

```

10 ! MULTIPLE CREEP TEST DATA ACQUISITION PROGRAM: "MultiCrp"
20
30 ! J. C. Gibeling ** April 25, 1983
40 ! revised ** January 06, 1989
41 ! revised ** December 10, 1999
50
60 ! Division of Materials Science and Engineering
70 ! Department of Mechanical Engineering
80 ! University of California
90 ! Davis, California 95616
100
110 ! This program is to be used for data acquisition from one to five
120 ! creep tests (tension or compression). It is designed to collect data
130 ! points at the chosen time interval for each test. The range of time
140 ! intervals which is available is from 2 sec to 60 min per point.
150
160 ! This program uses BASIC 3.0, 4.0 or 5.0 on an HP 9000 BASIC Language
170 ! Processor with an HP 3497A Data Acquisition and Control Unit and an
180 ! internal hard drive. An external printer is optional. The following
190 ! binary programs are required:
200 !
210 !           GRAPH          ERR
220 !           IO             DISC
230 !           TRANS         CS80
240 !           MAT            HPIB
250 !           KBD            SERIAL
260 !           CLOCK          CRTB
270 !           MS              CRTA
280 !
290 !
300 OPTION BASE 1
310 COM /Data/ Load(5,500),Elong(5,500),Time(5,500),Temp(5,500),Status(5),Tot
al_temp(5)
320 COM /Data/ Material$(5)[20],Sample_number$(5)[7],INTEGER Test_type(5),Po
int_counter(5),Number_points(5)
330 COM /Paths/ @Data_acq_unit
340 COM /Flags/ INTEGER Display_test,Test_running(5),Data_transfer(5),High_sp
d_flag(5),Control_flag(5),Fast_points(5),End_flag(5),Plot_flag
350 COM /Disc/ Store_device$[8],Drive$[5],Primary_label$[6],Backup_label$[6],
File_name$[10],Parm_file$(5)[10]
360 COM /Disc/ Check$[1],INTEGER File(5),Switch_drive(5),Primary_space,Backup
_space,Full_flag
370 COM /Display1/ Time_origin(5),Strain_origin(5),Time_span(5),Strain_span(5
)
380 COM /Parms/ Area(5),Gage_length(5),Elong_cal(5),Load_cal(5),Start(5),Fin
ish(5),Elong_0(5),Tc_cal(5),Max_load(5),Max_elong(5),Nom_load(5)
381 COM /Parms/ Comment$(5)[150],Computer$[8],INTEGER Test_frame
400 COM /Timing/ INTEGER Interval(5),Data_interval,Interval_flag
401 COM /Tparm/ Parm(0:8,1:5)
410 DIM Date$[11],Time$[8],Response$[20],Print_device$[5]
420 INTEGER I,J,Point,Check,Ser_poll
430 !
440 !
450 GRAPHICS OFF
460 DEG
470 CONTROL 2,1;0 ! Cancel PRINTALL mode
480 CONTROL 1,4;0 ! Cancel DISPLAY FUNCTIONS mode
490 CONTROL 2,2;1 ! Turn on USER softkeys
500 Computer$=SYSTEM$("SYSTEM ID") ! Determine computer model
510 DUMP DEVICE IS 26 ! Parallel printer

```

```

520 !ON TIMEOUT 26,.1 GOTO No_ext_printer
530 !Ser_poll=SPOLL(701)
540 !OUTPUT 701;CHR$(27)&"*rK"           ! Request printer model number
550 !ENTER 701 USING "5A";Print_device$ 
560 !
570 No_ext_printer:   !
580 !
590 !OFF TIMEOUT
591 !RESET 7
600 !SELECT Print_device$
610 !CASE "2673A"
620     !OUTPUT 701;CHR$(27)&"*r90X"      ! Horizontal graphics centering
630 !CASE ""
640     !IF Ser_poll<>0 THEN
650         !Print_device$="2225A"          ! Assume HP 2225A printer
660         !OUTPUT 701;CHR$(27)&"&11L"    ! Perforation skip
670         !OUTPUT 701;CHR$(27)&"&sOC"    ! End-of-line wrap-around
680     !END IF
690 !END SELECT
700 PRINTER IS 1                      ! CRT
710 MASS STORAGE IS "\BLP\DATA:DOS,C"  ! Use system disc for data
720 OUTPUT 2 USING "#,B";255,75        ! [CLR SCR]
730 !
740 !
750 PRINT TABXY(15,4);"MULTIPLE CREEP TEST DATA ACQUISITION PROGRAM"
760 FOR I=1 TO 5
770     Status(I)=0                    ! Clear test status signal values
780     Test_running(I)=0              ! Clear test running flags
790     Data_transfer(I)=1            ! Set data transferred flags
800     High_spd_flag(I)=0            ! Clear high speed data flag
810     End_flag(I)=0                ! Clear end-of-test flag
820 NEXT I
830 !
840 Interval_flag=0                  ! Clear data interval flag
850 Plot_flag=1                      ! Enable data plotting
860 Display_test=1                  ! Default is to display data from 1
oad frame 1
870 !
880 !ON ERROR GOTO 910
890 !DELSUB Phyread                 ! Delete previous disc utilities
900 !DELSUB Phywrite
910 !OFF ERROR
920 !LOADSUB ALL FROM "PHYREC"
921 !
923 !DISP USING "K";"Please insert the system data disc in the left hand driv
e, then press CONTINUE."
924 !CONTROL 2,2;0                  ! Turn on SYSTEM softkeys
925 !PAUSE
926 CONTROL 2,2;1                  ! Turn on USER softkeys
927 OUTPUT 2 USING "#,B";255,75    ! [CLR SCR]
930 !
940 !
950 CALL Set_clock
960 !
970 !
980 ! SET-UP 3497A DATA ACQUISITION/CONTROL UNIT
990 !
1000 ASSIGN @Data_acq_unit TO 709
1010 ON TIMEOUT 7,.1 CALL No_acq_unit
1020 Ser_poll=SPOLL(@Data_acq_unit)

```

```

1030 OFF TIMEOUT
1040 CLEAR @Data_acq_unit
1050 RESET 7
1060 !
1070 OUTPUT @Data_acq_unit USING "K";"VT4VD5VR3VA1VF1VN20VS1AF0AL19AC0AE1SO1SE
001"
1080 ! VT4: Hold DVM trigger
1090 ! VD5: 5-1/2 digit DVM output
1100 ! VR3: 10 Volt DVM range
1110 ! VA1: DVM Auto-zero on
1120 ! VF1: ASCII output format
1130 ! VN20: 20 readings per trigger
1140 ! VS1: DVM storage on
1150 ! AF0: First channel is 0
1160 ! AL19: Last channel is 19
1170 ! AC0: Close channel 0
1180 ! AE1: Enable external increment
1190 ! SO1: System output wait on
1200 ! SE001: Data ready SRQ mask
1201 OUTPUT @Data_acq_unit;"VC3" ! 1 mA current for status signals
1210 !
1220 !
1230 CALL Test_setup
1240 !
1250 OUTPUT 2 USING "#,B";255,75 ! [CLR SCR]
1260 ON INTR 7,5 CALL Data ! Set-up SRQ service routine
1270 ENABLE INTR 7;2 ! Enable SRQ interrupt on HPIB
1280 !
1290 OFF KEY
1300 SELECT Computer$[1,4]
1310 CASE "9817","S300","PC30"
1320     ON KEY 1 LABEL "Select plot" CALL Select_plot
1330     ON KEY 2 LABEL "New data rate" CALL Interval
1340     ON KEY 3 LABEL " New spans" CALL New_spans
1350     !ON KEY 4 LABEL " Data transfer" CALL Transfer
1360     ON KEY 5 LABEL " Start test" CALL Test_setup
1370     ON KEY 6 LABEL "End test" CALL End_of_test
1380     ON KEY 8 LABEL "End all tests" GOSUB End_all
1390 CASE ELSE
1400     ON KEY 0 LABEL "Select plot" CALL Select_plot
1410     ON KEY 1 LABEL "New data rate" CALL Interval
1420     ON KEY 2 LABEL "New spans" CALL New_spans
1430     !ON KEY 3 LABEL "Data transfer" CALL Transfer
1440     ON KEY 4 LABEL "Start test" CALL Test_setup
1450     ON KEY 5 LABEL "End test" CALL End_of_test
1460     ON KEY 9 LABEL "End all tests" GOSUB End_all
1470 END SELECT
1480 !
1490 !
1500 OUTPUT @Data_acq_unit;"VT2" ! DVM in external trigger mode
1510 OUTPUT 2 USING "#,B";255,75 ! [CLR SCR]
1520 !
1530 GRAPHICS ON
1540 !
1550 !
1560 Idle3: ! Main program loop
1570 Point=0
1580 IF Point_counter(Display_test)>1 THEN Point=Point_counter(Display_test)-1
1590 IF Point>0 THEN
1600     SYSTEM PRIORITY 2 ! Raise priority to avoid ON KEY's

```

```

1610      !
1620      !
1630      ! INCREASE SPANS IF NECESSARY
1640      !
1650      IF ABS(Time(Display_test,Point)-Start(Display_test))/3600>ABS(Time_span
(Display_test)) THEN
1660          Time_span(Display_test)=2*Time_span(Display_test)
1670          CALL Strain_plot(Display_test)
1680      END IF
1690      !
1700      IF Test_type(Display_test)=1 THEN
1705          Strain=100*ABS(Elong_cal(Display_test)*(Elong(Display_test,Point)-Elo
ng_0(Display_test))/Gage_length(Display_test))
1710          IF Strain>ABS(Strain_span(Display_test))+Strain_origin(Display_test)
THEN
1720              Strain_span(Display_test)=2*Strain_span(Display_test)
1730              CALL Strain_plot(Display_test)
1740          END IF
1750      ELSE
1760          IF 100*ABS(Elong_cal(Display_test)*(Elong(Display_test,Point)-Elong_0(
Display_test)))>ABS(Strain_span(Display_test))+Strain_origin(Display_test) THEN
1770              Strain_span(Display_test)=2*Strain_span(Display_test)
1780              CALL Strain_plot(Display_test)
1790          END IF
1800      END IF
1810      SYSTEM PRIORITY 0                      ! Return to normal priority
1820  END IF
1830  !
1840  !
1850  ! END A TEST IF THAT STATUS SIGNAL IS ZERO
1860  !
1870 FOR I=1 TO 5
1880     IF Test_running(I) THEN
1890         Point=0
1900         IF Point_counter(I)>1 THEN Point=Point_counter(I)-1
1910         IF Point>0 THEN
1920             IF (ABS(Load(I,Point))>ABS(Max_load(I))) THEN Max_load(I)=Load(I,Po
int)
1930             IF (ABS(Elong(I,Point))>ABS(Max_elong(I))) THEN Max_elong(I)=Elong(
I,Point)
1940             IF ABS(Status(I))<.001 THEN End_flag(I)=1
1950             IF End_flag(I) THEN CALL End_of_test(I)
1960             !
1970             IF High_spd_flag(I)=1 THEN
1980                 IF Point+1+(File(I)-1)*500>Fast_points(I) THEN
1990                     High_spd_flag(I)=0           ! Clear high speed data flag
2000                     Data_interval=Interval(I)
2010                     FOR J=1 TO 5               ! Find minimum data interval
2020                         IF (Interval(J)>0 AND Interval(J)<Data_interval) THEN Data_in
terval=Interval(J)
2030                         NEXT J
2040                         Interval_flag=1           ! Output selected data interval
2050                     END IF
2060                     END IF
2070                 END IF
2080             END IF
2090     NEXT I
2100     !
2110     IF Interval_flag THEN
2120         OUTPUT @Data_acq_unit USING "K,6Z";"VT2TI",100*(Data_interval DIV 60)+(

```

```

Data_interval MOD 60)
2130     Interval_flag=0
2140 END IF
2150 !
2160 !
2170 ! EXIT PROGRAM IF NO TESTS RUNNING
2180 !
2190 Check=0
2200 FOR I=1 TO 5
2210     Check=Check+Test_running(I)
2220 NEXT I
2230 IF NOT Check THEN GOSUB End_all
2240 !
2250 DISP                                ! Clear display line
2260 GOTO Idle3                          ! Loop to wait for data
2270 !
2280 !
2290 ! ***** SUBROUTINES FOR MAIN PROGRAM *****
2300 !
2310 !
2320 End_all: !
2330 !
2340 FOR I=1 TO 5
2350     IF Test_running(I) THEN
2360         INPUT "Do you wish to end all of the currently running tests? (Y/N)", Response$
2370     IF UPC$(Response$[1,1])<>"Y" THEN
2380         RETURN
2390     ELSE
2400         GOTO 2460
2410     END IF
2420 END IF
2430 NEXT I
2440 !
2450 OFF INTR
2460 CLEAR 709
2470 !
2480 FOR I=1 TO 5
2490     IF Test_running(I) THEN CALL End_of_test(I)
2500 NEXT I
2510 !
2520 FOR I=1 TO 5                         ! Transfer data to floppy disc
2530     IF NOT Data_transfer(I) THEN CALL Transfer(I)
2540 NEXT I
2550 !
2551 PRINTER IS 1
2560 GRAPHICS OFF
2570 DISP "All tests have ended."
2580 END
2590 !
2600 !
2610 !***** SUBPROGRAMS AND SUBFUNCTIONS*****
2620 !
2630 !
2640 SUB Disc_setup2(INTEGER File_size)
2650 !
2660     ! J. C. Gibeling ** November 06, 1982
2670     !             revised ** July 19, 1988
2680 !
2690     ! This subroutine is used to set_up discs for test data storage.

```

```

2700 ! Each disc is checked for sufficient space, new discs are
2710 ! initialized, and volume labels are checked and assigned.
2720 !
2730 ! Note that the primary disc is placed in drive 0 and the backup
2740 ! disc in drive 1.
2750 !
2760 OPTION BASE 1
2770 COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_name
$,$,Parm_file$(*)
2780     COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Backup
_space,Full_flag
2790 !
2800 Primary_label$=""
2810 Backup_label$=""
2820 SELECT Store_device$
2830     CASE "Internal"
2840         PRINT TABXY(1,3);;"Insert the primary data disc in the right hand d
rive, and a second disc in the"
2850         PRINT "left hand drive."
2860         CASE "HP_8290X","HP_9122D"
2870             PRINT TABXY(1,3);;"Insert the primary data disc in the left hand dr
ive, and a second disc in the"
2880             PRINT "right hand drive."
2890             CASE "HP_9133D"
2900                 PRINT TABXY(1,3);;"Insert the primary data disc in the floppy drive
."
2910 END SELECT
2920 !
2930 PRINT USING "2/,K";;"Test data will first be stored on the primary data
disc (until it"
2940 PRINT "is full), and then on the secondary data disc. It is your respo
nsibility"
2950 PRINT "to be certain that the data do not overflow the second disc."
2960 DISP USING "K";;"Press ",CHR$(129),"CONTINUE",CHR$(128)," when you are
ready."
2961 CONTROL 2,2;0                      ! Turn on SYSTEM softkeys
2962 PAUSE
2963 CONTROL 2,2;1                      ! Turn on USER softkeys
2964 OUTPUT 2 USING "#,B";255,75        ! [CLR SCR]
3010 SELECT Store_device$
3020     CASE "Internal"
3030         MASS STORAGE IS ":INTERNAL,4,0"
3040         Drive$="Right"
3050         CASE "HP_8290X"
3060             MASS STORAGE IS ":HP8290X,702,0"
3070             Drive$="Left "
3080             CASE "HP_9122D"
3090                 MASS STORAGE IS ":CS80,702,0"
3100                 Drive$="Left "
3110                 CASE "HP_9133D"
3120                     MASS STORAGE IS ":CS80,700,1"
3130                     Drive$="Left "
3140 END SELECT
3150 ON ERROR CALL Disc_error
3160 CALL Free_space(Primary_label$,Primary_space)
3170 IF Primary_space<File_size THEN
3180     PRINT TABXY(1,4);;"The primary data disc does not have"
3190     PRINT "enough room for more data. Insert a new disc and"
3200     PRINT USING "K";;"press ";CHR$(129);;"CONTINUE";CHR$(128);"."
3210     CONTROL 2,2;0                      ! Turn on SYSTEM softkeys

```

```

3220     PAUSE
3230     CONTROL 2,2;1
3240     OUTPUT 2 USING "#,B";255,75      ! Turn on USER softkeys
3250     GOTO 3160
3260 END IF
3270 SELECT Store_device$               ! [CLR SCR]
3280 CASE "Internal"
3290     MASS STORAGE IS ":INTERNAL,4,1"
3300     Drive$="Left"
3310 CASE "HP_8290X"
3320     MASS STORAGE IS ":HP8290X,702,1"
3330     Drive$="Right"
3340 CASE "HP_9122D"
3350     MASS STORAGE IS ":CS80,702,0"
3360     Drive$="Right"
3370 CASE "HP_9133D"
3380     GOTO 3510
3390 END SELECT
3400 CALL Free_space(Backup_label$,Backup_space)
3410 IF Backup_space<File_size THEN
3420     PRINT TABXY(1,4); "The backup data disc does not have"
3430     PRINT "enough room for more data. Insert a new disc and"
3440     PRINT USING "K"; "press ",CHR$(129),"CONTINUE",CHR$(128),"."
3450     CONTROL 2,2;0                      ! Turn on SYSTEM softkeys
3460     PAUSE
3470     CONTROL 2,2;1                      ! Turn on USRE softkeys
3480     OUTPUT 2 USING "#,B";255,75        ! [CLR SCR]
3490     GOTO 3400
3500 END IF
3510 OFF ERROR
3520 !
3530 ! Check to see if discs have same volume label
3540 !
3550 IF Backup_label$=Primary_label$ THEN
3560     PRINT TABXY(1,3); "The two discs have the same volume label. Do you
wish to change one? (Y/N)"
3570 INPUT Response$
3580 IF Response$[1,1]="Y" OR Response$[1,1]="y" THEN
3590     INPUT "Which label do you wish to change (Right or Left)?",Drive$
3600     IF Drive$[1,1]="R" OR Drive$[1,1]="r" THEN
3610         SELECT Store_device$           ! [CLR SCR]
3620         CASE "Internal"
3630             MASS STORAGE IS ":INTERNAL,4,0"
3640             Drive$="Right"
3650             CALL Vol_label(Drive$,Primary_label$)
3660 CASE "HP_8290X"
3670     MASS STORAGE IS ":HP8290X,702,1"
3680     Drive$="Right"
3690     CALL Vol_label(Drive$,Backup_label$)
3700 CASE "HP_9122D"
3710     MASS STORAGE IS ":CS80,702,1"
3720     Drive$="Right"
3730     CALL Vol_label(Drive$,Backup_label$)
3740 END SELECT
3750 ELSE
3760     SELECT Store_device$           ! [CLR SCR]
3770     CASE "Internal"
3780         MASS STORAGE IS ":INTERNAL,4,1"
3790         Drive$="Left"
3800         CALL Vol_label(Drive$,Backup_label$)

```

```

3810      CASE "HP_8290X"
3820          MASS STORAGE IS ":HP8290X,702,0"
3830          Drive$="Left "
3840          CALL Vol_label(Drive$,Primary_label$)
3850      CASE "HP_9122D"
3860          MASS STORAGE IS ":CS80,702,0"
3870          Drive$="Left "
3880          CALL Vol_label(Drive$,Primary_label$)
3890      END SELECT
3900  END IF
3910      OUTPUT 2 USING "#,B";255,75    ! [CLR SCR]
3920  END IF
3930 END IF
3940 !
3950 SELECT Store_device$
3960 CASE "Internal"
3970     MASS STORAGE IS ":INTERNAL,4,0" ! Start with right hand disc dri
ve
3980     Drive$="Right"
3990 CASE "HP_8290X"
4000     MASS STORAGE IS ":HP8290X,702,0" ! Start with left hand disc driv
e
4010     Drive$="Left "
4020 CASE "HP_9122D"
4030     MASS STORAGE IS ":CS80,702,0" ! Start with left hand disc driv
e
4040     Drive$="Left "
4050 CASE "HP_9133D"
4060     MASS STORAGE IS ":CS80,700,1" ! Start with "left" hand disc.
4070     Drive$="Left "
4080 END SELECT
4090 !
4100 SUBEND
4110 !
4120 !
4130 SUB Disc_error
4140 !
4150 ! J. C. Gibeling ** November 06, 1982
4160 ! revised ** May 16, 1985
4170 !
4180 ! Subprogram to trap disc related errors
4190 !
4200 COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_name
$,Parm_file$(*)
4210     COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Backup
_space,Full_flag
4220 !
4230     OUTPUT 2 USING "#,B";255,75
4240     SELECT ERRN
4250     CASE 80
4260         BEEP
4270         PRINT TABXY(1,12);"""
4280         IF Store_device$="HP_9133D" THEN
4290             PRINT USING "K";"Floppy disc changed or not in drive -- fix"
4300             PRINT USING "K";"and press ",CHR$(129),"CONTINUE",CHR$(128)," ."
4310         ELSE
4320             PRINT USING "K";Drive$" hand disc changed or not in drive -- fi
x"
4330             PRINT USING "K";"and press ",CHR$(129),"CONTINUE",CHR$(128)," ."
4340     END IF

```

```

4350     CONTROL 2,2;0           ! Turn on SYSTEM softkeys
4360     PAUSE
4370     CONTROL 2,2;1           ! Turn on USER softkeys
4380     !
4390 CASE 83
4400     BEEP
4410     PRINT TABXY(1,12);"""
4420     IF Store_device$="HP_9133D" THEN
4430         PRINT USING "K";"Floppy disc write protected -- fix and"
4440         PRINT USING "K";"press ",CHR$(129),"CONTINUE",CHR$(128),"."
4450     ELSE
4460         PRINT USING "K";Drive$;" hand disc write protected -- fix and"
4470         PRINT USING "K";"press ",CHR$(129),"CONTINUE",CHR$(128),"."
4480     END IF
4490     CONTROL 2,2;0           ! Turn on SYSTEM softkeys
4500     PAUSE
4510     CONTROL 2,2;1           ! Turn on USER softkeys
4520     !
4530 CASE 84
4540     BEEP
4550     PRINT TABXY(1,14);"Record not found."
4560     IF Store_device$="HP_9133D" THEN
4570         PRINT USING "K";"Is the floppy disc initialized? (Y/N)"
4580     ELSE
4590         PRINT USING "K";"Is the ",Drive$," hand disc initialized? (Y/N)"
4600     END IF
4610     INPUT Response$
4620     IF Response$[1,1]="N" OR Response$[1,1]="n" THEN
4630         OUTPUT 2 USING "#,B";255,75
4640         PRINT TABXY(1,16);"""
4650         IF Store_device$="HP_9133D" THEN
4660             PRINT USING "#,K";" To initialize the floppy disc, press ",CHR
$129,"CONTINUE"
4670             PRINT USING "K";CHR$(128),"."
4680         ELSE
4690             PRINT USING "#,K";" To initialize the ";Drive$;" hand disc, pr
ess ",CHR$(129),"CONTINUE"
4700             PRINT USING "K";CHR$(128),"."
4710         END IF
4720         CONTROL 2,2;0           ! Turn on SYSTEM softkeys
4730         PAUSE
4740         CONTROL 2,2;1           ! Turn on USER SOFTKEYS
4750         OUTPUT 2 USING "#,B";255,75
4760     IF Drive$="Right" THEN
4770         SELECT Store_device$
4780         CASE "Internal"
4790             INITIALIZE ":INTERNAL,4,0"
4800             CALL Vol_label(Drive$,Primary_label$)
4810             CASE "HP_8290X"
4820                 INITIALIZE ":HP8290X,702,1"
4830                 CALL Vol_label(Drive$,Backup_label$)
4840             CASE "HP_9122D"
4850                 INITIALIZE ":CS80,702,1"
4860                 CALL Vol_label(Drive$,Backup_label$)
4870             CASE "HP_9133D"
4880                 INITIALIZE ":CS80,700,1"
4890                 CALL Vol_label(Drive$,Backup_label$)
4900             END SELECT
4910         END IF
4920     IF Drive$="Left " THEN

```

```

4930      SELECT Store_device$  

4940          CASE "Internal"  

4950              INITIALIZE ":INTERNAL,4,1"  

4960              CALL Vol_label(Drive$,Backup_label$)  

4970          CASE "HP_8290X"  

4980              INITIALIZE ":HP8290X,702,0"  

4990              CALL Vol_label(Drive$,Primary_label$)  

5000          CASE "HP_9122D"  

5010              INITIALIZE ":CS80,702,0"  

5020              CALL Vol_label(Drive$,Primary_label$)  

5030          CASE "HP_9133D"  

5040              INITIALIZE ":CS80,700,1"  

5050              CALL Vol_label(Drive$,Primary_label$)  

5060      END SELECT  

5070  END IF  

5080 ELSE  

5090     CONTROL 2,2;0           ! Turn on SYSTEM softkeys  

5100     PAUSE  

5110     CONTROL 2,2;1           ! Turn on USER softkeys  

5120 END IF  

5130 !  

5140 CASE 85  

5150 BEEP  

5160 DISP "Disc not initialized. Please wait."  

5170 IF Drive$="Right" THEN  

5180     SELECT Store_device$  

5190         CASE "Internal"  

5200             INITIALIZE ":INTERNAL,4,0"  

5210             CALL Vol_label(Drive$,Primary_label$)  

5220         CASE "HP_8290X"  

5230             INITIALIZE ":HP8290X,702,1"  

5240             CALL Vol_label(Drive$,Backup_label$)  

5250         CASE "HP_9122D"  

5260             INITIALIZE ":CS80,702,1"  

5270             CALL Vol_label(Drive$,Backup_label$)  

5280         CASE "HP_9133D"  

5290             INITIALIZE ":CS80,700,1"  

5300             CALL Vol_label(Drive$,Backup_label$)  

5310     END SELECT  

5320 END IF  

5330 IF Drive$="Left " THEN  

5340     SELECT Store_device$  

5350         CASE "Internal"  

5360             INITIALIZE ":INTERNAL,4,1"  

5370             CALL Vol_label(Drive$,Backup_label$)  

5380         CASE "HP_8290X"  

5390             INITIALIZE ":HP8290X,702,0"  

5400             CALL Vol_label(Drive$,Primary_label$)  

5410         CASE "HP_9122D"  

5420             INITIALIZE ":CS80,702,0"  

5430             CALL Vol_label(Drive$,Primary_label$)  

5440         CASE "HP_9133D"  

5450             INITIALIZE ":CS80,700,1"  

5460             CALL Vol_label(Drive$,Primary_label$)  

5470     END SELECT  

5480 END IF  

5490 !  

5500 CASE 64  

5510 BEEP  

5520 IF Store_device$="HP_9133D" THEN

```

```

5530      DISP USING "K";" The floppy disc is full."
5540      ELSE
5550          DISP USING "K";" The ";Drive$;" disc is full."
5560      END IF
5570      SELECT Store_device$
5580          CASE "Internal"
5590              IF Drive$="Right" THEN
5600                  Drive$="Left "
5610                  Switch_drive(Test_frame)=File(Test_frame)
5620                  MASS STORAGE IS "INTERNAL,4,1"
5630              ELSE
5640                  End_flag=1
5650              END IF
5660          CASE "HP_8290X"
5670              IF Drive$="Left " THEN
5680                  Drive$="Right"
5690                  Switch_drive(Test_frame)=File(Test_frame)
5700                  MASS STORAGE IS ":HP8290X,702,1"
5710              ELSE
5720                  End_flag=1
5730              END IF
5740          CASE "HP_9122D"
5750              IF Drive$="Left " THEN
5760                  Drive$="Right"
5770                  Switch_drive(Test_frame)=File(Test_frame)
5780                  MASS STORAGE IS ":CS80,702,1"
5790              ELSE
5800                  End_flag=1
5810              END IF
5820          CASE "HP_9133D"
5830              IF Drive$="Left " THEN
5840                  Drive$="Right"
5850                  Switch_drive(Test_frame)=File(Test_frame)
5860                  MASS STORAGE IS ":CS80,700,1"
5870              ELSE
5880                  End_flag=1
5890              END IF
5900          END SELECT
5910      !
5920      CASE ELSE
5930          BEEP
5940          DISP USING "K,3D,X,K";"Unexpected error ";ERRN;ERRM$
5950          CONTROL 2,2;0           ! Turn on SYSTEM softkeys
5960          PAUSE
5970          CONTROL 2,2;1           ! Turn on USER softkeys
5980      END SELECT
5990      OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
6000      SUBEXIT
6010  SUBEND
6020  !
6030  !
6040  SUB Free_space(Label$,INTEGER Space)
6050  !
6060  ! J. C. Gibeling ** November 06, 1982
6070  !             revised ** May 03, 1985
6080  !
6090  ! Subprogram to determine the number of unused physical records
6100  ! remaining on a data disc.
6110  !
6120  ! This subprogram uses the CSUBs "Phyread" and "Phywrite", which

```

```

6130      ! must be appended.
6140      !
6150      OPTION BASE 1
6160      COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_nam
e$,Parm_file$(*)
6170      COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Backu
p_space,Full_flag
6180      DIM A$[2]
6190      INTEGER I,J,Records,Total_records,Vol(0:9),Dir(0:7,0:15)
6200      REAL Rpf,Address,Dir_start,Dir_length,D,E
6210      !
6220      SELECT Store_device$
6230      CASE "HP_9122D"
6240          Total_records=2460
6250      CASE ELSE
6260          Total_records=1056
6270      END SELECT
6280      !
6290      Records=0
6300      Phyread(0,Vol(*))           ! Obtain info from volume label
6310      Label$=""
6320      A$=""
6330      FOR I=1 TO 3             ! Construct volume label
6340          OUTPUT A$ USING "#,W";Vol(I)
6350          Label$=Label$&A$
6360      NEXT I
6370      !
6380      Dir_start=FNReal(Vol(4),Vol(5))    ! Find directory start
6390      Dir_length=FNReal(Vol(8),Vol(9))   ! Find directory length
6400      !
6410      FOR D=0 TO Dir_length-1           ! For each sector in the directory
6420          Phyread(Dir_start+D,Dir(*))   ! Read the sector
6430          FOR I=0 TO 7                 ! For each entry in the sector
6440              IF Dir(I,5)==-1 THEN
6450                  Eod_sector=D
6460                  Eod_entry=I
6470                  GOTO 6540
6480              ELSE
6490                  Rpf=FNReal(Dir(I,8),Dir(I,9))
6500                  Records=Records+Rpf
6510                  END IF
6520              NEXT I
6530      NEXT D
6540      Space=Total_records-Records-Dir_length-2
6550      !
6560      SUBEND
6570      !
6580      !
6590      DEF FNReal(INTEGER Hi,Lo)
6600      !
6610      ! J. C. Gibeling ** November 06, 1982
6620      !             revised ** January 20, 1983
6630      !
6640      ! Function to change two integers into a single real number.
6650      !
6660      REAL Result
6670      Result=Hi*65535+Lo
6680      IF Hi>0 AND Lo<0 THEN Result=Result+65535
6690      IF Hi<0 AND Lo>0 THEN Result=Result-65535
6700      IF Hi<0 THEN Result=Result+65535

```

```

6710      RETURN Result
6720      !
6730 FNEND
6740      !
6750      !
6760 SUB Vol_label(Drive$,Label$)
6770      !
6780      ! J. C. Gibeling ** November 06, 1982
6790      ! revised ** October 28, 1985
6800      !
6810      ! Subprogram to write disc volume label (Transposed from BASIC
6820      ! utility "INITIALIZE"). This subprogram uses the CSUBs "Phyread"
6830      ! and "Phywrite", which must be appended.
6840      !
6850 DIM C$(160),Label(0:2),Temp$(6]
6860 INTEGER Vol(0:255),I,J
6870 OFF KEY
6880      !
6890 Phyread(0,Vol(*))
6900      !
6910 Input:      !
6920 Label$="          "           ! Pad with blanks
6930 OUTPUT 2 USING "#,B";255,75
6940 PRINT TABXY(1,12); "The volume label must be 1 to 6 characters, upper
 case letters and numbers "
6950 PRINT "only. The first character must be an upper case letter."
6960 INPUT "Please enter the volume label:",Temp$
6970 Label$[1,LEN(Temp$)]=Temp$
6980      !
6990      ! Check label for blanks
7000      !
7010 FOR I=1 TO 5
7020     IF Label$[I,I]="" THEN
7030       FOR J=I TO 5
7040         IF Label$[J,J]<>" " THEN GOTO Input
7050       NEXT J
7060       IF Label$[1,1]="" THEN GOTO OK
7070     END IF
7080   NEXT I
7090   !
7100   ! Check for bad first character
7110   !
7120 IF Label$[1,1]<"A" OR Label$[1,1]>"Z" THEN GOTO Input
7130   !
7140 Flag=0
7150   !
7160   ! Check for illegal characters
7170   !
7180 FOR I=2 TO 6
7190   IF Label$[I,I]="" THEN GOTO Ok
7200   IF (Label$[I,I]<"A" OR Label$[I,I]>"Z") AND (Label$[I,I]<"0" OR Label$[I,I]>"9") THEN GOTO Input
7210   NEXT I
7220   !
7230 Ok:          ! Translate to ASCII
7240   FOR I=1 TO 6 STEP 2
7250     C=(I-1) DIV 2
7260     Label(C)=NUM(Label$[I])
7270     Label(C)=SHIFT(Label(C),-8)+NUM(Label$[I+1])
7280   NEXT I

```

```

7290      !
7300      Vol(1)=Label(0)          ! Transfer values to volume
7310      Vol(2)=Label(1)          ! label array
7320      Vol(3)=Label(2)
7330      !
7340      Phywrite(0,Vol(*))     ! Write volume label on disc
7350      !
7360  SUBEND
7370      !
7380      !
7390  SUB Strain_span(INTEGER Frame)
7400      !
7410      OPTION BASE 1
7420      COM /Display1/ Time_origin(*),Strain_origin(*),Time_span(*),Strain_s
pan(*)
7430      GRAPHICS OFF
7440      OUTPUT 2 USING "#,B";255,75 ! [CLR SCR]
7450      !
7460      PRINT TABXY(1,8); "Please select the time and strain spans for the re
al-time"
7470      PRINT "graphics display. As the test progresses, these scaling value
s will be"
7480      PRINT "increased as necessary to accomodate the data."
7490      !
7500      !
7510      INPUT "Time span (hours)? ",Time_span(Frame)
7520      IF Time_span(Frame)<=0 THEN
7530          DISP "The magnitude of the time span must be greater than 0"
7540          GOTO 7510
7550      END IF
7560      !
7570      INPUT "Minimum strain (%)? [+ for tension, - for compression]",Stra
in_origin(Frame)
7580      INPUT "Maximum strain (%)? [+ for tension, - for compression]",Max
7590      Strain_span(Frame)=Max-Strain_origin(Frame)
7600      IF Strain_span(Frame)=0 THEN
7610          DISP "The magnitude of the strain span must be greater than 0"
7620          GOTO 7570
7630      END IF
7640  SUBEND
7650      !
7660      !
7670  SUB Axes(Y_origin,X_origin,Y_span,X_span,Y_label$,X_label$)
7680      !
7690      INTEGER Power
7700      DIM Image$[10]
7710      !
7720      DEG
7730      GINIT
7740      PLOTTER IS 3,"INTERNAL"
7750      CSIZE 3.6,.6
7760      GRAPHICS ON
7770      OUTPUT 2 USING "#,B";255,75
7780      VIEWPORT 20,123,25,95
7790      WINDOW X_origin,X_origin+X_span,Y_origin,Y_origin+Y_span
7800      AXES X_span/50,Y_span/50,X_origin,Y_origin,5,5
7810      AXES X_span/50,Y_span/50,X_origin+X_span,Y_origin+Y_span,5,5
7820      CLIP OFF
7830      LORG 6
7840      !

```

```

7850     Power=INT(LGT(ABS(X_span)))+1      ! Determine IMAGE for X axis
7860     SELECT Power
7870         CASE <0
7880             Image$="MZ."&VAL$(ABS(Power)+2)&"D"
7890             CASE 0
7900                 Image$="MZ.DD"
7910             CASE 1
7920                 Image$="MDD.D"
7930             CASE >1
7940                 Image$="M"&VAL$(Power)&"D"
7950     END SELECT
7960     !
7970     FOR I=X_origin TO X_origin+X_span STEP X_span/2
7980         MOVE I,Y_origin
7990         LABEL USING Image$;DROUND(I,3)
8000     NEXT I
8010     !
8020     LORG 8
8030     Power=INT(LGT(ABS(Y_span)))+1      ! Determine IMAGE for Y axis
8040     SELECT Power
8050         CASE <0
8060             Image$="MZ."&VAL$(ABS(Power)+2)&"D"
8070             CASE 0
8080                 Image$="MZ.DD"
8090             CASE 1
8100                 Image$="MDD.D"
8110             CASE >1
8120                 Image$="M"&VAL$(Power)&"D"
8130     END SELECT
8140     !
8150     FOR I=Y_origin TO Y_origin+Y_span STEP Y_span/2
8160         MOVE X_origin,I
8170         LABEL USING Image$;DROUND(I,3)
8180     NEXT I
8190     !
8200     LORG 5
8210     CSIZE 4,.6
8220     MOVE X_origin+.5*X_span,Y_origin-.1*Y_span
8230     LABEL X_label$
8240     LDIR 90
8250     MOVE X_origin-.17*X_span,Y_origin+.5*Y_span
8260     LABEL Y_label$
8270     LDIR 0
8280     CSIZE 3.6,.6
8290     PENUP
8300     CLIP X_origin,X_origin+X_span,Y_origin,Y_origin+Y_span
8310 SUBEND
8320 !
8330 !
8340 SUB Strain_plot(INTEGER Frame)
8350 !
8360     ! Subprogram to plot strain-time data.
8370 !
8380     OPTION BASE 1
8390     COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
8400     COM /Data/ Material$(*),Sample_number$(*),INTEGER Test_type(*),Point
     _counter(*),Number_points(*)
8410     COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
     gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
8420     COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_na

```

```

me$,Parm_file$(*)
8430      COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Back
up_space,Full_flag
8440      COM /Display1/ Time_origin(*),Strain_origin(*),Time_span(*),Strain_s
pan(*)
8450      COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*)
,Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
8460      COM /Parms/ Comment$(*),Computer$,INTEGER Test_frame
8470      DIM Var1(500),Var2(500),Var3(500),Var4(500)
8480      INTEGER I,J
8490      !
8500      Plot_flag=0
8510      CALL Axes(Strain_origin(Frame),Time_origin(Frame),Strain_span(Frame)
,Time_span(Frame),"Strain (%)","Time (hrs)")
8520      !
8530      MOVE Time_origin(Frame)+.20*Time_span(Frame),Strain_origin(Frame)+.2
0*Strain_span(Frame)
8540      LABEL USING "K";Material$(Frame)
8550      LABEL USING "K";Sample_number$(Frame)
8560      LABEL USING "K,D";"Frame ",Frame
8570      PENUP
8580      !
8590      MOVE 0,0
8600      FOR I=1 TO File(Frame)-1
8610      ASSIGN @Data_path TO Sample_number$(Frame)&"."&VAL$(I)
8620      ENTER @Data_path;Var1(*),Var2(*),Var3(*),Var4(*)
8630      ASSIGN @Data_path TO *
8640      FOR J=1 TO 500
8650      ! Stress=Load_cal(Frame)*Var1(J)/Area(Frame)
8660      Tyme=(Var4(J)-Start(Frame))/3600
8670      IF Test_type(Frame)=1 THEN
8680      Strain=100*Elong_cal(Frame)*(Var2(J)-Elong_0(Frame))/Gage_leng
th(Frame)
8690      ELSE
8700      Strain=100*Elong_cal(Frame)*(Var2(J)-Elong_0(Frame))
8710      END IF
8720      !
8730      IF Strain>Strain_span(Frame)+Strain_origin(Frame) THEN
8740      Strain_span(Frame)=2*Strain_span(Frame)
8750      GOTO 8500
8760      END IF
8770      IF Tyme>Time_span(Frame) THEN
8780      Time_span(Frame)=2*Time_span(Frame)
8790      GOTO 8500
8800      END IF
8810      !
8820      DRAW Tyme,Strain
8830      NEXT J
8840      NEXT I
8850      !
8860      IF Point_counter(Frame)=1 THEN 9080
8870      FOR I=1 TO Point_counter(Frame)-1
8880      ! Stress=Load_cal(Frame)*Load(Frame,I)/Area(Frame)
8890      Tyme=(Time(Frame,I)-Start(Frame))/3600
8900      IF Test_type(Frame)=1 THEN
8910      Strain=100*Elong_cal(Frame)*(Elong(Frame,I)-Elong_0(Frame))/Gage
_length(Frame)
8920      ELSE
8930      Strain=100*Elong_cal(Frame)*(Elong(Frame,I)-Elong_0(Frame))
8940      END IF

```

```

8950 !
8960   IF Strain>Strain_span(Frame)+Strain_origin(Frame) THEN
8970     Strain_span(Frame)=2*Strain_span(Frame)
8980     GOTO 8500
8990   END IF
9000   IF Tyme>Time_span(Frame) THEN
9010     Time_span(Frame)=2*Time_span(Frame)
9020     GOTO 8500
9030   END IF
9040 !
9050   DRAW Tyme,Strain
9060 NEXT I
9070 !
9080 PENUP
9090 Plot_flag=1
9100 !
9110 SUBEND
9120 !
9130 !
9140 SUB Test_setup
9150 !
9160 ! Subprogram to set-up a new creep test.
9170 !
9180 OPTION BASE 1
9190 COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
9200 COM /Data/ Material$(*),Sample_number$(*),INTEGER Test_type(*),Point
counter(*),Number_points(*)
9220 COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*),
Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
9221 COM /Parms/ Comment$(*),Computer$,INTEGER Test_frame
9230 COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_na
me$,Parm_file$(*)
9240 COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Back
up_space,Full_flag
9250 COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
9260 COM /Paths/ @Data_acq_unit
9270 INTEGER Frame,I
9280 DIM Response$[20]
9290 !
9300 !
9310 GRAPHICS OFF
9320 Plot_flag=0
9330 OFF KEY
9340 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
9350 !
9360 PRINT TABXY(1,16); "Which test frame number do you wish to start (1-5
)?
"
9370 PRINT
9380 PRINT "Note: Use 1 for a single frame system."
9390 CALL Select_frame(Frame)
9400 Test_frame=Frame
9410 !
9420 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
9430 !
9440 IF Frame=0 THEN
9450   Plot_flag=1
9460   GRAPHICS ON
9470   SUBEXIT
9480 END IF

```

```

9490 !
9500 IF Test_running(Frame) THEN
9510   DISP USING "K,D,K";"A test is still running on test frame ",Frame,
9520   " !"
9530   Plot_flag=1
9540   GRAPHICS ON
9550   SUBEXIT
9560 END IF
9570 !
9580 IF NOT Data_transfer(Frame) THEN
9590   PRINT TABXY(1,16);" "
9600   PRINT USING "K,D,K";"Data from the previous test on frame ",Frame,
9610 " must be transferred"
9620   PRINT "from the system disc to a data disc before a new test can b
e started!"
9630   Plot_flag=1
9640   GRAPHICS ON
9650   SUBEXIT
9660 END IF
9670 !
9680 OUTPUT 2 USING "#,B";255,75 ! [CLR SCR]
9690 !
9700 Number_points(Frame)=0
9710 Point_counter(Frame)=1      ! First point number is 1
9720 File(Frame)=1              ! First file is Sample_number$&"_1"
9730 Switch_drive(Frame)=0
9740 Data_transfer(Frame)=0      ! Clear data transferred flag
9750 Max_load(Frame)=0
9760 Max_elong(Frame)=0
9770 High_spd_flag(Frame)=0      ! Clear initial high speed data flag
9780 Control_flag(Frame)=0      ! Clear test control flag
9790 Total_temp(Frame)=0
9800 !
9810 !
9820 !SELECT Computer$[1,4]
9830   !CASE "9817","S300","PC30"
9840     !ON KEY 1 LABEL "Displac.",2 GOTO Stroke
9850     !ON KEY 2 LABEL " Strain",2 GOTO Strain
9860   !CASE ELSE
9870     !ON KEY 0 LABEL "Displacement",2 GOTO Stroke
9880     !ON KEY 1 LABEL "Strain",2 GOTO Strain
9890 !END SELECT
9900 !
9910 Idle1: !
9920 !GOTO Idle1
9930 !
9940 Stroke:                      ! Displacement measured test
9950   Test_type(Frame)=1
9960   !GOTO Input
9970   !
9980 Strain:                      ! Strain measured test
9990   !Test_type(Frame)=2
10000  !GOTO Input
10010  !
10020 Input:!
10030  !
10040  !
10050  ! INPUT TEST PARAMETERS FOR THIS SAMPLE

```

```

10060      !
10070      OFF KEY
10080      INPUT "Material?",Material$(Frame)
10090      Response$=""
10100      CALL Sample_number(Response$)
10110      Sample_number$(Frame)=Response$[1,7]
10120      !
10130      OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
10140      !
10150      INPUT "Initial cross-sectional area (m^2)?",Area(Frame)
10160      IF Area(Frame)<=0 THEN
10170          DISP "The cross-sectional area of the sample must be greater than
0 !"
10180          GOTO 10150
10190      END IF
10191      Area(Frame)=Area(Frame)/.00645      ! Convert to sq. inches
10200      !
10210      INPUT "Gage length (m)?",Gage_length(Frame)
10211      Gage_length(Frame)=Gage_length(Frame)/.0254      ! Convert to inches
10220      IF Gage_length(Frame)<=0 THEN
10230          DISP "The gage length of the sample must be greater than 0 !"
10240          GOTO 10210
10250      END IF
10260      !
10270      LINPUT "Enter a test description comment for this frame. (150 charac
ters maximum)",Comment$(Frame)
10280      !
10290      INPUT "Nominal test load (N)?",Nom_load(Frame)
10300      IF Nom_load(Frame)<=0 THEN
10310          DISP "The nominal load on the sample must be greater than 0 !"
10320          GOTO 10290
10330      END IF
10331      Nom_load(Frame)=Nom_load(Frame)/4.448      ! Convert to lbf.
10340      !
10350      !
10360      CALL Calibration(Load_cal(Frame),Elong_cal(Frame),Tc_cal(Frame),Test
_type(Frame))
10370      !
10380      !
10390      ! CHECK DATA ACQUISITION UNIT INPUT CONNECTIONS
10400      !
10410      !OUTPUT 2 USING "#,B";255,75
10420      !PRINT TABXY(1,1);"Please check the following input connections to t
he 3497A Data Acquisition"
10430      !PRINT "and Control Unit:"
10440      !PRINT
10450      !PRINT USING "15X,31A,DD";"Test status signal to Channel ",4*(Frame
-1)
10460      !PRINT USING "15X,31A,DD";"Temperature signal to Channel ",1+4*(Fra
me-1)
10470      !IF Test_type(Frame)=1 THEN
10480          !PRINT USING "15X,31A,DD";"Displacement signal to Channel ",2+4*(F
rame-1)
10490      !ELSE
10500          !PRINT USING "15X,31A,DD";"Strain signal to Channel           ",2+4*(F
rame-1)
10510      !END IF
10520          !PRINT USING "15X,31A,DD";"Load signal to Channel           ",3+4*(Fra
me-1)
10530      !

```

```

10540      !PRINT USING "/,_K";"On the back of the 3497A check the following con-
nections:"  

10550      !PRINT USING "1/_,,15X,_K";"VM COMPLETE to EXT INCR"  

10560      !PRINT USING "15X,_K";"EXT TRIG to TIMER"  

10570      !  

10580      !DISP USING "K";"Press the CONTINUE softkey when you are ready."  

10590      !  

10600      !SELECT Computer$[1,4]  

10610          !CASE "9817","S300","PC30"  

10620              !ON KEY 1 LABEL "CONTINUE",2 GOTO Continue  

10630          !CASE ELSE  

10640              !ON KEY 0 LABEL "CONTINUE",2 GOTO Continue  

10650      !END SELECT  

10660      !  

10670 Idle2:!  

10680      !GOTO Idle2  

10690      !  

10700 Continue:!  

10710          !OFF KEY  

10720      OUTPUT 2 USING "#,B";255,75           ! [CLR SCR]  

10730      !  

10740      ON ERROR GOTO Error  

10750      Parm_file$(Frame)=Sample_number$(Frame)&".END"  

10760      CREATE BDAT Parm_file$(Frame),5      ! Create parameter file  

10770      !Primary_space=Primary_space-5  

10780      OFF ERROR  

10790      !  

10800      !  

10810      ! SET UP REAL-TIME GRAPHICS DISPLAYS  

10820      !  

10830      Time_origin(Frame)=0  

10840      Strain_origin(Frame)=0  

10850      CALL Strain_span(Frame)  

10860      OUTPUT 2 USING "#,B";255,75           ! Clear screen  

10870      !  

10880      !  

10890      ! SET UP HIGH SPEED DATA COLLECTION FOR TEST START  

10900      !  

10910      GRAPHICS OFF  

10920      Fast_points(Frame)=150                ! High speed for 5 minutes  

10930      High_spd_flag(Frame)=1                 ! Set flag for initial high speed  

data  

10940      !  

10950      !  

10960      CALL Interval(Frame)                 ! Select interval between points  

10970      !  

10980      !  

10990      IF High_spd_flag(Frame)=1 THEN  

11000          PRINT TABXY(1,5)  

11010          PRINT "At the beginning of the test, data will be collected at 2 s  

econd intervals"  

11020          PRINT USING "K,DDD,K";"for 5 minutes to monitor the primary region  

. Data will then be collected at the"  

11031          PRINT "selected interval."  

11040      END IF  

11050      PRINT TABXY(1,11);"You may change the data collection interval at an  

y time during "  

11060      PRINT "the test by pressing the appropriate softkey. In addition, th  

e test may be"  

11070      PRINT "ended by pressing the 'End of Test' key."

```

```

11080      !
11085      PRINT USING "2/,K";"Please be certain that the status signal is on,
then press the START softkey"
11090      PRINT USING "K";"when you are ready to begin collecting new data."
11100      !
11110      SELECT Computer$[1,4]
11120          CASE "9817","S300","PC30"
11130              ON KEY 1 LABEL "START",2 GOTO Start
11140          CASE ELSE
11150              ON KEY 0 LABEL "START",2 GOTO Start
11160      END SELECT
11170      !
11180 Idle3:!
11190      GOTO Idle3
11200      !
11210 Start:!
11220      OFF KEY
11221      Test_running(Frame)=1
11230      OUTPUT @Data_acq_unit;"VT3"           ! Take initial reading
11240      Start(Frame)=TIMEDATE
11250      OUTPUT @Data_acq_unit USING "K,6Z";"VT2TI",Data_interval
11270      Display_test=Frame                   ! Change display to new test
11280      CALL Strain_plot(Display_test)
11290      ! OUTPUT 2 USING "#,B";255,75        ! [CLR SCR]
11300      Plot_flag=1
11310      GRAPHICS ON
11320      SUBEXIT
11330      !
11340 Error: !
11350      IF ERRN=54 THEN
11360          PRINT TABXY(1,16);"A test with the same sample number already exists on the disc."
11370          PRINT "Please choose a new sample number."
11380          CALL Sample_number(Response$)
11390          Sample_number$(Frame)=Response$[1,7]
11400          GOTO 10750
11410      ELSE
11420          CALL Disc_error
11430          GOTO 10760
11440      END IF
11450      !
11460 SUBEND
11470 !
11480 !
11490 SUB Transfer(OPTIONAL INTEGER Load_frame)
11500      !
11510      ! Subprogram to transfer data from the system disc to a data disc
11520      ! for permanent storage.
11530      !
11540      OPTION BASE 1
11550      COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
11560      COM /Data/ Material$(*),Sample_number$(*),INTEGER Test_type(*),Point
      _counter(*),Number_points(*)
11570      COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_na
      me$,Parm_file$(*)
11580      COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Back
      up_space,Full_flag
11590      COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*)
      ,Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
11600      COM /Parms/ Comment$(*),Computer$,INTEGER Test_frame

```

```
11610      COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
11620      INTEGER I,Frame
11630      DIM Old_file_name$[10],Old_sample_num$[10]
11640      !
11650      !
11660      !OFF KEY
11670      !GRAPHICS OFF
11680      !IF NPAR=0 THEN
11690          !OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
11700          !
11710          !DISP "For which load frame do you wish to transfer the data (1-5)
?
11720          !CALL Select_frame(Frame)
11730      !ELSE
11740          !Frame=Load_frame
11750      !END IF
11761      Frame=Load_frame
11762      Test_frame=Frame
11770      !
11780      OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
11790      !
11800      IF Frame=0 THEN
11810          GRAPHICS ON
11820          SUBEXIT
11830      END IF
11840      !
11850      IF Test_running(Frame) THEN
11860          DISP USING "K,D,A";"A test is still running on load frame ",Frame,
"!"
11870          GRAPHICS ON
11880          SUBEXIT
11890      END IF
11900      !
11910      !IF Data_transfer(Frame) THEN
11920          !DISP USING "K,D,K";"The data from load frame ",Frame," have alrea
dy been transferred!"
11930          !GRAPHICS ON
11940          !SUBEXIT
11950      END IF
11960      !
11970      !
11980      ! SELECT MASS STORAGE DEVICE
11990      !
12000      !SELECT Computer$[1,4]
12010      !CASE "9817","S300"
12020          !ON KEY 1 LABEL "Internal",2 GOTO Internal
12030          !ON KEY 2 LABEL "HP 8290X",2 GOTO Hp_8290x
12040          !ON KEY 3 LABEL "HP 9122D",2 GOTO Hp_9122
12050          !ON KEY 4 LABEL "HP 9133 floppy",2 GOTO Hp_9133
12060      !CASE ELSE
12070          !ON KEY 0 LABEL "Internal",2 GOTO Internal
12080          !ON KEY 1 LABEL "HP 8290X",2 GOTO Hp_8290x
12090          !ON KEY 2 LABEL "HP 9122D",2 GOTO Hp_9122
12100          !ON KEY 3 LABEL "HP 9133 floppy",2 GOTO Hp_9133
12110      END SELECT
12120      !
12130      !DISP USING "K,D";"Please select a mass storage device to which to t
ransfer the data from frame ",Frame
12140      !
```

```
12150 !Idle0: !
12160     !GOTO Idle0
12170     !
12180 !Internal: !
12190     !Store_device$="Internal"
12200     !Msu$":INTERNAL,4,0"
12210     !DISP
12220     !GOTO Set_up
12230     !
12240 !Hp_8290x:!
12250     !Store_device$="HP_8290X"
12260     !Msu$":HP8290X,702,0"
12270     !DISP
12280     !GOTO Set_up
12290     !
12300 !Hp_9122: !
12310     !Store_device$="HP_9122D"
12320     !Msu$":CS80,702,0"
12330     !DISP
12340     !GOTO Set_up
12350     !
12360 !Hp_9133: !
12370     !Store_device$="HP_9133D"
12380     !Msu$":CS80,700,1"
12390     !DISP
12400     !GOTO Set_up
12401     !
12403 !DFS Hard Drive: !
12404     !Store_device$="DFS"
12405     !Msu$"\BLP\DATA:DOS,C"
12406     !DISP
12407     !GOTO Set_up
12410     !
12420     !
12430 Set_up:!
12440     !
12450     !OFF KEY
12460     !ASSIGN @Enter_path TO Parm_file$(Frame)
12470     !ENTER @Enter_path;Material$(Frame),Sample_number$(Frame),Test_type(
Frame),Area(Frame),Gage_length(Frame),Elong_cal(Frame),Load_cal(Frame)
12480     !ENTER @Enter_path;Tc_cal(Frame),Number_points(Frame),Primary_labels$,
Backup_labels$,Parm_file$(Frame)
12490     !ENTER @Enter_path;Elong_0(Frame),Start(Frame),Finish(Frame),File(Fr
ame),Switch_drive(Frame),Max_load(Frame),Max_elong(Frame)
12500     !ENTER @Enter_path;Comment$(Frame)
12510     !ASSIGN @Enter_path TO *
12520     !
12530     !CALL Disc_setup2(63)           ! Prepare discs for data storage
12540     !
12550     !MASS STORAGE IS "\BLP\DATA:DOS,C"
12560     !PURGE Parm_file$(Frame)
12570     !
12580     !Old_sample_num$=Sample_number$(Frame)
12590     !ON ERROR GOTO Error
12600     !CREATE BDAT Parm_file$(Frame)&Msu$,5
12610     !OFF ERROR
12620     !ON ERROR CALL Disc_error
12630     !ASSIGN @Output_path TO Parm_file$(Frame)&Msu$
12640     !OUTPUT @Output_path;Material$(Frame),Sample_number$(Frame),Test_typ
e(Frame),Area(Frame),Gage_length(Frame),Elong_cal(Frame),Load_cal(Frame)
```

```

12650      !OUTPUT @Output_path;Tc_cal(Frame),Number_points(Frame),Primary_labe
1$,Backup_label$,Parm_file$(Frame)
12660      !OUTPUT @Output_path;Elong_0(Frame),Start(Frame),Finish(Frame),File(
Frame),Switch_drive(Frame),Max_load(Frame),Max_elong(Frame)
12670      !OUTPUT @Output_path;Comment$(Frame)
12680      !ASSIGN @Output_path TO *
12690      !OFF ERROR
12700      !Primary_space=Primary_space-5
12710      !
12720      !FOR I=1 TO File(Frame)
12730          !Old_file_name$=Old_sample_num$&"_"&VAL$(I)
12740          !File_name$=Sample_number$(Frame)&"_"&VAL$(I)
12750          !
12760          !SELECT Store_device$
12770          !CASE "Internal"
12780              !IF Drive$="Right" AND Primary_space<63 THEN
12790                  !Drive$="Left "
12800                  !Msu$=:INTERNAL,4,1"
12810                  !Switch_drive(Frame)=I
12820          !END IF
12830          !CASE "HP_8290X"
12840              !IF Drive$="Left " AND Primary_space<63 THEN
12850                  !Drive$="Right"
12860                  !Msu$=:HP8290X,702,1"
12870                  !Switch_drive(Frame)=I
12880          !END IF
12890          !CASE "HP_9122D"
12900              !IF Drive$="Left " AND Primary_space<63 THEN
12910                  !Drive$="Right"
12920                  !Msu$=:CS80,702,1"
12930                  !Switch_drive(Frame)=I
12940          !END IF
12950          !CASE "HP_9133D"
12960              !IF Drive$="Left " AND Primary_space<63 THEN
12970                  !Drive$="Right"
12980                  !DISP "Please insert a new data disc in the drive, then pres
s ",CHR$(129),"CONTINUE",CHR$(128)
12990                  !CONTROL 2,2;0           ! Turn on SYSTEM softkeys
13000                  !PAUSE
13010                  !CONTROL 2,2;1           ! Turn on USER softkeys
13020                  !CALL Free_space(Backup_label$,Backup_space)
13030                  !IF Backup_space<63 THEN
13040                      !PRINT TABXY(1,4);"This disc does not have enough room for
more"
13050                      !PRINT USING "K";"data. Please insert a new disc and press
",CHR$(129),"CONTINUE",CHR$(128)," ."
13060                  !CONTROL 2,2;0           ! Turn on SYSTEM softkeys
13070                  !PAUSE
13080                  !CONTROL 2,2;1           ! Turn on USER softkeys
13090                  !OUTPUT 2 USING "#,B";255,75   ! [CLR SCR]
13100                  !GOTO 13020
13110          !END IF
13120          !Switch_drive(Frame)=I
13130      !END IF
13140  !END SELECT
13150  !
13160  !COPY Old_file_name$ TO File_name$&Msu$
13170  !PURGE Old_file_name$
13180  !
13190  !SELECT Store_device$

```

```

13200      !CASE "Internal"
13210          !IF Drive$="Right" THEN
13220              !Primary_space=Primary_space-63
13230          !ELSE
13240              !Backup_space=Backup_space-63
13250          !END IF
13260      !CASE "HP_8290X", "HP_9122D", "HP_9133D"
13270          !IF Drive$="Left" THEN
13280              !Primary_space=Primary_space-63
13290          !ELSE
13300              !Backup_space=Backup_space-63
13310          !END IF
13320      !END SELECT
13330      !
13340      !NEXT I
13350      !
13360      !
13370      Data_transfer(Frame)=1           ! Set data transferred flag
13380      !
13390 Print:!
13400      !
13410      !
13420      ! PRINT TEST PARAMETER INFORMATION
13430      !
13440      OFF TIMEOUT
13450      OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
13460      PRINTER IS 1
13470      PRINT TABXY(25,2);"CREEP TEST RESULTS"
13480      PRINT USING "1/"
13490      PRINT USING "K";"Sample number: ",Sample_number$(Frame),
    Material: ,Material$(Frame)
13500      PRINT USING "K";"Test start: ",DATE$(Start(Frame)),," ",TIME$(Start(
Frame))
13510      PRINT USING "K";"Test end: ",DATE$(Finish(Frame)),," ",TIME$(Finish(
Frame))
13520      PRINT USING "K";"Number of data points: ",Number_points(Frame)
13530      !
13540      IF Switch_drive(Frame)=0 THEN
13550          PRINT USING "K";"Data files ",Sample_number$(Frame)&".1"," to ",Sa
mple_number$(Frame)&".",&VAL$(File(Frame))," on hard drive."
13560      ELSE
13570          PRINT USING "K";"Data files ",Sample_number$(Frame)&".1"," to ",Sa
mple_number$(Frame)&".",&VAL$(Switch_drive(Frame)-1)," on hard drive."
13580          I=Switch_drive(Frame)
13590          PRINT USING "K";"Data files ",Sample_number$(Frame)&".",&VAL$(I)," to ",Sa
mple_number$(Frame)&".",&VAL$(File(Frame))," on hard drive."
13600      END IF
13610      PRINT USING "K";"Test parameters on hard drive in file ",Parm_file$(
Frame)
13620      !
13630      IMAGE "Average Temperature: ",4D.D," C"
13640      PRINT USING 13630;Total_temp(Frame)/Number_points(Frame)
13650      IMAGE "Total Elapsed Time: ",4D.DD," hours"
13660      PRINT USING 13650;(Finish(Frame)-Start(Frame))/3600
13670      !
13680      IMAGE "Nominal strain to failure= ",3D.2D," %"
13690      IF Test_type(Frame)=1 THEN
13700          PRINT USING 13680;100*Elong_cal(Frame)*(Max_elong(Frame)-Elong_0(F
rame))/Gage_length(Frame)
13710      ELSE

```

```

13720      PRINT USING 13680;100*Elong_cal(Frame)*(Max_elong(Frame)-Elong_0(F
rame))
13730      END IF
13740      PRINT USING "1/,K";Comment$(Frame)
13750      ON TIMEOUT 26,.1 GOTO End      ! Skip if no printer
13760      OUTPUT 26;CHR$(12)          ! Printer form feed
13770      OFF TIMEOUT
13780      DUMP ALPHA
13781      OUTPUT 26;CHR$(12)          ! Printer form feed
13790      !
13800 End:  !
13810      OFF TIMEOUT
13820      OUTPUT 2 USING "#,B";255,75    ! [CLR SCR]
13830      GRAPHICS ON
13840      SUBEXIT
13850      !
13860 Error:!
13870      IF ERRN=54 THEN
13880      PRINT TABXY(1,16); "A test with the same sample number already exists on the disc."
13890      PRINT "Please choose a new sample number."
13900      CALL Sample_number(Response$)
13910      Sample_number$(Frame)=Response$[1,7]
13920      Parm_file$(Frame)=Sample_number$(Frame)&".END"
13930      ELSE
13940      CALL Disc_error
13950      END IF
13960      GOTO 12600
13970      !
13980 SUBEND
13990 !
14000 !
14010 SUB Interval(OPTIONAL INTEGER Load_frame)
14020 !
14030      ! Subprogram to select the time interval between data points for
14040      ! a particular load frame.
14050 !
14060      OPTION BASE 1
14070      COM /Timing/ INTEGER Interval(*),Data_interval,Interval_flag
14080      COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
14090      COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*)
,Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
14100      COM /Parms/ Comment$(*),Computer$,INTEGER Test_frame
14110      INTEGER Frame
14120      !
14130      OUTPUT 2 USING "#,B";255,75
14140      Interval_flag=0           ! Clear interval selected flag
14150      !
14160      IF NPAR=0 THEN
14170      !
14180      GRAPHICS OFF
14190      DISP "For which load frame do you wish to select the data interval
(1-5)?"
14200      CALL Select_frame(Frame)
14210      !
14220      OUTPUT 2 USING "#,B";255,75    ! [CLR SCR]
14230      !
14240      IF Frame=0 THEN
14250      GRAPHICS ON

```

```

14260      SUBEXIT
14270      END IF
14280      !
14290      IF NOT Test_running(Frame) THEN
14300          DISP "A test is not running on the selected load frame!"
14310          GRAPHICS ON
14320          SUBEXIT
14330      END IF
14340      !
14350      ELSE
14360          Frame=Load_frame
14370      END IF
14380      !
14390      OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
14400      !
14410      !
14420      ! LABEL DATA INTERVAL SELECTIONS ON SOFTKEYS
14430      !
14440      DISP "Please select the desired time interval between data points."
14450      !
14460      GOSUB Slow_keys
14470 Idle0: !
14480      IF NOT Interval_flag THEN GOTO Idle0
14490      DISP
14500      Data_interval=1800
14510      FOR I=1 TO 5
14520          IF High_spd_flag(I) THEN Data_interval=2
14530          IF Interval(I)>0 AND Interval(I)<Data_interval THEN Data_interval=
Interval(I)
14540      NEXT I
14550      IF NPAR=0 THEN GRAPHICS ON
14560      SUBEXIT
14570      !
14580 Slow_keys:!
14590      SELECT Computer$[1,4]
14600          CASE "9817","S300","PC30"
14610              OFF KEY
14620                  ON KEY 1 LABEL " Other times",3 GOSUB Fast_keys
14630                  ON KEY 2 LABEL "2 sec",3 GOSUB Sec_2
14640                  ON KEY 3 LABEL "5 sec",3 GOSUB Sec_5
14650                  ON KEY 4 LABEL "10 sec",3 GOSUB Sec_10
14660                  ON KEY 5 LABEL "20 sec",3 GOSUB Sec_20
14670                  ON KEY 6 LABEL "40 sec",3 GOSUB Sec_40
14680                  ON KEY 7 LABEL "60 sec",3 GOSUB Sec_60
14690                  ON KEY 8 LABEL "2 min",3 GOSUB Min_2
14700          CASE ELSE
14710              OFF KEY
14720                  ON KEY 0 LABEL "Other times",3 GOSUB Fast_keys
14730                  ON KEY 1 LABEL "2 sec",3 GOSUB Sec_2
14740                  ON KEY 2 LABEL "5 sec",3 GOSUB Sec_5
14750                  ON KEY 3 LABEL "10 sec",3 GOSUB Sec_10
14760                  ON KEY 4 LABEL "20 sec",3 GOSUB Sec_20
14770                  ON KEY 5 LABEL "40 sec",3 GOSUB Sec_40
14780                  ON KEY 6 LABEL "60 sec",3 GOSUB Sec_60
14790                  ON KEY 7 LABEL "2 min",3 GOSUB Min_2
14800          END SELECT
14810          RETURN
14820          !
14830 Fast_keys: !
14840      SELECT Computer$[1,4]

```

```

14850      CASE "9817","S300","PC30"
14860          OFF KEY
14870              ON KEY 1 LABEL " Other times",3 GOSUB Slow_keys
14880              ON KEY 2 LABEL "4 min",3 GOSUB Min_4
14890              ON KEY 3 LABEL "8 min",3 GOSUB Min_8
14900              ON KEY 4 LABEL "15 min",3 GOSUB Min_15
14910              ON KEY 5 LABEL "30 min",3 GOSUB Min_30
14920              ON KEY 6 LABEL "60 min",3 GOSUB Min_60
14930      CASE ELSE
14940          OFF KEY
14950              ON KEY 0 LABEL "Other times",3 GOSUB Slow_keys
14960              ON KEY 1 LABEL "4 min",3 GOSUB Min_4
14970              ON KEY 2 LABEL "8 min",3 GOSUB Min_8
14980              ON KEY 3 LABEL "15 min",3 GOSUB Min_15
14990              ON KEY 4 LABEL "30 min",3 GOSUB Min_30
15000              ON KEY 5 LABEL "60 min",3 GOSUB Min_60
15010      END SELECT
15020      RETURN
15030      !
15040 Sec_2:!
15050      Interval(Frame)=2
15060      Interval_flag=1
15070      RETURN
15080 Sec_5:!
15090      Interval(Frame)=5
15100      Interval_flag=1
15110      RETURN
15120 Sec_10:!
15130      Interval(Frame)=10
15140      Interval_flag=1
15150      RETURN
15160 Sec_20:!
15170      Interval(Frame)=20
15180      Interval_flag=1
15190      RETURN
15200 Sec_40:!
15210      Interval(Frame)=40
15220      Interval_flag=1
15230      RETURN
15240 Sec_60:!
15250      Interval(Frame)=60
15260      Interval_flag=1
15270      RETURN
15280 Min_2:!
15290      Interval(Frame)=120
15300      Interval_flag=1
15310      RETURN
15320 Min_4:!
15330      Interval(Frame)=240
15340      Interval_flag=1
15350      RETURN
15360 Min_8:!
15370      Interval(Frame)=480
15380      Interval_flag=1
15390      RETURN
15400 Min_15:!
15410      Interval(Frame)=900
15420      Interval_flag=1
15430      RETURN
15440 Min_30:!

```

```

15450      Interval(Frame)=1800
15460      Interval_flag=1
15470      RETURN
15480 Min_60!:!
15490      Interval(Frame)=3600
15500      Interval_flag=1
15510      RETURN
15520      !
15530      !
15540 SUBEND
15550 !
15560 !
15570 SUB Select_plot
15580 !
15590 ! Subprogram to select the test for which to display data.
15600 !
15610 OPTION BASE 1
15620 COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
15630 COM /Data/ Material$(*),Sample_number$(*),INTEGER Test_type(*),Point
_counter(*),Number_points(*)
15640 COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
15650 INTEGER Frame
15660 !
15670 GRAPHICS OFF
15680 Plot_flag=0
15690 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
15700 DISP "For which load frame do you wish to display the data (1-5)?"
15710 CALL Select_frame(Frame)
15720 !
15730 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
15740 !
15750 IF Frame=0 THEN
15760   Plot_flag=1
15770   GRAPHICS ON
15780   SUBEXIT
15790 END IF
15800 !
15810 IF NOT Test_running(Frame) THEN
15820   DISP "A test is not running on the selected load frame!"
15830   Plot_flag=1
15840   GRAPHICS ON
15850   SUBEXIT
15860 END IF
15870 !
15880 Display_test=Frame
15890 DISP
15900 GCLEAR
15910 GRAPHICS ON
15920 CALL Strain_plot(Display_test)
15930 Plot_flag=1
15940 !
15950 SUBEND
15960 !
15970 !
15980 SUB Data_store(INTEGER Frame)
15990 !
16000 ! Subprogram to store data and test parameters from a single
16010 ! load frame.
16020 !

```

```

16030      OPTION BASE 1
16040      COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
16050      COM /Data/ Material$(*) ,Sample_number$(*) ,INTEGER Test_type(*),Point
16060      _counter(*),Number_points(*)
16070      COM /Paths/ @Data_acq_unit
16080      COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*)
16090      ,Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
16100      COM /Parms/ Comment$(*) ,Computer$,INTEGER Test_frame
16110      COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_na
16120      me$,Parm_file$(*)
16130      COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Back
16140      up_space,Full_flag
16150      COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
16160      gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
16170      DIM Var1(500),Var2(500),Var3(500),Var4(500)
16180      INTEGER I
16190      !
16200 Data_store:!
16210      !
16220      Finish(Frame)=TIMEDATE
16230      File_name$=Sample_number$(Frame)
16240      IF LEN(File_name$)<7 THEN
16250      FOR I=LEN(File_name$)+1 TO 7
16260      File_name$[I,I]=""           ! Pad file name with blanks
16270      NEXT I
16280      END IF
16290      File_name$=File_name$&"."&VAL$(File(Frame))
16300      !
16310      CREATE BDAT File_name$,63
16320      !
16330      FOR I=1 TO 500
16340      Var1(I)=Load(Frame,I)
16350      Var2(I)=Elong(Frame,I)
16360      Var3(I)=Temp(Frame,I)
16370      Var4(I)=Time(Frame,I)
16380      NEXT I
16390      !
16400      ASSIGN @Data_path TO File_name$
16410      OUTPUT @Data_path;Var1(*),Var2(*),Var3(*),Var4(*)
16420      ASSIGN @Data_path TO *
16430      !
16440      Max_elong(Frame)=Elong(Frame,Point_counter(Frame))
16450      Number_points(Frame)=Number_points(Frame)+Point_counter(Frame)
16460      Point_counter(Frame)=0
16470      !
16480 Parm_store: !
16490      !
16500      ASSIGN @Data_path TO Parm_file$(Frame)
16510      OUTPUT @Data_path;Material$(Frame),Sample_number$(Frame),Test_type(F
16520      rame),Area(Frame),Gage_length(Frame),Elong_cal(Frame),Load_cal(Frame)
16530      OUTPUT @Data_path;Tc_cal(Frame),Number_points(Frame),Primary_label$,
16540      Backup_label$,Parm_file$(Frame)
16550      OUTPUT @Data_path;Elong_0(Frame),Start(Frame),Finish(Frame),File(Fra
16560      me),Switch_drive(Frame),Max_load(Frame),Max_elong(Frame)
16570      OUTPUT @Data_path;Comment$(Frame)
16580      ASSIGN @Data_path TO *
16590      File(Frame)=File(Frame)+1
16600      DISP
16610      !
16620 SUBEND

```

```

16550 !
16560 !
16570 SUB End_of_test(OPTIONAL INTEGER Load_frame)
16580 !
16590 ! Subprogram to end the test on a single load frame.
16600 !
16610 OPTION BASE 1
16620 COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
16630 COM /Data/ Material$(*),Sample_number$(*),INTEGER Test_type(*),Point
 _counter(*),Number_points(*)
16640 COM /Timing/ INTEGER Interval(*),Data_interval,Interval_flag
16650 COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_na
 me$,Parm_file$(*)
16660 COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Back
 up_space,Full_flag
16670 COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
 gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
16680 COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*)
 ,Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
16690 COM /Parms/ Comment$(*),Computer$,INTEGER Test_frame
16700 INTEGER I,Frame
16710 !
16720 GRAPHICS OFF
16730 OFF KEY
16740 OUTPUT 2 USING "#,B";255,75 ! [CLR SCR]
16750 !
16760 IF NPAR=0 THEN
16770 DISP "Which test frame number do you wish to end (1-5)"
16780 CALL Select_frame(Frame)
16790 ELSE
16800 Frame=Load_frame
16810 END IF
16820 !
16830 OUTPUT 2 USING "#,B";255,75 ! [CLR SCR]
16840 !
16850 IF Frame=0 THEN
16860 GRAPHICS ON
16870 SUBEXIT
16880 END IF
16890 IF NOT Test_running(Frame) THEN
16900 DISP "A test is not running on the selected test frame!"
16910 GRAPHICS ON
16920 SUBEXIT
16930 ELSE
16940 Test_running(Frame)=0
16950 Interval(Frame)=0
16960 End_flag(Frame)=0
16970 END IF
16980 !
16990 IF Frame=1 AND Control_flag(Frame) THEN CALL Hold
17000 OFF KEY
17010 FOR I=Point_counter(Frame) TO 500
17020 Load(Frame,I)=0
17030 Elong(Frame,I)=0
17040 Temp(Frame,I)=0
17050 Time(Frame,I)=0
17060 NEXT I
17070 IF Point_counter(Frame)>3 THEN
17080 Point_counter(Frame)=Point_counter(Frame)-2 ! Ignore last data po
int

```

```

17090     Max_elong(Frame)=Elong(Frame,Point_counter(Frame))
17100     CALL Data_store(Frame)
17110 ELSE
17120     Max_elong(Frame)=Elong(Frame,500)
17130 END IF
17140 !
17150 File(Frame)=File(Frame)-1
17160 Point_counter(Frame)=Number_points(Frame)-(File(Frame)-1)*500
17170 !
17180 ON TIMEOUT 26,.1 GOTO End      ! Skip graphics dump if no printer
17190 !OUTPUT 26;CHR$(12)           ! Printer form feed
17200 OFF TIMEOUT
17210 Plot_flag=0
17220 CALL Strain_plot(Frame)
17230 DUMP GRAPHICS
17240 !
17250 !
17260 End: !
17270 OFF TIMEOUT
17280 DISP
17290 GCLEAR
17300 Plot_flag=1
17310 GRAPHICS ON
17320 !
17330 IF (Test_running(1) OR Test_running(2) OR Test_running(3) OR Test_ru
nning(4) OR Test_running(5)) THEN
17340     IF Display_test=Frame THEN
17350         IF NPAR=0 THEN
17360             PRINT TABXY(1,18)
17370             PRINT USING "K,D,K";"The test on frame ",Frame," has ended."
17380         END IF
17390         FOR I=5 TO 1 STEP -1
17400             IF Test_running(I) THEN Display_test=I
17410             NEXT I
17420         END IF
17430         !
17440         CALL Strain_plot(Display_test)
17450     END IF
17460     !
17470     Data_interval=1800          ! Reset interval between data scans
17480     FOR I=1 TO 5
17490         IF Interval(I)>0 AND Interval(I)<Data_interval THEN Data_interval=
Interval(I)
17500         NEXT I
17510         Interval_flag=1
17520 SUBEND
17530 !
17540 !
17550 SUB Data
17560 !
17570     ! Subprogram to read data from HP 3497A Data Acquisition and Control
Unit
17580 !
17590 OPTION BASE 1
17600     COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
17610     COM /Data/ Material$(*),Sample_number$(*),INTEGER Test_type(*),Point
_counter(*),Number_points(*)
17620     COM /Paths/ @Data_acq_unit
17630     COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag

```

```

17640      COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*)
,Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
17650      COM /Parms/ Comment$(*),Computer$,INTEGER Test_frame
17660      COM /Timing/ INTEGER Interval(*),Data_interval,Interval_flag
17670      COM /Disc/ Store_device$,Drive$,Primary_label$,Backup_label$,File_na
me$,Parm_file$(*)
17680      COM /Disc/ Check$,INTEGER File(*),Switch_drive(*),Primary_space,Back
up_space,Full_flag
17690      DIM Data(0:19)
17700      REAL Stress,Strain,Now,Last,Tyme
17710      INTEGER I,Frame,Ser_poll
17720      !
17730      Now=TIMEDATE
17740      Ser_poll=SPOLL(@Data_acq_unit)
17750      SELECT Ser_poll
17760      CASE 0,96,161           ! Re-program 3497 after power fail
17770      WAIT 1
17780      OUTPUT @Data_acq_unit USING "K";"VT4VD5VR3VA1VF1VN20VS1AF0AL19AC
0AE1SO1SE001"
17790      OUTPUT @Data_acq_unit USING "K,6Z";"TI",100*(Data_interval DIV 6
0)+(Data_interval MOD 60)
17800      CASE 65           ! Input data
17810      OUTPUT @Data_acq_unit;"VS"
17820      ON TIMEOUT 7,.1 GOTO 18460
17830      ON ERROR GOTO 18460
17840      ENTER @Data_acq_unit;Data(*)
17850      OFF TIMEOUT
17860      OFF ERROR
17870      !
17880      FOR Frame=1 TO 5
17890      IF Point_counter(Frame)>1 THEN
17900      Last=Time(Frame,Point_counter(Frame)-1) ! Compute time of l
ast saved point for this frame
17910      ELSE
17920      Last=0
17930      END IF
17931      !
17932      !
17950      ! Save data point for this frame if elapsed time > data interv
al.
17960      !
17970      IF Test_running(Frame) AND ((Now-Last)>=Interval(Frame) OR Hig
h_spd_flag(Frame)) OR Point_counter(Frame)=1) THEN
17980      Status(Frame)=Data(4*(Frame-1))
17990      Temp(Frame,Point_counter(Frame))=FNHctemp(Data(1+4*(Frame-1))
))
18000      Elong(Frame,Point_counter(Frame))=Data(2+4*(Frame-1))
18010      !Load(Frame,Point_counter(Frame))=Data(3*(Frame-1))
18020      Load(Frame,Point_counter(Frame))=Nom_load(Frame)
18030      Time(Frame,Point_counter(Frame))=Now
18040      Total_temp(Frame)=Total_temp(Frame)+Temp(Frame,Point_counter
(Frame))
18050      !
18060      IF (File(Frame)=1 AND Point_counter(Frame)=1) THEN Elong_0(F
rame)=Elong(Frame,1)
18070      IF Point_counter(Frame)=500 THEN CALL Data_store(Frame)
18080      Point_counter(Frame)=Point_counter(Frame)+1
18090      END IF
18100      NEXT Frame
18110      !

```

```

18120      !
18130      ! PLOT DATA
18140      !
18150      IF Point_counter(Display_test)>1 THEN
18160          I=Display_test
18170          Stress=Load_cal(I)*Load(I,Point_counter(I)-1)/Area(I)
18171          IF Test_type(Display_test)=1 THEN
18172              Strain=Elong_cal(I)*(Elong(I,Point_counter(I)-1)-Elong_0(I))
18173      /Gage_length(Display_test)
18174          ELSE
18175              Strain=Elong_cal(I)*(Elong(I,Point_counter(I)-1)-Elong_0(I))
18176          END IF
18177          Tyme=Time(I,Point_counter(I)-1)-Start(I)
18178          !
18179          CONTROL 1,0;1           ! Print position row 1, column 1
18180          CONTROL 1,1;1
18181          IMAGE 9X,7A,M6D.D,16A,MD.DDE,17A,4D.D,4A
18182          IMAGE 9X,7A,M3D.3D,16A,MD.DDE,17A,4D.D,4A
18183          !
18184          IF ABS(Stress)<999.9 THEN
18185              PRINT USING 18240;"Stress=";Stress;" psi     Strain=";Strain;
18186      " Elapsed Time=";(Tyme/3600);" hrs"
18187          ELSE
18188              PRINT USING 18230;"Stress=";Stress/1000;" ksi     Strain=";St
rain;" Elapsed Time=";(Tyme/3600);" hrs"
18189          END IF
18190          !
18191          IF NOT Plot_flag THEN 18450      ! Protect subroutines from ext
ra lines
18192          IF Point_counter(I)=2 THEN
18193              MOVE Tyme/3600,100*Strain
18194          END IF
18195          !
18196          IF Point_counter(I)>2 THEN
18197              DRAW Tyme/3600,100*Strain
18198          END IF
18199          END IF
18200          !
18201          END SELECT
18202          !
18203          ON TIMEOUT 7,.1 GOTO 18460
18204          OUTPUT @Data_acq_unit;"VT2"
18205          OFF ERROR
18206          OFF TIMEOUT
18207          ENABLE INTR 7;2           ! Re-enable interrupts on HPIB
18208          !
18209      SUBEND
18210      !
18211      !
18212      SUB New_spans
18213          !
18214          ! Subprogram to select new spans for graphics display.
18215          !
18216          OPTION BASE 1
18217          COM /Data/ Load(*),Elong(*),Time(*),Temp(*),Status(*),Total_temp(*)
18218          COM /Data/ Material$(*),Sample_number$(*),INTEGER Test_type(*),Point
18219          _counter(*),Number_points(*)
18220          COM /Flags/ INTEGER Display_test,Test_running(*),Data_transfer(*),Hi
gh_spd_flag(*),Control_flag(*),Fast_points(*),End_flag(*),Plot_flag
18221          !

```

```

18620    GRAPHICS OFF
18630    Plot_flag=0
18640    OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
18650    !
18660    GCLEAR
18670    CALL Strain_span(Display_test)
18680    CALL Strain_plot(Display_test)
18690    Plot_flag=1
18700    !
18710    SUBEND
18720    !
18730    !
18740    SUB Sample_number(Input$)
18750    !
18760    ! Subprogram to enter sample number and check for legal characters.
18770    !
18780    DIM Check$[1]
18790    INPUT "Sample number (7 characters maximum, letters, numbers and _ o
nly)?",Input$
18800    IF LEN(Input$)>7 THEN
18810        OUTPUT 2 USING "#,B";255,75
18820        PRINT TABXY(1,16); "The sample number must be 7 characters or less!
"
18830        GOTO 18790
18840    END IF
18850    ! Check for bad first character
18860    IF (Input$[1,1]<"A" OR Input$[1,1]>"Z") AND (Input$[1,1]<"0" OR Inpu
t$[1,1]>"9") THEN
18870        OUTPUT 2 USING "#,B";255,75
18880        PRINT TABXY(1,16); "The first character must be an upper case lette
r or a number."
18890        GOTO 18790
18900    END IF
18910    ! Check for illegal characters
18920    FOR I=2 TO LEN(Input$)
18930        Check$=Input$[I,I]
18940        IF (Check$<"0" OR Check$>"9") AND (Check$<"A" OR Check$>"Z") AND (
Check$<>"_") AND (Check$<"a" OR Check$>"z") THEN
18950            OUTPUT 2 USING "#,B";255,75
18960            PRINT TABXY(1,16); "Only letters, numbers, and _ are allowed in s
ample number."
18970            GOTO 18790
18980        END IF
18990        NEXT I
19000        !
19010    SUBEND
19020    !
19030    !
19040    SUB Select_frame(INTEGER Frame)
19050    !
19060    ! Subprogram to select desired load frame.
19070    !
19080    OPTION BASE 1
19090    COM /Parms/ Area(*),Gage_length(*),Elong_cal(*),Load_cal(*),Start(*),
,Finish(*),Elong_0(*),Tc_cal(*),Max_load(*),Max_elong(*),Nom_load(*)
19100    COM /Parms/ Comment$(*),Computer$,INTEGER Test_frame
19110    !
19120    OFF KEY
19130    Frame=0
19140    !

```

```

19150    SELECT Computer$[1,4]
19160        CASE "9817","S300","PC30"
19170            ON KEY 1 LABEL "Frame 1",2 GOTO 19390
19180            ON KEY 2 LABEL "Frame 2",2 GOTO 19380
19190            ON KEY 3 LABEL "Frame 3",2 GOTO 19370
19200            ON KEY 4 LABEL "Frame 4",2 GOTO 19360
19210            ON KEY 5 LABEL "Frame 5",2 GOTO 19350
19220            ON KEY 8 LABEL "Exit",2 GOTO 19410
19230        CASE ELSE
19240            ON KEY 0 LABEL "Frame 1",2 GOTO 19390
19250            ON KEY 1 LABEL "Frame 2",2 GOTO 19380
19260            ON KEY 2 LABEL "Frame 3",2 GOTO 19370
19270            ON KEY 3 LABEL "Frame 4",2 GOTO 19360
19280            ON KEY 4 LABEL "Frame 5",2 GOTO 19350
19290            ON KEY 9 LABEL "Exit",2 GOTO 19410
19300        END SELECT
19310        !
19320 Idle0: !
19330     GOTO Idle0
19340     !
19350     Frame=Frame+1
19360     Frame=Frame+1
19370     Frame=Frame+1
19380     Frame=Frame+1
19390     Frame=Frame+1
19400     !
19410 SUBEND
19420 !
19430 !
19440 SUB No_acq_unit
19450 !
19460 ! Subprogram to report no response from data acquisition unit.
19470 !
19480 !
19490 COM /Paths/ @Data_acq_unit
19500 INTEGER Ser_poll
19510 !
19520 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
19530 PRINT TABXY(1,14); "Please check to see that the data acquisition uni
t is turned on,"
19540 PRINT "connected to the computer via the IEEE-488 interface, and set
to"
19550 PRINT "select code 709, then press [CONTINUE]."
19560 CONTROL 2,2;0                  ! Turn on SYSTEM softkeys
19570 PAUSE
19580 !
19590 CONTROL 2,2;1                  ! Turn on USER softkeys
19600 ON TIMEOUT 7,.1 CALL No_acq_unit
19610 Ser_poll=SPOLL(@Data_acq_unit)
19620 OFF TIMEOUT
19630 RESET 7
19640 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
19650 SUBEND
19660 !
19670 !
19680 SUB Set_clock
19690 !
19700 ! Subprogram to set real_time clock.
19710 !
19720 DIM Data$[11],Time$[8]

```

```

19730 INPUT "Please enter the date (DD MMM YYYY)",Date$
19740 INPUT "Please enter the time in 24 hour format (HH:MM:SS)",Time$
19750 ON ERROR GOTO 19810
19760 SET TIMEDATE DATE(Date$)+TIME(Time$)
19770 OFF ERROR
19780 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
19790 SUBEXIT
19800 !
19810 OUTPUT 2 USING "#,B";255,75      ! [CLR SCR]
19820 OFF ERROR
19830 BEEP
19840 PRINT TABXY(1,16); "Please use the correct format to enter the date a
nd time."
19850 GOTO 19730
19860 !
19870 SUBEND
19880 !
19890 !
19900 SUB Calibration(Load_cal,Elong_cal,Tc_cal,INTEGER Test_type)
19910 !
19920 ! Subprogram to input transducer calibration data.
19930 !
19940 !INPUT "Load cell calibration (lbs/Volt)?",Load_cal
19941 Load_cal=1
19950 IF Test_type=1 THEN
19960     INPUT "Displacement calibration (inches/Volt)?",Elong_cal
19970 ELSE
19980     INPUT "Strain calibration (1/Volt)?",Elong_cal
19990 END IF
20000 !INPUT "Thermocouple calibration (degrees c/Volt)?",Tc_cal
20001 Tc_cal=1
20002 CALL Kcoeff          ! Load type K coefficients
20010 !
20020 SUBEND
20030 !
20040 !
20050 DEF FNHctemp(Voltage)
20060 !
20070 ! Subfunction to compute temperature from thermocouple voltage.
20080 !
20090 OPTION BASE 1
20100 COM /Tcparm/ Parm(*)
20110 INTEGER I
20120 !
20130 I=7
20140 IF Voltage<Parms(4,4) THEN I=3
20150 IF Voltage<Parms(I-1,4) THEN I=I-2
20160 IF Voltage<Parms(I,4) THEN I=I-1
20170 X=Voltage-Parms(I,4)
20180 RETURN Parm(I,5)+X*(Parms(I,1)+X*(Parms(I,2)+X*Parms(I,3)))
20190 FNEND
20200 !
20210 !
20220 SUB Kcoeff
20230 !
20240 ! Subprogram to read type K coefficients.
20250 !
20251 OPTION BASE 1
20260 COM /Tcparm/ Parm(*)
20270 INTEGER J,K

```

```
20280 !
20290     FOR J=0 TO 8
20300         FOR K=1 TO 5
20310             READParms(J,K)
20320             NEXT K
20330         NEXT J
20340 !
20350     DATA 1106070,-49675E6,1.14708E15,-.00645782,-270
20360     DATA 320939,-183443E4,6.17105E12,-.00643963,-259.411
20370     DATA 121274,-115989E3,782312E5,-.006322268,-237.096
20380     DATA 58575,-11949600,203467E4,-0.00578481,-193.25
20390     DATA 33405.4,-1689600,107971E3,-.00382995,-109.209
20400     DATA 23862,117909,-2267120,.00196027,48.5292
20410     DATA 24662.4,-92361.8,2373410,.00909134,223.865
20420     DATA 25167.5,56158.1,2787480,0.0377955,911.811
20430 ! S-C COEFFICIENTS
20440     DATA 6.65829E-8,3.94486E-5,2.4541E-8,-9.13087E-11,0
20450 SUBEND
20460 !
20470 !
```