27
3/23/79
4 copy to NTIS

MASTER

BDX-613-2085 (Rev.)

**AUTOMATION OF A THERMAL
EXPANSION INSTRUMENT**

By L. L. Holland

Published March 1979

Topical Report
D. S. Genser, Project Leader

Prepared for the United States Department of Energy
Under Contract Number DE-AC04-76-DP00613.

**Kansas City
Division**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America

Available From the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, Virginia 22161.

Price:	Microfiche	\$3.00
	Paper Copy	\$6.50

**AUTOMATION OF A THERMAL
EXPANSION INSTRUMENT**

By L. L. Holland

Published March 1979

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Topical Report
D. S. Genser, Project Leader

Project Team:
J. S. Beshoner
R. L. Davis
L. L. Holland
D. E. Lacey

Communications Services

**Kansas City
Division**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

AUTOMATION OF A THERMAL EXPANSION INSTRUMENT

BDX-613-2085 (Rev.), Topical Report, Published March 1979

Prepared by L. L. Holland

Automation of a thermal expansion instrument using a minicomputer system and with analog-to-digital converter inputs and flip-flop relay outputs is complete. The necessary hardware link is connected and the software developed to allow equipment control, data acquisition, data reduction, and report generation by the minicomputer. The design of the automation allows non-programmers to run the experiment, reduce the data, and generate the report.

FA:TR/7

This report was prepared as an account of work sponsored by the United States Government. Neither the United States, nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, expressed or implied or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

The Bendix Corporation
Kansas City Division
P. O. Box 1159
Kansas City, Missouri 64141

A prime contractor with the United States
Department of Energy under Contract Number
DE-AC04-78-0P00613

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

CONTENTS

Section	Page
SUMMARY	9
DISCUSSION	11
SCOPE AND PURPOSE.	11
PRIOR WORK	11
ACTIVITY	11
<u>Experimental Equipment and Operating Procedure</u> . . .	11
<u>Data Acquisition</u>	12
<u>Data Reduction and Plotting.</u>	13
ACCOMPLISHMENTS.	15
FUTURE WORK	16
APPENDICES	
A. INITIATION, EQUIPMENT CONTROL, AND DATA ACQUISITION SOFTWARE LISTINGS.	17
B. DATA REDUCTION AND PLOTTING SOFTWARE LISTINGS. . .	65
C. SAMPLES OF INTERACTIVE I/O AND OTHER OUTPUTS . . .	111
DISTRIBUTION	123

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

ILLUSTRATIONS

Figure		Page
1	Minicomputer System with Analog-To-Digital Converter Inputs and Flip-Flop Relay Outputs (P-102063)	12
2	Thermal Expansion Instrument (P-102065). . . .	13
3	Macro Flow Chart for Equipment Control	14

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

SUMMARY

The coefficient of linear thermal expansion over a temperature range is a materials property measurement that is routinely required at Bendix. The technique being used to determine the coefficient consists of subjecting the material to a range of temperatures. The expansion or contraction relative to a reference temperature is measured while the temperature is changed at a uniform rate.

The necessary hardware link between a minicomputer system and a thermal expansion instrument is designed and connected. The software that controls the temperature, records data, reduces data, and generates the report is written and functioning properly. The temperature is controlled with liquid nitrogen; the rate of flow is under program control. Data are recorded and stored on a disc at intervals specified by the operator. The recorded data include sample temperature, heater temperature, LVDT voltage, heater voltage, and the number of quarter turns the liquid nitrogen valve is open. The data reduction software is initiated by the operator, at which time the options desired for the particular application are entered. The minicomputer then outputs the data, in tabular or graphical form, and a completed test laboratory report cover sheet. The data is then archived on microfiche.

The manual method for running a typical two thermocycle test requires 30 labor-hours to complete. The automated test requires between 1 and 2 labor hours to complete, which reduces labor costs at least 15 times. Turnaround time is also greatly reduced, because more than one-third of the time required for the manual test is spent in data reduction which is completed in minutes by the computer system.

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

DISCUSSION

SCOPE AND PURPOSE

The objective of this portion of the Computerized Laboratory and Diagnostic Services (CLADS) project is to automate a thermal expansion instrument using a minicomputer system with analog-to-digital (A/D) converter inputs and flip-flop relay outputs. Functions to be automated are control of the liquid nitrogen valve and solenoid for temperature control, data acquisition, data reduction, and report generation.

Thermal expansion testing performed manually requires approximately 30 labor-hours to test each sample.

PRIOR WORK

Without the automated version of the thermal expansion test, manual control of the equipment and recording of data are necessary. The typical test requires 19 hours of testing time, during which the temperature and length change of the sample are constantly monitored with voltmeters. Data are manually recorded; upon completion of data acquisition, data reduction and plotting also are performed manually.

ACTIVITY

Experimental Equipment and Operating Procedure

The equipment used to determine the coefficient of linear thermal expansion consists of a minicomputer system with A/D converter inputs and flip-flop relay outputs (Figure 1) and a thermal expansion instrument (Figure 2). Sample and heater temperatures are monitored using two type K thermocouples.

In determining the coefficient of linear thermal expansion of a rigid solid using the thermal expansion instrument, a vitreous silica dilatometer tube is used. One thermocouple is placed on the surface of the 50-mm-long sample and the other is placed in a hole in the center of the sample. The sample and thermocouples are placed in the dilatometer tube, which is then placed in the vacuum chamber of the instrument. The temperature is cycled from room temperature down through a specified low temperature, up through a specified high temperature, and back to room temperature. This constitutes one thermocycle. The process can be reversed so that the sample is cycled through the high temperature prior to reaching the lower temperature. Sample temperature and length

Figure 1. Minicomputer System with Analog-To-Digital Converter Inputs and Flip-Flop Relay Outputs

change must be monitored at all times during the test. A typical test of one sample consists of two thermocycles with a temperature range from approximately -60 to $+150^{\circ}\text{C}$. A macro flow chart showing the interaction for equipment control of the minicomputer system and thermal expansion instrument is shown in Figure 3.

Data Acquisition

When the equipment and sample are stable at room temperature, the operator starts the initialization procedure software. This sets initial values of the parameters necessary to run the data acquisition and equipment control software. Two such parameters are the time interval between data readings and the number of quarter turns the liquid nitrogen valve is initially open.

Figure 2. Thermal Expansion Instrument

The software is finished recording data when the reference temperature has been reached once more than the number of thermocycles required (usually two). At this point, the equipment control software closes the liquid nitrogen solenoid and halts. The data acquisition software also halts, at which time data reduction can begin.

A listing of the initiation, equipment control, and data acquisition software, written partially in Fortran IV and partially in MACRO-11 assembly language is given in Appendix A.

Data Reduction and Plotting

The data reduction software offers the operator several options for examining and displaying the data. Some of these options allow the operator access to the data prior to the termination of

Figure 3. Macro Flow Chart for Equipment Control

the data acquisition portion of the test. The initial values of the test parameters can be examined to ensure that they were entered properly, and any portion of the existing data can be examined at any time during the experiment. This is important so that errors in the data may be discovered as soon as possible, instead of at the end of the test. For example, if a thermocouple breaks, the temperature recorded will always be 32°F (0°). This can be detected, the problem corrected, and the experiment restarted without waiting for the original data acquisition to finish. If, upon examining the data, it is necessary to change the time interval between readings, the data reduction software offers this option.

Upon completion of data acquisition, data reduction session can begin. All the data can be printed in tabular form either on the line printer or on the initiating terminal. All data between two specified data records and the initialization data also can be printed. In addition the data can be archived for permanent storage on microfiche.

The plotting software graphs the percent expansion or contraction relative to a reference temperature versus temperature. The options the operator chooses from as to the final form of the graph are:

- To plot the data corrected or uncorrected for thermocouple lead-lag;

- To change the default values of the scale factors for the percent expansion axis; and

- To calculate and print on the graph a table of coefficients of linear thermal expansion between temperatures specified by the operator.

The plotting program optionally prints a Test Laboratory Report cover sheet that is complete with all required information except the approval signature.

Software for data reduction and plotting, written in Fortran IV and MACRO-11 is given in Appendix B. Copies of the interactive I/O, along with other sample outputs, are given in Appendix C.

ACCOMPLISHMENTS

The test to determine the coefficient of linear thermal expansion is fully automated with the minicomputer system. The hardware and software necessary to provide the capability to run the test with the computer system controlling the equipment, recording and

reducing data, and generating a report are developed. Labor costs have been reduced from 30 labor-hours per sample tested to between 1 and 2 labor-hours per sample tested. Turnaround time is also greatly reduced, because no clerical help is needed to complete the test and because elapsed time for data reduction is a matter of minutes instead of hours.

As a result of the thermal expansion automation, several software modules for instrumentation are now available for use in automating other equipment. For example, an analog/digital (A/D) converter reader module is available that will read an A/D channel. Reading A/D channels is a part of every test instrument automation; thus, the availability of this and other modules will reduce software costs in future automation projects.

FUTURE WORK

The labor costs for running the automated version of the thermal expansion test are not likely to be reduced by further work. The bulk of the labor time required to run this test is used in preparing the sample and stabilizing the equipment at room temperature.

Appendix A

INITIATION, EQUIPMENT CONTROL, AND DATA ACQUISITION SOFTWARE LISTINGS

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

```

0001      PROGRAM THMEXP
           .IDENT/B42470/
C
C+
C      MARCH 1978                      RON DAVIS
C
C      THERMAL EXPANSION TASK INITIATION PROGRAM.
C
C      EDITED 3/13/78 - JSB
C      INITIAL READINGS ARE ALWAYS TAKEN AT 1 MINUTE INTERVALS
C      CHANGE TO LOOK AT SPECIMEN TEMPERATURE INSTEAD OF
C      HEATER FOR START OF CYCLE
C      CHANGED CALL TO SPLIT TO GET TRUE GAIN
C      CHANGED CALL TO TEMP TO USE MILLIVOLTS
C
C      3/15/78 - JSB - CHANGED TIME CONVERSION TO GIVE HOURS & MINUTES
C      CHANGED RECORD SIZE TO 8 WORDS/RECORD
C
C      4/17/78 - LLH- INCLUDED QUESTIONS FOR CHARGE NUMBER, PERSON
C      REQUESTING TEST, TEST SPECIFICATION, AND
C      MATERIAL. THESE QUANTITIES ARE WRITTEN TO
C      A FILE DK1:[300,300]LTR#ID.HDR
C
C      4/24/78 - LLH - CHANGED INITIAL LN2QTR TURNS OPEN
C      MINIMUM FROM 2 TO 1
C-
0002      LOGICAL*1 SPECID(2),EMPND(6),TIME(2),DATE(3),LTR(7)
0003      LOGICAL*1 NCYCLE,ISWING,CHAN(4),ITURNS,INTRVL,RELAY(3)
0004      LOGICAL*1 CYCLE,IQUAD,R(64),FILE(28),EXT(4),EXT2(4)
0005      LOGICAL*1 TSTSPC(12),MATERL(12),CHARG(8),REQDBY(14)
0006      LOGICAL*1 FILE2(28),FILE3(22)
C
0007      INTEGER*2 ISETS,IRESLT(4),ICHAN,RELAYZ
C
0008      REAL*4      SLENGT,STEMP,TEMPLO,TEMPI,RATIO,TASK
C
0009      EQUIVALENCE (R(1),LTR),(R(8),SPECID),(R(18),EMPND)
0010      EQUIVALENCE (R(16),TIME),(R(18),DATE),(R(21),SLENGTH)
0011      EQUIVALENCE (R(25),STEMP),(R(29),NCYCLE),(R(38),ISWING)
0012      EQUIVALENCE (R(31),TEMPLO),(R(35),TEMPI),(R(39),CHAN)

```

```

0013      EQUIVALENCE (R(43),RELAY),(R(46),ITURNS),(R(47),INTRVL)
0014      EQUIVALENCE (R(48),CYCLE),(R(49),RATIO),(R(53),ISETS)
0015      EQUIVALENCE(FILE3,FILE(5))
      C
      C
0016      DATA TASK /GRTE2DAC/
0017      DATA FILE /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H1,1H2,1H3,1H4,
1      1H0,1H1,15*0/
0018      DATA EXT /1H.,1HT,1HE,1HZ/
0019      DATA EXT2 /1H.,1HH,1HD,1HR/
      C
      C
      C      CHARGE NUMBER
      C      WRITTEN TO LTR*ID.HDR
      C
0020      WRITE(1,1001)
0021 1001  FORMAT('ENTER CHARGE NUMBER (8): ')
0022      READ(1,1002)NCHARG,CHARG
0023 1002  FORMAT(0,8A1)
      C
      C      REQUESTED BY
      C      WRITTEN TO LTR*ID.HDR
      C
0024      WRITE(1,1003)
0025 1003  FORMAT('ENTER 1ST INITIAL & LAST NAME OF PERSON REQUESTING'/
1      '$TEST (14):')
0026      READ(1,1004) NREQD,REQDBY
0027 1004  FORMAT(0,14A1)
      C
      C      DEPARTMENT OF PERSON REQUESTING TEST
      C      WRITTEN TO LTR*ID.HDR
      C
0028      WRITE(1,1011)
0029 1011  FORMAT('ENTER DEPT # OF PERSON REQUESTING TEST (4): ')
0030      READ(1,1012)IDEPTN
0031 1012  FORMAT(I4)
      C
      C      MATERIAL
      C      BYTE(12)
      C      WRITTEN TO LTR*ID.HDR
      C
0032      WRITE(1,1006)
0033 1006  FORMAT('ENTER MATERIAL TO BE TESTED (12): ')
0034      READ(1,1007)NMATRL,MATERL

```

```

C
C   TEST SPECIFICATION  BYTE(12)
C   WRITTEN TO LTR*ID.HDR
C
0035   WRITE(1,1005)
0036 1005   FORMAT('ENTER TEST SPECIFICATION, IF ANY (12): ')
0037   READ(1,1007) NTSTSP,TSTSPC
0038 1007   FORMAT(Q,12A1)
C
C
C   LTR NUMBER          BYTE(7)
C
0039   WRITE (1,1008)
0040 1008   FORMAT ('ENTER LTR* (7) : ')
0041   READ (1,1010) NLTR,LTR
0042 1010   FORMAT(Q,7A1)
C
C   SPECIMEN ID        BYTE(2)
C
0043   WRITE (1,1020)
0044 1020   FORMAT ('ENTER SPECIMEN ID (2) : ')
0045   READ (1,1030) (SPECID(I),I=1,2)
0046 1030   FORMAT (2A1)
C
C   EMPLOYEE NUMBER     BYTE(6)
C
0047   WRITE (1,1040)
0048 1040   FORMAT ('ENTER EMPLOYEE NUMBER (6) : ')
0049   READ (1,1050) (EMPNO(I),I=1,6)
0050 1050   FORMAT (6A1)
C
C   STARTING CYCLE NUMBER      (REAL)
C
0051   WRITE (1,1060)
0052 1060   FORMAT ('ENTER REFERENCE TEMPERATURE (F.P.) : ')
0053   READ (1,1070) STMP
0054 1070   FORMAT (F7.2)
C
C   SPECIMENT INITIAL LENGTH    (REAL)
C
0055   WRITE (1,1080)
0056 1080   FORMAT ('ENTER SPECIMEN'S INITIAL LENGTH (F.P.) : ')
0057   READ (1,1090) SLENGT

```



```

0058 1090 FORMAT (F8.3)
      C
      C NUMBER OF THERMAL CYCLES          (BYTE)
      C
0059      WRITE (1,1100)
0060 1100 FORMAT ('$NUMBER OF THERMAL CYCLES (1-127) : ')
0061      READ (1,1110) NCYCLE
0062 1110 FORMAT (I3)
      C
      C CYCLE DIRECTION SWING            (BYTE)
      C
0063      WRITE (1,1120)
0064 1120 FORMAT ('$DIRECTION OF TEMP CYCLE (1=UP,0=DOWN) : ')
0065      READ (1,1130) ISWING
0066 1130 FORMAT (I1)
      C
      C LOW TEMPERATURE                  (REAL)
      C
0067      WRITE (1,1140)
0068 1140 FORMAT ('$MINIMUM LOW TEMPERATURE (F.P.) : ')
0069      READ (1,1070) TEMPL0
      C
      C HIGH TEMPERATURE                 (REAL)
      C
0070      WRITE (1,1160)
0071 1160 FORMAT ('$MAXIMUM HIGH TEMPERATURE (F.P.) : ')
0072      READ (1,1070) TEMPH1
      C
      C A/D STARTING CHANNEL             (BYTE(4))?
      C
0073 100  WRITE (1,1180)
0074 1180 FORMAT ('$STARTING A/D CHANNEL NUMBER (0-68) : ')
0075      READ (1,1190) ICHAN
0076 1190 FORMAT (I2)
0077      IF (ICHAN .LT. 0 .OR. ICHAN .GT. 68) GOTO 100
0079      CHAN (1) = ICHAN
0080      CHAN (2) = ICHAN + 1
0081      CHAN (3) = ICHAN + 2
0082      CHAN (4) = ICHAN + 3
      C
      C RELAY STARTING CHANNEL (BYTE(3))
      C
0083 150  WRITE (1,1200)

```

```

0084 1200 FORMAT ('$STARTING RELAY (0-12 OR 16-20) : ')
0085 READ (1,1190) IRELAY
0086 IF (IRELAY .LT. 0 .OR. IRELAY .GT. 20) GOTO 150
0088 IF (IRELAY .GT. 12 .AND. IRELAY .LT. 16) GOTO 150
0090 RELAY (1) = IRELAY
0091 RELAY (2) = IRELAY + 1
0092 RELAY (3) = IRELAY + 2
C
C LN2 VALVE INITIAL POSITION (QTR TURNS 0-20) (BYTE)
C
0093 200 WRITE (1,1220)
0094 1220 FORMAT ('$INITIAL LN2 VALVE POSITION IN QTR TURNS (1-20) : ')
0095 READ (1,1190) Iturns
0096 IF (Iturns .LT. 1 .OR. Iturns .GT. 20) GOTO 200
C
C MINUTES BETWEEN READING INTERVALS (BYTE)
C
0098 WRITE (1,1240)
0099 1240 FORMAT ('$NUM MINUTES BETWEEN READINGS (1-127) : ')
0100 READ (1,1110) INTRVL
C
C LVDT/MV RATIO (REAL)
C
0101 WRITE (1,1260)
0102 1260 FORMAT ('$LVDT/MV RATIO (F.P.) : ')
0103 READ (1,1090) RATIO
C
C DATE MONTH, DAY, YEAR (BYTE (3))
C
0104 CALL IDATE (1,J,K)
0105 DATE (1) = I
0106 DATE (2) = J
0107 DATE (3) = K
C
C TIME HOUR, MINUTE (BYTE (2))
C
0108 SEC = SECNDS (0.0)
0109 MINUT = SEC / 60.
0110 TIME(1) = MINUT / 60 !TIME OF DAY - HOURS
0111 TIME(2) = MINUT - TIME(1)*60 !TIME OF DAY - MINUTES
C
C CYCLE NUMBER (BYTE)
C

```

```

0112      ISETS = 0
      C
      C
      C CREATE LTR FILE 'LTR*ID.TE2'
      C WHERE 'ID' IS THE SPECIMEN ID NAME.
      C AND LTR* IS THE 6 OR 7 DIGIT LTR NUMBER
      C EACH RECORD IS 8 WORDS LONG
      C LUN IS 2. INITIAL LENGTH IS 4 RECORDS FOR THE ABOVE INFO.
      C
0113      DEFINE FILE 2 (4,8,U,1REC)
0114      DO 250 I = 1,NLTR
0115      250  FILE(I+13) = LTR (I)
0116      FILE (NLTR+14) = SPECID (1)
0117      FILE (NLTR+15) = SPECID (2)
0118      DO 255 I=1,NLTR+15
0119      FILE2(I)=FILE(I)
0120      255  CONTINUE
0121      DO 260 I = 1,4
0122      FILE2(NLTR+15+I)=EXT2(I)
0123      260  FILE (NLTR+15+I) = EXT (I)
      C
      C      WRITE SEPARATE HEADER FILE
      C
      C      FILE2 HAS NLTR+19 CHARS IN ITS NAME.
0124      CALL ASSIGN(4,FILE2,NLTR+19)
0125      WRITE(4,1301)
0126      1301  FORMAT(' N1  N2  N3  N4')
0127      WRITE(4,1302)NCHARG,NREDD,NTSTSP,NMATRL
0128      1302  FORMAT(4I6)
0129      WRITE(4,1303)
0130      1303  FORMAT(' CHARGE *, REQUESTED BY, IDEPTH, TEST SPEC., MATERIAL')
0131      WRITE(4,1304)(CHARG(I),I=1,NCHARG)
0132      1304  FORMAT(14A1)
0133      WRITE(4,1304)(REQDBY(I),I=1,NREDD)
0134      WRITE(4,1306)IDEPTH
0135      1306  FORMAT(14)
0136      WRITE(4,1304)(TSTSPC(I), I=1,NTSTSP)
0137      WRITE(4,1304)(MATERL(I),I=1,NMATRL)
0138      CALL CLOSE(4)
      C
0139      CALL ASSIGN(2,FILE)
0140      CALL FDBSET(2,'NEW')
0141      IREC=1

```

```

C
C
C WRITE HEADER AS 4 SEQUENTIAL RECORDS
C 'R' IS HEADER.
C
0142      LL = 1
0143      DO 300 I =1,4
0144      WRITE (2, IREC) (R(I),L=LL,LL+15)
0145      LL = LL + 16
0146      300 CONTINUE
C
C
0147      CALL CLOSE (2)
C
C
C
C      CHANGE TO HIGHER PRIORITY
C
0148      CALL ALTPRI(,200.)
C
C
C      OPEN SOLENOID, USING BESHNER'S SUBROUTINE
C      WHERE RELAY(2) IS THE LN2 SOLENOID RELAY NUMBER
C      AND 3 IS THE      LUN
C
0149      RELAY2=IRELAY+1
C
0150      CALL CTRLSD(RELAY2,3)
C
C NOW BEGIN TO READ THE TEMPERATURE
C
0151      WRITE (1,2000)
0152      2000 FORMAT ('PRESS RETURN WHEN READY TO START : ')
0153      READ (1,1030)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C UPDATE # OF DATA SETS
C
0154      400 DEFINE FILE 2 (4,8,U,IREC)
0155      CALL ASSIGN (2,FILE)

```

```

0156      CALL FDBSET (2,'OLD')
0157      IREC = 4
0158      READ (2'IREC) (R(1),I=49,64)
      C
0159      ISETS = ISETS + 1
0160      IREC = 4
0161      WRITE (2'IREC) (R(1),I=49,64)
0162      CALL CLOSE (2)
      C
      C OPEN FOR CORRECT * OF RECORDS
      C
0163      NREC = ISETS + 4
0164      DEFINE FILE 2 (NREC,8,0,IREC)
0165      CALL ASSIGN (2,FILE)
0166      CALL FDBSET (2,'OLD')
      C
      C GET TIME
      C
0167      SEC = SECNDS (0.0)
0168      MINUT = SEC /60.
0169      TIME(1) = MINUT / 60          !TIME OF DAY - HOURS
0170      TIME(2) = MINUT - TIME(1)*60 !TIME OF DAY - MINUTES
      C
      C READ THE A/D CHANNELS
      C LUN IS 3
      C
0171      CALL ADREAD (ICHAN,4,IRESLT,IERR,3)
0172      IF (IERR. NE. 0) STOP 'A/D ERR'
      C
      C ZERO QUADRANT LOCATION NUM FOR THE CYCLE
      C USE INITIAL * TURNS FOR LN2
      C
0174      IQUAD = 0
0175      IREC = NREC
0176      IR = IRESLT(1)              !REVERSE HEATER & SPECIMEN VALUES
0177      IRESLT(1) = IRESLT(2)
0178      IRESLT(2) = IR
0179      WRITE (2'IREC) TIME,IRESLT,ITURNS,IQUAD
0180      CALL CLOSE (2)
      C
      C CALCULATE THE VOLTAGE FOR SPECIMEN TEMP FOR A/D VALUE AND GAIN
      C
0181      CALL SPLIT (IRESLT(2),IADV,GAIN)

```

```

0182      FIADV = IADV
0183      RMV = 4.8852 * FIADV / GAIN
      C
      C CALCULATE TEMP IN DEGREES F.
      C
0184      CALL TEMPF (RMV,T)
      C
      C DETERMINE IF PAST START TEMPERATURE. IF NOT,
      C WAIT UNTIL NEXT INTERVAL AND CHECK AGAIN.
      C
0185      IF (ISWING .EQ. 1 .AND. T .GE. STEMP) GOTO 500
0187      IF (ISWING .EQ. 0 .AND. T .LE. STEMP) GOTO 500
      C
      C WAIT 1 MINUTE FOR NEXT READING
      C
0189      WRITE (1,2400) T
0190 2400 FORMAT (' WAITING FOR NEXT READING. SPEC TEMP = ',F6.2)
0191      CALL WAIT (1,3,IDS)
      C WAIT ONE MINUTE FOR NEXT READING
0192      GOTO 400
      C
      C
0193 500 CALL REQUES (TASK)
0194      CALL SEND (TASK,FILE3)
0195      CALL EXIT
0196      END

```

ADREAD MACRO M1110 05-JUL-78 09:20 PAGE 1

```

1          .LIST TTM
2
3          .TITLE ADREAD
4          .IDENT/031070/
5
6          ;+
7          MARCH 1978
8          ;-
9          .MCALL QIOS,DIR$,UTSE$$
10
11          ;
12          ; CALL ADREAD (ISTART,NUM,IRESLT,IERR,IUNIT)
13          ;
14          ; ISTART - FIRST A/D CHANNEL NUM (0-71)
15          ; NUM - NUMBER OF CONSECUTIVE CHANNELS
16          ; IRESLT - BUFFER , LENGTH = TO NUM OF CHAN
17          ; ON INPUT WILL HAVE GAIN IN BITS 12-15
18          ; ON OUTPUT WILL HAVE GAIN AND AD VALUE IN BITS 0-11
19          ; IERR - 0 = NO ERROR
20          ; ELSE, RETURNS ICS ERROR CODE
21          ; IUNIT - LUN 1 TO 16
22          ;
23          ; LOCAL DATA
24          ;
25          000000 UNDFLG: .BLKW 1
26          000002 NCHAN: .BLKW 1
27          000004 LUN: .BLKW 1 ;QIO LUN
28          000006 CNTRL: .BLKW 72. ;QIO CHANNEL CONTROL BUFFER
29          000226 DATA: .BLKW 72. ;QIO DATA BUFFER A/D VALUES
30          000446 IOSB: .BLKW 2 ;STATUS BLOCK
31          ;
32          000452 ADDIO: QIOS 10,RBC,0,1,,IOSB,,<DATA,0,CNTRL>
33          ;
34          000000 G.1=0
35          000001 G.2=1
36          000004 G.10=4
37          000005 G.20=5
38          000010 G.50=10
39          000011 G.100=11
40          000014 G.200=14

```

```

41      000015  G.1000-15
42      ;EVENT FLAG = 1
43      ;
44      ;
45 000502      ADREAD::
46 000502      005725      TST      (R5)+
47 000504      017500      MOV      @ (R5),R0      ;FIRST CHANNEL NUM
48      000000
49 000510      017567      MOV      @2(R5),NCHAN      ;NUM OF CHANNELS
50      000002
51      177264
52 000516      016503      MOV      4(R5),R3      ;# OF IRESLT BUFFER
53      000004
54      ;
55      ; FROM ISTART CHANNEL, LOOP THRU
56      ; IRESLT AND EXTRACT GAIN SETTINGS
57      ;
58 000522      012704      MOV      @CNTRL,R4      ;# CNTRL BUFFER
59      000006
60 000526      012302      1$:      MOV      (R3)+,R2      ;GET GAIN
61 000530      042702      BIC      @7777,R2      ;CLEAR GARGAGE
62      007777
63 000534      050002      BIS      R0,R2      ;OR WITH CHANNEL #
64 000536      010224      MOV      R2,(R4)+      ;STORE IN CNTRL BUFFER
65 000540      005200      INC      R0      ;NEXT CHANNEL
66 000542      005367      DEC      NCHAN      ;ANY MORE LEFT
67      177234
68 000546      001367      BNE      1$      ;BIF YES
69      ;
70      ; CALL A/D DRIVER
71      ;
72 000550      012703      MOV      @ADG10,R3      ;GET @DPB
73      000452
74 000554      017501      MOV      @2(R5),R1      ;NUM OF CHANNELS
75      000002
76 000560      006301      ASL      R1      ;BYTE LENGTH CNTRL BUFFER
77 000562      010163      MOV      R1,@.10PL+2(R3)      ;STORE IN DPB
78      000016
79 000566      017563      MOV      @8.(R5),@.10LU(R3)      ;GET LUN
80      000010
81      000004
82 000574      017567      MOV      @8.(R5),LUN      ;AND SAVE FOR LATER
83      000010

```



```

177202
72 000602      DIR$      R3      ;0DPB
73 000606 103411      BCS      2$      ;BIF ERROR
74 000610      WTSE$S    *1      ;EVENT FLAG 1
75 000622 126727      CMPB      10SB,*IS.SUC      ;OR
      177620
      000000C
76 000630 001405      BEQ      3$      ;BIF NO ERROR
77 000632 116701 2$:   MOVB      10SB,R1      ;GET ERROR NUMBER
      177610
78 000636 010175      MOV      R1,06(R5)      ;RETURN ERROR CODE
      000006
79 000642 000207      RTS      PC      ;AND EXIT
81      ;CHECK RESULTS FOR CORRECT GAIN
82      ;SETTINGS, IF WRONG, ADJUST GAIN AND REMEASURE.
83      ;
84 000644 012703 3$:   MOV      *DATA,R3      ;A/D VALUES BUFFER
      000226*
85 000650 017567      MOV      02(R5),NCHAN      ;NUM OF CHANNELS
      000002
      177124
86 000656 012704      MOV      *CNTRL,R4      ;GAIN VALUES BUFFER
      000006*
87      ;
88 000662 011401 4$:   MOV      (R4),R1      ;GET CHANNEL AND GAIN
89 000664 011300      MOV      (R3),R0      ;GET DATA READING
90 000666 005067      CLR      UNDFLG      ;EXTERNAL FLAG FOR CHKGAN
      177106
91      ;
92 000672 004767      JSR      PC,CHKGAN      ;ADJUST FOR NEW GAIN
      000112
93      ;INPUT R0= A/D DATA
94      ;INPUT R1= CURRENT GAIN/CHANNEL
95      ;RETURN R0= A/D DATA RIGHT JUST
96      ;RETURN R1= NEW GAIN/CHANNEL
97 000676 126727      CMPB      10SB,*IS.SUC      ;ERROR
      177544
      000000C
98 000704 001405      BEQ      5$      ;BIF NO
99 000706 116701      MOVB      10SB,R1      ;GET ERROR NUM
      177534
100 000712 010175      MOV      R1,06(R5)      ;RET WITH ERROR CODE
      000006

```

```

101 000716 000207      RTS      PC
102
103
104 000720 010023 5$:      MOV      R0,(R3)+      ;SAVE A/D IN DATA
105 000722 010124      MOV      R1,(R4)+      ;SAVE GAIN IN CNTRL
106 000724 005367      DEC      NCHAN      ;ANY MORE
      177052
107 000730 001354      BNE      4$      ;BIF YES
109      ; RETURN GAIN AND A/D VALUE IN IRESLT
110
111 000732 017567      MOV      02(R5),NCHAN      ;NUMBER OF CHANNELS
      000002
      177042
112 000740 012703      MOV      *DATA,R3      ;DATA BUFFER
      000226
113 000744 012704      MOV      *CNTRL,R4      ;CNTRL BUFFER
      000006
114 000750 016502      MOV      4(R5),R2      ;IRESLT ADDRESS
      000004
115
116 000754 012301 6$:      MOV      (R3)+,R1      ;GET A/D VALUE
117 000756 042701      BIC      *170000,R1      ;CLR GARBAGE
      170000
118 000762 012403      MOV      (R4)+,R0      ;GET GAIN
119 000764 042700      BIC      *7777,R0      ;CLR GARBAGE
      007777
120 000770 050001      BIS      R0,R1      ;COMBINE GAIN AND DATA
121 000772 010122      MOV      R1,(R2)+      ;STORE IN IRESLT
122 000774 005367      DEC      NCHAN
      177002
123 001000 001365      BNE      6$      ;BIF MORE
124 001002 005075      CLR      06(R5)      ;NO ERRORS
      000006
125 001006 000207      RTS      PC
127      ;SUBROUTINE CHKGAN
128
129      ; UPON ENTRY      R1 = CURRENT GAIN/CHANNEL
130      ;                  R0 = A/D VALUE
131
132      ; UPON EXIT      R1 = NEW GAIN AND CHANNEL
133      ;                  R0 = A/D VALUE
134
135      ; NOTE *** THIS IS A RECURSIVE SUBROUTINE

```

```

136
137 001010 010346 CHKGAN: MOV R3,-(SP) ;SAVE
138 001012 010446 MOV R4,-(SP)
139 001014 072027 ASH *-4,R0 ;RIGHT JUSTIFY FROM A/D
      177774
140 001020 010102 MOV R1,R2 ;EXTRACT GAIN
141 001022 072227 ASH *-12.,R2 ;SHIFT RIGHT
      177764
142 001026 042702 BIC *-177760,R2
      177760
143
144 001032 020027 CMP R0,*3777 ;POS SATURATION
      003777
145 001036 001452 BEQ OVFL ;BIF YES
146 001040 020027 CMP R0,*-4000 ;NEG SATURATION
      174000
147 001044 001447 BEQ OVFL ;BIF YES
148
149 ;CHECK FOR UNDER FLOW SATURATION
150 ;AND INCREASE GAIN
151 ;
152 001046 020227 CMP R2,*G.1000 ;MAX GAIN
      000015
153 001052 001521 BEQ C.EXIT ;BIF YES
154 001054 010003 MOV R0,R3 ;POSITIVE READING
155 001056 100001 BPL 1$ ;BIF YES
156 001060 005403 NEG R3 ;ABSOLUTE VALUE
157 001062 020227 1$: CMP R2,*G.200 ;GAIN OF 200
      000014
158 001066 001412 BEQ 5$ ;BIF YES
159 001070 020227 2$: CMP R2,*G.20 ;GAIN OF 20
      000005
160 001074 001004 BNE 3$ ;BIF NO
161 001076 020327 CMP R3,*019. ;LESS THAN 2/5 MAX
      001463
162 001102 003105 BGT C.EXIT ;BIF NO
163 001104 000412 BR INCGAN
164 001106 020227 3$: CMP R2,*G.2 ;GAIN OF 2
      000001
165 001112 001004 BNE 4$ ;BIF NO
166 001114 020327 5$: CMP R3,*3777/5 ;LESS THAN 1/5 MAX
      000631

```

```

167 001120 003076 BGT C.EXIT ;BIF NO
168 001122 000403 BR INCGAN
169
170 ;ALL OTHER GAINS ARE
171 001124 020327 4$: CMP R3,*3777/2 ;INCREMENTED BY 2 FOLD
001777 ;LESS THAN 1/2 MAX
172 001130 003072 BGT C.EXIT ;BIF NO
173
174 ;FIND NEXT HIGHEST GAIN SETTING
175
176 001132 010027 INCGAN: MOV R0,(PC)+ ;SAVE OLD RESULTS
177
178 001134 000000 OLDRSL: .WORD 0
179 001136 010127 MOV R1,(PC)+ ;SAVE OLD ICNTL
180 001140 000000 OLD CNT: .WORD 0
181 001142 010227 MOV R2,(PC)+ ;SAVE OLD GAIN
182 001144 000000 OLDGAN: .WORD 0
183 001146 004767 JSR PC,GETGAN ;0 OF GAIN TABLE IN R4
000206
184 001152 022424 CMP (R4)+,(R4)+ ;POSITION TO NEXT HIGHEST GAIN
185 001154 012402 MOV (R4)+,R2 ;GET GAIN CODE
186 001156 005267 INC UNDFLG ;SHOW UNDER SATURATION
176616
187 001162 000421 BR C.ATOD ;CONVERT
188
189 ;CHECK FOR MIN GAIN ON OVERFLOW
190 ;AND IF DUE TOO MUCH INCREASE IN
191 ;GAIN FROM LAST OPERATION
192
193 001164 005767 OVFL: TST UNDFLG ;WAS LAST AN UNDER SATURATION
176610
194 001170 001407 BEQ 4$ ;BIF NO
195 001172 016701 MOV OLD CNT,R1 ;RESTORE OLD ICNTL
177742
196 001176 016700 MOV OLDRSL,R0 ;RESTORE OLD VALUE
177732
197 001202 016702 MOV OLDGAN,R2 ;RESTORE OLD GAIN
177736
198 001206 000443 BR C.EXIT
199 001210 120227 4$: CMPB R2,*G.1 ;MINIMUM GAIN
000000
200 001214 001440 BEQ C.EXIT ;BIF YES
201 001216 004767 JSR PC,GETGAN

```

```

000136
202 001222 005744      TST      -(R4)      ;SKIP GAIN VALUE
203 001224 014402      MOV      -(R4),R2    ;GET NEXT LOWEST GAIN CODE
204
205      ;CALL A/D DRIVER
206
207 001226 072227 C.ATOD: ASH      *12.,R2    ;SET NEW GAIN
000014
208 001232 042701      BIC      *170000,R1    ;SAVE CHANNEL NUMBER
170000
209 001236 050201      BIS      R2,R1
210 001240 010167      MOV      R1,CNTRL2    ;SET UP CONTROL INFO
000110
211 001244 012703      MOV      *ADD102,R3    ;# OF DPB
001324
212 001250 016763      MOV      LUN,Q.10LU(R3) ;INSERT LUN
176530
000004
213      ;
214 001256      DIRS      R3      ;CALL A/D DRIVER
215 001262 103415      BCS      C.EXIT    ;BIF ERROR
216 001264      WTSE$$    *1      ;WAIT
217 001276 126727      CMPB     10SB,*15.SUC ;OK
177144
000000G
218 001304 001004      BNE      C.EXIT    ;BIF NO
219 001306 016700      MOV      DATA2,R0    ;NEW A/D VALUE
000044
220      ;R1 = GAIN AND CHANNEL
221 001312 004767      JSR      PC,CHKGAN    ;ENTER RECURSIVELY
177472
222 001316 012604 C.EXIT: MOV      (SP)+,R4    ;POP REG
223 001320 012603      MOV      (SP)+,R3
224 001322 000207      RTS      PC
225      ;
226      ;
227 001324      ADD102: D10$    10.RBC,B.1.,10SB.,<DATA2,2,CNTRL2>
228 001354      CNTRL2: .BLKW    1
229 001356      DATA2: .BLKW    1
230

```

```

232      ;SUBROUTINE GETGAN
233      :
234      ;R2 = CURRENT GAIN CODE
235      ;RETURNS 0 OF TABLE IN R4 FOR GAIN
236      :
237
238 001360      GETGAN::
239 001360      0127B4      MOV      *G.TBL,R4      ;START 0
                001376
240 001364      120214      1$:      CMPB      R2,(R4)      ;GAIN CODE MATCH
241 001366      001402      BEQ      2$      ;BIF YES
242 001370      022424      CMP      (R4)+,(R4)+      ;POSITION TO NEXT ENTRY
243 001372      000774      BR      1$
244 001374      000207      2$:      RTS      PC      ;EXIT
245
246 001376      000000      G.TBL:  .WORD      0,      1,      ;GAIN CODE AND GAIN VALUE
                001400      000001
247 001402      000001      .WORD      1,      2.
                001404      000002
248 001406      000004      .WORD      4,      10.
                001410      000012
249 001412      000005      .WORD      5,      20.
                001414      000024
250 001416      000010      .WORD      10,     50.
                001420      000062
251 001422      000011      .WORD      11,     100.
                001424      000144
252 001426      000014      .WORD      14,     200.
                001430      000310
253 001432      000015      .WORD      15,     1000.
                001434      001750
254
255      000001      .END

```

FORTRAN IV V01C-03C
CORE=00K, UIC=1,111

LED 05-JUL-78 08:21:01

PAGE 001
LP=TEMP

```

0001      SUBROUTINE TEMP(V,T)
          C
          C      THIS SUBROUTINE CONVERTS EMF (V) VOLTAGES FROM CHROMEL-ALUMEL
          C      THERMOCOUPLES TO TEMPERATURE IN DEGREES 'F' (T)
          C
0002      IF(V .GE. 0.0) GO TO 70          !SKIP IF POSITIVE EMF
0004      IF(V .GT. -.692) GO TO 30
0006      IF(V .GT. -1.445) GO TO 40
0008      IF(V .EQ. -1.445) GO TO 50
0010      T = 35.8618 + 49.7259*V          !EMF .LT. -1.445
0011      GO TO 160
0012  30    T = 32.8339 + 46.1652*V          ! 0.0 .GT. EMF .GE. -.692
0013      GO TO 160
0014  40    T = 32.9694 + 47.6831*V          ! -.692 .GT. EMF .GT. -1.445
0015      GO TO 160
0016  50    T = -36.0                      ! EMF .EQ. -1.445
0017      GO TO 160
0018  70    C1 = -1.853306E01
0019          C2 = 3.891834E01
0020          C3 = 1.664515E-02
0021          C4 = -7.870237E-05
0022          C5 = 2.283579E-07
0023          C6 = -3.570823E-10
0024          C7 = 2.993291E-13
0025          C8 = -1.284985E-16
0026          C9 = 2.2240E-20
0027          N = 8
0028          T = 512.
0029  100    E = C1 + C2*T + C3*T**2 + C4*T**3 + C5*T**4 + C6*T**5 + C7*T**6
0030          E = E + C8*T**7 + C9*T**8
0031          E = .001 * ( E + 125.*EXP( - ( ( (T-127.)/65. )**2 )/2. ) )
0032          IF(V .EQ. E) GO TO 150
0034          IF(V .GT. E) GO TO 130
0036          T = T - 2.*N                      !HERE IF V .LT. E
0037          GO TO 140
0038  130    T = T + 2.*N                      !HERE IF V .GT. E
0039  140    N = N - 1
0040          IF(N .NE. -5) GO TO 100
0042  150    T = 1.8*T + 32.                    !'C' TO 'F' CONVERSION
0043  160    RETURN
0044      END

```

FORTRAN IV V01C-03C... WED 05-JUL-78 08:21:04
CORE=08K, UIC=E1,113

PAGE 001
LP=TEMP

```
      C
      C
      C
      C
0001      SUBROUTINE TEMPC(V,T)
      C
      C      THIS SUBROUTINE CONVERTS EMF VOLTAGES ('V') FROM
      C      CHROMEL-ALUMEL THERMOCOUPLES TO TEMPERATURE IN DEGREES 'C'(T)
      C
0002      CALL TEMPF(V,T)      (FIRST CONVERT TO DEGREES F
0003      T = ( T - 32.)/1.8    NOW CONVERT TO DEGREES C
0004      RETURN
0005      END
```


CTRLSD MACRO MILLD 05-JUL-78 00:21 PAGE 1

```

1          .TITLE CTRLSD
2          .IDENT /050078/
3          ;+
4          ; ROUTINE TO OPEN SOLENOID. WRITTEN BY J. S. BESHONER
5          ; INCLUDED IN RON DAVIS'S PROGRAM TO INSURE THAT
6          ; THE SOLENOID IS OPENED PRIOR TO TAKING ANY DATA.
7          ;
8          ;-
9          .GLOBAL CTRLSD          ;PROGRAM ENTRY
10         ;
11         .MCALL QIOU$,DIR$
12 CTRLSD:
13         ;
14 000000 005725          TST      (R5)+          ;PAST FORTRAN GARBAGE
15 000002 013567 000110   MOV      @ (R5)+,ST.PWR          ;SOLENOID RELAY NUMBER
16 000006 013567 000050   MOV      @ (R5)+,ST.QIO+Q.IOLU          ; LUN
17 000012 116767 000044 000044   MOVSB  ST.QIO+Q.IOLU,ST.QIO+Q.IDEF          ;SET EFN
18 000020 016700 000072   MOV      ST.PWR,R0          ;DETERMINE RELAY BIT
19 000024 012705 000120'   MOV      *ST.PWR+2,R5
20 000030 004767 000112   CALL     ST,20
21 000034 012704 000072'   MOV      *ST.QIO+Q.IDPL,R4          ;QIO PARAMETER LIST POINTER
22 000040 012524          MOV      (R5)+,(R4)+          ;SET RELAY CARD NUMBER
23 000042 011524          MOV      @R5,(R4)+          ;SET THE RELAY MASK BIT
24 000044 005014          CLR      @R4          ;SET TO OPEN RELAY
25 000046          DIR$     *ST.QIO          ;SERVICE THE SOLENOID
26 000054 000207          RETURN
27         ;+
28         ;
29         ; STORAGE AREA
30         ;
31         ;-
32 000056   ST.QIO: QIOU$   10.MLO,,,QIOSTA
33         ;
34 000106   QIOSTA: .BLKW   2
35         ;
36 000112   ST.ERR: .BLKW   1,          ;ERROR LOCATION
37         ;
38 000114   ST.STP: .BLKW   1          ;NUMBER OF STEPS
39         ;
40 000116 000000   ST.PWR: .WORD   0          ;RELAY NUMBER FOR POWER SUPPLY

```

41			:		
42	000120	000000	:	.WORD 0	;1ST RELAY ON POWER SUPPLY CARD
43			:		
44	000122	000000	:	.WORD 0	;POWER SUPPLY RELAY BIT POSITION
45			:		
46	000124	000000	:	.WORD 0	;DUMMY REQUIRED BT ST.20
47			:		
48	000126	000000	:	ST.MDT: .WORD 0	;RELAY NUMBER FOR 1ST STEPPER RELAY
49			:		
50	000130	000000	:	.WORD 0	;1ST RELAY ON STEPPER RELAY CARD
51			:		
52	000132	000000	:	.WORD 0	;1ST STEPPER RELAY POSITION
53			:		
54	000134	000000	:	.WORD 0	;2ND STEPPER RELAY BIT POSITION
55			:		
56	000136		:	ST.BFR: .BLKW 4	;BUFFER FOR 4 SEQUENTIAL STEPS
57			:		
58			:		
59			:		
60			:		
61			:	ST.20 SUBROUTINE TO DETERMINE BIT SETTING FOR RELAY CHANNELS	
62			:		
63			:	ENTRY R0 - CONTAINS THE LOGICAL RELAY NUMBER 0,1,2 ...ETC.	
64			:	R5 - CONTAINS ADDR OF 3-WORD STORAGE AREA	
65			:		
66			:	RETURN	
67			:	STORAGE AREA - STARTING RELAY NUMBER FOR THE PARTICULAR	
68			:		CARD ... 0,16.ETC.
69			:	+2	- BIT TO VIRTUAL RELAY, 0-15
70			:	+4	- BIT TO VIRTUAL RELAY + 1
71			:		
72			:	USER DEFINED PARAMETERS	
73			:		
74		000002	:	NUMRLY = 2	
75			:		
76	000146		:	ST.20:	
77	000146	005015	:	CLR R5	;ASSUME RELAY IS ON FIRST CARD
78	000150	012704 000002	:	MOV *NUMRLY,R4	;NUMBER OF CARDS IN ICS FILE
79	000154	005003	:	CLR R3	;1ST CARD HAS 0 RELAY TO START
80	000156		:		
81	000156	005304	:	DEC R4	;ANY MORE CARDS TO TRY?
82	000160	003406	:	BLE 20\$	;MUST HAVE BEEN LAST CARD
83	000162	062703 000020	:	ADD *16.,R3	;BUMP TO 1ST ON NEXT CARD

84	000166	020003		CMP	R0,R3	;THIS RELAY ON LAST CARD?
85	000170	002402		BLT	20\$	;BR IF YES
86	000172	010315		MOV	R3,0R5	;SAVE STARTING RELAY NUMBER
87	000174	000770		BR	10\$	;LOOK AT NEXT CARD
88	000176		20\$:			
89	000176	012701	000001	MOV	*1,R1	;INITIALIZE VIRTUAL AND
90	000202	012702	000002	MOV	*2,R2	;... VIRTUAL+1 BIT POSITIONS
91	000206	151500		SUB	0R5,R0	;GET LOGICAL BIT POSITIONS
92	000210	001403		BEQ	60\$	;DONE IF ZERO
93	000212		30\$:			
94	000212	005301		ASL	R1	;ROTATE TO
95	000214	005302		ASL	R2	;... PROPER POSITIONS
96	000216	077003		SUB	R0,30\$	
97	000220		60\$:			
98	000220	010165	000002	MOV	R1,2(R5)	;PASS BACK
99	000224	010265	000004	MOV	R2,4(R5)	;... TO CALLER
100	000230	000207		RETURN		
101		000001		.END		

FORTRAN IV V01C-03C
CORE=08K, UIC=[1,11]

WED 05-JUL-78 00:21:34

PAGE 001
.LP=TE2DAC

```
0001      PROGRAM TE2DAC  
         C      .IDENT      FEB 8, 1978  
         C  
         C      THIS PROGRAM IS NECESSARY TO ALLOW FOR THE  
         C      FORTRAN CALLS INTERNAL TO THE DATA ACQUISITION SOFTWARE  
         C  
0002      CALL      TE2MAC      !CALL MACRO VERSION OF TE2DAC  
0003      STOP  
0004      END
```

FORTRAN IV V01C-03C
CORE-08K, UIC-1113

WED 05-JUL-78 00:21:50

PAGE 001
.LP=TE2LN2

```

0001      SUBROUTINE TE2LN2(IVALUE,GAIN,NTURNS,ILUN,IPSC,ILNS,ILNC,IERR
          1      ,IVAL2,GAIN2)
          C      .IDENT      /042678/
          C+
          C      THIS SUBROUTINE SERVICES THE LN2 SUPPLY
          C
          C      EDITED 4/21/78 LLH - BRANCH ON SAMPLE TEMPERATURE TO CLOSE
          C      LN2 VALVE
          C
          C      IVALUE      = A/D MAGNITUDE OF THE SPLTMP
          C      GAIN      = A/D GAIN SETTING FOR SAMPLE TEMPERATURE
          C      NTURNS    = CURRENT LN2 VALVE OPENING IN QUARTER (90 DEG) INCREMENTS
          C      ILUN      = LOGICAL UNIT NUMBER
          C      IPSC      = POWER SUPPLY RELAY NUMBER
          C      ILNS      = LN2 SOLENOID VALVE
          C      ILNC      = LN2 VALVE STEPPER MOTOR RELAY NUMBER 1
          C      IERR      = 0 IF NO ERRORS, OTHERWISE CONTAINS I/O ERROR CODE
          C      IVAL2     = A/D MAGNITUDE OF HEATER VOLTAGE
          C      GAIN2     = A/D GAIN SETTING FOR HEATER VOLTAGE
          C
          C      J.S. BESHNER      JANUARY 20, 1978
          C-
          C
0002      VOLTS      = IVAL2      !CONVERT A/D HEATER READING TO VOLTS
0003      VOLTS      = 4.8852*VOLTS/(1000.*GAIN2)
          C
0004      VOLTS2     = IVALUE
0005      VOLTS2     = 4.8852*VOLTS2/GAIN
0006      CALL TEMPF(VOLTS2,SPLTMP)
          C
0007      IF(SPLTMP.LT.100.) GO TO 10
0009      CALL CLOSE (ILNS,ILUN)      !CLOSE SOLENOID IF SLTMP > 100 DEG F
0010      GO TO 999
          C
0011  10      CALL OPEN (ILNS,ILUN)      !OPEN SOLENOID IF HV <= 5 VOLTS
0012      N = -1      !SET TURNS TO OPEN VALVE 1/4 TURN
          C+
          C      LET THE HEATER VOLTAGE = HV
          C      IF HV < 2.00, OPEN THE LN2 VALVE 1/4 TURN

```

```

C      IF 2.00 <= HV < 2.75, TAKE NO ACTION
C      IF 2.75 <= HV < 3.25, CLOSE THE LN2 VALVE 1/4 TURN
C      IF 3.25 <= HV <= 5.0, CLOSE THE LN2 VALVE 2/4 TURNS
C
C      NEVER CLOSE THE VALVE DOWN TO LESS THAN 1/4 TURN NOR
C      OPEN IT MORE THAN A TOTAL OF 20/4 TURNS
C-
0013      IF(VOLTS.LT.2.0) GO TO 20      !OPEN VALVE 1/4 TRN IF < 2 VOLTS
0015      IF(VOLTS.LT.2.75) GO TO 999
0017      N = -N                          !SET TURNS TO CLOSE VALVE 1/4 TURN
0018      IF(VOLTS.LT.3.25) GO TO 30
0020      IF(NTURNS.LE.2) GO TO 30      !DON'T TRY 2 TURNS IF OPEN 2 OR LESS
0022      N = 2*N                        !SET TURNS TO CLOSE VALVE 2/4 TURNS
0023      GO TO 40
0024      20      IF(NTURNS.LT.20) GO TO 40
0026      GO TO 999
0027      30      IF(NTURNS.GT.1) GO TO 40
0029      GO TO 999
0030      40      NTURNS = NTURNS - N      !CALCULATE NEW TURN TOTAL
0031      CALL STEP(ILUN,IPSC,ILNC,N,IERR) !GO TURN THE VALVE
0032      999      RETURN                  !BACK TO CALLER
0033      STOP
0034      END

```

FORTRAN IV V01C-83C
CORE=08K, UIC=C1,111

WED 05-JUL-78 08:22:03

PAGE 001
,LP=TE2CYC

```

0001      SUBROUTINE TE2CYC(IVALUE,GAIN,ISWG,RTEMP,ICBYT,INEW)
           C      IDENT      /012078/
           C+
           C      DETERMINES THE CURRENT THERMAL EXPANSION CYCLE POSITION
           C      AND UPDATES ICBYT ACCORDING TO THE FOLLOWING CONVENTION:
           C      ICBYT = 0, CYCLE IS BELOW STARTING CYCLE TEMPERATURE
           C      ICBYT = 1, CYCLE IS ABOVE STARTING CYCLE TEMPERATURE
           C
           C      IVALUE      = A/D MAGNITUDE OF CURRENT SPECIMEN TEMPERATURE
           C      GAIN        = A/D GAIN VALUE
           C      ISWG        = INITIAL CYCLE SWING, 0 = DOWN ; 1 = UP
           C      RTEMP       = STARTING CYCLE TEMPERATURE
           C      ICBYT       = CURRENT CYCLE POSITION
           C      INEW        = SET TO 0 IF NOT NEW CYCLE, SET TO 1 IF NEW CYCLE
           C
           C      J. S. BESHONER      JANUARY 20, 1978
           C-
0002      INEW      = 0      !SHOW NO NEW CYCLE
0003      VOLTS      = IVALUE !CALCULATE CURRENT SPECIMEN TEMPERATURE
0004      VOLTS      = 4.8852*VOLTS/GAIN
0005      CALL TEMPF(VOLTS,STEMP) !CONVERT TO DEGREES
           C
           C      DETERMINE CYCLE POSITION
           C
0006      IF(STEMP-RTEMP)100,300,200
0007 100      NCBYT      = 0      !SPECIMEN TEMP BELOW STARTING TEMP
0008      IF((ISWG.EQ.0.AND.(ICBYT.EQ.1)) INEW=1
0009      GO TO 400
           C
0011 200      NCBYT      = 1      !SPECIMEN TEMP ABOVE STARTING TEMP
0012      IF((ISWG.EQ.1.AND.(ICBYT.EQ.0)) INEW=1
0014      GO TO 400
           C
0015 300      NCBYT      = ICBYT      !SPECIMEN TEMP = STARTING TEMP
           C
0016 400      ICBYT      = NCBYT      !SET NEW CYCLE POSITION
0017      RETURN
0018      STOP
0019      END

```

```

1      .MACRO TE$DEF
2      .ASECT
3      ;
4      ; HEADER RECORD OFFSET DEFINITIONS (4 DATA SETS LONG)
5      .#0
6      TE.LTR::      .BLKB 7      ;LTR NUMBER ASCII 7 CHAR
7      TE.SID::      .BLKB 2      ;SPECIMEN ID-ASCII 2 CHAR
8      TE.EIN::      .BLKB 6      ;EMP NUM-ASCII 6 CHAR
9      TE.TIM::      .BLKB 2      ;CURRENT TIME-BINARY HR. MIN
10     TE.DAT::      .BLKB 3      ;CURRENT DAT-BIN MON. DAY, YEAR
11     TE.LEN::      .BLKW 2      ;SPECIMEN INITIAL LENGTH-FLTPT
12     TE.TEM::      .BLKW 2      ;STARTING CYCLR TEMP-FLTPT
13     TE.MCY::      .BLKB 1      ;NUM THERMAL CYCLES-BIN
14     TE.SWG::      .BLKB 1      ;INITIAL SWING DIRECTION-BIN
15                                     ;+1= UPWARD SWING
16                                     ;0= DOWNWARD SWING
17     TE.LST::      .BLKW 2      ;LOW SIDE CYCLE TEMP-FLTPT
18     TE.HST::      .BLKW 2      ;HIGH SIDE CYCLE TEMP-FLTPT
19     TE.STC::      .BLKB 1      ;SPEC. TEMP A/D CHAN *-BIN
20     TE.HTC::      .BLKB 1      ;HEATER TEMP A/D CHAN *-BIN
21     TE.LVC::      .BLKB 1      ;LVDT VOLTAGE CHAN *-BIN
22     TE.HVC::      .BLKB 1      ;HEATER VOLTAGE CHAN *-BIN
23     TE.PSC::      .BLKB 1      ;P/S RELAY *-BIN
24     TE.LNS::      .BLKB 1      ;LIO N2 SOLENOID RELAY *-BIN
25     TE.LNC::      .BLKB 1      ;STEPPER MOTOR RELAY *-BIN
26     TE.LNV::      .BLKB 1      ;INITIAL LN2 QTR TURNS OPEN-BIN
27     TE.INT::      .BLKB 1      ;NUM MINUTES BETW. SAMPLES-BIN
28     TE.CCN::      .BLKB 1      ;CURRENT CYCLE NUM-BIN
29     TE.MVR::      .BLKW 2      ;LVDT/MV/INCH RATIO-FLTPT
30     TE.NDS::      .BLKW 1      ;NUM DATA SETS RECORDED-BIN
31     ; DATA RECORD OFFSET DEFINITIONS
32     .#0
33     TE.DTI::      .BLKB 2      ;CURRENT TIME HOUR,MIN-BIN
34     TE.DHT::      .BLKW 1      ;CURRENT HEATER TEMP-A/D VALUE
35     TE.DST::      .BLKW 1      ;CURRENT SPECIMEN TEMP(GAIN)
36     TE.DLV::      .BLKW 1      ;CURRENT LVDT VOLTAGE GAIN
37     TE.DHV::      .BLKW 1      ;CURRENT HEATER VOLTAGE GAIN
38     TE.DLN::      .BLKB 1      ;CURRENT LN2 QTR TURNS OPEN-BIN
39     TE.DXD::      .BLKB 1      ;NUM OF CROSS OVERS FOR CYCLE
40                                     ;0=LOWER SIDE OF CYCLE

```


41					
42			.BLKW	2	; 1-UPPER HALF OF CYCLE
43	TE.DSZ**.				; FILL BYTES
44	;				; LENGTH OF RECORD IN BYTES
45	TE.SOD**	4*TE.DSZ			; SIZE OF HEADER DATA
46	;				
47	N=1000-TE.SOD				
48	NRECB1**	N/TE.DSZ			; * OF RECORDS IN BLOCK 1
49	NRECBN**	1000/TE.DSZ			; * OF RECORDS IN BLOCK N
50		.PSECT			
51		.MACRO	TE\$DEF		

```

1                                     .TITLE TERECS
2                                     .IDENT/030170/
3
4                                     ;+
5                                     ;      MARCH 1970                                D. E. LACEY
6                                     ;
7                                     ;      THIS FORTRAN CALLABLE SUBROUTINE RETURNS AN ARRAY
8                                     ;      CONTAINING THE LENGTH IN BYTES OF THE HEADER AND DATA
9                                     ;      RECORDS AND THE NUMBER OF RECORDS CONTAINED IN BLOCKS
10                                    ;      ONE AND N OF THE TE2000 DATA FILE.
11                                    ;
12                                    ;      CALLING SEQUENCE:
13                                    ;
14                                    ;      CALL TERECS(1SIZS(1))
15                                    ;-
16 000000 TERECS::
17 000000 005725      TST      (R5)+
18 000002 012700 000052  MOV     @LIST, R0
19 000006 016720 000000G  MOV     TE.SOD, (R0)+
20 000012 016720 000000G  MOV     TE.DSZ, (R0)+
21 000016 016720 000000G  MOV     NRECB1, (R0)+
22 000022 016720 000000G  MOV     NRECBN, (R0)+
23 000026 011504      MOV     (R5), R4
24 000030 016724 000016  MOV     LIST, (R4)+
25 000034 016724 000014  MOV     LIST+2, (R4)+
26 000040 016724 000012  MOV     LIST+4, (R4)+
27 000044 016714 000010  MOV     LIST+6, (R4)
28 000050 000207      RETURN
29 000052
30      000001 LIST: .BLKW 4
                      .END

```

STEP ;STEPS RELAYS FOR STEPP MACRO M1110 05-JUL-78 08:22 PAGE 1

```

1          .TITLE STEP ;STEPS RELAYS FOR STEPPER MOTORS
2          .IDENT /022470/
3          ;+
4          ;STEP CALLED TO STEP RELAYS FOR STEPPER MOTORS.
5          ; A STEP CONSISTS OF 3 SETS OF 4 SEQUENTIAL OUTPUTS TO THE STEPPER
6          ; MOTOR AND ROTATES THE SHAFT 90 DEGREES.
7          ;
8          ; CALLING SEQUENCE FOLLOWS:
9          ;
10         ; CALL STEP(ILUN,IPOWER,IMOTOR,ISTEP,IERR)
11         ;
12         ; WHERE:
13         ; ILUN = LUN ASSIGNED
14         ; IPOWER = RELAY NUMBER CORRESPONDING TO POWER SUPPLY (0-31)
15         ; IMOTOR = RELAY NUMBER OF LEAST SIGNIFICANT STEPPER RELAY (0-31)
16         ; (2ND RELAY ASSUMED SEQUENTIALLY NEXT TO THIS RELAY)
17         ; ***** NOTE THAT CERTAIN VALUES ARE DECLARED ILLEGAL IN
18         ; THE INITIATOR SOFTWARE
19         ; ISTEP = THE NUMBER OF 1/4 STEPS DESIRED. A POSITIVE ISTEP
20         ; CAUSES THE MOTOR TO MOVE IN A CLOCKWISE DIRECTION AS
21         ; SEEN FROM AN OBSERVER SITTING ON THE MOTOR LOOKING DOWN
22         ; THE SHAFT. (CW CLOSES THE VALVE ON TE2000 LN2)
23         ; IERR = 0 IF NO ERRORS OCCUR OTHERWISE CONTAINS THE Q10
24         ; DIRECTIVE STATUS.
25         ;
26         ; J. S. BESHNER 9-JAN-78 (WRITTEN FOR TE2000)
27         ;-
28         .GLOBAL STEP,OPEN,CLOSE ;PROGRAM ENTRY
29         .GLOBAL $DSW
30         ;+
31         ; OPEN OPENS THE LN2 SOLENOID VALVE
32         ;
33         ; CLOSE CLOSES THE LN2 SOLENOID VALVE
34         ;
35         ;-
36         ;
37         .MCALL MKT$S,WTSE$S,Q10W$,DIR$
38         ;
39         ;+
40         ;USER DEFINED PARAMETERS
41         ;
42         ; TIMER = 7. ;CLOCK TICKS TO WAIT BETWEEN STEPS

```

000007

43	000002			NUMRLY = 2	:NUMBER OF RELAY CARDS IN THE SYSTEM
44					
45	000020			TIMEFN = 20	:EVENT FLAG FOR MARK-TIME DIRECTIVE
46	000021			QIOEFN = 21	:EVENT FLAG FOR QIO TO DIRECTIVE
47					
49	000000			STEP:	
50	000000	005725		TST (R5)+	:MOVE PAST FORTRAN COUNTER
51	000002	013567	000550	MOV @ (R5)+,ST.QIO+Q.IOLU	:SET LUN
52	000006	116767	000544 000544	MOVB ST.QIO+Q.IOLU,ST.QIO+Q.IOEF	:SET EFN
53	000014	013567	000574	MOV @ (R5)+,ST.PWR	:POWER SUPPLY RELAY *
54	000020	013567	000600	MOV @ (R5)+,ST.MOT	:LS MOTOR RELAY NUMBER
55	000024	013567	000560	MOV @ (R5)+,ST.STP	:NUMBER OF 1/4 STEPS REQUESTED
56	000030	011567	000552	MOV (R5),ST.ERR	:ADDR OF ERROR WORD
57	000034	016700	000554	MOV ST.PWR,R0	:RETRIEVE POWER RELAY
58	000040	012705	000616	MOV *ST.PWR+2,R5	:WHERE TO SAVE DATA
59	000044	004767	000254	CALL ST.20	:DETERMINE POWER RELAY BIT POSITION
60	000050	012704	000566	MOV *ST.QIO+Q.IOPL,R4	:ADDR OF QIO PARAMETER BLK
61	000054	012524		MOV (R5)+,(R4)+	:PUT STARTING CARD RELAY IN BLK
62	000056	011524		MOV @R5,(R4)+	:BIT LOCATION OF POWER RELAY
63	000060	011514		MOV @R5,@R4	:USE SAME BIT FOR MASK
64	000062			DIR* *ST.QIO,ST.90	:TURN ON THE POWER SUPPLY
65	000076	016700	000522	MOV ST.MOT,R0	:RELAY * FOR LS MOTOR RELAY
66	000102	012705	000626	MOV *ST.MOT+2,R5	:WHERE TO SAVE DATA
67	000106	004767	000212	CALL ST.20	:DETERMINE MOTOR RELAY BITS
68	000112	012704	000634	MOV *ST.BFR,R4	:STEP SEQUENCE BUFFER
69	000116	016702	000466	MOV ST.STP,R2	:GET STEP COUNT. POSITIVE ?
70	000122	003012		BGT 10\$	:BR IF POSITIVE
71	000124	002402		BLT 5\$	:IF NEGATIVE THEN CCW
72	000126	000167	000266	JMP ST.99	:DO NOTHING IF ZERO
73					
74				:TURN MOTOR COUNTER CLOCKWISE	
75					
76	000132			5\$:	
77	000132	016764	000472 000004	MOV ST.MOT+4,4(R4)	:3RD AND 1ST STEPS
78	000140	016714	000466	MOV ST.MOT+6,(R4)	:... FOR CCW DIRECTION
79	000144	005402		NEG R2	:MAKE STEP COUNT POSITIVE
80	000146	000405		BR 20\$	:DO COMMON LOGIC
81					
82				:TURN MOTOR COUNTER CLOCKWISE	
83					
84	000150			10\$:	
85	000150	016714	000454	MOV ST.MOT+4,(R4)	:1ST AND 3RD STEPS
86	000154	016764	000452 000004	MOV ST.MOT+6,4(R4)	:... FOR CW DIRECTION
87	000162			20\$:	
88	000162	011464	000002	MOV (R4),2(R4)	:DETERMINE 2ND STEP

89	000166	056464	000004	000002	BIS	4(R4),2(R4)	;... FOR EITHER DIRECTION.	
90	000174	005064	000006		CLR	6(R4)	;4TH STEP ALWAYS THE SAME	
91	000200	016767	000422	000360	MOV	ST.MOT+2,ST.Q10+Q.10PL	;1ST RELAY ON MOTOR CARD	
92	000206	016467	000002	000354	MOV	2(R4),ST.Q10+Q.10PL+2	; DRIVER MASK FOR 2 RELAYS	
93	000214							
94	000214	012705	000003		25%:	MOV	*3,R5	;# OF SETS PER STEP
95	000220				30%:			
96	000220	012703	000004		MOV	*4,R3	;NUMBER OF STEPS PER 1/4 TURN	
97	000224				40%:			
98	000224	012467	000342		MOV	(R4)+,ST.Q10+Q.10PL+4	;GET NEXT STEPPER BITS	
99	000230				DIR\$	*ST.Q10,ST.90	;OUTPUT THOSE BITS	
100	000244				MRKT\$	*TIMEFN,*TIMER,*1	;GIVE MOTOR TIME TO STEP	
101	000270				WTSE\$	*TIMEFN	;WAIT FOR TIMER	
102	000302	077330			SOB	R3,40\$	;LOOP IF MORE IN SEQUENCE	
103	000304	162704	000010		SUB	*10,R4	;RESET SEQUENCE POINTER	
104	000310	077535			SOB	R5,30\$	;LOOP IF MORE SETS	
105	000312	077240			SOB	R2,25\$	;LOOP IF MORE STEPS	
106	000314	005077	000266		CLR	*ST.ERR	;ZERO THE ERROR CODE	
107	000320	000167	000074		JMP	ST.99	;GO TERMINATE	
109								
110					ST.20	SUBROUTINE TO DETERMINE BIT SETTING FOR RELAY CHANNELS		
111								
112					ENTRY	R0 - CONTAINS THE LOGICAL RELAY NUMBER 0,1,2 ...ETC.		
113						R5 - CONTAINS ADDR OF 3-WORD STORAGE AREA		
114								
115					RETURN			
116						STORAGE AREA - STARTING RELAY NUMBER FOR THE PARTICULAR		
117						CARD ... 0,16,ETC		
118						+2 - BIT TO VIRTUAL RELAY, 0-15		
119						+4 - BIT TO VIRTUAL RELAY + 1		
120								
121	000324				ST.20:			
122	000324	005015			CLR	QRS	;ASSUME RELAY IS ON 1ST CARD	
123	000326	012704	000002		MOV	*NUMRLY,R4	;NUMBER OF CARDS IN ICS FILE	
124	000332	005003			CLR	R3	;1ST CARD HAS 0 RELAY TO START	
125	000334				10%:			
126	000334	005304			DEC	R4	;ANY MORE CARDS TO TRY ?	
127	000336	003406			BLE	20\$	;MUST HAVE BEEN LAST CARD	
128	000340	062703	000020		ADD	*16,R3	;BUMP TO 1ST ON NEXT CARD	
129	000344	020003			CMP	R0,R3	;THIS RELAY ON LAST CARD ?	
130	000346	002402			BLT	20\$	;BR IF YES	
131	000350	010315			MOV	R3,QRS	;SAVE STARTING RELAY NUMBER	
132	000352	000770			BR	10\$	;LOOK AT NEXT CARD	
133	000354				20%:			
134	000354	012701	000001		MOV	*1,R1	;INITIALIZE VIRTUAL AND	

135	000360	012702	000002	MOV	#2,R2	...	VIRTUAL + 1 BIT POSITIONS
136	000364	161503		SUB	0R5,R0	;	GET LOGICAL BIT POSITIONS
137	000366	001403		BED	60\$	;	DONE IF ZERO
138	000370			30\$:			
139	000370	006301		ASL	R1	;	ROTATE TO
140	000372	006302		ASL	R2	;	... PROPER POSITIONS
141	000374	077003		S08	R0,30\$		
142	000376			60\$:			
143	000376	010165	000002	MOV	R1,2(R5)	;	PASS BACK
144	000402	010265	000004	MOV	R2,4(R5)	;	... TO CALLER
145	000406	000207		RETURN			
147				;	+		
148				;	HERE IF ERROR OCCURS DURING Q10		
149				;	-		
150	000410			ST.90:			
151	000410	005726		TST	(SP)+	;	DROP ERROR CALL RETURN
152	000412	013777	000000G 000166	MOV	@#\$DW,@ST.ERR	;	PASS ERROR TO CALLER
153				;			
154				;	FALL THROUGH		
155				;			
156				;	+		
157				;	ST.99 - EVERYBODY TERMINATES THROUGH HERE		
158				;			
159				;	Turns OFF POWER SUPPLY AND RETURNS TO ORIGINAL CALLER		
160				;	-		
161	000420			ST.99:			
162	000420	016767	000172 000140	MOV	ST,PWR+2,0.10PL+ST.Q10	;	PROPER CARD
163	000426	016767	000166 000134	MOV	ST,PWR+4,0.10PL+2+ST.Q10	;	BIT MASK
164	000434	005067	000132	CLR	Q.10PL+4+ST.Q10	;	TURN OFF POWER
165	000440			DIRS	*ST.Q10		
166	000446	000207		RETURN		;	BACK TO CALLER
168				;	+		
169				;	ROUTINES TO OPEN OR CLOSE THE LN2 SOLENOID VALVE		
170				;			
171				;			
172				;	CALL OPEN(ILNS,ILUN)		
173				;			
174				;	ILNS	=	LN2 SOLENOID RELAY NUMBER
175				;	ILUN	=	LOGICAL UNIT NUMBER
176				;	-		
177	000450			CLOSE:			
178	000450	005067	000136	CLR	OPNFLG	;	SHOW US CLOSING SOLENOID
179	000454	000403		BR	OPN.00	;	DO COMMON LOGIC
180				;			
181	000456			OPEN:			

```

182 000456 012767 000001 000126      MOV      *I,OPNFLG      ;SHOW US OPENING THE SOLENOID
183
184 000464      ;
      OPN.00:
185 000464 005725      TST      (R5)+      ;PAST FORTRAN GARBAGE
186 000466 013567 000122      MOV      @R5+,ST,PWR      ;SOLENOID REALY NUMBER
187 000472 013567 000060      MOV      @R5+,ST,Q10+Q.10LU      ; LUN
188 000476 116767 000054 000054      MOV     ST,Q10+Q.10LU,ST,Q10+Q.10EF      ;SET EFN
189 000504 016700 000104      MOV      ST,PWR,R0      ;DETERMINE THE RELAY BIT
190 000510 012705 000616      MOV      *ST,PWR+2,R5
191 000514 004767 177604      CALL     ST,20
192 000520 012704 000566      MOV      *ST,Q10+Q.10PL,R4      ;Q10 PARAMETER LIST POINTER
193 000524 012524      MOV      (R5)+,(R4)+      ;SET THE RELAY CARD NUMBER
194 000526 011524      MOV      @R5,(R4)+      ;SET THE RELAY MASK BIT
195 000530 005014      CLR      @R4      ;SET TO OPEN RELAY
196 000532 005767 000054      TST      OPNFLG      ;ARE WE OPENING ?
197 000536 001001      BNE      10$      ;BR IF YES
198 000540 011514      MOV      @R5,@R4      ;IF CLOSING, SET THE BIT
199 000542      ;
200 000542      ;
201 000550 000207      DIR$     *ST,Q10      ;SERVICE THE SOLENOID
      RETURN
203      ;+
204      ;STORAGE AREA
205      ;~
206 000552      ST.Q10: Q10W$ 10.MLO,,,Q10STA
207      ;
208 000602      Q10STA: .BLKW 2
209      ;
210 000606      ST.ERR: .BLKW 1      ;ERROR LOCATION
211      ;
212 000610      ST.STP: .BLKW 1      ;NUMBER OF STEPS
213      ;
214 000612      OPNFLG: .BLKW 1      ;OPEN/CLOSE SOLENOID FLAG
215      ;
216 000614 000000      ST.PWR: .WORD 0      ;RELAY NUMBER FOR POWER SUPPLY
217      ;
218 000616 000000      .WORD 0      ;1ST RELAY ON POWER SUPPLY CARD
219      ;
220 000620 000000      .WORD 0      ;POWER SUPPLY RELAY BIT POSITION
221      ;
222 000622 000000      .WORD 0      ;DUMMY REQUIRED BY ST.20
223      ;
224 000624 000000      ST.MDT: .WORD 0      ;RELAY NUMBER FOR 1ST STEPPER RELAY
225      ;
226 000626 000000      .WORD 0      ;1ST RELAY ON STEPPER RELAY CARD
227      ;

```

228	000630	000000	.WORD	0	;1ST STEPPER RELAY BIT POSITION
229					
230	000632	000000	.WORD	0	;2ND STEPPER RELAY BIT POSITION
231					
232	000634		ST.BFR: .BLKW	4	;BUFFER FOR 4 SEQUENTIAL STEPS
233					
234		000001	.END		

TE2DAC ;TE2000 DATA ACQUISITION MACRO M1110 05-JUL-79 09:23 PAGE 1

```

1          .TITLE  TE2DAC ;TE2000 DATA ACQUISITION
2          .IDENT  /042178/
3          .SOTTL  TE2MAC
4
5          ;+
6          ;PERFORMS DATA ACQUISITION FOR THERMAL EXPANSION
7          ;--
8          .MCALL  RCVD$$,DIR$,QIOW$,GTIM$$,ALTP$$,MRKT$
9          .MCALL  WSIG$$,WTSE$$,ALUN$$
10         .MCALL  FDATE$,FDOF$,OPNS$,READ$,WAIT$,WRITES$,CLOSE$
11         .MCALL  FRSZ$,FDBDF$,FDRCS$,FDBK$,FDOF$
12         .GLOBL  $DSW
13
14         .GLOBL  TE2MAC          ;ENTRY POINT FOR FORTRAN CALL
15
16         ;EXTERNALS
17
18         .GLOBL  TE2CYC          ;DETERMINES THERMAL CYCLE POSITION
19         .GLOBL  TE2LN2          ;SERVICES THE LN2 SUPPLY
20         .LIST  MEB
21         TE$DEF          ;DEFINE FILE OFFSETS
22         .NLIST  MEB
23
24         ;
25         DKLUN  = 1
26         DKEFN  = DKLUN
27         ICSLUN = DKLUN + 1
28         ICSEFN = ICSLUN
29         TILUN  = ICSLUN + 1
30         TIEFN  = TILUN
31         MRKEFN = TILUN + 1
32
33         ;
34         ;+
35         ;
36         ;
37         ;
38         ;
39         ;
40         ;
41         ;
42         ;

```

000000 000000G
 000001
 000001
 000002
 000002
 000003
 000003
 000004

EDITED MARCH 13, 1979 - JSB

 ADDED LOGIC TO RECEIVE NEW TIME FRAME FROM DATA REDUC-
 TION PROGRAM AND TO HOLD OFF FILE EXTENSION UNTIL
 FIVE (5) BLOCKS HAVE BEEN WRITTEN - (INITIATION
 SOFTWARE CREATES A FILE 5 BLOCKS LONG).

 EDITED APRIL 21, 1978 - LLH

```

43      ;
44      ;
45      ;
46      ;
47      ;
48      ;
50 000002      TE2MAC:
51 000002      ALUN$S  *DKLUN,*DK,*1
52 000024      ALUN$S  *ICSLUN,*IC,*0
53 000044      ALUN$S  *TILUN,*TT,*0
54 000064      RCVD$S  ,*RCVBUF      ;GET DATA FROM INITIATION TASK
55 000102 126727 000000G 000000G      CMPB      $DSW,*IS.SUC      ;GET SOME ?
56 000110 001404      BEO      40$      ;OR IF YES
57 000112      DIR$      *Q10BLK      ;TELL SOMEBODY
58 000120 000207      RETURN      ;AND TERMINATE
59 000122
60 000122 012700 001714"      40$:      MOV      *RCVBUF+4,R0      ;RCV DATA BUFFER
61
62      ;
63      ;
64 000126      ;+
65 000126 122720 000135      ;
66 000132 001375      ;
67 000134 012701 002206"      ;
68 000140 005011      ;
69 000142 016102 000002      ;
70 000146      ;
71 000146 005211      45$:      INC      0R1      ;COPY
72 000150 112022      MOV      (R0)+,(R2)+      ;...FILE NAME
73 000152 001375      BNE      45$      ;.....TO DATA SET
74 000154 005311      DEC      0R1      ;DON'T COUNT ZERO BYTE
75 000156 005067 002066      CLR      VBNADR      ;READ
76 000162 012767 000001 002062      MOV      *1,VBNADR+2      ;...1ST BLOCK
77 000170 004767 001166      CALL      TE2.20      ;.....OF FILE
78
79      ;
80      ;
81 000174 012705 002256"      ;
82 000200 012767 000002 003106      ;BUILD A/D AND LN2 WORKING BLOCKS FOR FORTRAN CALLS
83 000206 116567 000000G 003102      ;
84 000214 116567 000000G 003076      ;
85 000222 116567 000000G 003072      ;
86 000230 112765 000001 000000G      ;
87
88 000236 116567 000000G 003112      ;

```

ADDED LOGIC TO SEND SAMPLE TEMPERATURE MAGNITUDE AND GAIN TO FORTRAN SUBROUTINE TE2LN2, SO THAT THE SUBROUTINE CAN TEST THE SAMPLE TEMPERATURE AND CLOSE THE LN2 VALVE WHEN NECESSARY.

TE2MAC:

ALUN\$S *DKLUN,*DK,*1

ALUN\$S *ICSLUN,*IC,*0

ALUN\$S *TILUN,*TT,*0

RCVD\$S ,*RCVBUF ;GET DATA FROM INITIATION TASK

CMPB \$DSW,*IS.SUC ;GET SOME ?

BEO 40\$;OR IF YES

DIR\$ *Q10BLK ;TELL SOMEBODY

RETURN ;AND TERMINATE

40\$: MOV *RCVBUF+4,R0 ;RCV DATA BUFFER

LOOP PAST THE UIC ON FRONT OF FILE NAME - DEFAULT TO [300,300]

42\$: CMPB *1,(R0)+ ;LOOKING FOR END OF UIC

BNE 42\$

MOV *DSDES+10,R1 ;DATASET FILE DESCRIPTOR

CLR 0R1 ;ZERO CHAR CNT OF FILE NAME

MOV 2(R1),R2 ;ADDR OF WHERE TO SAVE FILE NAME

45\$: INC 0R1 ;COPY

MOVB (R0)+,(R2)+ ;...FILE NAME

BNE 45\$;.....TO DATA SET

DEC 0R1 ;DON'T COUNT ZERO BYTE

CLR VBNADR ;READ

MOV *1,VBNADR+2 ;...1ST BLOCK

CALL TE2.20 ;.....OF FILE

;BUILD A/D AND LN2 WORKING BLOCKS FOR FORTRAN CALLS

MOV *IOAREA,R5 ;START OF DATA BLOCK

MOV *ICSLUN,LN2LUN ;ICS LUN FOR LIQUID NITROGEN

MOVB TE.PSC(R5),LN2PSC ;POWER SUPPLY RELAY NUMBER

MOVB TE.LNS(R5),LN2LNS ;SOLENOID RELAY NUMBER

MOVB TE.LNC(R5),LN2LNC ;LN2 STEPPER MOTOR RELAY *1

MOVB *1,TE.CCN(R5) ;INITIALIZE THE CYCLE COUNTER

MOVB TE.STC(R5),ADSTC ;A/D SPECIMEN TEMP CHANNEL *

```

89 000244 116567 000000G 003106      MOV8  TE,HTC(R5),ADHTC ; " HEATER " " "
90 000252 116567 000000G 003102      MOV8  TE,LVC(R5),ADLVC ; " LVDT VOLTAGE " "
91 000260 116567 000000G 003076      MOV8  TE,HVC(R5),ADHVC ; " HEATER " " "
92                                     ;
93 000266 116567 000000G 003122      MOV8  TE,SWG(R5),CYCSWG ; INITIAL CYCLE SWING
94 000274 016767 003116 003122      MOV    CYCSWG,1CBYT ;SAVE FOR FORTRAN CALL
95 000302 116567 000000G 003002      MOV8  TE,LNV(R5),NTURNS ;INITIAL LN2 VALVE 1/4 TURNS
96                                     ;
97 000310 016567 000000G 003102      MOV    TE,TEM(R5),CYCTEM ;START-OF-CYCLE
98 000316 016567 000002G 003076      MOV    TE,TEM+2(R5),CYCTEM+2 ;...TEMPERATURE
99                                     ;
100                                    ;
101                                    ;START REAL-TIME LOOP HERE
102                                    ;
103 000324      TEZ.05;
104 000324      GTIMMS  *TIMBFR ;GET CURRENT TIME OF DAY
105 000336 012705 003342*      MOV    *ADLIST,R5 ;FORTRAN A/D LIST
106 000342 004767 000000G      CALL    ADREAD ;READ THE DATA CHANNELS
107 000346 005767 002732      TST     LN2ERR ;ANY ERRORS ?
108 000352 001402      BEQ     5$ ;SKIP IF NONE
109 000354 000167 001256      JMP     ERR.20 ;PROCESS A/D ERROR
110 000360      5$;
111 000360 005767 001670      TST     LOOPFG ;1ST TIME IN THE LOOP ?
112 000364 001003      BNE     10$ ;BR IF NOT 1ST TIME
113 000366 005267 001652      INC     LOOPFG ;SET LOOP FLAG
114 000372 000407      BR       20$ ;AND SKIP DK READ
115 000374      10$;
116 000374 005067 001650      CLR     VBNADR ;ON SUBSEQUENT ENTRIES
117 000400 012767 000001 001644      MOV    *1,VBNADR+2 ;...WE NEED TO READ THE
118 000406 004767 000750      CALL    TE2.20 ;.....1ST BLK OF THE FILE
119 000412      20$;
120 000412      RCV0$S ;*RCVBUF ;SEE IF CHANGING LOOP TIME
121 000430 126727 000000G 000000G      CMP8  $DSW,*IS,SUC ;ARE THEY ???
122 000436 001004      BNE     22$ ;SKIP IF NOT
123 000440 116767 001250 000001C      MOV8  RCVBUF+4,TE,INT+10AREA ;IF YES THEN SAVE NEW TIME
124 000446 000761      BR       20$ ;SEE IF THEY CHANGED THEIR MIND
125 000450      22$;
126 000450 116767 000001C 001324      MOV8  TE,INT+10AREA,MRKTIM+M.KTMS ;CYCLE DELAY TIME
127 000456      DIR$  *MRKTIM,ERR.01 ;...START TIMER NOW
128                                     ;
129 000472      ALTP$S ;*50. ;LOWER OUR RUNNING PRIORITY
130                                     ;
131 000510 016700 002652      MOV    HVARV,R0 ;GET HEATER VOLTAGE
132 000514 004767 000542      CALL    TE2.10 ;CONVERT TO VALUE AND GAIN
133 000520 016767 002550 002600      MOV    IVALUE,IVAL2 ;SAVE HEATER VALUE

```

134	000526	016767	002554	002574	MOV	GAIN,GAIN2	;SAVE HEATER GAIN
135	000534	016767	002550	002570	MOV	GAIN+2,GAIN2+2	;SAVE HEATER GAIN
136	000542	016700	002622		MOV	STADV,R0	;GET SPECIMEN TEMPERATURE
137	000546	004767	000510		CALL	TE2,10	;CONVERT TO VALUE AND GAIN
138	000552	016767	002526	002554	MOV	IVALUE,IVLHD	;SAVE IVALUE OF SAMPLE TEMP
139	000560	016767	002522	002550	MOV	GAIN,GAINHD	;SAVE GAIN OF SAMPLE TEMP
140	000566	016767	002516	002544	MOV	GAIN+2,GAINHD+2	;SAVE GAIN OF SAMPLE TEMP
141	000574	012705	003256		MOV	*LN2LST,R5	;LN2 PARAMETER LIST
142	000600	004767	000000G		CALL	TE2LN2	;SERVICE THE LN2 SUPPLY
143							
144	000604	016767	002524	002472	MOV	IVLHD,IVALUE	;RESTORE SAMPLE TEMP VALUE
145	000612	016767	002520	002466	MOV	GAINHD,GAIN	;RESTORE SAMPLE TEMP GAIN
146	000620	016767	002514	002462	MOV	GAINHD+2,GAIN+2	;.....TO ORIGINAL VARIABLES
147							
148	000626	012705	003400		MOV	*CYCLST,R5	;CYCLE DATA LIST FOR FORTRAN CALL
149	000632	004767	000000G		CALL	TE2CYC	;DETERMINE CURRENT CYCLE POSITION
150	000636	005767	002564		TST	INew	;STARTING NEW CYCLE ?
151	000642	001412			BEQ	25%	;BR IF NO
152	000644	012700	002256		MOV	*IOAREA,R0	;POINT TO INITIAL DATA
153	000650	105260	000000G		INCB	TE.CCN(R0)	;INCREMENT CYCLE COUNTER
154	000654	126060	000000G	000000G	CMPE	TE.CCN(R0),TE.NCY(R0)	;WAS THAT THE LAST CYCLE ?
155	000662	003402			BLE	25%	;BR IF NO
156	000664	005267	001300		INC	CYCEND	;OTHERWISE SET END OF TEST FLAG
157							
158	000670						25%:
159	000670	012700	000000C		MOV	*TE.NDS+IOAREA,R0	;DATASET COUNTER ADDRESS
160	000674	005210			INC	R0	;ADD 1 FOR THIS READING
161	000676	011001			MOV	R0,R1	;GET NEW DATASET COUNT
162	000700	020127	000000G		CMPE	R1,*NRECB1	;WILL THIS RECORD FIT IN 1ST BLK ?
163	000704	003003			BGT	30%	;BR IF IT WON'T
164	000706	012702	000000C		MOV	*TE.SOD+IOAREA,R2	;POINT TO 1ST RECORD IN BLK
165	000712	000452			BR	72%	;DETERMINE WHERE TO STORE RECORD
166	000714						30%:
167	000714	012767	000001	001330	MOV	*1.VBNADR+2	;RESET VIRTUAL BLK NUMBER
168	000722	004767	000576		CALL	TE2,25	;WRITE THE 1ST BLK BACK TO DISK
169	000726	162701	000000G		SUB	*NRECB1,R1	;DEDUCT 1ST BLK'S RECORDS
170	000732						40%:
171	000732	005267	001314		INC	VBNADR+2	;POINT TO NEXT BLK
172	000736	020127	000000G		CMPE	R1,*NRECBN	;RECORD FIT INTO NEXT BLK ?
173	000742	003403			BLE	50%	;BR IF IT WILL
174	000744	162701	000000G		SUB	*NRECBN,R1	;DEDUCT RECORDS IN THIS BLK
175	000750	000770			BR	40%	;TRY AGAIN
176	000752						50%:
177	000752	020127	000001		CMPE	R1,*1	;THIS 1ST RECORD IN NEXT BLK ?
178	000756	001407			BEQ	60%	;IF YES, GO EXTEND FILE

179	000760				CLOSE\$	*TE2FDB		;CLOSE FILE BEFORE READING IT
180	000770	004767	000366		CALL	TE2.20		;OTHERWISE READ NEXT BLOCK
181	000774	000417			BR	70\$		;AND DETERMINE RECORD LOCATION
182	000776			60\$:				
183	000776	012704	002256		MOV	*IOAREA,R4		;ZERO THE NEW BLOCK
184	001002	012703	000400		MOV	*400,R3		
185	001006			65\$:				
186	001006	005024			CLR	(R4)+		
187	001010	077302			SQB	R3,65\$		
188	001012	022767	000005	001232	CMF	*5,VBNADR+2		;1ST FIVE BLKS ALREADY THERE ???
189	001020	002005			BGE	70\$		
190	001022	012700	002010		MOV	*TE2FDB,R0		;EXTEND THE FILE IF *5 OR BIGGER
191	001026	003002			CLR	R2		;...BY 1 NON-CONTIGUOUS
192	001030	004767	000000G		CALL	.EXTND		;.....BLOCK
193	001034			70\$:				
194	001034	012702	002256		MOV	*IOAREA,R2		;OFFSET TO 1ST RECORD IN BLK
195	001040			72\$:				
196	001040	005301			DEC	R1		;DETERMINE
197	001042	001403			BEQ	00\$		;...POINTER
198	001044	062702	000000G		ADD	*TE.DSZ,R2		;.....TO NEW
199	001050	000773			BR	72\$		;.....DATASET
200				:+				
201								;INSERT NEW DATASET VALUES INTO RECORD
202				:-				
203	001052			00\$:				
204	001052	116762	001100	000000G	MOV\$	TIMBFR+G.TIHR,TE.DTI(R2)		;HOURS
205	001060	116762	001074	000001G	MOV\$	TIMBFR+G.TIMI,TE.DTI+1(R2)		;MINUTES
206	001066	016762	002300	000000G	MOV	HTADV,TE.DHT(R2)		;HEATER TEMP
207	001074	016762	002270	000000G	MOV	STADV,TE.DST(R2)		;SPECIMEN TEMP
208	001102	016762	002266	000000G	MOV	LVADV,TE.DLV(R2)		;LVDT
209	001110	016762	002262	000000G	MOV	HVADV,TE.DHV(R2)		;HEATER VOLTS
210	001116	116762	002170	000000G	MOV\$	NTURNS,TE.DLN(R2)		;LN2 TURNS
211	001124	116762	002274	000000G	MOV\$	ICBYT,TE.DXO(R2)		;CYCLE POSITION
212	001132	004767	000366		CALL	TE2.25		;WRITE DATA TO DK:
213	001136				CLOSE\$	*TE2FDB		;CLOSE THE FILE
214				:				
215	001146	005767	001016		TST	CYCEND		;END OF TEST ?
216	001152	001025			BNE	100\$		;SKIP IF YES
217	001154				ALTP\$	*100.		;OTHERWISE RAISE OUR PRIORITY
218	001210				UTSE\$	*MRKEFN		;AND WAIT ON TIMER
219	001222	000167	177076		JMP	TE2.05		;LOOP
220	001226			100\$:				

221	001226	012705	003430'	MOV	*CLSLST,R5	;CLOSE PARAMETER LIST
222	001232	004767	000000G	CALL	CLOSE	;CLOSE SOLENOID BEFORE LEAVING
223	001236	012700	003512'	MOV	*MSG2,R0	;GET TERMINATION MESSAGE
224	001242			TE2.05:		
225	001242	010067	000514	MOV	R0,Q10BLK+0.10PL	;TELL OPERATOR
226	001246	112067	000512	MOVB	(R0)+,Q10BLK+0.10PL+2	; WE'RE LEAVING
227	001252			DIF\$	*Q10BLK	
228	001260	000207		RETURN		
230				;		
231				;	TE2.10	
232				;		
233				;		CONVERTS THE A/D VALUE IN R0 TO
234				;		A MAGNITUDE IN 'IVALUE' AND A FLOATING
235				;		POINT GAIN IN 'GAIN'.
236	001262			;		
237	001262	010067	002016	TE2.10:		
238	001266	042767	170000	MOV	R0,IVALUE	
239	001274	032767	004000	BIC	*170000,IVALUE	;SAVE MAGNITUDE
240	001302	001403		BIT	*4000,IVALUE	;NEGATIVE MAGNITUDE ?
241	001304	052767	170000	BEQ	5\$	;SKIP IF NOT
242	001312		001772	BIS	*170000,IVALUE	;OTHERWISE MAKE IT NEGATIVE
243	001312	000300		5\$:		
244	001314	006000		SWAB	R0	;GAIN TO LOW BYTE
245	001316	006000		ROR	R0	
246	001320	006000		ROR	R0	
247	001322	006000		ROR	R0	
248	001324	042703	177760	ROR	R0	;GAIN TO LS BITS
249	001330	000241		BIC	*177760,R0	;MASK TO GAIN
250	001332	006000		CLC		;CLEAR CARRY BIT
251	001334	103001		ROR	R0	
252	001336	005200		BCC	10\$	
253	001340			INC	R0	;CONVERT GAIN TO UNIT VALUE
254	001340	006300		10\$:		
255	001342	006300		ASL	R0	;MULTIPLY BY 4
256	001344	062700	001650'	ASL	R0	
257	001350	012067	001732	ADD	*G.TBL,R0	;GAIN TABLE POINTER
258	001354	012067	001730	MOV	(R0)+,GAIN	;SAVE GAIN VALUE
259	001360	000207		MOV	(R0)+,GAIN+2	
261				RETURN		
262				;		
263				;	TE2.20	READ A BLOCK OF THE DATA FILE INTO CORE
264	001362			;		
265	001362			TE2.20:		
266	001374			FDAT\$R	*TE2FDB,...*1000	;SET BLOCK SIZE
267	001402			FDOP\$R	R0,...*FO.UPDIFA.SHR	;OPEN FOR UPDATE AND SHARED
				OPNS\$U	R0,...,50\$	;OPEN THE FILE

```

268 001422                                READ$  R0,*IOAREA,,*VBNADR,,*,ERR.03 ;READ A BLOCK
269 001464                                WAIT$  R0,,*,ERR.03
270 001476 000207                                RETURN
271 001500                                50$:                                ;HERE IF ERROR OPENING FILE
272 001500 126027 000052 00000000                                CMPB  F.ERR(R0),*IE.WAC ;FILE ALREADY OPEN FOR A WRITE ?
273 001506 001402                                BEQ   60$ ;WAIT ON OPEN FILE
274 001510 000167 000100                                JMP   ERR.02 ;OTHERWISE ERROR
275 001514                                60$:
276 001514                                USIG$S ;WAIT FOR IT TO CLOSE
277 001522 000717                                BR    TE2.20
278
279 ;*****
280 ;AFTER A READ THE USER MUST PERFORM A CLOSE
281 ;*****
282
283
284
285
286 ;+
287 ;TE2.25                                WRITE A BLOCK OF DATA OUT TO DISK
288 ;-
289
290 001524                                TE2.25:
291 001524                                WRITE$ *TE2FDB,*IOAREA,,*VBNADR,,*,ERR.04 ;WRITE A BLOCK
292 001572                                WAIT$  R0,,*,ERR.04 ;WAIT ON COMPLETION
293 001604 000207                                RETURN ;BACK TO CALLER
294
295
296
297 001606                                ERR.01:
298 001606 012700 003543'                                MOV   *MSG3,R0 ;MARK TIME ERROR
299 001612 000413                                BR    ERR0UT
300 001614                                ERR.02:
301 001614 012700 003570'                                MOV   *MSG4,R0 ;OPEN FILE ERROR
302 001620 000410                                BR    ERR0UT
303 001622                                ERR.03:
304 001622 012700 003613'                                MOV   *MSG5,R0 ;READ FILE ERROR
305 001626 000405                                BR    ERR0UT
306 001630                                ERR.04:
307 001630 012700 003640'                                MOV   *MSG6,R0 ;WRITE FILE ERROR
308 001634 000402                                BR    ERR0UT
309 001636                                ERR.20:
310 001636 012700 003666'                                MOV   *MSG7,R0 ;A/D CONVERSION ERROR
311

```

312	001642			ERROUT:			
313	001642	005726			TST	(SP)+	;SET FOR RETURN TO MAIN PROGRAM
314	001644	000167	177372		JMP	TE2.09	;GO TERMINATE
316				;+			
317				;STORAGE AREA			
318				;-			
319	001650			G.TBL:			;A/D GAIN TABLE
320	001650	040200	000000		.FLT2	1.0	;GAIN OF 1
321	001654	040400	000000		.FLT2	2.0	
322	001660	041040	000000		.FLT2	10.0	
323	001664	041240	000000		.FLT2	20.0	
324	001670	041510	000000		.FLT2	50.0	
325	001674	041710	000000		.FLT2	100.0	
326	001700	042110	000000		.FLT2	200.0	
327	001704	042572	000000		.FLT2	1000.0	
328				;			
329	001710			RCVBUF:	.BLKW	15.	;RECEIVE DIRECTIVE BUFFER
330				;			
331	001746			QIOBLK:	QIOWS	10.WLB,TILUN,TIEFH,,IOSTAT,,<MSG1,MS1S12>	
332				;			
333	001776			MRKTIM:	MRKTS	MRKEFN,1,3	;SET FOR 1 MINUTE
334				;			
335	002010				FSRSZ%	0	
336	002010			TE2FDB:	FDBDF%		
337	002150				FDRCS%	FD.RWM	
338	002150				FDBK%	IOAREA,1000,,DKEFN,DKSTAT	
339	002150				FDOP%	DKLUN,DSDES,,FD.MFY	
340				;			
341	002150			TIMBER:	.BLKW	0.	
342				;			
343	002170			CYCEND:	.BLKW	1	;SET AT END OF TESTING
344				;			
345	002172			IOSTAT:			
346	002172			DKSTAT:	.BLKW	2	;I/O STATUS WORDS
347				;			
348	002176			DSDES:			;FDB DATASET DESCRIPTOR
349	002176	000004	002212		4,DRCT		;DEVICE
350	002202	000011	002216		9,,DUIC		;DIRECTORY
351	002206	000000	002227		0,DFILE		;FILE NAME
352	002212	104	113	061	DRCT:	.ASCII	/DK1:/
	002215	072					
353	002216	133	063	060	DUIC:	.ASCII	/1300,3001/
	002221	060	054	063			
	002224	060	060	135			
354	002227				DFILE:	.BLKB	16.


```

355                                     .EVEN
356
357 002250 000000 000001      ; VBNADR: .WORD 0,1      ; VBN FOR DK ACCESS
358
359 002254      ; LOOPFG: .BLKW 1      ; IF SET THEN TASK IS RUNNING
360
361 002256      ; IOAREA: .BLKB 1000      ; DK INPUT/OUTPUT BUFFER
362
363 003256      012      000      ; LN2LST: .BYTE 10,,0      ; FORTRAN LIST FOR LN2 SERVICE
364 003260 003304' 003306' 003312'      ; IVALUE,GAIN,NTURNS
365 003266 003314' 003316' 003320'      ; LN2LUN,LN2PSC,LN2LNS
366 003274 003322' 003324' 003326'      ; LN2LNC,LN2ERR,IVAL2
367 003302 003330'      ; GAIN2
368
369 003304 000000      ; IVALUE: .WORD 0      ; MAGNITUDE OF A/D READING SPLTMP
370
371 003306      ; GAIN: .BLKW 2      ; FLOATING POINT GAIN VALUE SPLTMP
372
373 003312 000000      ; NTURNS: .WORD 0      ; CURRENT LN2 1/4 TURNS OPEN
374
375 003314 000002      ; LN2LUN: .WORD 1CSLUN      ; ICS LUN
376
377 003316 000000      ; LN2PSC: .WORD 0      ; POWER SUPPLY RELAY *
378
379 003320 000000      ; LN2LNS: .WORD 0      ; LN2 SOLENOID RELAY *
380
381 003322 000000      ; LN2LNC: .WORD 0      ; LN2 STEPPER MOTOR RELAY *1
382
383 003324 000000      ; LN2ERR: .WORD 0      ; ERROR RETURN
384
385 003326 000000      ; IVAL2: .WORD 0      ; MAGNITUDE OF A/D READING HEATER VOLTAGE
386
387 003330      ; GAIN2: .BLKW 2      ; FLOATING POINT GAIN FOR HEATER VOLTAGE
388
389
390
391 003334 000000      ; IVLHD: .WORD 0      ; DUMMY HOLDING SPLTMP VALUE
392
393 003336      ; GAINHD: .BLKW 2      ; DUMMY HOLDING SPLTMP GAIN (F.P.)
394
395
396
397 003342      ; ADLIST:      ; FORTRAN LIST FOR ADREAD
398 003342      004      000      ; .BYTE 4,0
399 003344 003356' 003366' 003370'      ; ADSTC,MCHAN,STADV

```

400	003352	003324	003314	LNZERR, LN2LUN	;ERROR AND ICS LUN
401	003356	000000		ADSTC: .WORD 0	;A/D CHANNEL FOR SPECIMEN TEMP
402	003360	000000		ADHTC: .WORD 0	;HEATER TEMP A/D CHANNEL
403	003362	000000		ADLVC: .WORD 0	;LVDT A/D CHANNEL
404	003364	000000		ADHVC: .WORD 0	;HEATER VOLTAGE A/D CHANNEL
405				;	
406	003366	000004		NCHAN: .WORD 4	;NUMBER OF INPUT A/D CHANNELS
407				;	
408	003370	150000		STADV: .WORD 150000	;A/D VALUE FOR SPECIMEN TEMP (10MV GAIN)
409	003372	150000		HTADV: .WORD 150000	;A/D VALUE FOR HEATER TEMP
410	003374	150000		LVADV: .WORD 150000	;A/D VALUE FOR LVDT
411	003376	040000		HVADV: .WORD 40000	;A/D VALUE FOR HEATER VOLTAGE (5 V GAIN)
412				;	
413				;	
414	003400			CYCLST:	;FORTRAN LIST FOR CYCLE LOGIC
415	003400	007	000	.BYTE 7,0	
416	003402	003304	003306	IVALUE, GAIN, CYCSWG	
417	003410	003420	003424	CYCTEM, IC8YT, INEW	
418				;	
419	003416			CYCSWG: .BLKW 1	;INITIAL CYCLE SWING
420				;	
421	003420			CYCTEM: .BLKW 2	;STARTING CYCLE TEMPERATURE
422				;	
423	003424			IC8YT: .BLKW 1	;CURRENT CYCLE POSITION
424				;	
425	003426			INEW: .BLKW 1	;IF SET, NEW CYCLE STARTING
426	003430			CLSLSST:	;MACRO LIST TO CLOSE SOLENOID
427	003430	002	000	.BYTE 2,0	
428	003432	003320	003314	LN2LMS, LN2LUN	
430				;	
431				;	
432				;	
433				;	
434	003436	MSG1:		.NLIST BIN	
435	003440			.BYTE 15,12	
436	003510			.ASCII /TE2000 - NO DATA RECEIVED FROM INITIATOR/	
437				.BYTE 15,12	
438				MSISIZ = .-MSG1	
439	003512	MSG2:		.BYTE <MSG3-MSG2>, 15,12	
440	003515			.ASCII /TE2000 - END OF TEST/	
441	003541			.BYTE 15,12	
442	003543	MSG3:		.BYTE <MSG4-MSG3>, 15,12	
443	003546			.ASCII /TE2000 - MARK TIME/	
444	003570	MSG4:		.BYTE <MSG5-MSG4>, 15,12	

```
445 003573      .ASCII  /TE2000-OPEN FILE/
446 003613 MSG5:  .BYTE   <MSG6-MSG5>,15,12
447 003616      .ASCII  /TE2000 - READ FILE/
448 003640 MSG6:  .BYTE   <MSG7-MSG6>,15,12
449 003643      .ASCII  /TE2000 - WRITE FILE/
450 003666 MSG7:  .BYTE   <MSG8-MSG7>,15,12
451 003671      .ASCII  ?TE2000 - A/D ERR?
452 003711 MSG8:
453              .EVEN
454              .END
```

Appendix B

**DATA REDUCTION AND PLOTTING
SOFTWARE LISTINGS**

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

```

0001          PROGRAM TE2000
              .IDENT/050070/
C
C+
C      FEB. 1978          D. E. LACEY AND L. L. HOLLAND
C      MAY 1978          LATEST REVISION          D. E. LACEY.
C
C      THIS IS THE MAIN PROGRAM FOR THE TE2000 THERMAL
C      EXPANSION EXPERIMENT.  OPTIONAL SUBROUTINES AND
C      TASKS THAT DO I/O FUNCTIONS, DATA REDUCTION AND
C      PLOTTING ARE SELECTED AS REQUIRED.
C
C-
0002      REAL*4 TASK
0003      LOGICAL*1 FILNAM(26),ID(2),EXT(4),LTR(7),ANS(1),
C      EMPNO(6),EMPLN(6),OPT(7)
C
0004      EQUIVALENCE (EMPLN,EMPNO(1))
C
0005      DATA FILNAM/'D','K','I',' ',' ','C','3','0','0',' ',' ','3','0','0',' ','J',
C      13*1H /
0006      DATA EXT/' ','T','E','2'//, 1Y/1HY/
0007      DATA OPT/'H','P','N','B','T','G','E'/
0008      DATA TASK /6RTEPLOT/
C
C
0009      CALL ATTDEV
0010      WRITE(5,125)
0011      READ(5,107) IANS
0012      IF (IANS.NE.IY) GO TO 1
0014      CALL TIMING
0015      WRITE(5,130)
0016      READ(5,107) IANS
0017      IF (IANS.EQ.IY) CALL EXIT
0019  1      WRITE(5,100)
0020      READ(5,101) K1,LTR
0021      WRITE(5,105)
0022      READ(5,106) (ID(I), I=1,2)
C
C      FILNAM IS DK1:0300,300JLTR01D.TE2
C
0023      K=14
0024      DO 5 J=1,K1
0025          FILNAM(K)=LTR(J)
0026  5      K=K+1
0027      DO 6 J=1,2

```

```

0028      FILNAM(K)=ID(J)
0029 6      K=K+1
0030      DO 7 J=1,4
0031      FILNAM(K)=EXT(J)
0032 7      K=K+1
0033      K=K-1
      C
      C      CALL LPPRNT TO SEPARATE DATA INTO INDIVIDUAL
      C      THERMOCYCLES. WRITE THE DATA NEEDED BY TEPLT TO A TEMPORARY
      C      DISK FILE CALLED DK1:[300,300]PARAMS.DAT
      C
0034      CALL LPPRNT(FILNAM,K,1,RFTEMP,EMPNO)
0035      CALL ASSIGN(1,'DK1:[300,300]PARAMS.DAT')
0036      WRITE(1,133) K1
0037      WRITE(1,106) (ID(I),I=1,2)
0038      WRITE(1,135) (LTR(I),I=1,K1)
0039      WRITE(1,140) RFTEMP
0040      WRITE(1,133) K
0041      WRITE(1,145) (FILNAM(I),I=1,K)
0042      WRITE(1,155) (EMPLN(I),I=1,5)
0043      CALL CLOSE(1)
0044      WRITE(5,110)
0045      READ(5,111) ANS
0046 10      DO 15 J=1,7
0047          IF (ANS(1).EQ.OPT(J)) GO TO 20
0049 15      CONTINUE
0050      WRITE(5,120)
0051      READ(5,111) ANS
0052      GO TO 10
0053 20      GO TO (21,22,23,24,25,26,27), J
0054 21      CALL HEADER(FILNAM,K)
0055      GO TO 35
0056 22      CALL LPPRNT(FILNAM,K,0,RFTEMP,EMPNO)
0057      GO TO 35
0058 23      CALL NEWPR(FILNAM,K)
0059      GO TO 35
0060 24      CALL BETWEN(FILNAM,K)
0061      GO TO 35
0062 25      CALL TIMINC
0063      GO TO 35
0064 26      CALL DETDEV
0065      CALL REQUES(TASK)
0066      CALL EXIT
      C
      C      ERASE ALL DK1:[300,300]THMEXP.DAT FILES BEFORE EXIT

```

```

C
0067 27 CALL ASSIGN(1,'DK1:[300,300]THMEXP.DAT;1')
0068 WRITE(1,150)
0069 CALL DELFLE(1,IERR)
0070 CALL ASSIGN(1,'DK1:[300,300]THMEXP.DAT;2')
0071 WRITE(1,150)
0072 CALL DELFLE(1,IERR)
0073 CALL ASSIGN(1,'DK1:[300,300]PARAMS.DAT;1')
0074 WRITE(1,150)
0075 CALL DELFLE(1,IERR)
0076 CALL EXIT
0077 35 WRITE(5,115)
0078 READ(5,111) ANS
0079 GO TO 10

C
C FORMAT STATEMENTS.
C
0080 100 FORMAT('SLTR NUMBER? [7A1] ')
0081 101 FORMAT(0,7A1)
0082 105 FORMAT('$SPECIMEN IDENTIFICATION? [2A1] ')
0083 106 FORMAT(2A1)
0084 107 FORMAT(A1)
0085 110 FORMAT(' SELECT ONE OF THE FOLLOWING OPTIONS: ',/
1 ' HEADER (H), PRINT ALL DATA (P), PRINT ALL NEW DATA (N),'/1X,
2 'PRINT ALL DATA BETWEEN TWO RECORDS (B),CHANGE TIME INCREMENT'/
3 '$(T), GRAPH % EXP. VS. TEMP (G), EXIT (E): ')
0086 111 FORMAT(A1)
0087 115 FORMAT('$SELECT ANOTHER OPTION: ')
0088 120 FORMAT('$INPUT ERROR, SELECT ANOTHER OPTION: ')
0089 125 FORMAT('$TIME INCREMENT CHANGE? [Y/N] ')
0090 130 FORMAT('$EXIT PROGRAM? [Y/N] ')
0091 133 FORMAT(13)
0092 135 FORMAT(7A1)
0093 140 FORMAT(F5.1)
0094 145 FORMAT(26A1)
0095 150 FORMAT(1X,'ERASE FILE.')
0096 155 FORMAT(5A1)
0097 END

```



```

0001      SUBROUTINE LPPRNT(FILNAM,KK,JFLAG,RFTMP,EMPNO)
          .IDENT/051578/
          C
          C+
          C      MAR. 1978                      D. E. LACEY
          C      APR 1978      REVISED          L. L. HOLLAND
          C      MAY 1978      LATEST REVISION    D. E. LACEY
          C
          C      THIS SUBROUTINE OPTIONALLY PRINTS THE HEADER INFORMATION
          C      FOR THE TE2000 EXPERIMENT, THEN PRINTS ALL COLLECTED DATA
          C      AND NEW DATA, ALONG WITH THE REDUCED DATA. THE DISK DATAFILE
          C      IS CLOSED IMMEDIATELY FOLLOWING EACH READ.
          C
          C      UPON EXIT TO THE MAIN PROGRAM, THE USER MAY SELECT AN ADD-
          C      ITIONAL OPTION OR EXIT FROM THE PROGRAM.
          C
          C      THIS PROGRAM ALSO SEPARATES THE DATA INTO INDIVIDUAL
          C      THERMOCYCLES WHEN JFLAG=1.
          C-
0002      DIMENSION TUP(200),EXUP(200),IHOR(8)
0003      REAL*4 GAIN(4),TEMP(4),VOLTS(4),XMR
          C
0004      INTEGER*2 IADW(4),MAG(4),IWRD(6),IDUM(6)
          C
0005      LOGICAL*1 FILNAM(26),INFLE(14),IHEDR(64),XLEN1(4),ENO(6)
          1      ,KHOD(16),EMPNO(6),HOUR(8),XTEM1(4),ARCHVE(26),
          2      DUM1(26),DUM2(9),DUM3(1),IDATE(9)
          C
0006      EQUIVALENCE (IWRD,INFLE(1)),(XMR,IDUM(1)),
          1      (XLEN,XLEN1(1)),(XTEM,XTEM1(1)),(ENO,KHOD(16))
          C
0007      DATA IY/IHY/, JD/IHD/
          C
0008      CALL DATE(IDATE)
          C
          C      JFLAG = 0 SIGNALS THAT DATA HAS ALREADY BEEN SEPARATED INTO
          C      THERMOCYCLES, AND THAT DATA CAN NOW BE PRINTED.
          C
0009      IF(JFLAG.EQ.0)GOTO 1
          C
          C      ICYCLE IS THE NUMBER OF THE THERMOCYCLE THAT DATA IS BEING
          C      STORED IN CURRENTLY.
          C
0011      ICYCLE=1
          C

```

```

      C      WRITE THE DATA TO A SEPARATE VERSION OF THMEXP.DAT
      C      FOR EACH THERMOCYCLE.
      C
0012      CALL ASSIGN(1,'DK1:[300,300]THMEXP.DAT')
0013      WRITE(1,152)(FILNAM(I),I=1,KK)
0014      WRITE(1,152)(IDATE(I),I=1,9)
0015      WRITE(1,160)ICYCLE
0016      GOTO 6
0017 1      IO=5
0018      WRITE(5,115)
0019      READ(5,101) IANS
0020      IF (IANS.EQ.IY) IO=4
      C
      C      OUTPUT: 4=LINE PRINTER
      C      5=TERMINAL
      C      3=ARCHIVAL
      C
0022      IF (IO.EQ.4) GO TO 4
0024      WRITE(5,116)
0025      READ(5,101) IANS
0026      IF (IANS.NE.IY) GO TO 4
0028      IO=3
      C
      C      ASSIGN LUN TO ARCHIVAL FILE
      C
0029      DO 2 IL=1,KK-1
0030          ARCHVE(IL)=FILNAM(IL)
0031 2      CONTINUE
0032      ARCHVE(KK)=JD
0033      CALL ASSIGN(3,ARCHVE,KK)
      C
      C      WRITE HEADER?
      C
0034 4      WRITE(5,110)
0035      READ (5,101) IANS
0036      IF (IANS.NE.IY) GOTO 6
0038      CALL HEADER(FILNAM,KK)
      C
      C      IFLAG=0 SIGNALS THAT CROSSOVER TEMPERATURE HAS NOT BEEN REACHED.
      C      DO NOT CALCULATE X EXPANSION.
      C
0039 6      IFLAG=0
      C
      C      OPEN FILE TO READ HEADER INFO.
      C

```

```

0040      CALL ASSIGN(2,FILNAM,KK)
0041      CALL FDBSET(2,'R','SHARE')
0042      DEFINE FILE 2(4,8,U,NRECRD)
0043      READ(2'1)(KMOD(I),I=1,16)
0044      READ(2'2)(INFLE(I),I=1,14)
0045      READ(2'4)(IDUM(I),I=1,6)
0046      CALL CLOSE(2)
C
0047      DO 8 I=1,6
0048          EMPNO(I)=END(I)
0049      8      CONTINUE
0050      NREC=IDUM(3)+4
0051      ISWG=INFLE(14)
0052      DO 10 IJK=1,4
0053          XLEN(IJK)=INFLE(IJK+4)
0054      10      XTEM(IJK)=INFLE(IJK+8)
0055      RFTMP=XTEM
C
C      INITIALIZE COUNTERS
C      IPAGE IS THE PAGE NUMBER
C      JCONT CONTROLS NUMBER OF LINES PRINTED PER PAGE
C      ICOUNT IS COUNTER SPECIFYING EVERY 5TH LINE HAVE not'S
C      IFLAG1 IS THE NUMBER OF TIMES THE REF. TEMP. IS CROSSED.
C      NTTLP IS THE NUMBER OF TIMES CONTROL HAS PASSED THRU LOOP
C
0056      IPAGE=0
0057      JCONT=0
0058      ICOUNT=0
0059      IFLAG1=0
0060      NTTLP=0
C
C      RFVOLT DETERMINES THE LVDT VOLTAGE AT THE REFERENCE TEMP.
C
0061      CALL RFVOLT(FILNAM,KK,XTEM,XLEN,VREF,ISWG,XMR)
0062      DO 36 J=5,NREC
0063          DEFINE FILE 2 (NREC,8,U,NRECRD)
0064          CALL ASSIGN(2,FILNAM,KK)
0065          CALL FDBSET(2,'R','SHARE')
0066          READ(2'J,END=40,ERR=40)(1WRD(I),I=1,6)
0067          CALL CLOSE(2)
0068          IHR=INFLE(1)
0069          MIN=INFLE(2)
C
C      IDLN IS THE NUMBER OF QTR TURNS THE LIQUID NITROGEN

```

```

C      VALVE IS OPEN.IDXD IS THE NUMBER OF TIMES THE DATA ACQUISITION
C      PROGRAM HAS PASSED THE REF. TEMP.
C
0070      IDLN=INFLE(11)
0071      IDXD=INFLE(12)
0072      DO 15 K=1,3
0073 15      IADW(K)=IWRD(K+1)
C
C      SEPARATE READING INTO GAIN AND MAGNITUDE, THEN CALCULATE
C      VOLTAGE FROM MAG. AND GAIN
C
0074      CALL ADC(IADW(1),MAG(1),GAIN(1))
0075      DO 17 K=1,3
0076 17      VOLTS(K)=4.8852*FLOAT(MAG(K))
1          /GAIN(K)
C
C      FIND TEMP IN DEGREES F FOR SAMPLE AND HEATER.
C
0077      CALL TEMPF(VOLTS(1),HTRTMP)
0078      CALL TEMPF(VOLTS(2),SPLTMP)
C
C      VOLTS(3) IS LVDT VOLTAGE
C
0079      VOLTS(3)=VOLTS(3)*MMVR
C
C      IFLAG AND JFLAG DETERMINE WHETHER OR NOT % EXPANSIONS
C      NEED TO BE CALCULATED THIS TIME THRU LOOP.
C
0080      IF(JFLAG.EQ.1.AND.1FLAG.NE.0)GOTO 25
0082      IF(JFLAG.EQ.1)GOTO 21
0084      IF (JCONT.NE.0) GO TO 20
0086      JCONT=45
0087      IPAGE=IPAGE+1
C
C      READ TIME AND STIP PADDING FROM EACH BYTE.
C
0088      CALL TIME(HOUR)
0089      DO 19 I=1,8
0090 19      IHOR(I)=HOUR(I)
0091      WRITE(10,119)(FILNAM(I),I=1,KK)
0092      WRITE(10,120)(IDATE(I),I=1,9),(IHOR(I),I=1,8)
1          , IPAGE
0093      WRITE(10,121)
0094      WRITE(10,122)
0095 20      ICOUNT=ICOUNT+1

```

```

0096      IF (IFLAG.NE.0) GO TO 25
0098 21    IF (ISUG.EQ.0.AND.XTEM.LT.SPLTMP) GO TO 28
0100      IF (ISUG.EQ.1.AND.SPLTMP.LT.XTEM) GO TO 28
0102      IFLAG=1
      C    CORRECTION IS FOR QUARTZ
0103 25    CORRECT=.000031*(SPLTMP-XTEM)
0104      RAWEXP=XTMP*(VOLTS(3)-VREF)/XLEN*.1
0105      EXPNSN=RAWEXP+CORRECT
0106      IF (JFLAG.EQ.0) GO TO 27
0108      IF (IDXD.EQ.1) GO TO 26
0110      IFLAG1=1
0111      NTTLP=NTTLP+1
0112      IF (NTTLP.NE.2) GO TO 26
0114      CALL CLOSE(1)
      C
      C    OPEN FILE FOR SECOND THERMOCYCLE.
      C
0115      CALL ASSIGN(1,'DK1:C300,300JTHMEXP.DAT')
0116      WRITE(1,152)(FILNAM(1),I=1,KK)
0117      WRITE(1,152)(IDATE(1),I=1,9)
0118      ICYCLE=ICYCLE+1
0119      WRITE(1,150)ICYCLE
0120      IFLAG1=0
0121 26    WRITE(1,155) EXPNSN,SPLTMP
0122      GO TO 36
0123 27    IF (ICOUNT.NE.5) GO TO 32
0125      ICOUNT=0
0126      WRITE(10,130) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
      1      IDLN,RAWEXP,EXPNSN
0127      GO TO 34
0128 28    IF (JFLAG.EQ.1) GO TO 36
0130      IF (ICOUNT.NE.5) GO TO 29
0132      ICOUNT=0
0133      WRITE(10,140) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
      1      IDLN
0134      GO TO 34
0135 29    WRITE(10,135) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
      1      IDLN
0136      GO TO 34
0137 32    WRITE(10,125) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
      1      IDLN,RAWEXP,EXPNSN
0138 34    JCONT=JCONT-1
0139 36    CONTINUE
0140 40    CALL CLOSE(1)
      C

```

```

      C      IF NOT ARCHIVING DATA, SKIP THIS.
      C
0141      IF (IO.NE.3) GOTO 95
      C
0143      CALL ASSIGN(1,'DK1:(300,300)THMEXP.DAT;1')
0144      READ(1,152)DUM1
0145      READ(1,152)DUM2
0146      READ(1,152)DUM3
0147      WRITE(3,165)DUM2,DUM3,DUM1
      C
      C      ARCHIVE CORRECTED DATA?
      C
0148      WRITE(5,175)
0149      READ(5,101) IANS
0150      IF (IANS.NE.IY) GO TO 45
      C
      C      LEDLAG CORRECTS EACH THERMOCYCLE OF DATA FOR LEAD-LAG.
      C
0152      CALL LEDLAG(RFTEMP,TUP,EXUP,JCOUNT)
0153      WRITE(3,166)
      C
0154      DO 42 I=1,JCOUNT
0155 42      WRITE(3,170)TUP(I),EXUP(I)
0156      CALL CLOSE(1)
0157      WRITE(5,105)
0158      READ(5,101) IANS
0159      IF (IANS.NE.IY) GO TO 90
0161      CALL ASSIGN (1,'DK1:(300,300)THMEXP.DAT;2')
0162      READ(1,152)DUM1
0163      READ(1,152)DUM2
0164      READ(1,152)DUM3
0165      CALL LEDLAG(RFTEMP,TUP,EXUP,JCOUNT)
0166      WRITE(3,165)DUM2,DUM3,DUM1
0167      WRITE(3,166)
0168      DO 43 I=-1,JCOUNT
0169 43      WRITE(3,170) TUP(I),EXUP(I)
0170      CALL CLOSE(1)
0171      GO TO 90
0172 45      WRITE(3,176)
0173      DO 47 JKI=1,300
0174          READ(1,177,END=50,ERR=50)EXP,TMP
0175 47      WRITE(3,177)EXP,TMP
0176 50      CALL CLOSE(1)
0177      CALL ASSIGN(1,'DK1:(300,300)THMEXP.DAT;2')
0178      READ(1,152)DUM1

```

```

0179      READ(1,152)DUM2
0180      READ(1,152)DUM3
0181      WRITE(3,165)DUM2,DUM3,DUM1
0182      WRITE(3,176)
0183      DO 52 JKI=1,300
0184          READ(1,177,END=60,ERR=60)EXP,TMP
0185      52      WRITE(3,177)EXP,TMP
0186      60      CALL CLOSE(1)
0187      90      WRITE(3,180)
0188              WRITE(3,181)
0189              WRITE(3,182)
0190              WRITE(3,183)
0191              WRITE(3,184)
0192              WRITE(3,185)
0193              WRITE(3,186)
0194              WRITE(3,187)
0195              WRITE(3,188)
0196              WRITE(3,189)
0197      CALL CLOSE(3)
0198      95      RETURN
C
C      FORMAT STATEMENTS.
C
0199      101     FORMAT(A1)
0200      105     FORMAT('SARCHIVE THERMOCYCLE #27 [Y/N]: ')
0201      110     FORMAT('SPRINT HEADER? [Y/N]: ')
0202      115     FORMAT('SOUTPUT DATA TO THE LINE PRINTER? [Y/N]: ')
0203      116     FORMAT('SARCHIVE DATA? [Y/N]: ')
0204      117     FORMAT('SFILENAME? [26A1]: ')
0205      119     FORMAT('1',19X,26A1)
0206      120     FORMAT(/1X,9A1,10X,8A1,15X,'PAGE ',13)
0207      121     FORMAT(/3X,'TIME',4X,'HEATER',3X,'SPECIMEN',4X,
1 'LVDT',5X,'LN2 QTR.',5X,' % EXPANSION')
0208      122     FORMAT(1X,'HRS-MINS',3X,'TEMP.',5X,'TEMP.',3X,
2 'VOLTAGE',1X,'TURNS OPEN',4X,'RAW',4X,'COMPENSATED',/)
0209      125     FORMAT(2X,12,':',12,4X,F6.2,4X,F6.2,4X,F6.2,
1 4X,12,8X,F6.4,5X,F6.4)
0210      130     FORMAT(' *',12,':',12,' **',F6.2,' **',F6.2,' **',
1 ',F6.2,' ***',12,' ****',F6.4,' **',2X,F6.4)
0211      135     FORMAT(2X,12,':',12,4X,F6.2,4X,F6.2,4X,F6.2,
1 4X,12)
0212      140     FORMAT(' *',12,':',12,' **',F6.2,' **',F6.2,' **',
1 ',F6.2,' ***',12)
0213      150     FORMAT(12A1)
0214      152     FORMAT(26A1)

```

```

0001          SUBROUTINE BETWEN(FILNAM,KK)
              ,IDENT/050978/
              C
              C+
              C      FEB. 1978                      D. E. LACEY
              C      MAY 1978      REVISED          L. L. HOLLDAND
              C      MAY 1978      LATEST REVISION    D. E. LACEY
              C
              C      THIS SUBROUTINE OPTIONALLY PRINTS THE HEADER INFORMATION
              C      FOR THE TE2000 EXPERIMENT, THEN PRINTS COLLECTED DATA
              C      BETWEEN TWO USER SPECIFIED RECORD NUMBERS.  THE DISK DATA
              C      FILE IS CLOSED IMMEDIATELY FOLLOWING THE EACH READ.
              C      UPON EXIT TO THE MAIN PROGRAM, THE USER MAY SELECT AN ADD-
              C      ITIONAL OUTPUT OPTIONS OR EXIT FROM THE PROGRAM.
              C
              C-
              C
0002          DIMENSION IHOR(8)
              C
0003          REAL*4 GAIN(4),TEMP(4),VOLTS(4),XMYR
              C
0004          INTEGER*2 INFILE(6),IADW(4),MAG(4),IWRD(6)
              C
0005          LOGICAL*1 FILNAM(26),INFLE(14),XLEN1(4),XTEM1(4),
              I      HOUR(8),IDATE(9)
              C
0006          EQUIVALENCE (INFILE,INFLE(1)),(XMYR,IWRD(1)),
              I      (XLEN,XLEN1(1)),(XTEM,XTEM1(1))
              C
              C
0007          DATA IY/IHY/
0008          CALL DATE(IDATE)
0009          IO=5
              C
              C      WRITE HEADER?
              C
0010          WRITE(5,100)
0011          READ(5,101) IANS
0012          IF (IANS.NE.IY) GO TO 5
0014          CALL HEADER(FILNAM,KK)
              C
              C      OUTPUT:  4=LINE PRINTER  5=TERMINAL
              C      ISTART IS THE BEGINNING RECORD NUMBER
              C      IEND   IS THE END RECORD NUMBER
              C
0015          5  WRITE(5,105)

```



```

0016      READ(5,106) ISTART
0017      WRITE(5,110)
0018      READ(5,106) IEND
0019      WRITE(5,115)
0020      READ(5,101) IANS
0021      IF (IANS.EQ.IY) IO=4
      C
      C      READ HEADER INFORMATION
      C
0023      CALL ASSIGN(2,FILNAM,KK)
0024      CALL FDBSET(2,'R','SHARE')
0025      DEFINE FILE 2(5,8,U,NRECRD)
0026      READ(2*2) (INFLE(I),I=1,14)
0027      READ(2*4) (IWRD(I),I=1,6)
0028      CALL CLOSE(2)
      C
      C      NREC IS THE NUMBER OF RECORDS RECORDED.
      C
0029      NREC=IWRD(3)+4
      C
      C      ISWG IS THE SWING DIRECTION OF THE DATA:  B=DOWN,
      C      I=UP.
      C
0030      ISWG=INFLE(14)
0031      DO 7 IJK=1,4
0032          XLEN(IJK)=INFLE(IJK+4)
0033          XTEM(IJK)=INFLE(IJK+8)
0034      ?      CONTINUE
      C
      C      ADD 4 TO THE RECORD NUMBER TO SKIP OVER THE HEADER INFO.
      C
0035      ISTART=ISTART+4
0036      IEND=IEND+4
      C
      C      RFVOLT DETERMINES THE LVDT VOLTAGE AT THE REFERENCE TEMPERATURE.
      C
0037      CALL RFVOLT(FILNAM,KK,XTEM,XLEN,VREF,ISWG,XMVR)
      C
      C      INITIALIZE COUNTERS.
      C
0038      IPAGE=0
0039      ICOUNT=0
0040      JCONT=0
0041      DO 30 J=ISTART,IEND
0042          DEFINE FILE 2 (NREC,8,U,NRECRD)

```

```

0043      CALL ASSIGN(2,FILNAM,KK)
0044      CALL FDBSET(2,'R','SHARE')
0045      READ(2'J,END=40,ERR=40)(INFLE(I),I=1,6)
0046      CALL CLOSE(2)
0047      IHR=INFLE(1)
0048      MIN=INFLE(2)
0049      IDLN=INFLE(11)
0050      DO 20 K=1,3
0051      20      IADW(K)=INFLE(K+1)
      C
      C      DO THE ANALOG TO DIGITAL CONVERSION.  GET A MAGNITUDE
      C      AND A GAIN FOR EACH OF THE READINGS.
      C
0052      CALL ADC(IADW(1),MAG(1),GAIN(1))
      C
      C      CONVERT MAGNITUDES AND GAINS TO VOLTAGE READINGS.
      C
0053      DO 21 K=1,3
0054      21      VOLTS(K)=4.8852*FLOAT(MAG(K))/GAIN(K)
      C
      C      CONVERT VOLTAGES TO HEATER AND SAMPLE TEMPERATURES IN
      C      DEGREES F.
      C
0055      CALL TEMPF(VOLTS(1),HTRTMP)
0056      CALL TEMPF(VOLTS(2),SPLTMP)
      C
      C      VOLTS(3) IS THE LVDT VOLTAGE.
      C
0057      VOLTS(3)=VOLTS(3)*MMVR
      C
      C      JCONT CONTROLS THE NUMBER OF LINES PRINTED PER PAGE.
      C
0058      IF (JCONT.NE.0) GOTO 24
0059      JCONT=45
      C
      C      IPAGE IS THE CURRENT PAGE NUMBER.
      C
0061      IPAGE=IPAGE+1
      C
      C      READ THE CURRENT TIME AND STRIP THE PADDING FROM EACH BYTE.
      C
0062      CALL TIME(HOUR)
0063      DO 22 I=1,8
0064      22      IHOR(I)=HOUR(I)
0065      WRITE(IO,119)(FILNAM(I),I=1,KK)

```

```

0066      WRITE(10,120) (IDATE(I), I=1,9), (IHOR(I),
1          I=1,8), IPAGE
0067      WRITE(10,121)
0068      WRITE(10,122)
0069      ICOUNT=ICOUNT+1
      C
      C
      C      IFLAG = 0 SIGNALS THAT THE CROSSOVER TEMPERATURE HAS NOT
      C      BEEN REACHED, AND THAT % EXPANSION IS NOT CALCULATED.
      C
0070      IF(IFLAG.NE.0)GOTO 25
      C
      C      CHECK FOR CROSSOVER TEMPERATURE.
      C
0072      IF(ISWG.EQ.0.AND.XTEM.LT.SPLTMP)GOTO 28
0074      IF(ISWG.EQ.1.AND.SPLTMP.LT.XTEM) GO TO 28
0076      IFLAG=1
      C
      C      THIS CORRECTION IS FOR QUARTZ.
      C
0077      25      CORECT=.00031*(SPLTMP-XTEM)
0078      RAWEXP=XTRV*(VOLTS(3)-VREF)/XLEN*.1
0079      EXPNSN=RAWEXP+CORECT
      C
      C      ICOUNT = 5 SIGNALS THAT *****S ARE TO PRINTED ON
      C      EVERY FIFTH LINE OF DATA.
      C
0080      IF (ICOUNT.NE.5) GOTO 29
0082      ICOUNT=0
0083      WRITE(10,130) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
1          IDLN,RAWEXP,EXPNSN
0084      GO TO 30
0085      26      WRITE(10,135) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),IDLN
0086      GOTO 30
0087      28      IF(ICOUNT.NE.5)GOTO 26
0089      WRITE(10,140) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),IDLN
0090      GOTO 30
0091      29      WRITE(10,125) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
1          IDLN,RAWEXP,EXPNSN
0092      30      JCONT=JCONT-1
0093      GOTO 90
0094      40      WRITE(5,160) IEND=4, J=5
0095      CALL CLOSE(2)
0096      90      RETURN
      C
      C

```

```

C      FORMAT STATEMENTS.
C
0097 100  FORMAT('PRINT THE HEADER? (Y/N): ')
0098 101  FORMAT(A1)
0099 105  FORMAT('BEGINNING RECORD NUMBER? (I4): ')
0100 106  FORMAT(I4)
0101 110  FORMAT('ENDING RECORD NUMBER? (I4): ')
0102 115  FORMAT('OUTPUT TO THE LINE PRINTER? (Y/N) ')
0103 119  FORMAT('1',19X,26A1)
0104 120  FORMAT(/1X,9A1,10X,8A1,15X,'PAGE ',I3)
0105 121  FORMAT(/3X,'TIME',4X,'HEATER',3X,'SPECIMEN',4X,
1 'LVDT',5X,'LN2 QTR.',5X,'% EXPANSION')
0106 122  FORMAT(1X,'HRS-MINS',3X,'TEMP.',5X,'TEMP.',3X,
1 'VOLTAGE',1X,'TURNS OPEN',4X,'RAW',4X,'COMPENSATED'/)
0107 125  FORMAT(2X,I2,':',I2,4X,F6.2,4X,F6.2,4X,F6.2,
1 4X,I2,8X,F6.4,5X,F6.4)
0108 130  FORMAT(' *',I2,':',I2,' **',F6.2,' *ok',F6.2,' ***',
1 F6.2,' ****',I2,' *****',F6.4,' ***',2X,F6.4)
0109 135  FORMAT(2X,I2,':',I2,3(4X,F6.2),4X,I2)
0110 140  FORMAT(' *',I2,':',I2,' **',F6.2,' *ok',F6.2,' **',
1 F6.2,' ***',I2)
0111 150  FORMAT(I2A1)
0112 160  FORMAT('/ RECORD * ',I3,' HAS NOT BEEN RECORDED. THE LAST' /
1 ' RECORD RECORDED WAS * ',I3)
0113      END

```

```

0001      SUBROUTINE NEWPR(FILNAM,KK)
          .IDENT/050078/
          C
          C+
          C      MAR 1978                      D. E. LACEY
          C      MAY 1978      REVISED L. L. HOLLAND AND D. E. LACEY
          C
          C      THIS SUBROUTINE OPTIONALLY PRINTS THE HEADER INFORMATION
          C      FOR THE TE2000 EXPERIMENT, THEN PRINTS ALL COLLECTED DATA
          C      AND NEW DATA. THE DISK DATAFILE IS CLOSED IMMEDIATELY
          C      FOLLOWING EACH READ.
          C
          C      UPON EXIT TO THE MAIN PROGRAM, THE USER MAY SELECT AN ADD-
          C      ITIONAL OPTION OR EXIT FROM THE PROGRAM.
          C-
0002      DIMENSION IHOR(8)
          C
0003      REAL*4 GAIN(4),TEMP(4),VOLTS(4),XTMR
          C
0004      INTEGER*2 IADW(4),MAG(4),IWRD(6),IDUM(6)
          C
0005      LOGICAL*1 FILNAM(26),INFLE(14),IHEDR(64),XLEN1(4),
          C      1 XTEM1(4),HOUR(8),IDATE(9)
          C
0006      EQUIVALENCE (IWRD,INFLE(1)),(XTMR,IDUM(1)),
          C      1 (XLEN,XLEN1(1)),(XTEM,XTEM1(1))
          C
0007      DATA IY/IHY/
          C
0008      CALL DATE(IDATE)
0009      IO=5
          C
          C      WRITE HEADER?
          C
0010      WRITE(5,100)
0011      READ(5,101) IANS
0012      IF (IANS.NE.IY) GO TO 5
0014      CALL HEADER(FILNAM,KK)
0015      WRITE(5,115)
          C
          C      OUTPUT: 4=LINEPRINTER, 5=TERMINAL.
          C
0016      READ(5,101) IANS
0017      IF (IANS.EQ.IY) IO=4
          C

```

```

C      THIS SECTION READS THE HEADER TO GET NUMBER OF RECORDS IN
C      FILE AND CONSTANTS NEEDED FOR DATA REDUCTION.
C
0019      CALL ASSIGN(2,FILNAM,KK)
0020      CALL FDBSET(2,'R','SHARE')
0021      DEFINE FILE 2(4,B,U,NRECRD)
0022      READ(2'2) (INFLE(1),I=1,14)
0023      READ(2'4) (IDUM(1),I=1,6)
0024      CALL CLOSE(2)
C
C      ADD 4 TO NREC TO SKIP THE HEADER
C
0025      NREC=IDUM(3)+4
0026      ISUG=INFLE(14)
0027      DO 7 IJK=1,4
0028          XLEN1(IJK)=INFLE(IJK+4)
0029          XTEM1(IJK)=INFLE(IJK+8)
0030      7      CONTINUE
C
C      INITIALIZE COUNTERS
C
0031      IPAGE=0
0032      JCONT=0
0033      ICOUNT=0
C
C      RFVOLT DETERMINES THE LYDT VOLTAGE AT REFERENCE TEMPERATURE
C
0034      CALL RFVOLT(FILNAM,KK,XTEM,XLEN,VREF,ISUG,XMVR)
C
C      NUMHLD IS A DUMMY TO USE WHEN WAITING FOR NEW DATA.
C
0035      NUMHLD=5
0036      10  DO 30 J=NUMHLD,NREC
0037          DEFINE FILE 2 (NREC,B,U,NRECRD)
0038          CALL ASSIGN(2,FILNAM,KK)
0039          CALL FDBSET(2,'R','SHARE')
0040          READ(2'J,END=40,ERR=40) (IWRD(1),I=1,6)
0041          CALL CLOSE(2)
0042          IHR=INFLE(1)
0043          MIN=INFLE(2)
0044          IDLN=INFLE(11)
0045          DO 20 K=1,3
0046      20      IADW(K)=IWRD(K+1)
C
C      ADC CONVERTS ANALOG TO DIGITAL SIGNAL

```

```

C
0047      CALL ADC(IADW(1),MAG(1),GAIN(1))
0048      DO 22 K=1,3
0049 22      VOLTS(K)=4.8852*FLOAT(MAG(K))/GAIN(K)
C
C      FIND HEATER AND SPECIMEN TEMPERATURES IN DEGREES F.
C
0050      CALL TEMPF(VOLTS(1),HTRTMP)
0051      CALL TEMPF(VOLTS(2),SPLTMP)
0052      VOLTS(3)=VOLTS(3)*MVR
C
C      JCONT IS THE NUMBER OF LINES PRINTED PER PAGE.
C
0053      IF(JCONT.NE.0)GOTO 24
0055      JCONT=45
C
C      IPAGE IS THE PAGE NUMBER
C
0056      IPAGE=IPAGE+1
C
C      THE TIME MUST BE STRIPPED OF PADDING BEFORE PRINTING
C
0057      CALL TIME(HOUR)
0058      DO 23 I=1,8
0059 23      IHOR(I)=HOUR(I)
0060      WRITE(10,119)(FILNAM(I),I=1,KK)
0061      WRITE(10,120)(IDATE(I),I=1,9),(IHOR(I),
1          I=1,8),IPAGE
0062      WRITE(10,121)
0063      WRITE(10,122)
0064 24      ICOUNT=ICOUNT+1
C
C      IFLAG = 0 SIGNALS THAT THE CROSSOVER TEMPERATURE
C      HAS NOT BEEN REACHED, AND MUST BE REACHED BEFORE
C      CALCULATING % EXPANSION
C
0065      IF(IFLAG.NE.0)GOTO 25
C
C      TWO FOLLOWING IF STATEMENTS SEARCH FOR REFERENCE
C      TEMPERATURE.
C
0067      IF(ISWG.EQ.0.AND.XTEM.LT.SPLTMP)GOTO 26
0069      IF(ISWG.EQ.1.AND.SPLTMP.LT.XTEM)GOTO 26
C
C      SET IFLAG=1 TO INDICATE THE REFERENCE TEMP HAS BEEN REACHED.

```

```

C
0071      IFLAG=1
C
C      CORRECTION IS FOR QUARTZ
C
0072 25      CORECT=.000031*(SPLTMP-XTEM)
0073      RAWEXP=XPMR*(VOLTS(3)-VREF)/XLEN*.1
0074      EXPNSN=RAWEXP+CORECT
C
C      PRINT EVERY FIFTH LINE WITH *****5 TO SET IT APART.
C
0075      IF(ICOUNT.NE.5)GOTO 29
0077      ICOUNT=0
0078      WRITE(10,130) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
1          IDLN,RAWEXP,EXPNSN
0079      GOTO 30
0080 26      IF(ICOUNT.NE.5)GOTO 28
0082      ICOUNT=0
0083      WRITE(10,140) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
1          IDLN
0084      GOTO 30
0085 28      WRITE(10,135) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),IDLN
0086      GOTO 30
0087 29      WRITE(10,125) IHR,MIN,HTRTMP,SPLTMP,VOLTS(3),
1          IDLN,RAWEXP,EXPNSN
0088 30      JCONT=JCONT-1
0089 40      JJ=1
C
C      READ HEADER TO SEE IF MORE DATA HAS BEEN RECORDED SINCE
C      THE HEADER WAS READ THE LAST TIME.
C
0090      DEFINE FILE 2 (NREC,B,U,NRECD)
0091      CALL ASSIGN(2,FILNAM,KK)
0092      CALL FDBSET(2,'R','SHARE')
0093      DO 45 J=1,4
0094          READ(2,J) (IHEDR(I),I=JJ,JJ+15)
0095 45      JJ=JJ+16
0096      CALL CLOSE(2)
0097      INT=IHEDR(47)
0098      NCY=IHEDR(29)
0099      ICCN=IHEDR(48)
C
C      ICCN > NCY SAYS EXPERIMENT IS FINISHED.
C
0100      IF (ICCN.GT.NCY) RETURN

```



```

      C
      C      IF NOT FINISHED, WAIT FOR THE PERIOD OF TIME BETWEEN SCHEDULED
      C      READINGS.
      C
0102      CALL WAIT(INT,3)
0103      NREC=NREC+1
0104      NUMHLD=NREC
      C
      C      READ ANOTHER RECORD.
      C
0105      GO TO 10
      C
      C      FORMAT STATEMENTS.
      C
0106 100      FORMAT('PRINT THE HEADER? [Y/N]: ')
0107 101      FORMAT(A1)
0108 115      FORMAT('OUTPUT TO THE LINE PRINTER? [Y/N] ')
0109 119      FORMAT('1',19X,26A1)
0110 120      FORMAT(/1X,9A1,10X,8A1,15X,'PAGE ',I3)
0111 121      FORMAT(/1X,'TIME',4X,'HEATER',3X,'SPECIMEN',4X,
1 'LVDT',5X,'LN2 QTR.',5X,'% EXPANSION')
0112 122      FORMAT(1X,'HRS-MINS',3X,'TEMP.',3X,'TEMP.',6X,
1 'VOLTAGE',1X,'TURNS OPEN',4X,'RAW',4X,'COMPENSATED')
0113 125      FORMAT(2X,I2,':',I2,4X,F6.2,4X,F6.2,4X,F6.2,
1 4X,I2,8X,F6.4,5X,F6.4)
0114 130      FORMAT(' *',I2,':',I2,' ** ',F6.2,' ** ',F6.2,' ** '
1 ',F6.2,' ***',I2,' ****',F6.4,' ** ',F6.4)
0115 135      FORMAT(2X,I2,':',I2,2(4X,F6.2),4X,F6.2,4X,I2)
0116 140      FORMAT(' *',I2,':',I2,' ** ',F6.2,' ***',F6.2,' ** ',
1 F6.2,' ***',I2)
0117 150      FORMAT(12A1)
0118      END

```

```

0001      SUBROUTINE HEADER(FILNAM,K)
          C      .IDENT/050578/
          C+     FEB. 1978                      D. E. LACEY
          C
          C      THIS SUBROUTINE OPENS THE TE2000 DATA FILE ON DK1:
          C      FOR "READONLY" SHARED ACCESS AND PRINTS THE EXPERI-
          C      MENT HEADER INFORMATION, (EXPERIMENT PARAMETERS,
          C      DEVICE ASSIGNMENTS, TIME, DATE, LTR NO. AND ETC.),
          C      ON THE DEVICE SPECIFIED BY IO. THE DISK FILE IS CLOSED UPON
          C      COMPLETION OF THE READ. UPON EXIT TO THE MAIN PROG-
          C      RAM, THE USER MAY CHOOSE ANOTHER OPTIONAL SUBROUTINE
          C      FOR FURTHER OUTPUT.
          C
          C-
0002      INTEGER*2 INFLE(32)
          C
0003      DIMENSION ARRAY(3)
0004      LOGICAL*1 INFLE(64), HOUR(8)
0005      LOGICAL*1 FILNAM(26), EMPNO(6), SID(2),
          C      1 XLEN(4), XTEM1(4), XLST1(4),
          C      2 XHST1(4), XMVR1(4), LTR(7)
          C
0006      EQUIVALENCE (INFLE, INFLE(1))
0007      EQUIVALENCE (LTR, INFLE(1)), (SID, INFLE(8)), (EMPNO, INFLE(18)),
          C      1 (XLEN, XLEN(1)), (XTEM, XTEM1(1)), (XLST, XLST1(1)),
          C      2 (XHST, XHST1(1)), (XMVR, XMVR1(1)), (NDS, INFLE(53))
          C
0008      DATA IY/IHY/
          C
          C+
          C      OUTPUT: 5=TERMINAL
          C      4=LINE PRINTER
          C      3=ARCHIVE
          C-
0009      IO=5
0010      WRITE(5,175)
0011      READ(5,177) IANS
0012      IF (IANS.EQ.IY) GO TO 1
0014      WRITE(5,180)
0015      READ(5,177) IANS
0016      IF (IANS.NE.IY) GO TO 2
0018      IO=3
0019      GO TO 2
0020 1      IO=4

```

```

0021 2      CALL ASSIGN(2,FILNAM,K)
0022      CALL FDBSET(2,'R','SHARE')
0023      DEFINE FILE 2(4,8,U,NRECRD)

C
C+
C
C      READ HEADER FOR INFORMATION TO BE PRINTED ON HEADER.
C      IHR =      HOUR DATA WAS RECORDED
C      MIN =      MINUTE DATA WAS RECORDED
C      MON =      MONTH DATA WAS RECORDED
C      IDAY =     DAY DATA WAS RECORDED
C      IYR =      YEAR DATA WAS RECORDED
C      NCY =      NUMBER OF THERMAL CYCLES
C      S?
C      ISTC=      SPECIMEN TEMP A/D CHANNEL
C      INTC=      HEATER TEMP A/D CHANNEL
C      LVC =      LVDT VOLTAGE CHANNEL
C      IHVC=      HEATER VOLTAGE CHANNEL
C      IPSC=      POWER SUPPLY RELAY *
C      LNS =      LN2 SOLENOID RELAY *
C      LNC =      STEPPER MOTOR RELAY *
C      LNV =      INITIAL LN2 QTR TURNS OPEN
C      INT =      NUMBER OF MINUTES BETWEEN READINGS
C      ICCN=      CURRENT CYCLE NUMBER
C      LTR =      LAB TEST REQUEST *
C      SID =      SAMPLE ID *
C      EMPNO=     EMPLOYEE NUMBER
C      NDS =      NUMBER OF DATA SETS RECORDED
C      XLEN=      INITIAL SPECIMEN LENGTH
C      XMVR=      LVDT MV/INCH RATIO
C      XTEM=      STARTING CYCLE TEMPERATURE (REF. TEMP.)
C      XHST=      HIGH SIDE CYCLE TEMPERATURE
C      XLST=      LOW SIDE CYCLE TEMPERATURE
C
C-
0024      JJ=1
0025      DO 3 J=1,4
0026          READ(2*J) (INFILE(I), I=JJ,JJ+7)
0027      3      JJ=JJ+8
C
0028      CALL CLOSE(2)
C
0029      IHR=INFLE(16)
0030      MIN=INFLE(17)
0031      MON=INFLE(18)

```

```

0032      IDAY=INFLE(19)
0033      IYR=INFLE(20)
0034      NCY=INFLE(29)
0035      ISUG=INFLE(30)
0036      ISTC=INFLE(39)
0037      IHTC=INFLE(40)
0038      LVC=INFLE(41)
0039      IHVC=INFLE(42)
0040      IPSC=INFLE(43)
0041      LNS=INFLE(44)
0042      LNC=INFLE(45)
0043      LNV=INFLE(46)
0044      INT=INFLE(47)
0045      ICCN=INFLE(48)
0046      DO 6 J=1,4
0047 6      XLEN1(J)=INFLE(J+20)
0048      DO 7 J=1,4
0049 7      XTEM1(J)=INFLE(J+24)
0050      DO 8 J=1,4
0051 8      XLST1(J)=INFLE(J+30)
0052      DO 9 J=1,4
0053 9      XHST1(J)=INFLE(J+34)
0054      DO 10 J=1,4
0055 10     XNVRT(J)=INFLE(J+40)
        C
0056      CALL DATE(ARRAY)
0057      CALL TIME(HOUR)
0058      IH1=HOUR(1)
0059      IH2=HOUR(2)
0060      IM1=HOUR(4)
0061      IM2=HOUR(5)
0062      IH1=IH1-48
0063      IH2=IH2-48
0064      IM1=IM1-48
0065      IM2=IM2-48
0066      IH=IH1*10+IH2
0067      IM=IM1*10+IM2
        C
0068      WRITE(10,100)
0069      WRITE(10,101)(ARRAY(I),I=1,3),IH,IM
        C
0070      WRITE(10,105) LTR,SID,EMPNO
0071      WRITE(10,110) IHR,MIN,MON,1DAY,IYR
0072      WRITE(10,115) NDS,INT
0073      WRITE(10,120)

```

```

0074      WRITE(10,125) XLEN,XMYR
0075      WRITE(10,130) XTEM,NCY
0076      IF(1SWG.EQ.1)GOTO15
0078      WRITE(10,135) ICCN
0079      GOTO17
0080 15     WRITE(10,136) ICCN
0081 17     WRITE(10,140) XHST,XLST
0082      WRITE(10,145) LNV
0083      WRITE(10,150)
0084      WRITE(10,155) ISTC
0085      WRITE(10,160) IHTC
0086      WRITE(10,165) LVC
0087      WRITE(10,170) IHVC
0088      WRITE(10,171) IPSC,LNS,LNC

C
C      FORMAT STATEMENTS.
C
0089 100    FORMAT(1H1///11X,'T H E R M A L   E X P A N S I O N   I N F'.
1         ' O R M A T I O N')
0090 101    FORMAT(/26X,3A4,19X,12,' ',12)
0091 105    FORMAT(//10X,'LTR NO.: ',7A1/10X,'SPECIMEN IDENTIFICATION: ',
1 2A1/10X,'EMPLOYEE NO.: ',6A1)
0092 110    FORMAT(/10X,'DATA RECORDED: ',12,' ',12,' - ',12,'/',12,'/',12)
0093 115    FORMAT(/15X,'NO. OF DATA SETS RECORDED: ',14/15X,'NO OF MIN
1UTES BETWEEN SAMPLES: ',14)
0094 120    FORMAT(//10X,'EXPERIMENT PARAMETERS:')
0095 125    FORMAT(/15X,'INITIAL SPECIMEN LENGTH: ',G12.4/15X,'LVDT/MV/INCH
1 : ',G12.4)
0096 130    FORMAT(/15X,'STARTING CYCLE TEMP.: ',F6.2/15X,'NUMBER OF THERMA
1L CYCLES: ',14)
0097 135    FORMAT(/15X,'INITIAL SWING DIRECTION: DOWN'/15X,
1 'CURRENT CYCLE NUMBER: ',14)
0098 136    FORMAT(/15X,'INITIAL SWING DIRECTION: UP'/15X,
1 'CURRENT CYCLE NUMBER: ',14)
0099 140    FORMAT(15X,'HIGH-SIDE CYCLE TEMP.: ',F6.2/15X,'LOW-SIDE CYCLE
1 TEMP: ',F6.2)
0100 145    FORMAT(15X,'INITIAL LN2, (QUARTER TURNS OPEN): ',14)
0101 150    FORMAT(//10X,'DEVICE ASSIGNMENTS FOR THE EXPERIMENT:')
0102 155    FORMAT(/15X,'CHAN #',12,' = SPECIMEN TEMPERATURE')
0103 160    FORMAT(15X,'CHAN #',12,' = HEATER TEMPERATURE')
0104 165    FORMAT(15X,'CHAN #',12,' = LVDT VOLTAGE')
0105 170    FORMAT(15X,'CHAN #',12,' = HEATER VOLTAGE',//)
0106 171    FORMAT(/15X,'CHAN #',12,' = POWER SUPPLY RELAY'/15X,
1 'CHAN #',12,' = LIQUID NITROGEN RELAY'/15X,'CHAN #',
2 12,' = STEPPER MOTOR RELAY')

```

```
0107 175  FORMAT('$OUTPUT HEADER TO THE LINE PRINTER? [Y/N] ')
0108 177  FORMAT(A1)
0109 180  FORMAT('$ARCHIVE HEADER? [Y/N]; ')
0110      RETURN
0111      END
```

```

0001      SUBROUTINE TIMINC
           .IDENT/031478/
           C
           C+
           C      MARCH 1978                      D. E. LACEY
           C
           C      THIS SUBROUTINE WILL SEND A CHANGE OF THE TIME INTERVAL FOR
           C      TSK, TE2DAC, INSTALLED AND RUNNING UNDER TIME SCHEDULING.
           C      THE NEW TIME INTERVAL IS ARBITRARILY LIMITED TO THE
           C      INTERVAL, (GE.1.AND.LE.59), AND IS PASSED IN THE FIRST
           C      BYTE OF THE THIRD WORD OF THE RECIEVING TASK'S BUFFER.
           C
           C      CALLING SEQUENCE:
           C
           C      CALL TIMINC
           C-
0002      INTEGER*2 IBUF(13)
           C
0003      DATA TSK/6RTE2DAC/
           C
0004      WRITE(5,115)
0005      5    READ(5,120) IBUF(1)
0006          IF (IBUF(1).GE.1.AND.IBUF(1).LE.59) GO TO 10
0007      WRITE(5,125)
0008      GO TO 5
0009
0010      10    CALL SEND(TSK,IBUF)
0011      115   FORMAT('WHAT IS THE NEW TIME INCREMENT? ')
0012      120   FORMAT(I2)
0013      125   FORMAT('INPUT ERROR, ENTER THE NEW TIME INCREMENT: ')
0014      RETURN
0015      END

```

```

0001      SUBROUTINE LEDLAG(RFTEMP,TUP,EXUP,JCOUNT)
          C          .IDENT/051578/
          C+
          C      APR. 1978          D. E. LACEY AND L. L. HOLLAND
          C      MAY 1978      LATEST REVISION          L. L. HOLLAND
          C
          C      THIS SUBROUTINE CORRECTS TE2000 DAT FOR THERMOCOUPLE LEAD
          C      AND LAG.
          C
          C-
0002      DIMENSION TEMP(300),TUP(200),TDN(200),WK(300),
          1 EXPNSN(300),EXUP(200),EXDN(200),WK1(300)
          C
0003      DO 5 J=1,300
0004          READ(1,100,END=6,ERR=6) EXPNSN(J),TEMP(J)
0005      5      WK(J)=TEMP(J)
0006      6      ICOUNT=J-1
0007          IF (TEMP(1).GT.TEMP(ICOUNT)) ICOUNT=ICOUNT-1
0009      DO 7 IW=1,50
0010          IF (TEMP(IW).NE.TEMP(IW+1)) GOTO 8
0012      7      CONTINUE
0013          STOP 'THERMOCOUPLE BROKEN'
0014      8      ISWG=0
0015          IF (TEMP(IW).LT.TEMP(IW+1)) ISWG=1
0017          CALL MAXMIN(WK,ICOUNT,TMAX,TMIN)
0018          IF (ISWG.EQ.0) GO TO 11
0020          B=TMIN
0021          TMIN=TMAX
0022          TMAX=B
0023      11      DO 16 J=1,ICOUNT
0024          WK(J)=TEMP(J)
0025          WK1(J)=EXPNSN(J)
0026          IF (TEMP(J).EQ.TMIN)GOTO19
0028      16      CONTINUE
          C
          C      THIS SECTION DETERMINES IF THE MIN. TEMP OCCURS
          C      TWO OR MORE CONSECUTIVE TIMES.
          C
0029      18      DO 19 I=1,25
0030          J=J+1
0031          IF (TEMP(J).NE.TMIN)GOTO 20
0033          WK(J)=TEMP(J)
0034      19      WK1(J)=EXPNSN(J)
0035      20      J=J-1
0036          JK=J

```



```

0037      JJ=J+1
0038      KK=JJ
0039      K=1
0040      DO 25 J=JJ, ICOUNT
0041          IF (TEMP(J), EQ, TMAX) GO TO 30
0043          TUP(K)=TEMP(J)
0044          EXUP(K)=EXPNSN(J)
0045      25      K=K+1
0046      30      JJ=J
0047          NUP=J-JK-1
0048          K=1
0049          DO 35 J=JJ, ICOUNT
0050              TDN(K)=TEMP(J)
0051              EXDN(K)=EXPNSN(J)
0052      35      K=K+1
0053      KK=K
0054      K=1
0055      NDN=JK+KK-1
0056      DO 45 J=KK, NDN
0057          TDN(J)=WK(K)
0058          EXDN(J)=WK1(K)
0059      45      K=K+1
0060      IF (ISWG.EQ.0) GO TO 49
0062      DO 46 J=1, NDN
0063          WK(J)=TUP(J)
0064          WK1(J)=EXUP(J)
0065          TUP(J)=TDN(J)
0066      46      EXUP(J)=EXDN(J)
0067      DO 47 J=1, NUP
0068          TDN(J)=WK(J)
0069      47      EXDN(J)=WK1(J)
0070      49      N=1
0071      NPTSD=0
0072      DO 70 JKL=1, NUP
0073          DO 50 IK=1, NDN
0074              IF (EXUP(JKL), EQ, EXDN(IK)) GOTO 52
0076              IF (EXUP(JKL), GT, EXDN(IK)) GOTO 55
0078      50      CONTINUE
0079          WRITE(5, 115) TUP(JKL), EXUP(JKL)
0080          NPTSD=NPTSD+1
0081          GO TO 70
0082      52      WK(N)=TDN(IK)
0083          N=N+1
0084          GO TO 70
0085      55      CONTINUE

```

```

0086      DO 59 MEN=1,30
0087          IF (TDN(IK).NE.TDN(IK-MEN))GOTO 60
0089      59      CONTINUE
0090      STOP 'ERROR'
0091      60      DO 62 MN=1,25
0092          MAN=MN-1
0093          IF (EXDN(IK).NE.EXDN(IK-MEN-MAN))GOTO 65
0095      62      CONTINUE
0096      65      MANN=IK-MEN-MAN
0097      SLOPE=(EXDN(IK)-EXDN(MANN))/(TDN(IK)
1          -TDN(MANN))
0098      B=EXDN(IK)-SLOPE*TDN(IK)
0099      WK(N)=(EXUP(JKL)-B)/SLOPE
0100      N=N+1
0101      70      CONTINUE
0102      IF (NPTSD.EQ.0) GO TO 75
0104      WRITE(5,120) NPTSD
0105      75      JCOUNT=N-1
0106      DO 90 I=1,JCOUNT
0107          CORR=(TUP(I)-WK(I))*5
0108      90      TUP(I)=TUP(I)-CORR
0109      DO 91 I=1,JCOUNT
0110          IF (RFTEMP.EQ.TUP(I))GOTO 92
0112          IF (RFTEMP.LY.TUP(I))GOTO 93
0114      91      CONTINUE
0115      STOP 'REFERENCE TEMPERATURE NOT IN RANGE'
0116      92      ZEREXP=EXUP(I)
0117      GOTO 94
0118      93      IF (I.EQ.1) I=2
0120      SLOPE=(EXUP(I)-EXUP(I-1))/(TUP(I)-TUP(I-1))
0121      ZEREXP=SLOPE*(RFTEMP-TUP(I))+EXUP(I)
0122      94      DO 95 I=1,JCOUNT
0123      95      EXUP(I)=EXUP(I)-ZEREXP
0124      WRITE(5,105)
0125      READ(5,110) IDEL
0126      JCOUNT=JCOUN-DELE
0127      RETURN
0128      100      FORMAT(F7.4,F8.2)
0129      105      FORMAT(' HOW MANY POINTS ARE TO BE DELETED FROM THE HIGH',/
1          ' $TEMPERATURE END OF THE THERMOCYCLE? (13): ')
0130      110      FORMAT(13)
0131      115      FORMAT(' TEMP= ',F6.2,5X,' EXPANSION= ',F6.4,' EXPANSION'/
1          ' NOT FOUND. POINT DELETED.')
0132      120      FORMAT(' NUMBER OF POINTS DELETED= ',13)
0133      END

```

```

      C
      C
      C
0001      SUBROUTINE MAXMIN(ARRAY,N,MAX,MIN)
      C+
      C      1977                                D. E. LACEY
      C
      C      THIS SUBROUTINE DOES A PUSH DOWN SORT OF A ONE
      C      DIMENSIONED ARRAY OF N ELEMENTS.  THE MAXIMUM
      C      AND MINIMUM VALUES ARE RETURNED TO THE CALLING
      C      PROGRAM.
      C-
      C
0002      REAL*4 MAX,MIN
0003      DIMENSION ARRAY(N)
      C
0004      DO 15 M=1,N
0005      DO 15 K=2,N
0006          J=K-1
0007          IF (ARRAY(J).GT.ARRAY(K)) GO TO 15
0009          B=ARRAY(J)
0010          ARRAY(J)=ARRAY(K)
0011          ARRAY(K)=B
0012      15      CONTINUE
0013      MAX=ARRAY(1)
0014      MIN=ARRAY(N)
0015      RETURN
0016      END

```

```

0001      PROGRAM TEPLOT
          .IDENT /050978/

C
C+
C
C      APR 1978                      L. L. HOLLAND
C      MAY 1978      LATEST REVISION      D. E. LACEY
C
C      THIS PROGRAM PLOTS PERCENT EXPANSION VERSUS
C      TEMPERATURE FOR THE THERMAL EXPANSION EXPERIMENT.
C      INPUTS ARE THE DATA FILES CREATED BY SUBROUTINE LPPRNT.
C      DK1:[300,300]PARAMS.DAT
C
C-
C
0002      DIMENSION EXPNSN(500),TEMP(500),SLOPE(25),TEMI(25),
1      TEMP(25),TDATE(3),TEM(5),EXP(5)
0003      LOGICAL*1 FILE(26),ID(2),SPCNAM(26),
1      EMPNO(6),CHARG(8),TSTSPC(12),NAME(14),ENUMB(6),
2      MATERL(12),FILE2(26),REDDBY(14),LTR(7)
0004      DATA SAME/9999./,IY/IHY/,IH/IHW/,IMD/IMD/,IR/IHR/

C
C
0005      DATA TEM/-64.,00.,152.,260.,404./
0006      DATA EXP/-.1173.,.0116.,.0789.,.1828.,.3259/

C
C      READ PARAMETERS FROM DATA FILE CREATED BY SUBROUTINE LPPRNT
C      IN TASK TE2000.
C
0007      CALL ASSIGN(1,'DK1:[300,300]PARAMS.DAT')
0008      READ(1,105) K1
0009      READ(1,100) ID
0010      READ(1,101) (LTR(I),I=1,K1)
0011      READ(1,102) RFTMP
0012      READ(1,105) NFILE
0013      READ(1,103) (FILE(I),I=1,NFILE)
0014      READ(1,104) (EMPNO(I),I=1,5)
0015      CALL CLOSE(1)

C
0016      1      WRITE(5,120)
0017      2      READ(5,105) ITC
0018      IF (ITC.EQ.1) GOTO 3
0020      IF (ITC.EQ.2) GOTO 4
0022      WRITE(5,177)
0023      GOTO 2
0024      3      CALL ASSIGN(1,'DK1:[300,300]THMEXP.DAT;1')

```

```

0025      GOTO 5
0026  4    CALL ASSIGN(1,'DK1:(300,300)THMEXP,DAT;2')
0027  5    WRITE(5,135)
0028      READ(5,125)K2,SPCNAM
0029      READ(1,101)DUN
0030      READ(1,101)DUN
0031      READ(1,105)ICYCLE
0032      WRITE(5,140)
0033      READ(5,110)IANS
0034      IF(IANS.NE.1Y)GOTO 7
0036      CALL LEDLAG(RFTEMP,TEMP,EXPNSH,NEL)
0037      GOTO 12
0038  7    DO 10 I=1,1000
0039  10      READ(1,100,END=11,ERR=11)EXPNSH(I),TEMP(I)
0040  11      NEL=1
0041  12    CALL CLOSE(1)
0042      KFLAG=0
0043      INT=1
0044      WRITE(5,145)
0045      READ(5,110) IANS
0046      IF(IANS.NE.1Y)GOTO 30
0048      KFLAG=1
0049  15    WRITE(5,147)
0050      READ(5,140)TEMI(INT)
0051      IFG=0
0052      WRITE(5,195)
0053      READ(5,110)IANS
0054      IF (IANS.EQ.1Y) GOTO 17
0056      IFG=1
0057  17    DO 20 IN=1,NEL
0058          IF(TEMI(INT).GT.TEMP(IN))GOTO 20
0060          IF(IFG.EQ.0) GO TO 23
0062          IFG=0
0063  20      CONTINUE
0064      WRITE(5,151)
0065      GOTO 15
0066  23    WRITE(5,150)
0067      READ(5,140)TEMF(INT)
0068      DO 24 IM=1,NEL
0069          IF(TEMF(INT).LE.TEMP(IM))GOTO 25
0071  24      CONTINUE
0072      WRITE(5,152)
0073      GOTO 23
C
C      FIND COEFFICIENT, *EXP(-6) DEGREES F

```

```

      C      COEFF IS SLOPE/100
      C
0074 25      SLOPE(INT)=(EXPNSN(IN)-EXPNSN(IM))/(TEMP(IN)-
1          TEMP(IM))*10.**4
0075      INT=INT+1
0076      WRITE(5,154)
0077      READ(5,110) IANS
0078      IF(IANS.EQ.IY)GOTO 15
0080      INT=INT-1
      C
0081 30      CALL MODE(8,1.,1.,1.)
0082      CALL SCAN(TEMP,EXPNSN,-NEL,6440)
0083      WRITE(5,106)
0084      READ(5,110) IANS
0085      IF(IANS.NE.IY) GO TO 32
0087      WRITE(5,107)
0088      READ(5,148) YFACT
0089      WRITE(5,112)
0090      READ(5,148) YMIN
0091      XORG=-YMIN/YFACT
0092      CALL MODE(9,YMIN,YFACT,XORG)
0093      GO TO 34
0094 32      IF (X.GT..05) GO TO 34
0096      CALL MODE(-9,YMIN,X,XORG)
0097      XORG=-YMIN/.05
0098      CALL MODE(9,SAME,.05,XORG)
0099 34      CALL NOTE(TEMP,EXPNSN,30,-NEL)
0100      CALL MODE(5,SAME,SAME,6440.)
0101      CALL DRAW(TEMP,EXPNSN,NEL,6440)
0102      CALL AXES(22.0,'TEMPERATURE, DEGREES F',11.3,
1          '% EXPANSION')
0103      CALL NOTE(.5,8.9,'LTR# ',5)
0104      CALL NOTE(.5,8.7,'SPECIMEN ID# ',13)
0105      CALL NOTE(1.,8.9,LTR,K1)
0106      CALL NOTE(1.9,8.7,ID,2)
0107      CALL NOTE(.5,8.5,'CYCLE# ',7)
0108      CALL NOTE(1.2,8.5,ICYCLE,0)
0109      CALL MODE(4,.2,.134,SAME)
0110      CALL NOTE(2.5,9.,SPCNAM,K2)
0111      IF(KFLAG.EQ.0)GOTO 40
0113      CALL MODE(4,.062,.041,SAME)
0114      CALL NOTE(.5,8.,'TEMPERATURE',11)
0115      CALL NOTE(.5,7.8,' RANGE',8)
0116      CALL NOTE(.5,7.6,' (DEG F)',9)
0117      CALL NOTE(1.8,8.,'COEFFICIENT OF LINEAR',23)

```

```

0118      CALL NOTE(1.8,7.8,' THERMAL EXPANSION',28)
0119      CALL NOTE(1.8,7.6,' (MICROINCH/INCH/DEG F)',22)
0120      CALL MODE(4,.077,.051,SAME)
0121      SUB=7.3
0122      DO 35 I=1,INT
0123          CALL NOTE(.5,SUB,TEMP(I),1000)
0124          CALL NOTE(.75,SUB,' TO ',4)
0125          CALL NOTE(1.1,SUB,TEMP(I),1000)
0126          CALL NOTE(2.3,SUB,SLOPE(I),1000)
0127      35  SUB=SUB-.2
0128      40  CALL MODE(4,.1,.067,SAME)
0129      WRITE(5,175)
0130      READ(5,110) IANS
0131      IF(IANS.NE.IY)GOTO 45
      C
      C
      C
0133      CALL NOTE(TEM,EXP,35,-5)
0134      CALL DRAW(TEM,EXP,5,6440)
0135      CALL NOTE(.5,8.3,'NBS POINTS: 4',14)
0136      45  CALL DRAW(8.,8.,1,9000)
0137      CALL DRAW(8,8,8,999)
0138      WRITE(5,130)
0139      READ(5,110) IANS
0140      IF(IANS.EQ.IY)GOTO 1
0142      WRITE(5,156)
0143      READ(5,110) IANS
0144      IF(IANS.NE.IY)GOTO 80
0146      DO 50 I=1,NFILE-3
0147          FILE2(I)=FILE(I)
0148      50  CONTINUE
0149      FILE2(NFILE-2)=(H
0150      FILE2(NFILE-1)=(HD
0151      FILE2(NFILE)=(R
0152      CALL DATE(TDATE)
0153      WRITE(4,158)
0154      CALL ASSIGN(1,FILE2,NFILE)
0155      READ(1,101) DUM
0156      READ(1,160)N1,N2,N3,N4
0157      READ(1,101)DUM
0158      READ(1,101) (CHARG(I),I=1,N1)
0159      READ(1,101) (REQDBY(I),I=1,N2)
0160      READ(1,173) IDEPTH
0161      READ(1,101) (TSTSPC(I),I=1,N3)
0162      READ(1,101) (MATERL(I),I=1,N4)

```

```

0163      CALL CLOSE(1)
      C
      C      CREATE TEST LAB REPORT
      C
      C      ATTACH LINE PRINTER
      C
0164      CALL ATTDEV
0165      WRITE(4,162)(REGDBY(1),I=1,N2)
0166      WRITE(4,169) IDEPTH,TDATE
0167      WRITE(4,163)(LTR(1),I=1,7)
0168      WRITE(4,164)(CHARG(1),I=1,N1)
0169      WRITE(4,165)(MATERL(1),I=1,N4)
0170      WRITE(4,166)(TSTSPC(1),I=1,N3)
0171      WRITE(4,167)
      C
      C      FIND NAME TO MATCH EMPLOYEE NUMBER IN FILE
      C
0172      CALL ASSIGN(2,'DK1:(300,300)EMPLNO.DAT')
0173      DO 55 I=1,5
0174          READ(2,101)DUM
0175      55      CONTINUE
0176      DO 65 I=1,10
0177          READ(2,170,END=70,ERR=70)(ENUMB(I),I=1,6),
1              (NAME(I),I=1,14)
0178          DO 60 LI=1,5
0179              IF(EMPNO(LI).NE.ENUMB(LI))GOTO 65
0181      60      CONTINUE
0182          GOTO 75
0183      65      CONTINUE
0184      70      STOP 'EMPL. NUMBER NOT FOUND'
0185      75      CALL CLOSE(2)
0186      WRITE(4,168)(NAME(I),I=1,14)
0187      FILE(NFILE)=IHD
0188      WRITE(4,172)(FILE(I),I=1,NFILE)
0189      WRITE(4,170)
      C
      C      DELETE WORK FILES AND EXIT PROGRAM
      C
0190      00      CALL ASSIGN(1,'DK1:(300,300)THMEXP.DAT;1')
0191      WRITE(1,190)
0192      CALL DELFLE(1,1ERR)
0193      CALL ASSIGN(1,'DK1:(300,300)THMEXP.DAT;2')
0194      WRITE(1,190)
0195      CALL DELFLE(1,1ERR)
0196      CALL ASSIGN(1,'DK1:(300,300)PARAMS.DAT;1')

```



```

0197      WRITE(1,190)
0198      CALL DELFLE(1,IERR)
      C
      C      TELL OPERATOR WE'RE LEAVING.
      C
0199      WRITE(5,196)
0200      CALL EXIT
      C
      C      FORMAT STATEMENTS
      C
0201 100  FORMAT(F7.4,F8.2)
0202 101  FORMAT(30A1)
0203 105  FORMAT(13)
0204 106  FORMAT('$CHANGE Y SCALE FACTOR? [Y/N]: ')
0205 107  FORMAT('$ENTER Y SCALE FACTOR (F6.2): ')
0206 110  FORMAT(A1)
0207 112  FORMAT('$ENTER Y MINIMUM. (F6.2) ')
0208 120  FORMAT('$ENTER THERMOCYCLE # [1 OR 2]: ')
0209 125  FORMAT(0,26A1)
0210 130  FORMAT('$MORE PLOTS TO MAKE? [Y/N]: ')
0211 135  FORMAT('$ENTER SPECIMEN NAME. (26A1): ')
0212 140  FORMAT('$PLOT CORRECTED DATA? [Y/N]: ')
0213 145  FORMAT('$CALCULATE COEFFICIENTS? [Y/N]: ')
0214 147  FORMAT('$ENTER LOWER TEMPERATURE (F6.2): ')
0215 148  FORMAT(F6.2)
0216 150  FORMAT('$ENTER HIGHER TEMPERATURE. (F6.2): ')
0217 151  FORMAT(' INITIAL TEMPERATURE NOT IN RANGE.')
0218 152  FORMAT(' FINAL TEMPERATURE NOT IN RANGE.')
0219 154  FORMAT('$CALCULATE MORE COEFFICIENTS? [Y/N]: ')
0220 156  FORMAT('$MAKE COVER SHEET? [Y/N]: ')
0221 158  FORMAT('1',///43X,'TEST LABORATORY REPORT'
1      ///)
0222 160  FORMAT(416)
0223 162  FORMAT('$',3X,'TO: ',14A1)
0224 163  FORMAT(/4X,'LTR NO.: ',7A1)
0225 164  FORMAT(/4X,'E. P. NO.: ',14A1///)
0226 165  FORMAT(/4X,'MATERIAL: ',14A1)
0227 166  FORMAT(/////4X,'TEST SPECIFICATIONS: ',14A1)
0228 167  FORMAT(////////4X,'TEST RESULTS: '/21X,'SEE ATTACHED GRAPHS.')
0229 168  FORMAT(//////////'$',3X,'TESTED BY: ',14A1)
0230 169  FORMAT('+'2X,'D/',13,65X,'DATE: ',3A4/)
0231 170  FORMAT(6A1,2X,14A1)
0232 172  FORMAT('+' /TE2000-VERS1 //',26A1)
0233 173  FORMAT(14)
0234 175  FORMAT('$PLOT COPPER STANDARD POINTS? [Y/N]: ')

```

```

0235 177  FORMAT(' PROGRAM WORKS FOR TWO THERMOCYCLES ONLY.',
      1  // 'ENTER THERMOCYCLE # (1 OR 2): ')
0236 178  FORMAT(// ' APPROVED BY:')
0237 180  FORMAT(2A1)
0238 181  FORMAT(7A1)
0239 182  FORMAT(F5.1)
0240 183  FORMAT(26A1)
0241 184  FORMAT(6A1)
0242 190  FORMAT(1X, 'ERASE FILE.')
0243 195  FORMAT(' $FIRST OCCURRENCE OF INITIAL TEMP? (Y/N): ')
0244 196  FORMAT(// ' THERMAL EXPANSION DATA REDUCTION COMPLETED.')
0245      END

```

```

0001      SUBROUTINE TEMPF(V,T)
          C
          C      THIS SUBROUTINE CONVERTS EMF (V) VOLTAGES FROM CHROMEL-ALUMEL
          C      THERMOCOUPLES TO TEMPERATURE IN DEGREES 'F' (T)
          C
0002      IF(V .GE. 0.0) GO TO 70          !SKIP IF POSITIVE EMF
0004      IF(V .GT. -.692) GO TO 30
0006      IF(V .GT. -1.445) GO TO 40
0008      IF(V .EQ. -1.445) GO TO 50
0010      T = 35.8618 + 49.7259*V          !EMF .LT. -1.445
0011      GO TO 160
0012  30    T = 32.8339 + 46.1652*V          ! 0.0 .GT. EMF .GE. -.692
0013      GO TO 160
0014  40    T = 32.9694 + 47.6031*V          ! -.692 .GT. EMF .GT. -1.445
0015      GO TO 160
0016  50    T = -36.0                      ! EMF .EQ. -1.445
0017      GO TO 160
0018  70    C1 = -1.853306E01
0019      C2 = 3.891834E01
0020      C3 = 1.664515E-02
0021      C4 = -7.870237E-05
0022      C5 = 2.283579E-07
0023      C6 = -3.570023E-10
0024      C7 = 2.993291E-13
0025      C8 = -1.284985E-16
0026      C9 = 2.2240E-20
0027      N = 8
0028      T = 512.
0029  100   E = C1 + C2*T + C3*T**2 + C4*T**3 + C5*T**4 + C6*T**5 + C7*T**6
0030      E = E + C8*T**7 + C9*T**8
0031      E = .001 * ( E + 125.*EXP( - ( ( (T-127.)/65. )**2 )/2. ) )
0032      IF(V .EQ. E) GO TO 150
0034      IF(V .GT. E) GO TO 130
0036      T = T - 2.*N                      !HERE IF V .LT. E
0037      GO TO 140
0038  130   T = T + 2.*N                      !HERE IF V .GT. E
0039  140   N = N - 1
0040      IF(N .NE. -5) GO TO 100
0042  150   T = 1.8*T + 32.                  !'C' TO 'F' CONVERSION
0043  160   RETURN
0044      END

```

```
      C
      C
      C
      C
0001  SUBROUTINE TEMPC(V,T)
      C
      C    THIS SUBROUTINE CONVERTS EMF VOLTAGES ('V') FROM
      C    CHROMEL-ALUMEL THERMOCOUPLES TO TEMPERATURE IN DEGREES 'C'(T)
      C
0002  CALL TEMPF(V,T)          !FIRST CONVERT TO DEGREES F
0003  T = ( T - 32.)/1.8      !NOW CONVERT TO DEGREES C
0004  RETURN
0005  END
```

```

0001      SUBROUTINE RFVOLT(FILNAM, KK, XTEM, XLEN, VREF, ISUG, XTVR)
           C      IDENT/050078/
           C+
           C      APR 1978                      L. L. HOLLAND
           C      MAY 1978      LATEST REVISION      L. L. HOLLAND
           C
           C      THIS SUBROUTINE CALCULATES THE LVDT VOLTAGE AT THE
           C      REFERENCE TEMPERATURE FOR THE TE2000 DATA REDUCTION
           C      SOFTWARE.
           C-
0002      REAL*4 GAIN(4), VOLTS(4)
           C
0003      LOGICAL*1 FILNAM(26)
           C
0004      INTEGER*2 IWRD(6), IADW(4), MAG(4)
           C
0005      DO 30 J=5,500
0006          DEFINE FILE 2(504,8,U,NRECD)
0007          CALL ASSIGN (2,FILNAM, KK)
0008          CALL FDBSET (2, 'R', 'SHARE')
0009          READ(2,J,END=32,ERR=32) (IWRD(I), I=1,6)
0010          CALL CLOSE(2)
           C
           C
0011          IADW(1)=IWRD(3)
0012          IADW(2)=IWRD(4)
           C
           C      GET MAGNITUDE AND GAIN FROM A/D WORD
           C
0013          CALL ADC(IADW(1), MAG(1), GAIN(1))
0014          VOLTS(2)=4.0852*XTVR*FLOAT(MAG(2))/GAIN(2)
0015          VOLTS(1)=4.0852*FLOAT(MAG(1))/GAIN(1)
           C
           C      GET SAMPLE TEMP IN DEGREES F
           C
0016          CALL TEMPF(VOLTS(1), SPLTMP)
           C
           C      IF EITHER OF THE TWO FOLLOWING CONDITIONS HOLD,
           C      STORE THESE VALUES AND READ THE NEXT ONES. KEEP
           C      READING UNTIL THE REFERENCE TEMP HAS BEEN EXCEEDED
           C      IN THE PROPER DIRECTION.
           C
0017          IF (ISUG.EQ.0.AND.XTEM.LT.SPLTMP) GOTO 26
0019          IF (ISUG.EQ.1.AND.SPLTMP.LT.XTEM) GOTO 26

```

```

0021      IF (SPLTMP.NE,XTEM) GOTO 24
      C
      C      IF XTEM = SAMPLE TEMP, RETURN WITH REFERENCE TEMP
      C
0023      VREF=VOLTS(2)
0024      GOTO 25
0025 24      SLOPE=(VOLTS(2)-VHOLD)/(SPLTMP-THOLD)
0026      VREF=SLOPE*(XTEM-SPLTMP)+VOLTS(2)
0027 25      RETURN
0028 26      VHOLD=VOLTS(2)
0029      THOLD=SPLTMP
0030 30      CONTINUE
0031 32      STOP 'ERROR - REF TEMP NOT FOUND'
0032      END

```

ADC MACRO M1110 05-JUL-78 08:27 PAGE 1

```

1                                     .TITLE ADC
2                                     .IDENT/022778/
3
4                                     ;+
5                                     ;      FEB. 1978                                D. E. LACEY
6                                     ;
7                                     ;      THIS FORTRAN CALLABLE SUBROUTINE CONVERTS AN ARRAY,
8                                     ;      IADW, OF A/D READINGS INTO TWO ARRAY, (MAG AND GAIN),
9                                     ;      WHICH ARE RETURNED TO THE CALLING PROGRAM.
10                                    ;
11                                    ;      CALLING SEQUENCE:
12                                    ;
13                                    ;          CALL ADC(IADW(1),MAG(1),GAIN(1))
14                                    ;
15                                    ;
16 000000      ADC:
17 000000      005725      TST      (R5)+
18 000002      012504      MOV      (R5)+, R4                      ; R4 POINTS TO IADW
19 000004      012503      MOV      (R5)+, R3                      ; R3 POINTS TO MAG
20                                                    ; R5 POINTS TO GAIN
21
22                                    ;
23                                    ;      STORE CALLER'S ARRAY, IADW, IN SUBROUTINE ARRAY, ADW.
24 000006      012702      000220'    MOV      *ADW, R2
25 000012      012700      000004      MOV      #4, R0
26 000016      012422      10$:     MOV      (R4)+, (R2)+
27 000020      077002      SOB      R0, 10$
28
29                                    ;
30                                    ;      ISOLATE AND STORE MAGNITUDE BITS AS SIGNED INTEGER
31                                    ;      NUMBERS IN THE SUBROUTINE ARRAY, MAG.
32 000022      012704      000220'    MOV      *ADW, R4
33 000026      012702      000230'    MOV      *MAG, R2
34 000032      012700      000004      MOV      #4, R0
35 000036      012401      20$:     MOV      (R4)+, R1
36 000040      042701      170000      BIC      *170000,R1
37 000044      032701      004000      BIT      *4000, R1
38 000050      001001      30$      ONE      30$
39 000052      000402      40$      OR       40$
40 000054      052701      170000      30$:   BIS      *170000,R1

```

```

41 000060 010122
42 000062 077013
43
44
45
46
47 000064 012704 000220'
48 000070 012702 000240'
49 000074 012701 000004
50 000100 012700 000004
51 000104 000314
52 000106 006014
53 000110 077002
54 000112 042714 177760
55 000116 012422
56 000120 077111
57 000122 012703 000250'
58 000126 012702 000240'
59 000132 012700 000004
60 000136 012201
61 000140 000241
62 000142 006001
63 000144 103001
64 000146 005201
65 000150 006301
66 000152 006301
67 000154 062701 000270'
68 000160 012123
69 000162 012123
70 000164 077014
71
72
73
74 000166 012700 000004
75 000172 012702 000230'
76 000176 012701 000250'
77 000202 011504
78 000204 014503
79 000206 012223
80 000210 012124
81 000212 012124
82 000214 077004
83

```

```

40$: MOV R1, (R2)+
SOB R0, 20$
;
; ISOLATE, CONVERT AND STORE THE GAIN AS FLOATING POINT
; NUMBERS IN THE SUBROUTINE ARRAY, GAIN.
;
MOV *ADW, R4
MOV *IGAIN, R2
MOV *4, R1
50$: MOV *4, R0
SWAB (R4)
60$: ROR (R4)
SOB R0, 60$
BIC *177760, (R4)
MOV (R4)+, (R2)+
SOB R1, 50$
MOV *GAIN, R3
MOV *IGAIN, R2
MOV *4, R0
70$: MOV (R2)+, R1
CLC
ROR R1
BCC 00$
INC R1
80$: ASL R1
ASL R1
ADD *GTBL, R1
MOV (R1)+, (R3)+
MOV (R1)+, (R3)+
SOB R0, 70$
;
; RETURN ARRAYS, MAG AND GAIN, TO THE CALLING PROGRAM.
;
MOV *4, R0
MOV *MAG, R2
MOV *GAIN, R1
MOV (R5), R4
MOV -(R5), R3
90$: MOV (R2)+, (R3)+
MOV (R1)+, (R4)+
MOV (R1)+, (R4)+
SOB R0, 90$
;

```


84 000216 000207

85

86 000220

87 000230

88 000240

89 000250

90 000270 040200 000000

91 000274 040400 000000

92 000300 041040 000000

93 000304 041240 000000

94 000310 041510 000000

95 000314 041710 000000

96 000320 042110 000000

97 000324 042572 000000

98 000001

RETURN

;

ADW: .BLKW 4

MAG: .BLKW 4

IGAIN: .BLKW 4

GAIN: .BLKW 8.

GTBL: .FLT2 1.

.FLT2 2.

.FLT2 10.

.FLT2 20.

.FLT2 50.

.FLT2 100.

.FLT2 200.

.FLT2 1000.

.END

Appendix C

SAMPLES OF INTERACTIVE I/O AND OTHER OUTPUTS

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

FIELD
ACCOUNT OR NAME : HOLLAND
PASSWORD :

DEI LOGGED ONTO OTHER TERMINAL ALSO?

RSX TIM BL22 MULTI-USER SYSTEM

GOOD MORNING
28 OCT 78 10:33 LOGGED ON TERMINAL ITS:

PUN DELET+4 TIME XI
ENTER CHARGE NUMBER (8) : 4124304
ENTER 1ST INITIAL & LAST NAME OF PERSON REQUESTING
TEST (14) : J. McFARLAND
ENTER DEPT # OF PERSON REQUESTING TEST (4) : 814
ENTER MATERIAL ID OR TEST ID (12) : DC-306
ENTER TEST SPECIFICATION, IF ANY (12) : NONE
ENTER CURE (7) : 33/228
ENTER SPECIMEN ID (2) : 30
ENTER EMPLOYEE NUMBER (6) : 51545
ENTER DIFFERENCE TEMPERATURE (F.P.) : 22.
ENTER SPECIMEN'S INITIAL LENGTH (F.P.) : 2.01
NUMBER OF THERMAL CYCLES (1-122) : 2
DIRECTION OF TEMP CYCLE (1 UP+0-DOWN) : 0
MINIMUM LOW TEMPERATURE (F.P.) : 65.
MAXIMUM HIGH TEMPERATURE (F.P.) : 350.
STARTING A/D CHANNEL NUMBER (0-68) : 0
STARTING DELAY (0-12 OR 16 28) : 0
INITIAL INP. VAL. POSITION IN RTR TURNS (1-20) : 1
NUM. MINUTES BETWEEN READINGS (1-122) : 3
TWO-POW. PLOT (F.P.) : 1.
PRESS RETURN WHEN READY TO START :

>

THERMAL EXPANSION INFORMATION

05-JUL-78

0:50

LTR NO.: 337228

SPECIMEN IDENTIFICATION: 2A

EMPLOYEE NO.: 51545

DATA RECORDED: 0:30 - 5/18/78

NO. OF DATA SETS RECORDED: 357

NO OF MINUTES BETWEEN SAMPLES: 3

EXPERIMENT PARAMETERS:

INITIAL SPECIMEN LENGTH: 2.000

LVDT/MV/INCH : 1.000

STARTING CYCLE TEMP.: 77.00

NUMBER OF THERMAL CYCLES: 2

INITIAL SWING DIRECTION: DOWN

CURRENT CYCLE NUMBER: 3

HIGH-SIDE CYCLE TEMP.: 400.00

LOW-SIDE CYCLE TEMP: 70.00

INITIAL LN2, (QUARTER TURNS OPEN): 1

DEVICE ASSIGNMENTS FOR THE EXPERIMENT:

CHAN * 0 * SPECIMEN TEMPERATURE

CHAN * 1 * HEATER TEMPERATURE

CHAN * 2 * LVDT VOLTAGE

CHAN * 3 * HEATER VOLTAGE

CHAN * 0 * POWER SUPPLY RELAY

CHAN * 1 * LIQUID NITROGEN RELAY

CHAN * 2 * STEPPER MOTOR RELAY

DK1: [300.300]3372202A.TE2

23-OCT-70

10:23:26

PAGE 1

TIME HRS-MINS	HEATER TEMP.	SPECIMEN TEMP.	LVDT VOLTAGE	LN2 QTR. TURNS OPEN	X EXPANSION RAW	COMPENSATED
8:30	88.36	91.86	-3.33	1		
8:31	87.46	89.26	-3.39	1		
8:33	87.24	88.14	-3.43	1		
8:34	85.66	87.24	-3.46	1		
* 8:35 **	84.75 **	86.11 **	-3.49 ***	1		
8:36	83.64	84.99	-3.51	1		
8:37	81.61	84.09	-3.55	1		
8:38	79.81	82.74	-3.60	1		
8:39	76.66	80.71	-3.67	1		
* 8:40 **	75.76 **	79.59 **	-3.73 ***	1		
8:41	73.74	78.01	-3.80	1		
8:42	70.81	75.76	-3.87	1	-.0021	-.0022
8:42	70.81	75.76	-3.87	1	-.0019	-.0019
8:43	66.54	70.81	-4.00	1	-.0127	-.0120
* 8:48 **	66.99 **	67.44 **	-4.22 ***	1	-.0195 **	-.0190
8:51	65.64	65.86	-4.26	1	-.0217	-.0220
8:54	64.74	64.96	-4.30	1	-.0236	-.0240
8:57	63.39	63.61	-4.35	1	-.0261	-.0265
9: 0	61.59	62.04	-4.41	1	-.0290	-.0295
* 9: 3 **	59.56 **	60.46 **	-4.48 ***	1	-.0324 **	-.0330
9: 6	57.99	58.66	-4.55	1	-.0359	-.0364
9: 9	56.64	57.89	-4.61	1	-.0388	-.0394
9:12	55.86	55.51	-4.67	1	-.0420	-.0426
9:15	53.49	53.94	-4.73	1	-.0449	-.0456
* 9:18 **	53.49 **	53.84 **	-4.78 ***	1	-.0476 **	-.0483
9:21	53.29	53.49	-4.78	1	-.0476	-.0483
9:24	57.89	54.84	-4.77	1	-.0471	-.0478
9:27	59.11	56.64	-4.72	1	-.0444	-.0450
9:30	62.04	58.89	-4.65	1	-.0410	-.0415
* 9:33 **	65.19 **	61.81 **	-4.55 ***	1	-.0361 **	-.0366
9:36	68.79	65.19	-4.44	1	-.0305	-.0308
9:39	72.39	68.56	-4.38	1	-.0236	-.0239
9:42	76.66	72.39	-4.15	1	-.0161	-.0162
9:45	80.94	76.66	-4.00	1	-.0083	-.0083
* 9:48 **	85.66 **	81.16 **	-3.83 ***	1	0.0003 **	0.0004
9:51	89.94	85.44	-3.64	1	0.0096	0.0098
9:54	94.44	89.94	-3.46	1	0.0186	0.0190
9:57	98.26	94.21	-3.29	1	0.0269	0.0275
10: 0	101.64	97.01	-3.14	1	0.0347	0.0354
* 10: 3 **	104.79 **	101.19 **	-3.00 ***	1	0.0416 **	0.0423
10: 6	115.56	107.04	-2.81	1	0.0509	0.0518
10: 9	122.11	115.36	-2.46	1	0.0604	0.0606
10:12	130.66	123.69	-2.03	1	0.0699	0.0694
10:15	138.99	132.69	-1.52	1	0.1156	0.1173
* 10:18 **	143.44 **	141.69 **	-0.96 ***	1	0.1437 **	0.1457

23-OCT-78

10:23:54

PAGE 2

TIME HRS-MINS	HEATER TEMP.	SPECIMEN TEMP.	LVDT VOLTAGE	LN2 QTR TURNS OPEN	X EXPANSION RAW	COMPENSATED
10:21	156.99	150.69	-0.43	1	0.1703	0.1726
10:24	165.76	159.69	0.09	1	0.1962	0.1988
10:27	174.31	168.46	0.61	1	0.2221	0.2249
10:30	182.86	177.24	1.09	1	0.2468	0.2491
*10:33	** 191.19	** 185.34	1.55	****	0.2698	** 0.2723
10:36	199.29	193.66	1.99	1	0.2910	0.2946
10:39	206.94	201.76	2.45	1	0.3142	0.3180
10:42	214.81	209.64	2.89	1	0.3362	0.3403
10:45	222.69	217.51	3.32	1	0.3574	0.3618
*10:48	** 230.34	** 225.16	3.74	****	0.3787	** 0.3832
10:51	237.54	232.81	4.14	1	0.3987	0.4035
10:54	245.41	240.24	4.55	1	0.4192	0.4243
10:57	253.86	247.89	4.95	1	0.4392	0.4445
11: 0	260.49	255.54	5.38	1	0.4607	0.4663
*11: 3	** 268.14	** 263.19	5.79	****	0.4818	** 0.4868
11: 6	275.79	270.61	6.19	1	0.5013	0.5073
11: 9	283.21	278.04	6.59	1	0.5211	0.5273
11:12	290.64	285.69	7.03	1	0.5433	0.5498
11:15	298.29	293.11	7.44	1	0.5636	0.5703
*11:18	** 305.94	** 300.76	7.81	****	0.5819	** 0.5888
11:21	313.36	308.19	8.21	1	0.6019	0.6091
11:24	321.81	315.61	8.68	1	0.6217	0.6291
11:27	329.44	323.49	9.00	1	0.6415	0.6491
11:30	335.86	330.91	9.39	1	0.6613	0.6691
*11:33	** 343.51	** 338.34	9.79	****	0.6808	** 0.6889
11:36	350.94	345.99	10.16	1	0.6996	0.7079
11:39	358.36	353.64	10.55	1	0.7192	0.7277
11:42	365.79	361.06	10.94	1	0.7387	0.7475
11:45	373.66	368.49	11.33	1	0.7582	0.7673
*11:48	** 381.89	** 376.14	11.75	****	0.7790	** 0.7883
11:51	388.51	384.01	12.12	1	0.7973	0.8068
11:54	395.94	391.21	12.48	1	0.8156	0.8254
11:57	402.46	397.96	12.85	1	0.8348	0.8439
12: 0	408.31	404.26	13.17	1	0.8498	0.8600
*12: 3	** 412.81	** 409.21	13.41	****	0.8628	** 0.8723
12: 6	416.86	413.71	13.65	1	0.8743	0.8847
12: 9	420.81	416.86	13.88	1	0.8816	0.8921
12:12	422.49	419.79	13.92	1	0.8877	0.8983
12:15	424.29	421.81	14.02	1	0.8926	0.9033
*12:18	** 425.66	** 423.39	14.09	****	0.8962	** 0.9070
12:21	426.99	424.74	14.14	1	0.8987	0.9095
12:24	428.11	425.64	14.19	1	0.9011	0.9119
12:27	428.79	426.54	14.22	1	0.9023	0.9132
12:30	429.46	426.99	14.24	1	0.9036	0.9144
*12:33	** 429.69	** 427.66	14.24	****	0.9036	** 0.9144

DK1: (388,388)3372282A.TE2

23-OCT-78

18:24:19

PAGE 3

TIME HRS-MINS	HEATER TEMP.	SPECIMEN TEMP.	LVDT VOLTAGE	LN2 QTR. TURNS OPEN	* EXPANSION RAW	COMPENSATED
12:36	438.14	428.11	14.26	1	0.9848	0.9157
12:39	438.14	428.11	14.26	1	0.9848	0.9157
12:42	438.36	428.34	14.24	1	0.9836	0.9145
12:45	438.36	428.34	14.26	1	0.9848	0.9157
*12:48	*438.36	*428.56	*14.26	*1	*0.9848	*0.9157
12:51	438.36	428.56	14.26	1	0.9848	0.9157
12:54	438.36	428.56	14.24	1	0.9836	0.9145
12:57	438.59	428.56	14.26	1	0.9848	0.9157
13: 0	438.36	428.56	14.24	1	0.9836	0.9145
*13: 3	*438.59	*428.56	*14.26	*1	*0.9848	*0.9157
13: 6	438.59	428.56	14.26	1	0.9848	0.9157
13: 9	438.59	428.56	14.24	1	0.9836	0.9145
13:12	438.59	428.79	14.26	1	0.9848	0.9157
13:15	438.59	428.56	14.26	1	0.9848	0.9157
*13:18	*438.59	*428.56	*14.26	*1	*0.9848	*0.9157
13:21	438.36	428.56	14.26	1	0.9848	0.9157
13:24	438.59	428.56	14.26	1	0.9848	0.9157
13:27	438.36	428.56	14.24	1	0.9836	0.9145
13:30	428.34	426.76	14.17	1	0.8999	0.9107
*13:33	*426.76	*425.41	*14.09	*1	*0.8962	*0.9070
13:36	425.86	424.29	14.02	1	0.8926	0.9033
13:39	424.96	423.39	13.97	1	0.8981	0.9009
13:42	424.51	422.94	13.95	1	0.8889	0.8996
13:45	424.86	422.49	13.90	1	0.8865	0.8972
*13:48	*423.61	*421.81	*13.87	*1	*0.8852	*0.8959
13:51	422.94	421.14	13.83	1	0.8828	0.8935
13:54	421.81	420.24	13.75	1	0.8791	0.8898
13:57	419.79	418.66	13.68	1	0.8755	0.8861
14: 0	417.89	416.19	13.56	1	0.8694	0.8799
*14: 3	*413.71	*413.26	*13.39	*1	*0.8608	*0.8712
14: 6	409.44	409.21	13.17	1	0.8498	0.8601
14: 9	404.94	404.94	12.95	1	0.8388	0.8490
14:12	399.76	399.76	12.70	1	0.8266	0.8366
14:15	394.36	394.81	12.43	1	0.8132	0.8230
*14:18	*388.29	*388.96	*12.14	*1	*0.7985	*0.8082
14:21	382.89	383.34	11.85	1	0.7839	0.7934
14:24	376.59	377.49	11.55	1	0.7692	0.7785
14:27	370.74	371.41	11.24	1	0.7533	0.7625
14:30	364.21	363.11	10.92	1	0.7375	0.7464
*14:33	*357.91	*358.81	*10.60	*1	*0.7216	*0.7303
14:36	351.39	352.51	10.28	1	0.7057	0.7143
14:39	343.89	345.99	9.97	1	0.6898	0.6982
14:42	338.56	339.69	9.63	1	0.6730	0.6811
14:45	332.26	333.39	9.30	1	0.6566	0.6646
*14:48	*325.51	*326.86	*8.96	*1	*0.6398	*0.6475

DK1:1300.30013372202A.YE2

23-OCT-78

10:24:40

PAGE 4

TIME HRS-MINS	HEATER TEMP.	SPECIMEN TEMP.	LVDT VOLTAGE	LN2 QTR. TURNS OPEN	% EXPANSION RAW	% EXPANSION COMPENSATED
14:51	318.99	320.34	8.64	1	0.6236	0.6312
14:54	312.69	314.04	8.31	1	0.6078	0.6144
14:57	306.16	307.51	7.97	1	0.5902	0.5973
15: 0	299.41	300.99	7.64	1	0.5733	0.5803
*15: 3	** 292.89	** 294.69	** 7.29	** 1	** 0.5568	** 0.5627
15: 6	286.14	288.16	6.96	1	0.5394	0.5459
15: 9	279.61	281.64	6.61	1	0.5223	0.5286
15:12	272.86	275.11	6.28	1	0.5057	0.5118
15:15	266.34	268.59	5.94	1	0.4886	0.4945
*15:18	** 259.59	** 262.29	** 5.60	** 1	** 0.4715	** 0.4772
15:21	253.06	255.76	5.25	1	0.4541	0.4597
15:24	246.54	249.01	4.90	1	0.4365	0.4419
15:27	239.79	242.71	4.56	1	0.4194	0.4246
15:30	233.26	236.19	4.21	1	0.4021	0.4070
*15:33	** 226.74	** 229.89	** 3.86	** 1	** 0.3848	** 0.3895
15:36	220.21	223.36	3.52	1	0.3677	0.3722
15:40	213.91	217.06	3.18	1	0.3503	0.3547
15:43	207.16	210.76	2.82	1	0.3327	0.3369
15:46	200.64	204.01	2.48	1	0.3154	0.3193
*15:49	** 194.34	** 197.94	** 2.17	** 1	** 0.3002	** 0.3048
15:52	188.04	191.64	1.82	1	0.2827	0.2862
15:55	181.51	185.11	1.47	1	0.2651	0.2684
15:58	175.21	179.04	1.13	1	0.2490	0.2511
16: 1	169.14	172.96	0.79	1	0.2311	0.2341
*16: 4	** 163.51	** 167.11	** 0.47	** 1	** 0.2152	** 0.2188
16: 7	157.89	161.26	0.17	1	0.2001	0.2027
16:10	152.71	156.09	-0.12	1	0.1857	0.1881
16:13	147.76	150.91	-0.39	1	0.1720	0.1743
16:16	143.04	145.96	-0.65	1	0.1591	0.1612
*16:19	** 138.76	** 141.69	** -0.88	** 1	** 0.1473	** 0.1493
16:22	134.49	137.41	-1.11	1	0.1361	0.1380
16:25	130.66	133.36	-1.33	1	0.1251	0.1269
16:28	127.06	129.54	-1.53	1	0.1149	0.1165
16:31	123.69	126.16	-1.71	1	0.1058	0.1073
*16:34	** 120.76	** 122.56	** -1.88	** 1	** 0.0975	** 0.0989
16:37	117.84	119.64	-2.03	1	0.0899	0.0913
16:40	114.91	116.71	-2.17	1	0.0829	0.0841
16:43	112.44	113.79	-2.31	1	0.0763	0.0774
16:46	109.96	111.54	-2.43	1	0.0702	0.0712
*16:49	** 107.71	** 109.29	** -2.55	** 1	** 0.0643	** 0.0653
16:52	105.46	106.81	-2.65	1	0.0589	0.0598
16:55	103.66	104.79	-2.75	1	0.0543	0.0551
16:58	101.64	102.99	-2.83	1	0.0499	0.0507
17: 1	100.06	100.96	-2.92	1	0.0455	0.0462
*17: 4	** 98.26	** 99.39	** -3.00	** 2	** 0.0416	** 0.0423

TEST LABORATORY REPORT

TO: J. MCFARLAND D/814

DATE: 19-JUL-78

LTR NO.: 337228

E. P. NO.: 4124384

MATERIAL: DC-306

TEST SPECIFICATIONS: NONE

TEST RESULTS:

SEE ATTACHED GRAPHS.

TESTED BY: L. HOLLAND /TE2000-VERS1 /DK1:C300,30013372282A.TED

APPROVED BY:

DC-306

LTR# 337228
SPECIMEN ID# 2A
CYCLE# 1

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

DISTRIBUTION

	Copy
R. Bulcock, DOE-KCAO	1
R. Caudle, DOE-MA, Washington, DC	2
A. M. Kray, LLL	3
J. E. Long, D/531, 1A46	4
H. T. Barnes, D/554, BD50	5-6
L. Stratton, D/554, 2C44	7-9
R. P. Frohberg, D/800, 2A39	10
W. H. Deterding, D/814, 2C43	11
D. S. Genser, D/816, SA-1	12
L. L. Holland, D/816, SA-1	13-15
D. E. Lacey, D/816, SA-1	16
G. L. MacCurdy, D/816, SA-1	17
W. J. Stone, D/816, SA-1	18
R. E. Kessler, D/865, 2C40	19

BDX-613-2085 (Rev.)

AUTOMATION OF A THERMAL EXPANSION INSTRUMENT,
L. L. Holland, Topical, March 1979.

Automation of a thermal expansion instrument using a minicomputer system and with analog-to-digital converter inputs and flip-flop relay outputs is complete. The necessary hardware link is connected and the software developed to allow equipment control, data acquisition, data reduction, and report generation by the minicomputer. The design of the automation allows non-programmers to run the experiment, reduce the data, and generate the report.

SPECIAL PROJECTS: Automated Testing

AUTOMATION OF A THERMAL EXPANSION INSTRUMENT,
L. L. Holland, Topical, BDX-613-2085 (Rev.),
March 1979.

Automation of a thermal expansion instrument using a minicomputer system and with analog-to-digital converter inputs and flip-flop relay outputs is complete. The necessary hardware link is connected and the software developed to allow equipment control, data acquisition, data reduction, and report generation by the minicomputer. The design of the automation allows non-programmers to run the experiment, reduce the data, and generate the report.

AUTOMATION OF A THERMAL EXPANSION INSTRUMENT,
L. L. Holland, Topical, BDX-613-2085 (Rev.),
March 1979.

Automation of a thermal expansion instrument using a minicomputer system and with analog-to-digital converter inputs and flip-flop relay outputs is complete. The necessary hardware link is connected and the software developed to allow equipment control, data acquisition, data reduction, and report generation by the minicomputer. The design of the automation allows non-programmers to run the experiment, reduce the data, and generate the report.