THE DESIGN OF A DOUBLE-POINT CALIBRATION ROUTINE FOR A MULTICHANNEL HYPERTHERMIA THERMOMETRY SYSTEM

BY

MARY AGNES OZARKA

B.S., University of Illinois, 1983

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1984

Urbana, Illinois

ACKNOWLEDGEMENTS

The author would like to express her deep appreciation to Professor Charles A. Cain and Professor Richard L. Magin for their guidance. A word of thanks is extended to Steve Foster for his assistance with the thermometry hardware. Thank you to all of the graduate students in the lab for their help and suggestions. The efforts of all those who aided in the preparation of this thesis are greatly appreciated. Thank you to my parents and family for their love and assistance. Finally, a special thank you to Stanley Scontras for his support and encouragement throughout this endeavor.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

Hyperthermia is the use of elevated temperatures to treat cancerous tumors. The selective heating of the unhealthy tissue alters the state of the abnormal cells allowing the radiation treatment which follows to be more effective.

A block diagram of a hyperthermia system is shown in Figure 1.1. The thermometry unit is a critical part of this system. The temperature of the tumor and the normal tissue surrounding it must be monitored at several locations. For the best results, the temperature of the abnormal tissue is kept above the minimum treatment level of about $43^{o}C$, but below an upper tolerance level of approximately $50^{o}C$. In addition, the encircling healthy tissue is maintained below $40^{o}C$, as near to normal tissue temperature as possible. The temperature information obtained is used to determine the level of ultrasound or microwave energy the controller sends to the tissue via the applicator.

The thermometry system may also function as a stand-alone unit. As indicated by the dashed line in Figure 1.1, the thermometry unit may connect directly to a terminal, via an RS-232 connection, for the input and output interactions with an operator.

Since the thermometry hardware has varying gain and offset characteristics for each channel, a calibration of the system is required before treatment may begin. The double-point calibration scheme will be discussed and analyzed in this thesis.

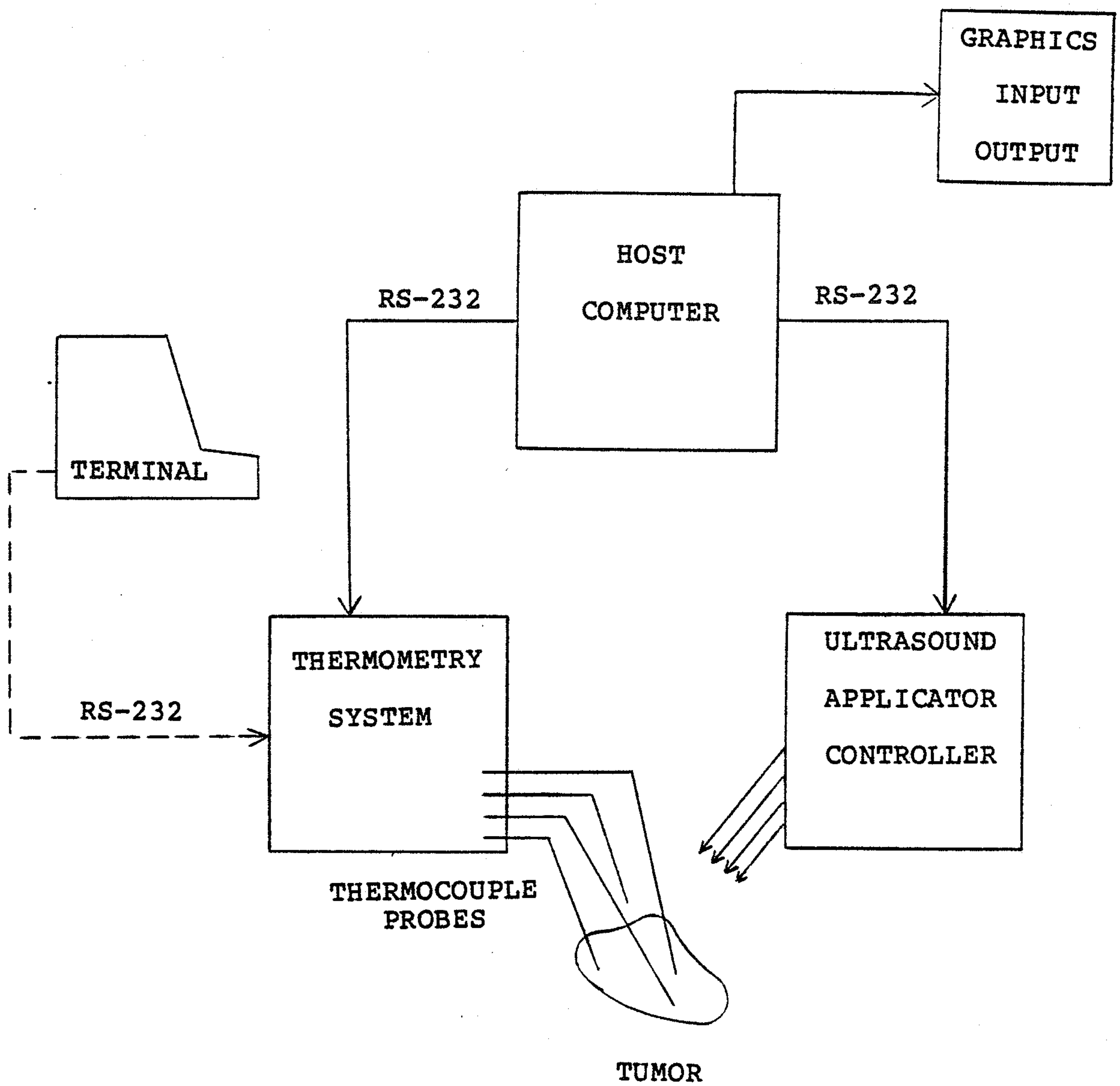


Figure 1.1. Block diagram of a hyperthermia system.

CHAPTER II

THERMOMETRY HARDWARE OVERVIEW

The thermometry system is based on the Intel 8031 microcontroller as designed by Alfred Gharakhani [1]. A block diagram of the system is displayed in Figure 2.1. The temperatures are obtained through the use of type T (copper-constantan) thermocouple probes. The probes generate voltages based on the temperature at the copper-constantan junctions. Then, the voltages are amplified and one channel is selected by the analog multiplexer. At this time, the error detection circuitry signals the 8031 if the probe for the channel chosen is broken or missing. Otherwise, the voltage from the analog multiplexer is scaled so that the analog-to-digital (A/D) converter functions correctly. The twelve bits of digitized voltage are sent to the 8031 for further software processing. The temperature corresponding to the particular voltage is transmitted to the terminal via the RS-232 connection in appropriate ASCII code.

The conversion between thermocouple voltage and temperature requires either a look-up table or an equation which models the table. Complications encountered in the look-up table method necessitated the use of a polynomial equation to model the conversion. In addition, each channel of hardware may have a different gain and offset which requires a linear fit. These two situations are dealt with by calibrating the thermometry system to a precision thermometer before any readings are made.

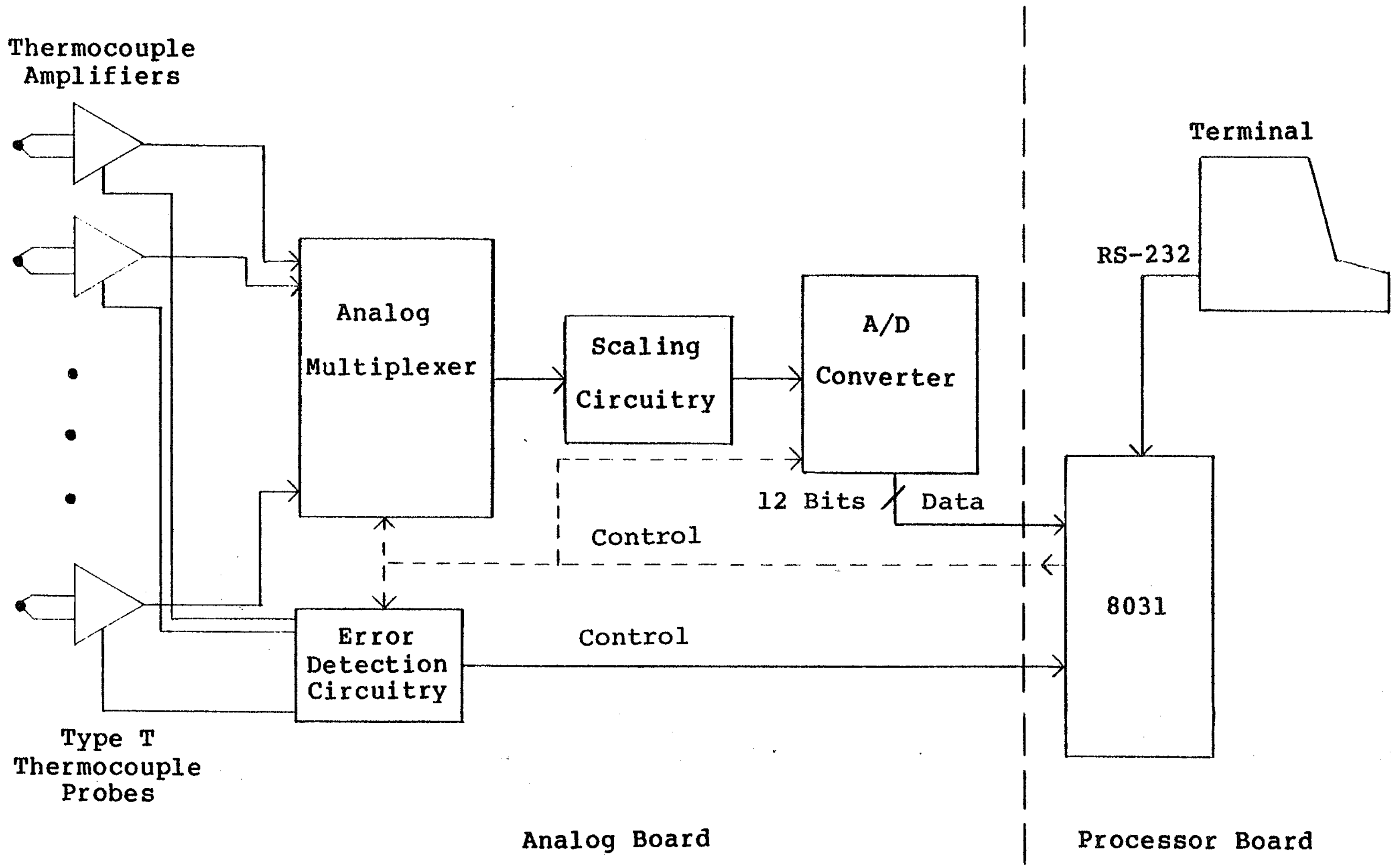


Figure 2.1. Block diagram of the thermometry unit (from [1]).

To improve the thermometry system and its calibration, several changes have been made to Gharakhani's design. A summary of the major modifications is described below.

The type J (iron-constantan) thermocouple probes and the corresponding thermocouple amplifier chips, which Gharakhani used, were replaced by type T (copper-constantan) thermocouple probes and amplifiers. The type T probes are more resistant to corrosion in moist surroundings [2], whereas the type J probes will rust if placed in a water bath for an extended length of time.

The valid temperature range of $0^{\circ}C$ to $51.1^{\circ}C$ was changed to $20^{\circ}C$ to $60^{\circ}C$, and the least significant digit changed from $0.1^{\circ}C$ to $0.01^{\circ}C$. It will turn out that the calibration calculations work for temperatures in the range $0^{\circ}C$ to $99.99^{\circ}C$, but the thermocouple voltage to temperature conversion is accurate only for the range $20^{\circ}C$ to $60^{\circ}C$.

Since the temperature output is to the hundredth place, the look-up table scheme for the conversion between thermocouple voltage and temperature could no longer be used. For the range $0^{\circ}C$ to $51.1^{\circ}C$, the look-up table consisted of 512 memory locations for each of the three numeric places (stored as ASCII characters). That is equivalent to 1,536 locations. For output temperatures to the hundredth place, in the range $20^{\circ}C$ to $60^{\circ}C$, the look-up table scheme would require 4001 * 4 = 16,004 bytes, a considerable amount of memory. Moreover, if the temperature was out of the range of values in the look-up table, an underfow or overflow error would occur. The extended range of $0^{\circ}C$ to $99.99^{\circ}C$ would require 10,000 * 4 = 40,000 memory locations. Obviously, it would be wasteful to attempt this method. Fortunately, the look-up table may be replaced by a polynomial equation approximating the conversion. This scheme will be discussed further in Chapter III.

Another significant change is that all twelve bits of data from the analog-to-digital converter are used rather than only nine bits. This will increase the accuracy of the displayed temperatures.

Furthermore, the two calibration points are user selectable rather than set at 25°C and 35°C.

These changes will result in a more accurate and "user-friendly" system.

CHAPTER III

THERMOCOUPLE CHARACTERISTICS

3.1. Nonlinearity

Thermocouples have many characteristics advantageous for ultrasound hyperthermia temperature measurement. They are sturdy, small in size, and respond quickly; however, the thermocouple output voltage is nonlinear with respect to temperature [3]. A plot of the type T thermocouple voltage versus temperature, for the range $20^{o}C$ to $60^{o}C$, is shown in Figure 3.1. This is a graphical representation of the thermocouple reference table from [4], which is reproduced in Table 3.1.

As can be seen, the nonlinearity does not appear to be severe. Nevertheless, further calculations will indicate the advantage of taking the nonlinearity into account. Note also that the design will actually have one hundred points between the ones shown in Figure 3.1, since the resolution is to $0.01^{o}C$.

For the double-point calibration and correction of the thermometry system, four main equations are required. Two of these are the nonlinear conversions between thermocouple voltage and temperature. The other two involve a simple linear model of the system hardware. The thermocouple conversion equations are based on the type T standard reference table displayed in Table 3.1. These equations will be discussed in the following two subsections. The other equations and the calibration scheme itself will be presented in Chapter IV.

3.1.1. Conversion of Thermocouple Voltage to Temperature

The conversion from thermocouple voltage to temperature, for

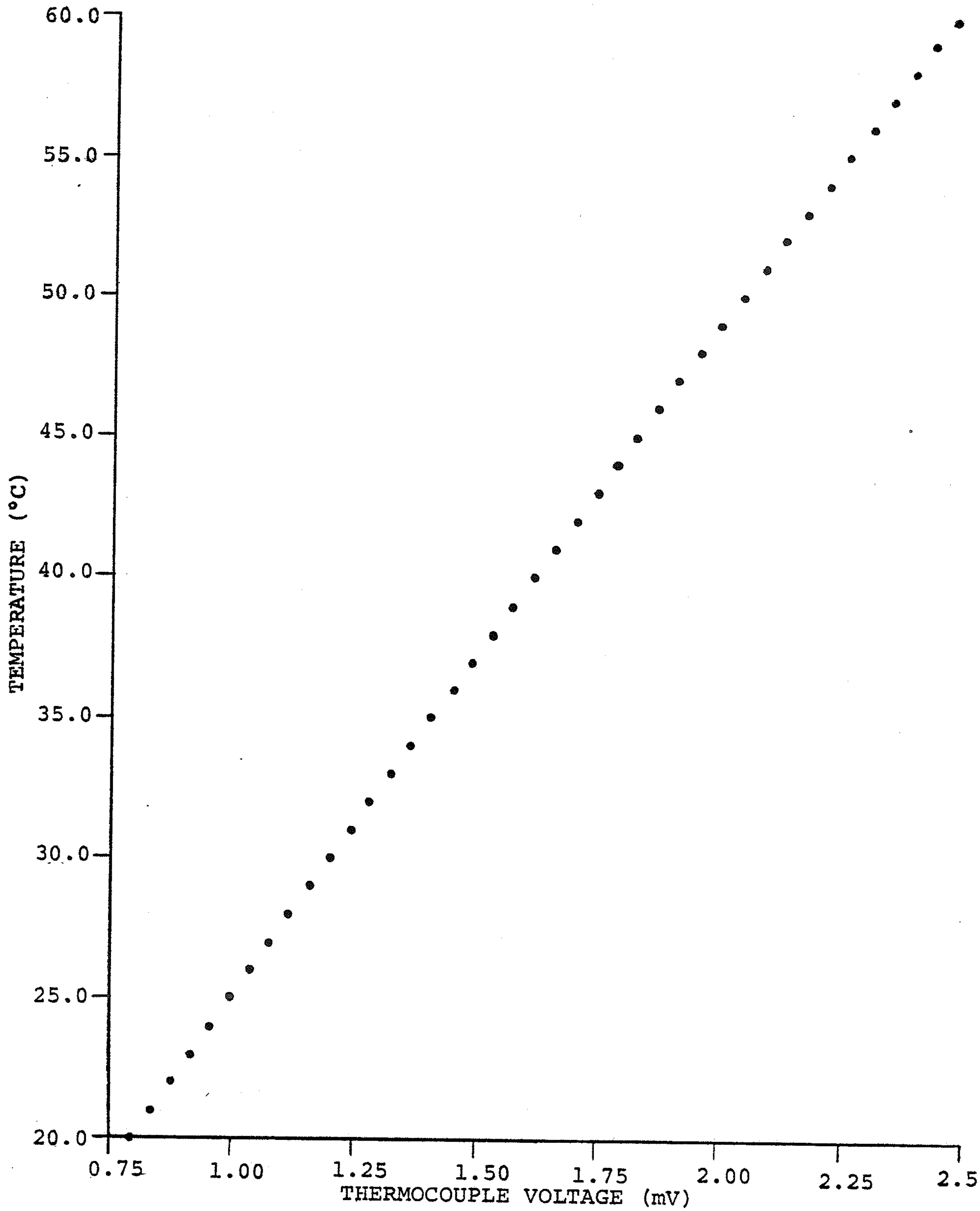


Figure 3.1. Temperature - voltage correspondence for type T thermocouples.

Table 3.1. Temperature - Voltage Correspondence for Type T Thermocouples (from [4]).

Temperature (^{o}C)	Voltage (mV)	Temperature (^{o}C)	Voltage (mV)
20	0.789	40	1.611
21	0.830	41	1.653
22	0.870	42	1.695
23	0.911	43	1.738
24	0.951	44	1.780
25	0.992	45	1.822
26	1.032	46	1.865
27	1.073	47	1.907
28	1.114	48	1.950
29	1.155	49	1.992
30	1.196	50	2.035
31	1.237	51	2.078
32	1.279	52	2.121
33	1.320	53	2.164
34	1.361	54	2.207
35	1.403	55	2.250
36	1.444	56	2.294
37	1.486	57	2.337
38	1.528	58	2.380
39	1.569	59	2.424
40	1.611	60	2.467

$20^{o}C$ to $60^{o}C$, was calculated using a best polynomial fit. The resulting equations are of the form

$$T = A3 * V^{3} + A2 * V^{2} + A1 * V + A0 \qquad (3.1)$$

where A3, A2, A1, and A0 are real constants, which may be zero, T is temperature in degrees Celsius, and V is voltage in millivolts. The constants are shown in Table 3.2 for zero order, linear, quadratic, and cubic fits. The zero order equation is obviously not a good fit, but it is included for the purpose of comparison.

Also shown in Table 3.2 are the mean square error (or variance) and the root mean square (RMS) error (or standard deviation) as defined by

$$\text{Mean Square} = [\Sigma(T - Ti)^{2}] / N \qquad (3.2)$$

and

$$RMS = [[\Sigma(T - Ti)^{2}] / N]^{1/2} \qquad (3.3)$$

The value of T_i is the ideal temperature from Table 3.1, and its corresponding voltage is V. T is the temperature calculated from Equation (3.1), using the value of voltage, V, and the appropriate values of the constants, for each particular fit in Table 3.2. N is the number of data points, a total of forty-one, in Table 3.1.

To determine which equation to use, the RMS error values were compared. As the order of the equation increases, the RMS error decreases. Notice that there is an improvement of $0.1197^{o}C$ in the RMS error between the linear and quadratic equations. However, there is only a difference of $0.0007^{o}C$ in the RMS error between the quadratic and cubic approximations. In other words, the RMS error accompanying the second and third order fits is smaller than

Table 3.2. Polynomial Fits for the Conversion from Thermocouple Voltage to Temperature.

	ORDER OF FIT			
	ZERO	LINEAR	QUADRATIC	CUBIC
A3	0	0	0	0.0329830244
A2	0	0	-0.577516124	-0.7384273168
A1	0	23.8349418	25.7112919	25.95831205
A0	40	1.45134171	0.0696651191	-0.048435657
MEAN SQUARE ($^{\circ}$C)2	140	0.0162	0.000057891	0.000048047
RMS ($^{\circ}$C)	11.8322	0.1274	0.0076	0.0069
MAXIMUM DEVIATION ($^{\circ}$C)	+20	+0.2571	-0.0154	+0.0133

the desired resolution of 0.01°C.

The values in Table 3.1 have finite precision and, thus, an inherent round-off error. The precision in the voltage values is 1 μV and the theoretical round-off error [5] is

$$1\ \mu V\ /\ [(12^{1/2}] = 0.289\ \mu V. \qquad (3.4)$$

This corresponds to

$$0.289\ \mu V\ /\ (40.44\ \mu V/^{o}C) = 0.0071^{o}C \qquad (3.5)$$

where 40.44 μV/°C is the input offset voltage per degree of temperature change (from [3]). This value of 0.0071°C is the theoretical value of the RMS error, and it indicates the smallest error that can be statistically realized. As seen in Table 3.2, the higher order fits approach this value and, in fact, the cubic fit has an RMS error that is smaller than 0.0071°C. This is misleading, though, since as N approaches infinity, the RMS error has to approach 0.0071°C.

Another factor to consider is the maximum deviation, also shown in Table 3.2. The calculated and ideal temperatures were compared for every degree between 20°C and 60°C. Again, the magnitudes of the deviation for the quadratic and cubic fits are close in value.

There are trade-offs in choosing which equation is the best to use for this application. The cubic equation has slightly smaller errors, but the extra calculations and several special routines required were not considered to be worth the small gain in accuracy. The equation is then

$$T = -0.577516124 * V^2 + 25.7112919 * V + 0.0696651191 \qquad (3.6)$$

where T is the temperature in degrees Celsius and V is the voltage in millivolts.

3.1.2. Conversion of Temperature to Thermocouple Voltage

The best polynomial fit for the conversion from temperature to thermocouple voltage is of the form

$$V = B3 * T^3 + B2 * T^2 + B1 * B0 \qquad (3.7)$$

where B3, B2, B1, and B0 are real constants, V is voltage in millivolts, and T is temperature in degrees Celsius. The constants are shown in Table 3.3 for zero order, linear, quadratic, and cubic fits. Also displayed are the mean square error, RMS error, and maximum deviation.

The round-off error due to the finite precision of Table 3.1 is 0.289 μV, as stated in Equation (3.4). For this conversion, the RMS errors for the quadratic and cubic fits are exactly equal. In addition, the quadratic fit has a smaller maximum deviation for the points examined. For these reasons, the quadratic fit has been chosen again. This equation is

$$V = 0.0000426489542 * T^2 + 0.0385384869 * T + 0.00157115981 \qquad (3.8)$$

where V is voltage in millivolts and T is temperature in degrees Celsius.

Equations (3.6) and (3.8) are shown with constants that have several significant digits. Naturally, when these equations are implemented on the eight bit Intel 8031 microcontroller, some of the accuracy will be lost. A further discussion on this matter appears in Chapter IV.

Table 3.3. Polynomial Fits for the Conversion from Temperature to Thermocouple Voltage.

	ORDER OF FIT			
	ZERO	LINEAR	QUADRATIC	CUBIC
B3	0	0	0	6.190*10-13
B2	0	0	0.0000426489542	4.2606*10-5
B1	0	0.0419503489	0.0385384869	0.038541969
B0	1.611	-0.0606968815	0.00157115981	0.001502803
MEAN SQUARE (μV)2	243.8693	28.5543	0.0848	0.0848
RMS (μV)	493.8312	5.3436	0.2912	0.2912
MAXIMUM DEVIATION (μV)	+856	-10.6899	-0.5265	-0.5334

3.2. Self-Heating Effects

The tip of the type T thermocouple is a junction of two metals, copper and constantan. At the point of contact, a potential difference exists which is a repeatable function of temperature [3]. In order to relate this voltage to temperature, a known reference temperature is necessary.

Usually, standard thermocouple voltage to temperature conversion tables assume a reference junction at $0^{o}C$. The thermocouple amplifier used in the thermometry system, the AD595, has an effective reference junction which is not at $0^{o}C$. This junction is shown in Figure 3.2 at point A. Here, a second thermocouple junction exists which must be kept at the same temperature as the AD595. If the temperature is maintained, the cold junction compensation feature of the AD595 will work properly. Internally, the compensation voltage (V_o in Figure 3.2) adds to the thermocouple voltage an amount proportional to the difference between $0^{o}C$ and the AD595 temperature. Thus, the output voltage will be equivalent to a reading from a thermocouple referenced to an ice bath [3].

This design seems to account for the self-heating effects of the thermocouple amplifier chip; however, there is a problem. The specification for the AD595 [3] indicates that the input offset change with respect to temperature is typically

$$dV_o/dT = 40.44\ \mu V/^{o}C \qquad (3.9)$$

at $25^{o}C$. This implies that, ideally, the change in thermocouple voltage with respect to temperature should also be $40.44\ \mu V/^{o}C$ for the compensation to function correctly. The change in

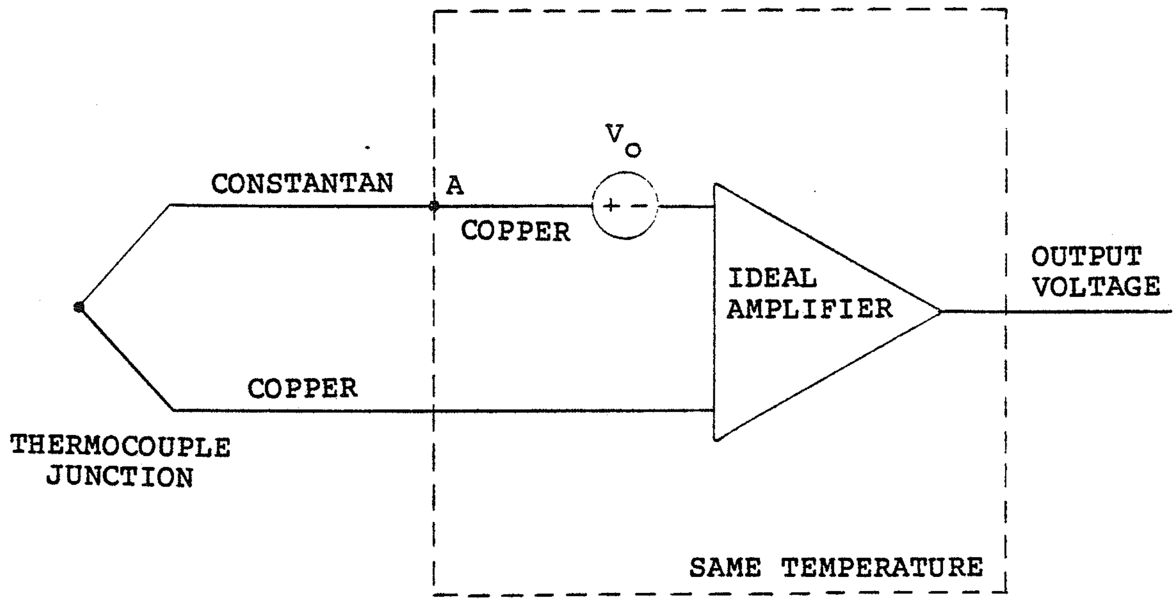


Figure 3.2. Thermocouple amplifier showing the effective reference junction at A.

thermocouple voltage with respect to temperature may be calculated by finding the derivative of Equation (3.8). That is

$$dV_{th}/dT = 0.0852979084 * T + 38.5384869 \quad \mu V/^{o}C. \qquad (3.10)$$

Table 3.4 shows the voltage change as the ambient temperature increases. Here, compensation and deviation are defined as

$$\text{Compensation} = dV_{th}/dT - 40.44 \ \mu V/^{o}C \qquad (3.11)$$

and

$$\text{Deviation} = (dV_{th}/dT - 40.44) \ / \ (dV_{th}/dT). \qquad (3.12)$$

Table 3.4. Ambient Temperature Effects.

Ambient Temperature (^{o}C)	dV_{th}/dT ($\mu V/^{o}C$)	Compensation ($\mu V/^{o}C$)	Deviation ($^{o}C/^{o}C$)
20	40.24	-0.20	-0.005
22.3	40.44	0.00	0.000
25	40.67	+0.23	+0.006
30	41.1	+0.66	+0.016
35	41.52	+1.08	+0.026
40	41.95	+1.51	+0.036

The cold junction compensation of the AD595 compensates incorrectly for ambient temperatures other that 22.3^{o}C. Obviously, if the chip temperature is allowed to increase much above 25^{o}C, some error will result. Notice that the dV_{o}/dT and dV_{th}/dT do not match at 25^{o}C. This is because the AD595 is intended to operate with type K thermocouples (the copper is replaced by a nickel-chromium alloy in the ratio 90:10) and produce zero compensation at 25^{o}C.

Another specification of the AD595, related to self-heating effects, is the stability versus temperature. This value is rated at $\pm 0.025^{o}C/^{o}C$ [3]. Since type T rather than type K thermocouples are being used, the $+0.006^{o}C/^{o}C$ deviation, taken from Table 3.4, should be added to the $\pm 0.025^{o}C/^{o}C$ rating at $25^{o}C$. For example, suppose a reading of $37^{o}C$ were obtained when the ambient temperature was $25^{o}C$. Then, if the ambient temperature increases by one degree to $26^{o}C$, the uncertainty in the reading will be $-0.02^{o}C$ to $+0.03^{o}C$. This implies that the reading may be between $36.98^{o}C$ and $37.03^{o}C$ just due to the increase in ambient temperature by one degree.

At other ambient temperatures, the results get worse. For instance, at $30^{o}C$, the error is $\pm 0.025^{o}C/^{o}C$, plus the value $+0.016^{o}C/^{o}C$ from Table 3.4. Thus, the uncertainty is $-0.01^{o}C$ to $+0.04^{o}C$, and the reading may be between $36.99^{o}C$ and $37.04^{o}C$, just because of the ambient temperature change.

From this discussion, it is apparent that some means of controlling the ambient AD595 temperature is advisable.

CHAPTER IV

THE CALIBRATION SCHEME

4.1. Description and Procedure

Since each channel of the thermometry system has different thermocouple amplifier chips with slightly different characteristics, each probe, if uncalibrated, may output a different temperature. The purpose of the calibration program is to correct hardware deviations.

From the user's point of view, the double - point calibration of the thermometry system is a quick and easy process. The steps are:

1. Place the probes to be used in a water bath (or oven) of constant temperature, T1.
2. Enter temperature T1 on the terminal.
3. Place the probes in a bath of constant temperature, T2, of greater value than T1.
4. Enter temperature T2 on the terminal.

Temperatures T1 and T2 may take on any value in the range from 20°C to 60°C. Note that the calculations in the program work for temperatures between 0°C and 99.99°C, but the nonlinear conversion equations were based on temperatures in the range 20°C to 60°C. The speed of the calibration process is limited only by the input and output interactions and the time it takes for the water bath (or oven) to stabilize at T1 and T2. After the four steps, the system is calibrated. From then on, every time a reading is made, the output temperature is modified to the correct value before being displayed.

An overview of the calibration and correction process is shown in Figure 4.1. In addition, a detailed flow chart is presented in Appendix A. After the operator places the probes in temperature T1 and enters its value on the terminal, all of the probes are read. The digitized thermocouple output voltages are stored in memory. The same process of reading and storing is repeated when the probes are in temperature T2.

Next, entered temperatures T1 and T2 are converted to voltages using a form of equation (3.8). The equation actually used in the program is

$$V = 426.5 * T^2 + 38{,}538{,}486 * T + 157{,}115{,}981. \qquad (4.1)$$

Here, V is in 10^{-14} volts and T is in $10^{2\,o}C$. Notice that the constants have been multiplied by 10^{11} and that some of the significant digits have been dropped. This makes calculation easier by eliminating most of the fractional parts. Actually, in the program, the value 4265 is multiplied by T^2 and then the product is divided by ten. Since all of the arithmetic operations on the Intel 8031 involve one byte long numbers, several special subroutines were written to perform multiple byte multiplication and division. In addition, some values are multiplied by factors of 256 to avoid division resulting in a fractional number. It is necessary to keep track of these adjustments so that the reverse process can be performed later to get the true values. The comments in the program listing in Appendix B explain the procedure.

The next step in the calibration, as shown in Figure 4.1, is the approximation of the system hardware by a linear fit. That

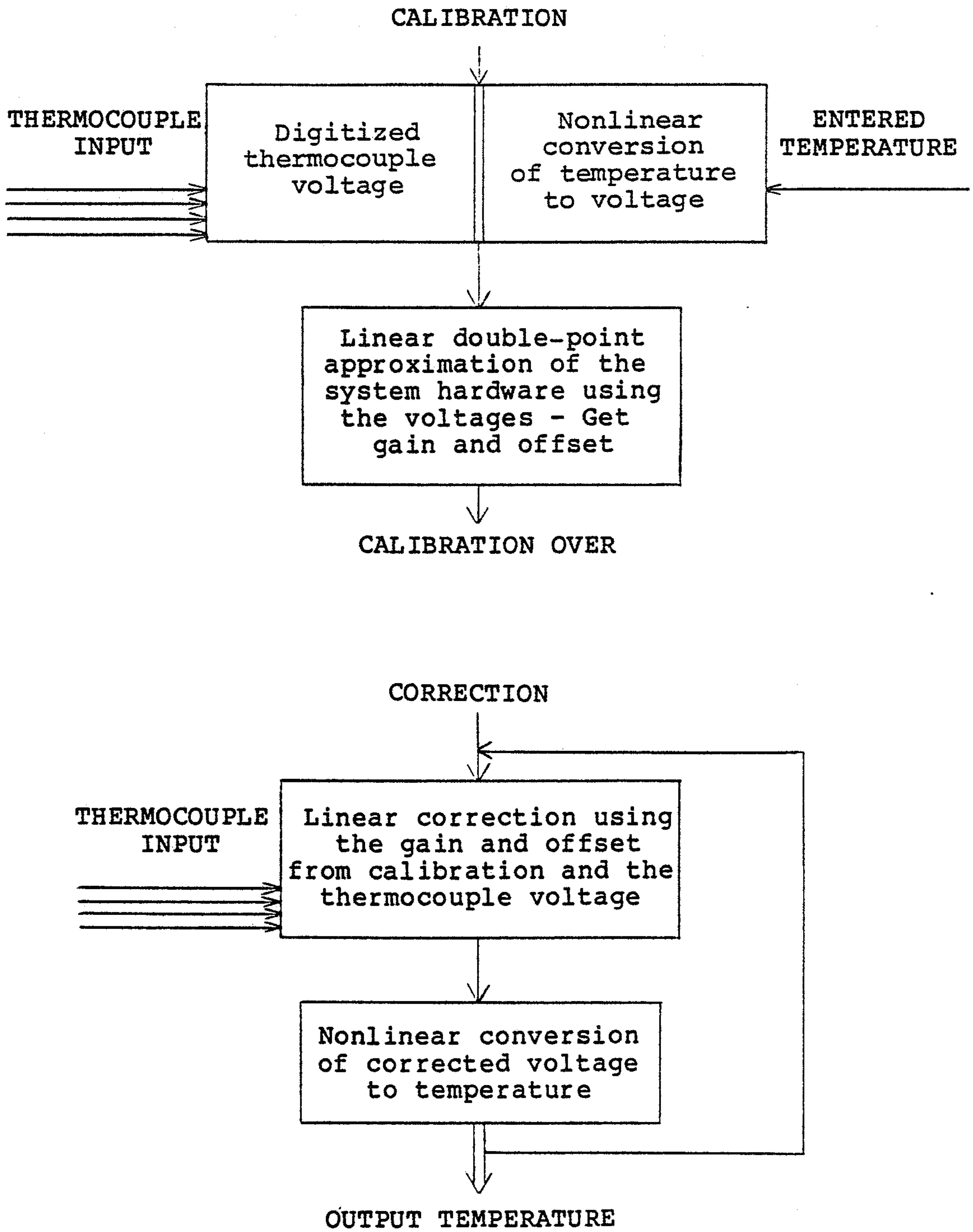


Figure 4.1. Overall calibration and correction scheme.

is, the gain and offset for each channel are found using

$$Y = M * X + B \qquad (4.2)$$

where X is the known voltage obtained from the entered temperature, Y is the digitized thermocouple voltage, M is the gain, and B is the offset. Once calculated, the gain and offset for each channel are stored in memory and, at this point, the calibration is complete.

From now on, when the probes are sampled, Y, the digitized voltage obtained is corrected using the stored gain, the stored offset, and the linear equation

$$X = (Y - B) / M. \qquad (4.3)$$

The corrected voltage, X, is then converted to a temperature using a form of equation (3.6). Again, due to the limitations of the microprocessor, the actual equation used is

$$T = -27.27 * V^2 + 7,237,085 * V + 116,878,675. \qquad (4.4)$$

Here, T is temperature in $(100 * 256^3)$ degrees Celsius and V is voltage in $(10^{-14} * 256^3)$ volts. These units may seem odd, but the equation was put in this form to make the calculations easier.

Any of the calibrated probes may be chosen for a reading as many times as desired. However, if the probe corresponding to a particular channel was missing or broken during the calibration, then it will remain uncalibrated unless the calibration procedure is performed again. The flow chart in Appendix A shows some of the error checking present in the routine. Also, see the listing of the calibration program in Appendix B for more details [6].

4.2. An Example of Calibration

The example which follows is intended to highlight and clarify the flow chart in Appendix A. The calibration of only one probe will be considered, but the extension to any number of probes is straightforward. Notice that truncation is used in the calculations.

- Let the temperature, T1, of the first water bath and the twelve bit digitized voltage, Y1, of the probe in this water bath be:

  $T1 = 37.06^{o}C$

  Y1 = 05DB in hexadecimal = 1499 in decimal.

- Let the temperature, T2, of the second water bath and the twelve bit digitized voltage, Y2, of the probe in this water bath be:

  $T2 = 50.04^{o}C$

  Y2 = 07F9 in hexadecimal = 2041 in decimal.

- Temperatures (T1 * 100) and (T2 * 100) are converted to voltages V1 and V2 using equation (4.1). Then, they are each divided by 256^3, obtaining X1 and X2 respectively, so that the numbers are easier to process in the 8031. However, the results are still accurate.

  $V1 = 1.488384*10^{11}$ $\quad\quad$ X1 = 8871

  $V2 = 2.036832*10^{11}$ $\quad\quad$ X2 = 12,140

- Next, the points (X1,Y1) and (X2,Y2) are fitted to a line using equation(4.2).

  1499 = M * 8871 + B

  2041 = M * 12,140 + B

- The gain, $M = m * 256^2$ is found.

$$m = (Y2 - Y1) / (X2 - X1) = 542 / 3269$$

Since (542/3269) will be a fraction, the numerator is multiplied by 256^2, then the division is performed.

$$M = m * 256^2 = (542*256^2) / 3269 = 10,865$$

- The offset, $B = b * 256^2$ is found.

$$b = Y1 - M * X1$$

Since M is actually $(m*256^2)$, we need $(Y1*256^2)$ to get $B = (b*256^2)$.

$$B = Y1 * 256^2 - M * X1 = 1,855,049.$$

- The gain and offset are stored in memory. Three bytes of memory for gain and five bytes of memory for offset are reserved for each channel.
- The calibration is over.

- The probe is placed in a water bath of temperature T. The digitized voltage, Y, of the probe is read.

Y = 06DB in hexadecimal = 1755 in decimal

$T = 43.24^{\circ}C$ = the desired output temperature

- The correction to Y, the voltage, is made using equation (4.3) with the extra factor of 256^2 included.

$$X = (Y * 256^2 - B) / M = 10,415$$

- In this example, X is playing the role of the corrected thermocouple voltage, V. This value of voltage is used in equation (4.4) to convert to temperature, T. The result is divided by $(100 * 256^3)$ to get the actual temperature.

$$T = 7.253308*10^{10}$$

Temperature = $43.23^{\circ}C$

- The temperature is transmitted to the terminal as ASCII characters for display.

The truncation of intermediate results during the calculations and the other manipulation of the values cause the final temperature to have an error of 0.01°C.

CHAPTER V

EXPERIMENTAL DATA AND ANALYSIS

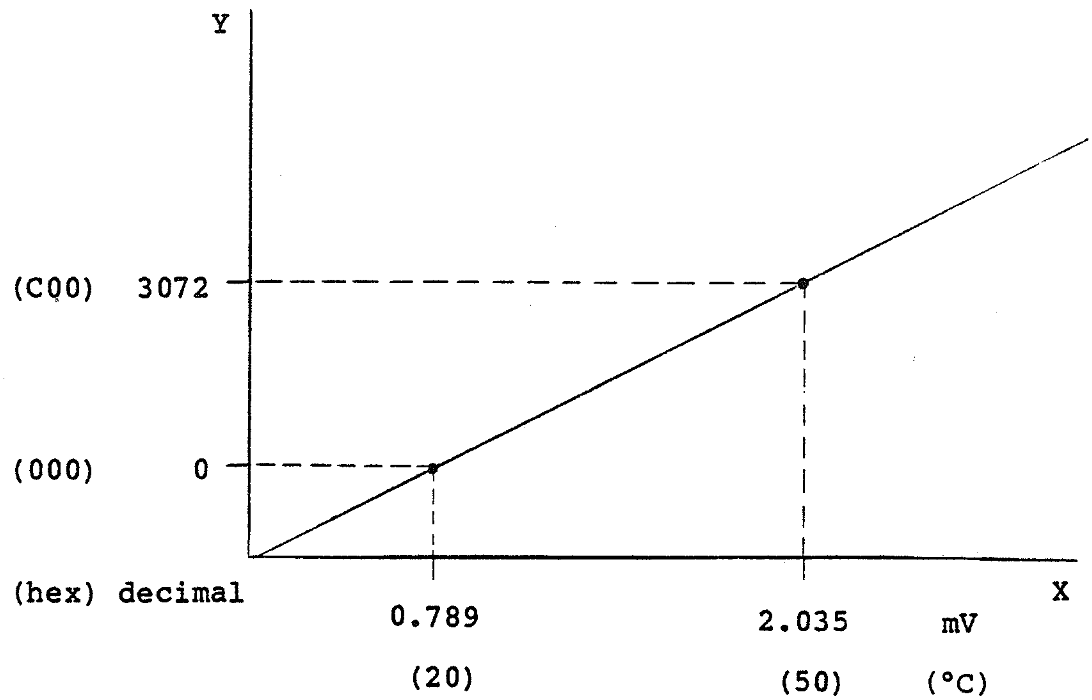
5.1. Overview

Several experimental tests of the calibration routine have been performed. These tests were designed to study the properties of accuracy, precision, stability, and repeatability. Due to hardware failure of Gharakhani's thermometry system, the input and output parts of the program were modified and the tests were performed on URI THERM-X's version of the system. It will be shown that most of the error that occurs is due to analog hardware variations. Once the calibration is complete, the correction routine will always output the same temperature when given a particular voltage.

5.2. Software Test

A test of the software was devised which eliminated the analog hardware. The calibration points were chosen to be 20°C and 50°C. Then, the twelve bit digitized voltages corresponding to these temperatures, 000 and C00 (in hexadecimal), respectively, were used instead of the usual thermocouple output voltage obtained from the analog circuitry. This means that all parts of the program were run with known numerical values.

Figure 5.1 shows the linear fit involved. On the x-axis are the temperatures, in degrees Celsius, and their equivalent voltages, in millivolts, obtained from Table 3.1. The y-axis shows the expected digitized voltage from the analog-to-digital converter corresponding to a particular temperature. The digitized voltages for various temperatures in the range 20°C to



M = (3072 - 0) / (2.035 - 0.789) = 2465.489
B=-1945.271

Figure 5.1. The linear fit of the calibration software test.

$60^{o}C$ were calculated by hand using the method in the calibration routine. Then, the program was executed. The results are shown in Table 5.1. The deviation displayed is the difference between the temperature obtained from the program and the ideal temperature.

There is one deviation of $0.02^{o}C$, but the rest of the time it is less. This error is directly related to the calculations in the program. As discussed in Chapter IV, the microprocessor's limitation of eight bit arithmetic operation causes unusual manipulations of intermediate values. With this in mind, the error involved is small.

In review, Table 5.1 and the discussion above indicate that an error of up to $0.02^{o}C$ may be attributed to the calibration calculations in the program. Errors greater than this are hardware related.

5.3. Test Setup

All of the other tests made use of a test setup as shown in Figure 5.2. The probes were placed in a water bath of constant temperature. The accurate temperature of the water was measured using the platinum resistance thermometer (PRT). The printer gave a permanent record of the temperatures of the probes for any chosen interval of time.

An important consideration for obtaining reliable data was to make sure that the temperature of the water bath was stable, as indicated by a stable readout on the PRT. The thermocouple probes will react much quicker to a temperature change than will the PRT. Therefore, if the temperature indicated by the PRT is still changing when a reading is made, erroneous results will occur.

Table 5.1. Results of Calibration Program Check.

Ideal Temperature (°C)	Temperature from Hand Calculation (°C)	Temperature from Program (°C)	Deviation (°C)
20	20.00	20.00	0.00
23	23.00	23.01	0.01
25	25.00	25.01	0.01
28	27.99	28.00	0.00
30	29.99	29.99	0.01
33	33.00	33.00	0.00
35	34.99	35.00	0.00
38	38.00	38.01	0.01
40	39.99	39.99	0.01
43	43.00	43.01	0.01
45	44.99	44.99	0.01
48	48.00	48.01	0.01
50	50.00	50.00	0.00
52	52.00	52.00	0.00
54	53.99	54.00	0.00
58	57.98	57.98	0.02

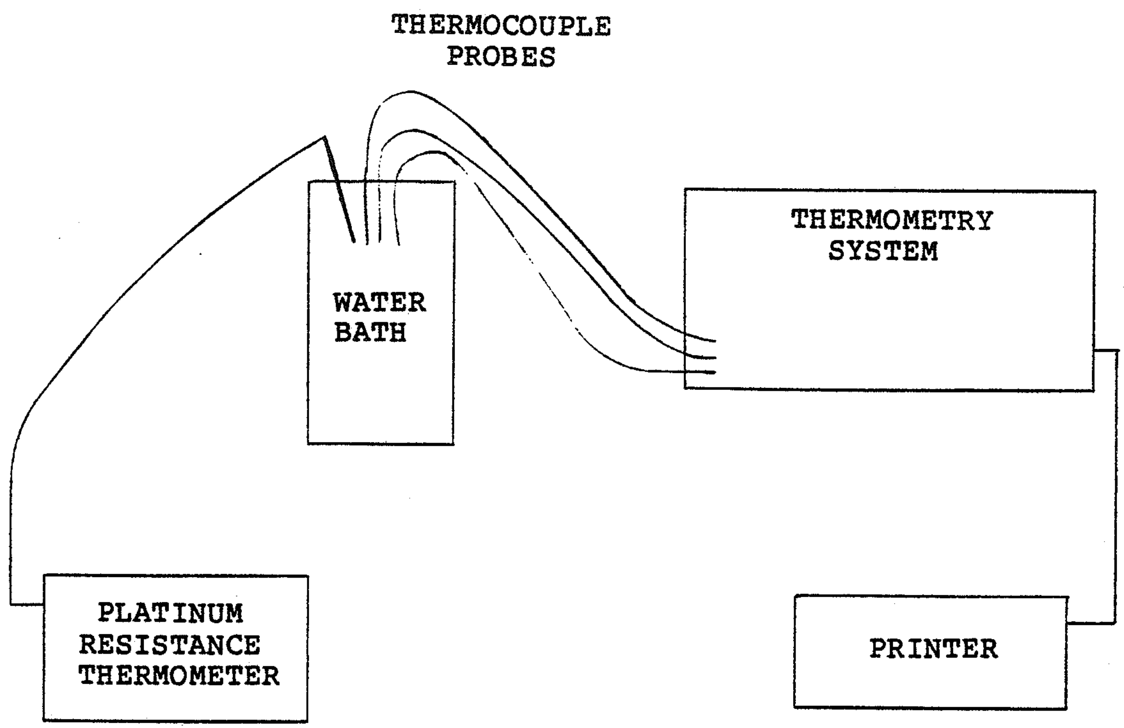


Figure 5.2. Experimental setup for taking data.

Since the water bath takes a moderate amount of time to stabilize, some of the experiments were aided by the use of two water baths.

5.3.2. Accuracy of the Two-Point Calibration

The purpose of this test, which included all the hardware, was to establish that a calibration routine based on two points could work with acceptable accuracy.

In order to set up the most rigid conditions possible, the probes were calibrated at 19.94°C and 59.95°C, which are near the extremes of the valid temperature range. The water bath temperature was then increased from 20°C to 60°C, in steps of about 4°C. At each point, the PRT temperature was read and the probe temperatures were printed out. In Table 5.2 are the PRT temperatures and their corresponding probe output temperatures from the five best channels.

It should be noted that a total of eight channels was observed during this test. However, the intent of the test was to evaluate the performance of the calibration routine only, and probes thought to be in error for some other reason, e.g., hardware failure, were not considered. These other kinds of probe error will be discussed later.

In Figure 5.3, PRT temperature minus thermometry output temperature is plotted versus PRT temperature, for the five best channels. As can be seen, the deviations of thermometry output temperature from PRT temperature are independent of PRT temperature and are concentrated within a range of ± 0.03°C.

At each PRT temperature in Table 5.2, the mean (average temperature), the standard deviation from the mean and the PRT temperature minus the mean were calculated for the same five

Table 5.2. Thermometry Output at Various Water Bath Temperatures.

PRT (°C)	Channel 1 (°C)	5 (°C)	7 (°C)	8 (°C)	10 (°C)
19.95	19.96	19.96	20.03	20.00	19.99
24.05	24.03	24.05	24.05	24.07	24.07
28.02	27.99	28.00	28.04	28.03	28.02
32.05	31.99	32.02	32.03	32.04	32.05
36.05	36.04	36.03	36.06	36.04	36.05
40.14	40.12	40.12	40.14	40.14	40.17
44.03	44.02	44.01	44.00	44.02	44.03
48.02	48.02	48.01	48.03	48.02	48.03
52.11	52.09	52.10	52.11	52.11	52.11
56.14	56.11	56.13	56.14	56.12	56.14
60.11	60.08	60.07	60.08	60.07	60.08

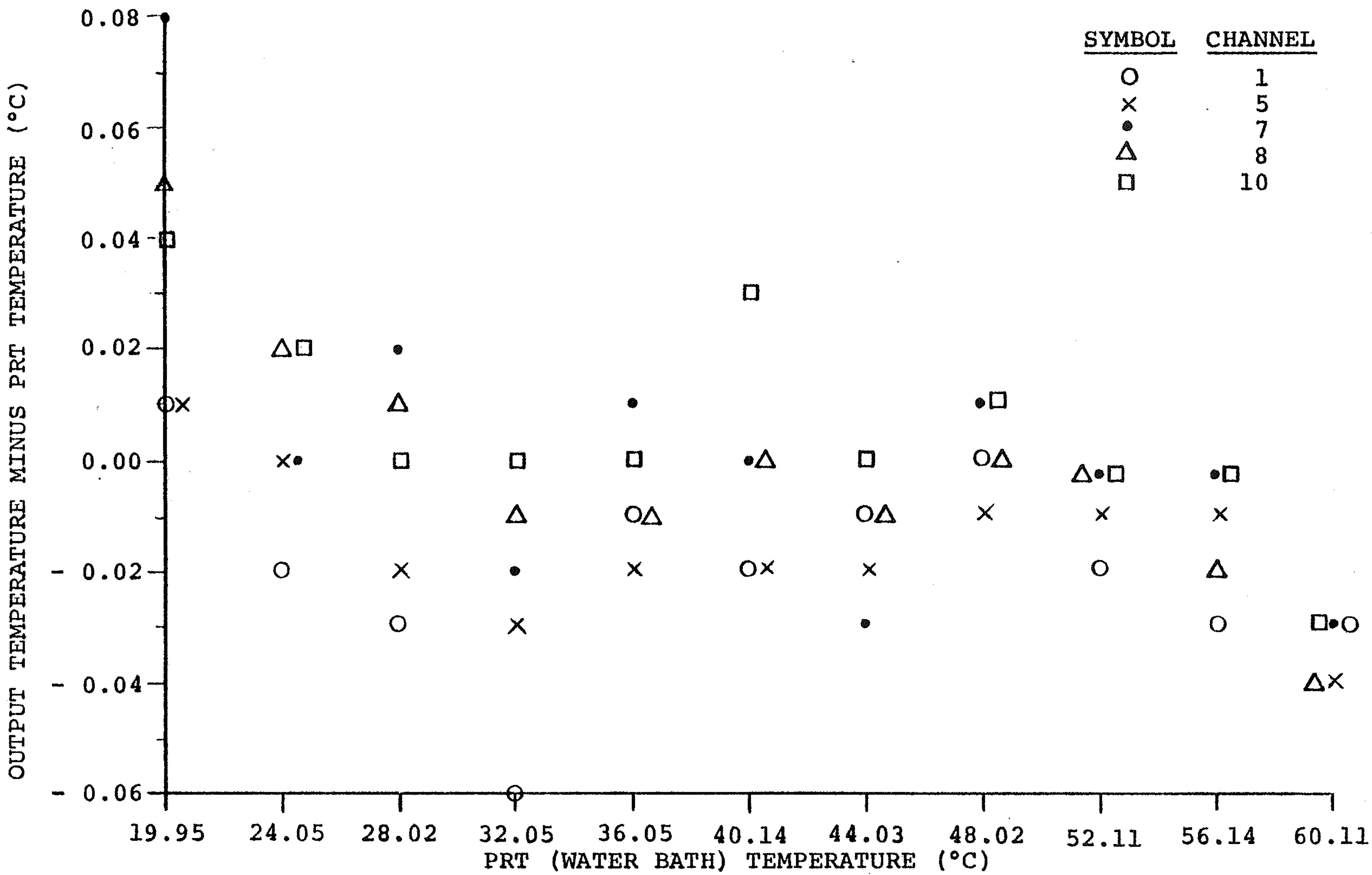


Figure 5.3. Thermometry output temperature minus PRT temperature plotted versus PRT temperature.

channels. These are tabulated in Table 5.3. This estimate of standard deviation is the same as that in Equation (3.3), except now the "ideal temperature" is replaced by the mean. According to the American Society for Testing and Materials (ASTM) [7], accuracy is defined as the degree of agreement between the true value of a property being tested and the average of many observations. The accuracy of the thermometry system is thus represented by the quantity "PRT-MEAN" and, as can be seen, the thermometry system is accurate to within $\pm 0.01^{\circ}C$ most of the time.

The standard deviation can provide quantitative information on the amount of scatter in the data. The above calculations of standard deviation from the mean will reveal how well localized all the measured temperatures were around their average value. A small value of standard deviation implies a high degree of precision in the measurements, since all the temperatures would be grouped closely together. With this in mind, one can determine an estimate of the precision of the system from the calculated standard deviation. In Table 5.3, the largest value of standard deviation is $0.03^{\circ}C$ (rounded to the nearest $0.01^{\circ}C$). The average of all the values of the standard deviation is $0.014^{\circ}C$. Therefore, an estimate of the precision of the system is $0.014^{\circ}C$, with an upper limit of approximately $0.03^{\circ}C$.

This test has demonstrated that the calibration routine will give favorable results over the desired range of calibrated temperatures.

Table 5.3. Channels at Each PRT Temperature.

PRT ($^{\circ}$C)	Mean ($^{\circ}$C)	Standard Deviation ($^{\circ}$C)	(PRT-Mean) ($^{\circ}$C)
19.95	19.99	0.026	-0.04
24.05	24.05	0.015	0.00
28.02	28.02	0.019	0.00
32.05	32.03	0.021	+0.02
36.05	36.04	0.011	+0.01
40.14	40.14	0.018	0.00
44.03	44.02	0.010	+0.01
48.02	48.02	0.007	0.00
52.11	52.10	0.008	+0.01
56.14	56.13	0.012	+0.01
60.11	60.08	0.005	+0.03

5.3.3. Stability - The Twenty-Four Hour Run

The stability of the thermometry system over long periods of time involves consideration of many factors. For example, it was discovered that the temperature of the water bath could fluctuate by $\pm 0.01^{\circ}C$ from the desired temperature during long runs. Also, changes in room temperature could have an effect on the internal compensation of the thermocouple amplifier chips. An increase in the ambient chip temperature of only $5^{\circ}C$ could cause an error of $0.01^{\circ}C$, as mentioned in Section 3.2.

For this test, the thermometry system was calibrated at $19.97^{\circ}C$ and $60.14^{\circ}C$. Then, with the probes in a water bath of about $40.10^{\circ}C$, the output temperatures were sent to the printer every ten minutes for twenty-four hours. Twelve channels were used in this test. Figures 5.4 and 5.5 show the thermometry output temperatures taken every hour for ten of the channels. It is seen that the probe temperatures deviated by $-0.07^{\circ}C$ to $+0.04^{\circ}C$ from the PRT temperature of $40.10^{\circ}C$. Channel sixteen's probe was placed inside the thermometry system after a few readings to monitor the temperature of the thermocouple amplifier chips. The output temperatures of probe six all deviated by about $0.16^{\circ}C$ from the PRT temperature. This channel's output was not shown on these graphs, but will be considered later.

Figures 5.6 through 5.16 show the variation of temperature every ten minutes for each of the ten channels, including channel six. Notice that most of the channels are fairly well behaved, with the exception of channels five and six. Channel five has one spurious reading, which may have been caused by analog hardware noise, and the readings for channel six are all centered around

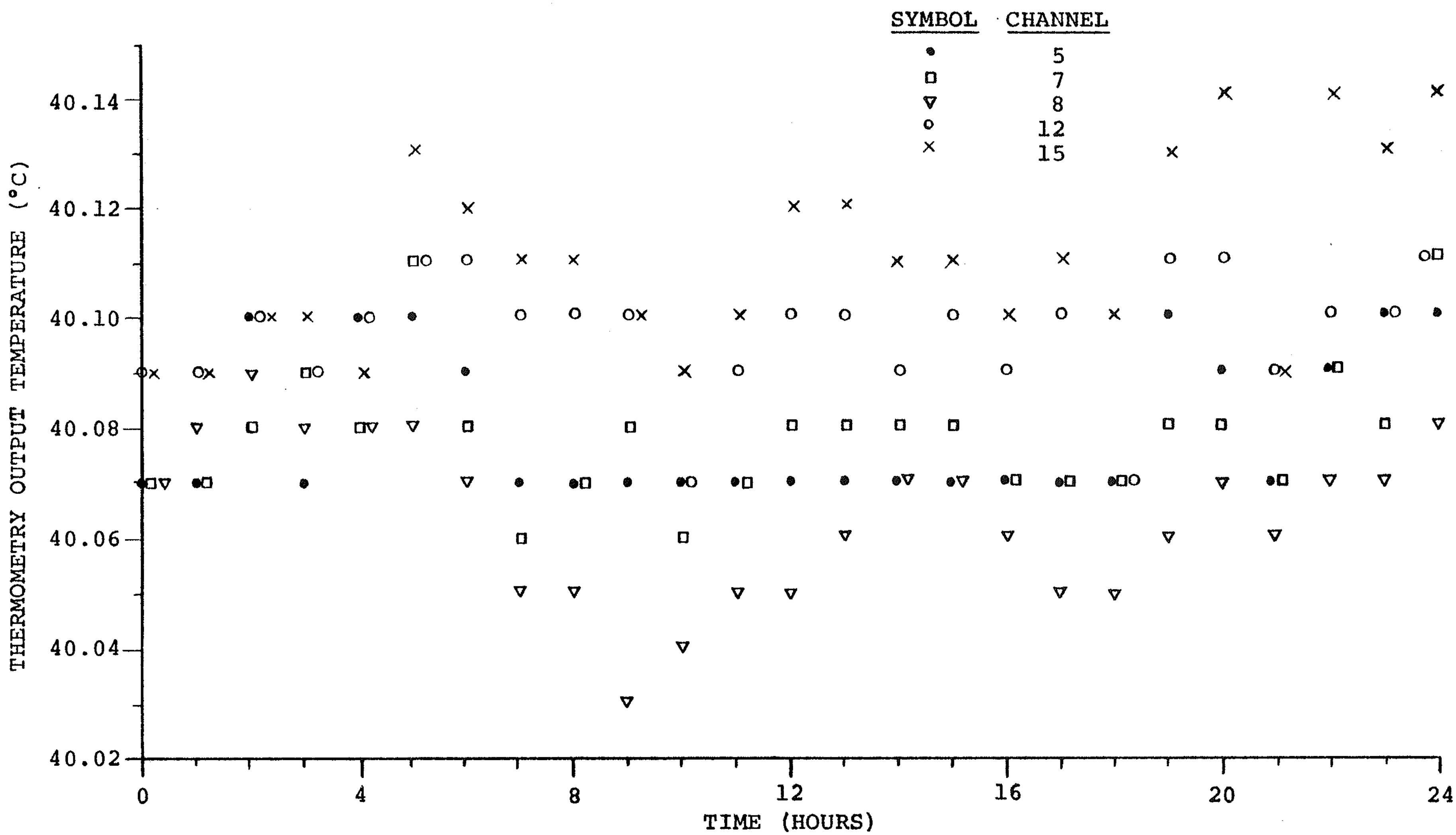


Figure 5.4. Thermometry output over 24 hours with probes in a water bath of 40.10 °C.

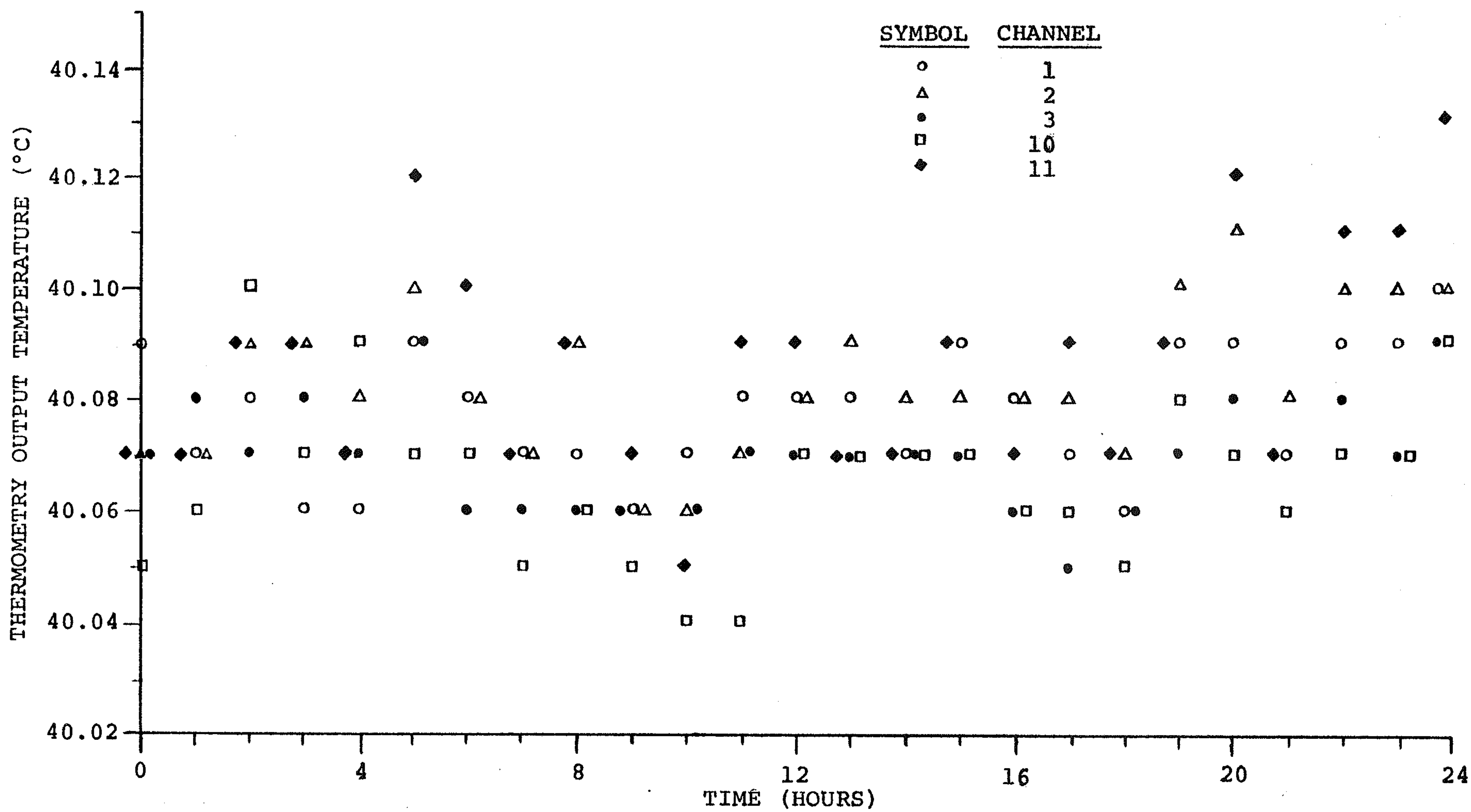


Figure 5.5. Thermometry output over 24 hours with probes in a water bath of 40.10 °C.

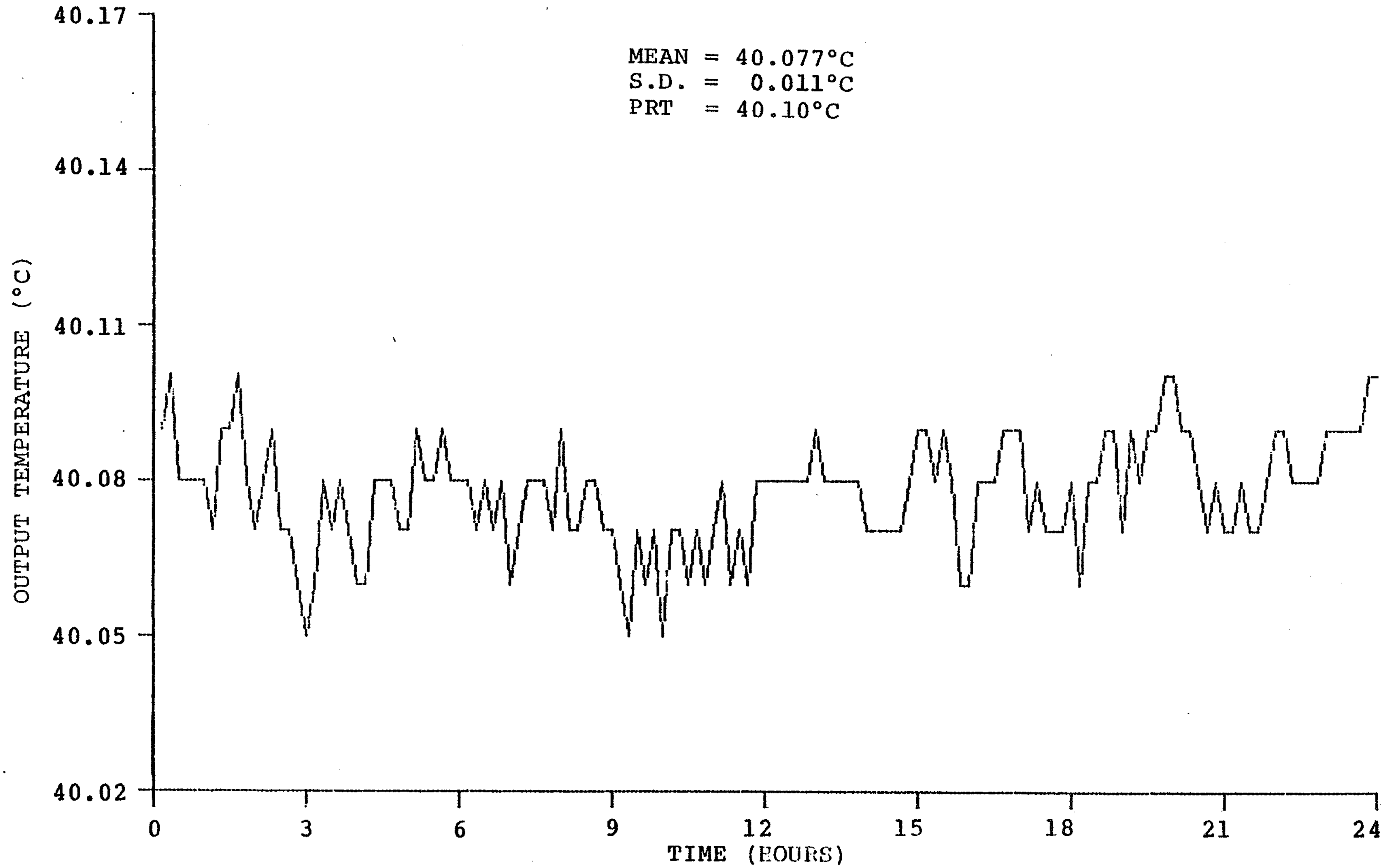


Figure 5.6. Temperature variation of channel 1 with readings made every 10 minutes.

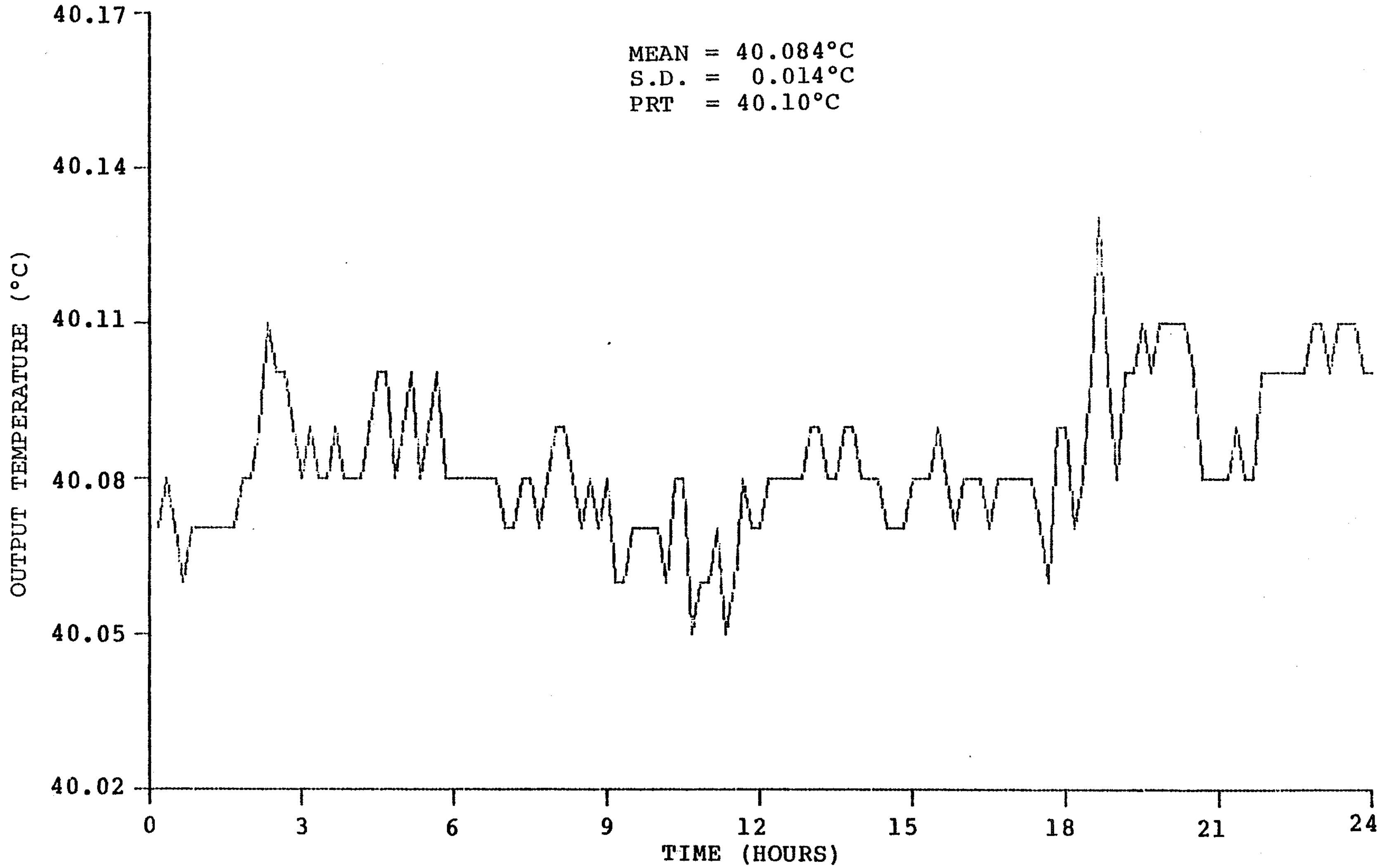


Figure 5.7. Temperature variation of channel 2 with readings made every 10 minutes.

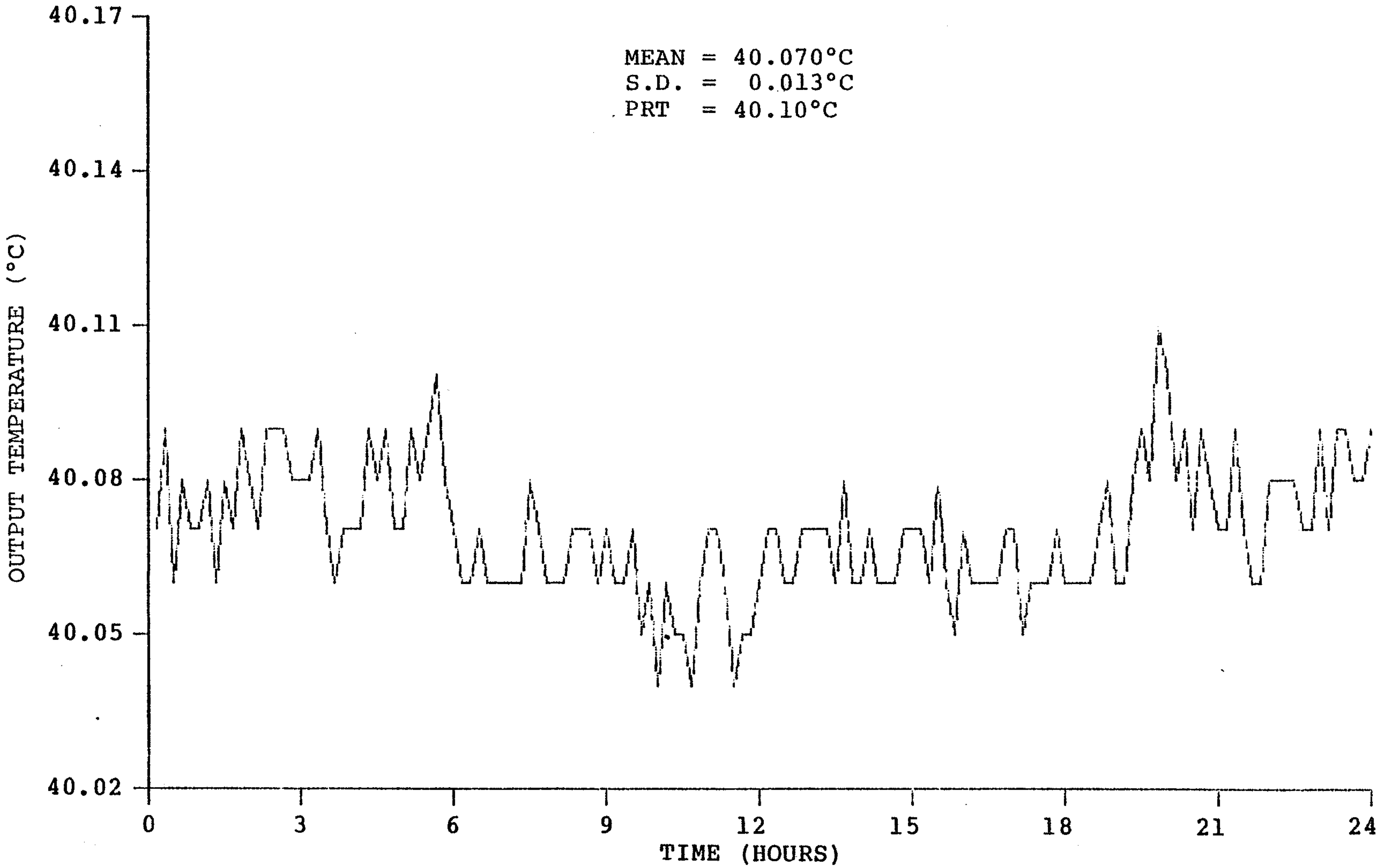


Figure 5.8. Temperature variation of channel 3 with readings made every 10 minutes.

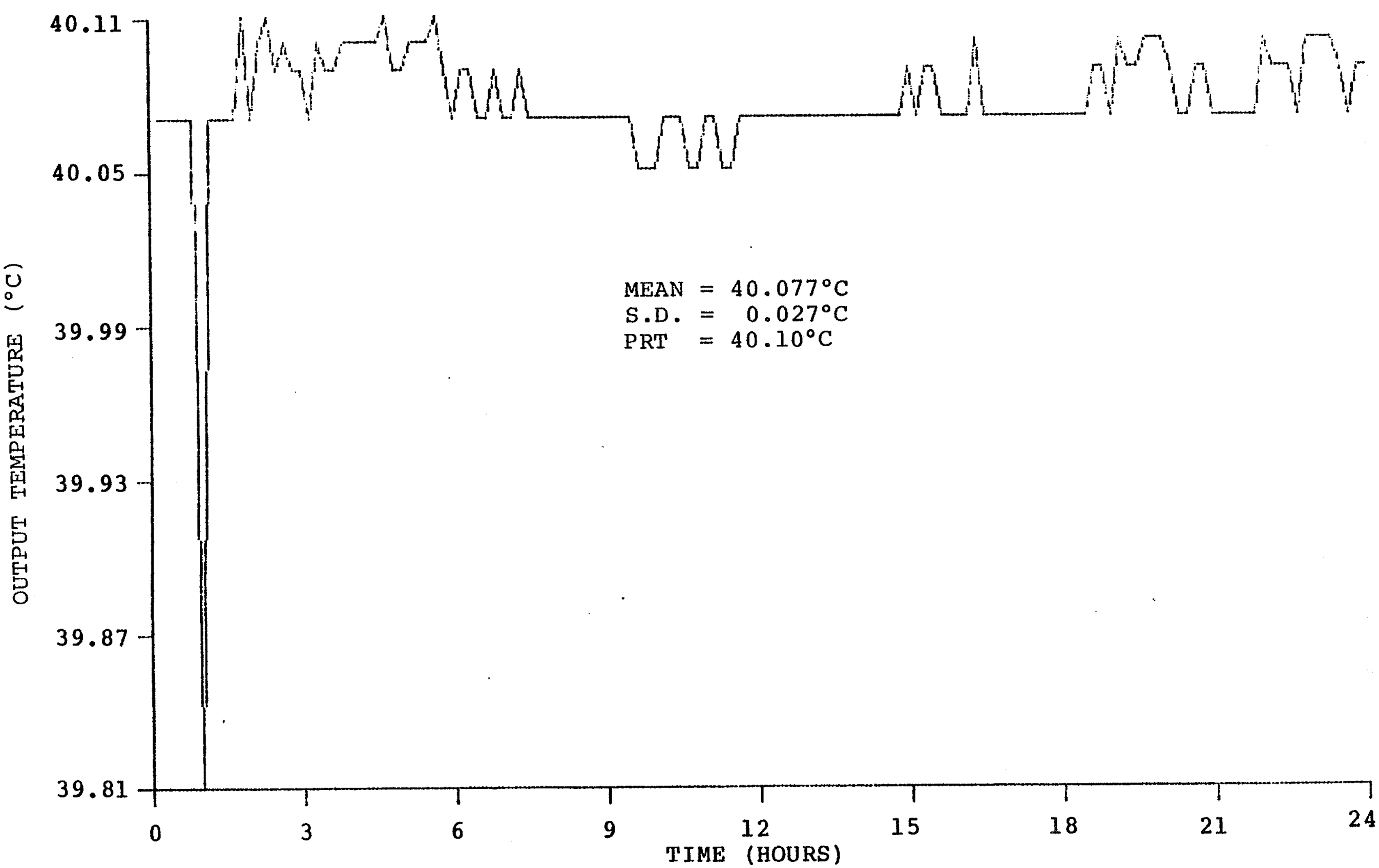


Figure 5.9. Temperature variation of channel 5 with readings made every 10 minutes.

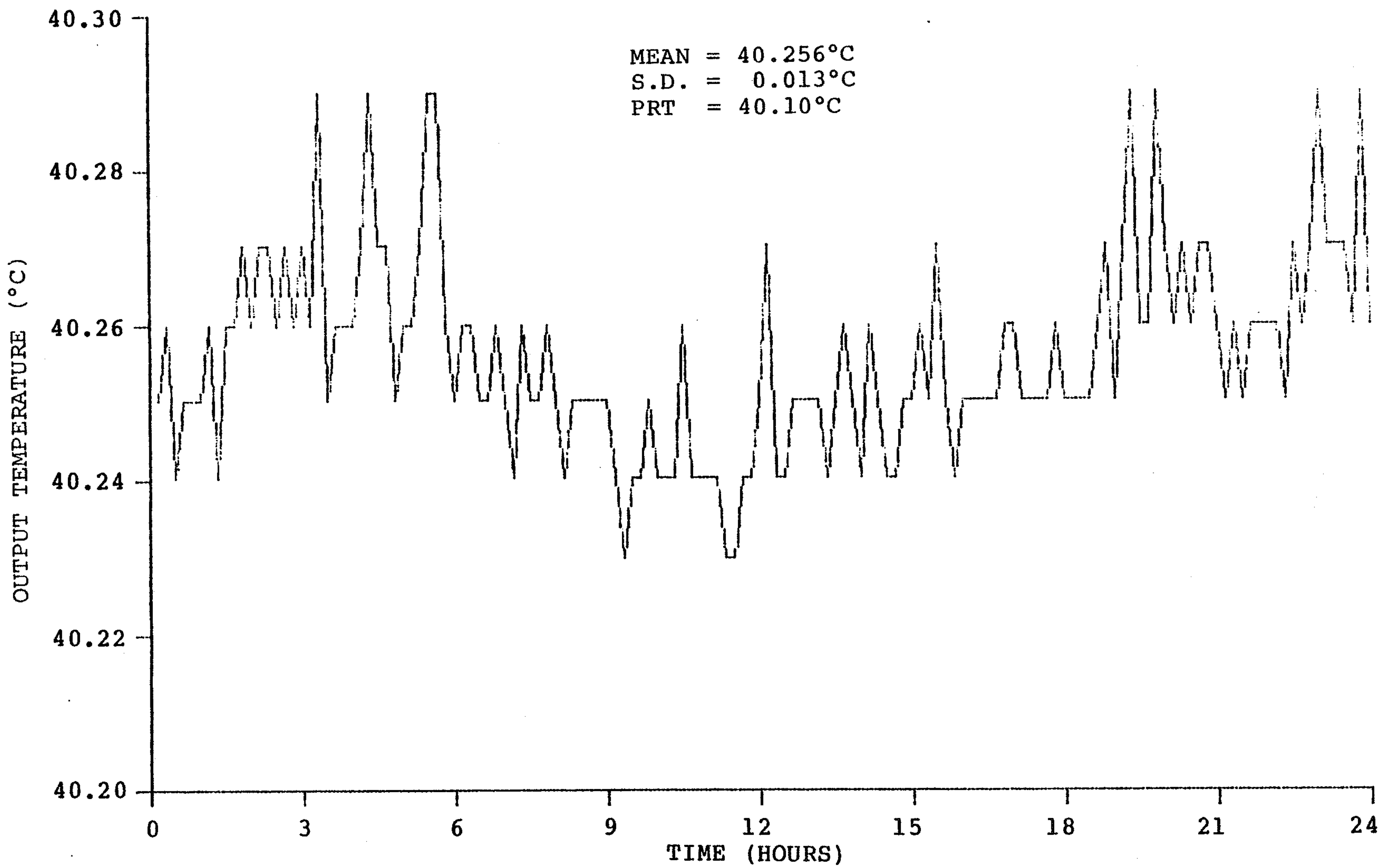


Figure 5.10. Temperature variation of channel 6 with readings made every 10 minutes.

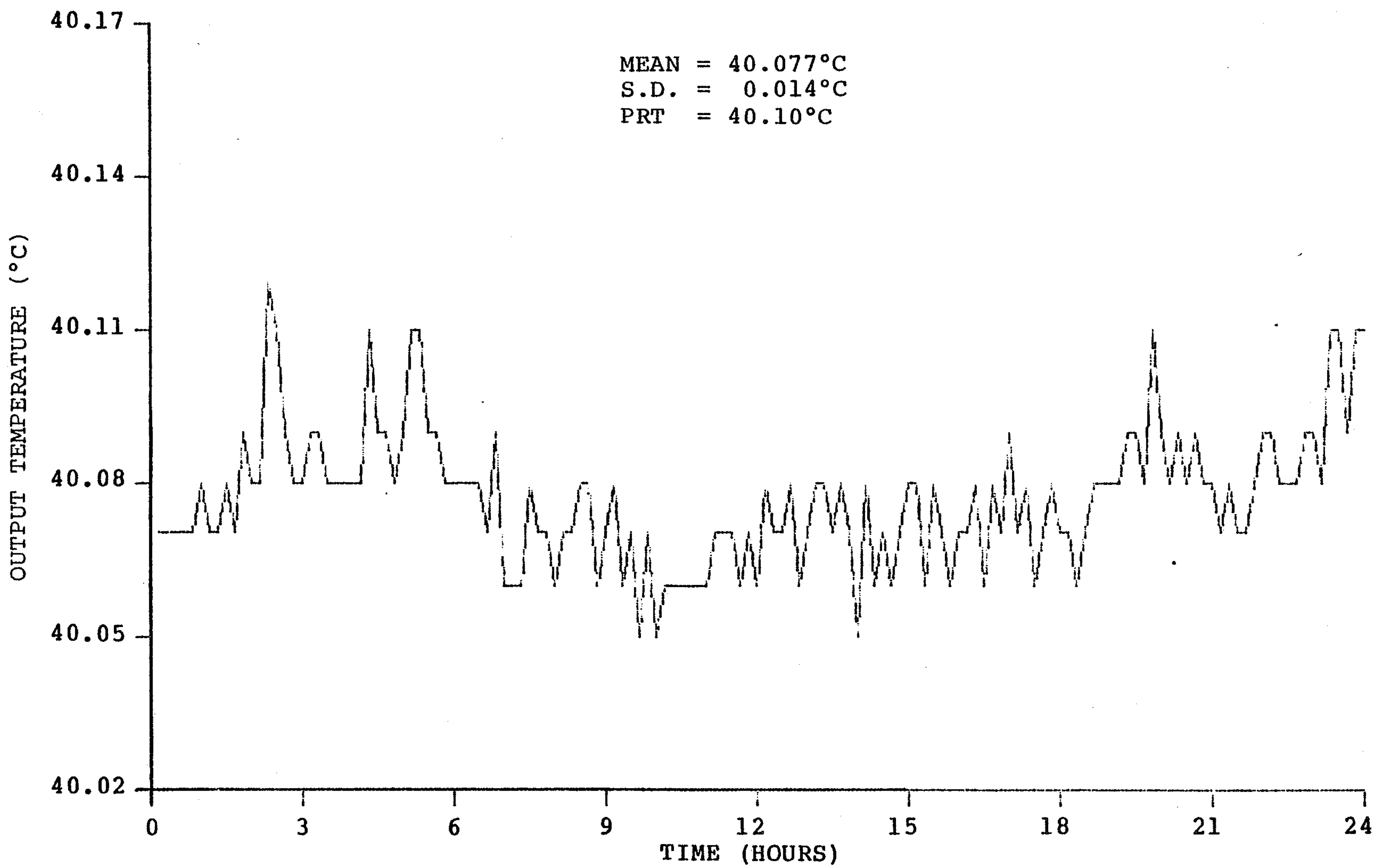


Figure 5.11. Temperature variation of channel 7 with readings made every 10 minutes.

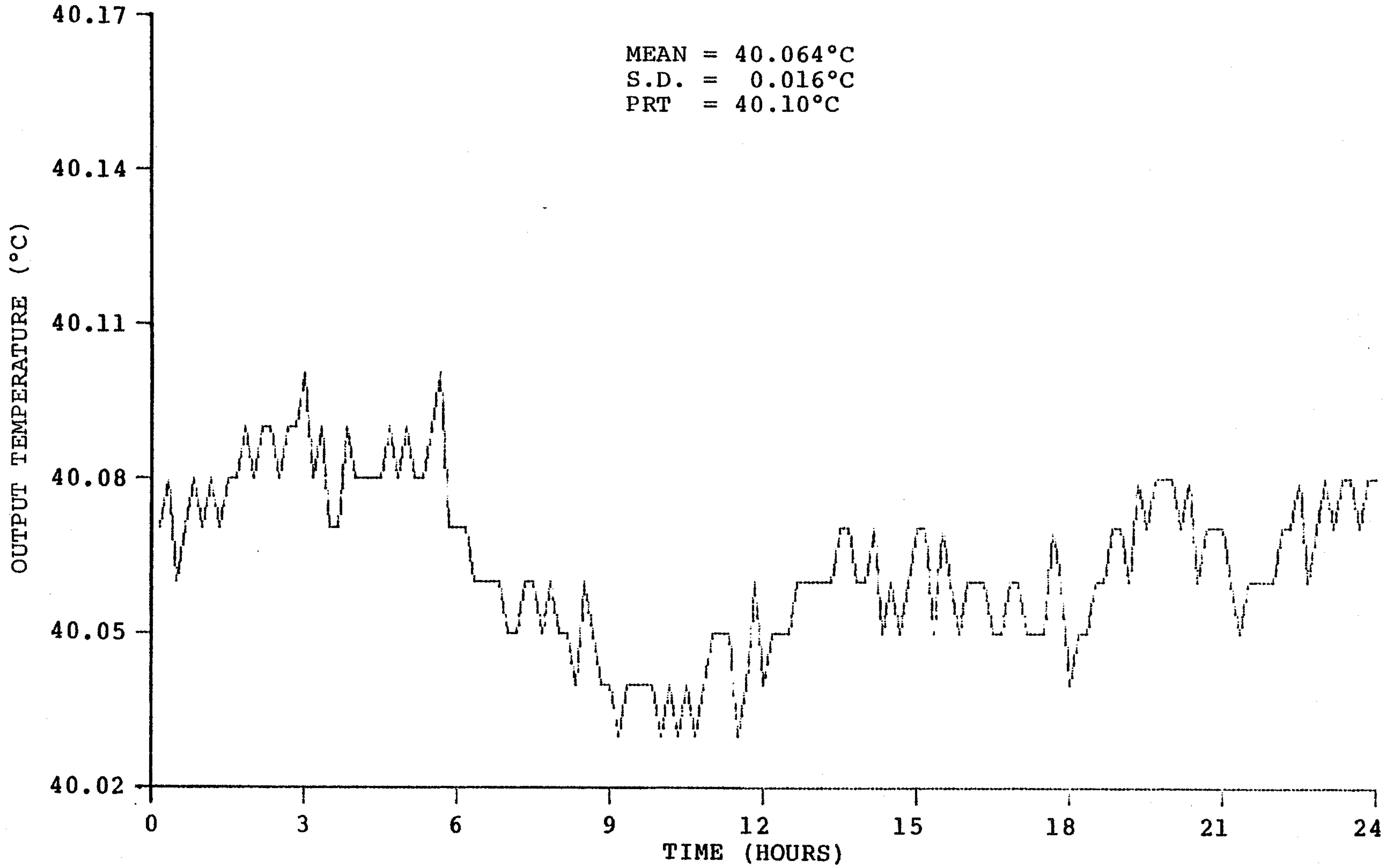


Figure 5.12. Temperature variation of channel 8 with readings made every 10 minutes.

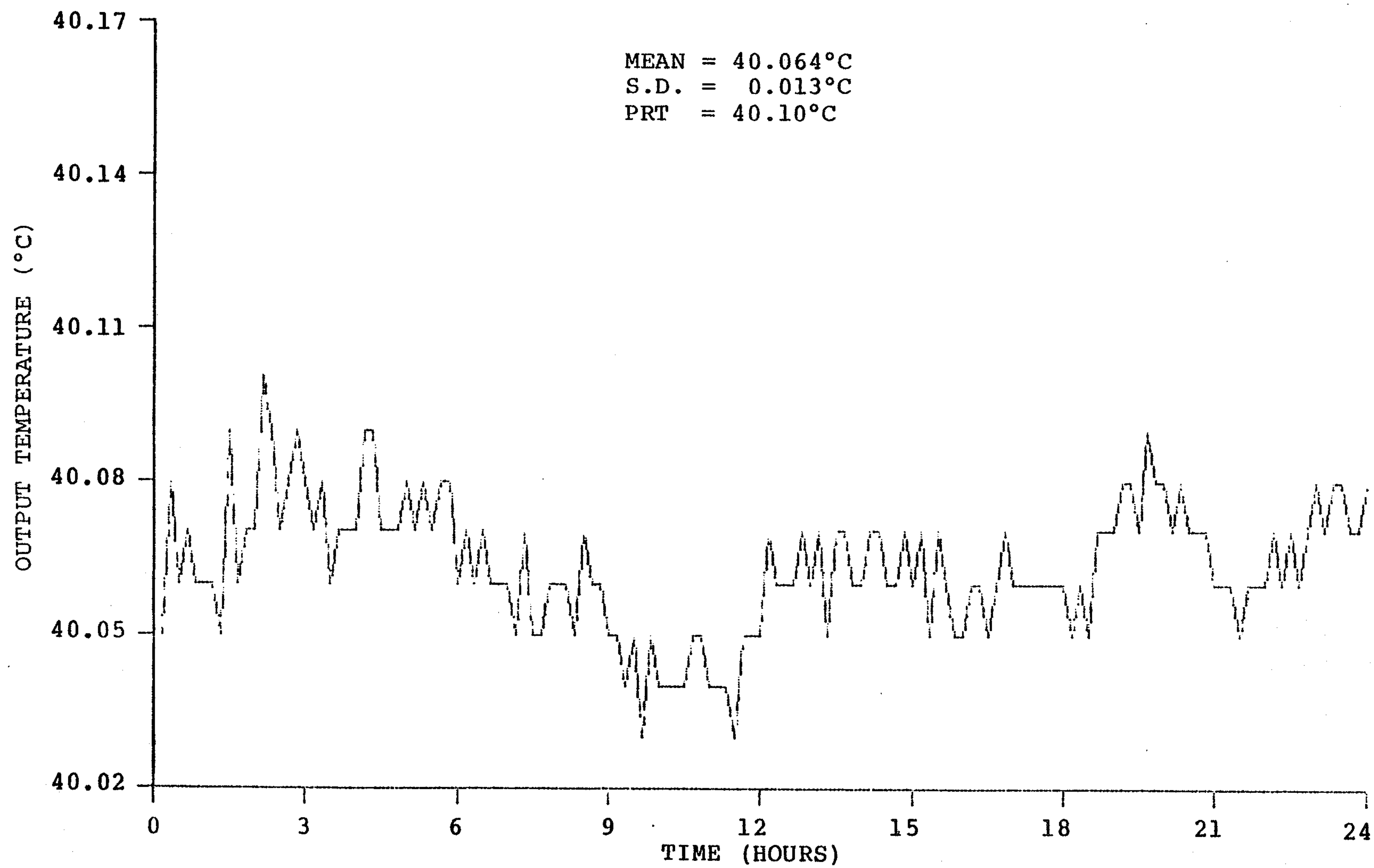


Figure 5.13. Temperature variation of channel 10 with readings made every 10 minutes.

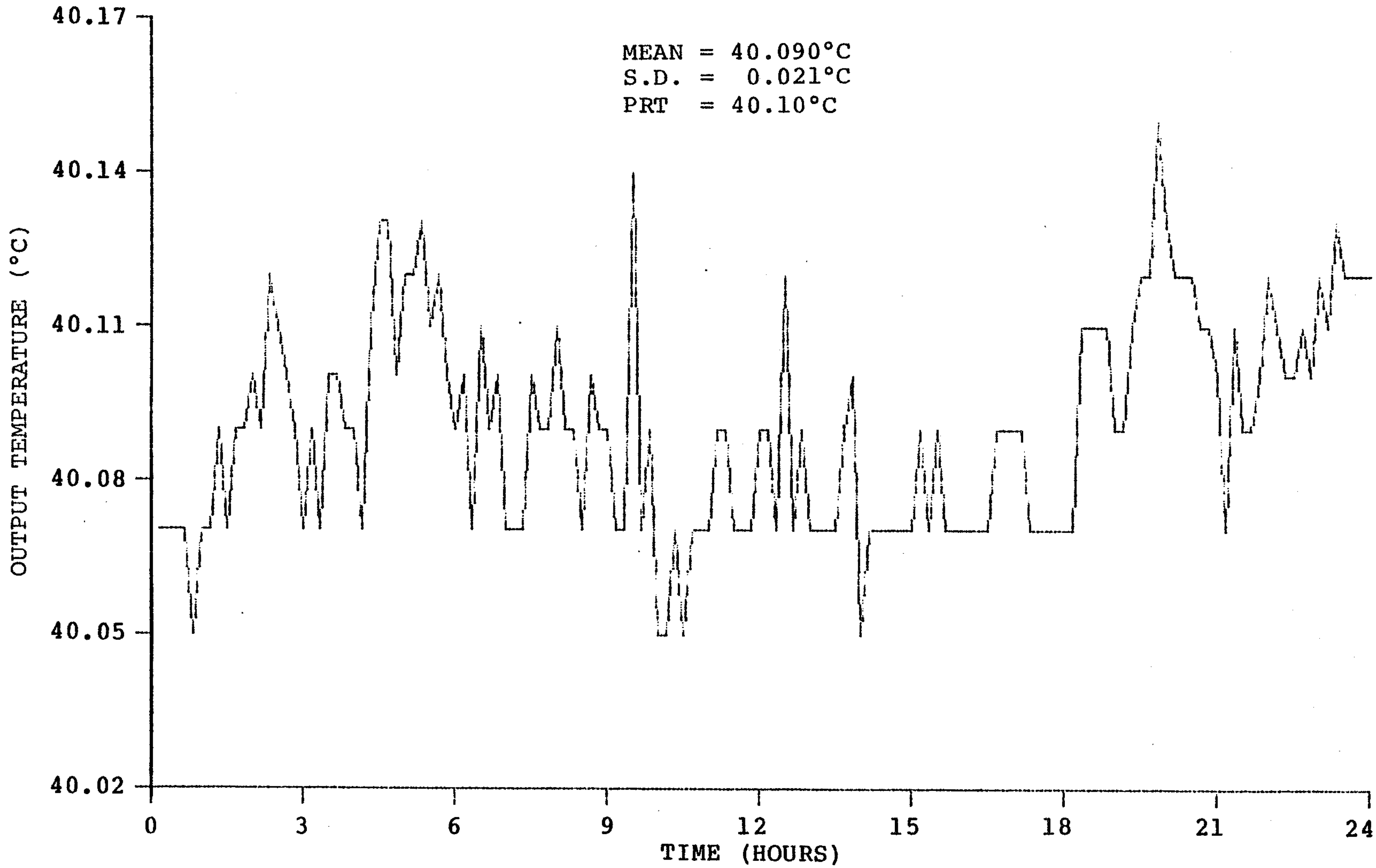


Figure 5.14. Temperature variation of channel 11 with readings made every 10 minutes.

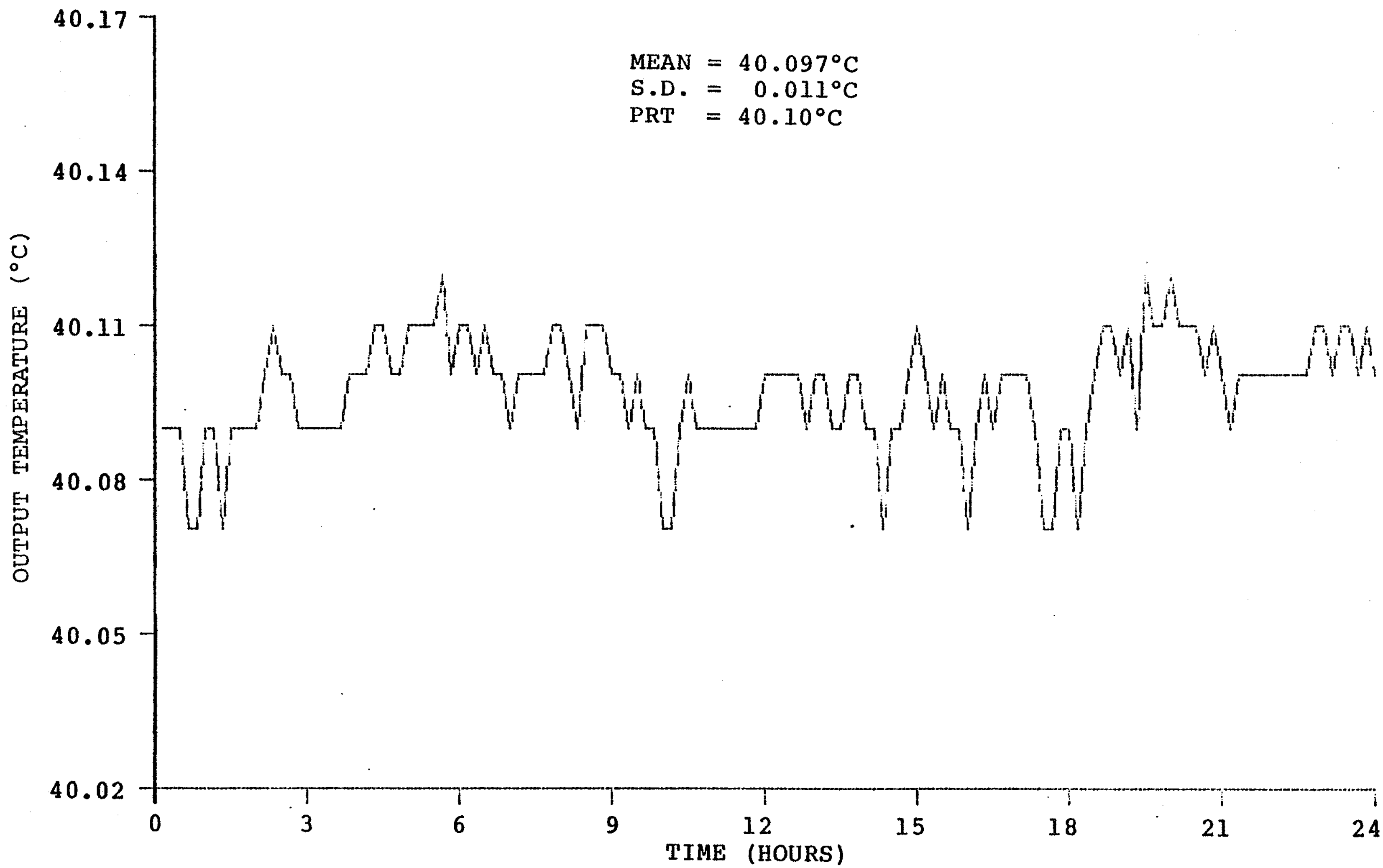


Figure 5.15. Temperature variation of channel 12 with readings made every 10 minutes.

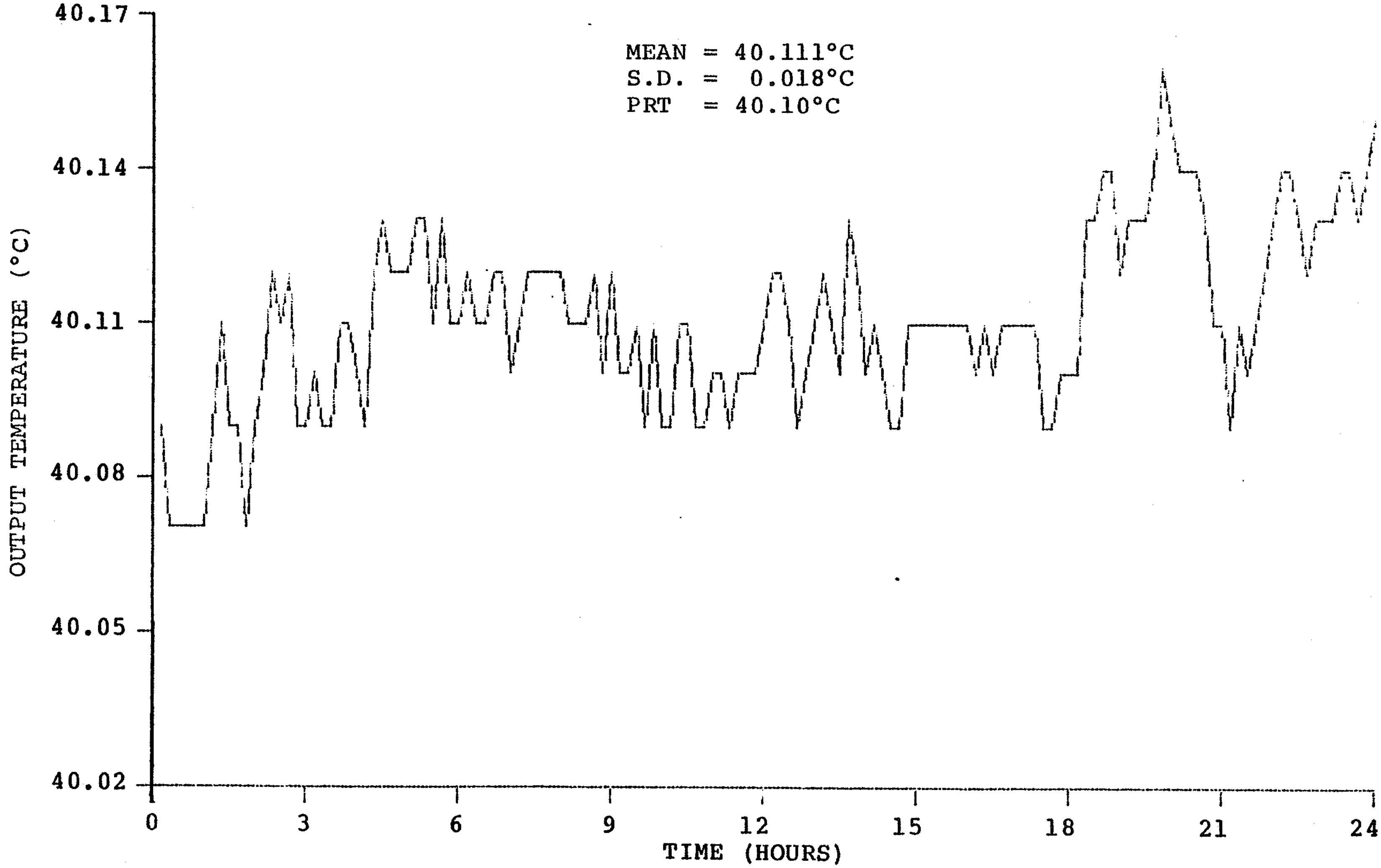


Figure 5.16. Temperature variation of channel 15 with readings made every 10 minutes.

40.26°C. Also, the mean and standard deviation for each channel are printed on the graphs and tabulated in Table 5.4.

Comparing the mean values to the PRT value of 40.10°C will give a measure of the accuracy of the thermometry system, as discussed in the previous subsection. As seen in Table 5.4, all channels, except channel six, are accurate to within ±0.04°C. However, all the channels are precise. The standard deviation, which is a measure of the precision, is less than 0.03°C for all channels.

5.3.5. Repeatability

The repeatability of accurate, calibrated temperatures was examined in a series of structured tests. Before each of the twelve experiments, described in Table 5.5, a calibration was performed at approximately 20°C and 60°C. Then, the probes were read at particular intervals of time while in a constant temperature water bath, stable to within ±0.01°C.

Twelve channels were used in each test. For a particular test, all twelve channels were read at each reading. The output temperatures of all twelve channels were then plotted together in the form of a histogram. For the tests where 100 readings were made, twenty representative readings were chosen from each channel. Therefore, each histogram contains 20 * 12 = 240 data points, which represents twenty temperature readings from each of the twelve channels. Figures 5.17 through 5.28 are the point histograms plotting the number of times each temperature occurs versus temperature, for each test. Also, the PRT temperature and the mean temperature have been marked on the plots. This method of presenting the data gives additional insight into the

Table 5.4. Properties of Each Channel in the Twenty-four Hour Run.

Channel	Mean (^{o}C)	Standard Deviation (^{o}C)	($40.1^{o}C$-Mean) (^{o}C)
1	40.08	0.011	+0.02
2	40.08	0.014	+0.02
3	40.07	0.013	+0.03
5	40.08	0.027	+0.02
6	40.26	0.013	-0.16
7	40.08	0.014	+0.02
8	40.06	0.016	+0.04
10	40.06	0.013	+0.04
11	40.09	0.021	+0.01
12	40.10	0.011	0.00
15	40.11	0.018	-0.01

Table 5.5. Format of the Twelve Tests.

Test Number	Temperature of Water Bath(^{o}C)	Time Between Reading(sec)	Number of Readings
1	20	10	20
2	20	36	100
3	20	60	20
4	20	180	100
5	40	10	20
6	40	36	100
7	40	60	20
8	40	180	100
9	60	10	20
10	60	36	100
11	60	60	20
12	60	180	100

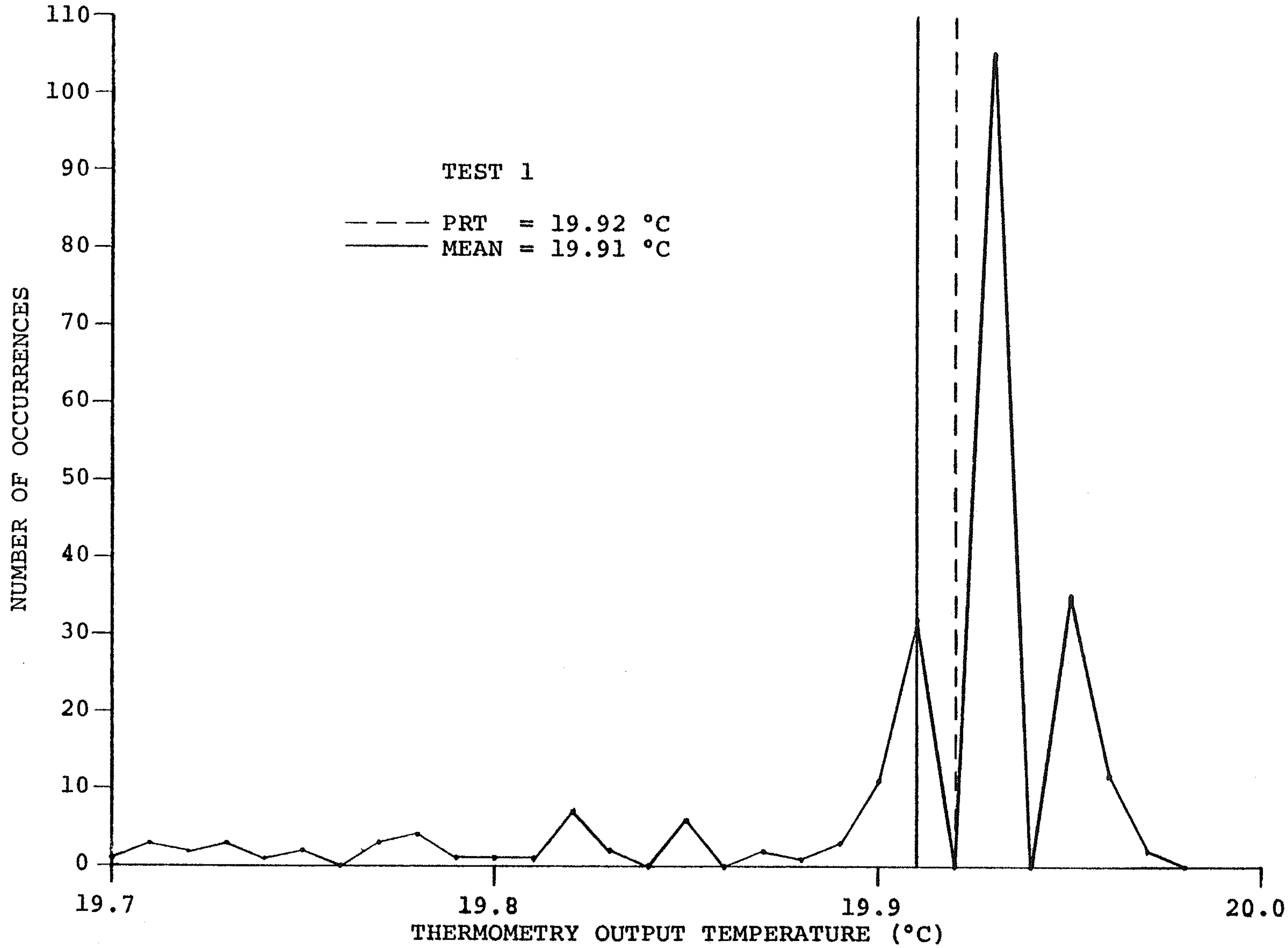


Figure 5.17. Distribution of output temperatures when probes are read every 10 seconds.

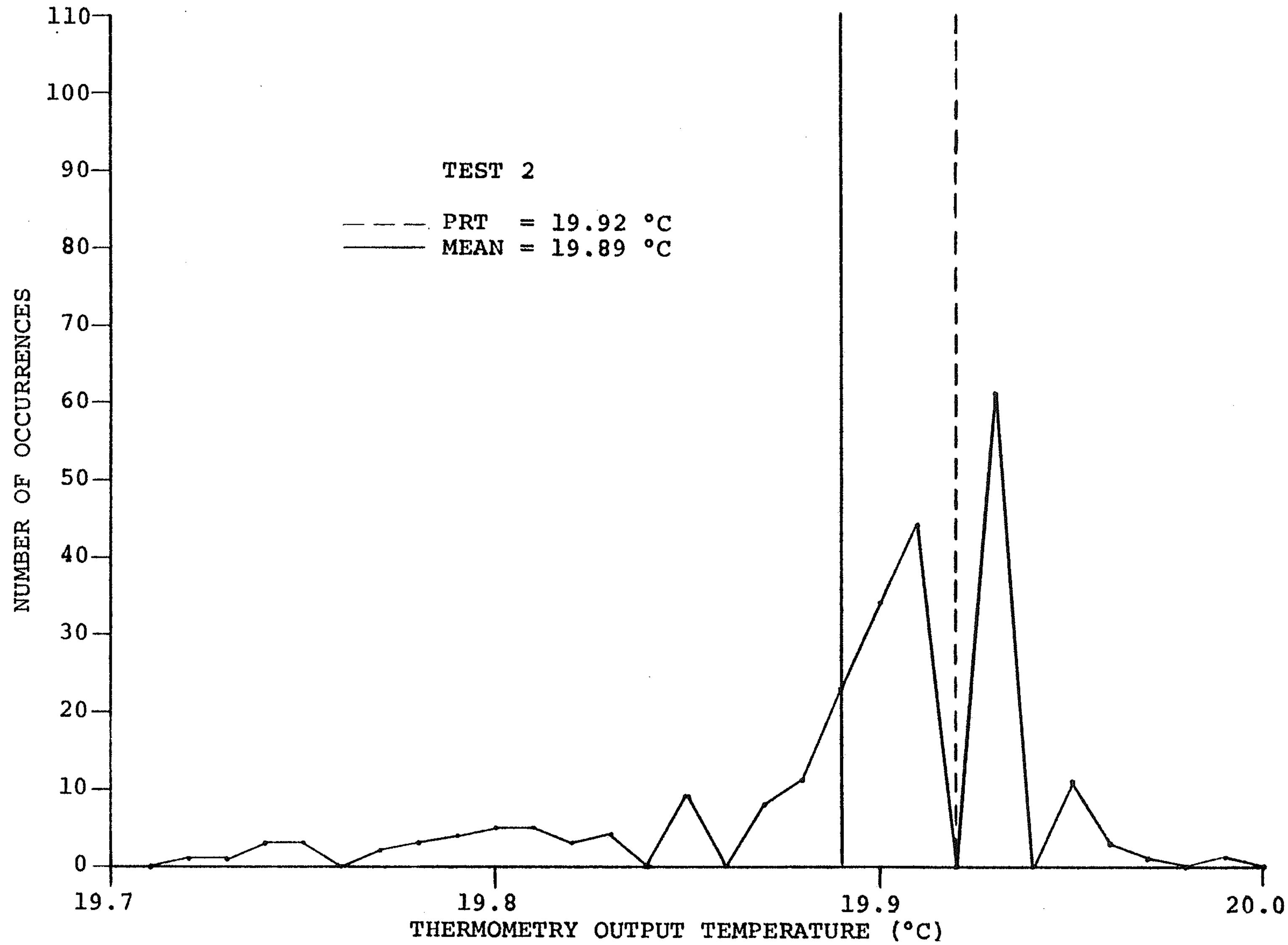


Figure 5.18. Distribution of output temperatures when probes are read every 36 seconds.

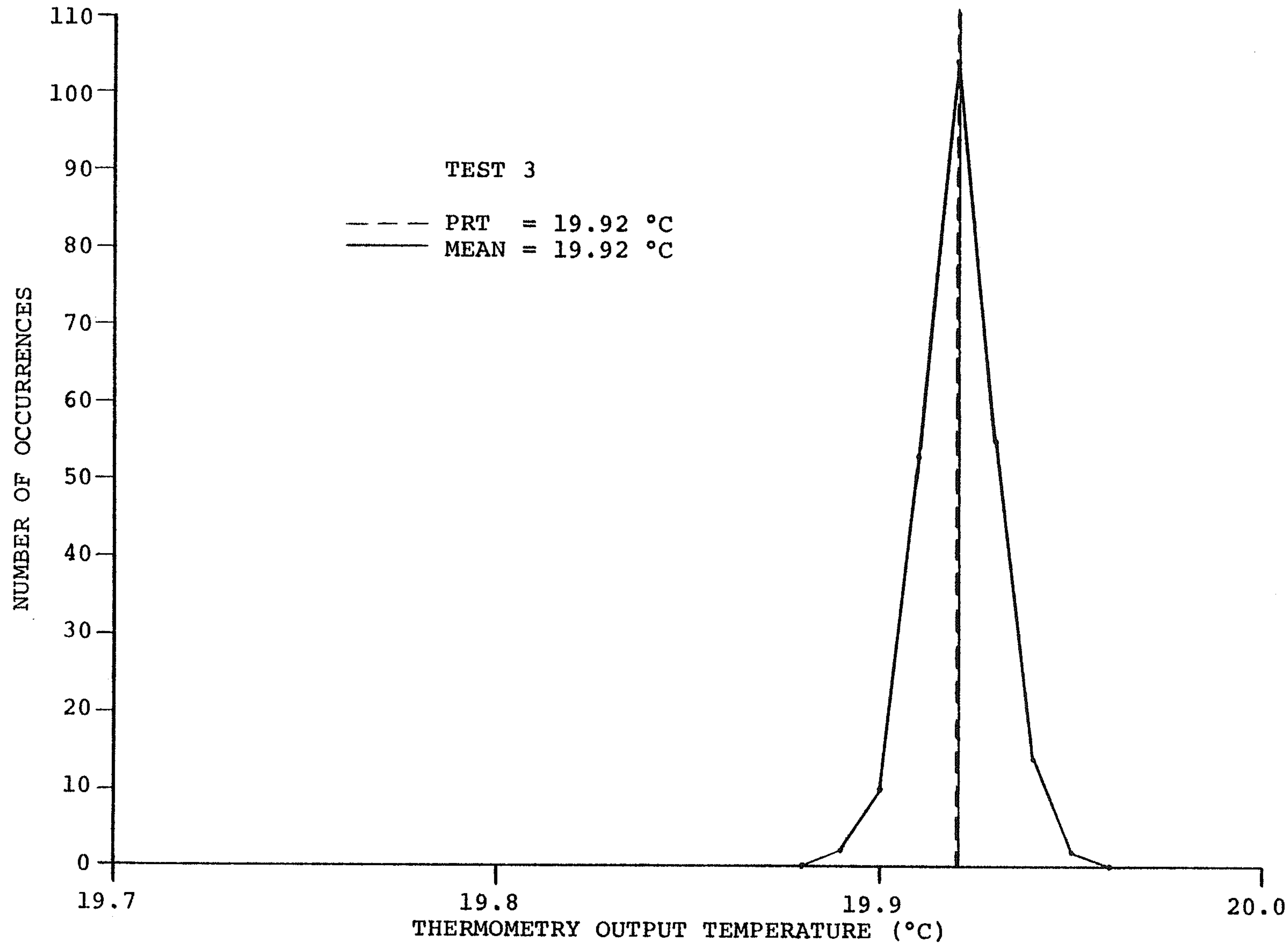


Figure 5.19. Distribution of output temperature when probes are read every 60 seconds.

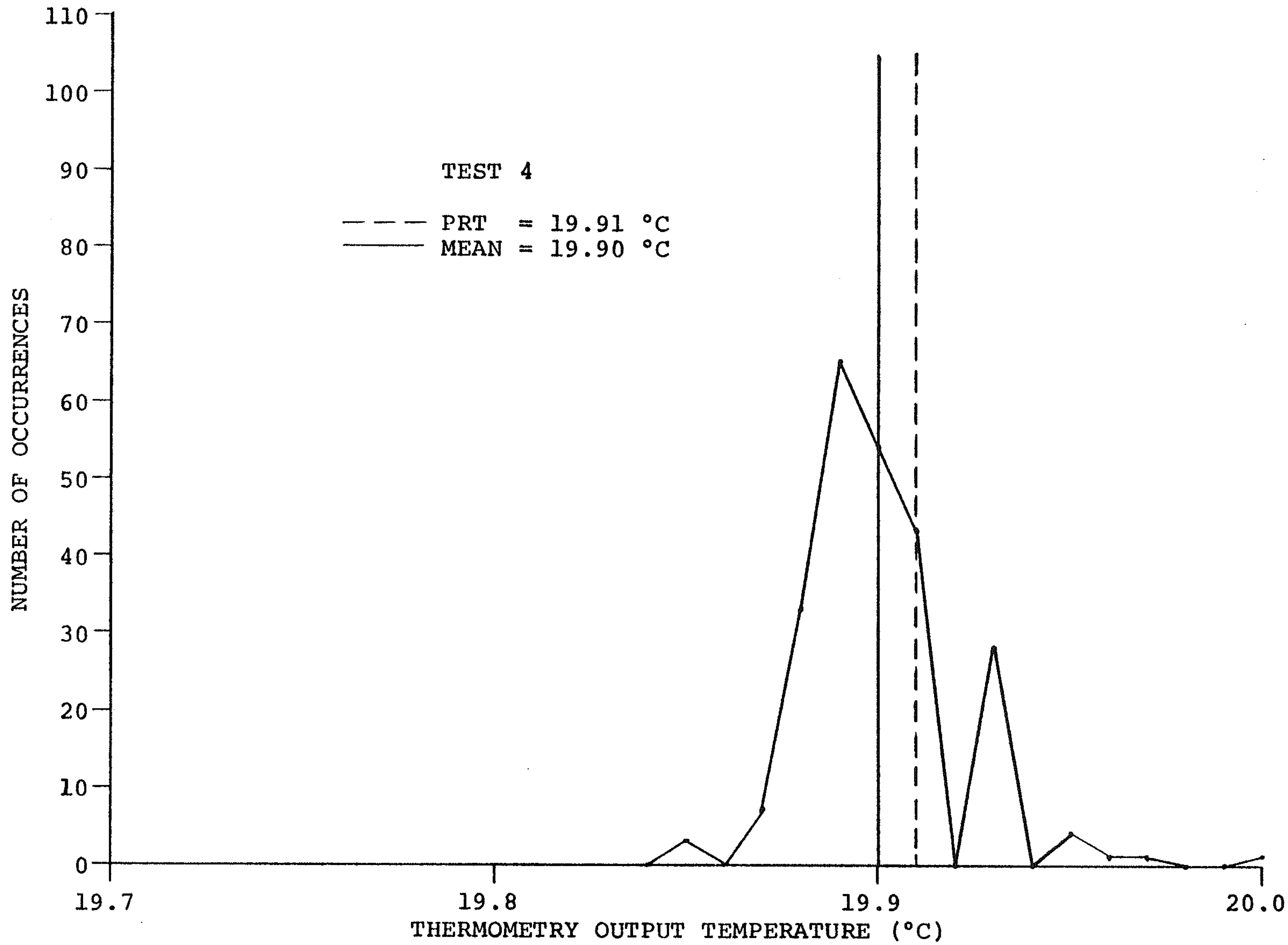


Figure 5.20. Distribution of output temperatures when probes are read every 180 seconds.

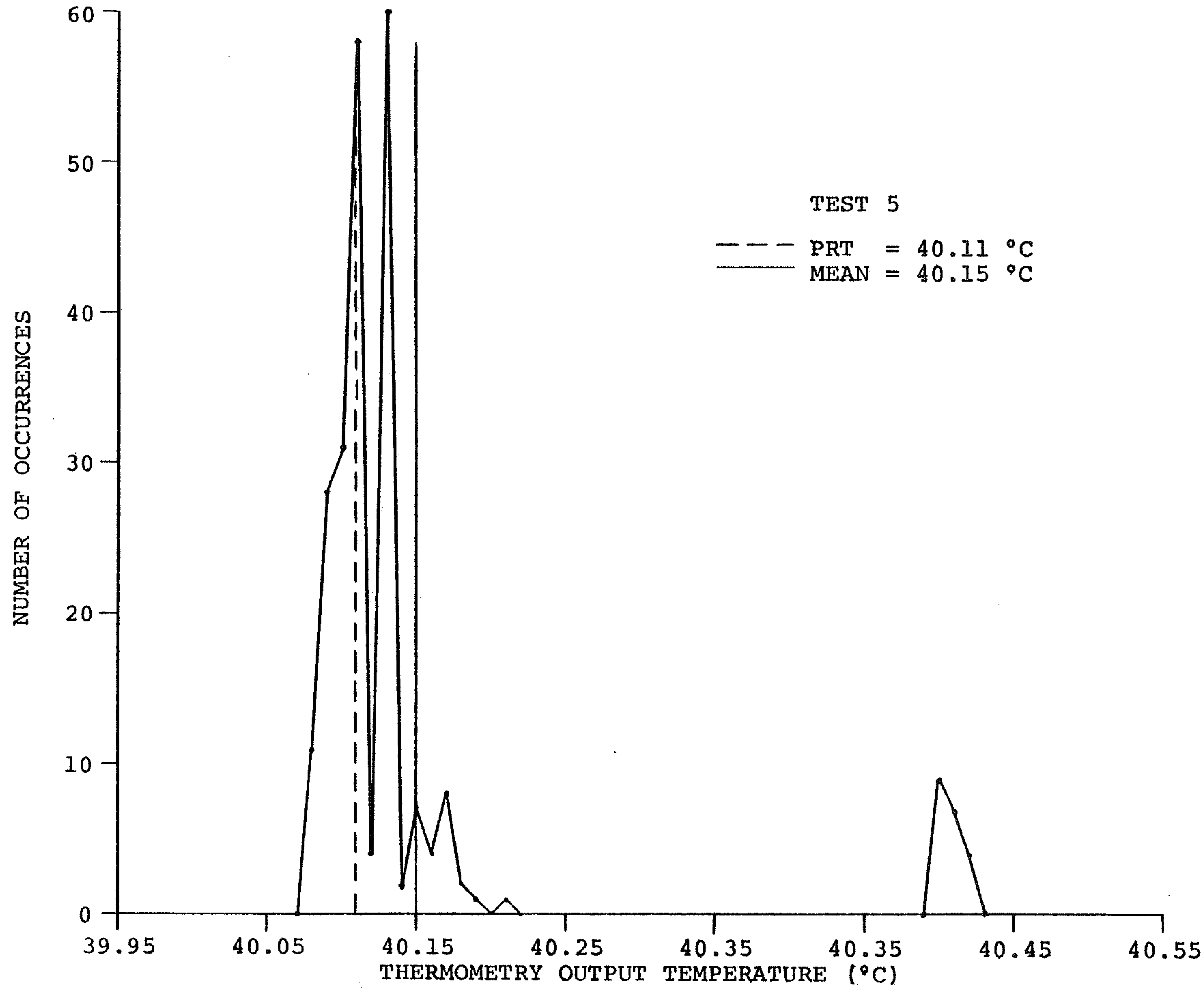


Figure 5.21. Distribution of output temperatures when probes are read every 10 seconds.

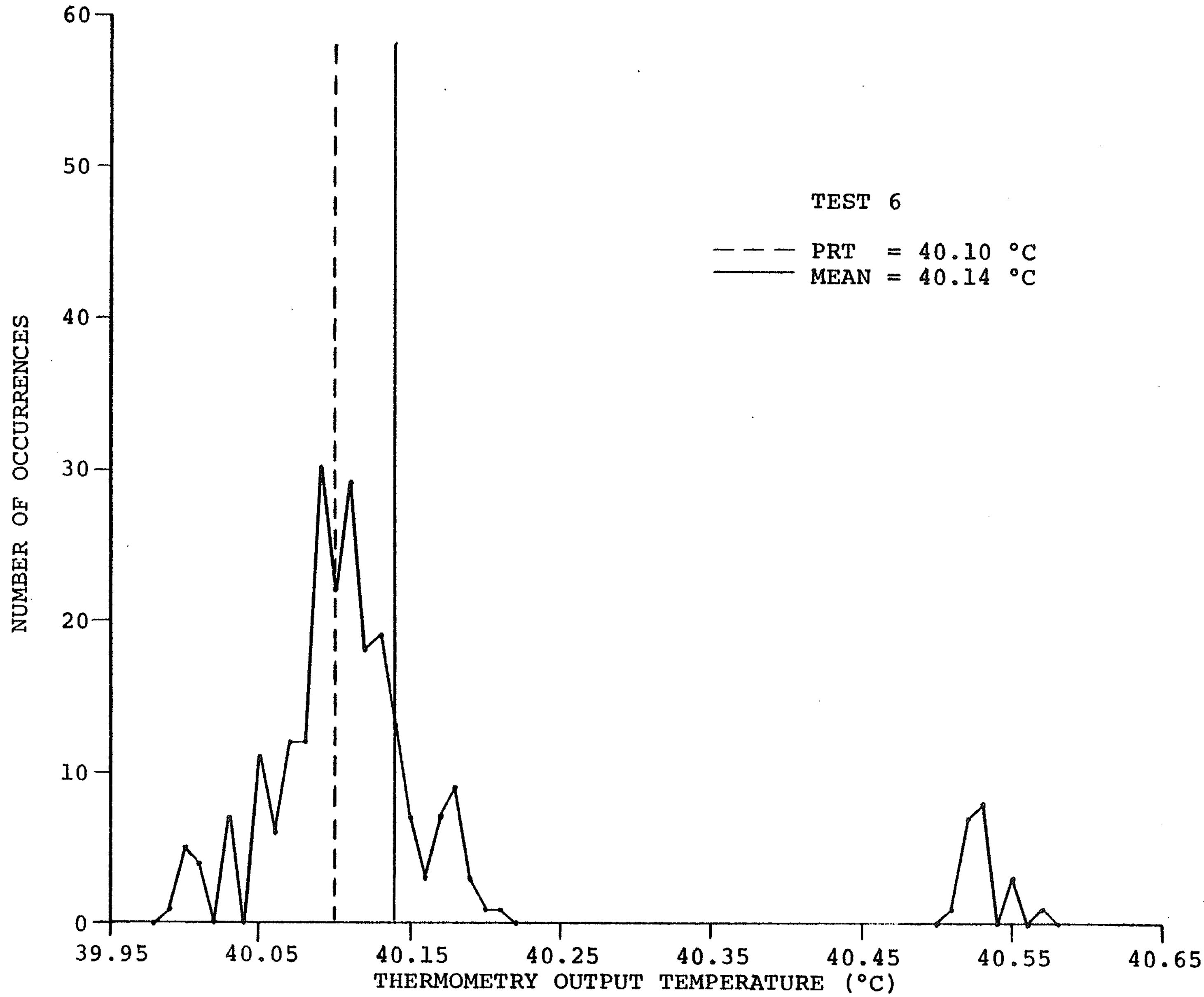


Figure 5.22. Distribution of output temperatures when probes are read every 36 seconds.

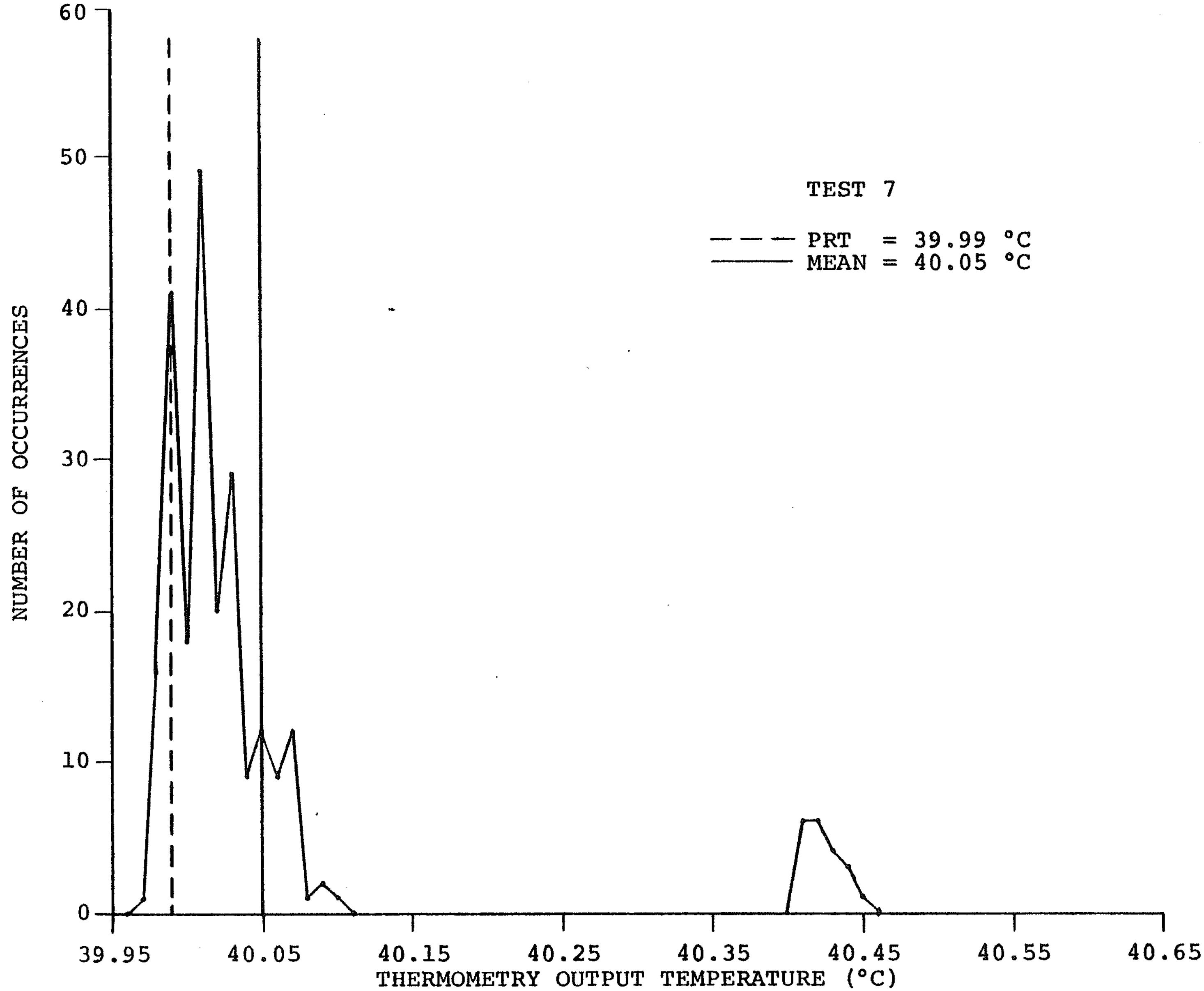


Figure 5.23. Distribution of output temperatures when probes are read every 60 seconds.

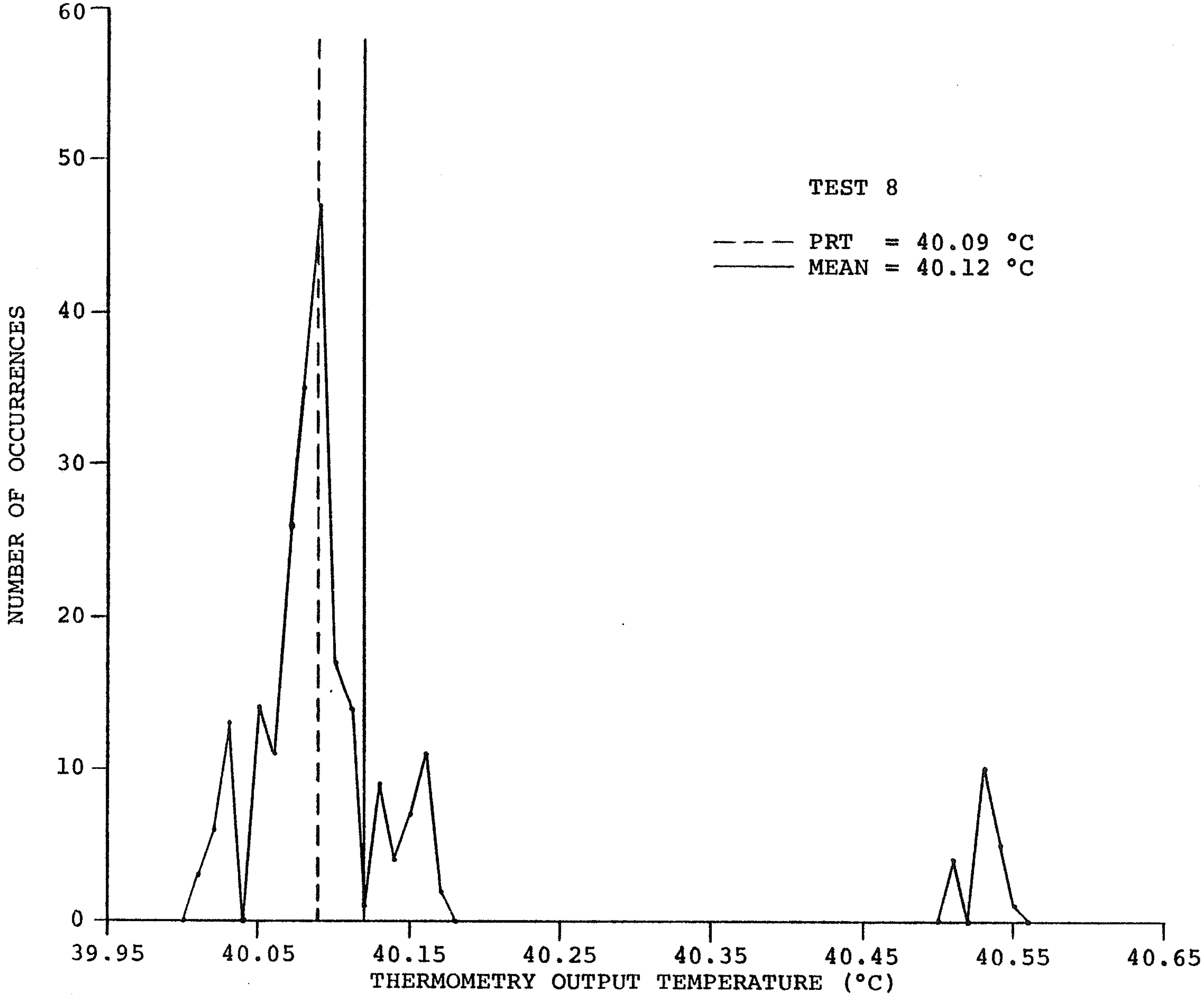


Figure 5.24. Distribution of output temperatures when probes are read every 180 seconds.

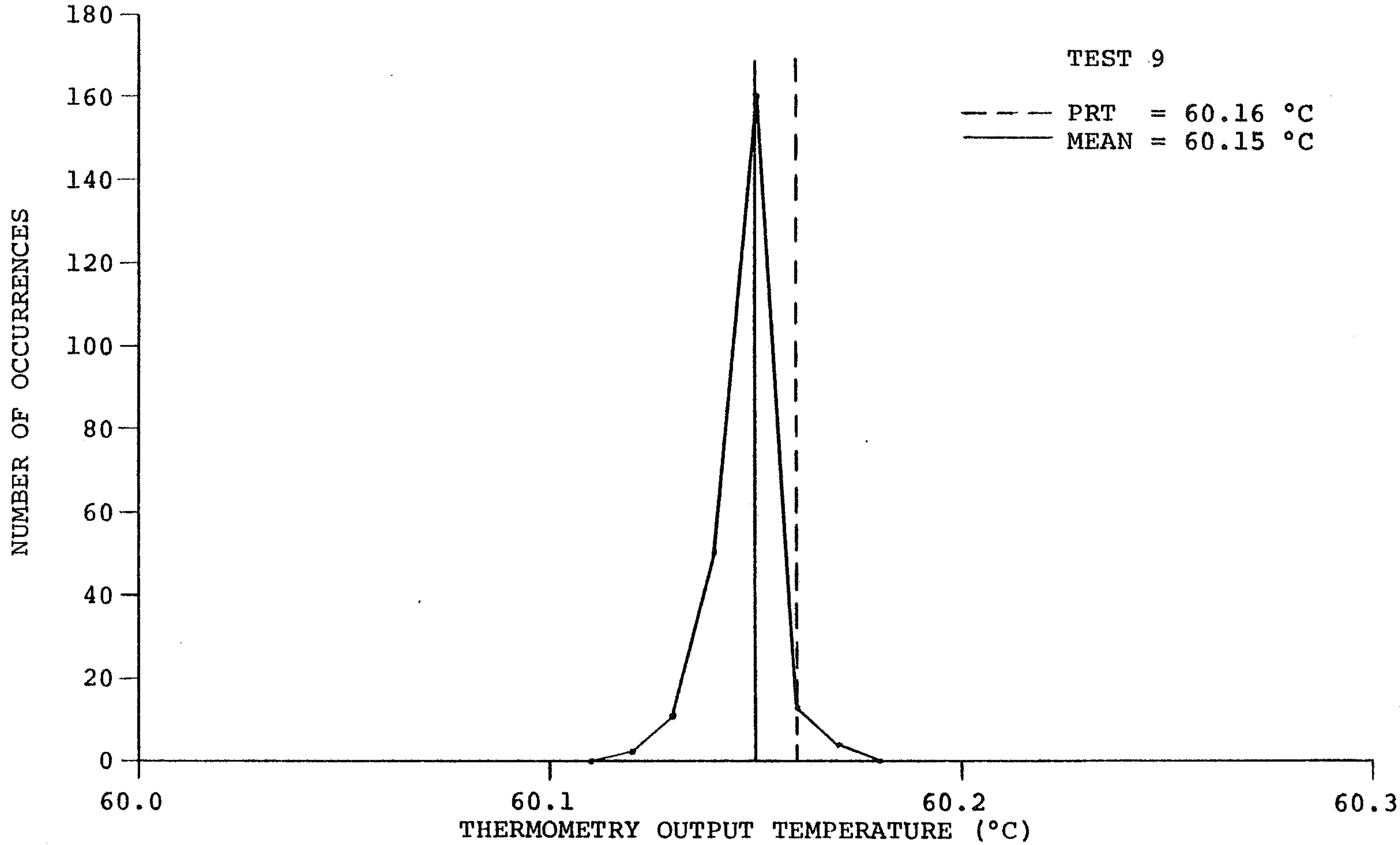


Figure 5.25. Distribution of output temperatures when probes are read every 10 seconds.

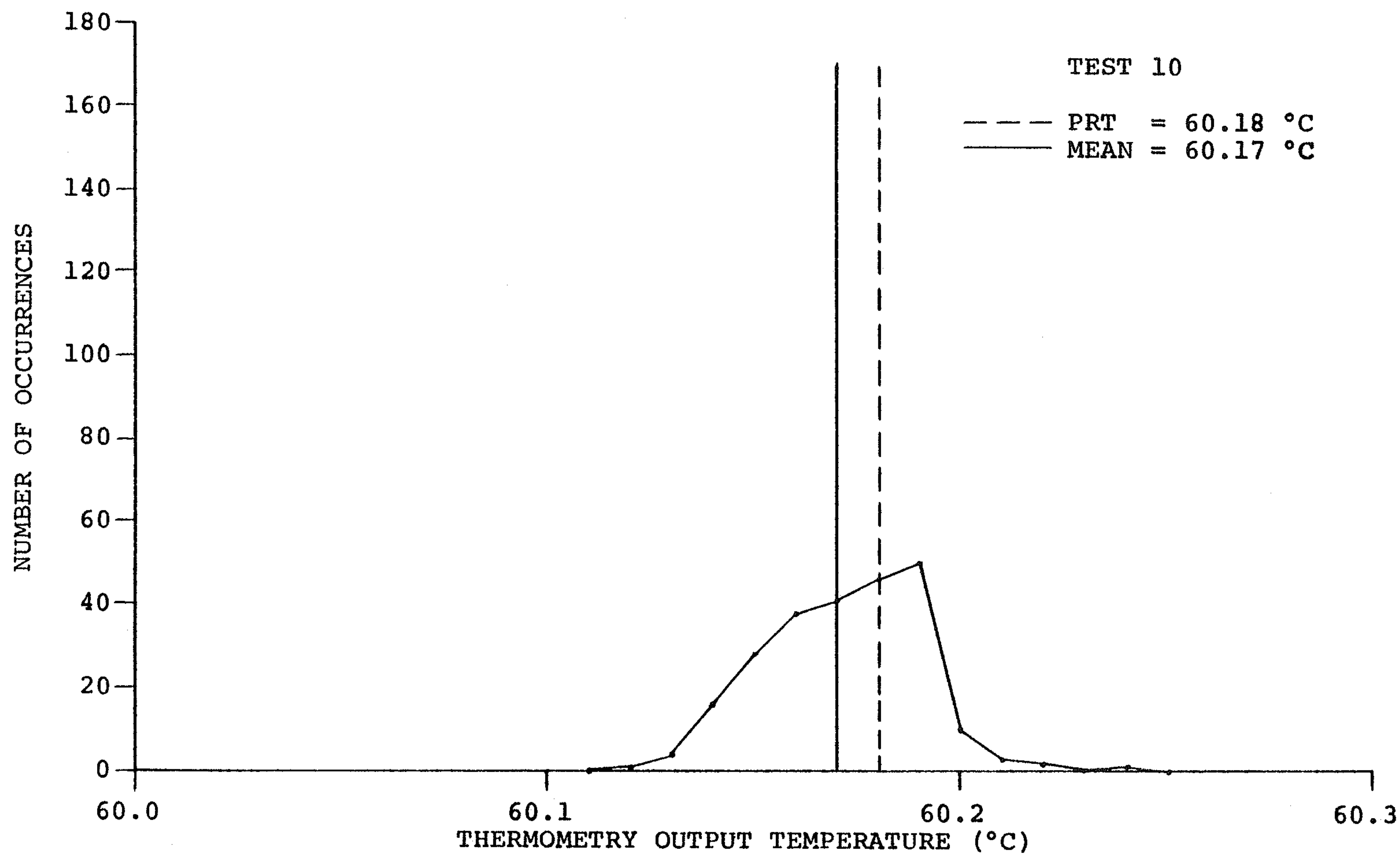


Figure 5.26. Distribution of output temperatures when probes are read every 36 seconds.

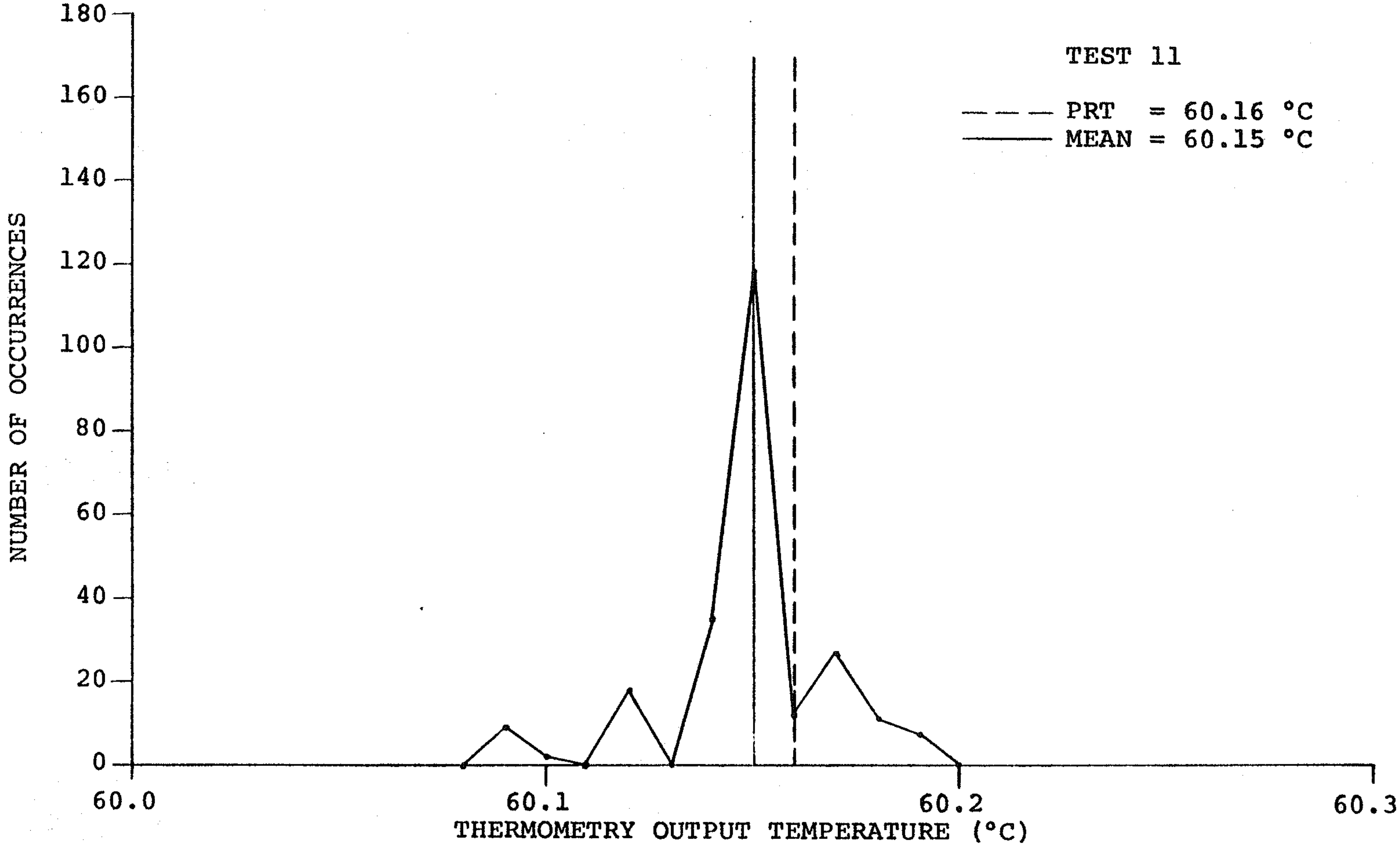


Figure 5.27. Distribution of output temperatures when probes are read every 60 seconds.

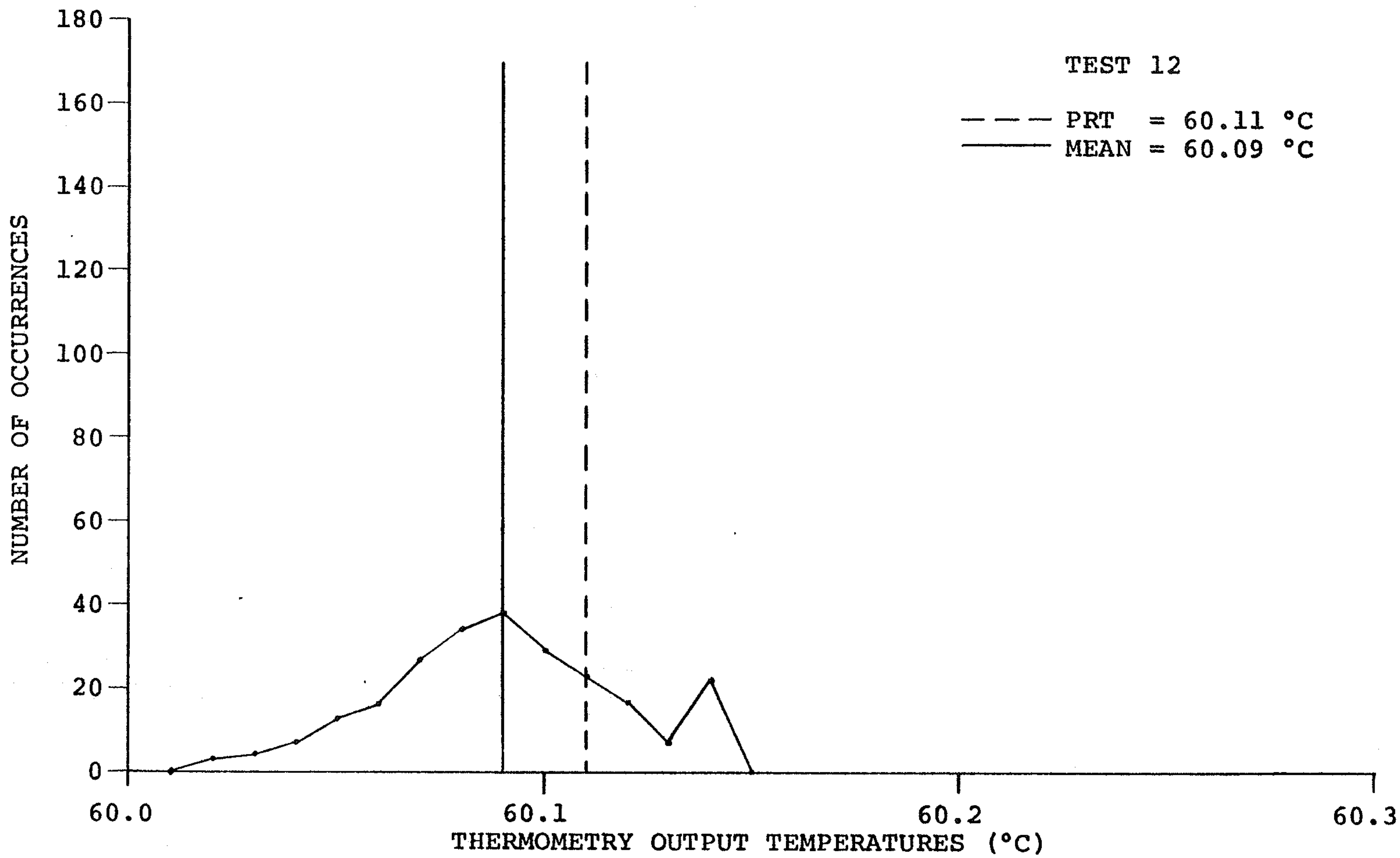


Figure 5.28. Distribution of output temperatures when probes are read every 180 seconds.

performance of the thermometry system. Figure 5.17 illustrates this point. In this plot, channels three and fifteen are responsible for all the readings below 19.85^{o}C, causing the mean to be less than the PRT temperature. However, it is seen that the "center" of the distribution is actually above the PRT temperature. In fact, most of the points are above the PRT temperature. This means that a typical reading of output temperature will exceed the PRT temperature with a probability of greater than 50%.

Similarly, in Figures 5.21 through 5.24, channel six produced readings far above the rest of the distribution, and had the effect of pulling the mean above the PRT temperature. Channel six continued to act suspiciously in the tests illustrated in Figures 5.25 through 5.28. For each of the four tests, its output remained fixed at the second calibration temperature for the duration of the test. This peculiar behavior was masked somewhat by the fact that the second calibration point was always near the PRT temperature of about 60.14^{o}C.

Qualitatively, one could determine the accuracy of the thermometry system, for a particular test, by how much the PRT temperature deviates from the center of the distribution. Similarly, the precision could be defined as the symmetric variation about the center of the distribution that encompasses two-thirds of the points. These definitions will de-emphasize the erratic channels and enhance the presentation of the good channels. These values of accuracy and precision are presented in Table 5.6.

Table 5.6. Rough Estimates of Accuracy and Precision from the Point Histograms.

Figure	Accuracy (°C)	Precision (°C)
5.17	0.01	0.02
5.18	0.01	0.03
5.19	0.00	0.01
5.20	0.01	0.01
5.21	0.00	0.02
5.22	0.00	0.04
5.23	0.02	0.03
5.24	0.00	0.03
5.25	0.01	0.00
5.26	0.01	0.02
5.27	0.01	0.01
5.28	0.02	0.03

On the whole, these values are smaller than those in Subsection 5.3.3, where no attempt was made to distinguish poor channels from good ones. However, they are very similar to the values obtained in Subsection 5.3.2, where only the five best channels were analyzed. This is as expected. Another interesting point is seen by dividing the histograms into three groups, corresponding to the PRT temperature during the tests. These groups are Figures 5.17 through 5.20, Figures 5.21 through 5.24, and Figures 5.25 through 5.28. The three groups correspond to water bath temperatures of approximately 20°C, 40°C, and 60°C, respectively. One can see that there is a definite trade-off between precision and accuracy between these groups.

5.4. Explanation of the Errors

When the calibration routine did not calibrate a particular channel properly, several possible errors might have occurred.

Small deviations in precision may be attributed to the slight temperature instability of the water bath. Small errors in accuracy may be due to errors caused by the manipulation of intermediate results in the calibration routine.

A spurious data point which was not consistent with the rest of the readings, as seen in Figure 5.9, was probably due to noise on the analog board. To eliminate such a problem, several readings of the digitized voltage from the analog-to-digital converter could be averaged. Then, this result could be used in the calculations.

When the readings were not accurate or precise, a serious problem with the probe or thermocouple amplifier chip may have existed. Another possibility is that the probe may have been placed in the water bath improperly. The position of the thermocouple tip in relation to the PRT probe may have been such that they were actually at different temperatures when it was assumed that they were at the same temperature.

It is possible for the readings of a particular channel to be inaccurate but remain precise. The twenty-four hour run contained such an example. As can be seen in Figure 5.10, all of the readings for channel six exceed the PRT temperature of 40.10^{o}C by greater than 0.10^{o}C. The reason for the larger than normal deviation is probably a bad calibration. In other words, one or both of the calibration voltages obtained from the analog board were most likely in error. If the probes were placed in the exact same calibration temperature again, the voltage might have been different. If this were the case, the gain and the offset calculated would have been based on a bad input point and every output temperature would deviate greatly from the true value. As a result, all of the readings deviate by about the same amount.

One way to check for such problems would be to verify that the calibration is successful immediately after the calibration is completed. The probes could be placed in a third temperature water bath, which could be at the same temperature as the first. This temperature would be entered and the probes read. If there is a discrepancy of more than, say, 0.1^{o}C, the particular channel would be considered to be uncalibrated. The option to recalibrate the faulty channels, without altering the correct channels, would be available. If a channel fails again, the probe or thermocouple chip is probably defective, and so the channel should not be used.

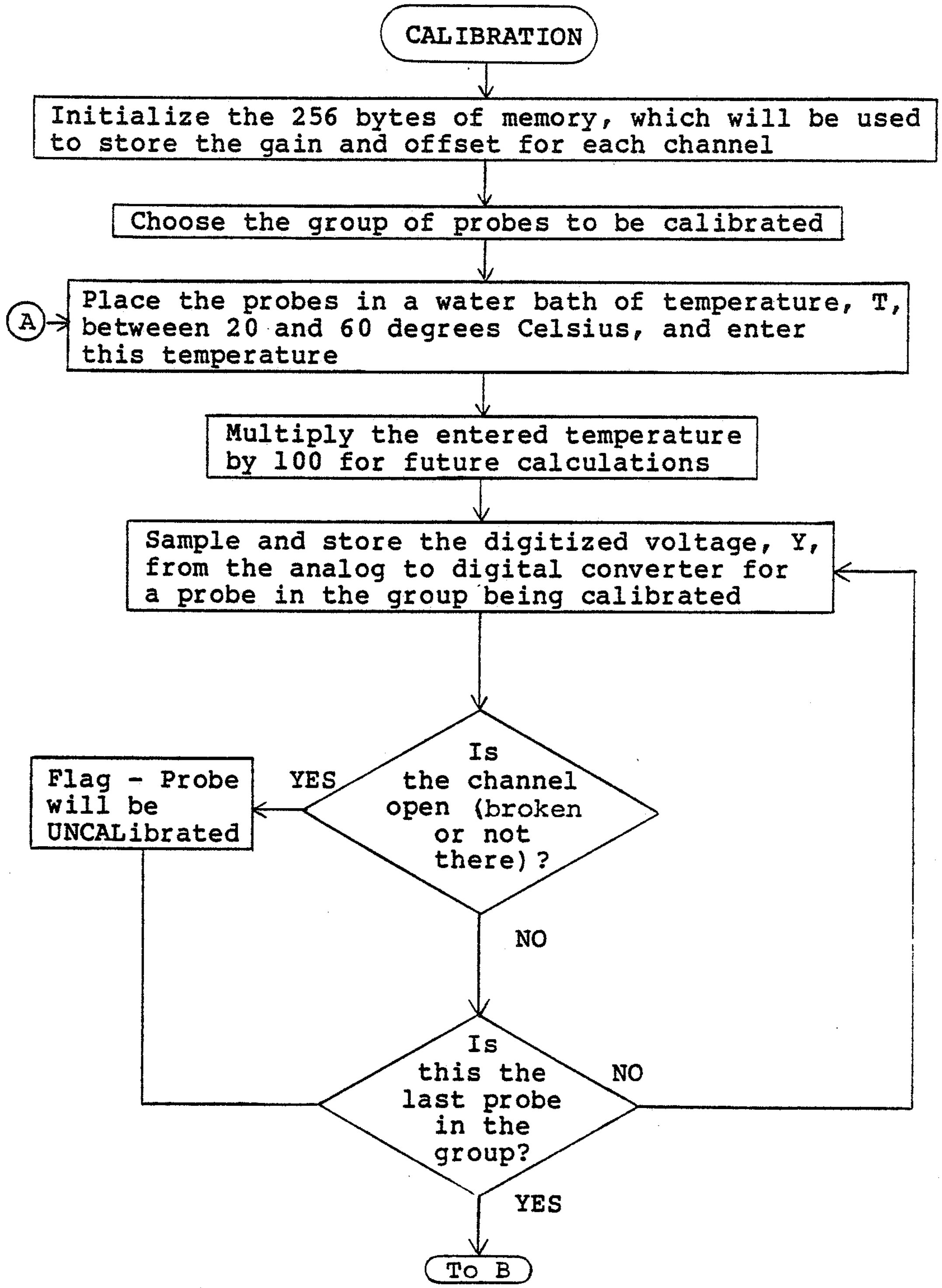
CHAPTER VI

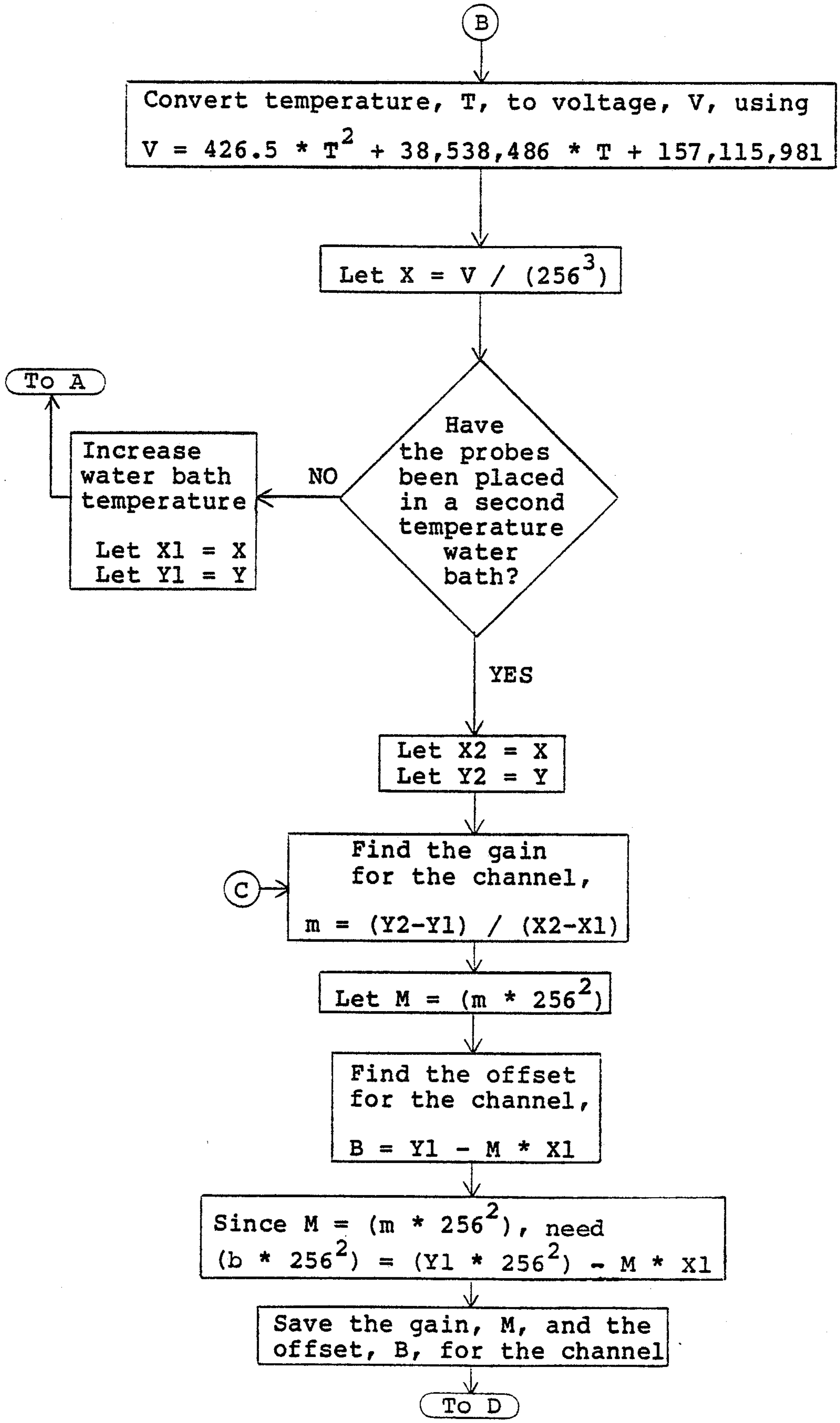
SUMMARY

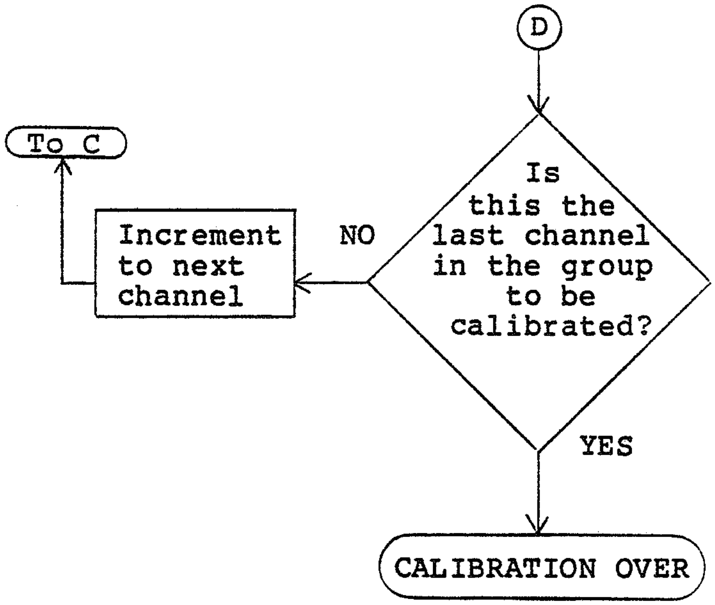
The double-point calibration routine for the thermometry system has been shown to function properly. The tests that have been performed indicate an accuracy of $\pm 0.01^{\circ}C$ and a precision of $\pm 0.03^{\circ}C$. In addition, the results are stable over long periods of time and repeatable under various conditions. Some of the reasons for unusual deviations have been discussed and a method of checking for incorrect calibrations has been suggested.

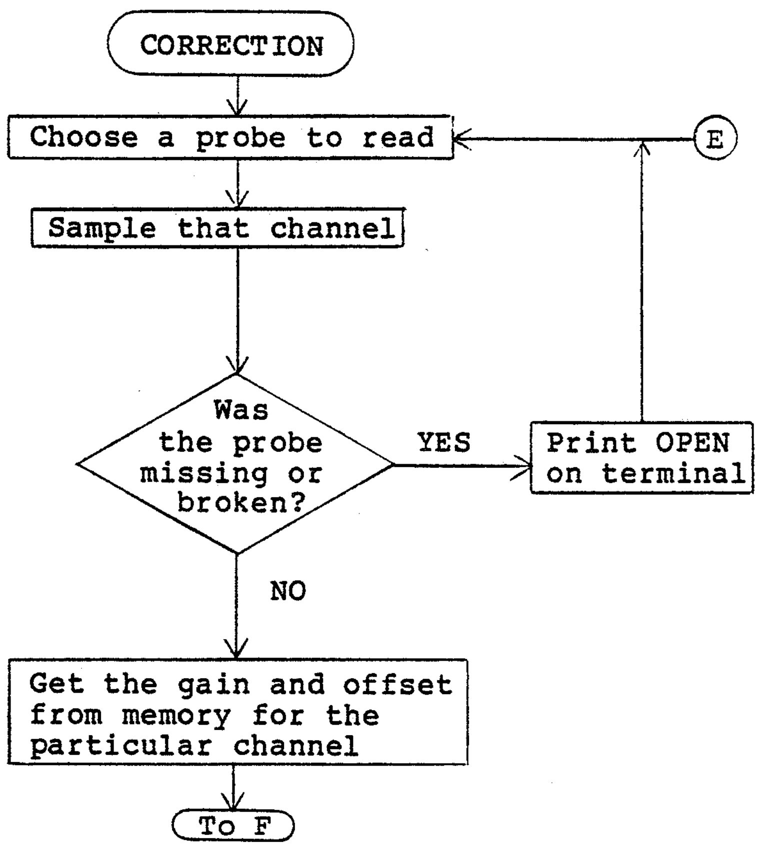
APPENDIX A

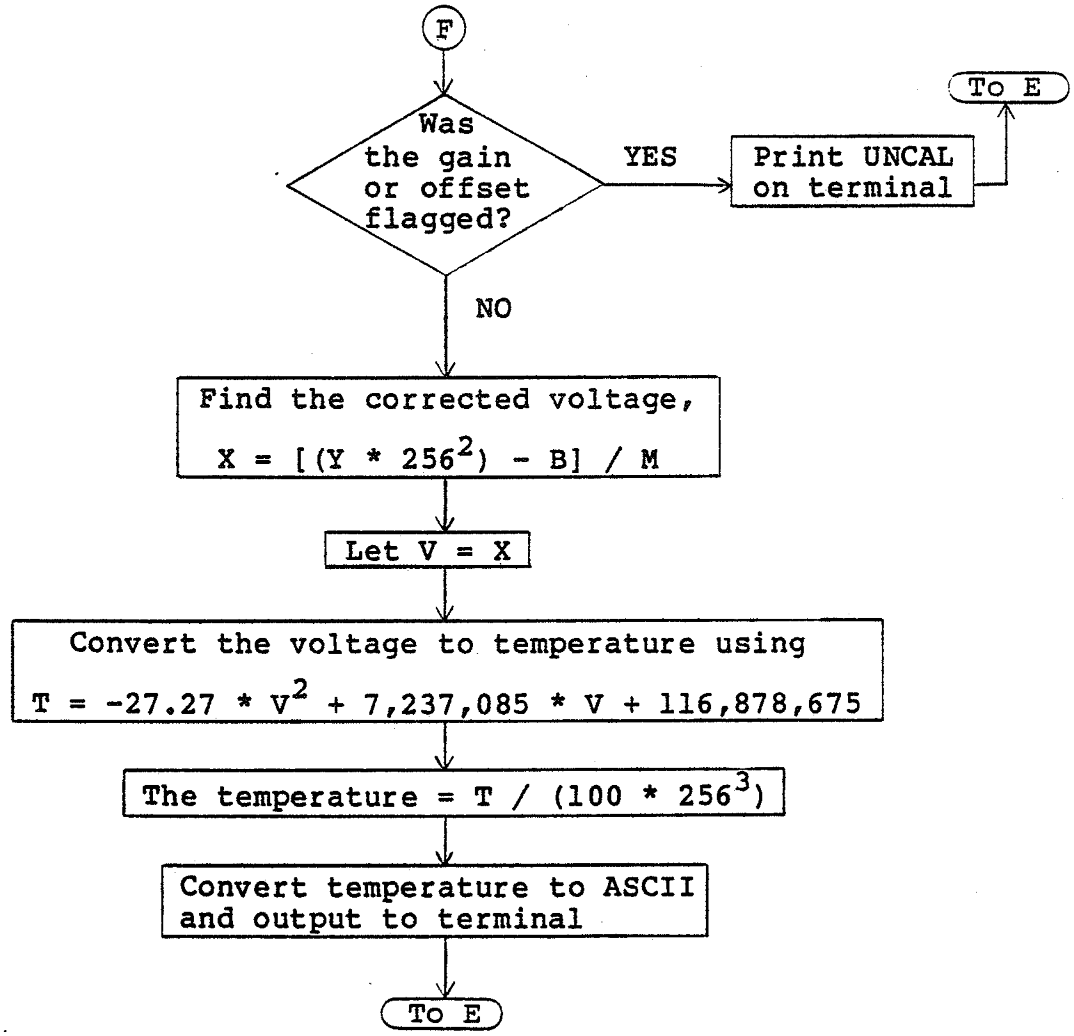
FLOW CHART FOR THE CALIBRATION ROUTINE




B
Convert temperature, T, to voltage, V, using
V = 426.5 * T2 + 38,538,486 * T + 157,115,981
Let X = V / (2563)
Have the probes been placed in a second temperature water bath?
NO
Increase water bath temperature
Let X1 = X
Let Y1 = Y
To A
YES
Let X2 = X
Let Y2 = Y
C
Find the gain for the channel,
m = (Y2-Y1) / (X2-X1)
Let M = (m * 2562)
Find the offset for the channel,
B = Y1 - M * X1
Since M = (m * 2562), need
(b * 2562) = (Y1 * 2562) - M * X1
Save the gain, M, and the offset, B, for the channel
To D


D
Is this the last channel in the group to be calibrated?
NO
Increment to next channel
To C
YES
CALIBRATION OVER


CORRECTION
Choose a probe to read
E
Sample that channel
Was the probe missing or broken?
YES
Print OPEN on terminal
NO
Get the gain and offset from memory for the particular channel
To F


F
Was the gain or offset flagged?
YES
Print UNCAL on terminal
To E
NO
Find the corrected voltage,
X = [(Y * 256^2) - B] / M
Let V = X
Convert the voltage to temperature using
T = -27.27 * V^2 + 7,237,085 * V + 116,878,675
The temperature = T / (100 * 256^3)
Convert temperature to ASCII and output to terminal
To E

APPENDIX B

PROGRAM LISTING WITH CALIBRATION

```
                ;**********************************************************
                ;*            16-CHANNEL THERMOMETRY SYSYTEM              *
                ;*                   WITH CALIBRATION                     *
                ;*                          BY                            *
                ;*                      MARY OZARKA                       *
                ;*      UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN        *
                ;*THIS PROGRAM CONTAINS ALFRED GHARAKHANI'S CONTROL CODE  *
                ;*FOR THE THERMOMETRY SYSTEM RELATED TO INITIALIZATION,   *
                ;*CHARACTER INPUT AND OUTPUT, AND SAMPLING OF THE A/D     *
                ;*CONVERTER.  THE CALIBRATION CODE HAS BEEN DEVELOPED FOR *
                ;*TYPE T THERMOCOUPLES.  THE MESSAGES ARE STORED BEGINNING *
                ;*AT 900 HEXADECIMAL IN MEMORY.  IN THE PRESENT FORM, SOME *
                ;*OF THE RESPONSES TO THE MESSAGES ARE NOT IN A           *
                ;*"USER-FRIENDLY" FORM.  THESE MAY BE EASILY MODIFIED.    *
                ;**********************************************************
                ;
                ;
                ;
                ;LABELS.
0094            ALTCH    EQU P1.4
0095            START    EQU P1.5
0096            RDY      EQU P1.6
0007            BELL     EQU $07
000A            LF       EQU $0A
0020            BLANK    EQU $20
003E            PRMT     EQU $3E
000D            CR       EQU $0D
002E            POINT    EQU $2E
0087            PCON     EQU $87
F000            DATAL    EQU $F000
E000            DATAH    EQU $E000
D000            PDATAL   EQU $D000
C000            PDATAH   EQU $C000
B000            DSPLY    EQU $B000
0000            ZERO     EQU $00
000A            TEN      EQU $0A
0020            TWEN     EQU $20
0021            TWEN1    EQU $21
0040            FORT     EQU $40
0041            FORT1    EQU $41
                ;
                ;
                ;THIS PORTION OF THE PROGRAM CONFIGURES THE 8031 TO
                ;PERFORM THE DESIRED FUNCTIONS. TO UNDERSTAND THE DETAILS
                ;REFER TO MCS-51 USERS MANUAL(CHAP. 2).   TO SIMPLIFY THIS
                ;PAGE NUMBERS ARE INCLUDED.
```

AVOCET SYSTEMS 8051 CROSS-ASSEMBLER - VERSION 1.09

SOURCE FILE NAME: CAL2.ASM

```
                        ;
0100                    INIT:   ORG $100        ;START LOADING THE PROGRAM FROM
                        ;100H.
0100 74FF                       MOV A,#$FF      ;TURN OFF THE 7 SEGMENTS
0102 90B000                     MOV DPTR,#DSPLY
0105 F0                         MOVX @DPTR,A
                        ;IN THE NEXT THREE LINES PORT1 WILL BE CONFIGURED. RDY (P1.6)
                        ;IS SET TO BE AN INPUT. ALTCH(P1.4) IS SET AND START(P1.5)
                        ;IS CLEARED
                        ;CHANNEL SELECT BITS ARE ALL CLEARED. PORT3 IS CONFIGURED
                        ;FOR ALTERNATE FUNCTIONS (REFER TO PP.13-15).
0106 759050                     MOV P1,#%01010000       ;MAKE RDY AN INPUT.
0109 43B0FF                     ORL P3,#%11111111
                        ;IN THE END INTERRUPT SYSTEM IS SET UP
                        ;       MOV IE,#%10000001       ;ENABLE INTO ONLY.
                        ;       MOV IP,#%00000001       ;INTO HAS HIGHEST PRIORITY.
                        ;FOR MORE INFO. ABOUT INTERRUPTS REFFER TO PP. 6-10
                        ;
                        ;
010C 1202A8                     CALL BAUD       ;SET UP THE SERIAL PORT.
010F 7407                       MOV A,#BELL
0111 1202F8                     CALL OUTCH      ;BEEP THE TERMINAL
0114 9009E5                     MOV DPTR,#$09E5
0117 120323                     CALL MESS       ;PRINT "URI TERM-X"
011A 12031C                     CALL PROMT      ;PROMT THE USER.
                        ;
                        ;
                        ;THE MAIN ROUTINE CALIBRATES THE SET OF PROBES REQUESTED.
                        ;THEN THE CALIBRATED TEMPERATURE OF A REQUESTED PROBE IS
                        ;DISPLAYED.
                        ;
011D                    MAIN    EQU $
                        ;INITIALLY LOAD THE 256 EXTERNAL MEMORY LOCATION USED TO
                        ;STORE GAIN AND OFFSET WITH #$99
011D 120177                     CALL NN
0120 120185                     CALL PROBE      ;GET THE CHANNELS TO BE CALIBRATED
0123 120336                     CALL CALIB      ;CALIBRATE THESE PROBES
0126 757000                     MOV $70,#ZERO
0129 12023C             MORE:   CALL CHNL       ;SELECT A CHANNEL TO OBSERVE
012C 8F71                       MOV $71,R7      ;FOR USE LATER
012E EF                 SAM:    MOV A,R7
012F 120275                     CALL SAMPLE     ;SAMPLE IT
                        ;CHECK IF PROBE MISSING WHEN SAMPLED
0132 E9                         MOV A,R1
0133 B4990C                     CJNE A,#$99,CHX
0136 EA                         MOV A,R2
0137 B49908                     CJNE A,#$99,CHX
                        ;PRINT O
013A 744F                       MOV A,#$4F
013C 1202F8                     CALL OUTCH
013F 020160                     JMP MORO        ;TRY AGAIN WITH WHICH PROBE?
```

```
                ;
0142 120711     CHX:    CALL CORR     ;CORRECTION BASED ON OFFSET AND GAIN
                ;CHECK IF THE CHANNEL IS UNCALIBRATED
0145 E57C               MOV A,$7C
0147 B4990D             CJNE A,#$99,FN1
014A E57D               MOV A,$7D
014C B49908             CJNE A,#$99,FN1
                ;PRINT U
014F 7455               MOV A,#$55
0151 1202F8             CALL OUTCH
0154 020160             JMP MORO
                ;
0157 12077A     FN1:    CALL TEMP     ;CONVERT THE VOLTAGE TO TEMPERATURE
                ;
015A 1207D7             CALL CONV     ;CONVERT TO ASCII
015D 120826             CALL OUT      ;DISPLAY TEMPERATURE TO TERMINAL
0160 E570       MORO:   MOV A,$70
0162 B40005             CJNE A,#ZERO,MOR1
0165 12030A             CALL CRLF
0168 80BF               JMP MORE
016A 7420       MOR1:   MOV A,#$20
016C 1202F8             CALL OUTCH
016F 0571               INC $71
0171 AF71               MOV R7,$71
0173 1570               DEC $70
0175 80B7               JMP SAM
                ;
                ;WHEN COMMUNICATION WITH A HOST COMPUTER IS IMPLENTED,
                ;THE IDEA WILL BE TO KEEP SAMPLING ALL 32 (16) PROBES AND
                ;STORE THE TEMPERATURES IN A BUFFER.
                ;
                ;
                ;THE NN ROUTINE LOADS THE EXTERNAL MEMORY LOCATION 0 THROUGH
                ;255 WITH #$99
                ;
0177            NN      EQU $
0177 900000             MOV DPTR,#$0000
017A 7499       NN1:    MOV A,#$99
017C F0                 MOVX @DPTR,A
017D 0582               INC DPL
017F E582               MOV A,DPL
0181 B400F6             CJNE A,#ZERO,NN1
0184 22                 RET
                ;
                ;THE PROBE ROUTINE ASKS THE USER FOR THE CHANNELS DESIRED
                ;(SPECIAL CODE).
                ;IT EXPECTS ONLY ONE CHARACTER TO BE ENTERED.
                ;THIS CHARACTER IS DECODED AND THE CHANNEL NUMBER OF THE
                ;FIRST PROBE TO BE CALIBRATED IS STORED IN LOCATION $70.
                ;THE CHANNEL NUMBER OF THE LAST PROBE TO BE CALIBRATED
                ;PLUS ONE IS STORED IN LOCATION $71.
```

AVOCET SYSTEMS 8051 CROSS-ASSEMBLER - VERSION 1.09

SOURCE FILE NAME: CAL2.ASM

```
                ;PROBE IS SETUP TO WORK ON A THERMOMETRY SYSTEM WITH UP
                ;TO 32 CHANNELS.
                ;
0185            PROBE    EQU $
0185 9009BF              MOV DPTR,#$09BF
0188 120323              CALL MESS
018B 120317              CALL ECHO
018E 12030A              CALL CRLF
                ;
0191 B46109              CJNE A,#$61,PR1
0194 757000              MOV $70,#ZERO
0197 757104              MOV $71,#$04
019A 02023B              JMP PRE
                ;
019D B46209     PR1:     CJNE A,#$62,PR2
01A0 757004              MOV $70,#$04
01A3 757108              MOV $71,#$08
01A6 02023B              JMP PRE
                ;
01A9 B46309     PR2:     CJNE A,#$63,PR3
01AC 757008              MOV $70,#$08
01AF 75710C              MOV $71,#$0C
01B2 02023B              JMP PRE
                ;
01B5 B46409     PR3:     CJNE A,#$64,PR4
01B8 75700C              MOV $70,#$0C
01BB 757110              MOV $71,#$10
01BE 02023B              JMP PRE
                ;
01C1 B46509     PR4:     CJNE A, #$65,PR5
01C4 757000              MOV $70,#ZERO
01C7 757108              MOV $71,#$08
01CA 02023B              JMP PRE
                ;
01CD B46609     PR5:     CJNE A,#$66,PR6
01D0 757008              MOV $70,#$08
01D3 757110              MOV $71,#$10
01D6 02023B              JMP PRE
                ;
01D9 B46709     PR6:     CJNE A,#$67,PR7
01DC 757000              MOV $70,#ZERO
01DF 757110              MOV $71,#$10
01E2 02023B              JMP PRE
                ;
01E5 B46809     PR7:     CJNE A,#$68,PR8
01E8 757010              MOV $70,#$10
01EB 757114              MOV $71,#$14
01EE 02023B              JMP PRE
                ;
01F1 B46909     PR8:     CJNE A,#$69,PR9
01F4 757014              MOV $70,#$14
```

SOURCE FILE NAME: CAL2.ASM

```
 01F7 757118            MOV $71,#$18
 01FA 02023B            JMP PRE
                  ;
 01FD B46A09      PR9:     CJNE A,#$6A,PR10
 0200 757018            MOV $70,#$18
 0203 75711C            MOV $71,#$1C
 0206 02023B            JMP PRE
                  ;
 0209 B46B09      PR10:    CJNE A,#$6B,PR11
 020C 75701C            MOV $70,#$1C
 020F 757120            MOV $71,#$20
 0212 02023B            JMP PRE
                  ;
 0215 B46C09      PR11:    CJNE A,#$6C,PR12
 0218 757010            MOV $70,#$10
 021B 757118            MOV $71,#$18
 021E 02023B            JMP PRE
                  ;
 0221 B46D09      PR12:    CJNE A,#$6D,PR13
 0224 757018            MOV $70,#$18
 0227 757120            MOV $71,#$20
 022A 02023B            JMP PRE
                  ;
 022D B46E09      PR13:    CJNE A,#$6E,PR14
 0230 757010            MOV $70,#$10
 0233 757120            MOV $71,#$20
 0236 02023B            JMP PRE
                  ;
 0239 2185        PR14:    JMP PROBE      ;INVALID INPUT SO TRY AGAIN
                  ;
 023B 22          PRE:     RET
                  ;
                  ;
                  ;
                  ;THE CHNL ROUTINE ASKS THE USER TO ENTER THE PROBE NUMBER
                  ;FOR A TEMPERATURE READING.
                  ;CHNL IS SETUP TO WORK ON A THERMOMETRY SYSTEM WITH UP TO 32
                  ;CHANNELS.
                  ;MAY ENTER VALUES 0 THROUGH 31 USING THE SPECIAL CODE (PAGE 52)
                  ;THE VALUE WILL BE PUT IN R7.
                  ;THERE IS A CHECK FOR INVALID INPUTS.
                  ;
 023C             CHNL     EQU $
 023C 900950            MOV DPTR,#$0950
 023F 120323            CALL MESS            ;PRINT THE MESSAGE.
 0242 120317            CALL ECHO
 0245 F5F0              MOV B,A
 0247 54F0              ANL A,#$F0
 0249 B43008            CJNE A,#$30,FR
 024C E5F0              MOV A,B
 024E 540F              ANL A,#$0F
```

AVOCET SYSTEMS 8051 CROSS-ASSEMBLER - VERSION 1.09

SOURCE FILE NAME: CAL2.ASM

```
0250 FF                 MOV R7,A        ;HAVE PROBE NUMBER IN R7
0251 02025D             JMP FRE
0254 B42010     FR:     CJNE A,#$20,ERFR
0257 E5F0               MOV A,B
0259 C3                 CLR C
025A 9410               SUBB A,#$10
025C FF                 MOV R7,A        ;HAVE PROBE NUMBER IN R7
025D 120300     FRE:    CALL INCH
0260 B40DD9             CJNE A,#CR,CHNL
0263 12030A             CALL CRLF
0266 22                 RET
                ;
0267 E5F0       ERFR:   MOV A,B
0269 B441D0             CJNE A,#$41,CHNL
026C 757007             MOV $70,#$07
026F 7F00               MOV R7,#ZERO
0271 12030A             CALL CRLF
0274 22                 RET
                ;
                ;
                ;THE SAMPLE ROUTINE RECEIVES A CHANNEL NUMBER IN THE ACC.
                ;THE REQUESTED CHANNEL IS SAMPLED AND THE 12-BIT NUMBER FROM
                ;THE A/D IS RETURNED IN (R1,R2)
                ;
                ;ALSO, BROKEN T-COUPLES IN THE CHANNELS BEING SAMPLED
                ;WILL BE DETECTED AND FLAGGED, (R1,R2) = #$9999
                ;
0275            SAMPLE  EQU $
0275 D294               SETB ALTCH
0277 5390F0             ANL P1,#$F0     ;CLEAR PREVIOUS CHANNEL NUMBER
027A 4290               ORL P1,A        ;SEND OUT CHANNEL NUMBER TO AIS
027C C294               CLR ALTCH       ;LATCH THE CHANNEL NUMBER
027E D295               SETB START
0280 C295               CLR START       ;START THE A/D
                ;
                ;AT THIS POINT THE A/D WILL START ITS 25 MICROSECOND
                ;CONVERSION.
0282 90E000             MOV DPTR,#DATAH
0285 3096FD             JNB RDY,$       ;WAIT UNTIL THE END OF CONVERSION
0288 E0                 MOVX A,@DPTR    ;LOAD ACC WITH HIGHEST 8 BITS
0289 F9                 MOV R1,A
028A 90F000             MOV DPTR,#DATAL
028D E0                 MOVX A,@DPTR    ;LOAD ACC WITH THE LOWER 4 BITS
028E FA                 MOV R2,A
028F 30E305             JNB ACC.3,NOERR         ;CHECK FOR OPEN CHANNEL
                ;
0292 7999       ERR:    MOV R1,#$99     ;THE SAMPLED CHANNEL IS OPEN
0294 7A99               MOV R2,#$99
0296 22                 RET
                ;
                ;SHIFT BITS IN (R1,R2) RIGHT 4 TIMES APPENDING ZEROS ON LEFT
```

```
                ;
0297 EA         NOERR:  MOV A,R2
0298 54F0               ANL A,#$F0
029A C4                 SWAP A
029B FA                 MOV R2,A
029C E9                 MOV A,R1
029D 540F               ANL A,#$0F
029F C4                 SWAP A
02A0 4A                 ORL A,R2
02A1 FA                 MOV R2,A
02A2 E9                 MOV A,R1
02A3 54F0               ANL A,#$F0
02A5 C4                 SWAP A
02A6 F9                 MOV R1,A
                ;
                ;TST1:  MOV A,R1        ;FOR TEST
                ;       MOV B,R2        ;FOR TEST
                ;       CALL TEST3      ;FOR TEST
                ;
02A7 22                 RET
                ;
                ;
                ;
                ;THE FOLLOWING SUBROUTINE WILL FIGURE OUT THE BAUD RATE
                ;SETTING. A PROPER RELOAD VALUE WILL BE LOADED ACCORDINGLY.
                ;IN THE END COUNTER1 WILL BE STARTED. FOR MORE INFO.
                ;ABOUT SERIAL PORT SET UP REFER TO PP.18-28 .
02A8 758801     BAUD:   MOV TCON,#%00000001
02AB 759850             MOV SCON,#%01010000
02AE 758920             MOV TMOD,#%00100000
02B1 758780             MOV PCON,#$80  ;DOUBLE THE BAUD RATE
02B4 90F000             MOV DPTR,#DATAL
02B7 E0                 MOVX A,@DPTR   ;READ THE SETTING.
02B8 5407               ANL A,#%00000111
02BA B40006             CJNE A,#$0,B600
02BD 758D30             MOV TH1,#$30   ;300 BAUD.
02C0 0202F3             JMP GOTIT
02C3 B40106     B600:   CJNE A,#$1,B1200
02C6 758D98             MOV TH1,#$98   ;600 BAUD
02C9 0202F3             JMP GOTIT
02CC B40206     B1200:  CJNE A,#$2,B2400
02CF 758DCC             MOV TH1,#$CC   ;1200 BAUD.
02D2 0202F3             JMP GOTIT
02D5 B40306     B2400:  CJNE A,#$3,B4800
02D8 758DE6             MOV TH1,#$E6   ;2400 BAUD.
02DB 0202F3             JMP GOTIT
02DE B40406     B4800:  CJNE A,#$4,B9600
02E1 758DF3             MOV TH1,#$F3   ;4800 BAUD.
02E4 0202F3             JMP GOTIT
02E7 B40506     B9600:  CJNE A,#$5,B192
02EA 758DF9             MOV TH1,#$F9   ;9600 BAUD.
```

```
02ED 0202F3              JMP GOTIT
02F0 758DFD     B192:    MOV TH1,#$FD    ;19.2K BAUD.
02F3 D28E       GOTIT:   SETB TR1        ;START THE TIMER.
02F5 D299                SETB TI         ;SET TI FOR 1ST CHARACTER.
02F7 22                  RET
                ;OUTCH: IS A SUBROUTINE WHICH SENDS OUT THE CONTENT OF
                ;THE ACC. TO THE SERIAL BUFFER OF THE 8031,ACC. MUST
                ;CONTAIN THE 7BIT ASCII REPRESENTATION OF DESIRED
                ;CHARACTER.
                ;NOTICE THAT NO PARITY IS USED.
                ;VALUE OF ACC. REMAINS UNTOUCHED.
                ;
02F8 3099FD     OUTCH:   JNB TI,$        ;WAIT UNTIL THE END OF THE PREVIOUS
                ;TRANSMISSION.
02FB C299                CLR TI          ;GET READY FOR THE CURRENT CHAR.
02FD F599                MOV SBUF,A      ;SEND THE CURRENT CHAR.
02FF 22                  RET
                ;
                ;
                ;
                ;INCH: IS A SUBROUTINE WHICH RECEIVES AN ASCII CHAR. FROM
                ;THE SERIAL BUFFER OF THE 8031. THIS CHAR. WILL BE PLACED
                ;IN THE ACC.IN THE END ACC.7 WILL BE CLEARED. PARITY IS
                ;IGNORED.
                ;
                ;
0300 3098FD     INCH:    JNB RI,$        ;WAIT FOR RECEIVE INTERRUPT
0303 C298                CLR RI          ;CLEAR THE RECEIVE INTERRUPT.
0305 E599                MOV A,SBUF
0307 547F                ANL A,#$7F
0309 22                  RET
                ;
                ;
                ;
                ;CRLF: THIS IS A SUBROUTINE WHICH SENDS OUT A CR FOLLOWED
                ;BY A LF.
                ;THE VALUE IN THE ACCUM. DOES NOT CHANGE
                ;
                ;
030A F51A       CRLF:    MOV $1A,A
030C 740D                MOV A,#CR
030E 51F8                CALL OUTCH
0310 740A                MOV A,#LF
0312 51F8                CALL OUTCH
0314 E51A                MOV A,$1A
0316 22                  RET
                ;
                ;
                ;
                ;ECHO: IS A SUBROUTINE WHICH RECEIVES CHAR FROM THE
                ;KEYBOARD AND SENDS IT TO THE SCREEN.ACC. WILL PRESERVE
```

```
                ;THE RECEIVED CHAR.
                ;
                ;
                ;
0317 7100       ECHO:   CALL INCH       ;GET THE CHAR. FROM THE KEYBOARD.
0319 51F8               CALL OUTCH      ;ECHO THE CHARTO THE SCREEN
031B 22                 RET
                ;
                ;
                ;
                ;PROMT: IS A SUBROUTINE WHICH PRINTS A PROMT IN THE
                ;BEGINNING OF THE NEXT LINE.
                ;
                ;
                ;
031C 710A       PROMT:  CALL CRLF
031E 743E               MOV A,#PRMT
0320 51F8               CALL OUTCH
0322 22                 RET
                ;
                ;
                ;
                ;MESS: SUBROUTINE WILL PRINT DIFFERENT MESSAGES ON THE
                ;SCREEN. THESE MESSAGES ARE STORED IN THE PROGRAM MEMORY
                ;(900H-A00H). EACH MESSAGE IS IDENTIFIED BY THE ADDRESS OF
                ;ITS FIRST CHARACTER. THE LAST CHAR. OF EACH MESSAGE MUST
                ;BE A CR. THIS ROUTINE EXPECTS THE ADDRESS OF MESSAGE IN
                ;THE DPTR.IN THE END ACC. AND DPTR WILL BE CLEARED.
                ;
                ;
0323 710A       MESS:   CALL CRLF
0325 7C00               MOV R4,#$00
0327 EC         NEXT:   MOV A,R4          ;SAVE THE INDEX
0328 93                 MOVC A,@A+DPTR
0329 B40D03             CJNE A,#CR,AGAIN ;STOP IF CR
032C 020334             JMP FINISH
032F 51F8       AGAIN:  CALL OUTCH      ;DISPLAY THE CHAR.
0331 0C                 INC R4          ;INCREMENT THE INDEX.
0332 80F3               JMP NEXT
0334 E4         FINISH: CLR A
0335 22                 RET
                ;
                ;
                ;
                ;THE CALIB ROUTINE GENERATES A CORRECTION GAIN AND OFFSET
                ;FOR EACH CHANNEL.  THESE VALUES ARE STORED IN EXTERNAL
                ;NONVOLATILE RAM.
                ;
                ;THE NUMBER OF THE FIRST PROBE TO BE CALIBRATED IS PASSED
                ;IN LOCATION $70 AND THE LAST PROBE TO BE CALIBRATED PLUS
                ;ONE IS PASSED IN LOCATION $71.
```

SOURCE FILE NAME: CAL2.ASM

```
                ;MEMORY LOCATION $72 WILL HOLD A NUMBER COUNTER.
                ;
0336            CALIB   EQU $
0336 900900             MOV DPTR,#$0900        ;SETUP FOR FIRST MESSAGE
0339 1203A8             CALL MT
                ;
                ;MT SENDS OUT THE FIRST MESSAGE AND RETURNS THE INPUT
                ;TEMPERATURE, X1 IN (R4,R5)
                ;
033C 7820               MOV R0,#TWEN   ;BEGIN STORING AT $20
033E 857072             MOV $72,$70    ;INITIALIZE $72 = FIRST CHANNEL
0341 12044E             CALL LOAD
                ;LOAD TEMP1, Y1, 12 BIT NUMBER FROM A/D FOR ALL CHANNELS
                ;TO CALIBRATE.
                ;FIND THE VOLTAGE CORRESPONDING TO TEMPERATURE X1 IN (R4,R5)
0344 120460             CALL VOLT
                ;VOLT RETURNS THE VOLTAGE IN (R3,R4)
                ;STORE IN ($7E,$7F) SINCE VOLT IS USED AGAIN
0347 8B7E               MOV $7E,R3
0349 8C7F               MOV $7F,R4
                ;
034B 900968             MOV DPTR,#$0968
034E 1203A8             CALL MT
                ;MT SENDS OUT THE SECOND MESSAGE AND RETURNS THE TEMPERATURE
                ;X2 IN (R4,R5)
0351 7840               MOV R0,#FORT   ;BEGIN STORING AT $40
0353 857072             MOV $72,$70    ;INITIALIZE,$72 = FIRST CHANNEL
0356 12044E             CALL LOAD
                ;LOAD TEMP2, Y2, 12 BIT NUMBER FROM A/D FOR ALL CHANNELS
                ;TO BE CALIBRATED
                ;FIND THE VOLTAGE CORRESPONDING TO TEMPERATURE X IN (R4,R5)
0359 120460             CALL VOLT
                ;
                ;VOLT RETURNS THE VOLTAGE IN (R3,R4).
                ;REMEMBER X2 > X1 SO Y2 > Y1 TOO.
                ;FIRST FIND (X2-X1) = (R3,R4)-($7E,$7F) AND STORE IN
                ;($74,$7A) BECAUSE WILL BE NEEDED FOR OTHER CHANNELS.
                ;
035C C3                 CLR C
035D EC                 MOV A,R4
035E 957F               SUBB A,$7F
0360 F57A               MOV $7A,A
0362 EB                 MOV A,R3
0363 957E               SUBB A,$7E     ;WITH BORROW
0365 F574               MOV $74,A
                ;
                ;FIND THE GAIN = (Y2-Y1)/(X2-X1)
0367 857072             MOV $72,$70
036A 7821               MOV R0,#TWEN1
036C 7941               MOV R1,#FORT1
036E AC74       LPP:    MOV R4,$74
```

AVOCET SYSTEMS 8051 CROSS-ASSEMBLER - VERSION 1.09

SOURCE FILE NAME: CAL2.ASM

```
0370 AD7A               MOV R5,$7A
0372 1205DD             CALL GAIN
               ;THE GAIN IS RETURNED IN ($7B,$7C,$7D)
               ;FIND THE OFFSET
0375 12069C             CALL INTC
               ;OFFSET RETURNED IN ($75,$76,$77,$78,$79)
               ;STORE THE GAIN AND OFFSET IN EXTERNAL RAM
0378 1206E6             CALL SAVE
               ;SETUP FOR THE NEXT CHANNEL
037B 08                 INC R0
037C 08                 INC R0
037D 08                 INC R0
037E 09                 INC R1
037F 09                 INC R1
0380 09                 INC R1
               ;
0381 0572               INC $72
0383 E572               MOV A,$72
0385 B571E6             CJNE A,$71,LPP
0388 22                 RET           ;END
               ;
               ;
               ;THE TEST3 ROUTINE IS FOR DEBUGGING PURPOSES ONLY.
               ;THE CONTENTS OF (A,B) ARE SENT TO THE ERROR DISPLAY ON THE
               ;FRONT OF THE THERMOMETRY SYSTEM.
0389           TEST3    EQU $
0389 F51C               MOV $1C,A
038B 85831A             MOV $1A,DPH
038E 85821B             MOV $1B,DPL
0391 90B000             MOV DPTR,#DSPLY
0394 F0                 MOVX @DPTR,A
0395 7100               CALL INCH
0397 E5F0               MOV A,B
0399 90B000             MOV DPTR,#DSPLY
039C F0                 MOVX @DPTR,A
039D 7100               CALL INCH
039F E51C               MOV A,$1C
03A1 851A83             MOV DPH,$1A
03A4 851B82             MOV DPL,$1B
03A7 22                 RET
               ;
               ;
               ;THE MT ROUTINE SENDS OUT A MESSAGE AND RECEIVES THE
               ;TEMPERATURE VALUE IN THE FORM 23.64, FOR EXAMPLE, FOLLOWED
               ;BY A CR.  THIS ROUTINE IS FOR THE STAND ALONE UNIT ONLY.
               ;THE TEMPERATURE IS MULTIPLIED BY 100 AND RETURNED IN (R4,R5)
               ;THE TEMPERATURE MUST BE IN THE RANGE BETWEEN 0 AND 99.99
               ;THERE IS NO CHECK THAT THE CHARACTERS TYPED IN ARE NUMBERS.
               ;
03A8           MT       EQU $
03A8 7C00               MOV R4,#ZERO
```

```
03AA 7D00                MOV R5,#ZERO   ;CLEAR R4 AND R5
03AC 7123      REDO:     CALL MESS      ;PLACE PROBES IN WATER BATH
03AE 7117                CALL ECHO      ;RETURNS CHARACTER IN ACC.
03B0 B40D02              CJNE A,#CR,NT1
03B3 80F7                JMP REDO       ;TRY AGAIN
               ;
03B5 B42E03    NT1:      CJNE A,#POINT,NT2
03B8 0203E9              JMP PT         ;FIRST CHARACTER IS A DECIMAL POINT
               ;
03BB 540F      NT2:      ANL A,#$0F     ;CONVERT FROM ASCII TO HEX
03BD FD                  MOV R5,A
03BE 7117                CALL ECHO      ;INPUT SECOND CHARACTER
03C0 B40D03              CJNE A,#CR,NT3
03C3 020443              JMP ET100      ;MULTIPLY BY 100 TO GET VALUE
               ;
03C6 B42E03    NT3:      CJNE A,#POINT,NT4
03C9 0203F5              JMP PTS
               ;
03CC 540F      NT4:      ANL A,#$0F
03CE FE                  MOV R6,A
03CF ED                  MOV A,R5
03D0 75F00A              MOV B,#TEN
03D3 A4                  MUL AB         ;B HERE SHOULD BE ZERO
03D4 C3                  CLR C
03D5 2E                  ADD A,R6
03D6 FD                  MOV R5,A
03D7 AEF0                MOV R6,B       ;LIKE CLR R6 SINCE B IS ZERO
               ;
03D9 7117                CALL ECHO
03DB B40D03              CJNE A,#CR,NT5
03DE 020443              JMP ET100      ;MULTIPLY BY 100 TO GET VALUE
               ;
03E1 B42E03    NT5:      CJNE A,#POINT,NT6
03E4 0203F5              JMP PTS
               ;
03E7 80C3      NT6:      JMP REDO       ;TRY AGAIN
               ;
03E9 7117      PT:       CALL ECHO
03EB B42E02              CJNE A,#POINT,PTER
03EE 80BC                JMP REDO
03F0 B40D0F    PTER:     CJNE A,#CR,ET2
03F3 80B7                JMP REDO
               ;
03F5 7117      PTS:      CALL ECHO
03F7 B42E02              CJNE A,#POINT,PTST
03FA 80B0                JMP REDO
03FC B40D03    PTST:     CJNE A,#CR,ET2
03FF 020443              JMP ET100      ;MULTIPLY BY 100 TO GET VALUE
               ;
0402 540F      ET2:      ANL A,#$0F
0404 FE                  MOV R6,A
```

```
0405 ED                 MOV A,R5
0406 75F00A             MOV B,#TEN
0409 A4                 MUL AB
040A ACF0               MOV R4,B
040C C3                 CLR C
040D 2E                 ADD A,R6
040E FD                 MOV R5,A
              ;NOW (R4,R5) HOLD THE TEMPERATURE TO THE TENTH PLACE * 10
040F 5001               JNC ET3
0411 0C                 INC R4
              ;
0412 7117     ET3:      CALL ECHO
0414 B40D12             CJNE A,#CR,NT7
              ;MULTIPLY BY TEN
0417 ED                 MOV A,R5
0418 75F00A             MOV B,#TEN
041B A4                 MUL AB
041C FD                 MOV R5,A
041D EC                 MOV A,R4
041E ACF0               MOV R4,B
0420 75F00A             MOV B,#TEN
0423 A4                 MUL AB          ;B SHOULD BE ZERO
0424 2C                 ADD A,R4        ;SHOULD NOT CAUSE A CARRY
0425 FC                 MOV R4,A
0426 02044B             JMP ETT
              ;
0429 540F     NT7:      ANL A,#$0F
042B FE                 MOV R6,A
042C ED                 MOV A,R5
042D 75F00A             MOV B,#TEN
0430 A4                 MUL AB
0431 C3                 CLR C
0432 2E                 ADD A,R6
0433 FD                 MOV R5,A
0434 AFF0               MOV R7,B
0436 5001               JNC NT8
0438 0F                 INC R7
              ;
0439 EC       NT8:      MOV A,R4
043A 75F00A             MOV B,#TEN
043D A4                 MUL AB          ;B SHOULD EQUAL ZERO
043E 2F                 ADD A,R7
043F FC                 MOV R4,A
0440 02044B             JMP ETT
              ;
0443 ED       ET100:    MOV A,R5
0444 75F064             MOV B,#$64
0447 A4                 MUL AB
0448 ACF0               MOV R4,B
044A FD                 MOV R5,A        ;HAVE VALUE
              ;
```

```
044B 710A      ETT:    CALL CRLF
               ;REALLY ONLY WANT LF SINCE ALREADY HAVE CR
044D 22                RET            ;RETURNS TEMP IN (R4,R5)
               ;
               ;
               ;THE LOAD ROUTINE LOADS MEMORY LOCATIONS (BEGINNING WITH $20
               ;FOR TEMP1 AND $40 FOR TEMP2) IN BYTE PAIRS WITH THE VALUE
               ;THE A/D CONVERTER SENDS TO THE DIGITAL BOARD.
               ;
               ;R0 MUST HOLD THE BEGINNING MEMORY LOCATION.
               ;$71 HOLDS 1 PLUS THE LAST CHANNEL NUMBER.
               ;$72 HOLDS THE FIRST CHANNEL NUMBER
               ;
044E           LOAD    EQU $
044E E572              MOV A,$72      ; ACC HOLDS THE CHANNEL NUMBER
               ;
0450 5175              CALL SAMPLE
               ;SAMPLE RETURNS THE 12 BIT NUMBER FROM THE A/D CORRESPONDING
               ;TO THE CHANNEL WHOSE NUMBER WAS IN THE ACC.
               ;THE 12 BITS ARE RETURNED IN (R1,R2)
               ;IF A PROBE IS BROKEN OR NOT THERE, RETURNS (R1,R2)=$9999.
               ;
               ;STORE THIS NUMBER IN BYTE PAIRS BEGINNING AT $20, $21 FOR
               ;TEMP1 AND $40, $41 FOR TEMP2.
               ;
0452 E9                MOV A,R1
0453 F6                MOV @R0,A
0454 08                INC R0
0455 EA                MOV A,R2
0456 F6                MOV @R0,A
0457 08                INC R0
0458 0572              INC $72        ;NEXT CHANNEL
045A E572              MOV A,$72
               ;
               ;COMPARE CHANNEL NUMBER (IN $72) WITH ONE PLUS THE NUMBER OF
               ;THE LAST PROBE TO BE CALIBRATED (IN $71)
               ;
045C B571EF            CJNE A,$71,LOAD
045F 22                RET
               ;
               ;
               ;
               ;THE VOLT ROUTINE FINDS THE THERMOCOUPLE VOLTAGE BASED ON
               ;THE THERMOMETER TEMPERATURE TYPED IN AND STORED IN (R4,R5).
               ;THE EQUATION USED IS BASED ON THE TEMPERATURE-VOLTAGE
               ;STANDARD TABLES FOR T-TYPE THERMOCOUPLE PROBES
               ;THE EQUATION IS V = 426.5 * T**2 + 38,538,487 * T + 157,115,981
               ;WHERE V, VOLTAGE IS IN 10**-14 VOLTS AND T, TEMPERATURE
               ;IS (100 * DEGREE C)
               ;THE VOLTAGE CALCULATED IS DIVIDED BY (256)**3 = 16,777,216
               ;AND USED IN THIS FORM WITHOUT LOSS OF ACCURACY
```

```
                ;
0460            VOLT     EQU $
0460 8C7A                MOV $7A,R4
0462 8D7B                MOV $7B,R5
0464 1204BF              CALL SQU
                ;SQU RETURNS THE SQUARE OF (R4,R5) IN (R4,R5,R6,R7)
                ;THIS IS T**2
                ;MULTIPLY 426 * T**2
0467 7A10                MOV R2,#$10
0469 7BA9                MOV R3,#$A9    ;#$10A9 = 4265
046B 1204E7              CALL MUL24
                ;MUL24 MULTIPLIES (R2,R3) * (R4,R5,R6,R7)
                ;RESULT RETURNED IN ($75,$76,$77,$78,$79)
                ;DIVIDE BY TEN SINCE WANTED TO MULTIPLY BY 426.5
                ;INSTEAD OF 4265
046E 120595              CALL DIV51
                ;
                ;SINCE WILL NEED TO USE THE MUL24 ROUTINE AGAIN, STORE
                ;THE RESULT IN ($65,$66,$67,$68,$69)
                ;
0471 857565              MOV $65,$75
0474 857666              MOV $66,$76
0477 857767              MOV $67,$77
047A 857868              MOV $68,$78
047D 857969              MOV $69,$79
                ;NOW MULTIPLY 38,538,487 * T
                ;T WAS STORED IN ($7A,$7B)
                ;
0480 AA7A                MOV R2,$7A
0482 AB7B                MOV R3,$7B
0484 7C02                MOV R4,#$02
0486 7D4C                MOV R5,#$4C
0488 7E0C                MOV R6,#$0C
048A 7FF7                MOV R7,#$F7
048C 1204E7              CALL MUL24
                ;ADD ($75,$76,$77,$78,$79) + ($65,$66,$67,$68,$69)
                ;STORE VALUE IN (R3,R4,R5,R6,R7)
048F C3                  CLR C
0490 E579                MOV A,$79
0492 2569                ADD A,$69
0494 FF                  MOV R7,A
0495 E578                MOV A,$78
0497 3568                ADDC A,$68     ;WITH CARRY
0499 FE                  MOV R6,A
049A E577                MOV A,$77
049C 3567                ADDC A,$67     ;WITH CARRY
049E FD                  MOV R5,A
049F E576                MOV A,$76
04A1 3566                ADDC A,$66     ;WITH CARRY
04A3 FC                  MOV R4,A
04A4 E575                MOV A,$75
```

```
04A6 3565             ADDC A,$65     ;THERE SHOULD BE NO
04A8 FB               MOV R3,A       ;CARRY GENERATED HERE
              ;
              ;ADD 157,115,981 = 095D664DH
04A9 C3               CLR C
04AA EF               MOV A,R7
04AB 244D             ADD A,#$4D
04AD FF               MOV R7,A
04AE EE               MOV A,R6
04AF 3466             ADDC A,#$66    ;WITH CARRY
04B1 FE               MOV R6,A
04B2 ED               MOV A,R5
04B3 345D             ADDC A,#$5D    ;WITH CARRY
04B5 FD               MOV R5,A
04B6 EC               MOV A,R4
04B7 3409             ADDC A,#$09    ;WITH CARRY
04B9 FC               MOV R4,A
04BA EB               MOV A,R3
04BB 3400             ADDC A,#ZERO   ;TAKES CARE OF PREVIOUS
04BD FB               MOV R3,A       ;CARRY, IF ANY
              ;TRUNCATE THE LOWER THREE BYTES WHICH IS THE SAME AS
              ;DIVIDING BY (256)**3.
              ;SO THE RESULT IS IN (R3,R4).
              ;THIS IS DONE TO SIMPLIFY THE CODE WITHOUT LOSS OF ACCURACY
              ;OF THE FINAL RESULT.
              ;
04BE 22               RET
              ;
              ;
              ;
              ;THE SQU ROUTINE FINDS THE SQUARE OF THE TWO BYTE NUMBER
              ;STORED IN (R4,R5) AND RETURNS THE VALUE IN (R4,R5,R6,R7)
              ;
04BF          SQU     EQU $
              ;FIRST MULTIPLY THE LSB TIMES ITSELF, THAT IS, (R5 * R5)
              ;
04BF ED               MOV A,R5
04C0 8DF0             MOV B,R5
04C2 A4               MUL AB
04C3 FF               MOV R7,A
04C4 AEF0             MOV R6,B
              ;
              ;MULTIPLY THE LSB BY THE MSB (R5 * R4) AND DOUBLE THE RESULT
              ;
04C6 EC               MOV A,R4
04C7 8DF0             MOV B,R5
04C9 A4               MUL AB
04CA ADF0             MOV R5,B
04CC FB               MOV R3,A       ;TEMPORARY STORAGE
04CD C3               CLR C
04CE 2B               ADD A,R3
```

```
 04CF FB                MOV R3,A
 04D0 ED                MOV A,R5
 04D1 3D                ADDC A,R5      ;WITH CARRY
 04D2 FD                MOV R5,A
               ;ADD SHIFTED RESULT TO (R6,R7) AND GET (R5,R6,R7)
 04D3 C3                CLR C
 04D4 EE                MOV A,R6
 04D5 2B                ADD A,R3
 04D6 FE                MOV R6,A
 04D7 5001              JNC SQP1
 04D9 0D                INC R5
               ;
               ;MULTIPLY THE MSB BY THE MSB (R4 * R4)
               ;
 04DA EC       SQP1:    MOV A,R4
 04DB 8CF0              MOV B,R4
 04DD A4                MUL AB
 04DE ACF0              MOV R4,B
 04E0 C3                CLR C
 04E1 2D                ADD A,R5
 04E2 FD                MOV R5,A
 04E3 5001              JNC SQP2
 04E5 0C                INC R4         ;RESULT IS IN (R4,R5,R6,R7)
 04E6 22       SQP2:    RET
               ;
               ;
               ;
               ;THE MUL24 ROUTINE MULTIPLIES A TWO BYTE NUMBER STORED IN
               ;(R2,R3) BY A FOUR BYTE NUMBER STORED IN (R4,R5,R6,R7).
               ;THE RESULT IS RETURNED IN ($75,$76,$77,$78,$79).
               ;
 04E7          MUL24    EQU $
 04E7 757600            MOV $76,#ZERO
 04EA 757500            MOV $75,#ZERO
 04ED EF                MOV A,R7
 04EE 8BF0              MOV B,R3
 04F0 A4                MUL AB
 04F1 85F078            MOV $78,B
 04F4 F579              MOV $79,A
               ;
 04F6 EF                MOV A,R7
 04F7 8AF0              MOV B,R2
 04F9 A4                MUL AB
 04FA C3                CLR C
 04FB 2578              ADD A,$78
 04FD F578              MOV $78,A
 04FF 85F077            MOV $77,B
 0502 500A              JNC MS1
 0504 E577              MOV A,$77
 0506 3400              ADDC A,#ZERO   ;TAKES CARE OF CARRY IF IT EXISTS
 0508 F577              MOV $77,A
```

SOURCE FILE NAME: CAL2.ASM

```
 050A 5002              JNC MS1
 050C 0576              INC $76
                ;
 050E EE        MS1:    MOV A,R6
 050F 8BF0              MOV B,R3
 0511 A4                MUL AB
 0512 C3                CLR C
 0513 2578              ADD A,$78
 0515 F578              MOV $78,A
 0517 500A              JNC MS2
 0519 E577              MOV A,$77
 051B 3400              ADDC A,#ZERO   ;TAKES CARE OF CARRY IF IT EXISTS
 051D F577              MOV $77,A
 051F 5002              JNC MS2
 0521 0576              INC $76
                ;
 0523 E5F0      MS2:    MOV A,B
 0525 C3                CLR C
 0526 2577              ADD A,$77
 0528 F577              MOV $77,A
 052A 5002              JNC MS3
 052C 0576              INC $76
                ;
 052E EE        MS3:    MOV A,R6
 052F 8AF0              MOV B,R2
 0531 A4                MUL AB
 0532 C3                CLR C
 0533 2577              ADD A,$77
 0535 F577              MOV $77,A
 0537 5002              JNC MS4
 0539 0576              INC $76
                ;
 053B E5F0      MS4:    MOV A,B
 053D C3                CLR C
 053E 2576              ADD A,$76
 0540 F576              MOV $76,A
 0542 5002              JNC MS5
 0544 0575              INC $75
                ;
 0546 EB        MS5:    MOV A,R3
 0547 8DF0              MOV B,R5
 0549 A4                MUL AB
 054A C3                CLR C
 054B 2577              ADD A,$77
 054D F577              MOV $77,A
 054F 500A              JNC MS6
 0551 E576              MOV A,$76      ;INSTEAD OF INC $76
 0553 3400              ADDC A,#ZERO
 0555 F576              MOV $76,A
 0557 5002              JNC MS6
 0559 0575              INC $75
```

```
                ;
055B E5F0       MS6:    MOV A,B
055D C3                 CLR C
055E 2576               ADD A,$76
0560 F576               MOV $76,A
0562 5002               JNC MS7
0564 0575               INC $75
                ;
0566 EA         MS7:    MOV A,R2
0567 8DF0               MOV B,R5
0569 A4                 MUL AB
056A C3                 CLR C
056B 2576               ADD A,$76
056D F576               MOV $76,A
056F 5002               JNC MS8
0571 0575               INC $75
                ;
0573 E5F0       MS8:    MOV A,B
0575 2575               ADD A,$75
0577 F575               MOV $75,A      ;WILL NOT HAVE CARRY HERE
                ;
0579 EB                 MOV A,R3
057A 8CF0               MOV B,R4
057C A4                 MUL AB
057D C3                 CLR C
057E 2576               ADD A,$76
0580 F576               MOV $76,A
0582 5002               JNC MS9
0584 0575               INC $75
                ;
0586 E5F0       MS9:    MOV A,B
0588 2575               ADD A,$75
058A F575               MOV $75,A
                ;
058C EA                 MOV A,R2
058D 8CF0               MOV B,R4
058F A4                 MUL AB         ;B SHOULD EQUAL ZERO
0590 2575               ADD A,$75
0592 F575               MOV $75,A
                ;
0594 22                 RET            ;RESULT IN ($75,$76,$77,$78,$79)
                ;
                ;
00595           THE DIV51 ROUTINE DIVIDES ($75,$76,$77,$78,$79) BY (R2).
                ;THE RESULT IS STORED IN ($75,$76,$77,$78,$79).
                ;
0595            DIV51   EQU $
0595 7B00               MOV R3,#ZERO
0597 AC75               MOV R4,$75
0599 7D00               MOV R5,#ZERO
059B 1205C8             CALL SUBIT
```

SOURCE FILE NAME: CAL2.ASM

```
 059E 8D75              MOV $75,R5
               ;
 05A0 EC                MOV A,R4      ;SHIFT LEFT
 05A1 FB                MOV R3,A      ;R3 SHOULD HAVE BEEN ZERO
 05A2 AC76              MOV R4,$76
 05A4 7D00              MOV R5,#ZERO
005A6                   VALL SUBIT
 05A6 8D76              MOV $76,R5
               ;
 05A8 EC                MOV A,R4
 05A9 FB                MOV R3,A
 05AA AC77              MOV R4,$77
 05AC 7D00              MOV R5,#ZERO
 05AE 1205C8            CALL SUBIT
 05B1 8D77              MOV $77,R5
               ;
 05B3 EC                MOV A,R4
 05B4 FB                MOV R3,A
 05B5 AC78              MOV R4,$78
005B7                   MNOV R5,#ZERO
 05B7 1205C8            CALL SUBIT
 05BA 8D78              MOV $78,R5
               ;
 05BC EC                MOV A,R4
 05BD FB                MOV R3,A
 05BE AC79              MOV R4,$79
 05C0 7D00              MOV R5,#ZERO
 05C2 1205C8            CALL SUBIT
 05C5 8D79              MOV $79,R5
 05C7 22                RET
               ;
               ;
               ;THE SUBIT ROUTINE PERFORMS THE SUBTRTACTION (R3,R4) - R2
               ;ONLY WHEN NO CARRY (FROM R3) EXISTS.
               ;
 05C8          SUBIT    EQU $
 05C8 C3                CLR C
 05C9 EC                MOV A,R4
U05CA 9500              SUBB A,R2K
 05CC EB                MOV A,R3
 05CD 9400              SUBB A,#ZERO   ;TAKE CARE OF CARRY FROM R4
 05CF 5001              JNC SOSUB
 05D1 22                RET
               ;
 05D2 0D       SOSUB    INC R5
 05D3 C3                CLR C
 05D4 EC                MOV A,R4
 05D5 9A                SUBB A,R2
 05D6 FC                MOV R4,A
 05D7 EB                MOV A,R3
 05D8 9400              SUBB A,#ZERO
```

```
05DA FB                 MOV R3,A
05DB 80EB               JMP SUBIT       ;END OF SUBIT
               ;
               ;
               ;
               ;THE GAIN ROUTINE FINDS THE GAIN, M, BASED ON THE POINTS
               ;(X1,Y1) AND (X2,Y2)
               ;M = (Y2-Y1)/(X2-X1)
               ;(X2-X1) IS STORED IN (R4,R5) AND Y1 AND Y2 ARE STORED
               ;BEGINNING AT MEMORY LOCATION $20 AND $40, RESPECTIVELY,
               ;IN BYTE PAIRS.
               ;THE DIFFERENCE, (Y2-Y1) IS MULTIPLIED BY (256)**2 TO MAKE
               ;THE DIVISION EASIER.
               ;
05DD           GAIN     EQU $
               ;FIRST CHECK TO SEE IF THE PROBE WAS DISCONNECTED.
               ;THAT IS, IF Y1 OR Y2 = #$9999
               ;USUALLY IF ONE DOES THEY BOTH DO
05DD E6                 MOV A,@R0
05DE 18                 DEC R0
05DF B49908             CJNE A,#$99,CH1
05E2 E6                 MOV A,@R0
05E3 B49904             CJNE A,#$99,CH1
               ;HAVE Y1 = #$9999
05E6 12061C             CALL FLAG
05E9 22                 RET
               ;CHECK IF Y2 = #$9999
05EA E7        CH1:     MOV A,@R1
05EB 19                 DEC R1
05EC B49908             CJNE A,#$99,CH2
05EF E7                 MOV A,@R1
05F0 B49904             CJNE A,#$99,CH2
               ;HAVE Y2 = #$9999
05F3 12061C             CALL FLAG
05F6 22                 RET
               ;
               ;NEITHER Y1 OR Y2 EQUAL #$9999 SO PROCEED WITH THE REST OF
               ;THE GAIN ROUTINE
05F7 08        CH2:     INC R0
05F8 09                 INC R1          ;RESTORE R0 AND R1
               ;
05F9 C3                 CLR C
05FA E7                 MOV A,@R1
05FB 96                 SUBB A,@R0      ;A=(LSB OF Y2)-(LSB OF Y1)
05FC FF                 MOV R7,A        ;STORE LSB OF THE DIFFERENCE IN R7
               ;
05FD 18                 DEC R0
05FE 19                 DEC R1
05FF 8873               MOV $73,R0      ;SAVE R0
0601 E7                 MOV A,@R1
0602 96                 SUBB A,@R0      ;A = (MSB OF Y2)-(MSB OF Y1)-BORROW
```

```
 0603 FE                MOV R6,A       ;SAVE MSB OF DIFFERENCE IN R6
              ;
              ;DIVIDE (R6,R7,00,00) BY (R4,R5)
              ;(R6,R7,00,00) IS (R6,R7) * (256**2)
              ;
 0604 757500            MOV $75,#ZERO
 0607 8E76              MOV $76,R6
 0609 8F77              MOV $77,R7
 060B 757800            MOV $78,#ZERO
 060E 757900            MOV $79,#ZERO
              ;
 0611 7B00              MOV R3,#ZERO
 0613 757B00            MOV $7B,#ZERO
              ;
 0616 120626            CALL DIV43
              ;THE GAIN IS RETURNED IN ($7B,$7C,$7D)
              ;THIS IS THE GAIN * (256)**2
 0619 A873              MOV R0,$73     ;RESTORE R0
 061B 22                RET
              ;
              ;
              ;THE FLAG ROUTINE FLAGS THE GAIN BY ASSIGNING ($7B,$7C,$7D)
              ;TO BE EQUAL TO (#$999999)
              ;
 061C         FLAG      EQU $
 061C 757B99            MOV $7B,#$99
 061F 757C99            MOV $7C,#$99
 0622 757D99            MOV $7D,#$99
 0625 22                RET
              ;
              ;
              ;THE DIV43 ROUTINE DIVIDES ($76,$77,$78,$79) BY (R3,R4,R5)
              ;THE METHOD USED IS LONG DIVISION
              ;THE RESULT IS STORED IN ($7B,$7C,$7D)
              ;
 0626         DIV43     EQU $
              ;CHECK IF R3 = 0
 0626 EB                MOV A,R3
 0627 6012              JZ DD1
 0629 7800              MOV R0,#ZERO
 062B AF79              MOV R7,$79
 062D 120663            CALL SUBOK
 0630 887C              MOV $7C,R0
              ;
 0632 7800              MOV R0,#ZERO
 0634 7F00              MOV R7,#ZERO   ;VALUE OF R7 HERE DOES NOT MATTER
 0636 120663            CALL SUBOK
 0639 887D              MOV $7D,R0
              ;
 063B 857778  DD1:      MOV $78,$77
 063E 857677            MOV $77,$76
```

```
0641 857576          MOV $76,$75
0644 757500          MOV $75,#ZERO
0647 7800            MOV R0,#ZERO
0649 7F00            MOV R7,#ZERO
064B 120663          CALL SUBOK
064E 887B            MOV $7B,R0
                ;
0650 7800            MOV R0,#ZERO
0652 7F00            MOV R7,#ZERO
0654 120663          CALL SUBOK
0657 887C            MOV $7C,R0
                ;
0659 7800            MOV R0,#ZERO
065B 7F00            MOV R7,#ZERO   ;VALUE OF R7 HERE DOES NOT MATTER
065D 120663          CALL SUBOK
0660 887D            MOV $7D,R0
                ;
0662 22              RET
                ;
                ;
                ;
                ;THE SUBOK ROUTINE CHECKS IF THE SUBTRACTION OF (R3,R4,R5)
                ;FROM ($75,$76,$77,$78) WILL CAUSE A NEGATIVE RESULT.
                ;IF IT WILL, LFSH IS CALLED TO SHIFT ($76,$77,$78) ONE BYTE
                ;LEFT.  OTHERWISE, THE SUBTRACTION IS PERFORMED UNTIL IT
                ;CAUSES A NEGATIVE RESULT.
                ;
                ;THE NUMBER OF SUBTRACTIONS IS RETURNED IN R0
                ;
0663            SUBOK   EQU $
0663 C3              CLR C
0664 E578            MOV A,$78
0666 9D              SUBB A,R5
0667 E577            MOV A,$77
0669 9C              SUBB A,R4      ;WITH BORROW
066A E576            MOV A,$76
066C 9B              SUBB A,R3      ;WITH BORROW
066D E575            MOV A,$75
066F 9400            SUBB A,#ZERO   ;WITH BORROW
0671 5004            JNC QR1        ;MAY SUBTRACT
                ;IF THERE IS A CARRY HERE, NEED TO SHIFT ONE BYTE LEFT
0673 120690          CALL LFSH
0676 22              RET
                ;
0677 08         QR1:  INC R0
0678 C3              CLR C          ;THIS TIME STORE THE RESULT
0679 E578            MOV A,$78
067B 9D              SUBB A,R5
067C F578            MOV $78,A
067E E577            MOV A,$77
0680 9C              SUBB A,R4      ;WITH BORROW
```

```
0681 F577            MOV $77,A
0683 E576            MOV A,$76
0685 9B              SUBB A,R3      ;WITH BORROW
0686 F576            MOV $76,A
0688 E575            MOV A,$75
068A 9400            SUBB A,#ZERO   ;WITH BORROW
068C F575            MOV $75,A
068E 80D3            JMP SUBOK      ;END
               ;
               ;
               ;
               ;THE LFSH ROUTINE SHIFTS THE BYTES IN ($76,$77,$78,$79)
               ;LEFT ONE BYTE.  THE VALUE IN R7 IS SHIFTED IN ON THE RIGHT
               ;AND THE LEFT BYTE IS SHIFTED INTO $75.
               ;THE RESULT IS IN ($75,$76,$77,$78).
               ;
0690           LFSH    EQU $
0690 857675            MOV $75,$76
0693 857776            MOV $76,$77
0696 857877            MOV $77,$78
0699 8F78              MOV $78,R7
069B 22                RET
               ;
               ;
               ;
               ;THE INTC ROUTINE FINDS THE OFFSET B = Y1 - M * X1.
               ;FIRST MULTIPLY M * X1
               ;X1 IS STORED IN ($7E,$7F), (VOLTAGE)
               ;(M * 256**2) IS STORED IN ($7B,$7C,$7D)
               ;
069C           INTC    EQU $
               ;FIRST CHECK IF THE GAIN IS FLAGGED #$999999
069C E57B              MOV A,$7B
069E B4991A            CJNE A,#$99,W1
06A1 E57C              MOV A,$7C
06A3 B49915            CJNE A,#$99,W1
06A6 E57D              MOV A,$7D
06A8 B49910            CJNE A,#$99,W1
               ;HAVE GAIN FLAGGED SO FLAG THE OFFSET WITH #$99 99 99 99 99
06AB 757599            MOV $75,#$99
06AE 757699            MOV $76,#$99
06B1 757799            MOV $77,#$99
06B4 757899            MOV $78,#$99
06B7 757999            MOV $79,#$99
06BA 22                RET
               ;
               ;GAIN NOT FLAGGED SO PROCEED WITH NORMAL INTC ROUTINE
06BB AA7E      W1:     MOV R2,$7E
06BD AB7F              MOV R3,$7F
06BF 7C00              MOV R4,#ZERO
06C1 AD7B              MOV R5,$7B
```

```
06C3 AE7C               MOV R6,$7C
06C5 AF7D               MOV R7,$7D
06C7 91E7               CALL MUL24
             ;RESULT RETURNED IN ($75,$75,$77,$78,$79)
             ;SUBTRACT Y1 - (M * X1)
06C9 08                 INC R0
06CA C3                 CLR C
06CB E4                 CLR A
06CC 9579               SUBB A,$79
06CE F579               MOV $79,A
06D0 E4                 CLR A
06D1 9578               SUBB A,$78     ;WITH BORROW
06D3 F578               MOV $78,A
06D5 E6                 MOV A,@R0
06D6 9577               SUBB A,$77     ;WITH BORROW
06D8 F577               MOV $77,A
06DA 18                 DEC R0
06DB E6                 MOV A,@R0
06DC 9576               SUBB A,$76     ;WITH BORROW
06DE F576               MOV $76,A
06E0 E4                 CLR A
06E1 9575               SUBB A,$75     ;WITH BORROW
06E3 F575               MOV $75,A      ;IGNORE THE CARRY
             ;
06E5 22                 RET
             ;
             ;
             ;
             ;THE SAVE ROUTINE LOADS THE GAIN AND THE OFFSET INTO
             ;EXTERNAL RAM.
             ;THE GAIN REQUIRES 3 BYTES AND THE OFFSET REQUIRES 5 BYTES
             ;SO EVERY EIGHTH BYTE BEGINS A NEW GAIN AND OFFSET FOR A
             ;DIFFERENT CHANNEL.
             ;
06E6         SAVE       EQU $
06E6 E572               MOV A,$72
06E8 75F008             MOV B,#$08
06EB A4                 MUL AB         ;B IS ZERO HERE
06EC 758300             MOV DPH,#ZERO
06EF F582               MOV DPL,A
06F1 E57B               MOV A,$7B
06F3 F0                 MOVX @DPTR,A
06F4 A3                 INC DPTR
06F5 E57C               MOV A,$7C
06F7 F0                 MOVX @DPTR,A
06F8 A3                 INC DPTR
06F9 E57D               MOV A,$7D
06FB F0                 MOVX @DPTR,A   ;HAVE GAINED STORED
06FC A3                 INC DPTR
06FD E575               MOV A,$75
06FF F0                 MOVX @DPTR,A
```

SOURCE FILE NAME: CAL2.ASM

```
0700 A3                 INC DPTR
0701 E576               MOV A,$76
0703 F0                 MOVX @DPTR,A
0704 A3                 INC DPTR
0705 E577               MOV A,$77
0707 F0                 MOVX @DPTR,A
0708 A3                 INC DPTR
0709 E578               MOV A,$78
070B F0                 MOVX @DPTR,A
070C A3                 INC DPTR
070D E579               MOV A,$79
070F F0                 MOVX @DPTR,A    ;HAVE OFFSET STORED
0710 22                 RET
                ;
                ;
                ;
                ;THE CORR ROUTINE FINDS THE CORRECTED TEMPERATURE USING THE
                ;STORED GAIN AND OFFSET FOR THE PARTICULAR CHANNEL.
                ;IT IS CALLED WITH (R1,R2) = Y, THE 12 BIT NUMBER READ FROM
                ;THE A/D AND R7 HOLDING THE CHANNEL NUMBER.
                ;
                ;FINDS X IN Y = M * X + B
                ;THAT IS X = (Y - B) / M
                ;
0711            CORR    EQU $
0711 120750             CALL REC
                ;
                ;REC RETURNS THE GAIN IN ($7B,$7C,$7D) AND THE OFFSET IN
                ;($75,$76,$77,$78,$79) FOR THE CHANNEL WHOSE NUMBER IS IN R7
                ;
                ;CHECK IF THE GAIN IS FLAGGED WHICH IMPLIES THAT THE OFFSET
                ;IS ALSO, SO NO NEED TO CHECK BOTH.
0714 E57B               MOV A,$7B
0716 B49911             CJNE A,#$99,Q1
0719 E57C               MOV A,$7C
071B B4990C             CJNE A,#$99,Q1
071E E57D               MOV A,$7D
0720 B49907             CJNE A,#$99,Q1
                ;THE GAIN IS FLAGGED SO FLAG X IN ($7C,$7D)
0723 757C99             MOV $7C,#$99
0726 757D99             MOV $7D,#$99
0729 22                 RET
                ;
                ;SUBTRACT (Y-B), THAT IS ACTUALLY ((Y*(256**2) - B)
                ;
072A C3         Q1:     CLR C
072B 7400               MOV A,#ZERO
072D 9579               SUBB A,$79
072F F579               MOV $79,A
0731 7400               MOV A,#ZERO
0733 9578               SUBB A,$78      ;WITH BORROW
```

```
0735 F578              MOV $78,A
0737 EA                MOV A,R2
0738 9577              SUBB A,$77     ;WITH BORROW
073A F577              MOV $77,A
073C E9                MOV A,R1
073D 9576              SUBB A,$76     ;WITH BORROW
073F F576              MOV $76,A
0741 7400              MOV A,#ZERO
0743 9575              SUBB A,$75     ;WITH BORROW
0745 F575              MOV $75,A      ;$75 HERE SHOULD BE 0
                 ;
                 ;($75,$76,$77,$78,$79) = (Y - B)
                 ;DIVIDE (Y - B) / M = X
                 ;
0747 AB7B              MOV R3,$7B
0749 AC7C              MOV R4,$7C
074B AD7D              MOV R5,$7D
                 ;
074D D126              CALL DIV43     ;RETURNS X IN ($7B,$7C,$7D)
                 ;
074F 22                RET
                 ;
                 ;
                 ;
                 ;THE REC ROUTINE LOADS THE GAIN AND OFFSET FROM RAM, FOR THE
                 ;CHANNEL NUMBER STORED IN R7, INTO MEMORY LOCATIONS
                 ;($7B,$7C,$7D) AND ($75,$76,$77,$78,$79) RESPECTIVELY.
                 ;
0750             REC   EQU $
                 ;FIND THE BEGINNING ADDRESS FOR THE PARTICULAR CHANNEL
0750 EF                MOV A,R7
0751 75F008            MOV B,#$08
0754 A4                MUL AB         ;B IS ZERO
0755 758300            MOV DPH,#ZERO
0758 F582              MOV DPL,A
                 ;
075A E0                MOVX A,@DPTR
075B F57B              MOV $7B,A
075D A3                INC DPTR
075E E0                MOVX A,@DPTR
075F F57C              MOV $7C,A
0761 A3                INC DPTR
0762 E0                MOVX A,@DPTR
0763 F57D              MOV $7D,A      ;HAVE GAIN IN ($7B,$7C,$7D)
                 ;
0765 A3                INC DPTR
0766 E0                MOVX A,@DPTR
0767 F575              MOV $75,A
0769 A3                INC DPTR
076A E0                MOVX A,@DPTR
076B F576              MOV $76,A
```

```
 076D A3                      INC DPTR
 076E E0                      MOVX A,@DPTR
 076F F577                    MOV $77,A
 0771 A3                      INC DPTR
 0772 E0                      MOVX A,@DPTR
 0773 F578                    MOV $78,A
 0775 A3                      INC DPTR
 0776 E0                      MOVX A,@DPTR
 0777 F579                    MOV $79,A      ;HAVE OFFSET IN ($75,$76,$77,$78,$79)
                      ;
 0779 22                      RET
                      ;
                      ;
                      ;
                      ;THE TEMP ROUTINE FINDS THE TEMPERATURE BASED ON THE VOLTAGE
                      ;STORED IN ($7C,$7D).  THE EQUATION USED IS
                      ;T = -27.27 * V**2 + 7,237,085 * V + 116,878,675
                      ;WHEN THE RESULTING T IS DIVIDED BY (256**3), HAVE THE
                      ;TEMPERATURE * 100 WHICH IS STORED IN (R0,R1).
                      ;
 077A                 TEMP    EQU $
 077A AC7C                    MOV R4,$7C
 077C AD7D                    MOV R5,$7D
 077E 91BF                    CALL SQU
                      ;SQU RETURNS THE SQUARE OF (R4,R5) IN (R4,R5,R6,R7)
                      ;MULTIPLY 2727 * (V**2)
 0780 7A0A                    MOV R2,#$0A
 0782 7BA7                    MOV R3,#$A7
 0784 91E7                    CALL MUL24
                      ;MUL24 MULTIPLIES (R2,R3) * (R4,R5,R6,R7)
                      ;RESULT RETURNED IN ($75,$76,$77,$78,$79)
                      ;NEXT DIVIDE BY 100 SINCE REALLY WANTED 27.27 INSTEAD OF
                      ;2727 MULTIPLIED BY V**2.
 0786 7A64                    MOV R2,#$64
 0788 B195                    CALL DIV51
                      ;SINCE NEED TO USE MUL24 ROUTINE AGAIN, STORE THE RESULT IN
                      ;($65,$66,$67,$68,$69).
                      ;
 078A 857565                  MOV $65,$75
 078D 857666                  MOV $66,$76
 0790 857767                  MOV $67,$77
 0793 857868                  MOV $68,$78
 0796 857969                  MOV $69,$79
                      ;NOW MULTIPLY 7,237,085 * V
                      ;V WAS STORED IN ($7C,$7D)
                      ;
 0799 AA7C                    MOV R2,$7C
 079B AB7D                    MOV R3,$7D
 079D 7C00                    MOV R4,#ZERO
 079F 7D6E                    MOV R5,#$6E
 07A1 7E6D                    MOV R6,#$6D
```

```
07A3 7FDD            MOV R7,#$DD
07A5 91E7            CALL MUL24
              ;SUBTRACT ($65,$66,$67,$68,$69) FROM THE RESULT IN
              ;($75,$76,$77,$78,$79)
              ;STORE THE DIFFERENCE IN (R0,R1,R2,R3,R4)
07A7 C3              CLR C
07A8 E579            MOV A,$79
07AA 9569            SUBB A,$69
07AC FC              MOV R4,A
07AD E578            MOV A,$78
07AF 9568            SUBB A,$68     ;WITH BORROW
07B1 FB              MOV R3,A
07B2 E577            MOV A,$77
07B4 9567            SUBB A,$67     ;WITH BORROW
07B6 FA              MOV R2,A
07B7 E576            MOV A,$76
07B9 9566            SUBB A,$66     ;WITH BORROW
07BB F9              MOV R1,A
07BC E575            MOV A,$75
07BE 9565            SUBB A,$65     ;WITH BORROW
07C0 F8              MOV R0,A       ;THERE SHOULD BE NO BORROW GENERATED
              ;
              ;ADD 116,878,675
07C1 C3              CLR C
07C2 EC              MOV A,R4
07C3 2453            ADD A,#$53
07C5 FC              MOV R4,A
07C6 EB              MOV A,R3
07C7 346D            ADDC A,#$6D    ;WITH CARRY
07C9 FB              MOV R3,A
07CA EA              MOV A,R2
07CB 34F7            ADDC A,#$F7    ;WITH CARRY
07CD FA              MOV R2,A
07CE E9              MOV A,R1
07CF 3406            ADDC A,#$06    ;WITH CARRY
07D1 F9              MOV R1,A
07D2 E8              MOV A,R0
07D3 3400            ADDC A,#ZERO   ;WITH CARRY
07D5 F8              MOV R0,A
              ;
              ;TRUNCATE THE LOWER THREE BYTES.
              ;SAME AS DIVIDING BY (256)**3
              ;SO RESULT IS IN (R0,R1).
              ;
07D6 22              RET
              ;
              ;
              ;THE CONV ROUTINE WILL CONVERT THE NUMBER IN (R0,R1) TO THE
              ;FOUR BYTE ASCII EQUIVALENT, WHICH WILL BE STORED IN
              ;(R4,R5,R6,R7).
              ;
```

```
07D7            CONV     EQU $
                ;DIVIDE (R0,R1) BY 10.  THE REMAINDER IS THE LEAST
                ;SIGNIFICANT DIGIT OF THE ASCII EQUIVALENT
07D7 7A00                MOV R2,#ZERO
07D9 7B00                MOV R3,#ZERO
07DB 1207FF              CALL DIV21
                ;RETURNS THE QUOTIENT IN (R2,R3) AND THE REMAINDER IN R1
07DE E9                  MOV A,R1
07DF 4430                ORL A,#$30     ;TO CONVERT TO ASCII
07E1 FF                  MOV R7,A
                ;
                ;REPEAT THE DIVISION TWO MORE TIMES TO GET THE NEXT 2 BYTES
                ;SETUP FOR DIV21
07E2 EA                  MOV A,R2
07E3 F8                  MOV R0,A
07E4 EB                  MOV A,R3
07E5 F9                  MOV R1,A
07E6 7A00                MOV R2,#ZERO
07E8 7B00                MOV R3,#ZERO
07EA 1207FF              CALL DIV21     ;QUOTIENT IN (R2,R3), BUT R2=00
                ;
07ED E9                  MOV A,R1
07EE 4430                ORL A,#$30
07F0 FE                  MOV R6,A
                ;
07F1 EB                  MOV A,R3
07F2 75F00A              MOV B,#TEN
07F5 84                  DIV AB
                ;THE REMAINDER B IS THE THIRD BYTE
                ;A HOLDS THE FOURTH BYTE
07F6 4430                ORL A,#$30
07F8 FC                  MOV R4,A
                ;
07F9 E5F0                MOV A,B
07FB 4430                ORL A,#$30
07FD FD                  MOV R5,A
                ;
07FE 22                  RET
                ;
                ;
                ;THE DIV21 ROUTINE DIVIDES THE TWO BYTE NUMBER IN (R0,R1).
                ;BY 10 USING LONG DIVISION
                ;STORE THE RESULT IN (R2,R3) AND THE REMAINDER IN R1.
                ;
07FF            DIV21    EQU $
                ;CHECK IF THE SUBTRACTION CAUSED A BORROW
07FF C3                  CLR C
0800 E8                  MOV A,R0
0801 940A                SUBB A,#TEN
0803 5003                JNC RR1
0805 020810              JMP RR2
```

```
                ;
0808 0A         RR1:    INC R2
0809 C3                 CLR C
080A E8                 MOV A,R0
080B 940A               SUBB A,#TEN
080D F8                 MOV R0,A        ;STORE THIS TIME
080E 80EF               JMP DIV21
                ;
0810 C3         RR2:    CLR C
0811 E9                 MOV A,R1
0812 940A               SUBB A,#TEN
0814 E8                 MOV A,R0
0815 9400               SUBB A,#ZERO    ;WITH BORROW
0817 5001               JNC RR3
0819 22                 RET
                ;QUOTIENT RETURNED IN (R2,R3) AND REMAINDER IN R1
                ;
081A 0B         RR3:    INC R3
081B C3                 CLR C
081C E9                 MOV A,R1
081D 940A               SUBB  A,#TEN
081F F9                 MOV R1,A
0820 E8                 MOV A,R0
0821 9400               SUBB A,#ZERO    ;WITH BORROW
0823 F8                 MOV R0,A
0824 80EA               JMP RR2         ;END OF DIV21
                ;
                ;
                ;
                ;THE OUT ROUTINE WILL OUTPUT THE TEMPERATURE STORED IN
                ;(R4,R5,R6,R7) IN ASCII BYTES TO THE TERMINAL.
                ;
0826            OUT     EQU $
0826 EC                 MOV A,R4
0827 1202F8             CALL OUTCH
082A ED                 MOV A,R5
082B 1202F8             CALL OUTCH
082E 742E               MOV A,#POINT
0830 1202F8             CALL OUTCH
0833 EE                 MOV A,R6
0834 1202F8             CALL OUTCH
0837 EF                 MOV A,R7
0838 1202F8             CALL OUTCH
083B 22                 RET
                ;
0000                    END
```

AVOCET SYSTEMS 8051 CROSS-ASSEMBLER - VERSION 1.09

SOURCE FILE NAME: CAL2.ASM
---- SYMBOL TABLE ----

ACC	00E0	DSPLY	8000	MS2	0523	PR1	019D	RR3	081A
AGAIN	032F	ECHO	0317	MS3	052E	PR10	0209	SAM	012E
ALTCH	0094	ERFR	0267	MS4	053B	PR11	0215	SAMPLE	0275
B	00F0	ERR	0292	MS5	0546	PR12	0221	SAVE	06E6
B1200	02CC	ET100	0443	MS6	055B	PR13	022D	SBUF	0099
B192	02F0	ET2	0402	MS7	0566	PR14	0239	SCON	0098
B2400	02D5	ET3	0412	MS8	0573	PR2	01A9	SOSUB	05D2
B4800	02DE	ETT	044B	MS9	0586	PR3	01B5	SQP1	04DA
B600	02C3	FINISH	0334	MT	03A8	PR4	01C1	SQP2	04E6
B9600	02E7	FLAG	061C	MUL24	04E7	PR5	01CD	SQU	04BF
BAUD	02A8	FN1	0157	NEXT	0327	PR6	01D9	START	0095
BELL	0007	FORT	0040	NN	0177	PR7	01E5	SUBIT	05C8
BLANK	0020	FORT1	0041	NN1	017A	PR8	01F1	SUBOK	0663
CALIB	0336	FR	0254	NOERR	0297	PR9	01FD	TCON	0088
CH1	05EA	FRE	025D	NT1	03B5	PRE	023B	TEMP	077A
CH2	05F7	GAIN	05DD	NT2	03BB	PRMT	003E	TEN	000A
CHNL	023C	GOTIT	02F3	NT3	03C6	PROBE	0185	TEST3	0389
CHX	0142	INCH	0300	NT4	03CC	PROMT	031C	TH1	008D
CONV	07D7	INIT	0000	NT5	03E1	PT	03E9	THE	0595
CORR	0711	INTC	069C	NT6	03E7	PTER	03F0	TI	0099
CR	000D	LF	000A	NT7	0429	PTS	03F5	TMOD	0089
CRLF	030A	LFSH	0690	NT8	0439	PTST	03FC	TR1	008E
DATAH	E000	LOAD	044E	OUT	0826	Q1	072A	TWEN	0020
DATAL	F000	LPP	036E	OUTCH	02F8	QR1	0677	TWEN1	0021
DD1	063B	MAIN	011D	P1	0090	RDY	0096	VOLT	0460
DIV21	07FF	MESS	0323	P3	00B0	REC	0750	W1	06BB
DIV43	0626	MOR0	0160	PCON	0087	REDO	03AC	ZERO	0000
DIV51	0595	MOR1	016A	PDATAH	C000	RI	0098		
DPH	0083	MORE	0129	PDATAL	D000	RR1	0808		
DPL	0082	MS1	050E	POINT	002E	RR2	0810		

REFERENCES

[1] A. Gharakhani, "The Design of Multiple Channel Thermometry System for Use in Hyperthermia," M.S. Thesis, University of Illinois, Urbana, unpublished, 1984.

[2] "Manual on the Use of Thermocouples in Temperature Measurement," American Society for Testing and Materials, pp. 20-23, 1981.

[3] Analog Devices, "Monolithic Thermocouple Amplifier with Cold Junction Compensation," 1982.

[4] R. L. Powell, W. J. Hall, C. H. Hyink, Jr., and L. L. Sparks, "Thermocouple Reference Tables Based on the IPTS-68," National Bureau of Standards MN-125, Department of Commerce, Washington, D.C., 1974.

[5] A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975), pp. 413-418.

[6] Intel Corporation, "MCS-51 Family of Single Chip Microcomputers User's Manual," Chapter 4, 1981.

[7] "ASTM Standards on Precision and Accuracy for Various Applications," American Society for Testing and Materials, 1977.