127
10/24/80
SAND 79-2313
Unlimited Release
UC-37

A Versatile Microcomputer-Based Temperature Controller

MASTER

Victor R. Yarberry

**Distribution
Category UC-37**

See

CONTENTS

	<u>Page</u>
Introduction	7
Hardware Description	9
Software Description	10
Operation	13
Parameter Determination	13
Programming	14
Performance	18
Summary	20
References	21
APPENDIX A - Suppliers	23
APPENDIX B - Program Listings	27
APPENDIX C - Baud Rates and Addresses	47

ILLUSTRATIONS

Figure

1 Functional Diagram of Temperature Controller	7
2 System Block Diagram	9
3 System Flow Chart	11
4 Open Loop Response With Stepped Power Input	19
5 Controlled Step Input Response-Parameters Optimized	19
6 Linear Ramp Generated By Temperature Controller	20

A VERSATILE MICROCOMPUTER-BASED TEMPERATURE CONTROLLER

Introduction

A number of analog temperature controllers are commercially available today, but they have certain limitations, such as inflexible parameters, insufficient precision, or lack of digital interface capability, which severely limits their usefulness. The microcomputer-based temperature controller described here uses a feedback method which helps to overcome these limitations. It is based on instrumentation supplied by several commercial contractors. Figure 1 provides a functional diagram of the controller.

Figure 1. Functional Diagram of Temperature Controller

Since the control function is implemented digitally by a microcomputer, temperature readings are taken at discrete points in time. The interval between readings is the sampling period, T . This period can be specified to allow for the different response times found in various systems. The user supplies the set point or target temperature for the controller; various temperature profiles may be generated by varying the target temperature as a function of time. The numerical value of this temperature, in digital form (the command signal), is compared to a feedback signal, which is the actual digitized temperature value. The difference is the error signal. The error signal at the n th instant in

time, $e(n)$, is applied in a control algorithm to produce a corrective action, $E(n)$, which counters the error. This corrective action, in the form of a binary number, is applied to a digital-to-analog converter (DAC), which controls a power supply driving the heater. This power level causes the sample holder to acquire a specific temperature, T_F , which is sensed by a thermocouple whose output is sampled and digitized by a digital voltmeter (DVM). The DVM output is the signal which is fed back to the controller. The object of the temperature controller is to force the controlled variable, T_F , to assume the value of the command signal, whether that signal is static or dynamic.

This controller uses the proportional-integral-derivative (PID) algorithm. The continuous time PID equation is

$$E(t) = K_p \left[e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right], \quad (1)$$

where

K_p = proportional gain

T_i = integral time

T_d = derivative time.

The time-sampled version of the PID algorithm is given by

$$E(n) = E(n-1) + A_0 e(n) + A_1 e(n-1) + A_2 e(n-2), \quad (2)$$

where

$E(n)$ = past algorithm output actions

$E(n-1)$ = present algorithm output actions

$e(n)$, $e(n-1)$, $e(n-2)$ = present and past input errors

$$A_0 = K_p \left(1 + \frac{T}{T_i} + \frac{T_d}{T} \right)$$

$$A_1 = - K_p \left(1 + \frac{2T_d}{T} \right)$$

$$A_2 = K_p \left(\frac{T_d}{T} \right)$$

Hardware Description

The temperature controller hardware is based on a Z-80 micro-computer system manufactured by Cromemco. The basic modules of the S-100 bus microcomputer and the peripherals complete the system (Figure 2). The cards described below are used in a TEI mainframe.

Figure 2. System Block Diagram

The Z-80 processor is located on the single card computer (SCC). This board also features an RS-232 interface for use with the ASR 33 Teletype. Two of the three input/output (IO) ports interface with the display panel and the other port receives data from the DANA 5900 digital voltmeter. The temperature control program resides on the 32-kbyte erasable

programmable read only memory (EPROM); the 4-kbyte random access memory (RAM) board is used for temporary storage. These are all S-100 bus Cromemco cards.

A PT-488 interface card provides capability for remote control by a system controller.* This card, along with supplied software, implements the IEEE-488 bus interface.

The last S-100 card is a DAC-110 Digital-to-Analog Converter. This 12-bit card provides the high resolution necessary to maintain close temperature control. The analog output controls the heater power supply, a voltage-programmable Kepco Model 120-S. The maximum power delivered to the heater is controlled by a potentiometer on the power supply.

Also residing in the TEI mainframe is a custom-built display panel. This is a 32-character unit made up of Litronix 1416 14-segment displays. Most ASCII characters can be shown. The panel is used to display the current temperature and provide visual feedback to the operator on parameter updates.

Thermocouples provide temperature sensing. The thermocouple electromotive force (emf) is measured by the DVM. The DVM provides a binary coded decimal (BCD) output, which is coupled to the controller through a custom interface board. The interface uses a multiplexing arrangement to transmit the 6-digit BCD information to the controller using only one I/O port.

Software Description

The temperature controller program is broken down into four sections: an initialization routine, the temperature control routine, the

*An IEEE-488 bus system controller, such as a Hewlett-Packard HP9825A desktop computer, may be used.

IEEE-488 bus interface routine, and the ancillary driver routines. The temperature control routine is written in a compiled version of FORTRAN written by Microsoft. The other routines are written in assembly language. A complete program listing may be found in Appendix B. Figure 3 gives a simplified flow chart of the program; general descriptions of each routine are given below.

Figure 3. System Flow Chart

Upon power-up or reset, an initialization routine is run. This routine clears the IEEE-488 bus interface, erases the display panel, turns the heater power supply off, and sets the control parameters to their power-up conditions. The main body of the program is then entered.

The temperature control program is a continuous loop. The output (heater power supply voltage) is updated during each loop or sampling period. At the beginning of each loop, the thermocouple emf is sampled by the DVM. A FORTRAN routine, called DVMIN, transfers the BCD data from the DVM to the microcomputer. The voltage reading is then converted to a temperature by a thermocouple linearization subroutine. The temperature value is stored in a buffer for later use by the IEEE-488 interface subroutine and is also sent to the display panel for a visual readout. The PID control algorithm is then used to compute a new output value. This value is converted by the DAC to a voltage, which controls the heater power supply.

At the end of each loop, the TTY and the IEEE-488 flags are checked for activity. When the user wishes to enter the data input sequence he sets the TTY flag by pressing a key on the TTY keyboard. This allows the user to change the control parameters during the course of an experiment. The IEEE-488 flag is set whenever the system controller has passed new parameters to the temperature controller via the IEEE-488 interface bus. A FORTRAN subroutine, named HPIBIN, is called to decode and store these parameters.

After the flags are checked, a wait subroutine is used to finish out the remainder of the sampling period. The loop is then started again.

The IEEE-488 bus interface routine, named 488INT, is an interrupt-driven package based on software supplied by Pickles & Trout. In the interrupt-driven mode, an interrupt occurs whenever there are commands on the IEEE-488 bus. The IEEE-488 routine is called and interprets these commands, taking whatever action is indicated--listening, talking, etc. If new control parameters are passed, the IEEE-488 flag is set, and at the end of the IEEE-488 routine, program control is restored to the interrupted program. This makes the IEEE-488 bus interactions transparent to the user.

The ancillary routines include the TTY drivers and display panel driver. The TTY drivers are special I/O routines written for the compiled version of FORTRAN, which allow the use of a TTY without a disk operating system (for which compiled FORTRAN was written). The display panel driver allows the use of the Litronix displays as an output device.

All software was developed on a Cromemco disk operating system. It requires approximately 16 kbytes of EPROM and 2 kbytes of RAM to run.

Operation

The operations section covers two areas--parameter determination and programming.

Parameter Determination

Choosing optimum parameters is largely done by trail and error. Other methods are available but have been tried and are grossly inaccurate. To find the optimum step response, the gain (K_p) is varied first with the integration and derivative terms set to zero. Generally the gain and the maximum power settings have the most effect on the dampening and the settling time. For the step response, the initial temperature (T_o) and the final temperature (T_f) are far apart so that the maximum power, along with the thermal mass of the system, control the rate of change of the temperature. If the system gains too much momentum, it will overshoot and ring. There is a compromise between the gain and the maximum power so that T_f is reached with minimum overshoot and settling time.

In any finite gain system response, there also is a steady-state error. This error can be reduced considerably through the use of the integration term. Generally a long integration time constant (TC) is used for a large thermal TC. There is some interaction between the gain and the integration TC. If the integration TC is too big, it will cause the

response to oscillate or ring. Therefore, the gain may have to be reduced when the integration TC is added.

For systems that experience short disturbances (impulses), the derivative term may be added. This TC should reduce the effects of sudden changes from outside the system, but little experimentation has been done with this parameter.

In systems that have a very fast step response, the sampling rate may have to be increased. Too slow a sampling rate makes the system unresponsive to fast changes.

The parameters optimized for a step response will also work well when following a linear ramp. The step response is a worst case.

Programming

The following two subsections describe programming techniques for the temperature controller. First, parameter input using a TTY is discussed, with an example. The second section details parameter passing via the IEEE-488 bus interface. The examples shown use an HP 9825A controller.

Teletype -- The temperature controller may be programmed by using a teletype set for 110 baud (see Appendix C for other baud rates). The programming is done by typing in the desired response to questions displayed on the TTY.

To enter the data input sequence press the carriage return (CR). The controller responds by asking the first question. The questions are listed below, in order, with typical responses.

Gain	90
Integration TC	250
Sampling time	.5
Maximum power	100 (X)
Derivative TC	0
Thermocouple	1
Temperature setpoint	150

Each question must be answered every time the data input sequence is run. A maximum of seven digits, including decimal points and signs, is allowed. Each response is terminated by pressing CR. After the last response is entered, the temperature controller will start the temperature controlling sequence. A new data input sequence may be entered at any time by pressing CR, and the system may be reset by pressing the RESET switch on the front panel of the TEI mainframe. Note that pressing the RESET switch will turn the power supply off.

IEEE-488 Bus -- The temperature controller may be programmed via the IEEE-488 bus. The device address is preset to 01 by software code in the temperature controller (see Appendix C for other addresses). The temperature controller command codes are listed below.

Control

D	Display all variables
Sx.x	If 0, output normal
	If 1, output maximum

Variables

Gx.x	Gain
Px.x	Max power
Tx.x	Sampling time
Ux.x	Integration TC
Vx.x	Drivative TC

Wx.x

Thermocouple type

1 - Chromel-alumel

2 - Copper-constantan

NOTE: All numerical data must have a decimal point (e.g. x.x, 2.0). Maximum number of parameters transmitted in a line is 10. All command strings must begin with a blank or space. This first character is disregarded.

The first character of any string sent to the temperature controller must be a blank. This is because the first character of a string is not recognized by the temperature controller. Only 10 parameters can be sent in one line. These parameters may be transferred at any time and may have a maximum field length of seven spaces. Note that all data must include a decimal point in the field.

Two special control codes are also recognized. The code "D" causes the temperature controller to display a descriptive mnemonic and the stored value of all parameters on the display panel. This is done sequentially at approximately 2-s intervals. This control code allows the user to verify the parameters being used at any time, and does not require a data field to be associated with it. The control code 'S' is used to raise the heater power supply (PS) output to the 100% level by forcing the output of the DAC to full on (10 V). This allows the user to preset the maximum output of the PS and maintain the best resolution for that range. The following examples use the control codes found in the list above. In these examples an HP 9825A controller was used.

EXAMPLE 1. This example shows the settings for an apparatus that has a very large thermal mass and very poor coupling between the heat source and the sensor.

Gain	90
Integration TC	250
Sampling time	.5
Maximum power	100 %
Derivative TC	0

Thermocouple	1
Temperature setpoint	150

The following program code was written for an HP 9825A to program the temperature controller to the settings listed above.

HP 9825A Program Code:

```
wrt 701, " G90.U250.T.5P100.V0.W1.R150."
```

EXAMPLE 2. This example shows how to read the current temperature (in °C) into the HP9825A using variable A.

HP 9825 Program Code:

```
red 701,A
```

EXAMPLE 3. This example shows the settings for an apparatus that has a very small thermal mass and poor coupling between the sensor and the heat source.

Gain	.03
Integration TC	.5
Sampling time	2
Maximum power	100%
Derivative TC	0
Thermocouple	1
Temperature setpoint	150

HP 9825A Program Code:

```
.03→G; .5→U; 2→T; 100→P; 0→V; 1→W; 150→R
fmt1, 1x, "G", f6.2, "U", f6.2, "T", f5.2, "P", f6.1, "V", f
5.1, "W", f5.1, "R", f6.2; wrt 201, G, U, T, P, V,
W,R
```

EXAMPLE 4. This example shows how to set the maximum power using the control code 'S'. First the control code to raise the DAC output to maximum is sent.

HP 9825A Program Code:

wrt 701, " S1."

This causes the temperature controller to be overridden, and the PS to output full voltage. Or adjustable power supplies this voltage is then set to the desired output. Then the program code is used.

HP 9825A Program Code:

wrt 701, " S0."

The temperature controller will then resume controlling the temperature using the new power range.

Performance

The temperature controller was tested in an experimental high pressure (14 kbar) apparatus at various pressure and temperature levels. The apparatus represents a large thermal time constant with poor coupling between sensor and heat source. Figure 4 shows the open loop response with a stepped power input. The curve is a very slowly rising exponential that tends to level out after a long period of time. Note that this temperature level never becomes flat, but continues to increase gradually. Figure 5 shows the controlled step input response with the parameters optimized. The settling time is greatly reduced compared to the open loop response; the steady-state error is also reduced to about 0.1°C.

Figure 4. Open Loop Response With Stepped Power Input

Figure 5. Controlled Step Input Response - Parameter Optimized

The graph in Figure 6 shows a linear ramp generated by the temperature controller. As long as the slope of the desired ramp is within limits dictated by the thermal mass of the apparatus, the temperature will track very closely--usually to within 1°C.

Figure 6. Linear Ramp Generated by Temperature Controller

SUMMARY

Both step response and linear ramp mode temperature profiles have been generated for a wide variety of operating conditions. These include high and low pressures (0-30 kbar), large and small thermal mass, and various temperatures (25-250°C). Optimal parameters were found using the methods in the text. For optimized parameters, the response to a step input was an exponential rise with minimal overshoot and low steady-state error ($<0.1^{\circ}\text{C}$). The response to a linear ramp input was a well-controlled ramp whose standard deviation approached the steady-state error for ramping rates not limited by the thermal mass of the system. These results show that the performance of the temperature controller meets the design goals of precision and flexibility.

References

1. IEEE Std. 488-1978. IEEE Standard Digital Interface for Programmable Instrumentation. (New York: Institute of Electrical and Electronic Engineers, Inc., 1978)
2. Y. Takahasi, M. J. Rabins, and D. M. Aslander. Control and Dynamic Systems. (Menlo Park, California: Addison-Wesley, 1970)

APPENDIX A
Suppliers

APPENDIX A

The following companies manufacture instrumentation essential to the successful development of the microcomputer-based temperature controller:

Cromemco, Inc., Mountain View, CA 94040, developed the Z-80 microcomputer system on which the temperature controller hardware is based.

TEI, Inc., Houston, TX 77033, manufactures the TEI RM12 mainframe.

Teletype Corp., Skokie, IL 60077, manufactures the ASR 33 teletype.

Racal-Dana Instruments, Inc., Irvine, CA 92715, manufactures the DANA Model 5900 Digital Multimeter.

Pickles & Trout, Goleta, CA 93017, manufactures the PT-488 interface card.

California Data Corp., Newbury Park, CA 91320, manufactures the DAC-110 digital-analog converter.

Kepco, Inc., Flushing, NY 11352, manufactures the KS 120-5M power supply.

Microsoft, Inc., Bellevue, WA 98004, wrote the compiled version of FORTRAN used in the temperature control routine.

APPENDIX B

Program Listings

1. TEMP
2. 486INT
3. INTDSP
4. NEWIO

```

C      PROGRAM TEMP
      PID TEMPERATURE CONTROL PROGRAM
      INTEGER FORBUF, TLKBUF
      LOGICAL DSTAT
      COMMON /PARAM/DSTAT, FORBUF, TLKBUF
      E=0.
      F=0.
      G=0.
      N=1
      P=0.
      R=0.
      S=0.
      T=.5
      U=10000.
      V=0.
      W=1.
      X=0.
      Y=0.
      Z=0.
      CALL INTPNT
      CALL INTDSP
      CALL DACOUT(Y)
5      CALL DVMIN(X0)
      IF(W.EQ.1) CALL TEMPCR(R,X0)
      IF(W.EQ.2) CALL TEMPCC(R,X0)
      GOTO 100
7      WRITE(1,10)
10     FORMAT(' ', 'GAIN', 1X)
      READ(1,70)G
      WRITE(1,15)
15     FORMAT(' ', 'INTEGRATION TIME', 1X)
      READ(1,70)U
      WRITE(1,20)
20     FORMAT(' ', 'SAMPLING TIME', 1X)
      READ(1,70)T
      WRITE(1,25)
25     FORMAT(' ', 'MAXIMUM POWER', 1X)
      READ(1,70)P
      WRITE(1,30)
30     FORMAT(' ', 'DERIVATIVE TERM', 1X)
      READ(1,70)V
      WRITE(1,60)
60     FORMAT(' ', 'TEMPERATURE SET POINT', 1X)
      READ(1,70)R
      FORMAT(F7.0)
70     Z=0
100    A=G*(1.+T/U+V/T)
      B=C*(1.+2.*V/T)
      C=G*(V/T)
      PMAX=P*4095.0/100.

```

```

150      N=2
        CALL PRINT(X,N)
        CALL DVMIN(X0)
        IF (X.EQ.1) CALL TEMPCR(X,X0)
        IF (X.EQ.2) CALL TEMPCC(X,X0)
        CALL PRINT(X,N)
        D=R-X
        Y=Y+A*D
        IF (Y.LT.0) Y=0
        IF (Y.GT.PMAX) Y=PMAX
        IF (S.EQ.0) CALL DACOUT(Y)
        IF (S.EQ.1) CALL DACOUT(4095.)
        F=E
        E=D
        Y=Y-B*E+C*F
        IF (Y.LT.0) Y=0
        IF (Y.GT.PMAX) Y=PMAX
        IF (DSTAT.GT.0) CALL HPIBIN(DSTAT,FORBUF,G,P,R,S,T,U,V,W,Z)
        IF (Z.GT.0) GOTO 100
        CALL CONCHK(1,CHK)
        IF (INCHK.EQ.1) GOTO 7
        CALL HPIBOT(X,TLKBUF)
        N=1
        CALL PRINT(X,N)
        CALL WAIT(T)
        GOTO 150
      END
      SUBROUTINE HPIBIN(DSTAT,FORBUF,G,P,R,S,T,U,V,W,Z)
      INTEGER FORBUF
      LOGICAL VAR,DSTAT,IN(7)
      TM=2
      N=0
      NDATA=DSTAT
      DSTAT=0
      Z=1
      LOC=FORBUF
      DO 300 I=1,NDATA
        VAR=PEEK(LOC)+1
        DO 5 J=1,7
          IN(J)=PEEK(LOC+J)
          IF (VAR.EQ.4) GOTO 40
          IF (VAR.EQ.7) GOTO 70
          IF (VAR.EQ.16) GOTO 160
          IF (VAR.EQ.18) GOTO 180
          IF (VAR.EQ.19) GOTO 190
          IF (VAR.EQ.20) GOTO 200
          IF (VAR.EQ.21) GOTO 210
          IF (VAR.EQ.22) GOTO 220
          IF (VAR.EQ.23) GOTO 230
        GOTO 300
      CALL PRINT(G,70)
      CALL WAIT(TM)
      CALL PRINT(P,160)
      CALL WAIT(TM)
      CALL PRINT(R,180)
      CALL WAIT(TM)
      CALL PRINT(T,200)
      CALL WAIT(TM)

```

```

CALL PRINT(U,210)
CALL WAIT(TM)
CALL PRINT(V,220)
CALL WAIT(TM)
CALL PRINT(W,230)
CALL WAIT(TM)
GOTO 300
70  DECODE(IN,270)G
GOTO 300
160  DECODE(IN,270)P
GOTO 300
180  DECODE(IN,270)R
GOTO 300
190  DECODE(IN,270)S
IF(S.EQ.1) CALL PRINT(S,190)
CALL WAIT(TM)
GOTO 300
200  DECODE(IN,270)T
GOTO 300
210  DECODE(IN,270)U
GOTO 300
220  DECODE(IN,270)V
GOTO 300
230  DECODE(IN,270)W
GOTO 300
270  FORMAT(F7.0)
300  LOC=LOC+8
RETURN
END
SUBROUTINE HPIDOT(X,TLKBUF)
INTEGER TLKBUF
LOGICAL OUT(6)
ENCODE(OUT,10)X
10  FORMAT(F8.2)
DO 20 I=1,8
20  CALL POKE(TLKBUF+I-1,OUT(I))
RETURN
END
SUBROUTINE DVMIN (X0)
10  IF ((INP(16).AND.16).EQ.0) GOTO 10
CALL OUT(16,Z'FE')
CALL OUT(22,1)
IX=INP(17)
IF (IX.LT.0) IX=IX+256
IX=IX-IX/16*6
X0=IX
CALL OUT(16,Z'FD')
IX=INP(17)
IF (IX.LT.0) IX=IX+256
IX=IX-IX/16*6
X0=X0+100*IX
CALL OUT(16,Z'FB')
IX=INP(17)
IF (IX.LT.0) IX=IX+256
X0=X0+10000*(IX.AND.Z'F')
IF((IX.AND.16).EQ.0) X0=-1*X0
CALL OUT(22,0)
RETURN
END

```



```

SUBROUTINE TEMPCR(X,X0)
X0=X0/1000.
IF(X0.LE.-.1) GOTO 10
IF(X0.LE.1.4) GOTO 20
IF(X0.LE.5.33) GOTO 30
IF(X0.LE.8.14) GOTO 40
IF(X0.LE.12.21) GOTO 50
IF(X0.LE.16.40) GOTO 60
IF(X0.LE.20.64) GOTO 70
IF(X0.LE.25.33) GOTO 80
10 X=99999.
CALL PRINT(X,300)
GOTO 100
20 X=-.30784*X0*X0+25.3078*X0+.000656417
GOTO 100
30 X=-.015145*X0*X0+24.078*X0+1.2
GOTO 100
40 X=-.08937*X0*X0+23.735*X0+.9935
GOTO 100
50 X=-.12203*X0*X0+27.049*X0-12.015
GOTO 100
60 X=-.047458*X0*X0+25.231*X0-.94237
GOTO 100
70 X=-.024678*X0*X0+24.469*X0+5.4492
GOTO 100
80 X=-.0063448*X0*X0+23.171*X0+18.961
100 X0=1000.*X0
RETURN
END
SUBROUTINE TEMPC(X,X0)
X1=X0
X0=1.0E6*X0
IF(X0.LE.-13535.) GOTO 200
IF(X0.LE.-9392.) GOTO 10
IF(X0.LE.-6803.) GOTO 20
IF(X0.LE.-4277.) GOTO 30
IF(X0.LE.-1071.1) GOTO 40
IF(X0.LE.868.6) GOTO 50
IF(X0.LE.2244.6) GOTO 60
IF(X0.LE.3450.8) GOTO 70
IF(X0.LE.4402.2) GOTO 80
IF(X0.LE.5008.8) GOTO 90
IF(X0.LE.5380.6) GOTO 100
IF(X0.LE.5625.1) GOTO 110
IF(X0.LE.5839.4) GOTO 120
IF(X0.LE.6005.2) GOTO 130
IF(X0.LE.6108.5) GOTO 140
IF(X0.LE.6178.6) GOTO 150
IF(X0.LE.6220.1) GOTO 160
IF(X0.LE.6231.4) GOTO 170
IF(X0.LE.6238.0) GOTO 180
GOTO 200
10 X=-1.50553E-7*X0*X0-.0215498*X0+2.86044E2
GOTO 300
20 X=-2.15121E-7*X0*X0-.0227903*X0+2.80073E2
GOTO 300
30 X=-2.91602E-7*X0*X0-.023817*X0+2.76632E2
GOTO 300
40 X=-4.78135E-7*X0*X0-.025328*X0+2.73563E2
GOTO 300

```

```

50      X=-8.13065E-7*X0*X0-.0259243*X0+2.73148E2
      GOTO 300
60      X=-1.24002E-6*X0*X0-.0251936*X0+2.7281E2
      GOTO 300
70      X=-1.99717E-6*X0*X0-.0217596*X0+2.68889E2
      GOTO 300
80      X0=X0-3.0E3
      X=-3.56121E-6*X0*X0-.0322469*X0+1.8524E2
      GOTO 300
90      X0=X0-4.4E3
      X=-6.18989E-6*X0*X0-.0423423*X0+1.33082E2
      GOTO 300
100     X0=X0-5.0E3
      X=-1.0198E-5*X0*X0-.0497968*X0+1.05433E2
      GOTO 300
110     X0=X0-5.3E3
      X=-1.54408E-5*X0*X0-.055159*X0+89.5529
      GOTO 300
120     X0=X0-5.6E3
      X=-2.7043E-5*X0*X0-.0627787*X0+71.5858
      GOTO 300
130     X0=X0-5.8E3
      X=-5.52758E-5*X0*X0-.0707562*X0+57.8613
      GOTO 300
140     X0=X0-6.0E3
      X=-1.26675E-4*X0*X0-.0918664*X0+41.471
      GOTO 300
150     X0=X0-6.1E3
      X=-3.50935E-4*X0*X0-.1113336*X0+30.9484
      GOTO 300
160     X0=X0-6.17E3
      X=-1.33144E-3*X0*X0-.136493*X0+21.2272
      GOTO 300
170     X0=X0-6.22E3
      X=-6.97451E-3*X0*X0-.273145*X0+11.0228
      GOTO 300
180     X0=X0-6.23E3
      X=-2.74622E-2*X0*X0-.325922*X0+7.49877
      GOTO 300
200     X=9999.99
      CALL PRINT(X,300)
300     X0=X1
      RETURN
      END
      SUBROUTINE DACOUT (Y)
      IY=Y
      CALL OUT(24,IY/256)
      CALL OUT(25,IY)
      RETURN
      END
      SUBROUTINE PRINT (A,N)
      IF(N.EQ.1) WRITE(1,1)A
      IF(N.EQ.2) WRITE(1,2)A
      IF(N.EQ.70) WRITE(1,70)A
      IF(N.EQ.160) WRITE(1,160)A
      IF(N.EQ.180) WRITE(1,180)A
      IF(N.EQ.190) WRITE(1,190)
      IF(N.EQ.200) WRITE(1,200)A
      IF(N.EQ.210) WRITE(1,210)A

```

```

IF(N.EQ.220) WRITE(1,220)A
IF(N.EQ.230) WRITE(1,230)A
IF(N.EQ.300) WRITE(1,300)
1  FORMAT(' ',F8.2)
2  FORMAT(' ',F8.2,4X,'SAMPLING')
70  FORMAT(' ',GAIN=',F8.3)
160  FORMAT(' ',POWER=',F8.0)
180  FORMAT(' ',SET POINT=',F8.1)
190  FORMAT(' ',POWER SUPPLY FULL ON')
200  FORMAT(' ',TIME=',F8.2)
210  FORMAT(' ',INTEGRATION=',F8.2)
220  FORMAT(' ',DERIVATIVE=',F8.3)
230  FORMAT(' ',THERMOCOUPLE=',F8.0)
300  FORMAT(' ',THERMOCOUPLE EMF OUT OF RANGE')
RETURN
END
SUBROUTINE WAIT(T)
K=10.0*T
DO 20 J=1,K
DO 10 I=1,4000
10  CONTINUE
20  CONTINUE
RETURN
END
SUBROUTINE CONCHK (INCHK)
INCHK=INP(0).AND.2'40'
IF (INCHK.EQ.2'40') INCHK=1
IF (INCHK.EQ.1) I=INP(1)
RETURN
END

```

B.

```

0002 ;      HPID INTERRUPT PROGRAM
0003 ;
0004 ENTRY  INTPT
0005 EXT    INIT,TALK,LISTN,STADR,STALE,XCTRL
0006 EXT    LSTAT,LEYTE
0007 ;
0008 ;
(007C) 0009 ISRPT: EQU 07CH ;ADDR OF 488 INTERRUPT STATUS PORT
(007D) 0010 CMDPT: FOU 07DH ; " " " COMMAND PORT
(007E) 0011 DATPT: EQU 07EH ; " " " DATA PORT
(007F) 0012 PPORT: EQU 07FH ; " " " PARALLEL POLL RESPONSE PORT
0013 ;
(003B) 0014 MNITR: EQU 003BH ;SYSTEM MONITOR ENTRY ADDRESS
(0452) 0015 CBASIC: EQU 0452H ;CONTROL BASIC ENTRY ADDRESS
(0145) 0016 PRT: EQU 0145H ;CONSOLE PRINT ROUTINE ENTRY ADDRESS
0017 ;
0018 COM    PARAM
0019 ;
0000* (0001) 0020 DSTAT: DS 1 ;CHARACTER COUNTER FOR FORTRAN
0001* (0002) 0021 FBFADD: DS 2 ;ADDRESS OF BEGINNING OF FORTRAN BUFFER
0003* (0002) 0022 TBFADD: DS 2 ;TALK BUFFER ADDRESS STORAGE
0023 ;
0024 DATA
0025 ;
0000" (0050) 0026 FORBUF: DS 50H ;FORTRAN BUFFER AREA FOR INPUTTING DATA
0050" (0001) 0027 CHSTAT: DS 1 ;CHARACTER COUNTER FOR CBASIC
0051" (0008) 0028 TLKBUF: DS 8 ;TALK BUFFER AREA
0029 ;
(0059") 0030 TLKEND: EQU TLKBUF+8 ;END OF TALK BUFFER
(2100) 0031 NMIADD: EQU 2100H ;NMI JUMP ADDRESS LOCATION
0032 ;
0033 REL
0034 ;
0000' 3EFE 0035 NMI: LD A,0FEH ;DISABLE PNT488 NMI
0002' D37C 0036 OUT ISRPT,A
0004' DB7C 0037 IN A,ISRPT
0006' CB67 0038 BIT 4,A ;TEST FOR IFC
0008' CA8800' 0039 JP 2,ICLR
000B' CB5F 0040 BIT 3,A ;TEST FOR ATN
000D' CA7801' 0041 JP 2,XTN

```

```

0010' CB47          0042      BIT      0,A          ;TEST FOR POC
0012' CA6A01'      0043      JP       Z,POCRST
0015' D9            0044      NMIRET: EXX
0016' FDE1          0045      POP      IY
0018' DDE1          0046      POP      IX
001A' 3E03          0047      LD       A,3          ;RE-ENABLE PNT488 NMI
001C' D37C          0048      OUT     ISRPT,A
001E' 08            0049      EX       AF,AF'        ;RESTORE A
001F' ED45          0050      RETN
0021' CD3400'      0051      INTPNT: CALL    PWRUP        ;INITIALIZE RAM AREA
0024' 21AF01'      0052      LD       HL,ADDRS    ; SET UP LISTEN, TALK ADDRESSES
0027' CD0000#      0053      CALL    STADR
002A' 0602          0054      LD       B,2          ;CLEAR 488 INTERFACE, BUT DO NOT SEND IFC
002C' CD0000#      0055      CALL    INIT
002F' 3E03          0056      LD       A,3          ;ENABLE PNT488 NMI
0031' D37C          0057      OUT     ISRPT,A      ;SET B0,B1 HIGH, RESET ALL OTHERS
0033' C9            0058      RET
                    0059      ;
0034' 3E00          0060      PWRUP: LD       A,0
0036' 325000"      0061      LD       (CHSTAT),A
0039' 320000"      0062      LD       (DSTAT),A
003C' 210000"      0063      LD       HL,FORBUF    ;SAVE FORTRAN BUFFER LOCATION
003F' 220100"      0064      LD       (FBFADD),HL  ;IN COMMON (PARAM) AREA
0042' 215100"      0065      LD       HL,TLKBUF    ;SAVE TALK BUFFER LOCATION
0045' 220300"      0066      LD       (TBFADD),HL  ;IN COMMON (PARAM) AREA
0048' 210000'      0067      LD       HL,NMI      ;SAVE NMI JUMP ADDRESS
004B' 220021       0068      LD       (NMIADD),HL
004E' C9            0069      RET
                    0070      ;
                    0071      ;      USER-SUPPLIED JUMP TABLE
                    0072      ;
004F' C36A00'      0073      JTBL: JP       TRGR
0052' C36B00'      0074      JP       DVCL
0055' C38500'      0075      JP       BFL
0058' C38800'      0076      JP       ICLR
005B' C39600'      0077      JP       BRK
005E' C36601'      0078      JP       NLS
0061' C36901'      0079      JP       SREQ
0064' C36A01'      0080      JP       POCRST
0067' C37801'      0081      JP       XTN
                    0082      ;
006A' C9            0083      TRGR: RET
006B' CD2100'      0084      DVCL: CALL    INTPNT

```

006E' 3E00	0085	LD	A,0	
0070' D312	0086	OUT	(18),A	
0072' D313	0087	OUT	(19),A	
0074' F680	0088	OR	80H	
0076' D313	0089	OUT	(19),A	
0078' 97	0090	SUB	A	
0079' D313	0091	OUT	(19),A	
007B' F680	0092	OR	80H	
007D' D313	0093	OUT	(19),A	
007F' 97	0094	SUB	A	
0080' D313	0095	OUT	(19),A	
0082' C35204	0096	JP	CBASIC	
0085' C35204	0097 BFL:	JP	CBASIC	;BUFFER OVERFLOW, GOTO RESTART
0088' 21AF01'	0098 ICLR:	LD	HL,ADDRS	;RESET LISTEN, TALK ADDRESSES
008B' CD2800#	0099	CALL	STADR	
008E' 0602	0100	LD	B,2	
0090' CD2D00#	0101	CALL	INIT	
0093' C31500'	0102	JP	NMIRET	
0096' 3A0000#	0103 BRK:	LD	A,(LSTAT)	
0099' E601	0104	AND	1	
009B' CA6501'	0105	JP	Z,NBRK	
009E' DB7D	0106	IN	A,CMDPT	
00A0' E608	0107	AND	8H	
00A2' CA6501'	0108	JP	Z,NBRK	
00A5' 3A0000#	0109 DECDT:	LD	A,(LBYTE)	
00A8' E6FF	0110	AND	OFFH	
00AA' CAFF00'	0111	JP	Z,DECDTO	
00AD' C6A5	0112	ADD	A,0A5H	
00AF' DAFF00'	0113	JP	C,DECDTO	
00B2' 3AA600#	0114	LD	A,(LBYTE)	
00B5' D641	0115	SUB	A,041H	
00B7' DAFF00'	0116	JP	C,DECDTO	
00BA' F5	0117	PUSH	AF	
00BB' FD210000"	0118	LD	IY,FORBUF	
00BF' 3A0000#	0119	LD	A,(DSTAT)	
00C2' 87	0120	ADD	A,A	
00C3' 87	0121	ADD	A,A	
00C4' 87	0122	ADD	A,A	
00C5' 0600	0123	LD	B,C	
00C7' 4F	0124	LD	C,A	
00C8' FD09	0125	ADD	IY,BC	
00CA' 3E30	0126	LD	A,30H	
00CC' FD7701	0127	LD	(IY+1),A	

;TURN OFF POWER SUPPLY

;CHECK FOR LISTENER ADDRESSED AND ACTIVE
;KEEP ONLY LISTENER BIT
;RETURN IF NOT ADDRESSED
;CHECK FOR LISTENER ACTIVE (ATN HIGH)
;KEEP ONLY ATN BIT

;GET CHARACTER
;CHECK FOR ZERO

;SCALE AND CHECK FOR A THRU Z
;OUT CF RANGE, GO CHECK FOR NUMBER

;SET IY TO FORTRAN BUFFER AREA

;#2
;#4
;#8
; (8 BYTE BUFFER FOR EACH LETTER)

;ADD SCALING FACTOR TO IY

00CF' FD7702	0128	LD	(IX+2),A	
00D2' FD7703	0129	LD	(IX+3),A	
00D5' FD7704	0130	LD	(IX+4),A	
00D8' FD7705	0131	LD	(IX+5),A	
00DB' FD7706	0132	LD	(IX+6),A	
00DE' FD7707	0133	LD	(IX+7),A	
00E1' 3A0000*	0134	LD	A,(DSTAT)	;INCREMENT DATA COUNTER
00E4' 3C	0135	INC	A	
00E5' 320000*	0136	LD	(DSTAT),A	
00E8' F1	0137	POP	AF	
00E9' FD7700	0138	LD	(IX),A	;INSERT KEY CHARACTER INTO STRING FOR
00EC' FD23	0139	INC	IX	; LATER DECODING BY FORTRAN PROGRAM
00EE' DD210020	0140	LD	IX,02000H	;SET IX TO BEGINNING OF VARIABLE SECTION
00F2' 87	0141	ADD	A,A	;COMPUTE RELATIVE ADDRESS OF VARIABLE
00F3' 0600	0142	LD	B,0	;FOR CONTROL BASIC
00F5' 4F	0143	LD	C,A	
00F6' DD09	0144	ADD	IX,BC	
00F8' 97	0145	SUB	A	;CLEAR A
00F9' 325000*	0146	LD	(CHSTAT),A	;CLEAR CHSTAT
00FC' C36401'	0147	JP	DECDT4	;RETURN FOR NEXT CHARACTER
00FF' 3A5000*	0148	LD	A,(CHSTAT)	
0102' D608	0149	SUB	8	;CHECK FOR POSSIBLE OVERFLOW
0104' F21901'	0150	JP	P,DECDT1	
0107' 3AB300#	0151	LD	A,(LBYTE)	
010A' FE0D	0152	CP	0DH	;DON'T PUT CR OR LF IN BUFFER
010C' CA1901'	0153	JP	Z,DECDT1	
010F' FE0A	0154	CP	0AH	
0111' CA1901'	0155	JP	Z,DECDT1	
0114' FD7700	0156	LD	(IX),A	;SAVE CHARACTER IN BUFFER
0117' FD23	0157	INC	IX	
0119' 3A5000*	0158	LD	A,(CHSTAT)	;NO MORE THAN 4 NUMBERS
011C' D604	0159	SUB	4	
011E' F25D01'	0160	JP	P,DECDT3	
0121' 3A0801#	0161	LD	A,(LBYTE)	;GET CHARACTER
0124' E630	0162	AND	30H	;CHECK FOR DECIMAL NUMBER
0126' FE30	0163	CP	30H	
0128' C25D01'	0164	JP	NZ,DECDT3	;IGNORE IF NOT DEC NUMBER
012B' 3A5000*	0165	LD	A,(CHSTAT)	;CHECK FOR FIRST TIME THRU
012E' FE00	0166	CP	0-	
0130' C24201'	0167	JP	NZ,DECDT2	;IF FIRST TIME, STUFF NUMBER INTO SLOT
0133' 3A2201#	0168	LD	A,(LBYTE)	
0136' D630	0169	SUB	A,30H	;CONVERT TO DECIMAL
0138' DD360100	0170	LD	(IX+1),0	;SAVE IN DESIGNATED SLOT
013C' DD7700	0171	LD	(IX),A	
013F' C35D01'	0172	JP	DECDT3	

0142' 3A3401#	0173 DECDT2: LD	A,(LBYTE)	;GET CHARACTER
0145' D630	0174 SUB	30H	;CONVERT TO DEC
0147' DD6601	0175 LD	H,(IX+1)	;GET PREVIOUS INPUT AND MULTIPLY BY TEN
014A' DD6E00	0176 LD	L,(IX)	
014D' 29	0177 ADD	HL,HL	;RESULT *2
014E' E5	0178 PUSH	HL	
014F' 29	0179 ADD	HL,HL	;RESULT *4
0150' 29	0180 ADD	HL,HL	;RESULT *8
0151' D1	0181 POP	DE	
0152' 19	0182 ADD	HL,DE	;RESULT *10
0153' 1600	0183 LD	D,0	;ADD IN NEW LSD
0155' 5F	0184 LD	E,A	
0156' 19	0185 ADD	HL,DE	
0157' DD7401	0186 LD	(IX+1),H	;SAVE BACK IN SLOT
015A' DD7500	0187 LD	(IX),L	
015D' 3A5000"	0188 DECDT3: LD	A,(CHSTAT)	;UPDATE CHSTAT
0160' 3C	0189 INC	A	
0161' 325000"	0190 LD	(CHSTAT),A	
0164' 00	0191 DECDT4: NOP		
0165' C9	0192 NBRK: RET		
0166' C35204	0193 NLS: JP	CBASIC	;GO RESTART IF NO LISTENERS
0169' C9	0194 SREQ: RET		;LET THE CONTROLLER-IN-CHARGE TAKE CARE
	0195 ;	OF THE SERVICE REQUEST	
016A' 21AF01'	0196 POCRST: LD	HL,ADDRS	;RESET LISTEN, TALK ADDRESSES
016D' CD8C00#	0197 CALL	STADR	
0170' 0602	0198 LD	B,2	;CLEAR INTERFACE, BUT DO NOT SEND IFC
0172' CD9100#	0199 CALL	INIT	
0175' C31500'	0200 JP	NMIRET	
0178' 014F00'	0201 XTN: LD	BC,JTBL	
017B' CD0000#	0202 CALL	XCTRL	;LISTEN TO THE COMMANDS AND UPDATE
	0203		; STATE OF INTERFACE
017E' CD0000#	0204 CALL	STATE	;FIND OUT WHAT STATE PNT488 IS IN
0181' FE02	0205 CP	2	;KEEP ONLY LISTENER BIT
0183' C29101'	0206 JP	NZ,CKTLK	
0186' 014F00'	0207 LD	BC,JTBL	;SETUP JUMP TABLE
0189' 3E02	0208 LD	A,2	;NON-BUFFERED OPERATION, END ON EOS
018B' CD0000#	0209 CALL	LISTN	;LISTEN FOR DATA
018E' C31500'	0210 JP	NMIRET	
0191' FE01	0211 CKTLK: CP	1	;KEEP ONLY TALKER BIT
0193' C2A701'	0212 JP	NZ,NTLSN	
0196' 3E01	0213 LD	A,1	
0198' 014F00'	0214 LD	BC,JTBL	


```

019B' 115900"      0215      LD      DE,TLKEND
019E' 215100"      0216      LD      HL,TLKEUF
01A1' CD0000#      0217      CALL    TALK                      ;GO TALK
01A4' C31500'      0218      JP      NMIRET
01A7' 0602         0219 NTLSN: LD      B,2                      ;CLEAR INTERFACE, BUT DO NOT SEND IFC
01A9' CD7301#      0220      CALL    INIT                      ;IGNORE INFORMATION-NOT ADDRESSED
01AC' C31500'      0221      JP      NMIRET
                   0222 ;
                   0223 ; LISTEN, TALK ADDRESSES-
                   0224 ;
01AF' 21           0225 ADDR: DEFB  '!'                      ;LISTEN ADDRESS 1
01B0' 41           0226      DEFB  'A'                      ;TALK ADDRESS 1
01B1' FF           0227      DEFB  OFFH                      ;PARALLEL POLL RESPONSE BYTE (NO RESPONSE)
01B2' FF           0228      DEFB  OFFH                      ;SERIAL POLL RESPONSE BYTE (NO RESPONSE)
01B3' 0A           0229      DEFB  0AH                      ;EOS CHARACTER
01B4' (0000)      0230      END

Errors            0

Program Length   01B4 (436)
Data Length      0059 (89)
COMMON Lengths
/PARAM/          0005 (5)
B.

```

	0002	ENTRY	INTDSP	
	0003	EXT	PRTPTR	
(000B)	0004	DPORT: EQU	11	
(0020)	0005	BUFLEN: EQU	32	;DISPLAY PORT MOD1=21, MOD2=11
0000' F5	0006	INTDSP: PUSH	AF	
0001' C5	0007	PUSH	BC	
0002' D5	0008	PUSH	DE	
0003' 3E00	0009	LD	A,0	
0005' 320000#	0010	LD	(PRTPTR),A	
0008' 0600	0011	LD	B,0	
000A' 0E0B	0012	LD	C,DPORT	
000C' DB16	0013	IN	A,(22)	
000E' 5F	0014	LD	E,A	
000F' F630	0015	OR	30H	
0011' D316	0016	OUT	(22),A	
0013' 3E00	0017	LD	A,0	
0015' 0D	0018	CLRCS: DEC	C	
0016' ED79	0019	OUT	(C),A	
0018' 0C	0020	INC	C	
0019' ED41	0021	OUT	(C),B	
001B' 3C	0022	INC	A	
001C' 3C	0023	INC	A	
001D' 3C	0024	INC	A	
001E' 3C	0025	INC	A	
001F' FE20	0026	CP	BUFLEN	
0021' C21500'	0027	JP	NZ,CLRCS	
0024' 0C	0028	INC	C	
0025' ED59	0029	OUT	(C),E	
0027' D1	0030	POP	DE	
0028' C1	0031	POP	BC	
0029' F1	0032	POP	AF	
002A' C9	0033	RET		
002B' (0000)	0034	END		

Errors 0

Program Length 002B (43)

	0002	ENTRY	\$TTYIN,\$TTYOT,\$CPMBF,\$LINBF,\$LNPTR,\$LUNTB,PRTPTR	
	0003	EXT	\$DRV3	
(000B)	0004	DPORT: EQU	11	;DISPLAY PORT MOD1=21, MOD2=11
(0020)	0005	BUFLN: EQU	32	;WRITE BUFFER LENGTH
(0452)	0006	CBASIC: EQU	0452H	;CONTROL BASIC WARM ENTRY
0000' E5	0007	\$TTYIN: PUSH	HL	
0001' 2A5400"	0008	LD	HL,(\$LNPTR)	
0004' 7E	0009	LD	A,(HL)	
0005' FE0A	0010	CP	A,0AH	
0007' CC1100'	0011	CALL	Z,\$TTY1	
000A' 23	0012	INC	HL	
000B' 7E	0013	LD	A,(HL)	
000C' 225400"	0014	LD	(\$LNPTR),HL	
000F' E1	0015	POP	HL	
0010' C9	0016	RET		
0011' C5	0017	\$TTY1: PUSH	BC	
0012' D5	0018	PUSH	DE	
0013' CD2D00'	0019	CALL	INLN	
0016' 210200"	0020	LD	HL,\$LINBF	
0019' 2B	0021	DEC	HL	
001A' 5E	0022	LD	E,(HL)	
001B' 1600	0023	LD	D,00	
001D' EB	0024	EX	DE,HL	
001E' 19	0025	ADD	HL,DE	
001F' 23	0026	INC	HL	
0020' 360D	0027	LD	(HL),ODH	
0022' 23	0028	INC	HL	
0023' 3E0A	0029	LD	A,0AH	
0025' CD7400'	0030	CALL	\$TTYOT	
0028' 77	0031	LD	(HL),A	
0029' EB	0032	EX	DE,HL	
002A' D1	0033	POP	DE	
002B' C1	0034	POP	BC	
002C' C9	0035	RET		
002D' 1600	0036	INLN: LD	D,0	
002F' 1E00	0037	LD	F,0	
0031' 210200"	0038	LD	HL,\$LINBF	
0034' 2B	0039	DEC	HL	
0035' CD5C00'	0040	INLN1: CALL	KEYIN	

0038' FE03	0041	CP	03H	;COMPARE FOR CONTROL C TO JUMP TO
003A' CA5204	0042	JP	Z,CBASIC	;CONTROL BASIC
003D' FE7F	0043	CP	07FH	;COMPARE FOR RUBOUT TO EDIT
003F' CA5100'	0044	JP	Z,EDIT	
0042' FE0D	0045	CP	0DH	
0044' CA4D00'	0046	JP	Z,INLN2	
0047' 23	0047	INC	HL	
0048' 1C	0048	INC	E	
0049' 77	0049	LD	(HL),A	
004A' C33500'	0050	JP	INLN1	
004D' ED52	0051 INLN2:	SBC	HL,DE	
004F' 73	0052	LD	(HL),E	
0050' C9	0053	RET		
0051' 1D	0054 EDIT:	DEC	E	;DON'T RUBOUT NON-EXISTENT CHARACTERS
0052' E6FF	0055	AND	OFFH	
0054' CA3500'	0056	JP	Z,INLN1	
0057' 2B	0057	DEC	HL	
0058' 1D	0058	DEC	E	
0059' C33500'	0059	JP	INLN1	
005C' CD6300'	0060 KEYIN:	CALL	GBYTE	
005F' CD7400'	0061	CALL	\$TTYOT	
0062' C9	0062	RET		
0063' CD6C00'	0063 GBYTE:	CALL	CHKIN	
0066' CA6300'	0064	JP	Z,GBYTE	
0069' E67F	0065	AND	7FH	
006B' C9	0066	RET		
006C' DB00	0067 CHKIN:	IN	A,(0)	
006E' E640	0068	AND	40H	
0070' C8	0069	RET	Z	
0071' DB01	0070	IN	A,(1)	
0073' C9	0071	RET		
0074' C5	0072 \$TTYOT:	PUSH	BC	
0075' D5	0073	PUSH	DE	
0076' E5	0074	PUSH	HL	
0077' F5	0075	PUSH	AF	
0078' CD8000'	0076	CALL	PRINT	
007B' F1	0077	POP	AF	
007C' E1	0078	POP	HL	
007D' D1	0079	POP	DE	
007E' C1	0080	POP	BC	
007F' C9	0081	RET		
0080' 325600'	0082 PRINT:	LD	(PBYTE),A	;SAVE CHARACTER FOR LATER

0083' FE0A	0083	CP	0AH	;CHECK FOR LF
0085' CCBB00'	0084	CALL	Z,CLRBF	;IF LF, GO CLEAR BUFFER
0088' FE7F	0085	CP	7FH	;CHECK FOR RUBOUT
008A' CCEF00'	0086	CALL	Z,DEL	;IF RUBOUT, GO DELETE LAST CHAP
008D' D620	0087	SUB	20H	;CHECK FOR PRINTABLE CHAR
008F' FA9A00'	0088	JP	M,PRINT1	
0092' 3A5600"	0089	LD	A,(PBYTE)	
0095' D660	0090	SUB	60H	
0097' FC9B00'	0091	CALL	M,UPDTBF	
009A' C9	0092	PRINT1: RET		;UPDATE AND PRINT BUFFER
009B' 3A5700"	0093	UPDTBF: LD	A,(PRTPTR)	
009E' 4F	0094	LD	C,A	
009F' 0600	0095	LD	B,0	
00A1' FE20	0096	CP	BUFLEN	
00A3' CCBB00'	0097	CALL	Z,CLRBF	
00A6' 215800"	0098	LD	HL,PRTBF	
00A9' 09	0099	ADD	HL,BC	
00AA' 3A5600"	0100	LD	A,(PBYTE)	
00AD' 77	0101	LD	(HL),A	
00AE' 3E80	0102	LD	A,80H	
00B0' B6	0103	OR	(HL)	
00B1' 77	0104	LD	(HL),A	
00B2' 0C	0105	INC	C	
00B3' 79	0106	LD	A,C	
00B4' 325700"	0107	LD	(PRTPTR),A	
00B7' CDD100'	0108	CALL	WRTBF	
00BA' C9	0109	RET		
00BB' F5	0110	CLRBF: PUSH	AF	
00BC' 3E00	0111	LD	A,0	
00BE' 325700"	0112	LD	(PRTPTR),A	
00C1' 0E00	0113	LD	C,0	
00C3' 215800"	0114	LD	HL,PRTBF	
00C6' 3E20	0115	LD	A,BUFLEN	
00C8' 3680	0116	CLRBF1: LD	(HL),80H	
00CA' 23	0117	INC	HL	
00CB' 3D	0118	DEC	A	
00CC' C2C80C'	0119	JP	NZ,CLRBF1	
00CF' F1	0120	POP	AF	
00D0' C9	0121	RET		
00D1' 215800"	0122	WRTBF: LD	HL,PRTBF	
00D4' 161F	0123	LD	D,BUFLEN-1	
00D6' 0E0B	0124	LD	C,DPORT	

00D8'	0620	0125	LD	B,BUFLEN
00DA'	DB16	0126	IN	A,(22)
00DC'	5F	0127	LD	E,A
00DD'	F630	0128	OR	30H
00DF'	D316	0129	OUT	(22),A
00E1'	0D	0130	WRTBF1: DEC	C
00E2'	ED51	0131	OUT	(C),D
00E4'	0C	0132	INC	C
00E5'	15	0133	DEC	D
00E6'	EDA3	0134	OUTI	
00E8'	C2E100'	0135	JP	NZ,WRTBF1
00EB'	7B	0136	LD	A,E
00EC'	D316	0137	OUT	(22),A
00EE'	C9	0138	RET	
00EF'	F5	0139	DEL: PUSH	AF
00F0'	3A5700"	0140	LD	A,(PRTPTR)
00F3'	3D	0141	DEC	A
00F4'	FA0601'	0142	JP	M,DEL1
00F7'	215800"	0143	LD	HL,PRTBF
00FA'	325700"	0144	LD	(PRTPTR),A
00FD'	4F	0145	LD	C,A
00FE'	0600	0146	LD	B,0H
0100'	09	0147	ADD	HL,BC
0101'	3680	0148	LD	(HL),80H
0103'	CDD100'	0149	CALL	WRTBF
0106'	F1	0150	DEL1: POP	AF
0107'	C9	0151	RET	
0108'	C6	0152	\$LUNTB: DB	6
0109'	0000#	0153	DW	\$DRV3
010B'	0901#	0154	DW	\$DRV3
010D'	0B01#	0155	DW	\$DRV3
010F'	0D01#	0156	DW	\$DRV3
0111'	0F01#	0157	DW	\$DRV3
		0158	DATA	
0000"	(0002)	0159	\$CPMBF: DS	2
0002"	(0052)	0160	\$LINBF: DS	82
0054"	(0002)	0161	\$LNPTF: DS	2
0056"	(0001)	0162	PBYTE: DS	1
0057"	(0001)	0163	PRTPTR: DS	1
0058"	(0020)	0164	PRTBF: DS	32
0078"	(0000)	0165	END	

Errors

0

Program Length 0113 (275)
Data length 0078 (120)

APPENDIX C

Baud Rates and Addresses

Appendix C

TTY Baud Rates

To change the TTY baud rate, the program located at page 16 (hex 1000) must be changed. This program, written in Control Basic, is listed below.

10	AUTOSTART
20	OUT (0) = 1
30	CALL % 4000
40	END

Line 20 programs the baud rate to 110. Table 1 shows values for different baud rates.

Table 1

Baud Rate	Program
110	OUT(0) = 1
150	OUT(0) = 2
300	OUT(0) = 4
1200	OUT(0) = 8
2400	OUT(0) = 16
4800	OUT(0) = 32
9600	OUT(0) = 64

IEEE-488 Device Addresses

The device address for the temperature controller is preset to 01. This address may be changed by changing lines 225 and 226 in program 488INT. The 488INT program must be reassembled with the new address from Table 2 below. This program must then be relinked with the other modules and a new set of EPROMs programmed.

Table 2

ADDRESS	TALK CODE	LISTEN
00	@	SP
01	A	!
02	B	"
03	C	#
04	D	\$
05	E	%
06	F	&
07	G	' (apos.)
08	H	(
09	I	)
10	J	*
11	K	+
12	L	, (comma)
13	M	-
14	N	.
15	O	/
16	P	0
17	Q	1
18	R	2
19	S	3
20	T	4
21	U	5
22	V	6
23	W	7
24	X	8
25	Y	9
26	Z	:
27	[	;
28		
29	]	=
30		