
1           ENDIF
      100   Y1(NUM+1)=PRTMIN
            Y1(NUM+2)=PRTSF
      Y14(NUM+1)=BBPMIN
      Y14(NUM+2)=BBPSF
      Y15(NUM+1)=BBRMIN
      Y15(NUM+2)=BBRSF
      CALL LERDY
      CALL CALPLT(2.,1.,-3)
      X(NUM+1)=0.
      X(NUM+2)=XSCALE
      CALL AXES(0.,0.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,ISTRIN,.14,-18)
      CALL AXES(0.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,'PRT',.14,3)
      CALL AXES(0.,5.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,' ',0.0,1)
      CALL AXES(9.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,' ',0.0,-1)
      CALL NEWPEN (1)
      CALL LINPLT(X,Y1,NUM,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,Y14(NUM+1),Y14(NUM+2),1.,10.,'BBP',
     *.14,3)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(X,Y14,NUM,1,0,0,1,2)
      CALL NEWPEN (3)
      CALL CALPLT(10.,0.,-3)
      CALL AXES(0.,0.,90.,5.,Y15(NUM+1),Y15(NUM+2),1.,10.,'BBR',
     *.14,3)
      CALL CALPLT(-10.,0.,-3)
      CALL LINPLT(X,Y15,NUM,1,0,0,1,3)
      CALL NEWPEN(1)
      CALL HEADR
      CALL NFRAME
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES-

  ANS          0B  /TOP/                     CHAR*1                  EMNTH        2B  /DATE/
  BBPMIN       2B  /KEY/                     REAL                    FDAY         1B  /DATE/
  BBPSF        3B  /KEY/                     REAL                    FHR          1B  /TIM/
  BBRMIN       4B  /KEY/                     REAL                    FHRE         3B  /TIM/
  BBRSF        5B  /KEY/                     REAL                    FMIN         2B  /TIM/
  EDAY         3B  /DATE/                    REAL                    FMINE        4B  /TIM/
```

```
      C
            SUBROUTINE SUB3(Y1,Y4,Y6,X,Y11,Y12)
      C
      C
      C*****THE FOLLOWING SUBROUTINE SUB3 PLOTS PRT,CIRCUIT TEMP*
      C*****AND BB CHAMBER TEMP**********
      C
            COMMON/TOP/ANS,ISTRIN
            COMMON/SCALE/XSCALE
            COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
            COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
            COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
            COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
           *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
            DIMENSION Y1(NUM+2),Y4(NUM+2),Y6(NUM+2),X(NUM+2),Y11(NUM+2),
           *Y12(NUM+2)
            CHARACTER ANS*1,ISTRIN*18
            DATA ISTRIN/'ELAPSED TIME(HRS)'/
            PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS  :'
            PRINT *,'PRT MIN ........980 K'
            PRINT *,'PRT SF ........10'
            PRINT *,'TCKT MIN ........20 DEG'
            PRINT *,'TCKT SF ........5.'
            PRINT *,'TCH MIN  ........15 DEG'
            PRINT *,'TCH SF  ........(5.)'
            PRINT *,'BBP MIN .........2 WATTS'
            PRINT *,'BBP SF .........2'
            PRINT *,'BBR MIN ............2.5 OHMS'
            PRINT *,'BBR SF ............(.5)'
      5     PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF ?(Y/N)'
            READ (*,'(A1)')ANS
            IF(ANS .EQ. 'Y')THEN
1           PRINT *,'INPUT THE NEW MIN AND SF PRT VALUES'
1           READ *,PRTMIN,PRTSF
1           PRINT *,'INPUT THE NEW MIN AND SF T CKT VALUES'
1           READ *,TCKMIN,TCKSF
1           PRINT *,'INPUT THE NEW MIN AND SF T CHBR VALUES'
1           READ *,TCHMIN,TCHSF
1           PRINT *,'INPUT THE NEW BBP MIN AND SF'
1           READ *,BBPMIN,BBPSF
```

```
      PRINT *,'INPUT THE NEW BBR MIN AND SF'
      READ *,BBRMIN,BBRSF
          ELSEIF(ANS .EQ.'N')THEN
             GO TO 100
             ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
                PRINT *,'WRONG INPUT'
                GO TO 5
      ENDIF
100   Y1(NUM+1)=PRTMIN
      Y1(NUM+2)=PRTSF
      Y4(NUM+1)=TCKMIN
      Y4(NUM+2)=TCKSF
      Y6(NUM+1)=TCHMIN
      Y6(NUM+2)=TCHSF
      CALL LEROY
      CALL CALPLT(2.,1.,-3)
      X(NUM+1)=0.
      X(NUM+2)=XSCALE
      CALL AXES(0.,0.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,ISTRIN,.14,-18)
      CALL AXES(0.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,'PRT',.14,3)
      CALL AXES(0.,5.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,' ',0.0,1)
      CALL AXES(9.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,' ',0.0,-1)
      CALL NEWPEN (1)
      CALL LINPLT(X,Y1,NUM,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,Y4(NUM+1),Y4(NUM+2),1.,10.,'CIRCUIT',
     *.14,7)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(X,Y4,NUM,1,0,0,1,2)
      CALL NEWPEN (3)
      CALL CALPLT(10.,0.,-3)
      CALL AXES(0.,0.,90.,5.,Y6(NUM+1),Y6(NUM+2),1.,10.,'CHAMBER',
     *.14,7)
      CALL CALPLT(-10.,0.,-3)
      CALL LINPLT(X,Y6,NUM,1,0,0,1,3)
      CALL NEWPEN(1)
      CALL HEADR
      CALL NFRAME
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      C
              SUBROUTINE SUB4(Y1,Y5,Y7,Y11,X)
      C
      C
      C      *THE FOLLOWING SUBROUTINE SUB4 PLOTS PRT,ISOLATOR TEMP***
      C      *BB CASE TEMP BBP******
      C
              COMMON/TOP/ANS,ISTRIN
              COMMON/SCALE/XSCALE
              COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
              COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
              COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
              COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
             *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
              CHARACTER ANS*1,ISTRIN*18
              DIMENSION Y1(NUM+2),Y5(NUM+2),Y7(NUM+2),Y11(NUM+2),X(NUM+2)
              DATA ISTRIN/'ELAPSED TIME(HRS)'/
              PRINT *,'THE DEFAULT GRAFING LIMITS ARE AS FOLLOWS : '
              PRINT *,'PRTMIN .......980 K'
              PRINT *,'PRTSF .......10'
              PRINT *,'T ISOL MIN ...110 DEG'
              PRINT *,'T ISOL SF ...(5.)'
              PRINT *,'T CASE MIN ...115 DEG'
              PRINT *,'T CASE SF ...5.'
              PRINT *,'BBP MIN...2 VOLTS'
              PRINT *,'BBP SF...10VOLTS'
      5       PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF (Y/N)'
              READ (*,'(A1)')ANS
              IF(ANS .EQ. 'Y')THEN
1             PRINT *,'INPUT THE NEW MIN AND SF PRT VALUES'
1             READ *,PRTMIN,PRTSF
1             PRINT *,'INPUT THE NEW MIN AND SF ISOL VALUES'
1             READ *,TSOLMIN,TSOLSF
1             PRINT *,'INPUT THE NEW MIN AND SF T CASE VALUES'
1             READ *,TCASMIN,TCASSF
1             PRINT *,'INPUT THE NEW BBP MIN ANS SF VALUES'
1             READ *,BBPMIN,BBPSF
1             ELSEIF(ANS .EQ. 'N')THEN
1                     GO TO 100
1                     ELSEIF(ANS.NE.'Y' .OR. ANS.NE.'N')THEN
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
1                     PRINT *,'WRONG INPUT'
1                     GO TO 5
1              ENDIF
        100    Y1(NUM+1)=PRTMIN
               Y1(NUM+2)=PRTSF
               Y11(NUM+1)=BBPMIN
               Y11(NUM+2)=BBPSF
               Y5(NUM+1)=TSOLMIN
               Y5(NUM+2)=TSOLSF
               Y7(NUM+1)=TCASMIN
               Y7(NUM+2)=TCASSF
               CALL LEROY
               CALL CALPLT(2.,1.,-3)
               X(NUM+1)=0.
               X(NUM+2)=XSCALE
               CALL AXES(0.,0.,0.,8.,X(NUM+1),X(NUM+2),1.,10.,ISTRIN,.14,-18)
               CALL AXES(0.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,'PRT',.14,3)
               CALL AXES(0.,5.,0.,8.,X(NUM+1),X(NUM+2),1.,10.,' ',0.0,1)
               CALL AXES(8.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,' ',0.0,-1)
               CALL NEWPEN (1)
               CALL LINPLT(X,Y1,NUM,1,0,0,1,1)
               CALL NEWPEN (2)
               CALL CALPLT(-1.,0.,-3)
               CALL AXES(0.,0.,90.,5.,Y11(NUM+1),Y11(NUM+2),1.,10.,'BBP',
              *.14,3)
               CALL CALPLT(1.,0.,-3)
               CALL LINPLT(X,Y11,NUM,1,0,0,1,2)
               CALL NEWPEN (3)
               CALL CALPLT(9.,0.,-3)
               CALL AXES(0.,0.,90.,5.,Y5(NUM+1),Y5(NUM+2),1.,10.,'ISOLATOR',
              *.14,8)
               CALL CALPLT(-9.,0.,-3)
               CALL LINPLT(X,Y5,NUM,1,0,0,1,3)
               CALL NEWPEN(4)
               CALL CALPLT(10.,0.,-3)
               CALL AXES(0.,0.,90.,5.,Y7(NUM+1),Y7(NUM+2),1.,10.,'CASE',
              *.14,4)
               CALL CALPLT(-10.,0.,-3)
               CALL LINPLT(X,Y7,NUM,1,0,0,1,3)
               CALL NEWPEN(1)
               CALL HEADR
               CALL NFRAME
```

ORIGINAL PAGE IS
OF POOR QUALITY.

```
      SUBROUTINE SUB4        74/860  OPT=1,ROUND= A/ S/ M/-D,-DS     FTN 5.1+642          87/04/30. 09.48.

     83                 RETURN
     84                 END


--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE      -NAME---ADDRESS --BLOCK-----PROPERTIES

  ANS          0B   /TOP/                     CHAR*1                   PRTSF         1B   /KEY/
  BBPMIN       2B   /KEY/                     REAL                     RADMIN       13B   /KEY1/
  BBPSF        3B   /KEY/                     REAL                     RADSF        14B   /KEY1/
  BBRMIN       4B   /KEY/                     REAL                     TCASMIN       7B   /KEY1/
  BBRSF        5B   /KEY/                     REAL                     TCASSF       10B   /KEY1/
  EDAY         3B   /DATE/                    REAL                     TCHMIN        2B   /KEY1/
  EMNTH        2B   /DATE/                    REAL                     TCHSF         3B   /KEY1/
  FDAY         1B   /DATE/                    REAL                     TCKMIN        0B   /KEY1/
  FHR          1B   /TIM/                     REAL                     TCKSF         1B   /KEY1/
  FHRE         3B   /TIM/                     REAL                     TSOLMIN       4B   /KEY1/
  FMIN         2B   /TIM/                     REAL                     TSOLSF        5B   /KEY1/
  FMINE        4B   /TIM/                     REAL                     VPSMIN       11B   /KEY1/
  FMNTH        0B   /DATE/                    REAL                     VPSSF        12B   /KEY1/
  FYEAR        0B   /TIM/                     REAL                     X             5    DUMMY-ARG
  ISTRIN       0B   /TOP/                     CHAR*18                  XSCALE        0B   /SCALE/
  NUM          6B   /KEY1/                    INTEGER                  Y1            1    DUMMY-ARG
  PREMIN      15B   /KEY1/                    REAL                     Y11           4    DUMMY-ARG
  PRESF       16B   /KEY1/                    REAL                     Y5            2    DUMMY-ARG
  PRTMIN       0B   /KEY/                     REAL                     Y7            3    DUMMY-ARG

--PROCEDURES--(LO=A)
 -NAME------TYPE--------ARGS------CLASS-----      -NAME------TYPE--------ARGS------CLASS-----

  AXES                     11       SUBROUTINE      LINPLT                    8       SUBROUTINE
  CALPLT                    3       SUBROUTINE      NEWPEN                    1       SUBROUTINE
  HEADR                     0       SUBROUTINE      NFRAME                    0       SUBROUTINE
  LEROY                     0       SUBROUTINE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
SUBROUTINE SUB5        74/860  OPT=1,ROUND= A/ S/ M/-D,-DS     FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.

           C
                 SUBROUTINE SUB5(Y1,Y10,Y8,X)
           C
           C
           C*****THE FOLLOWING SUBROUTINE SUB5 PLOTS PRT,VPS AND**
           C*****RADIOMETRIC DATA***
           C
                 COMMON/TOP/ANS,ISTRIN
                 COMMON/SCALE/XSCALE
                 COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
                 COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
                 COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
                 COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
                *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
                 DIMENSION Y1(NUM+2),Y10(NUM+2),Y8(NUM+2),X(NUM+2)
                 CHARACTER ANS*1,ISTRIN*18
                 DATA ISTRIN/'ELAPSED TIME(HRS)'/
                 PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS :'
                 PRINT *,'PRT MIN ......980 K'
                 PRINT *,'PRT SF ......10'
                 PRINT *,'VPS MIN ......0  VOLTS'
                 PRINT *,'VPS SF ......(2.)'
                 PRINT *,'RAD MIN .....(-2)  C'
                 PRINT *,'RAD SF ......2.'
           5     PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
                 READ (*,'(A1)')ANS
                 IF(ANS .EQ. 'Y')THEN
1                   PRINT *,'INPUT THE NEW MIN AND SF PRT VALUES '
1                   READ *,PRTMIN,PRTSF
1                PRINT *,'INPUT THE NEW MIN AND SF VPS VALUES '
1                READ *,VPSMIN,VPSSF
1                PRINT *,'INPUT THE NEW MIN AND SF RADIOMETRIC VAL'
1                READ *,RADMIN,RADSF
1                ELSEIF(ANS.EQ.'N')THEN
1                    GO TO 100
1                    ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
1                       PRINT *,'WRONG INPUT'
1                       GO TO 5
1                ENDIF
           100   Y1(NUM+1)=PRTMIN
```

```
      Y1(NUM+2)=PRTSF
      Y10(NUM+1)=VPSMIN
      Y10(NUM+2)=VPSSF
      Y8(NUM+1)=RADMIN
      Y8(NUM+2)=RADSF
      CALL LEROY
      CALL CALPLT(2.,1.,-3)
      X(NUM+1)=0.
      X(NUM+2)=XSCALE
      CALL AXES(0.,0.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,ISTRIN,.14,-18)
      CALL AXES(0.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,'PRT',.14,3)
      CALL AXES(0.,5.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,' ',0.0,1)
      CALL AXES(9.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,' ',0.0,-1)
      CALL NEWPEN (1)
      CALL LINPLT(X,Y1,NUM,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,Y10(NUM+1),Y10(NUM+2),1.,10.,'VPS',
     *.14,3)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(X,Y10,NUM,1,0,0,1,2)
      CALL NEWPEN (3)
      CALL CALPLT(10.,0.,-3)
      CALL AXES(0.,0.,90.,5.,Y8(NUM+1),Y8(NUM+2),1.,10.,'RADIOMETRIC',
     *.14,11)
      CALL CALPLT(-10.,0.,-3)
      CALL LINPLT(X,Y8,NUM,1,0,0,1,3)
      CALL NEWPEN(1)
      CALL HEADR
      CALL NFRAME
      RETURN
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES-

  ANS          0B   /TOP/                     CHAR*1                 EDAY          3B   /DATE/
  BBPMIN       2B   /KEY/                     REAL                   EMNTH         2B   /DATE/
  BBPSF        3B   /KEY/                     REAL                   FDAY          1B   /DATE/
  BBRMIN       4B   /KEY/                     REAL                   FHR           1B   /TIM/
  BBRSF        5B   /KEY/                     REAL                   FHRE          3B   /TIM/
```

```
SUBROUTINE SUB6       74/860  OPT=1,ROUND= A/ S/ M/-D,-DS     FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.

C
      SUBROUTINE SUB6(Y11,Y12,Y4,Y6,X)
C
C*****THE FOLLOWING SUBROUTINE SUB6 PLOTS CIRCUIT TEMP
C*****CHAMBER TEMP BBP AND BBR*****
C
      COMMON/TOP/ANS,ISTRIN
      COMMON/SCALE/XSCALE
      COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
      COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
      COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
      COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
     *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
      DIMENSION Y11(NUM+2),Y12(NUM+2),Y4(NUM+2),Y6(NUM+2),X(NUM+2)
      CHARACTER ANS*1,ISTRIN*18
      DATA ISTRIN/'ELAPSED TIME(HRS)'/
      PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
      PRINT *,'TCKMIN .......20 C'
      PRINT *,'TCKSF.......(5.)'
      PRINT *,'TCHMIN .......15 C'
      PRINT *,'TCHSF .......(5.)'
      PRINT *,'BBPMIN ....... 2 WATTS'
      PRINT *,'BBPSF .......2'
      PRINT *,'BBRMIN .......3.5 OHMS'
      PRINT *,'BBRSF .........(.5)'
5     PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF(Y/N)?'
      READ (*,'(A1)')ANS
      IF(ANS .EQ.'Y')THEN
         PRINT *,'INPUT THE NEW MIN AND SF TCK VALUES '
         READ *,TCKMIN,TCKSF
         PRINT *,'INPUT THE NEW MIN AND SF TCH VALUES '
         READ *,TCHMIN,TCHSF
         PRINT *,'INPUT THE NEW MIN AND SF BBP VALUES'
         READ *,BBPMIN,BBPSF
         PRINT *,'INPUT THE NEW MIN AND SF BBR VALUES '
         READ *,BBRMIN,BBRSF
         ELSEIF(ANS.EQ.'N')THEN
            GO TO 100
            ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
               PRINT *,'WRONG INPUT'
```

```
1                        GO TO 5
1                  ENDIF
             100   Y4(NUM+1)=TCKMIN
                   Y4(NUM+2)=TCKSF
                   Y11(NUM+1)=BBPMIN
                   Y11(NUM+2)=BBPSF
                   Y12(NUM+1)=BBRMIN
                   Y12(NUM+2)=BBRSF
                   Y6(NUM+1)=TCHMIN
                   Y6(NUM+2)=TCHSF
                   CALL LEROY
                   CALL CALPLT(2.,1.,-3)
                   X(NUM+1)=0.
                   X(NUM+2)=XSCALE
                   CALL AXES(0.,0.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,ISTRIN,.14,-18)
                   CALL AXES(0.,0.,90.,5.,Y4(NUM+1),Y4(NUM+2),1.,10.,
                  *'CIRCUIT',.14,7)
                   CALL AXES(0.,5.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,' ',0.0,1)
                   CALL AXES(9.,0.,90.,5.,Y4(NUM+1),Y4(NUM+2),1.,10.,' ',0.0,-1)
                   CALL NEWPEN (1)
                   CALL LINPLT(X,Y4,NUM,1,0,0,1,1)
                   CALL NEWPEN (2)
                   CALL CALPLT(-1.,0.,-3)
                   CALL AXES(0.,0.,90.,5.,Y11(NUM+1),Y11(NUM+2),1.,10.,'BBP',
                  *.14,3)
                   CALL CALPLT(1.,0.,-3)
                   CALL LINPLT(X,Y11,NUM,1,0,0,1,2)
                   CALL NEWPEN (3)
                   CALL CALPLT(10.,0.,-3)
                   CALL AXES(0.,0.,90.,5.,Y12(NUM+1),Y12(NUM+2),1.,10.,'BBR',
                  *.14,3)
                   CALL CALPLT(-10.,0.,-3)
                   CALL LINPLT(X,Y12,NUM,1,0,0,1,3)
                   CALL NEWPEN(1)
                   CALL HEADR
                   CALL NFRAME
                   RETURN
                   END
```

```
SUBROUTINE SUB7       74/860  OPT=1,ROUND= A/ S/ M/-D,-DS     FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.

      C*****THE FOLLOWING SUBROUTINE SUB7 PLOTS PRT AND PRESSURE*
      C
            SUBROUTINE SUB7(Y1,Y16,X)
      C
      C*****THE FOLLOWING SUBROUTINE SUB7 PLOTS PRT AND PRESSURE*
      C
            COMMON/TOP/ANS,ISTRIN
            COMMON/SCALE/XSCALE
            COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
            COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
            COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
            COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
           *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
            DIMENSION Y1(NUM+2),Y16(NUM+2),X(NUM+2)
            CHARACTER ANS*1,ISTRIN*18
            DATA ISTRIN/'ELAPSED TIME(HRS)'/
            PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS :'
            PRINT *,'PRT MIN ...980'
            PRINT *,'PRT SF ...1020'
            PRINT *,'PRESSURE MIN ....-.01'
            PRINT *,'PRESSURE SF ....-.1 '
      5     PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
            READ (*,'(A1)')ANS
            IF(ANS.EQ.'Y')THEN
              PRINT *,'INPUT THE NEW PRT MIN AND SF'
              READ *,PRTMIN,PRTSF
              PRINT *,'INPUT THE NEW PRESSURE MIN AND SF'
              READ *,PREMIN,PRESF
              ELSEIF(ANS.EQ.'N')THEN
                GO TO 100
                ELSEIF(ANS.NE.'Y'.OR.ANS.NE.'N')THEN
              PRINT *,'WRONG INPUT'
              GO TO 5
            ENDIF
      100   Y1(NUM+1)=PRTMIN
            Y1(NUM+2)=PRTSF
            Y16(NUM+1)=PREMIN
            Y16(NUM+2)=PRESF
            CALL LEROY
            CALL CALPLT(2.,1.,-3)
      X(NUM+1)=0.
      X(NUM+2)=XSCALE
      CALL AXES(0.,0.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,ISTRIN,.14,-18)
      CALL AXES(0.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,'PRT',.14,3)
      CALL AXES(0.,5.,0.,9.,X(NUM+1),X(NUM+2),1.,10.,' ',0.0,1)
      CALL AXES(9.,0.,90.,5.,Y1(NUM+1),Y1(NUM+2),1.,10.,' ',0.0,-1)
      CALL NEWPEN (1)
      CALL LINPLT(X,Y1,NUM,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,Y16(NUM+1),Y16(NUM+2),1.,10.,'PRESSURE',
     *.14,8)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(X,Y16,NUM,1,0,0,1,2)
      CALL NEWPEN(1)
      CALL HEADR
      CALL NFRAME
      RETURN
      END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES

  ANS          0B  /TOP/                      CHAR*1                PRTMIN       0B  /KEY/
  BBPMIN       2B  /KEY/                      REAL                  PRTSF        1B  /KEY/
  BBPSF        3B  /KEY/                      REAL                  RADMIN      13B  /KEY1/
  BBRMIN       4B  /KEY/                      REAL                  RADSF       14B  /KEY1/
  BBRSF        5B  /KEY/                      REAL                  TCASMIN      7B  /KEY1/
  EDAY         3B  /DATE/                     REAL                  TCASSF      10B  /KEY1/
  EMNTH        2B  /DATE/                     REAL                  TCHMIN       2B  /KEY1/
  FDAY         1B  /DATE/                     REAL                  TCHSF        3B  /KEY1/
  FHR          1B  /TIM/                      REAL                  TCKMIN       0B  /KEY1/
  FHRE         3B  /TIM/                      REAL                  TCKSF        1B  /KEY1/
  FMIN         2B  /TIM/                      REAL                  TSOLMIN      4B  /KEY1/
  FMINE        4B  /TIM/                      REAL                  TSOLSF       5B  /KEY1/
  FMNTH        0B  /DATE/                     REAL                  VPSMIN      11B  /KEY1/
  FYEAR        0B  /TIM/                      REAL                  VPSSF       12B  /KEY1/
  ISTRIN       0B  /TOP/                      CHAR*18               X            3   DUMMY-ARG
  NUM          6B  /KEY1/                     INTEGER               XSCALE       0B  /SCALE/
  PREMIN      15B  /KEY1/                     REAL                  Y1           1   DUMMY-ARG
  PRESF       16B  /KEY1/                     REAL                  Y16          2   DUMMY-ARG
```

```
SUBROUTINE SUB8       74/860  OPT=1,ROUND= A/ S/ M/-D,-DS      FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.


                    SUBROUTINE SUB8(DDAY,DPRT,DBBP,DBBR,DCKT,DISOL,
                   *   DCHBR,DCASE,DRAD,DVPS,DPRESS,M,IDAYS,IDAYE)
              C
              C    *THIS SUBROUTINE PLOTS THE DAILY AVERAGES OF ALL THE
              C    *PARAMETERS.
              C
                    COMMON/TOP/ANS,ISTRIN
                    COMMON/SCALE/XSCALE
                    COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
                    COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
                    COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
                    COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
                   *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
                    DIMENSION DDAY(M+2),DPRT(M+2),DBBP(M+2),DBBR(M+2)
                    DIMENSION DCKT(M+2),DISOL(M+2),DCHBR(M+2),DCASE(M+2),
                   *DRAD(M+2),DVPS(M+2),DPRESS(M+2)
                    CHARACTER ANS*1,ISTRIN*18
                    CHARACTER NSTRIN*14
              C     DATA ISTRIN /'ELAPSED TIME(DAYS)'/
                    DATA NSTRIN/'DAILY  AVERAGE'/
                    ISTRIN='ELAPSED TIME(DAYS)'
                    DAYS=FLOAT(IDAYS)
                    DAYE=FLOAT(IDAYE)
                    PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
                    PRINT *,'PRT MIN ....980K'
                    PRINT *,'PRT SF ....10 '
                    PRINT *,'BBP MIN ...2 WATTS'
                    PRINT *,'BBP SF ...2'
                    PRINT *,'BBR MIN ...2.5 OHMS'
                    PRINT *,'BBR SF ...(.5)'
              5     PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
                    READ (*,'(A1)')ANS
                    IF(ANS.EQ.'Y')THEN
  1                   PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
  1                   READ *,PRTMIN,PRTSF
  1                 PRINT *,'INPUT THE NEW BBP MIN AND SF'
  1                 READ *,BBPMIN,BBPSF
  1                 PRINT *,'INPUT THE NEW BBR MIN AND SF'
```

```
1                        READ *,BBRMIN,BBRSF
1                          ELSEIF(ANS.EQ.'N')THEN
1                             GO TO 100
1                             ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
1                               PRINT *,'WRONG INPUT'
1                               GO TO 5
1                        ENDIF
                   100   DPRT(M+1)=PRTMIN
                         DPRT(M+2)=PRTSF
                         DBBP(M+1)=BBPMIN
                         DBBP(M+2)=BBPSF
                         DBBR(M+1)=BBRMIN
                         DBBR(M+2)=BBRSF
                         CALL LEROY
                         CALL CALPLT(2.,1.,-3)
                         DDAY(M+1)=DAYS-1.
                         DDAY(M+2)=6.
                         CALL AXES(0.,0.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,ISTRIN,
                        *.14,-18)
                         WRITE(6,16)ISTRIN
                   16    FORMAT(1X,A18)
                         CALL AXES(0.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,'PRT',
                        *.14,3)
                         CALL AXES(0.,5.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,' ',0.0,1)
                         CALL AXES(9.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,' ',0.0,-1)
                         CALL NEWPEN (1)
                         CALL LINPLT(DDAY,DPRT,M,1,0,0,1,1)
                         CALL NEWPEN (2)
                         CALL CALPLT(-1.,0.,-3)
                         CALL AXES(0.,0.,90.,5.,DBBP(M+1),DBBP(M+2),1.,10.,'BBP',
                        *.14,3)
                         CALL CALPLT(1.,0.,-3)
                         CALL LINPLT(DDAY,DBBP,M,1,0,0,1,2)
                         CALL NEWPEN (3)
                         CALL CALPLT(10.,0.,-3)
                         CALL AXES(0.,0.,90.,5.,DBBR(M+1),DBBR(M+2),1.,10.,'BBR',
                        *.14,3)
                         CALL CALPLT(-10.,0.,-3)
                         CALL LINPLT(DDAY,DBBR,M,1,0,0,1,3)
                         CALL NEWPEN (1)
                         CALL HEADR
                         CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
```

```
      CALL NFRAME
      PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
      PRINT *,'PRT MIN ....980K'
      PRINT *,'PRT SF ....10 '
      PRINT *,'TCK MIN ...20'
      PRINT *,'TCK SF ...5.'
      PRINT *,'TCH MIN ...15'
      PRINT *,'TCH SF ...(5.)'
15    PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
      READ (*,'(A1)')ANS
      IF(ANS.EQ.'Y')THEN
        PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
        READ *,PRTMIN,PRTSF
      PRINT *,'INPUT THE NEW TCK MIN AND SF'
      READ *,TCKMIN,TCKSF
      PRINT *,'INPUT THE NEW TCH MIN AND SF'
      READ *,TCHMIN,TCHSF
        ELSEIF(ANS.EQ.'N')THEN
           GO TO 200
           ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
             PRINT *,'WRONG INPUT'
             GO TO 15
      ENDIF
200   DCKT(M+1)=TCKMIN
      DCKT(M+2)=TCKSF
      DCHBR(M+1)=TCHMIN
      DCHBR(M+2)=TCHSF
      DPRT(M+1)=PRTMIN
      DPRT(M+2)=PRTSF
      DDAY(M+1)=DAYS-1.
      DDAY(M+2)=6.
      CALL LEROY
      CALL CALPLT(2.,1.,-3)
      DDAY(M+1)=DAYS-1.
      DDAY(M+2)=6.
      WRITE(6,16)ISTRIN
      CALL AXES(0.,0.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,
     *'ELAPSED TIME$(DAYS$)',.14,-20)
      WRITE(6,16)ISTRIN
      CALL AXES(0.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,'PRT',
     *.14,3)
      CALL AXES(0.,5.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,' ',0.0,1)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL AXES(9.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,' ',0.0,-1)
      CALL NEWPFN (1)
      CALL LINPLT(DDAY,DPRT,M,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,DCKT(M+1),DCKT(M+2),1.,10.,'CIRCUIT',
     *.14,7)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(DDAY,DCKT,M,1,0,0,1,2)
      CALL NEWPEN (3)
      CALL CALPLT(10.,0.,-3)
      CALL AXES(0.,0.,90.,5.,DCHBR(M+1),DCHBR(M+2),1.,10.,'CHBR',
     *.14,4)
      CALL CALPLT(-10.,0.,-3)
      CALL LINPLT(DDAY,DCHBR,M,1,0,0,1,3)
      CALL NEWPEN (1)
      CALL HEADR
      CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
      CALL NFRAME
      PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
      PRINT *,'PRT MIN ....980K'
      PRINT *,'PRT SF ....10 '
      PRINT *,'ISOL MIN ...110'
      PRINT *,'ISOL SF ...5'
      PRINT *,'CASE MIN ...115'
      PRINT *,'CASE SF ...(5.)'
25    PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
      READ (*,'(A1)')ANS
      IF(ANS.EQ.'Y')THEN
        PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
        READ *,PRTMIN,PRTSF
      PRINT *,'INPUT THE NEW ISOL MIN AND SF'
      READ *,TSOLMIN,TSOLSF
      PRINT *,'INPUT THE NEW CASE MIN AND SF'
      READ *,TCASMIN,TCASSF
        ELSEIF(ANS.EQ.'N')THEN
           GO TO 300
           ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
             PRINT *,'WRONG INPUT'
             GO TO 25
      ENDIF
300   DPRT(M+1)=PRTMIN
```

```
      DPRT(M+2)=PRTSF
      DISOL(M+1)=TSOLMIN
      DISOL(M+2)=TSOLSF
      DCASE(M+1)=TCASMIN
      DCASE(M+2)=TCASSF
      CALL LEROY
      CALL CALPLT(2.,1.,-3)
      DDAY(M+1)=DAYS-1.
      DDAY(M+2)=6.
      CALL AXES(0.,0.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,
     *'ELAPSED TIME$(DAYS$)',.14,-20)
      CALL AXES(0.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,'PRT',
     *.14,3)
      CALL AXES(0.,5.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,' ',0.0,1)
      CALL AXES(9.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,' ',0.0,-1)
      CALL NEWPEN (1)
      CALL LINPLT(DDAY,DPRT,M,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,DISOL(M+1),DISOL(M+2),1.,10.,'ISOLATOR',
     *.14,8)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(DDAY,DISOL,M,1,0,0,1,2)
      CALL NEWPEN (3)
      CALL CALPLT(10.,0.,-3)
      CALL AXES(0.,0.,90.,5.,DCASE(M+1),DCASE(M+2),1.,10.,'CASE',
     *.14,4)
      CALL CALPLT(-10.,0.,-3)
      CALL LINPLT(DDAY,DCASE,M,1,0,0,1,3)
      CALL NEWPEN(1)
      CALL HEADR
      CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
      CALL NFRAME
      PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
      PRINT *,'PRT MIN ....980K'
      PRINT *,'PRT SF ....10 '
      PRINT *,'RAD MIN ...0 '
      PRINT *,'RAD SF ...2.'
      PRINT *,'VPS MIN ...0.'
      PRINT *,'VPS SF ...(2.)'
35    PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
      READ (*,'(A1)')ANS
```

```
              IF(ANS.EQ.'Y')THEN
1               PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
1               READ *,PRTMIN,PRTSF
1             PRINT *,'INPUT THE NEW RAD MIN AND SF'
1             READ *,RADMIN,RADSF
1             PRINT *,'INPUT THE NEW VPS MIN AND SF'
1             READ *,VPSMIN,VPSSF
1               ELSEIF(ANS.EQ.'N')THEN
1                  GO TO 400
1                  ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
1                    PRINT *,'WRONG INPUT'
1                    GO TO 35
1             ENDIF
        400   DPRT(M+1)=PRTMIN
              DPRT(M+2)=PRTSF
              DRAD(M+1)=RADMIN
              DRAD(M+2)=RADSF
              DVPS(M+1)=VPSMIN
              DVPS(M+2)=VPSSF
              CALL LERDY
              CALL CALPLT(2.,1.,-3)
              DDAY(M+1)=DAYS-1.
              DDAY(M+2)=6.
              CALL AXES(0.,0.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,
             *'ELAPSED TIME$(DAYS$)',.14,-20)
              CALL AXES(0.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,'PRT',
             *.14,3)
              CALL AXES(0.,5.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,' ',0.0,1)
              CALL AXES(9.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,' ',0.0,-1)
              CALL NEWPEN (1)
              CALL LINPLT(DDAY,DPRT,M,1,0,0,1,1)
              CALL NEWPEN (2)
              CALL CALPLT(-1.,0.,-3)
              CALL AXES(0.,0.,90.,5.,DRAD(M+1),DRAD(M+2),1.,10.,'RADIOMETRIC',
             *.14,11)
              CALL CALPLT(1.,0.,-3)
              CALL LINPLT(DDAY,DRAD,M,1,0,0,1,2)
              CALL NEWPEN (3)
              CALL CALPLT(10.,0.,-3)
              CALL AXES(0.,0.,90.,5.,DVPS(M+1),DVPS(M+2),1.,10.,'VPS',
             *.14,3)
              CALL CALPLT(-10.,0.,-3)
```

```
          CALL LINPLT(DDAY,DVPS,M,1,0,0,1,3)
          CALL NEWPEN(1)
          CALL HEADR
          CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
          CALL NFRAME
          PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
          PRINT *,'PRT MIN ....980K'
          PRINT *,'PRT SF ....10 '
          PRINT *,'PRE MIN ...-.01'
          PRINT *,'PRE SF ...4.'
 45       PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
          READ (*,'(A1)')ANS
          IF(ANS.EQ.'Y')THEN
            PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
            READ *,PRTMIN,PRTSF
          PRINT *,'INPUT THE NEW PRE MIN AND SF'
          READ *,PREMIN,PRESF
            ELSEIF(ANS.EQ.'N')THEN
               GO TO 500
               ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
                 PRINT *,'WRONG INPUT'
                 GO TO 45
          ENDIF
 500      DPRT(M+1)=PRTMIN
          DPRT(M+2)=PRTSF
          DPRESS(M+1)=PREMIN
          DPRESS(M+2)=PRESF
          CALL LEROY
          CALL CALPLT(2.,1.,-3)
          DDAY(M+1)=DAYS-1.
          DDAY(M+2)=6.
          CALL AXES(0.,0.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,
         *'ELAPSED TIME$(DAYS$)',.14,-20)
          CALL AXES(0.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,'PRT',
         *.14,3)
          CALL AXES(0.,5.,0.,9.,DDAY(M+1),DDAY(M+2),1.,6.,' ',0.0,1)
          CALL AXES(9.,0.,90.,5.,DPRT(M+1),DPRT(M+2),1.,10.,' ',0.0,-1)
          CALL NEWPEN (1)
          CALL LINPLT(DDAY,DPRT,M,1,0,0,1,1)
          CALL NEWPEN (2)
          CALL CALPLT(-1.,0.,-3)
          CALL AXES(0.,0.,90.,5.,DPRESS(M+1),DPRESS(M+2),1.,10.,
                 *'PRESSURE',.14,8)
                  CALL CALPLT(1.,0.,-3)
                  CALL LINPLT(DDAY,DPRESS,M,1,0,0,1,2)
                  CALL NEWPEN (1)
                  CALL HEADR
                  CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
                  CALL NFRAME
                  RETURN
                  END
```

```
--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES

  ANS          0B  /TOP/                        CHAR*1                FMNTH        0B  /DATE/
  BBPMIN       2B  /KEY/                        REAL                  FYEAR        0B  /TIM/
  BBPSF        3B  /KEY/                        REAL                  IDAYE       14   DUMMY-ARG
  BBRMIN       4B  /KEY/                        REAL                  IDAYS       13   DUMMY-ARG
  BBRSF        5B  /KEY/                        REAL                  ISTRIN       0B  /TOP/
  DAYE     2513B                *S*             REAL                  M           12   DUMMY-ARG
  DAYS     2512B                                REAL                  NSTRIN   2510B
  DBBP         3   DUMMY-ARG                    REAL       ADJ-ARY    NUM          6B  /KEY1/
  DBBR         4   DUMMY-ARG                    REAL       ADJ-ARY    PREMIN      15B  /KEY1/
  DCASE        8   DUMMY-ARG                    REAL       ADJ-ARY    PRESF       16B  /KEY1/
  DCHBR        7   DUMMY-ARG                    REAL       ADJ-ARY    PRTMIN       0B  /KEY/
  DCKT         5   DUMMY-ARG                    REAL       ADJ-ARY    PRTSF        1B  /KEY/
  DDAY         1   DUMMY-ARG                    REAL       ADJ-ARY    RADMIN      13B  /KEY1/
  DISOL        6   DUMMY-ARG                    REAL       ADJ-ARY    RADSF       14B  /KEY1/
  DPRESS      11   DUMMY-ARG                    REAL       ADJ-ARY    TCASMIN      7B  /KEY1/
  DPRT         2   DUMMY-ARG                    REAL       ADJ-ARY    TCASSF      10B  /KEY1/
  DRAD         9   DUMMY-ARG                    REAL       ADJ-ARY    TCHMIN       2B  /KEY1/
  DVPS        10   DUMMY-ARG                    REAL       ADJ-ARY    TCHSF        3B  /KEY1/
  EDAY         3B  /DATE/                       REAL                  TCKMIN       0B  /KEY1/
  EMNTH        2B  /DATE/                       REAL                  TCKSF        1B  /KEY1/
  FDAY         1B  /DATE/                       REAL                  TSOLMIN      4B  /KEY1/
  FHR          1B  /TIM/                        REAL                  TSOLSF       5B  /KEY1/
  FHRE         3B  /TIM/                        REAL                  VPSMIN      11B  /KEY1/
  FMIN         2B  /TIM/                        REAL                  VPSSF       12B  /KEY1/
  FMINE        4B  /TIM/                        REAL                  XSCALE       0B  /SCALE/
```

```
       C


       C
                SUBROUTINE SUB9(WWK,WPRT,WBBP,WBBR,WCKT,WISOL,
       C
       C       *THIS SUBROUTINE PLOTS THE WEEKLY AVERAGES OF ALL
       C       *PARAMETERS
       C
               *   WCHBR,WCASE,WRAD,WVPS,WPRESS,NW)
                COMMON/TOP/ANS,ISTRIN
                COMMON/SCALE/XSCALE
                COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
                COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
                COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
                COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
               *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
                DIMENSION WWK(NW+2),WPRT(NW+2),WBBP(NW+2),WBBR(NW+2)
                DIMENSION WCKT(NW+2),WISOL(NW+2),WCHBR(NW+2),WCASE(NW+2),
               *WRAD(NW+2),WVPS(NW+2),WPRESS(NW+2)
                CHARACTER ISTRIN*19,ANS*1,NSTRIN*14
                DATA NSTRIN/'WEEKLY AVERAGE'/
                ISTRIN='ELAPSED TIME(WEEKS)'
                DAYS=FLOAT(IDAYS)
                DAYE=FLOAT(IDAYE)
                PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
                PRINT *,'PRT MIN ....980K'
                PRINT *,'PRT SF ....10 '
                PRINT *,'BBP MIN ...2 WATTS'
                PRINT *,'BBP SF ...2'
                PRINT *,'BBR MIN ...2.5 OHMS'
                PRINT *,'BBR SF ...(.5)'
        5       PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
                PEAD (*,'(A1)')ANS
                IF(ANS.EQ.'Y')THEN
 1                PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
 1                READ *,PRTMIN,PRTSF
 1              PRINT *,'INPUT THE NEW BBP MIN AND SF'
 1              READ *,BBPMIN,BBPSF
 1              PRINT *,'INPUT THE NEW BBR MIN AND SF'
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  1                READ *,BBRMIN,BBRSF
  1                  ELSEIF(ANS.EQ.'N')THEN
  1                     GO TO 100
  1                     ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
  1                       PRINT *,'WRONG INPUT'
  1                       GO TO 5
  1                ENDIF
          100      WPRT(NW+1)=PRTMIN
                   WPRT(NW+2)=PRTSF
                   WBBP(NW+1)=BBPMIN
                   WBBP(NW+2)=BBPSF
                   WBBR(NW+1)=BBRMIN
                   WBBR(NW+2)=BBRSF
                   CALL LEROY
                   CALL CALPLT(2.,1.,-3)
                   WWK(NW+1)=0.
                   WWK(NW+2)=1.
                   CALL AXES(0.,0.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,ISTRIN,
                  *.14,-19)
                   CALL AXES(0.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,'PRT',
                  *.14,3)
                   CALL AXES(0.,5.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,' ',0.0,1)
                   CALL AXES(9.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,' ',0.0,-1)
                   CALL NEWPEN (1)
                   CALL LINPLT(WWK,WPRT,NW,1,0,0,1,1)
                   CALL NEWPEN (2)
                   CALL CALPLT(-1.,0.,-3)
                   CALL AXES(0.,0.,90.,5.,WBBP(NW+1),WBBP(NW+2),1.,10.,'BBP',
                  *.14,3)
                   CALL CALPLT(1.,0.,-3)
                   CALL LINPLT(WWK,WBBP,NW,1,0,0,1,2)
                   CALL NEWPEN (3)
                   CALL CALPLT(10.,0.,-3)
                   CALL AXES(0.,0.,90.,5.,WBBR(NW+1),WBBR(NW+2),1.,10.,'BBR',
                  *.14,3)
                   CALL CALPLT(-10.,0.,-3)
                   CALL LINPLT(WWK,WBBR,NW,1,0,0,1,3)
                   CALL NEWPEN (1)
                   CALL HEADR
                   CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
                   CALL NFRAME
                   PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
```

```
                PRINT *,'PRT MIN ....980K'
                PRINT *,'PRT SF ....10 '
                PRINT *,'TCK MIN ...20'
                PRINT *,'TCK SF ...5.'
                PRINT *,'TCH MIN ...15'
                PRINT *,'TCH SF ...(5.)'
          15    PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
                READ (*,'(A1)')ANS
                IF(ANS.EQ.'Y')THEN
 1                PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
 1                READ *,PRTMIN,PRTSF
 1              PRINT *,'INPUT THE NEW TCK MIN AND SF'
 1              READ *,TCKMIN,TCKSF
 1              PRINT *,'INPUT THE NEW TCH MIN AND SF'
 1              READ *,TCHMIN,TCHSF
 1                ELSEIF(ANS.EQ.'N')THEN
 1                   GO TO 200
 1                   ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
 1                     PRINT *,'WRONG INPUT'
 1                     GO TO 15
 1              ENDIF
          200   WCKT(NW+1)=TCKMIN
                WCKT(NW+2)=TCKSF
                WCHBR(NW+1)=TCHMIN
                WCHBR(NW+2)=TCHSF
                WPRT(NW+1)=PRTMIN
                WPRT(NW+2)=PRTSF
                WWK(NW+1)=0.
                WWK(NW+2)=1.
                CALL LEROY
                CALL CALPLT(2.,1.,-3)
                WWK(NW+2)=1.
                CALL AXES(0.,0.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,
               *'ELAPSED TIME$(WEEKS$)',.14,-21)
                CALL AXES(0.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,'PRT',
               *.14,3)
                CALL AXES(0.,5.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,' ',0.0,1)
                CALL AXES(9.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,' ',0.0,-1)
                CALL NEWPEN (1)
                CALL LINPLT(WWK,WPRT,NW,1,0,0,1,1)
                CALL NEWPEN (2)
                CALL CALPLT(-1.,0.,-3)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL AXES(0.,0.,90.,5.,WCKT(NW+1),WCKT(NW+2),1.,10.,'CIRCUIT',
     *.14,7)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(WWK,WCKT,NW,1,0,0,1,2)
      CALL NEWPEN (3)
      CALL CALPLT(10.,0.,-3)
      CALL AXES(0.,0.,90.,5.,WCHBR(NW+1),WCHBR(NW+2),1.,10.,'CHAMBER',
     *.14,7)
      CALL CALPLT(-10.,0.,-3)
      CALL LINPLT(WWK,WCHBR,NW,1,0,0,1,3)
      CALL NEWPEN (1)
      CALL HEADR
      CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
      CALL NFRAME
      PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
      PRINT *,'PRT MIN ....980K'
      PRINT *,'PRT SF ....10 '
      PRINT *,'ISOL MIN ...110'
      PRINT *,'ISOL SF ...5'
      PRINT *,'CASE MIN ...115'
      PRINT *,'CASE SF ...(5.)'
   25 PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
      READ (*,'(A1)')ANS
      IF(ANS.EQ.'Y')THEN
        PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
        READ *,PRTMIN,PRTSF
      PRINT *,'INPUT THE NEW ISOL MIN AND SF'
      READ *,TSOLMIN,TSOLSF
      PRINT *,'INPUT THE NEW CASE MIN AND SF'
      READ *,TCASMIN,TCASSF
        ELSEIF(ANS.EQ.'N')THEN
           GO TO 300
           ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
             PRINT *,'WRONG INPUT'
             GO TO 25
      ENDIF
  300 WPRT(NW+1)=PRTMIN
      WPRT(NW+2)=PRTSF
      WISOL(NW+1)=TSOLMIN
      WISOL(NW+2)=TSOLSF
      WCASE(NW+1)=TCASMIN
      WCASE(NW+2)=TCASSF
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                CALL LEROY
                CALL CALPLT(2.,1.,-3)
                WWK(NW+1)=0.
                WWK(NW+2)=1.
                CALL AXES(0.,0.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,
               *'ELAPSED TIME$(DAYS$)',.14,-21)
                CALL AXES(0.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,'PRT',
               *.14,3)
                CALL AXES(0.,5.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,' ',0.0,1)
                CALL AXES(9.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,' ',0.0,-1)
                CALL NEWPEN (1)
                CALL LINPLT(WWK,WPRT,NW,1,0,0,1,1)
                CALL NEWPEN (2)
                CALL CALPLT(-1.,0.,-3)
                CALL AXES(0.,0.,90.,5.,WISOL(NW+1),WISOL(NW+2),1.,10.,'ISOLATOR',
               *.14,8)
                CALL CALPLT(1.,0.,-3)
                CALL LINPLT(WWK,WISOL,NW,1,0,0,1,2)
                CALL NEWPEN (3)
                CALL CALPLT(10.,0.,-3)
                CALL AXES(0.,0.,90.,5.,WCASE(NW+1),WCASE(NW+2),1.,10.,'CASE',
               *.14,4)
                CALL CALPLT(-10.,0.,-3)
                CALL LINPLT(WWK,WCASE,NW,1,0,0,1,3)
                CALL NEWPEN(1)
                CALL HEADR
                CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
                CALL NFRAME
                PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
                PRINT *,'PRT MIN ....980K'
                PRINT *,'PRT SF ....10 '
                PRINT *,'RAD MIN ...0 '
                PRINT *,'RAD SF ...2.'
                PRINT *,'VPS MIN ...0.'
                PRINT *,'VPS SF ...(2.)'
          35    PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
                READ (*,'(A1)')ANS
                IF(ANS.EQ.'Y')THEN
1                 PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
1                 READ *,PRTMIN,PRTSF
1               PRINT *,'INPUT THE NEW RAD MIN AND SF'
1               READ *,RADMIN,RADSF
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PRINT *,'INPUT THE NEW VPS MIN AND SF'
      READ *,VPSMIN,VPSSF
        ELSEIF(ANS.EQ.'N')THEN
           GO TO 400
           ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
             PRINT *,'WRONG INPUT'
             GO TO 35
      ENDIF
 400  WPRT(NW+1)=PRTMIN
      WPRT(NW+2)=PRTSF
      WRAD(NW+1)=RADMIN
      WRAD(NW+2)=RADSF
      WVPS(NW+1)=VPSMIN
      WVPS(NW+2)=VPSSF
      CALL LEROY
      CALL CALPLT(2.,1.,-3)
      WWK(NW+1)=0.
      WWK(NW+2)=2.
      CALL AXES(0.,0.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,
     *'ELAPSED TIME$(WEEKS$)',.14,-21)
      CALL AXES(0.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,'PRT',
     *.14,3)
      CALL AXES(0.,5.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,' ',0.0,1)
      CALL AXES(9.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,' ',0.0,-1)
      CALL NEWPEN (1)
      CALL LINPLT(WWK,WPRT,NW,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,WRAD(NW+1),WRAD(NW+2),1.,10.,'RADIOMETRIC',
     *.14,11)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(WWK,WRAD,NW,1,0,0,1,2)
      CALL NEWPEN (3)
      CALL CALPLT(10.,0.,-3)
      CALL AXES(0.,0.,90.,5.,WVPS(NW+1),WVPS(NW+2),1.,10.,'VPS',
     *.14,3)
      CALL CALPLT(-10.,0.,-3)
      CALL LINPLT(WWK,WVPS,NW,1,0,0,1,3)
      CALL NEWPEN(1)
      CALL HEADR
      CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
      CALL NFRAME
```

C-2

```
      PRINT *,'THE DEFAULT GRAPHING LIMITS ARE AS FOLLOWS : '
      PRINT *,'PRT MIN ....980K'
      PRINT *,'PRT SF ....10 '
      PRINT *,'PRE MIN ...-.01'
      PRINT *,'PRE SF ...4.'
45    PRINT *,'DO YOU WANT TO CHANGE THE DEFAULT MIN AND SF?(Y/N)'
      READ (*,'(A1)')ANS
      IF(ANS.EQ.'Y')THEN
        PRINT * ,'INPUT THE NEW PRT MIN ANS SF'
        READ *,PRTMIN,PRTSF
      PRINT *,'INPUT THE NEW PRE MIN AND SF'
      READ *,PREMIN,PRESF
        ELSEIF(ANS.EQ.'N')THEN
           GO TO 500
           ELSEIF(ANS.NE.'Y' .OR. ANS .NE. 'N')THEN
             PRINT *,'WRONG INPUT'
             GO TO 45
      ENDIF
500   WPRT(NW+1)=PRTMIN
      WPRT(NW+2)=PRTSF
      WPRESS(NW+1)=PREMIN
      WPRESS(NW+2)=PRESF
      CALL LEROY
      CALL CALPLT(2.,1.,-3)
      WWK(NW+1)=0.
      WWK(NW+2)=1.
      CALL AXES(0.,0.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,
     *'ELAPSED TIMES(WEEKS$)',.14,-21)
      CALL AXES(0.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,'PRT',
     *.14,3)
      CALL AXES(0.,5.,0.,9.,WWK(NW+1),WWK(NW+2),1.,6.,' ',0.0,1)
      CALL AXES(9.,0.,90.,5.,WPRT(NW+1),WPRT(NW+2),1.,10.,' ',0.0,-1)
      CALL NEWPEN (1)
      CALL LINPLT(WWK,WPRT,NW,1,0,0,1,1)
      CALL NEWPEN (2)
      CALL CALPLT(-1.,0.,-3)
      CALL AXES(0.,0.,90.,5.,WPRESS(NW+1),WPRESS(NW+2),1.,10.,
     *'PRESSURE',.14,8)
      CALL CALPLT(1.,0.,-3)
      CALL LINPLT(WWK,WPRESS,NW,1,0,0,1,2)
      CALL NEWPEN (1)
      CALL HEADR
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                    CALL CHARACT(2.5,7.75,.10,NSTRIN,0,14,.2)
                    CALL NFRAME
                    RETURN
                    END

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES

  ANS          0B  /TOP/                      CHAR*1                PRTSF          1B  /KEY/
  BBPMIN       2B  /KEY/                      REAL                  RADMIN        13B  /KEY1/
  BBPSF        3B  /KEY/                      REAL                  RADSF         14B  /KEY1/
  BBRMIN       4B  /KEY/                      REAL                  TCASMIN        7B  /KEY1/
  BBRSF        5B  /KEY/                      REAL                  TCASSF        10B  /KEY1/
  DAYE     2510B              *S*             REAL                  TCHMIN         2B  /KEY1/
  DAYS     2506B              *S*             REAL                  TCHSF          3B  /KEY1/
  EDAY         3B  /DATE/                     REAL                  TCKMIN         0B  /KEY1/
  EMNTH        2B  /DATE/                     REAL                  TCKSF          1B  /KEY1/
  FDAY         1B  /DATE/                     REAL                  TSOLMIN        4B  /KEY1/
  FHR          1B  /TIM/                      REAL                  TSOLSF         5B  /KEY1/
  FHRE         3B  /TIM/                      REAL                  VPSMIN        11B  /KEY1/
  FMIN         2B  /TIM/                      REAL                  VPSSF         12B  /KEY1/
  FMINE        4B  /TIM/                      REAL                  WBBP           3   DUMMY-ARG
  FMNTH        0B  /DATE/                     REAL                  WBBR           4   DUMMY-ARG
  FYEAR        0B  /TIM/                      REAL                  WCASE          8   DUMMY-ARG
  IDAYE    2507B              UND/*S*         INTEGER               WCHBR          7   DUMMY-ARG
  IDAYS    2505B              UND/*S*         INTEGER               WCKT           5   DUMMY-ARG
  ISTRIN       0B  /TOP/                      CHAR*19               WISOL          6   DUMMY-ARG
  NSTRIN   2503B                              CHAR*14               WPRESS        11   DUMMY-ARG
  NUM          6B  /KEY1/                     INTEGER               WPRT           2   DUMMY-ARG
  NW          12   DUMMY-ARG                  INTEGER               WRAD           9   DUMMY-ARG
  PREMIN      15B  /KEY1/                     REAL                  WVPS          10   DUMMY-ARG
  PRESF       16B  /KEY1/                     REAL                  WWK            1   DUMMY-ARG
  PRTMIN       0B  /KEY/                      REAL                  XSCALE         0B  /SCALE/
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
SUBROUTINE HEADR       74/860  OPT=1,ROUND= A/ S/ M/-D,-DS      FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.

C
      SUBROUTINE HEADR
      COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
      COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
      COMMON/TOP/ANS,ISTRIN
      CHARACTER ANS*1,ISTRIN*18
      CHARACTER LSTRIN*28,ITIT1*5,ITIT2*6,ITIT3*4,ISTR5*5
      DATA LSTRIN /'HALOE BLACKBODY PERFORMANCE'/
      DATA ITIT1/'YEAR:'/
      DATA ITIT2/'MONTH:'/
      DATA ITIT3/'DAY:'/
      DATA ISTR5/'TIME:'/
C     DATA ISTR4/'START TIME        :'/
      CALL CHARST8
      CALL CHARACT(2.5,8.0,.15,LSTRIN,0,28,.2)
      CALL CHARACT(2.5,7.5,.10,ITIT1,0,5,.2)
      CALL  CHARACT(2.5,7.0,.10,ITIT2,0,6,.2)
      CALL CHARACT(2.5,6.5,.10,ITIT3,0,4,.2)
      CALL CHARACT(2.5,6.0,.10,ISTR5,0,5,.2)
      CALL NUMBER(4.0,7.5,.10,FYEAR,0.,-1)
      CALL NUMBER(4.0,7.0,.10,FMNTH,0.,-1)
      CALL NUMBER(4.0,6.5,.10,FDAY,0.,-1)
      CALL NUMBER(6.0,7.5,.10,FYEAR,0.,-1)
      CALL NUMBER(6.0,7.0,.10,EMNTH,0.,-1)
      CALL NUMBER(6.0,6.5,.10,EDAY,0.,-1)
      CALL NUMBER(4.0,6.0,.10,FHR,0.,-1)
      CALL NUMBER(4.8,6.0,.10,FMIN,0.,-1)
      CALL NUMBER(6.0,6.0,.10,FHRE,0.,-1)
      CALL NUMBER(6.8,6.0,.10,FMINE,0.,-1)
      CALL CHARST1
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE SUB9       74/860  OPT=1,ROUND= A/ S/ M/-D,-DS     FTN 5.1+642        87/04/30. 09.48.

 --PROCEDURES--(LO=A)
  -NAME------TYPE---------ARGS------CLASS-----      -NAME------TYPE---------ARGS------CLASS-----

   AXES                     11       SUBROUTINE       LEROY                    0       SUBROUTINE
   CALPLT                    3       SUBROUTINE       LINPLT                   8       SUBROUTINE
   CHARACT                   7       SUBROUTINE       NEWPEN                   1       SUBROUTINE
   FLOAT      REAL           1       INTRINSIC        NFRAME                   0       SUBROUTINE
   HEADR                     0       SUBROUTINE

 --STATEMENT LABELS--(LO=A)
  -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----PROPERTIES----DEF       -LABEL-ADDRESS-----

       5     30B                     33          45    1136B                     256          300     537B
      15    252B                     89         100      71B                      48          400     761B
      25    476B                    146         200     313B                     104          500    1173B
      35    720B                    202

 --ENTRY POINTS--(LO=A)
  -NAME---ADDRESS--ARGS---

   SUB9         3B    12

 --STATISTICS--

   PROGRAM-UNIT LENGTH           2511B =  1353
   CM LABELLED COMMON LENGTH       41B =  33
   CM STORAGE USED             62000B =  25600
   COMPILE TIME                5.764 SECONDS
```

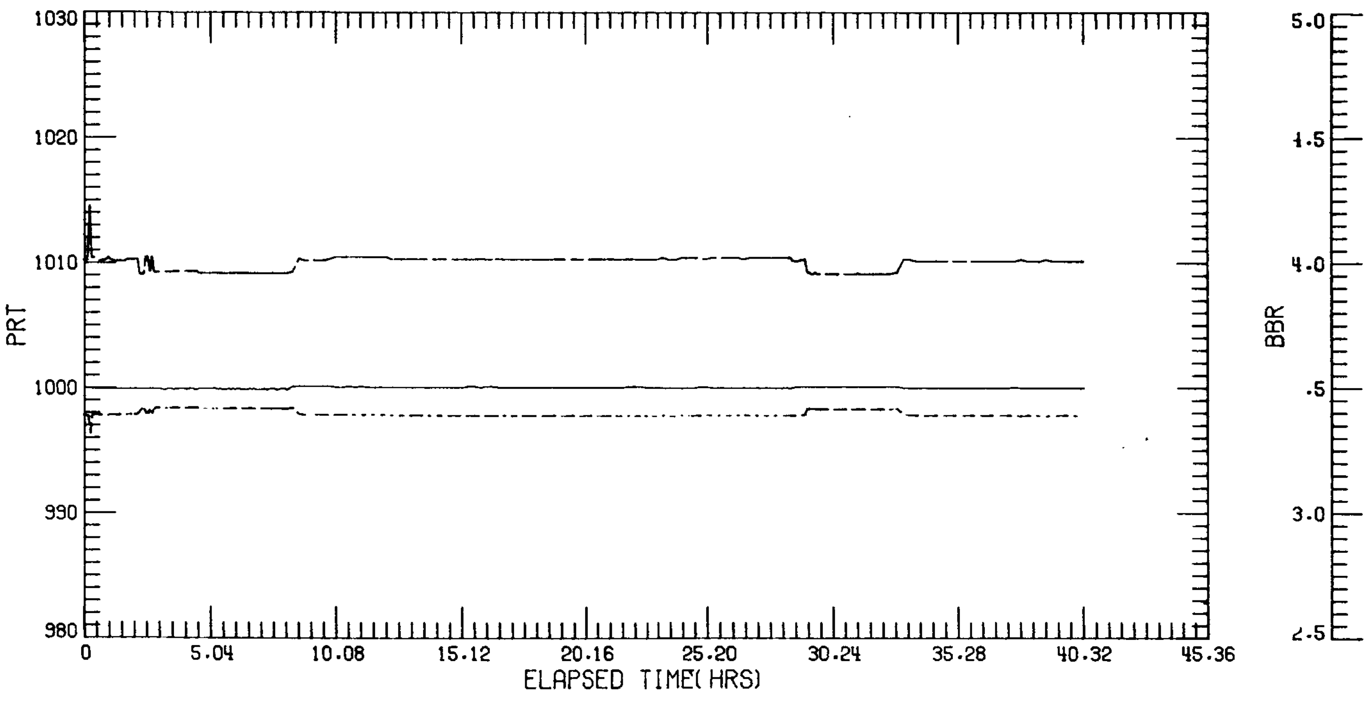
ORIGINAL PAGE IS
OF POOR QUALITY

```
    BLOCKDATA COM          74/860  OPT=1,ROUND= A/ S/ M/-D,-DS      FTN 5.1+642          87/04/30. 09.48.
DO=-LONG/-OT,ARG= COMMON/-FIXED,CS= USER/-FIXED,DB=-TB/-SB/-SL/-ER/-ID/-PMD/-ST,-AL,PL=5000
FTN5,I=HPLOT,L=LF.


                  BLOCK DATA COM
      C           COMMON/JTB/NFR,JREQ,IBAUD,HDR,IJO,TFAC,IJTB(4)
                  COMMON/SCALE/XSCALE
                  COMMON/TOP/ANS,ISTRIN
                  COMMON/TIT/IDEN1,IDEN2,IDEN3,IDEN4,IDEN5,IL1,IL2,IL3,IL4,IL5
                  COMMON/TIM/FYEAR,FHR,FMIN,FHRE,FMINE
                  COMMON/DATE/FMNTH,FDAY,EMNTH,EDAY
                  COMMON/KEY/ PRTMIN,PRTSF,BBPMIN,BBPSF,BBRMIN,BBRSF
                  COMMON/KEY1/ TCKMIN,TCKSF,TCHMIN,TCHSF,TSOLMIN,TSOLSF,NUM,
                 *TCASMIN,TCASSF,VPSMIN,VPSSF,RADMIN,RADSF,PREMIN,PRESF
                  CHARACTER ANS*1,ISTRIN*18
      C           DATA JREQ /2/
                  DATA PRTMIN,PRTSF/980,10./
                  DATA BBPMIN,BBPSF/2,2./
                  DATA BBRMIN,BBRSF/2.5,.5/
                  DATA TCKMIN,TCKSF/20,5./
                  DATA TCHMIN,TCHSF/15,5./
                  DATA TSOLMIN,TSOLSF/110,5./
                  DATA TCASMIN,TCASSF/115,5./
                  DATA VPSMIN,VPSSF/0,2./
                  DATA RADMIN,RADSF/0,2./
                  DATA PREMIN,PRESF/-.01,4./
                  END

--VARIABLE MAP--(LO=A)
 -NAME---ADDRESS --BLOCK-----PROPERTIES-------TYPE---------SIZE     -NAME---ADDRESS --BLOCK-----PROPERTIES-

  ANS          0B  /TOP/                      CHAR*1                  FMINE        4B  /TIM/
  BBPMIN       2B  /KEY/                      REAL                    FMNTH        0B  /DATE/
  BBPSF        3B  /KEY/                      REAL                    FYEAR        0B  /TIM/
  BBRMIN       4B  /KEY/                      REAL                    IDEN1        0B  /TIT/
  BBRSF        5B  /KEY/                      REAL                    IDEN2        1B  /TIT/
  EDAY         3B  /DATE/                     REAL                    IDEN3        2B  /TIT/
  EMNTH        2B  /DATE/                     REAL                    IDEN4        3B  /TIT/
  FDAY         1B  /DATE/                     REAL                    IDEN5        4B  /TIT/
  FHR          1B  /TIM/                      REAL                    IL1          5B  /TIT/
  FHRE         3B  /TIM/                      REAL                    IL2          6B  /TIT/
  FMIN         2B  /TIM/                      REAL                    IL3          7B  /TIT/
```

ORIGINAL PAGE IS
OF POOR QUALITY

HALOE BLACKBODY PERFORMANCE

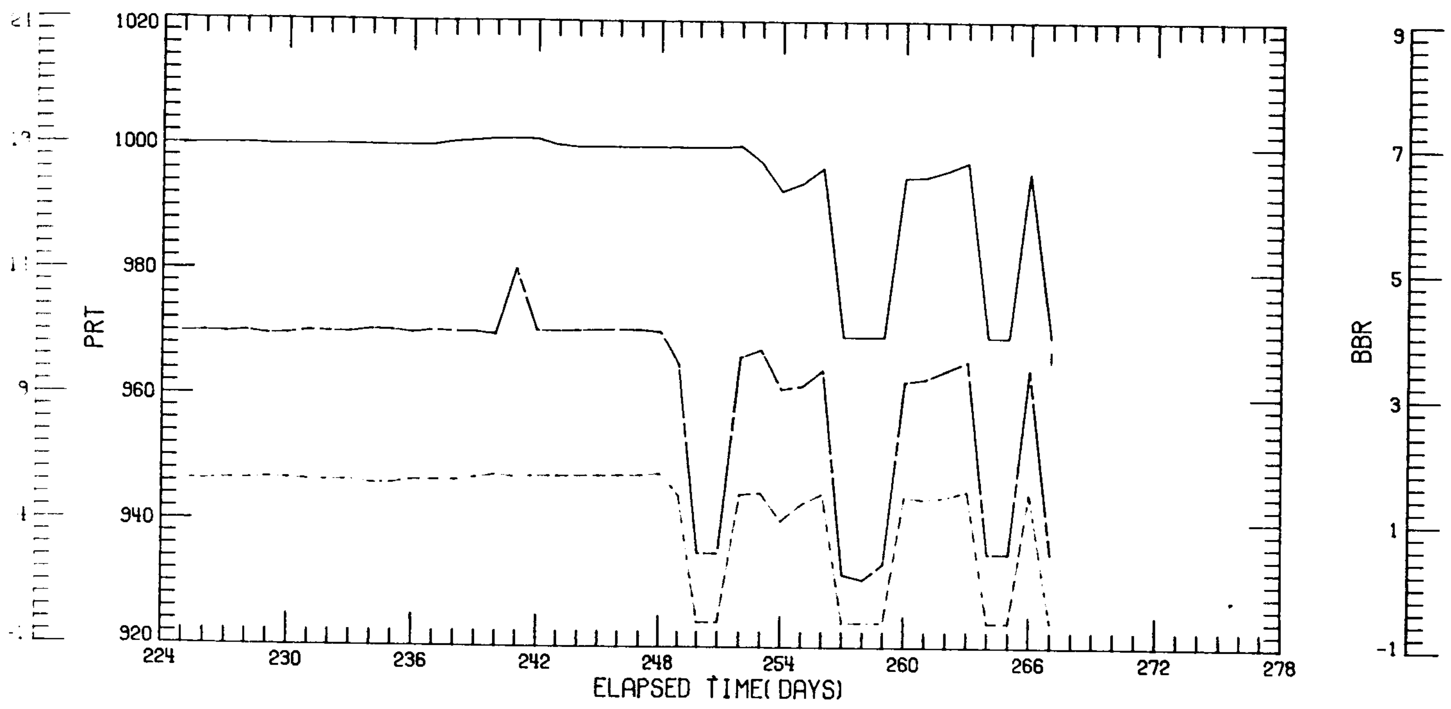
YEAR:	1985		1985	
MONTH:	8		8	
DAY:	13		14	
TIME:	7	36	23	55



HALOE BLACKBODY PERFORMANCE

DAILY AVERAGE

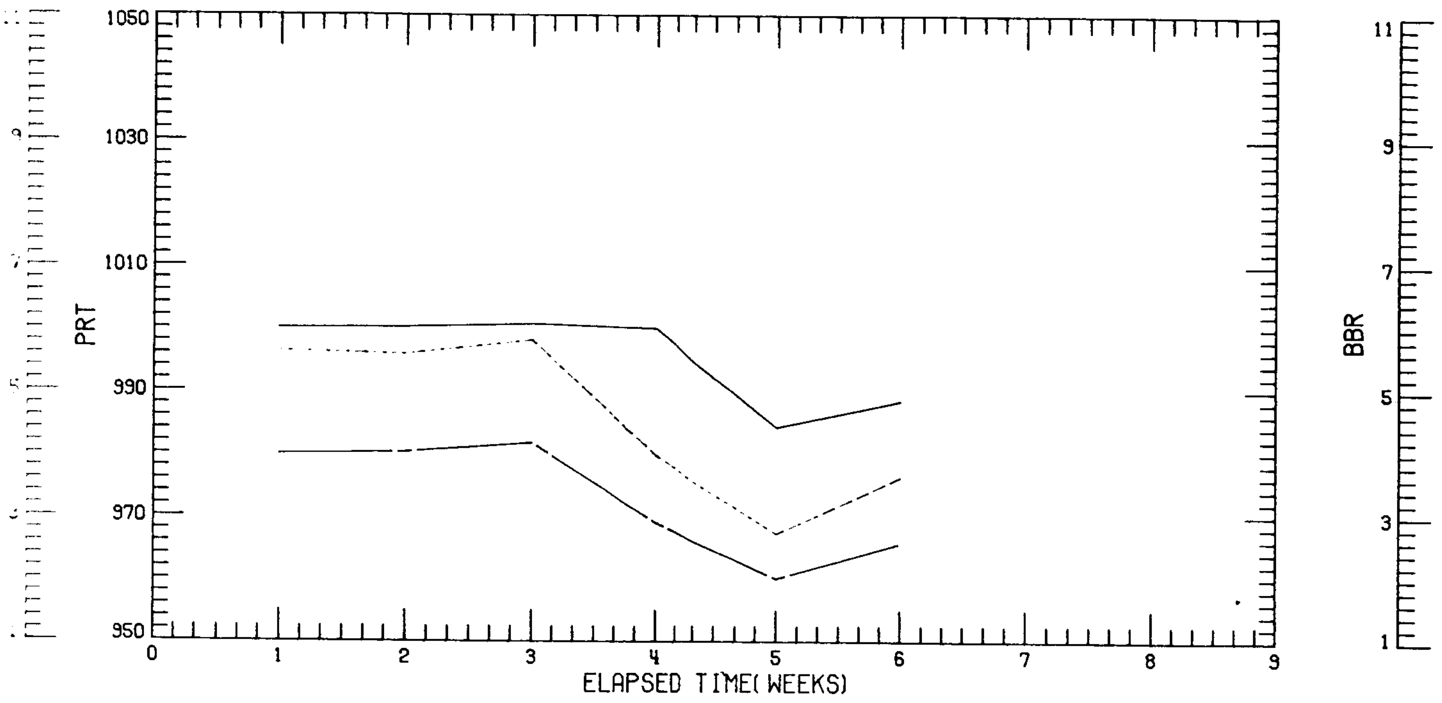
YEAR:	1985		1985	
MONTH:	8		9	
DAY:	13		24	
TIME:	7	36	4	45



HALOE BLACKBODY PERFORMANCE

WEEKLY AVERAGE

YEAR:	1985		1985	
MONTH:	8		9	
DAY:	13		24	
TIME:	7	36	4	45



APPENDIX C - SPECRES

Program Name:	SPECRES.PAS
Function:	SPECRES is designed to acquire data from the HALOE GCETS (Gas Correlation Electronic Test Set) during the Spectral Response Test.
Description:	SPECRES is written in Turbo Pascal on and for an IBM-XT or compatible. The program uses an RS232 line to communicate with the CD2A Compudrive which drives the spectrometer during the spectral response test. SPECRES also uses a Lab Master card to acquire data from the GCETS which is in turn connected to channels of interest in the HALOE instrument.
Use:	SPECRES is invoked on the IBM-XT by typing SPECRES. The program prompts the user for the run-time parameters and file names as needed. Data is saved to disk file for plotting and tabulating after each spectral run is completed.

Listing of: SPECRES.PAS

```
        PROGRAM SPECRES ;

  {
        Haloe Spectral Response data acquisition program.  This program
        communicates with the CD2A Compudrive to determine the wavenumber
        setting of the spectrometer. Each time the wavenumber changes,
        Specres will acquire a number of data points for all the selected
        channels. The data is recorded on disk to be plotted and analyzed
        immediately following a spectral response run.

  }


  {$U-}
  { RS232 INPUT/OUTPUT ROUTINES }
  TYPE REGPACK = RECORD
        AX,BX,CX,DX,BP,DI,SI,DS,ES,FLAGS:INTEGER ;
        END;

  CONST
      SIX: BYTE = 6 ;
      LF : BYTE = 10 ;

  VAR
      INSTRING : STRING[80] ;
      RECPACK : REGPACK ;
      AH,AL: BYTE ;
      OLDSER,SER :INTEGER ;
      Baud,StopBits,DataBits,PAR: Integer;
      Message: String[80];
      PORT1 : INTEGER ;
      INCHAR,OUTCHAR : BYTE ;
      INPCHAR: CHAR ABSOLUTE INCHAR ;
      OUTPCHAR: CHAR ABSOLUTE OUTCHAR ;
      ONLINE : BOOLEAN ;
      printer : boolean ;
    type
      String19=String[19];
    Type
      __RegisterSet=Record case Integer of
                      1: (AX,BX,CX,DX,BP,DI,SE,DS,ES,Flags: Integer);
                      2: (AL,AH,BL,BH,CL,CH,DL,DH: Byte);
                    end;
      __ParityType=(None,Even,Odd);

    var
      __Regs: __RegisterSet;
      InError,OutError: Array [1..2] of Byte;
```

```
{      SPECTRAL RESPONSE DATA ACQUISITION PROGRAM  }
       TYPE
           Filename = String[12] ;
           Name = String[10] ;
           Names = Array[0..4] of Name ;
           Samples = Array[0..4] of real ;
           descript = string[80] ;

       LABEL STOP ;

       CONST
            STARTLOC : INTEGER = $710 ;
            factor : array[0..2] of real = (1.0,10.0,100.0) ;
            rgain: array[0..2] of integer = (1,10,100) ;
            MAXCHANnum : INTEGER = 5 ;
      VAR

           PROMPT : DESCRIPT ;
           MONTH,DAY,HR,MIN,SEC:INTEGER ;
           bell:char;
           NCHAN : INTEGER ;
           ITER : INTEGER ;
           NITER : INTEGER ;
           NPTS : INTEGER ;
           COUNTS : SAMPLES ;
           i,j,k : integer ;
           IT: INTEGER ;
           CTRLBYTE : BYTE ;
           STATBYTE : BYTE ;
           Inch : integer ;
           INPCH : ARRAY[0..15] OF INTEGER ;
           IND : INTEGER ;
           HIGH : BYTE ;
           LOW : BYTE ;
           val : real ;
           ref : real ;
           ICHAN : INTEGER ;
           CHAN : ARRAY[0..5,0..1000] OF REAL ;
           gain : array[0..15] of byte ;
           igain : byte ;
           sum,sumx2,mean,minx,maxx,std,nopts : samples ;
           tsum,tsumx2,tmean,tminx,tmaxx,tstd,tnopts:samples;
           NAM : NAMES;
           F1 : TEXT ;
           FNAME : FILENAME ;
           PLOTS : TEXT ;
           PNAME : FILENAME ;
           IOerror : integer ;
           answer : string[1] ;
           WAVEL : REAL ;
           WAVELENGTH : real ;      { USED FOR WAVENUMBER }
```

```
          DELTA : REAL ;          { STEP SIZE }
          DWELL , STEPS: INTEGER ;
          denom : real ;
          RADICAL : REAL ;
          descrip : descript ;
          RSINT : ARRAY[0..1] OF INTEGER ABSOLUTE $0000:$0030 ;
          OLDINT : ARRAY[0..1] OF INTEGER ;
          BUF,PTR,BASE :INTEGER ;
          BUFOUT,BUFIN : INTEGER ;
  Function Binary(V: Integer): String19;

    var
      I: Integer;
      B: Array [0..3] of String[4];

    begin
      For I:=0 To 15 do
        if (V and (1 Shl (15-I)))<>0 then B[I Div 4][(I Mod 4)+1]:='1'
        else B[I Div 4][(I Mod 4)+1]:='0';
      For I:=0 To 3 do B[I][0]:=Chr(4);
      Binary:=B[0]+' '+B[1]+' '+B[2]+' '+B[3];
    end;


function KEYIN : INTEGER ;
begin
     with recpack do
       begin
               ah := 6 ;
               al := 0;
               ax := ah shl 8 + al  ;
               dx := $ff ;
          intr($21,recpack);
          al := ax and $ff ;
          KEYIN := al ;
       END;
end;






FUNCTION CHANNEL(CHANNUM:INTEGER):REAL;
begin
      ctrlbyte := 128 or gain[CHANNUM] ;
      port[startloc+4] := ctrlbyte ;
      PORT[STARTLOC+5] := INPCH[CHANNUM] ;
      PORT[STARTLOC+6] := 0 ;
      WHILE PORT[STARTLOC+4] and 128 = 0 DO
        BEGIN
```

```
            statbyte := port[startloc+4] ;
        END;
      LOW := PORT[STARTLOC+5] ;
      HIGH := PORT[STARTLOC+6] ;
      VAL := high*256.0 + low ;
      if VAL > 32767.0 then CHANNEL := VAL -65536.0
      ELSE CHANNEL := VAL ;

end;

PROCEDURE SETGAINS ;
BEGIN
  { determine best gain value for each channel }
INCH := 0 ;
  repeat
      igain := 0 ;
      VAL := CHANNEL(15);
      VAL := CHANNEL(INCH) ;
      counts[inch] := val ;
      if (abs(val)<200.0)then
      begin
           igain := 1 ;
           if (abs(val)<20.0)then
           begin
               igain := 2 ;
           end;
      end;
      gain[inch] := igain ;
      inch := inch +1 ;
   until inch = nchan ;

END;
{ read a burst of data }
procedure readburst ;
BEGIN
{ initialize stats and gains }
      for ichan := 0 to NCHAN - 1 do
      begin
          sum[ichan] := 0.0  ;
          sumx2[ichan] := 0.0 ;
          minx[ichan] := 1.0e+33 ;
          maxx[ichan] := -1.0E+33 ;
          nopts[ichan] := 0 ;
          gain[ichan] := 0 ;
      end;

SETGAINS ;           { DETERMINE BEST GAIN SETTING FOR EACH CHANNEL }

{ acquire data }

ind := 0 ;
```

Listing of: SPECRES.PAS

```
repeat
  INCH := 0 ;
   repeat
      NOPTS[INCH] := NOPTS[INCH]+1 ;
      VAL := CHANNEL(15);
      VAL := CHANNEL(INCH) ;       { read ground, REF , THEN CHANNEL }
      {  IF(REF<>0.0) THEN
            val := val/ref
         ELSE
            WRITELN(' DIVIDE BY ZERO REF VOLTS');
                                                  }
      val := val/(204.75*factor[gain[inch]]) ;
      sum[inch] := sum[inch] + val ;
      sumx2[inch] := sumx2[inch] + val*val ;
      if val < minx[inch] then minx[inch] := val ;
      if val > maxx[inch] then maxx[inch] := val ;
      inch := inch + 1 ;
   until inch = nchan ;
   IND := IND + 1;
UNTIL IND = ITER ;  { ITER IS NUMBER ITERATIONS PER BURST }
end ;


Procedure MAKEfile(VAR FL:TEXT;PROMPT:DESCRIPT ;
                        VAR FNAME:FILENAME;var ioerror:integer) ;
LABEL AGIN ;
begin
{$I-}       { turn off i/o error checking }
AGIN:     Writeln(PROMPT );
   Readln(FNAME) ;
   Assign(fl,FNAME);
   Reset(fl);  { try to rewind the file }
   IOerror := IOresult ;
   if(IOerror <> 0) then  { an error will occur if it doesn't exist }
   begin
     Rewrite(Fl) ;     { try to create the file }
     IOerror := IOresult ;
     if(IOerror <> 0)then writeln(' error in creating file: ',IOerror:5);
   end
   else
   begin
     writeln(' FILE ALREADY EXISTS, DO YOU WANT TO OVERWRITE IT? (Y/N)');
     READLN(ANSWER);
     IF (UPCASE(ANSWER)= 'Y' ) THEN
     BEGIN
            CLOSE(FL);
           GOTO AGIN ;
     END;
   end;
end;
```

Listing of: SPECRES.PAS

```
FUNCTION BCD2DEC(X:INTEGER) : INTEGER ;
BEGIN
     BCD2DEC := (X DIV 16 ) * 10 + (X MOD 16) ;
END ;

PROCEDURE TIME(VAR MONTH,DAY,HR,MIN,SEC:INTEGER) ;
CONST TIMEBASE = 893 ;
BEGIN
     PORT[TIMEBASE] := 2 ;    { SELECT SECONDS REGISTER }
     SEC := BCD2DEC(PORT[TIMEBASE+2]);
     PORT[TIMEBASE] := 3 ;    { SELECT MINUTES REGISTER }
     MIN := BCD2DEC(PORT[TIMEBASE+2]);
     PORT[TIMEBASE] := 4 ;    { SELECT HOURS REGISTER }
     HR := BCD2DEC(PORT[TIMEBASE+2]);
     PORT[TIMEBASE] := 6 ;    { SELECT DAY OF MONTH }
     DAY := BCD2DEC(PORT[TIMEBASE+2]);
     PORT[TIMEBASE] := 7 ;    { SELECT MONTH REGISTER }
     MONTH := BCD2DEC(PORT[TIMEBASE+2]);
END;

Procedure Selectchannels ;
Var i: integer ;
Begin
  WRITELN(' ENTER THE NUMBER OF CHANNELS');
  READLN(NCHAN);
  for i:= 0 to NCHAN - 1 do
  begin
        writeln(' Enter description of channel# ',i:5);
        readln(NAM[I]);
        WRITELN(' ENTER PLUG POSITION# FOR THIS CHANNEL');
        READLN(INPCH[I]);
  end;
  WRITELN(' BE SURE THAT THE GROUND (SHORTING) PLUG IS IN POSITION 15');
  INPCH[15] := 15 ;
end;

PROCEDURE ASCIN ; EXTERNAL 'ASCIN.COM' ;

PROCEDURE ASCINIT ;
BEGIN
     BASE := OFS(ASCIN) ;
     PTR := BASE + $2D ;
     BUF  :=BASE + $2F ;
     MEMW[CSEG:BASE+$10] := PTR ;
     MEMW[CSEG:BASE+$14] := BUF ;
     MEMW[CSEG:BASE+$21] := PTR ;

END;
```

```
procedure ASCII_ENABLE ;
BEGIN
     PORT[$3FC] := $0B ;
     PORT[$21] := PORT[$21] AND $EF ;
     PORT[$3F9] := 1 ;
END;

FUNCTION DATA_AVAIL : BOOLEAN ;
BEGIN
     DATA_AVAIL := TRUE ;
     BUFIN := MEMW[CSEG:PTR] ;
     IF BUFIN = BUFOUT THEN DATA_AVAIL := FALSE ;

END;

{ Beginning of Main Program ----------------------------------------}
Begin

OLDINT[0] := RSINT[0] ;
OLDINT[1] := RSINT[1] ;
ASCINIT ;
RSINT[0] := OFS(ASCIN);
RSINT[1] := CSEG ;
BUFOUT := 0 ;
ASCII_ENABLE ;
REPEAT
      OUTCHAR := KEYIN ;
      IF OUTCHAR <> 0 THEN
      BEGIN
{           REPEAT
           UNTIL ((PORT[$3FD] AND $20) <> 0 ) ;}
           PORT[$3F8] := OUTCHAR ;
      END ;
      WHILE DATA_AVAIL DO
      BEGIN
           INCHAR := MEM[CSEG:BUF+BUFOUT] ;
           BUFOUT := BUFOUT+ 1;
           IF BUFOUT > 255 THEN BUFOUT := 0 ;
           CASE INCHAR OF
           32..128,10,13: WRITE(INPCHAR);
           5: BEGIN
{                   REPEAT
                   UNTIL ((PORT[$3FD] AND $20) <> 0 ) ;}
                   PORT[$3F8] := 6 ;
              END;
           END ;
       END;
UNTIL INCHAR = 26 ;
```

```
WRITELN(' SPECTRAL RESPONSE DATA ACQUISITION PROGRAM' );
WRITELN ;
WRITELN(' written by William L. Edmonds ' );
writeln;
writeln;
writeln;
bell := chr($07);
PROMPT := ' ENTER FILE NAME FOR SPECTRAL RESPONSE DATA (ALL PTS)' ;
MAKEFILE(FL,PROMPT,FNAME,IOERROR) ;
IF(IOERROR <> 0 ) THEN GOTO STOP ;
PROMPT := ' ENTER FILE NAME FOR PLOT FILE' ;
MAKEFILE(PLOTS,PROMPT,PNAME,IOERROR);
Selectchannels ;
WRITELN(' ENTER TOTAL NUMBER OF DATA POINTS FOR EACH WAVELENGTH');
READLN(NPTS);
ITER := 10 ;
NITER := NPTS DIV ITER ;

     writeln(' Enter description of this run (80 chars)');
     readln(descrip);
     writeln(' Enter START WAVENUMBER (real number with decimal)');
     readln(WAVELENGTH);
     WRITELN(' ENTER DELTA WAVENUMBER (REAL NUMBER )');
     READLN(DELTA);
     WRITELN(' ENTER NUMBER OF STEPS (INTEGER)');
     READLN(STEPS);
     WRITELN(' ENTER DWELL TIME IN SECONDS (INTEGER)');
     READLN(DWELL);
     writeln(' Type G when ready to start taking data ') ;
     writeln(' OR enter Q to quit') ;
     readln(answer);
     IF(UPCASE(ANSWER)<>'G') THEN GOTO STOP;
  WAVEL := WAVELENGTH ;
  writeln(fl,descrip);
FOR j:= 1 TO STEPS DO    { wavenumber loop }
BEGIN
  if (UPCASE(answer) <> 'G' ) then goto stop ;
  WRITELN(' WAVELENGTH = ' ,WAVEL:10:2);
  WRITELN(LST,' WAVELENGTH = ',WAVEL:10:2);
  TIME(MONTH,DAY,HR,MIN,SEC);
  WRITELN(fl,MONTH:2,'/',DAY:2,'/86  ',HR:2,':',MIN:2,':',SEC:2);
  writeln(fl,
     'parameter    minimum   maximum      mean    std dev    num pts');
  WRITELN(lst,
     MONTH:2,'/',DAY:2,'/86  ',HR:2,':',MIN:2,':',SEC:2);
  writeln(lst,
     'parameter    minimum   maximum      mean    std dev    num pts');
  WRITELN(MONTH:2,
     '/',DAY:2,'/86  ',HR:2,':',MIN:2,':',SEC:2);
  writeln(
     'parameter    minimum   maximum      mean    std dev    num pts');
```

```
   WRITELN(FL,' WAVELENGTH = ',WAVEL:10:2);
   for ind := 0 to nchan -1 do
   BEGIN
      tmean[ind]:=0.;
      tsum[ind]:=0.;
      tsumx2[ind]:=0.;
      tminx[ind]:=1.0e+33;
      tmaxx[ind]:=-1.e+33;
      tnopts[ind]:=0.;
      tstd[ind]:=0.;
    end;
{   readburst ;   read each channel to initialize process }
    for k:= 1 to niter do
    begin
{     readburst ; }
            for ind := 0 to NCHAN - 1 do
            begin
                mean[ind] := sum[ind]/nopts[ind] ;
                RADICAL := (nopts[ind]*sumx2[ind]-sum[ind]*sum[ind])/
                                  ((nopts[ind]-1)*nopts[ind]) ;
                tsum[ind]:= tsum[ind]+sum[ind];
                tsumx2[ind]:= tsumx2[ind]+sumx2[ind];
                if(minx[ind]<tminx[ind])then tminx[ind]:=minx[ind];
                if(maxx[ind]>tmaxx[ind])then tmaxx[ind] :=maxx[ind];
                tnopts[ind]:=tnopts[ind]+nopts[ind];
                IF(RADICAL>0.0) THEN
                BEGIN
                STD[IND]:= SQRT(RADICAL) ;
                END
                ELSE
                BEGIN
                STD[IND] := 0.0 ;
                END;
{                writeln(NAM[ind]:10,mean[ind]:8:4,std[ind]:10:4
                          ,factor[gain[ind]]:5:1);
 }                writeln(Fl,nam[IND]:10,minX[IND]:10:5,maxX[IND]:10:5,
                 mean[IND]:10:5,std[IND]:10:5,NOPTS[IND]:10:0);
{                 writeln(lst,nam[IND]:10,minX[IND]:10:5,maxX[IND]:10:5,
                 mean[IND]:10:5,std[IND]:10:5,NOPTS[IND]:10:0);
 }         end; { of for loop }
           writeln(fl);
end;

           for ind := 0 to nchan - 1 do
           begin
                tmean[ind] := tsum[ind]/tnopts[ind] ;
                radical := 0.0 ;
                denom := ((tnopts[ind]-1)*tnopts[ind]);
                if(denom<>0.0) then
                   radical := (tnopts[ind]*tsumx2[ind]-tsum[ind]*tsum[ind])
                   / denom;
```

```
              if(radical>0.0)then
              begin
                  tstd[ind] := sqrt(radical);
              end
              else
              begin
                  tstd[ind] := 0. ;
             end;
             writeln(f1,nam[ind]:10,tminx[ind]:10:5,tmaxx[ind]:10:5,
             tmean[ind]:10:5,tstd[ind]:10:5,tnopts[ind]:10:5);
             writeln(lst,nam[ind]:10,tminx[ind]:10:5,tmaxx[ind]:10:5,
             tmean[ind]:10:5,tstd[ind]:10:5,tnopts[ind]:10:5);
             writeln(nam[ind]:10,tminx[ind]:10:5,tmaxx[ind]:10:5,
             tmean[ind]:10:5,tstd[ind]:10:5,tnopts[ind]:10:5);
        end;
              WAVEL := WAVELENGTH + j*DELTA ;
              WRITELN(bell,
                'ENTER G WHEN READY TO TAKE DATA FOR WAVELENGTH ='
                ,WAVEL:10:2);
                READLN(ANSWER);
    END;
    stop:
         WRITELN(FL);
         close(f1) ;
END.
```

APPENDIX D - SPECPLT

Program Name: SPECPLT.PAS

Function: SPECPLT is designed to plot HALOE spectral response data on an HP pen plotter.

Description: SPECPLT is written in Turbo Pascal for an IBM-XT or compatible.

Use: After each spectral response run is made, it is essential to plot the data to determine the quality of the data and whether or not an additional run under the same conditions is necessary. SPECPLT gives the capability of plotting the data quickly, allowing several parameters to be plotted in different colors on the same graph.

Listing of: A:SPECPLT.PAS

```
(***********************************************************************)
(***********************************************************************)
(*                                                                     *)
(*                TURBO PASCAL PLOT PROGRAM for Spectral               *)
(*                  Response using IEEE 488 BUS DRIVER                 *)
(*                                                                     *)
(*                                                                     *)
(***********************************************************************)
(***********************************************************************)
Program Specplt ;
type
  filename = string[12] ;
  name = string[10] ;
  names = array[0..16] of name ;
  cmd = string[127];
  VALUE = STRING[10];
  vax = string[80];
  flg = integer;
  bad = integer;
  INTS = ARRAY[0..10] OF INTEGER ;
  ANTS = ARRAY[0..21] OF BYTE ;
  param = array[1..200] of real ;
  STRG = STRING[40] ;
CONST ZERO : STRING[3] = ' 0 ' ;
      MINEQ : STRING[6] = 'MIN = ' ;
      MAXEQ : STRING[6] = 'MAX = ' ;
      MINIMUM : REAL = 1.0E+33 ;
      MAXIMUM : REAL = -1.0E+33 ;
      ET : BYTE = 3 ;

Label TOP,NEWPLOT,theEnd ;
var
  ETX : CHAR ABSOLUTE ET ;
  PENPOS : VAX ;
  LAB : STRG ;
  ANSWER : CHAR ;
  nparam,CHAN : integer ;
  params : array[1..16] of param ;
  PARVAL : string[10] ;
  parnam : ARRAY[0..16] OF name ;
  PARNAME : NAME ;
  waveleng : param ;
  WAVEVAL : string[10] ;
  parmin,parmax : array[1..16] of real ;
  wavemin,wavemax : real ;
  title : STRG ;    { title of plot can be up to 40 characters }
  XLAB,YLAB,DIR:VALUE ;
  date,datime : value ;  { 10 character strings for date and time }
  XCOORD,YCOORD : REAL ;
  I,J,npt:INTEGER ;
```

```
     X,Y: VALUE ;
     MINX,MINY,MAXX,MAXY : REAL ;
     XSF,YSF,XOF,YOF : REAL ;  { X&Y SCALE FACTORS AND OFFSETS }
     XTIC : VALUE ;
     XPOS : REAL ;
     XDIV,YDIV : INTEGER ;
     XDEL,YDEL,ydelta : REAL ;
     syscon:cmd;
     f:flg;
     b:bad;
     v:vax;
     RX,RY:REAL ;
     c:cmd;
     IANS: CHAR ;
     NUMS: INTS ABSOLUTE V ;
     BYTES : ANTS ABSOLUTE V ;
     TEMP : BYTE ;
     specfile : text ;
     specfilename : filename ;
     ioerror : integer ;
     PEN : CHAR ;

   Procedure Openfile(var FL:TEXT;var FNAME:FILENAME;var ioerror:integer) ;
   LABEL AGIN ;
   begin
   {$I-}       { turn off i/o error checking }
   AGIN:      Writeln(' Enter plot data file name ' );
        Readln(FNAME) ;
        Assign(fl,FNAME);
        Reset(fl);
        IOerror := IOresult ;
        if(IOerror <> 0) then
        begin
            writeln(' File : ',fname,' does not exist! ');
            writeln(' DO YOU WANT TO TRY AGAIN? (Y/N)');
            READLN(ANSWER);
            IF(UPCASE(ANSWER) = 'Y' ) THEN
            goto agin ;
        end
        else
        begin
            writeln(' OPENING FILE: ',FNAME);
        end;
   end;

   procedure ReadInData ( var ioerror : integer ) ;
   VAR PRINT : BOOLEAN ;
   LABEL FINIS ;
   begin
       WRITELN(' DO YOU WANT TO PRINT THE DATA?');
       READLN(ANSWER);
```

```
    IF UPCASE(ANSWER) = 'Y' THEN PRINT := TRUE ELSE PRINT := FALSE ;
    readln(specfile,title);
    writeln(' title : ', title ) ;
    readln(specfile,date,datime);
    writeln(' date and time : ',date,datime);
    readln(specfile,nparam);
    writeln(' number of parameters = ',nparam:5);
    READ(SPECFILE,PARNAM[0]);
    WAVEMIN := MINIMUM ;
    WAVEMAX := MAXIMUM ;
IF PRINT THEN
BEGIN
    WRITELN(LST,TITLE);
    WRITELN(LST,DATE,DATIME);
    WRITELN(LST,' NUMBER OF PARAMETERS = ',NPARAM);
    WRITE(LST,PARNAM[0]);
END;
    for i := 1 to nparam do
    begin
        read(specfile,parnam[i]);
        IF PRINT THEN WRITE(LST,PARNAM[I]);
        PARMIN[I] := MINIMUM ;
        PARMAX[I] := MAXIMUM ;
    end;
    IF PRINT THEN WRITELN(LST);
    npt := 0 ;
repeat
        npt := npt + 1 ;
        read(specfile,waveLENG[npt]);
        IF EOF(SPECFILE) THEN GOTO FINIS ;
        IF PRINT  THEN WRITE(LST,WAVELENG[NPT]:10:2);
        for j:= 1 to nparam do
        begin
           read(specfile,parAMS[j,npt]);
           IF EOF(SPECFILE) THEN GOTO FINIS ;
           IF PRINT THEN WRITE(LST,PARAMS[J,NPT]:10:5);
        end;
        IF PRINT THEN WRITELN(LST);
until eof(specfile) ;
FINIS: NPT := NPT-1 ;
FOR I := 1 TO NPT DO
BEGIN
        IF WAVELENG[I] < WAVEMIN THEN WAVEMIN := WAVELENG[I] ;
        IF WAVELENG[I] > WAVEMAX THEN WAVEMAX := WAVELENG[I] ;
    FOR J := 1 TO NPARAM DO
    BEGIN
        IF PARAMS[J,I] < PARMIN[J] THEN PARMIN[J] := PARAMS[J,I] ;
        IF PARAMS[J,I] > PARMAX[J] THEN PARMAX[J] := PARAMS[J,I] ;
    END ;
END;
end;
```

```
procedure IE488 ( VAR c:cmd;
                  VAR v:vax;
                  VAR f:flg;
                  VAR b:bad ); external 'IE488.COM';

PROCEDURE LABELIT(VAR LAB:STRG; VAR X,Y, DIRECTION: VALUE);
BEGIN
    V:='DI ' + DIRECTION + ' ; ' ;
    IE488(C,V,F,B);
    V := 'PU PA ' + X + Y + ' ; ' ;
    IE488(C,V,F,B);
    V := 'LB ' + LAB + ETX ;
    IE488(C,V,F,B);
END;


PROCEDURE INITIEEE ;
BEGIN

  f := 1;
  b := 0;
  syscon := 'SYSCON MAD=3, CIC=1, NOB=1, BAO=&H200';
  v := '                              ';
  IE488(syscon,v,f,b);
  if f<> 0 then
  writeln('RETURNED FROM IE488 SYSCON PROCEDURE flg = ', f);
  F:= 0;
  B:= 0 ;
  C := 'TIMEOUT' ;
  V := chr(1) ;
  IE488(C,V,F,B);
  if f<>0 then
  WRITELN(' TIMEOUT PROC RETURN WITH FLAG =',F);
  C:= 'OUTPUT 5[$#]' ;
END;

PROCEDURE INITPLOT ;
BEGIN

  V := 'DF IN PS 4 IP 0,0,9865,7462;' ;
  IE488(C,V,F,B);
  V := ' SC -20,100,-10,110 ;';
  IE488(C,V,F,B);
  if f<>0 then
  WRITELN(' INITIALIZED PLOTTER, FLAG = ',F);
  WRITELN(' WHAT PEN NUMBER DO YOU PREFER?');
  READLN(PEN);
  V:= 'SP ' + PEN + ';' ;
  IE488(C,V,F,B);
```

```
     V := 'PA 0,0,PD 100,0,100,100,0,100,0,0 ;' ;
     IE488(C,V,F,B);
     V:=    ' PU 0,0 ;' ;
     IE488(C,V,F,B);

   END;

   PROCEDURE AXES;
   BEGIN
       XDEL := 100.0/XDIV ;
       YDEL := 100.0/YDIV ;
       V:= ' ' ;
       FOR I:= 1 TO XDIV DO
       BEGIN
         XPOS := I*XDEL ;
         STR(XPOS:8:4,XTIC);
         V :=  'PA ' + XTIC + ','+ ZERO + ';'+'XT;' ;
         IE488(C,V,F,B);
       END;
       FOR I := 1 TO YDIV DO
       BEGIN
         XPOS  := I * YDEL ;
         STR(XPOS:8:4,XTIC);
         V := 'PA ' + ZERO + ',' + XTIC + ';' + 'YT ;';
         IE488(C,V,F,B);
       END;
         V := 'PU PA 0,0 ;'   ;
       IF F<>0 THEN WRITELN(' ERROR IN AXES = ',F);

   END;
   procedure plotline ;
   BEGIN
      I := 1 ;
      XCOORD := (WAVELENG[I]-XOF)*XSF ;
      YCOORD := (PARAMS[CHAN,I]-YOF)*YSF ;
      STR(XCOORD:10:2,WAVEVAL);
      STR(YCOORD:10:2,PARVAL);
      penpos := 'PU ' ;
      V := penpos +  WAVEVAL + ',' + PARVAL + ';'   ;
      IE488(C,V,F,B) ;
      PENPOS := 'PD ' ;
      FOR I := 1 TO NPT DO
      BEGIN
           XCOORD := (WAVELENG[I]-XOF)*XSF ;
           YCOORD := (PARAMS[CHAN,I]-YOF)*YSF ;
           STR(XCOORD:10:2,WAVEVAL);
           STR(YCOORD:10:2,PARVAL);
           V := penpos +  WAVEVAL + ',' + PARVAL + ';'   ;
           IE488(C,V,F,B) ;
       END ;
```

```
END;

PROCEDURE SETSCALES ;
BEGIN

     WRITELN(' CURRENT WAVENUMBER MIN AND MAX ARE: ',WAVEMIN:10:2,
     WAVEMAX:10:2);
     WRITELN(' CURRENT MIN AND MAX FOR ',PARNAM[CHAN],':',
     PARMIN[CHAN]:10,'   ',PARMAX[CHAN]:10);
     writeln(' DO YOU WANT TO ADJUST THESE? (Y/N)');
     READLN(ANSWER);
     IF(UPCASE(ANSWER) = 'Y') THEN
REPEAT
     WRITELN(' ENTER WAVENUMBER MINIMUM: ');
     READLN(WAVEMIN);
     WRITELN(' ENTER WAVENUMBER MAXIMUM: ');
     READLN(WAVEMAX);

     WRITELN(' ENTER MIN FOR:',PARNAM[CHAN]);
     READLN(PARMIN[CHAN]);
     WRITELN(' ENTER MAX FOR:',PARNAM[CHAN]);
     READLN(PARMAX[CHAN]);
     WRITELN(' MIN AND MAX WAVENUMBERS: ',WAVEMIN:10:2,WAVEMAX:10:2);
     WRITELN(' MIN AND MAX FOR ',PARNAM[CHAN],PARMIN[CHAN]:10,
     '   ',PARMAX[CHAN]:10);
     WRITELN(' ARE THESE VALUES OK? (Y/N)');
     READLN(ANSWER);
UNTIL UPCASE(ANSWER) = 'Y' ;
XDEL := WAVEMAX-WAVEMIN ;
YDEL := PARMAX[CHAN]-PARMIN[CHAN] ; ;
XSF := 100.0/XDEL ;
YSF := 100.0/YDEL ;
XOF := WAVEMIN ;
YOF := PARMIN[CHAN] ;
ydelta := ydel ;
END ;


PROCEDURE YLABEL(pmin,pmax:real;pnam:name) ;
BEGIN

V := 'PU PA 0 0 ;';
IE488(C,V,F,B);
YLAB := ' 0 ' ;
STR(Pmin:10,LAB);
LAB := MINEQ + LAB ;
DIR := '0 1 ' ;
LABELIT(LAB,XLAB,YLAB,DIR);

YLAB := '40 ' ;
LAB := Pnam ;
```

```
DIR := '0 1 '   ;
LABELIT(LAB,XLAB,YLAB,DIR);

YLAB := ' 70 ' ;
STR(Pmax:10,LAB);
LAB := MAXEQ + LAB ;
LABELIT(LAB,XLAB,YLAB,DIR);

END;

{------------------------------ S P E C P L T  MAIN PROGRAM ---------------

BEGIN
INITIEEE ;              { INITIALIZE IEEE BUS FOR PLOTTING }

TOP:      OPENFILE(specfile,specfilename,ioerror);

if ioerror <>0 then goto theEnd ;
Readindata(ioerror) ;
if ioerror <>0 then goto theEnd ;

NEWPLOT:

XDIV := 10 ;
YDIV := 10 ;

FOR I:= 1 TO NPARAM DO
WRITELN('CHANNEL# ',I:5,PARNAM[I]:12) ;
WRITELN(' ENTER CHANNEL # TO PLOT AGAINST WAVELENGTH');
READLN(CHAN);
SETSCALES ;

INITPLOT ;
AXES ;
 XLAB := ZERO ;
YLAB := '100 ' ;
DIR := ' 1 0 ' ;
LABELIT(TITLE,XLAB,YLAB,DIR);
XLAB := ' 50 ' ;
LAB := DATE + ' ' + DATIME ;

LABELIT(LAB,XLAB,YLAB,DIR);

PLOTLINE;


XLAB := ' 0 ' ;
YLAB := '-5 ' ;
STR(WAVEMIN:8:2,LAB);
LAB := MINEQ + LAB ;
LABELIT(LAB,XLAB,YLAB,DIR);
```

```
XLAB := '40 ' ;
YLAB := '-5 ' ;
LAB := PARNAM[0] ;
 LABELIT(LAB,XLAB,YLAB,DIR);

XLAB := ' 70 ' ;
STR(WAVEMAX:8:2,LAB);
LAB := MAXEQ + LAB ;
LABELIT(LAB,XLAB,YLAB,DIR);

XLAB := '-5 ' ;
YLABEL(parmin[chan],parmax[chan],parnam[chan]) ;

WRITELN(' DO YOU WANT TO PLOT ANOTHER CHAN ON SAME PLOT? (Y/N)');
READLN(ANSWER);
IF UPCASE(ANSWER) = 'Y' THEN
BEGIN
    WRITELN(' WHAT PEN NUMBER DO YOU PREFER?');
    READLN(PEN);
    V:= 'SP ' + PEN + ';' ;
    IE488(C,V,F,B);
    XLAB := '-10 ' ;
    FOR I:= 1 TO NPARAM DO
    WRITELN('CHANNEL# ',I:5,PARNAM[I]:12) ;
    WRITELN(' ENTER CHANNEL # TO PLOT AGAINST WAVELENGTH');
    READLN(CHAN);
    WRITELN(' DO YOU WANT TO USE THE SAME SCALE-FACTOR (Y/N)  ');
    READLN(ANSWER);
    IF UPCASE(ANSWER) = 'N' THEN
        SETSCALES
    else
        YOF := 0.0 ;

    YLABEL(0.0,ydelta,parnam[chan]) ;
    PLOTLINE ;

END;



v := ' SP 0 ; ' ;
IE488(C,V,F,B);
WRITELN(' DO YOU WANT TO CONTINUE? (Y/N)');
READLN(ANSWER);
IF (UPCASE(ANSWER)='N') THEN
GOTO THEEND
ELSE
BEGIN
     WRITELN(' SAME FILE?(Y/N)');
     READLN(ANSWER);
```

```
            IF(UPCASE(ANSWER)='Y') THEN GOTO NEWPLOT;
            CLOSE(SPECFILE);
            GOTO TOP;
       END;
       THEEND:
       CLOSE(SPECFILE);
       END.
```

APPENDIX E - MONITOR

Program Name: MONITOR.PAS

Function: MONITOR is designed to acquire HALOE major frames of data and to limit check the data before displaying it on a color monitor in color coded form. MONITOR will also archive data to disk for off-line processing.

Description: MONITOR is a Turbo Pascal program written on an IBM-XT.

Use: MONITOR will be used to limit check, display and archive HALOE major frames of data during refurb testing and UARS I & T (Upper Atmosphere Research Satellite Integration and Testing). It will be part of an overall quick-look system for HALOE.

```
 PROGRAM MONITOR ;
{

      Monitor is a HALOE program designed to process HALOE
  major frames of data sent to the IBM-XT (or compatible)
  by the IETS HP-1000 over the HPIB (IEEE-488 interface bus).
  Monitor will convert the raw counts to engineering units
  and perform limit checking and color coding of the data
  before display on the color monitor. Monitor will also
  archive data to disc for transfer later to an off-line
  system for further processing and evaluation.

  THIS PROGRAM WILL SET UP AN INTERRUPT VECTOR TO ITSELF,
  AND LOCK ITSELF IN MEMORY TO BE CALLED BY FORTH LATER
  USING AN INTERRUPT 48 (HEX) }

type
    ivdt = record         { variable definition data }
           leng : byte ;
           loc : integer ;
           bitpos, equatnum : byte ;
           SCRPOS : INTEGER ;  { SCREEN POSITION }
           IDNAM : STRING[8] ;
           end;
    icoef = record       { coefficients for conversion equations }
            slope, offset : real ;
            end;
    regs = record
           AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER ;
           END;

var
    REGSET : REGS ;
    CSEGM,OFFS : INTEGER ;
    ID1,ID2 : INTEGER ;
    ANSWER : CHAR ;
    VDTfileNAM : STRING[15] ;
    vdt :  ivdt ;
    vtble : array[1..200] of ivdt ;
    VDTFILE : FILE OF IVDT ;
    coefFILEnam : string[15] ;
    coef : icoef ;
    coefTBLE : ARRAY[1..50] OF ICOEF ;
    COEFfile : file of icoef ;
    WORDNUM : INTEGER ;
    BYTEDISP : INTEGER ;
    BITDISP : BYTE ;


const datseg :ARRAY[0..1] OF integer = (0,0) ;
```

```
          STSEG : INTEGER = 0 ;
          EXSEG : INTEGER = 0 ;
          STPT : INTEGER = 0 ;
          oldstseg : integer = 0 ;
          oldstpt : integer = 0 ;
          base : integer = $200 ;
          HEXDIG : ARRAY[0..15] OF CHAR = '0123456789ABCDEF' ;
     var
          SCRNMODE : ARRAY[0..15] OF BYTE ; {DISPLAY PARAMETERS FOR GRAPHICS}
          dataseg : ARRAY[0..1] OF integer absolute datseg ;
          STACKSEG : INTEGER ABSOLUTE STSEG ;
          STACKPT : INTEGER ABSOLUTE STPT ;
          ESSEG : INTEGER ABSOLUTE EXSEG ;
          ZILCH : integer ;
          INTVEC : ARRAY[0..1] OF INTEGER ABSOLUTE $0000:$0120;
          basearray : array[0..15] of byte absolute $0000:$0200 ;
          year,day: string[5] ;
          hours,minutes,seconds : string[3] ;
          DELTA,START,STOP: REAL ;
          sorc : string[80] ;
    type
      ABC = STRING[80] ;
      cmd = string[127];
      vax = string[255];
      flg = integer;
      bad = integer;
      INTS = ARRAY[0..302] OF INTEGER ;
      ANTS = ARRAY[0..604] OF BYTE ;
      INTEGBUFF = ARRAY[0..4000] OF INTEGER ;
      BYTEBUFF = ARRAY[0..8000] OF BYTE ;
      HEXVAL = STRING[4] ;

    var
      COMM : INTEGER ; { HOLDS COMMAND VALUE FROM ODD OR EVEN COMMAND WORD }
      INDEX : INTEGER; { COMM IS USED TO CALCULATE INDEX OF COMMAND IN TBLE}
      port21 : byte ;  { 8259 interrupt mask register }
      txt : text ;
      txtfile : string[10] ;
      att : integer ;
      I,J,ind:INTEGER ;
      COUNT : INTEGER ;
      syscon:cmd;
      f:flg;
      b:bad;
      needmoredata : boolean ;
      STATUS : INTS  ;
      STAT : VAX ABSOLUTE STATUS ;
      numsaddr : INTS ;
      NUMSAD : VAX ABSOLUTE NUMSADDR ;
      c:cmd;
      IANS: CHAR ;
```

```
  NUMS: INTS  ;
  BYTES : ANTS ABSOLUTE NUMS ;
  V : VAX ABSOLUTE NUMS ;
  TEMP : BYTE ;
  FRAME : INTEGBUFF ABSOLUTE $B800:$0000;
  BFRAME : BYTEBUFF ABSOLUTE $B800:$0000;
  TIMER : BYTE ABSOLUTE $0040:$006C ;
  mask,mask2,num1,num2,shift: integer ;  { used by bits function }
  LINENUM,CHARNUM :INTEGER ;
  SCRNINT : ARRAY[0..1] OF INTEGER ABSOLUTE $0000:$0014 ;
  STORINT : ARRAY[0..1] OF INTEGER ;
  STATPR : BYTE ABSOLUTE $0050:$0000 ;


PROCEDURE SETINTVEC(SEGM,OFFS:INTEGER) ;
{ set up interrupt vector number $48 (hex) to point to
  the main subroutine }
var ah,al : byte ;
BEGIN
     WITH REGSET DO
     BEGIN
          DS := SEGM ;
          DX := OFFS ;
          ah := $25 ;
          AX :=( ah shl 8) or $48 ;
          INTR($21,REGSET);
     END;
END;


FUNCTION HEX(VAL:INTEGER): HEXVAL ;
{ convert val into a hex string }
BEGIN
     HEX := HEXDIG[VAL SHR 12] +
            HEXDIG[(VAL SHR 8) AND 15] +
            HEXDIG[(VAL SHR 4) AND 15] +
            HEXDIG[VAL AND 15] ;
END;


FUNCTION BITS(NUMS:ints;IND:INTEGER;BITPOS,LENGTH:BYTE):INTEGER ;
{ extract length bits from bitpos of nums[ind] }
BEGIN
     BITPOS :=  16 - BITPOS ;
     NUM1 := NUMS[IND];
     NUM2 := NUMS[IND+1] ;
     SHIFT := BITPOS - LENGTH ;
     IF SHIFT < 0 THEN
     BEGIN
         MASK := ($FFFF SHR (16 - BITPOS)) ;
```

Listing of: MONITOR.PAS

```
         MASK2 := $FFFF SHR (16+SHIFT) ;
         BITS := ((NUM1 AND MASK ) SHL -SHIFT) OR
                 ((NUM2) SHR ( 16 + SHIFT)) AND MASK2  ;
     END
     ELSE
     IF SHIFT = 0 THEN
     BEGIN
         MASK := $FFFF SHR ( 16 - LENGTH ) ;
         BITS := MASK AND NUM1 ;
     END
     ELSE
     BEGIN
          MASK := $FFFF SHR (16 - LENGTH) ;
          BITS := (NUM1 SHR SHIFT) AND MASK ;
     END;
END;


procedure SCRDUMP(var i,j: integer) ;
TYPE CHARBUFF = ARRAY[0..8000] OF CHAR ;
VAR CFRAME: CHARBUFF ABSOLUTE $B800:$0000;
PRFRAME: ARRAY[0..4000] OF CHAR ;
K ,l: INTEGER ;
begin
IF (I+J = 0) THEN
BEGIN
FOR K := 0 TO 3999 DO
BEGIN
PRFRAME[K] := CFRAME[K*2];
END;
END;
for l:= 0 to 4 do
begin
   if (j<79)then
   begin
       WRITE(LST,PRFRAME[I*80 +j])   ;
   end
   else
   begin
       writeln(lst,PRFRAME[I*80+j]);
   end;
   j:= j+1;
end;
if (j>79) then
begin
j:=0;
i := i+1;
if (i>48) then
begin
i := 0;
statpr :=0;
```

```
end;
end;
end;


FUNCTION STACK : INTEGER ; EXTERNAL 'STACK.COM' ;
{ STACK RETURNS VALUE OF STACK POINTER }

FUNCTION ESEGM : INTEGER ; EXTERNAL 'ESEG.COM' ;
{ RETURNS VALUE OF ES ..EXTRA SEGMENT REGISTER }

procedure IE488 ( VAR c:cmd;
                  VAR v:vax;
                  VAR f:flg;
                  VAR b:bad ); external 'IE488.COM';


PROCEDURE S5080(var  i :byte); EXTERNAL 'CONO.COM';
{ S5080 PUTS THE CONOGRAPHICS SYSTEM IN THE DESIRED MODE:
  At program start, it puts the screen in 50 row,80 column mode.
  At termination, it returns the screen to 25 X 80 . }

PROCEDURE PUTOUT(VAR SORC:ABC;VAR FRAME:INTEGER;ATTR:INTEGER);
 EXTERNAL 'PUTOUT.COM';
{ PUTOUT places a string and its color attributes
 in the screen memory area }

FUNCTION PRSTAT:INTEGER; EXTERNAL 'PRSTAT.COM';
{ PRSTAT responds to the shift-PrtSC keys by setting a flag.
  The program will then dump the screen to the printer
  50 rows by 80 columns }

FUNCTION XYPOS(ROW,COL:INTEGER ):INTEGER ;
BEGIN
     XYPOS := ROW * 80 + COL;
END;


procedure NEWSCREEN ;
{ set up conographics screen mode with 80 columns and 50 rows }
BEGIN
     SCRNMODE[0] := $71;
     SCRNMODE[1] := $50;
     SCRNMODE[2] := $5A;
     SCRNMODE[3] := $0F;
     SCRNMODE[4] := $1B;
     SCRNMODE[5] := 6;
```

```
        SCRNMODE[6] := $19;
        SCRNMODE[7] := $1A;
        SCRNMODE[8] := 3;
        SCRNMODE[9] := 7;
        SCRNMODE[10] := $20 ;
        SCRNMODE[11] := $20 ;
        SCRNMODE[12] := 0;
        SCRNMODE[13] := 0;
        SCRNMODE[14] := 0;
        SCRNMODE[15] := 0;
        S5080(SCRNMODE[0]);

   END;

   PROCEDURE OLDSCREEN ;
   { restore old screen mode }
   VAR LOC : INTEGER ;
   BEGIN
         FOR LOC := 0 TO 3999 DO
         FRAME[LOC] := $F00 ;

         SCRNMODE[4] := $1F ;
         SCRNMODE[7] := $1C ;
         SCRNMODE[8] := 2;
         SCRNMODE[10] := 6;
         SCRNMODE[11] := 7;
         S5080(SCRNMODE[0]);
   END;

   PROCEDURE DISPLAYACRO ;
   { display background for limit check screen }
   VAR I: INTEGER;
   BEGIN
    txtfile := 'HALOE.SCR' ;
    assign(txt,txtfile);
    reset(txt);
    att := 15 ;
   i := 0 ;
   while not eof(txt) do
   begin
   readln(txt,sorc);
   sorc := sorc + '                ';
   putout(sorc,frame[i],att);
   i := i + 80 ;
   end;
   close (txt);

   END;
```

```
{ ---------------PROCEDURES & FUNCTIONS--------------}

PROCEDURE OUTPUT(VAR SORC: ABC; VAR FRAME: INTEGER; ATTR: INTEGER);
VAR BLANKS : ABC ;
BEGIN
{      BLANKS := '             ' ;  10 BLANKS }
 {     PUTOUT(BLANKS,FRAME,WHITE);    }
      PUTOUT(SORC,FRAME,ATTR);
END;

function bcd2dec(x:integer):integer;
{ convert bcd value x into decimal value }
begin
     bcd2dec :=(x div 16 )*10 + (x mod 16) ;
end;


FUNCTION TIME: REAL ;
CONST TIMEBASE = 893 ;
VAR TSEC,HUNDSEC,SEX,MENS: INTEGER ;
BEGIN
     PORT[TIMEBASE] := 0 ;    { SELECT THOUSANDTHS OF SECONDS REGISTER }
     TSEC := BCD2DEC(PORT[TIMEBASE+2]);
     PORT[TIMEBASE] := 1 ;    { SELECT HUNDREDTHS AND TENTHS REGISTER }
     HUNDSEC := BCD2DEC(PORT[TIMEBASE+2]);
     PORT[TIMEBASE] := 2 ;
     SEX := BCD2DEC(PORT[TIMEBASE+2]) ;
     PORT[TIMEBASE] := 3 ;
     MENS := BCD2DEC(PORT[TIMEBASE+2]) ;

     TIME := TSEC/1000. + HUNDSEC/100. + SEX + MENS*60.0;
END;

PROCEDURE DISPLAY(ITEM,NDEC,XPOS,YPOS,COLR:INTEGER ) ;
BEGIN
      STR(ITEM:NDEC,SORC);
      PUTOUT(SORC,FRAME[XYPOS(XPOS,YPOS)],COLR);

END;

PROCEDURE DISPLAYDATA ;
VAR VALU,K : INTEGER ;
    xvalu : real ;
    VDT1,VDT2 : IVDT ;
    KDV,XV,XDV,BBI,BBV : REAL ;

CONST COLON : CHAR = ':' ;
LABEL THEexit ;
BEGIN

      str(nums[10]:5,year);
str(nums[9]:5,day);
str(nums[8]:3,hours);
str(nums[7]:2,minutes);
str(nums[6]:2,seconds);
sorc := year + day + hours + COLON
         + minutes + COLON + seconds ;
putout(sorc,frame[xypos(0,57)],yellow);
FOR I:= 1 TO 10 DO
BEGIN
      VDT := VTBLE[I] ;
      IF VDT.SCRPOS > 0 THEN
      BEGIN
             K := VDT.LOC -1 ;
             VALU := BITS(NUMS,K,VDT.BITPOS,VDT.LENG);
             J := VDT.EQUATNUM ;
             IF J > 128 THEN J := J -256 ;
             IF J > 0 THEN
             BEGIN
                 if( J < 51) and (j <> 2)then
                 begin
                     COEF := COEFTBLE[J] ;
                     xvalu :=  valu*(COEF.SLOPE) + COEF.OFFSET ;
                     str(xvalu:6:3,sorc);
                 end
                 else
                   IF (I=94) OR (I=95) THEN
                   BEGIN    { EVEN OR ODD COMMAND }
                    SORC := HEX(VALU) +'   ';
                    PUTOUT(SORC,FRAME[VDT.SCRPOS],GREEN);
                    COMM := VALU SHR 12 ;  { GET COMMAND NUMBER }
                    VALU := VALU AND 4095 ;
                    CASE COMM OF
                    1,3,5,7,9,11,13,15: INDEX := COMM div 2 + 110 ;
                    0,2,4,6,8,10,12,14: INDEX := COMM div 2 + 100 ;
                    END;

                    VDT := VTBLE[INDEX] ;  { SELECT TABLE ENTRY FOR
                                                          THIS COMMAND}
                    sorc := hex(valu);
                   END
                   else
                       STR(VALU:6,SORC);
             END
             ELSE
                 BEGIN
 { SPECIAL PROCESSING.. EQUIVALENT OF ISPCL IN HP SOFTWARE}
                       J := ABS(J) -25 ;
                       IF (J<0) OR (J>6) THEN GOTO THEexit ;
                       CASE J OF
                       1,2,3,4 :
                       BEGIN
```

```
                               ID2 := 2*J ;
                               ID1 := ID2 -1 ;
                               VDT1 := VTBLE[ID1] ;
                               VDT2 := VTBLE[ID2] ;
                               K := VDT1.LOC -1 ;
                               XV := BITS(NUMS,K,VDT1.BITPOS,VDT1.LENG);
                               COEF := COEFTBLE[VDT1.EQUATNUM] ;
                               XV := XV*COEF.SLOPE + COEF.OFFSET ;
                               K := VDT2.LOC -1 ;
                               XDV := BITS(NUMS,K,VDT2.BITPOS,VDT2.LENG);
                               COEF := COEFTBLE[VDT2.EQUATNUM] ;
                               XDV := XDV*COEF.SLOPE + COEF.OFFSET ;
                               KDV := 58.0;
                               IF J=4 THEN
                                     BEGIN
                                          XDV := XDV + 4.639 ;
                                          KDV := 29.0 ;
                                     END;
                                XVALU := XDV/KDV*1.E6 ;
                             END;
                             5,6:
                             BEGIN
                                VDT1 := VTBLE[21] ;   { BBI }
                                VDT2 := VTBLE[48] ;   { BBV }
                                K:= VDT1.LOC -1 ;
                                BBI :=BITS(NUMS,K,VDT1.BITPOS,VDT1.LENG);
                                COEF := COEFTBLE[VDT1.EQUATNUM];
                                BBI := BBI*COEF.SLOPE + COEF.OFFSET ;
                                K := VDT2.LOC -1 ;
                                BBV := BITS(NUMS,K,VDT2.BITPOS,VDT2.LENG);
                                COEF := COEFTBLE[8] ;
                                bbv := bbv*coef.slope + coef.offset ;
                                XVALU := BBV - BBI*0.5 ;
                                IF J=6 THEN XVALU := XVALU/BBI ;
                             END ;
                           END; { OF CASE }
                           STR(XVALU:10:4,SORC);
                      END;
              PUTOUT(SORC,FRAME[VDT.SCRPOS],GREEN);
TheExit:            END;
        END;
END;


procedure main ;
begin
     port21 := port[*21] ;
     port[*21] := port21 or 1 ;
    numsaddr[0] := seg(nums[0]) ;
    numsaddr[1] := ofs(nums[0]) ;
```

```
{ CHECK FOR SCREEN DUMP }
{ IF STATPR = 1 THEN
REPEAT
    SCRDUMP(LINENUM,CHARNUM);
UNTIL STATPR = 0 ;}
      STR(TIMER:4,SORC);
      SORC := 'TIMER = ' + SORC ;
      PUTOUT(SORC,FRAME[1220],WHITE);


    if needmoredata then
    begin
       C:= 'ENTER [WD,0,301]' ;   { set up for DMA transfer of 604 bytes }
       IE488(C,numsad,F,B);     { input 302 words of data input v array }
       needmoredata := FALSE ;
{      START := TIME ;}
    end
    else
    begin
{           COUNT := COUNT + 1 ;}
         C:= 'REQUEST' ;
         STATUS[0] := 0 ;
         IE488(C,STAT,F,B);
         if ((STATUS[0] AND $200) = 0) then
         begin
{              STOP := TIME ;}
{              DELTA := STOP - START ;}
{              WRITELN(' ITERATIONS = ', COUNT:5,'  TIME =',DELTA:10:5);}
{              COUNT := 0 ;}
              FOR I := 1 TO 302 DO
              BEGIN
                   J := 2*I ;
                   TEMP := BYTES[J] ;
                   BYTES[J] := BYTES[J+1] ;
                   BYTES[J+1] := TEMP ;
              end;
              displaydata ;              needmoredata := true ;
         end;
      end;
port[$21] := port21 ;   { restore interrupt mask for 8259 }
end;


procedure INTieee;
begin
inline(   $FB/       { STI   ENABLE INTERRUPTS }
          $50/       { PUSH AX }
          $53/       { PUSH BX }
          $51/       { PUSH CX }
          $52/       { PUSH DX }
          $56/       { PUSH SI }
          $57/          { PUSH DI }
          $1e/          { PUSH DS }
          $06/          { PUSH ES }
          $55           { PUSH BP }
          );
INLINE($2E/$C5/$3E/DATSEG);  { SET DS REG TO DATA SEG }
inline(
          $1e/           { push ds }
          $07 ) ;        { pop es }  { turbo ds & es are same }
inline($2e/$89/$26/oldstpt); { save old stack pointer }
INLINE($2E/$8B/$26/STPT);   { SET STACK POINTER }
inline($2e/$8c/$16/oldstseg); { save old stack seg }
INLINE($2E/$8E/$16/STSEG); { SET STACK SEGMENT REG }
MAIN ;   { CALL MAIN PROCEDURE }
inline($2e/$8b/$26/oldstpt);  { restorr old stack pointer }
inline($2e/$8e/$16/oldstseg); { restore old stack segment }

inline(   $5d/            { POP BP }
          $07/            { POP ES }
          $1f/            { POP DS }
          $5f/            { POP DI }
          $5e/            { POP SI }
          $5a/            { POP DX }
          $59/            { POP CX }
          $5b/            { POP BX }
          $58/            { POP AX }

          $cf             { IRET   }
          ); { RETURN TO 4TH }

end;




{ ------------------PAS4TH MAIN PROGRAM--------------------- }


  BEGIN
COUNT := 0 ;
newscreen ;
displayacro ;
needmoredata := true ;
  f := 1;
  b := 0;
STORINT[0] := SCRNINT[0] ;   {SAVE PRINT SCREEN VECTOR }
STORINT[1] := SCRNINT[1] ;
SCRNINT[0] := OFS(PRSTAT) ;
SCRNINT[1] := CSEG ;
STATPR := 0 ;
LINENUM := 0 ;
```

```
CHARNUM := 0 ;

{ get coefficient file name }

writeln(' enter coefficient file name (usually coef.dat)');
readln(coefFileNam);
{coefFILEnam := 'coef.dat' ;}
assign(coefFILE,COEFfileNAM);
RESET(COEFFILE);

{ read in coefficients }
FOR I:= 1 TO 50 DO
READ(COEFFILE,COEFTBLE[I]);
CLOSE(COEFFILE) ;

{ get variable definition table file name }
WRITELN(' ENTER VARIABLE DEFINITION FILE NAME (USUALLY VARDEF.DAT)');
READLN(VDTFILENAM);
ASSIGN(VDTFILE,VDTFILENAM);
RESET(VDTFILE);
{ read in variable definition table }
FOR I:= 1 TO 200 DO
READ(VDTFILE,VTBLE[I]);     { READ IN THE VARIABLE DEFINITION TABLE }
CLOSE(VDTFILE) ;

{ set up ieee-488 bus. my address = 3 (MAD=3)
  computer in charge= 1,  number of ieee-488 cards = 1,
  base address for ieee card = 200 hex }
  syscon := 'SYSCON MAD=3, CIC=1, NOB=1, BA0=&H200';
  v := '                                     ';
{ send initialization command contained in string syscon }
  IE488(syscon,v,f,b);
  f :=2;
  b :=0;
  C:= 'PASCTL 0' ;
{  writeln('PASSING CONTROL TO HP'); }
{ need to send control to HP-1000 }
  IE488(c,v,f,b);
F:= 0;
B:= 0 ;
C := 'TIMEOUT' ;
V := chr(1) ;
{ set up for infinite time out value }
IE488(C,V,F,B);

  ESSEG := ESEGM ;
  dataseg[0] := DSEG ;
  DATASEG[1] := DSEG ;
  WRITELN(' ESEG & DESG =',ESSEG:6,DATASEG[0]:6);
  STACKSEG := SSEG ;
  STACKPT := STACK ;
  csegm := cseg ;
  offs := ofs(INTieee)+ 7 ; {  THE + 7 SKIPS OVER TURBO PROCEDURE CODE }
  SETINTVEC(CSEGM,OFFS);
  writeln('    PAS4TH  CS,OFS :',intvec[0]:6,intvec[1]:6);
{  writeln('    Datseg =',datseg[0]:6,datseg[1]:6); }
{  port[$208] := 1 ; }

{ interrupt type 27 hex allows a program to terminate while locking
  itself in memory. This main program is never re-entered, but interrupt
  type 48 hex will cause the main procedure to be called which in turn
  utilizes the rest of this program code }
  intr($27,zilch);
  END.
```

Standard Bibliographic Page

1. Report No. NASA CR-178339	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle HALOE TEST AND EVALUATION SOFTWARE		5. Report Date June 1987
		6. Performing Organization Code
7. Author(s) W. Edmonds S. Natarajan		8. Performing Organization Report No.
9. Performing Organization Name and Address ST Systems Corporation (STX) 28 Research Drive Hampton, VA 23666		10. Work Unit No. 678-12-04-09
		11. Contract or Grant No. NAS1-18022
12. Sponsoring Agency Name and Address NASA Langley Research Center Hampton, VA 23665		13. Type of Report and Period Covered Contractor Report
		14. Sponsoring Agency Code
15. Supplementary Notes Langley Technical Monitor: R. L. Baker		

16. Abstract

Computer programming, system development and analysis efforts during this contract were carried out in support of the Halogen Occultation Experiment (HALOE) at NASA/Langley. Support in the major areas of data acquisition and monitoring, data reduction and system development are described along with a brief explanation of the HALOE project. Documented listings of major software are located in the appendix.

17. Key Words (Suggested by Authors(s)) HALOE, UARS, ozone depletion, software development, system development, data acquisition data monitoring, Forth, Pascal	18. Distribution Statement Unclassified - Unlimited Subject Category 61		
19. Security Classif.(of this report) Unclassified	20. Security Classif.(of this page) Unclassified	21. No. of Pages 142	22. Price

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA Langley Form 63 (June 1985)