DRA/LANGLEY

OLD DOMINION UNIVERSITY RESEARCH FOUNDATION

DEPARTMENT OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

ULTRASOUND INSTRUMENTATION FOR THE 7" MACH SEVEN TUNNEL

By

David S. Mazel

and

Roland R. Mielke, Principal Investigator

Final Report
For the period August 28, 1984 to December 31, 1984

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Master Contract Agreement NAS1-17099
Task Authorization No. 40
Allan J. Zuckerwar, Technical Monitor
IRD-Acoustics and Vibration Instrumentation Section

January 1985

DEPARTMENT OF ELECTRICAL ENGINEERING
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

ULTRASOUND INSTRUMENTATION FOR THE 7" MACH SEVEN TUNNEL

By

David S. Mazel

and

Roland R. Mielke, Principal Investigator

Final Report
For the period August 28, 1984 to December 31, 1984

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Master Contract Agreement NAS1-17099
Task Authorization No. 40
Allan J. Zuckerwar, Technical Monitor
IRD-Acoustics and Vibration Instrumentation Section

Submitted by the
Old Dominion University Research Foundation
P. O. Box 6369
Norfolk, Virginia 23508

January 1985

ABSTRACT

Three areas of research are discussed. The first area is a discussion of the use of an APPLE II+ microcomputer to collect data during the operation of the 7" Mach Seven Tunnel. The second area of investigation is a method by which the contamination of liquid oxygen is monitored with sound speed techniques. The last subject area is a study of the electrical equivalent of a transducer bonded to a high pressure fill plug. The three areas are briefly explained and data gathered for each area is presented.

TABLE OF CONTENTS

LIST OF FIGURES

ULTRASOUND INSTRUMENTATION FOR THE 7" MACH SEVEN TUNNEL

By

David S. Mazel[1] and Roland R. Mielke[2]

INTRODUCTION

During the past four months, three areas of study have been investigated. The first area is the use of an APPLE II+ microcomputer to gather data for the 7" Mach Seven Wind Tunnel. Appropriate software was written and tested along with the development of necessary hardware. The next subject area is the monitoring of the contamination of liquid oxygen (LOX) by liquid nitrogen (LN_2) with sound speed techniques. The last area of investigation was modeling a transducer with electrical components. The progress in each area is briefly presented. Program listings are given in the appendices. Data from the LOX-LN_2 study is given graphically.

SUMMARY OF WORK

The developed software for the 7" Mach 7 wind tunnel allows the user to view instrument readings prior to a test. Immediately before the test begins, a pushbutton is pressed which triggers the APPLE II+ to record data from various instruments. After the test is completed, another pushbutton is pressed which stops the APPLE II+ from taking data. The recorded data are saved on a diskette. Thus, the data can be reviewed at later times, or printed on paper for permanent record. Flowcharts are shown in Figures 1 and 2. A user's guide is provided in the next section. The program listings are shown in Appendices A, B, and C.

[1] Undergraduate Research Assistant, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.

[2] Chairman/Associate Professor, Department of Electrical Engineering, Old Dominion University, Norfolk, Virginia 23508.

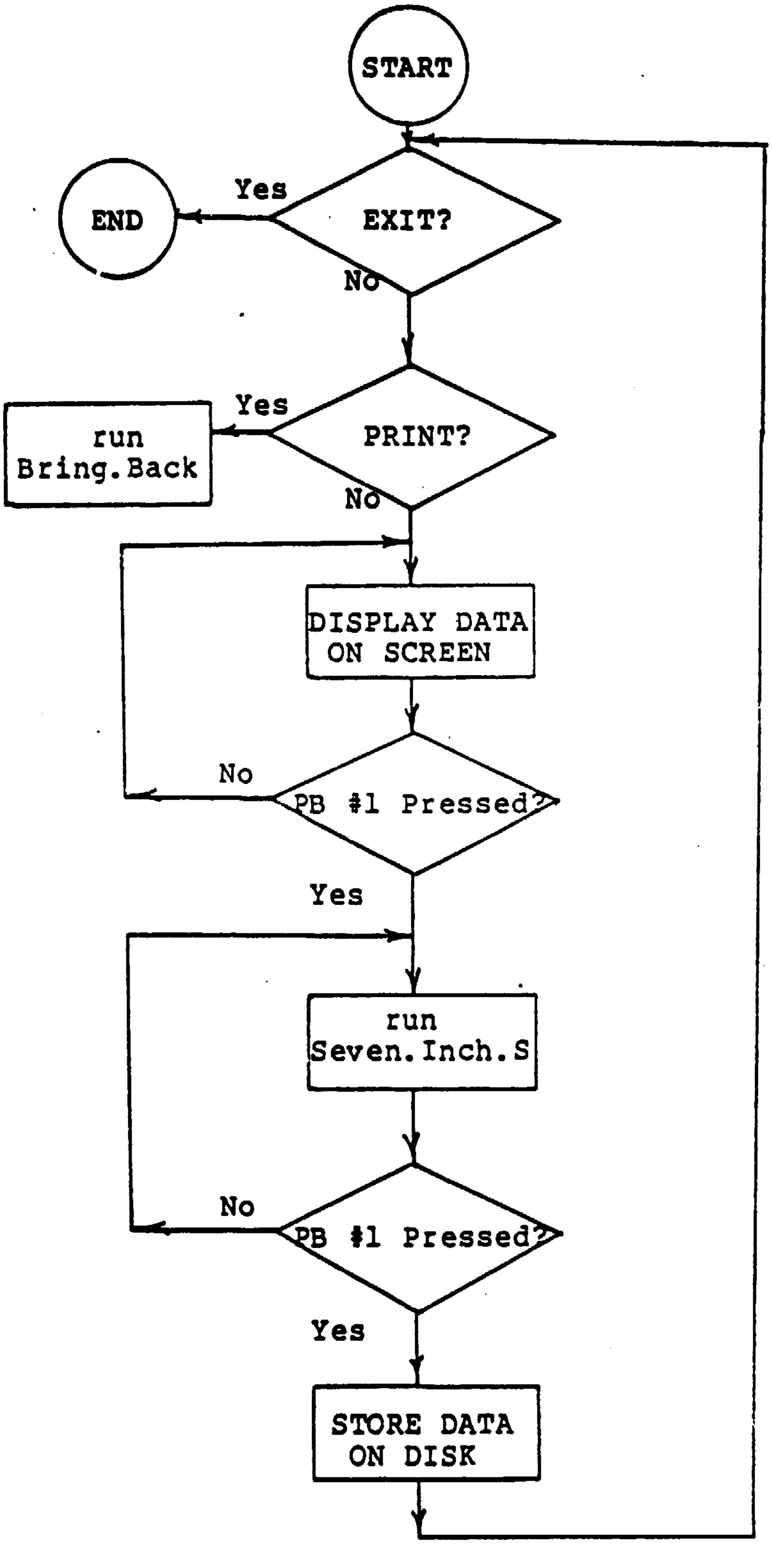


Figure 1. Software Configuration of the Data Acquisition System: Start.Fetch Program.

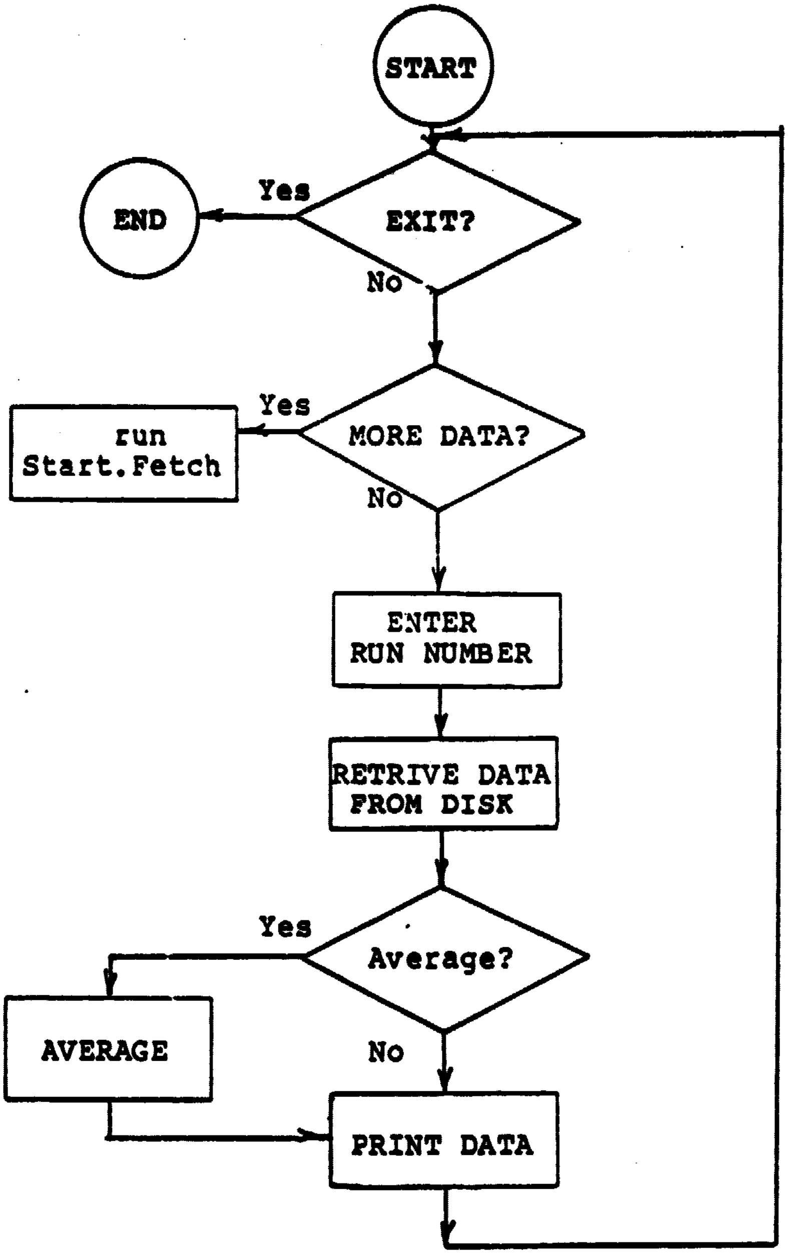


Figure 2. Software Configuration of the Data Acquisition System: Bring.Back Program.

The second area of interest, LN_2-LOX contamination monitoring, was completed. Data representing the results of numerous experiments is presented graphically in Figures 3 and 4. A paper concerning LN_2-LOX contamination monitoring is being written for a NASA reference publication.

The last area of research concerned an invention, NASA case number LAR13300-1-CU, to monitor liquid levels in high pressure tanks. A transducer was bonded to a high pressure fill plug. The fill plug is placed in direct contact witn the liquid and the transducer is excited. The plug and transducer comprise a composite resonator which emits an ultrasonic pulse into the liquid. The transducer also receives the reflected pulse. The time difference between the transmitted and reflected pulses indicate the liquid deptn. The technique is pulse-echo.

A transducer can be modelled as two inductor-capacitor (L-C) circuits. When the transducer is bonded to the plug, however, additional modes of excitation exist. Each additional mode is represented by an additional L-C circuit. The circuit which models the composite resonator is shown in Figure 5. The circuit was analyzed with E.C.A.P. (Electronic Circuit Analysis Program) to compare the frequency response of the model with the frequency response of the actual transducer. A sample graph, as generated witn E.C.A.P., is given in Figure 6. This graph compares favorably with the actual response of the composite resonator (not shown).

TUNNEL SOFTWARE OPERATING PROCEDURE

A. Initialization

1. Turn on the video monitor (Monitor III)

   Turn on the printer (Epson)

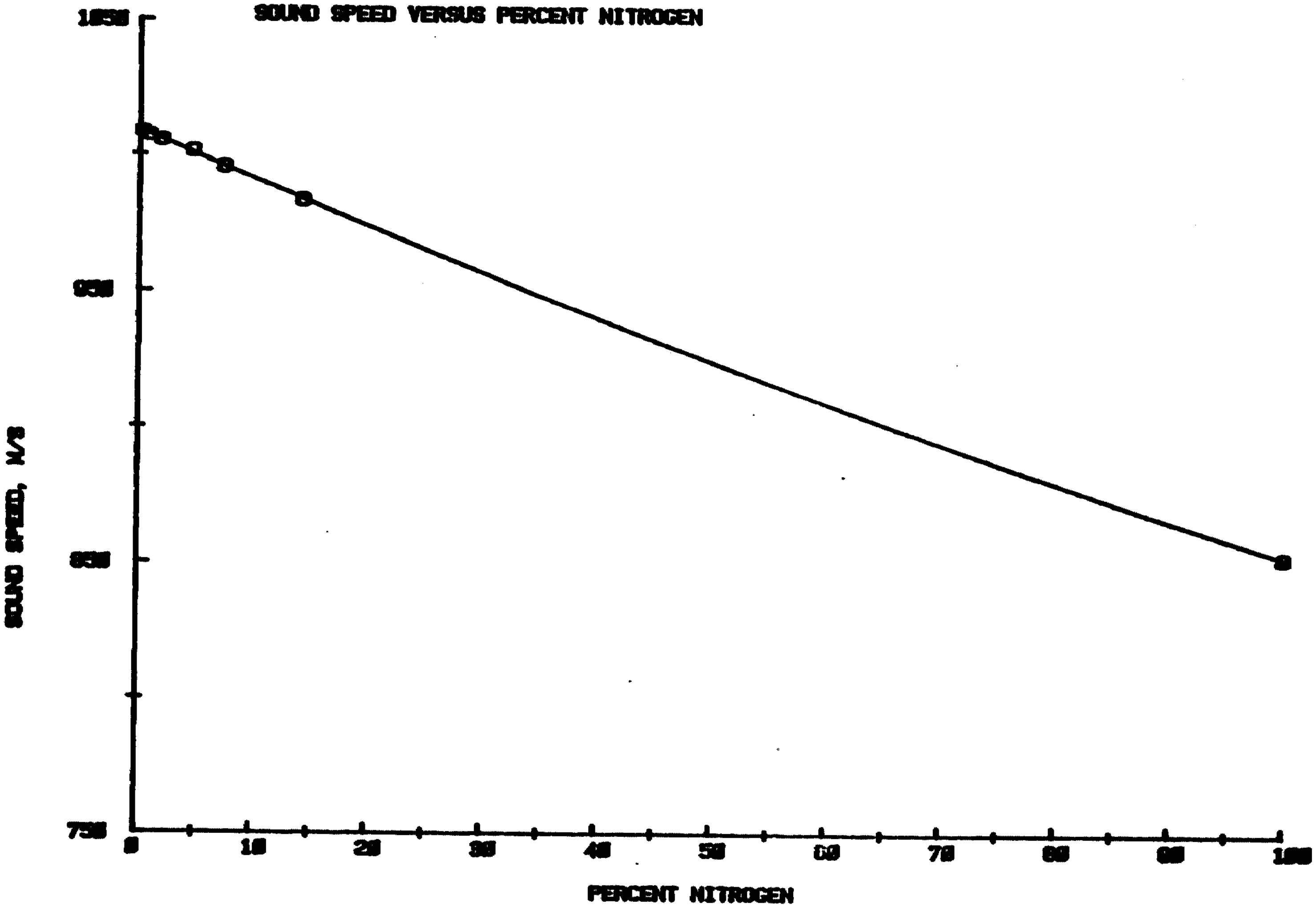


Figure 3. Graph of Sound Speed Versus Percent Nitrogen (0-100%).

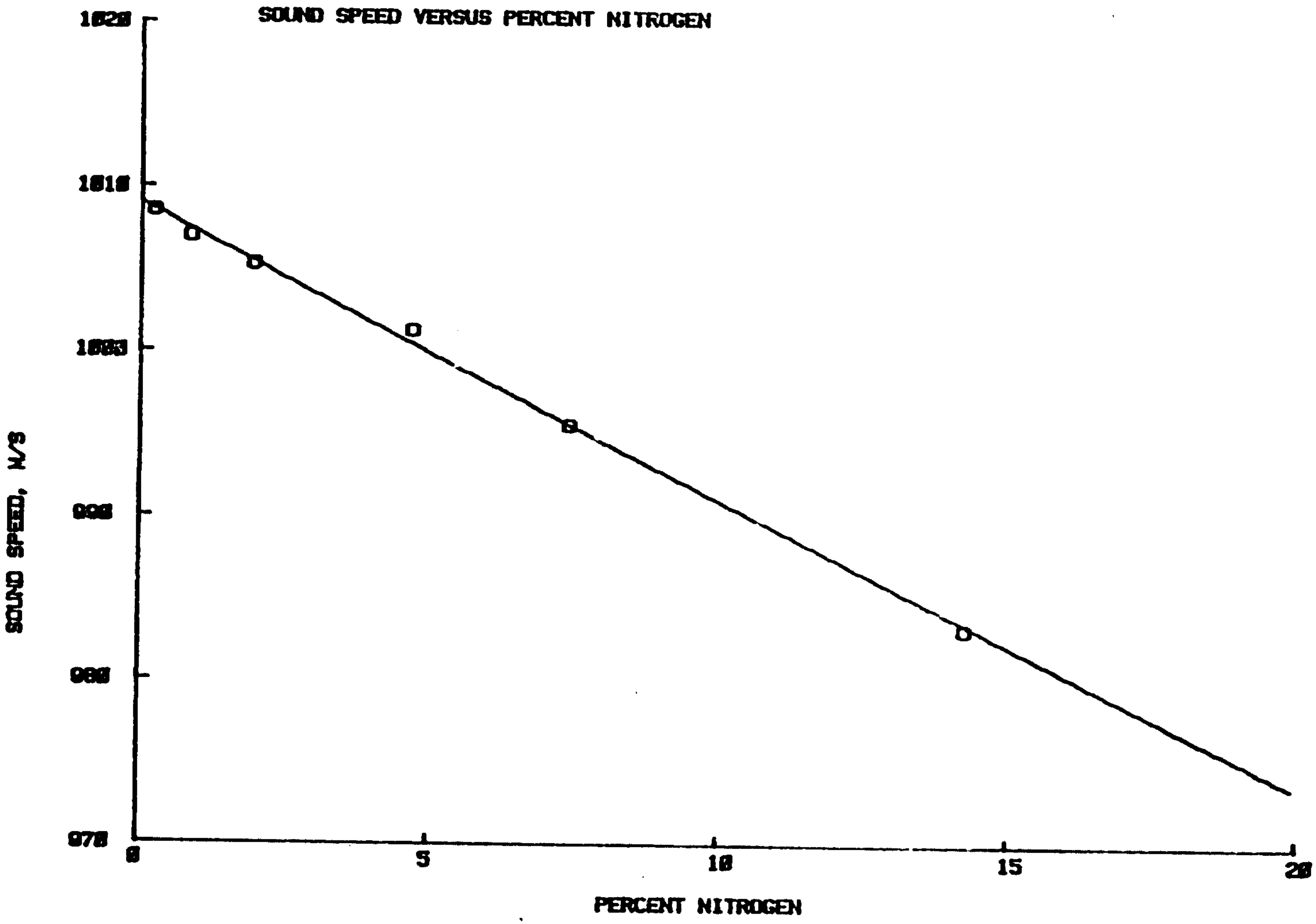
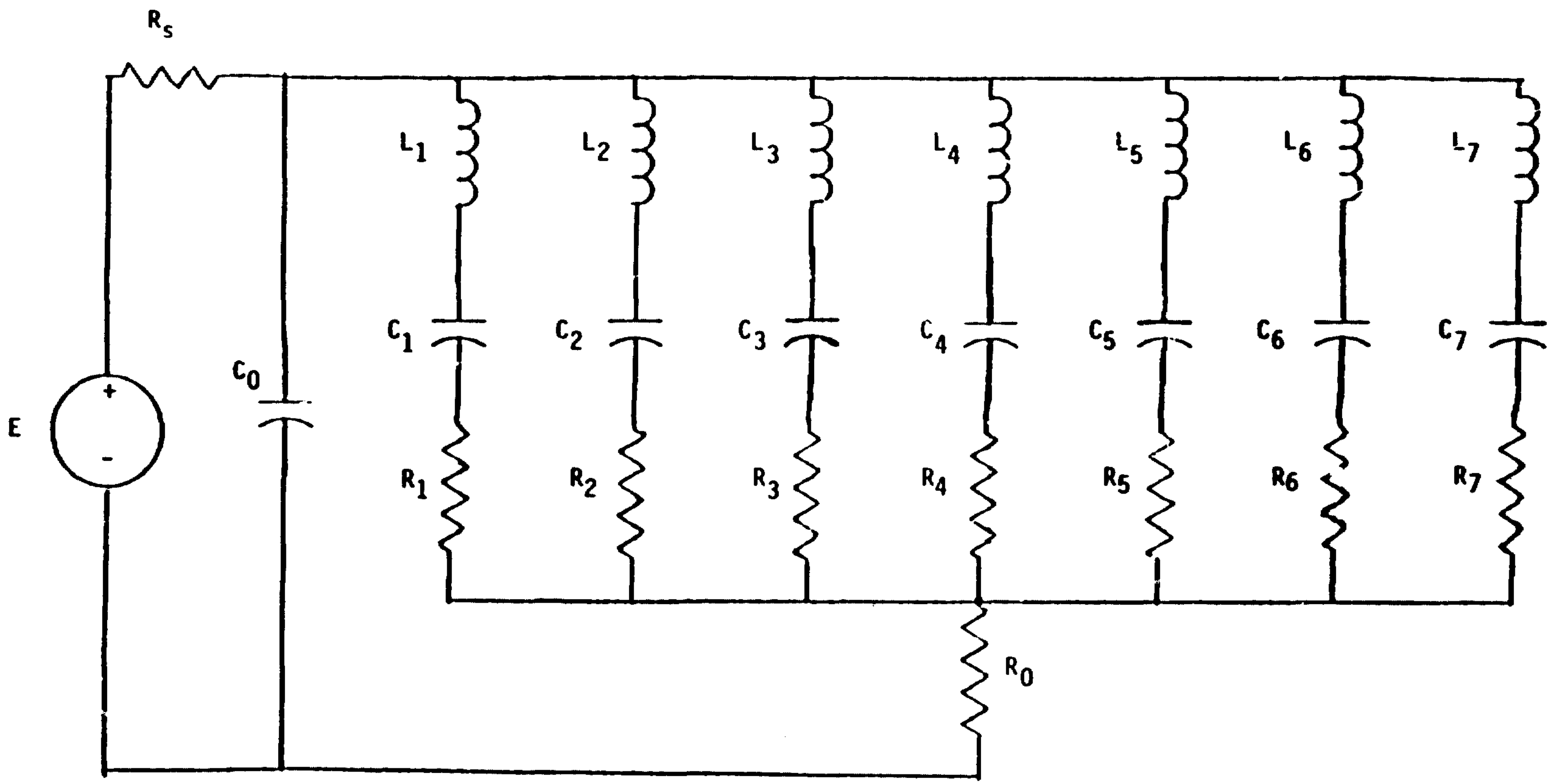


Figure 4. Graph of Sound Speed Versus Percent Nitrogen (0-20%).



E	= 1000V
R_S	= 0.1Ω
L_1	= 2.77 μH
L_2	= 620 μH
L_3	= 1.26 μH
L_4	= 585 μH
L_5	= 1.8 μH
L_6	= 892 μH
L_7	= 518 μH
C_0	= 600 pF
C_1	= 295 pF
C_2	= 207 pF
C_3	= 39 pF
C_4	= 71.2 pF
C_5	= 211 pF
C_6	= 107 pF
C_7	= 82 pF
R_0	= 25 Ω
R_1	= 26 Ω
R_2	= 14 Ω
R_3	= 131 Ω
R_4	= 1×10^{-7}
R_5	= 26 Ω
R_6	= 26 Ω
R_7	= 5.9 Ω

Figure 5. Electrical Equivalent Circuit of a Transducer.

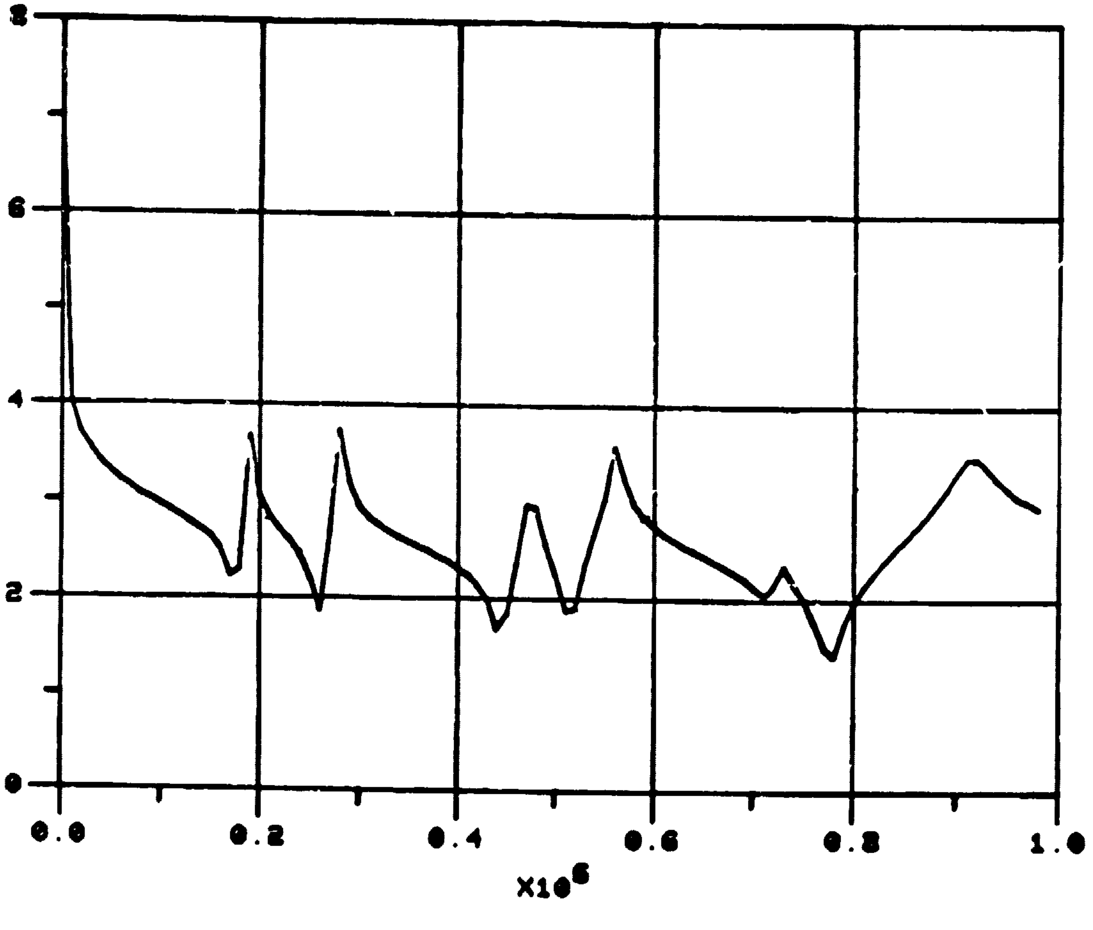


Figure 6. Frequency Response of an Electrical Circuit Modelling a Transducer.

Turn on the computer (APPLE II+)

Note: Power indicator lights on the three units should be on.

The "IN USE" light on Disk Drive 1 should be on.

2. Insert a prepared disk into Drive 1.

3. Close the door of the disk drive.

4. Wait for the "IN USE" light of Drive 1 to go off.

5. Observe the video monitor

One will see

```
ENTER YOUR CHOICE

(1) RETRIEVE DATA FROM DISK

(2) DISPLAY AND COLLECT DATA FROM INSTRUMENTS

(3) EXIT PROGRAM
```

6. If one wants to print data already saved, type "1" followed by <return>.

7. If one wants to gather new data from the instruments, type "2" followed by <return>.

8. If one wants to quit, type "3" followed by <return>.

B. Retrieval of Data from Disk

1. If one just typed "1", as described above, the disk will have been activated for approximately ten seconds.

The video monitor will show:

```
ENTER YOUR CHOICE

(1) RETRIEVE DATA FROM DISK

(2) RECORD DATA FROM INSTRUMENTS

(3) EXIT PROGRAM
```

2. If one wishes to retrieve already stored data, type "1" followed by

<return>.

3. If one wishes to gather new data from the instruments, type "2" followed by <return>.

4. If one wishes to exit, type "3" followed by <return>.

(Data Retrieval)

5. You will see

   ENTER THE RUN NUMBER

6. Type the run number of the desired data followed by <return>.

7. Disk drive 1 will be activated.

8. Wait for the "IN USE" light to go off.

9. You will see

   (1) OUTPUT TO PRINTER

   (2) OUTPUT TO PRINTER WITH AVERAGING

10. Type the number of your choice, followed by <return>.

11. Be certain the "ON LINE" light of the printer is illuminated. If the light is off, press the "ON LINE" button. The light should now be on.

12. Check all cable connections if a problem occurs.

13. After the printing is completed, the computer will display the menu shown in B1 above.

Notes: (1) Choice 1 prints all data taken. Time is minute:second.

(2) Choice 2 prints the data averaged together within each second. Time is minute:second.

C. Collecting Data

1. One will see the menu listed in A5.

2. Enter your choice as described in A6, 7 and 8.

3. One will see

```
ENTER THE RUN NUMBER
```

4. Type the run number for the data followed by <return>.

5. The screen will show the present readings of the instruments.

6. The screen should show, for example:

```
1) CELL OUTPUT     0.00   MV

2) AMP OUTPUT      0.00   VOLTS

3) CELL TEMP       8.54   DEGREES

4) SAMPLE TEMP      0     DEGREES

5) SAMPLE FLOW      53    CCM

6) REF FLOW         22    CCM

7) CAVITY PRESS    -10    TORR

8) INLET PRESS      2     TORR

                TUE   OCT   9   3:30:40 pm
```

7. To save data, press the button on game paddle 1.

8. To stop gathering data, press the button on game paddle Ø.

9. The disk drive will be activated.

10. The menu in A5 will be displayed.

11. Proceed as desired.

D. Exiting the Program

This choice ends the program.

One can observe the files saved on disk by typing "CATALOG" <return>.

Other available commands may be found in the APPLE DOS MANUAL.

APPENDIX A:

SOFTWARE FOR DATA ACQUISITION SYSTEM

START.FETCH

```
]


]
]LIST

0  REM    START.FETCH 9/20/84
1  ONERR  GOTO 2000
3 CLECT = 34048: REM   $8500
5  HIMEM: 8192: REM     $2000
7  DIM A%(1,8)
10 D$ =  CHR$ (4):F1 = 0
15  PRINT D$;"BLOAD GETAI13,A$9500"
17  PRINT D$;"BLOAD SEVEN.INCH.0,A$8500"
18  PR# 0: HOME : CALL 1002:F1 = 0
19  POKE 34214,100: REM     CHANGE FOR SAMPLE RATE
20  PRINT "ENTER YOUR CHOICE"
30  PRINT "(1)  RETRIEVE DATA FROM DISK"
40  PRINT "(2)  DISPLAY AND COLLECT DATA FROM INSTRUMENTS"
45  PRINT "(3)  EXIT PROGRAM"
50  INPUT CHOICE
60  IF CHOICE <  > 1 AND CHOICE <  > 2 AND CHOICE <  > 3 THEN  GOTO 20
65  IF CHOICE = 3 THEN  END
70  IF CHOICE = 2 THEN  GOTO 500
80  REM  RETRIEVE DATA
90  PRINT D$;"RUN BRING.BACK"
500  REM  DISPLAY AND DATA COLLECTION
510  INPUT "ENTER THE RUN NUMBER ";RU
520  GOTO 1000: REM   DISPLAY
530  CALL CLECT
540 SIZE = ( PEEK (2) * 256 +  PEEK (1)) - 2 * 16 ^ 3
545 D$ =  CHR$ (4)
546  CALL 1002
547  PRINT
550  PRINT D$;"BSAVE RUN";RU;",A$2000,L";SIZE
560  GOTO 18
900  DATA     5,8,5,1,0,1,0,1,0,1,0,1,0,1,5,1,5,1
910  REM  THE ABOVE NUMBERS ARE  SLOT,SUM OF AVERGS,GAIN,AVERG FOR CHNL,G
     AIN,AVERG,ETC.
1000  REM   DISPLAY ROUTINE
1010  HOME
1020  VTAB 2: HTAB 2: PRINT "1. CELL OUTPUT ";
1030  HTAB 33: PRINT "  MV ";
1040  VTAB 4: HTAB 2: PRINT "2. AMP OUTPUT";
1050  HTAB 33: PRINT "VOLTS";
1060  VTAB 6: HTAB 2: PRINT "3. CELL TEMP";
1070  HTAB 33: PRINT "DEGREES ";
1080  VTAB 8: HTAB 2: PRINT "4. SAMPLE TEMP";
1090  HTAB 33: PRINT "DEGREES";
1100  VTAB 10: HTAB 2: PRINT "5. SAMPLE FLOW";
1110  HTAB 33: PRINT "  CCM";
1120  VTAB 12: HTAB 2: PRINT "6. REF FLOW";
1130  HTAB 33: PRINT "  CCM";
1140  VTAB 14: HTAB 2: PRINT "7. CAVITY PRES";
1150  HTAB 33: PRINT " TORR"
1160  VTAB 16: HTAB 2: PRINT "8. INLET PRES";
1170  HTAB 33: PRINT " TORR";
1172  POKE 33,33: REM   KEEP UNITS ON THE SCREEN
1173  RESTORE
1175  IF F1 THEN  GOTO 1510
1210  READ SLOT
```

```
1220 A%(0,0) = SLOT
1230  READ M
1240 A%(1,0) =  - M
1245 DUM = 1
1250  FOR CHNL = 1 TO 8
1255  READ GAIN,AVERG
1260  FOR J = DUM TO (AVERG + DUM - 1)
1270 A%(0,J) = CHNL + 16 * GAIN
1275  NEXT J
1280 DUM = DUM + AVERG
1285  NEXT CHNL
1290  POKE 8,1: CALL 38144
1295 DUM = 1
1300  RESTORE : READ A,B
1305  FOR CHNL = 1 TO 8
1306 SUM = 0: READ C,AVERG
1307  FOR J = DUM TO (AVERG + DUM - 1)
1308 SUM = A%(1,J) + SUM
1309  NEXT J
1311 A%(1,CHNL) =  INT (SUM / AVERG)
1313 DUM = DUM + AVERG
1315  NEXT CHNL
1370  FOR I = 1 TO 8
1380 TEMP = A%(1,I)
1390  ON I GOSUB 11000,12000,13000,14000,15000,16000,17000,18000
1400  NEXT I
1402  PR# 4: IN# 4
1404  INPUT "%";T$
1406  PR# 0: IN# 0
1409  VTAB 20: HTAB 10: PRINT T$;" ";
1410  IF  PEEK ( - 16286) > 127 THEN  POKE 33,40: HOME : GOTO 530
1420  GOTO 1290
1500 J = 2 * 16 ^ 3 + 5: REM  $2005
1505  GOTO 1000
1510 COUNT = 1
1530  FOR I = 1 TO 8
1540 A(1,I) =  PEEK (J) * 256 +  PEEK (J + 1)
1550 J = J + 2
1560  NEXT I
1570  FOR I = 1 TO 8
1572 TEMP = A%(1,I)
1574  ON I GOSUB 11000,12000,13000,14000,15000,16000,17000,18000
1576  NEXT I
1580 COUNT = COUNT + 1
1590  IF J >  = LN + 8192 THEN  GOTO 18
1600  IF COUNT = 21 THEN J = J + 4: GOTO 1510
1610  GOTO 1530
2000  REM   ERROR HANDLING ROUTINE (DOS)
2010 QZ =  PEEK (222): REM   ERROR CODE
2020  FLASH
2030  IF QZ = 8 THEN  PRINT "I/O ERROR": PRINT "CHECK THE DISK DRIVE": GOTO
     2060
2040  IF QZ = 9 THEN  PRINT "DISK FULL": PRINT "CHANGE DISKS": GOTO 2060
2050  PRINT "FATAL ERROR"
2060  PRINT "PRESS 'RETURN' WHEN READY"
2065  NORMAL : PRINT
2070  GET AA$
2080  IF  ASC (AA$) <  > 13 THEN  GOTO 2070
2100  PRINT
2105  IF QZ <  > 8 AND QZ <  > 9 THEN  GOTO 18
2110  GOTO 540
3800  REM
3900  REM
11000  VTAB 2: HTAB 22
11010 QZ = 2 / 4096
11020 RESULT = TEMP * QZ - 1.0
```

```
11030 RESULT = 100 * RESULT
11035 RESULT =  INT (100 * RESULT) / 100
11040 Z$ =  STR$ (RESULT) + "       "
11050  PRINT  LEFT$ (Z$,6);
11060  RETURN
12000  VTAB 4: HTAB 22
12010 QZ = 5 / 4096
12012  IF TEMP < 9 THEN  PRINT "0.00";: GOTO 12050
12020 RESULT = TEMP * QZ
12030 Z$ =  STR$ (RESULT) + "       "
12040  PRINT  LEFT$ (Z$,4);
12050  RETURN
13000  VTAB 6: HTAB 22
13010 QZ = 5000 / 4096
13020 RESULT = TEMP * QZ
13030 Z$ =  STR$ (RESULT) + "        "
13040  PRINT  LEFT$ (Z$,6);
13050  RETURN
14000  VTAB 8: HTAB 22
14010 QZ = 5 / 4096
14020 RESULT = TEMP * QZ * 1000
14030 RESULT =  INT (RESULT + 0.5)
14040 Z$ =  STR$ (RESULT)
14070 Z$ = Z$ + "       "
14080  PRINT  LEFT$ (Z$,6);
14090  RETURN
15000  VTAB 10: HTAB 22
15010 QZ = 20000 / 4096
15020 RESULT = TEMP * QZ
15030 P$ =  STR$ (RESULT) + ".       "
15040 Z$ = ""
15050  FOR QZ = 1 TO 6
15060  IF  MID$ (P$,QZ,1) <  > "." THEN Z$ = Z$ +  MID$ (P$,QZ,1): NEXT Q
     Z
15070 Z$ = Z$ + "       "
15080  PRINT  LEFT$ (Z$,6);
15100  RETURN
16000  VTAB 12: HTAB 22
16010 QZ = 2000 / 4096
16020 RESULT = TEMP * QZ
16030 Z$ = ""
16040 P$ =  STR$ (RESULT) + ".       "
16050  FOR QZ = 1 TO 6
16060  IF  MID$ (P$,QZ,1) <  > "." THEN Z$ = Z$ +  MID$ (P$,QZ,1): NEXT Q
     Z
16070 Z$ = Z$ + "       "
16080  PRINT  LEFT$ (Z$,6);
16100  RETURN
17000  VTAB 14: HTAB 22
17010 QZ = 2 / 4096
17020 RESULT = (TEMP - 2048) * QZ
17030 RESULT = 100 * RESULT
17035 RESULT =  INT (10 * (RESULT + 0.05)) / 10
17040 Z$ = ""
17055  IF  ABS (RESULT) < 0.05 THEN Z$ = "0.00": GOTO 17080
17056  IF  ABS (RESULT) =  > 0.05 THEN Z$ =  STR$ (RESULT)
17080 Z$ = Z$ + "      "
17090  PRINT  LEFT$ (Z$,6);
17100  RETURN
18000  VTAB 16: HTAB 22
18010 QZ = 2 / 4096
18020 RESULT = (TEMP - 2048) * QZ
18030 RESULT = RESULT * 1000
18040 Z$ = ""
18050 P$ =  STR$ (RESULT) + ".       "
18055  IF  ABS (RESULT) < 1 THEN P$ = "0." + P$
```

```
18060   FOR QZ = 1 TO 6
18070   IF  MID$ (P$,QZ,1) <   > "." THEN Z$ = Z$ +  MID$ (P$,QZ,1): NEXT Q
    Z
18080 Z$ = Z$ + "       "
18090   PRINT  LEFT$ (Z$,6);
18100   RETURN
```

APPENDIX B:

SOFTWARE FOR DATA ACQUISITION SYSTEM

BRING.BACK

```
]
]LIST

0  REM      BRING.BACK 9/20/84
5  HIMEM: 8192: REM     $2000
7  DIM XX(4)
10 D$ =  CHR$ (4):F1 = 0
18  PR# 0: HOME : CALL 1002:F1 = 0
20  PRINT "ENTER YOUR CHOICE"
30  PRINT "(1)  RETRIEVE DATA FROM DISK"
40  PRINT "(2)  RECORD DATA FROM INSTRUMENTS"
45  PRINT "(3)  EXIT PROGRAM"
50  INPUT CHOICE
60  IF CHOICE <  > 1 AND CHOICE <  > 2 AND CHOICE <  > 3 THEN  GOTO 20
65  IF CHOICE = 3 THEN  END
70  IF CHOICE = 2 THEN  PRINT D$;"RUN START.FETCH"
80  PRINT : PRINT
90  INPUT "ENTER THE RUN NUMBER  ";RU
95 FL$ = "RUN" +  STR$ (RU)
100  REM  RETRIEVE DATA FROM DISK
110  PRINT D$;"BLOAD ";FL$;",A$2000"
120 LN =  PEEK (43617) * 256 +  PEEK (43616): REM   LENGTH OF THE DATA
140  PRINT "(1)  OUTPUT TO PRINTER"
145  PRINT "(2)  OUTPUT TO PRINTER WITH AVERAGING "
150  INPUT CHOICE
160  IF CHOICE <  > 1 AND CHOICE <  > 2 THEN  GOTO 130
175  IF CHOICE = 2 THEN  GOTO 3000
180  REM  PRINTING ROUTINE
200  PR# 1
210  FOR I = 1 TO 5
220  PRINT
230  NEXT I
235  PRINT "         RUN NUMBER ";RU: PRINT : PRINT
240  PRINT "            TIME     CELL OUTPUT    AMP OUTPUT    CELL TEMP    SAM
     PLE TEMP"
250  PRINT "            (SEC)      (MV)            (V)       (DEGREES)    (DEG
     REES)"
260 FLAG = 0
265 I = 1: IF FLAG THEN I = 9
275  IF FLAG THEN I = I - 8: REM  RESET I TO GET TIME
280 A$ =  CHR$ ( PEEK (8192 + I)) +  CHR$ ( PEEK (8193 + I)) + ":" +  CHR$
     ( PEEK (8194 + I)) +  CHR$ ( PEEK (8195 + I))
290  IF FLAG THEN I = I + 12: GOTO 300: REM    CHECK THIS FOR THE FIRST T
     IME IN THE LOOP
295 I = I + 4
300  PRINT "            ";A$;"        ";
305 ON = 1
310  FOR J = I TO I + 6 STEP 2
320 TEMP =  PEEK (8192 + J) * 256 +  PEEK (8193 + J)
330  IF  NOT (FLAG) THEN  ON ON GOSUB 11010,12010,13010,14010
340  IF FLAG THEN  ON ON GOSUB 15010,16010,17010,18010
350  PRINT "      ";
355 ON = ON + 1
360  NEXT J
370  PRINT
380 I = I + 16
410  IF I >  = LN AND  NOT (FLAG) THEN  GOTO 430
415  IF I >  = LN AND FLAG THEN  GOTO 18
420  GOTO 275
430  PRINT : PRINT : PRINT
435  PRINT "         RUN NUMBER ";RU: PPINT : PRINT
440  PRINT "            TIME    SAMPLE FLOW    REF FLOW    CAVITY PRES    I
     NLET PRES"
450  PRINT "            (SEC)       (CCM)          (CCM)       (TORR)
```

```
      (TORR)"
470 FLAG = 1
480  GOTO 265
3000  REM   AVERAGING ROUTINE
3005  PR# 1
3010  FOR I = 1 TO 5: PRINT : NEXT I
3020  PRINT "          RUN NUMBER ";RU: PRINT : PRINT
3030  PRINT "                 TIME     CELL OUTPUT   AMP OUTPUT   CELL TEMP   SA
     MPLE TEMP"
3040  PRINT "                 (SEC)      (MV)            (V)        (DEGREES)   (DE
     GREES)"
3050 I = 1
3060 A$ =  CHR$ ( PEEK (8192 + I)) +  CHR$ ( PEEK (8193 + I)) + ":" +  CHR$
     ( PEEK (8194 + I)) +  CHR$ ( PEEK (8195 + I))
3070 TM =  PEEK (8195 + I)
3080 I = I + 4
3090  FOR X = 1 TO 4:XX(X) = 0: NEXT X
3100 COUNT = 0
3110  FOR X = 1 TO 4
3120 XX(X) =  PEEK (8192 + I) * 256 +  PEEK (8193 + I) + XX(X)
3130 I = I + 2
3140  NEXT X
3150 COUNT = COUNT + 1
3155 I = I + 8
3160 TT =  PEEK (8195 + I)
3170  IF TT = TM THEN I = I + 4: GOTO 3110
3180  REM  AVERAGE DATA
3190  FOR X = 1 TO 4
3200 XX(X) = XX(X) / COUNT
3210  NEXT X
3220  PRINT "              ";A$;"        ";
3230  FOR X = 1 TO 4
3240 TEMP = XX(X)
3250  ON X GOSUB 11010,12010,13010,14010
3260  PRINT "         ";
3270  NEXT X
3275  PRINT
3280  IF I < LN - 8 THEN  GOTO 3060
3500 I = 1
3510  FOR I = 1 TO 3: PRINT : NEXT I
3515  PRINT "          RUN NUMBER ";RU: PRINT : PRINT
3530  PRINT "                 TIME     SAMPLE FLOW    REF FLOW    CAVITY PRES
     INLET PRES"
3535  PRINT "                 (SEC)        (CCM)          (CCM)        (TORR)
       (TORR)"
3550 I = 1
3560 A$ =  CHR$ ( PEEK (8192 + I)) +  CHR$ ( PEEK (8193 + I)) + ":" +  CHR$
     ( PEEK (8194 + I)) +  CHR$ ( PEEK (8195 + I))
3570 TM =  PEEK (8195 + I)
3580 I = I + 12
3590  FOR X = 1 TO 4:XX(X) = 0: NEXT X
3600 COUNT = 0
3610  FOR X = 1 TO 4
3620 XX(X) =  PEEK (8192 + I) * 256 +  PEEK (8193 + I) + XX(X)
3630 I = I + 2
3640  NEXT X
3650 COUNT = COUNT + 1
3660 TT =  PEEK (8195 + I)
3670  IF TT = TM THEN I = I + 12: GOTO 3610
3680  REM    AVERAGE DATA
3690  FOR X = 1 TO 4
3700 XX(X) = XX(X) / COUNT
3710  NEXT X
3720  PRINT "              ";A$;"        ";
3730  FOR X = 1 TO 4
3740 TEMP = XX(X)
```

```
3750  ON X GOSUB 15010,16010,17010,18010
3760  PRINT "        ";
3770  NEXT X
3775  PRINT
3780  IF I < LN THEN  GOTO 3560
3790  GOTO 18
3800  REM
3900  REM
11000  VTAB 2: HTAB 22
11010 QZ = 2 / 4096
11020 RESULT = TEMP * QZ - 1
11030 RESULT = 100 * RESULT
11035 RESULT =  INT (100 * RESULT) / 100
11040 Z$ =  STR$ (RESULT) + "      "
11050  PRINT  LEFT$ (Z$,6);
11060  RETURN
12000  VTAB 4: HTAB 22
12010 QZ = 5 / 4096
12012  IF TEMP < 9 THEN  PRINT "0.00";: GOTO 12050
12020 RESULT = TEMP * QZ
12030 Z$ =  STR$ (RESULT) + "      "
12040  PRINT  LEFT$ (Z$,4);
12050  RETURN
13000  VTAB 6: HTAB 22
13010 QZ = 5000 / 4096
13020 RESULT = TEMP * QZ
13030 Z$ =  STP$ (RESULT) + "       "
13040  PRINT  LEFT$ (Z$,6);
13050  RETURN
14000  VTAB 8: HTAB 22
14010 QZ = 5 / 4096
14020 RESULT = TEMP * QZ * 1000
14030 RESULT =  INT (RESULT + 0.5)
14040 Z$ =  STR$ (RESULT)
14070 Z$ = Z$ + "      "
14080  PRINT  LEFT$ (Z$,6);
14090  RETURN
15000  VTAB 10: HTAB 22
15010 QZ = 20000 / 4096
15020 RESULT = TEMP * QZ
15030 P$ =  STR$ (RESULT) + ".       "
15040 Z$ = ""
15050  FOR QZ = 1 TO 6
15060  IF  MID$ (P$,QZ,1) <  > "." THEN Z$ = Z$ +  MID$ (P$,QZ,1): NEXT Q
      Z
15070 Z$ = Z$ + "       "
15080  PRINT  LEFT$ (Z$,6);
15100  RETURN
16000  VTAB 12: HTAB 22
16010 QZ = 2000 / 4096
16020 RESULT = TEMP * QZ
16030 Z$ = ""
16040 P$ =  STR$ (RESULT) + ".      "
16050  FOR QZ = 1 TO 6
16060  IF  MID$ (P$,QZ,1) <  > "." THEN Z$ = Z$ +  MID$ (P$,QZ,1): NEXT Q
      Z
16070 Z$ = Z$ + "      "
16080  PRINT  LEFT$ (Z$,6);
16100  RETURN
17000  VTAB 14: HTAB 22
17010 QZ = 2 / 4096
17020 RESULT = (TEMP - 2048) * QZ
17030 RESULT = 100 * RESULT
17035 RESULT =  INT (10 * (RESULT + 0.05)) / 10
17040 Z$ = ""
17055  IF  ABS (RESULT) < 0.05 THEN Z$ = "0.00": GOTO 17080
```

```
17056  IF  ABS (RESULT) =  > 0.05 THEN Z$ =  STR$ (RESULT)
17080 Z$ = Z$ + "      "
17090  PRINT  LEFT$ (Z$,6);
17100  RETURN
18000  VTAB 16: HTAB 22
18010 QZ = 2 / 4096
18020 RESULT = (TEMP - 2048) * QZ
18030 RESULT = RESULT * 1000
18040 Z$ = ""
18050 P$ =  STR$ (RESULT) + ".      "
18055  IF  ABS (RESULT) < 1 THEN P$ = "0." + P$
18060  FOR QZ = 1 TO 6
18070  IF  MID$ (P$,QZ,1) <  > "." THEN Z$ = Z$ +  MID$ (P$,QZ,1): NEXT Q
     Z
18080 Z$ = Z$ + "      "
18090  PRINT  LEFT$ (Z$,6);
18100  RETURN
```

APPENDIX C:

SOFTWARE FOR DATA ACQUISITION SYSTEM

SEVEN.INCH.S

```
:ASM
                * SEVEN.INCH.S, 10/16/84
                *
                                SYM
                                ORG    $8500
                RDTCP           EQU    $C408      ; READ ENTRY POINT
                WTTCP           EQU    $C40B      ; WRITE ENTRY POINT
                SLOT            EQU    $C080+$50
                PTR1            EQU    $01
                PTR2            EQU    $02
                UPBND           EQU    $80
                LOBND           EQU    $00
                PSHBTM          EQU    $C061
                GN1             EQU    $51
                GN2             EQU    $02
                GN3             EQU    $13
                GN4             EQU    $14
                GN5             EQU    $05
                GN6             EQU    $06
                GN7             EQU    $57
                GN8             EQU    $58
                DEL1            EQU    $03
                DEL2            EQU    $04
                TEMP1           EQU    $05
                * INSERT MY VARIABLES HERE
                *
                * INITIALIZE ROUTINE HERE
                *
8500: A9 20                     LDA    #$20
8502: 85 02                     STA    PTR2       ; CHECK VALUES HERE
8504: A9 00                     LDA    #$00
8506: 85 01                     STA    PTR1
8508: 20 25 85  START           JSR    TIME       ; GET THUNDERCLOCK DATA
850B: 20 46 85                  JSR    ATOD      ; GET DATA FROM A/D
850E: 20 A5 85                  JSR    DELAY     ; DELAY FOR 20 PER SECOND
8511: A5 02     HALT            LDA    PTR2       ; CHECK IF TOO MUCH DATA
8513: C9 80                     CMP    #UPBND          ;
8515: D0 03                     BNE    CONT      ; KEEP GOING
8517: 18                        CLC
8518: 90 05                     BCC    WAIT
851A: 2C 61 C0  CONT            BIT    PSHBTM ; CHECK PUSH BUTTON
851D: 10 E9                     BPL    START      ; TAKE MORE DATA
851F: 2C 61 C0  WAIT            BIT    PSHBTM ;
8522: 10 ED                     BPL    HALT          ; TOO MUCH DATA SO WAIT
8524: 60                        RTS
                *
                *****
                *
8525: A9 A3     TIME            LDA    #$A3     ; "#"= NUMERIC MODE
8527: 20 0B C4                  JSR    WTTCP      ; SET NUMERIC READ MODE
852A: 20 08 C4                  JSR    RDTCP      ; PUT TIME IN GETLN BUFFER
852D: AD 0C 02                  LDA    $20C        ; 1ST DIGIT OF MINUTE
8530: 20 9A 85                  JSR    SAVE
8533: AD 0D 02                  LDA    $20D       ; 2ND DIGIT OF MINUTE
8536: 20 9A 85                  JSR    SAVE
8539: AD 0F 02                  LDA    $20F        ; 1ST DIGIT OF SECOND
853C: 20 9A 85                  JSR    SAVE
853F: AD 10 02                  LDA    $210        ; 2ND DIGIT OF SECOND
8542: 20 9A 85                  JSR    SAVE
8545: 60                        RTS
                *
                *****
                *
```

```
8546: A9 51    63   ATOD     LDA  #GN1
8548: 8D DO CO 64            STA  SLOT
854B: 20 87 85 65            JSR  STORE
854E: A9 02    66            LDA  #GN2
8550: 8D DO CO 67            STA  SLOT
8553: 20 87 85 68            JSR  STORE
8556: A9 13    69            LDA  #GN3
8558: 8D DO CO 70            STA  SLOT
855B: 20 87 85 71            JSR  STORE
855E: A9 14    72            LDA  #GN4
8560: 8D DO CO 73            STA  SLOT
8563: 20 87 85 74            JSR  STORE
8566: A9 05    75            LDA  #GN5
8568: 8D DO CO 76            STA  SLOT
856B: 20 87 85 77            JSR  STORE
856E: A9 06    78            LDA  #GN6
8570: 8D DO CO 79            STA  SLOT
8573: 20 87 85 80            JSR  STORE
8576: A9 57    81            LDA  #GN7
8578: 8D DO CO 82            STA  SLOT
857B: 20 87 85 83            JSR  STORE
857E: A9 58    84            LDA  #GN8
8580: 8D DO CO 85            STA  SLOT
8583: 20 87 85 86            JSR  STORE
8586: 60       87            RTS
               88   *
               89   *****
               90   *
8587: 48       91   STORE    PHA            ; DELAY TO COMPLETE
8588: 68       92            PLA            ; 4 BITS
8589: AD D1 CO 93            LDA  SLOT+1  ; MSB
858C: 29 OF    94            AND  #$OF      ;
858E: EA       95            NOP            ; DELAY
858F: AE DO CO 96            LDX  SLOT
8592: 20 9A 85 97            JSR  SAVE
8595: 8A       98            TXA
8596: 20 9A 85 99            JSR  SAVE
8599: 60       100           RTS
               101  *
               102  *****
               103  *
859A: E6 01    104  SAVE     INC  PTR1
859C: DO 02    105           BNE  SKIP
859E: E6 02    106           INC  PTR2
85A0: A0 00    107  SKIP     LDY  #$00
85A2: 91 01    108           STA  (PTR1),Y
85A4: 60       109           RTS
               110  *
               111  *****
               112  *
85A5: A9 OF    113  DELAY    LDA  #$OF
85A7: 65 03    114           STA  DEL1
85A9: A9 FF    115  B        LDA  #$FF
85AB: 85 04    116           STA  DEL2
85AD: C6 04    117  A        DEC  DEL2
85AF: DO FC    118           BNE  A
85B1: C6 03    119           DEC  DEL1
85B3: DO F4    120           BNE  B
85B5: 60       121           RTS
               122  *
               123  *****
               124  *


---- END ASSEMBLY ---
```

```
TOTAL ERRORS: 0

182 BYTES GENERATED THIS ASSEMBLY
```