THE DESIGN OF MULTIPLE CHANNEL THERMOMETRY SYSTEM FOR USE IN HYPERTHERMIA

BY

ALFRED GHARAKHANI

B.S., University of Illinois, 1982

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

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author would like to express gratitude to his advisors Professor Charles A. Cain and Professor Richard L. Magin for their guidance and encouragement. Special thanks are due to Mary Ozarka for development of the calibration software, and Steven Silverman and John McCarthy for their efforts in preparation of the software development system. Thanks are extended to Mr. Joseph Cobb and Mr. Billy McNeill for their assistance in building the hardware. The efforts of Mrs. Wanda Elliott in the preparation of this thesis are also greatly appreciated.

The author is grateful to Professor Edward S. Davidson for his assistance and cooperation during the past five years.

CHAPTER I

INTRODUCTION

In recent years there has been an increasing amount of research about treatment of cancerous tumors by hyperthermia. In hyperthermia the toxic effects of heat are used to destroy the cancerous cells. In order to have an effective treatment the temperature of the tumor must be monitored in several points because excessive heat can harm the healthy tissues as well as cancerous cells. To measure the temperature in several different locations a multichannel thermometry system is required.

The prototype of a 16 channel thermometry system with thermocouple probes was developed. This system can be used in conjunction with a complete hyperthermia system whose central computer will use the digital information from this thermometry unit to calculate an appropriate scan path for a phased array heating applicator.

A block diagram of this thermometry system is shown in Figure 1.1. Solid lines represent the analog and digital data paths, dashed lines indicate control signals. The developed prototype has the following specifications:

- 16 channels of thermocouple inputs.
- Type J (iron-constantan) thermocouples, reconfigurable to operate with other types of thermocouples.
- Extended biological range, $0^{o}C$ - $51.2^{o}C$ with an accuracy of $\pm 0.1^{o}C$.
- Graphics display capabilities.
- Selectable temperature display rate (300 baud - 4800

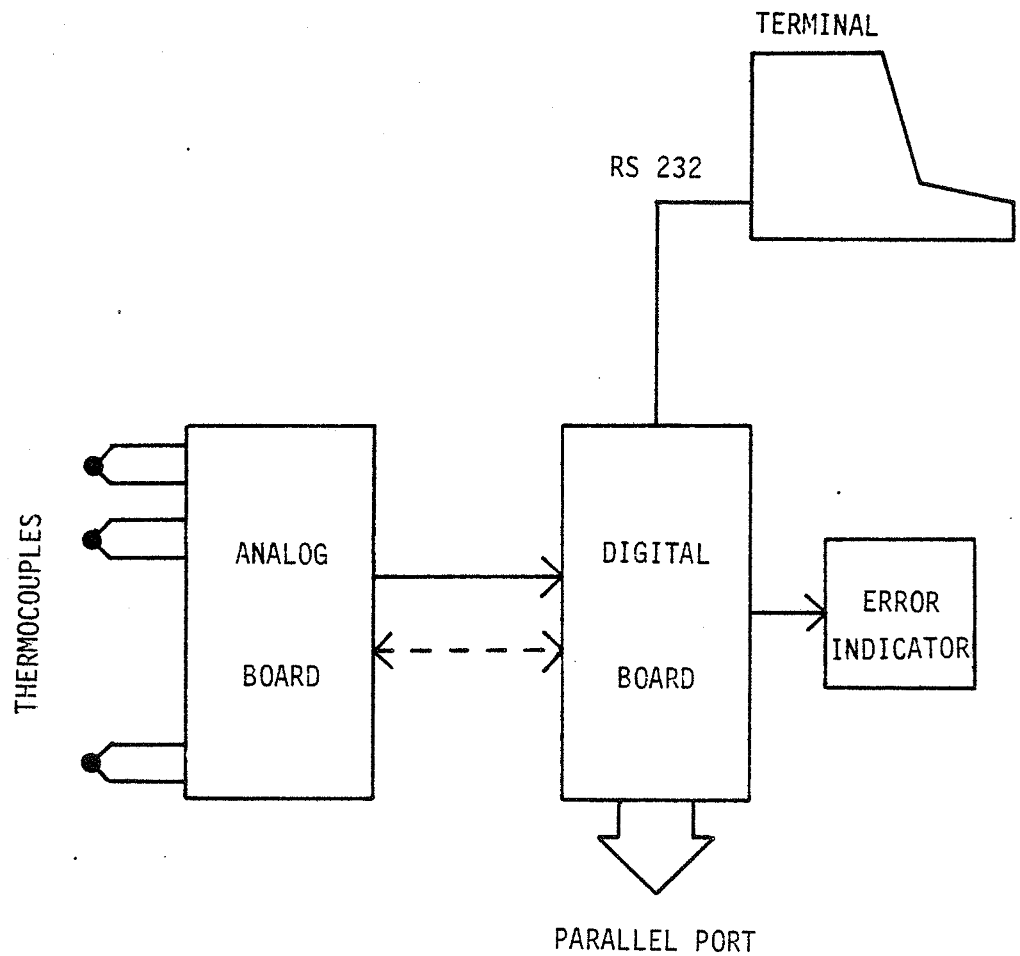

Figure 1.1. The thermometry system block diagram.

baud).

- Parallel port for communication with a central computer or printer.
- Controlled by a 12 MHz Intel 8031 microcomputer.
- Double point self calibration routine.
- Capability of detecting faulty probes.

This system consists of two boards and two power supplies. They are installed in a 17" x 9.5" x 9" metal box.

The main purpose of this thesis is to describe the design criteria and operation of this system as clearly as possible. Thus it can be used as a guide for further improvement and modification.

CHAPTER II

THE HARDWARE DESCRIPTION

2.1. Overview

The hardware of this thermometry system can be divided into two distinct parts: an analog unit and a digital unit. The analog unit is responsible for the conditioning and digitizing of low level signals produced by thermocouples. However, the digital unit controls the analog unit and processess the raw digital data produced by the analog unit. The digital unit has many other responsibilities such as user interface, error detection, and calibration.

The analog and digital hardware were built on two separate boards. This reduces the undesirable effects of fast switching TTL voltages on the low level analog signals. In addition, two separate boards simplify any future changes significantly. For example, the procssor board can be easily replaced by another computer board or a host computer. In the current configuration, the processor board can be easily connected to another data acquisition system.

In the remainder of this chapter the analog and digital hardware will be explained and the problem of noise will be examined thoroughly. The noise problem receives extended discussion because the low level analog signals are very susceptible to high frequency digital signals. Therefore the performance of the system could be degraded significantly if the noise is not eliminated.

2.2. The Analog Board

The function of the analog board is to amplify and digitize the low level voltage produced by a thermocouple junction. A block diagram of this board is shown in Figure 2.1. The complete schematic of the analog board is shown in Figure 2.2.

The temperature sensing element was chosen to be a type-J, iron-constantan thermocouple. Measurements have shown that type-J thermocouples are very linear over the range of 0^oC - 1000^oC [Omega, 1983a]. First stage amplification is done by the AD594C. This is a Monolithic Thermocouple Amplifier with Cold Junction Compensation, manufactured by Analog Devices. This chip has a built-in ice point compensation as well as a thermocouple failure alarm [Analog Devices, 1982].

The AD594C is designed to operate with a type-J thermocouple. It can be recalibrated for use with other types of thermocouples by the addition of two external resistors. The output voltage of AD594C in an ideal case is given by

$$V_{594} = 192.3 \ (V_{TC} + 23 \quad V) \tag{2.1}$$

where V_{TC} is the thermoelectric voltage created by thermocouple junction. These values for type-J thermocouples are tabulated in ANSI tables [Omega, 1983b]. The V_{594} is the voltage at the output of AD594C. For example, the first stage amplifiers will produce voltage of V_{594} = 4.423 mV at 0^oC and V_{594} = 513.75 mV at 51.2^oC ($V_{TC}(0)$ = 0 mV and $V_{TC}(51.2)$ = 2.699 mV).

In order to multiplex sixteen analog signals, an AD363(AIS) is used. This package has a sample-and-hold in addition to a 16 channel analog MUX. In order to match the maximum voltage

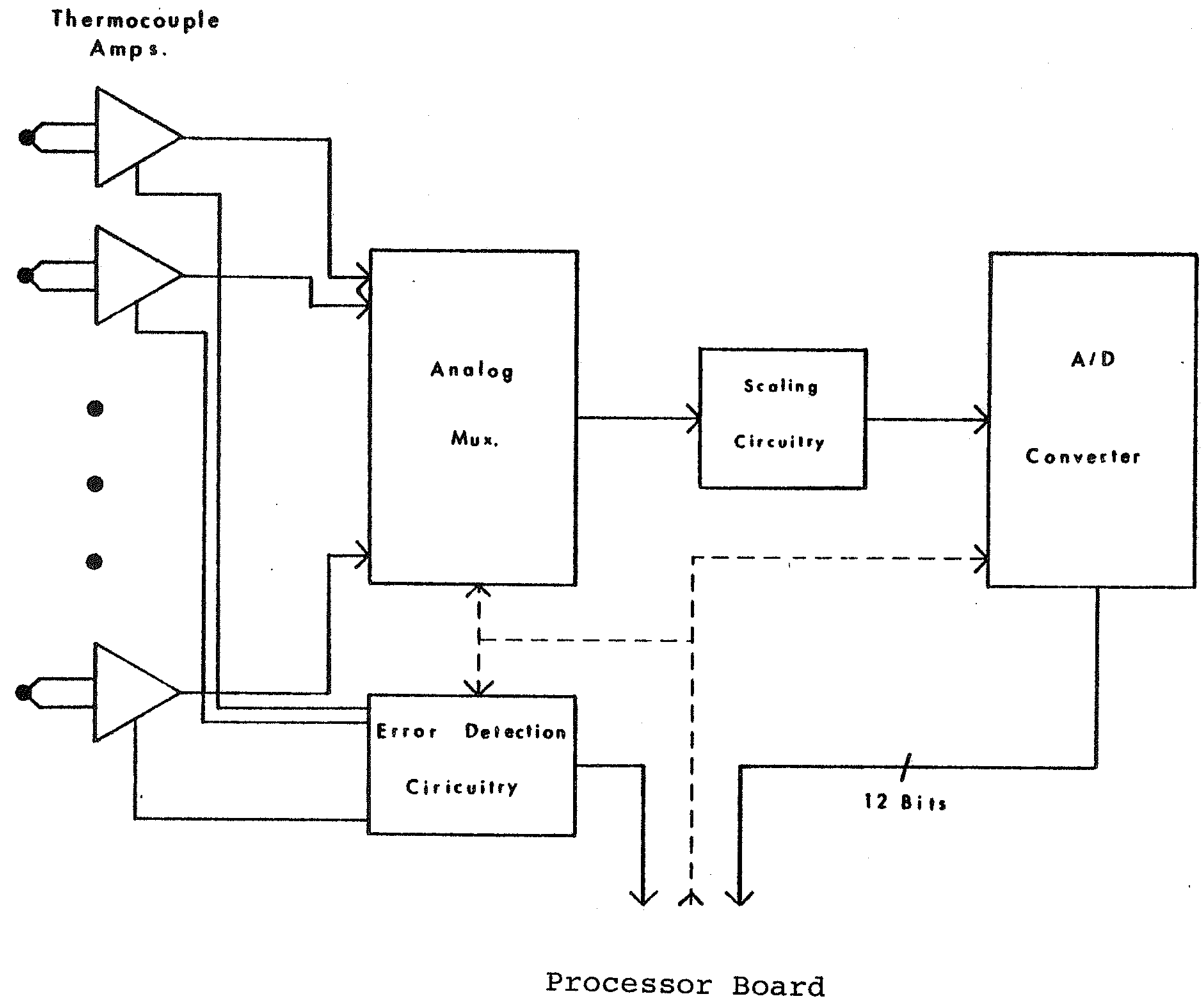

Figure 2.1. The analog board block diagram.

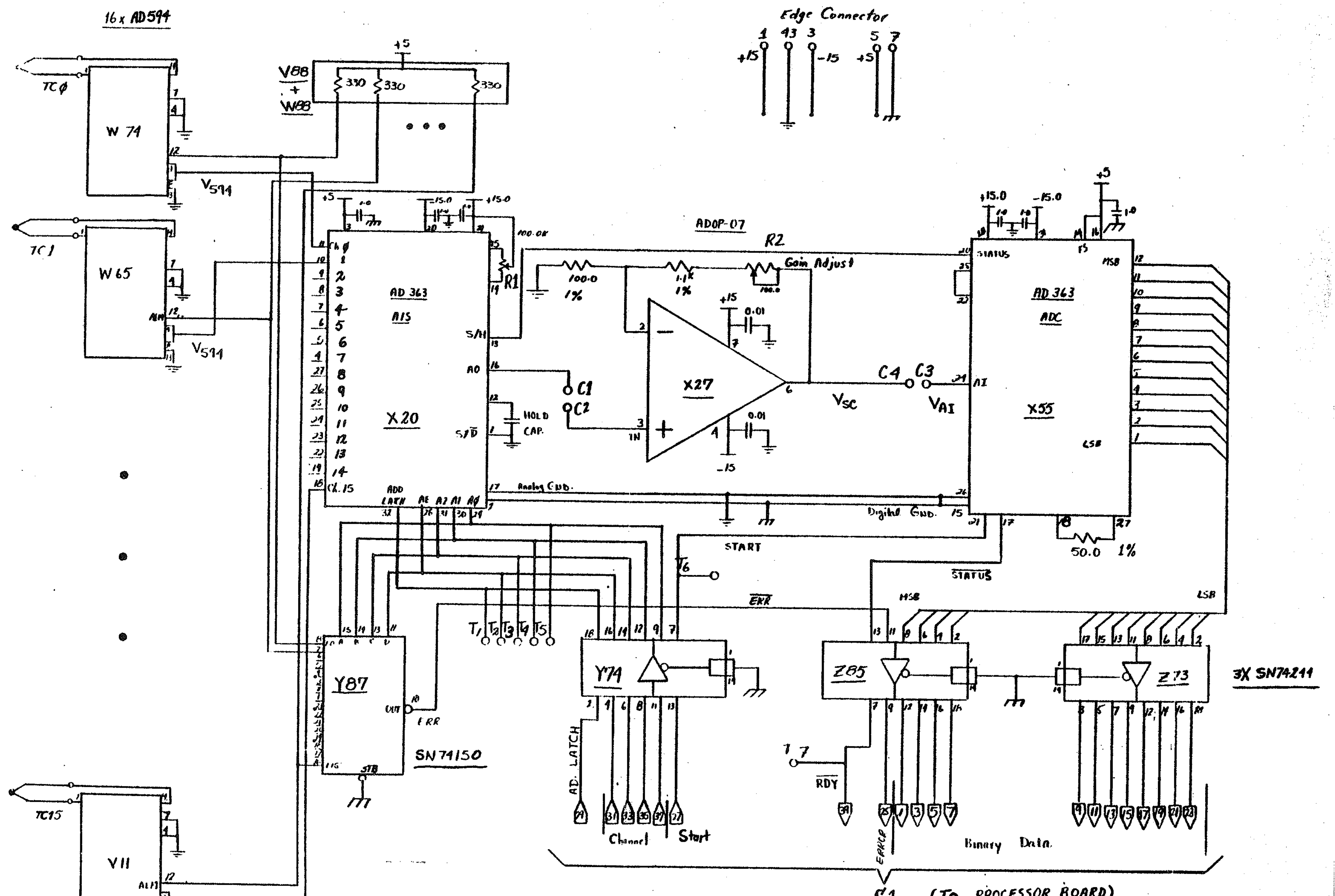

Figure 2.2. The analog board schematic.

produced by the first stage amplifier with the reference voltage of the A/D converter (5 volts), a scaling circuit is needed. The scaling circuitry consists of a non-inverting amplifier with a variable gain of 8 to 12. The gain is adjusted by a 10 turn trim pot during manual calibration (to obtain an accurate and stable gain, 1% precision resistors are used). This amplifier is constructed using an Analg Devices AD-OP-07. This instrumentation operational amplifier has ultra-low offset voltage and offset voltage drift. The output of scaling circuitry (V_{SC}) is directly connected to the analog input of the A/D converter (V_{AI}) where

$$V_{AI} = V_{SC} = G_{SC} \times 192.3 \times (V_{TC} + 23\ \mu V) \qquad (2.2)$$

To digitize the analog signal produced by the scaling circuitry, a AD363(ADC) is used. This is a 12 bit Analog-to-Digital converter which has a 25 µsec conversion time. A resolution of $0.1^{o}C$ over a range of $0^{o}C$ - $51.2^{o}C$ is achieved by only 9 bits of digital data, but a 12 bit A/D converter is used, primarily for future development and enhancement. After being buffered the digitized 12 bit data is transmitted to processor board via a flat cable.

The process of sampling a particular channel is as follows:

1. An appropriate channel address is selected by the processor.
2. The selected channel is latched into Address latch register.
3. A START command from processor initiates the conversion of A/D converter.

4. Upon receiving a START pulse, the A/D converter puts the sample-and-hold in the hold position and starts the 25 μsec conversion.
5. The A/D converter notifies the processor using the RDY signal, as soon as the conversion is over.
6. The processor reads the valid data upon receiving of the RDY signal.

This process is illustrated by a timing diagram shown in Figure 2.3.

The function of the sample-and-hold is to keep the analog signal in a constant level while the A/D converter is busy with the conversion. Since the temperature of the body changes fairly slowly, the sample-and-hold may be disabled. Calculation shows that this will introduce a maximum error of $0.07^{o}C$ in the final readout, which is quite acceptable. Disabling the sample-and-hold will eliminate a digital signal (STATUS), which like any other digital signal can create a spike on the analog signal. This particular spike may harm the system more than others, because it will be coupled to the input lines of the analog MUX right before the sample-and-hold freezes the analog signal.

The analog circuitry will not provide accurate data unless it is calibrated. This is a manual calibration and has to be done only once. It consists of adjusting the offset voltage of AD363(AIS) and the gain of the scaling circuitry (G_{SC}). To adjust the offset voltage, the thermocouples must be placed in an ice bath; and variable resistor R1 has to be turned until all data bits of A/D converter are zeroes.

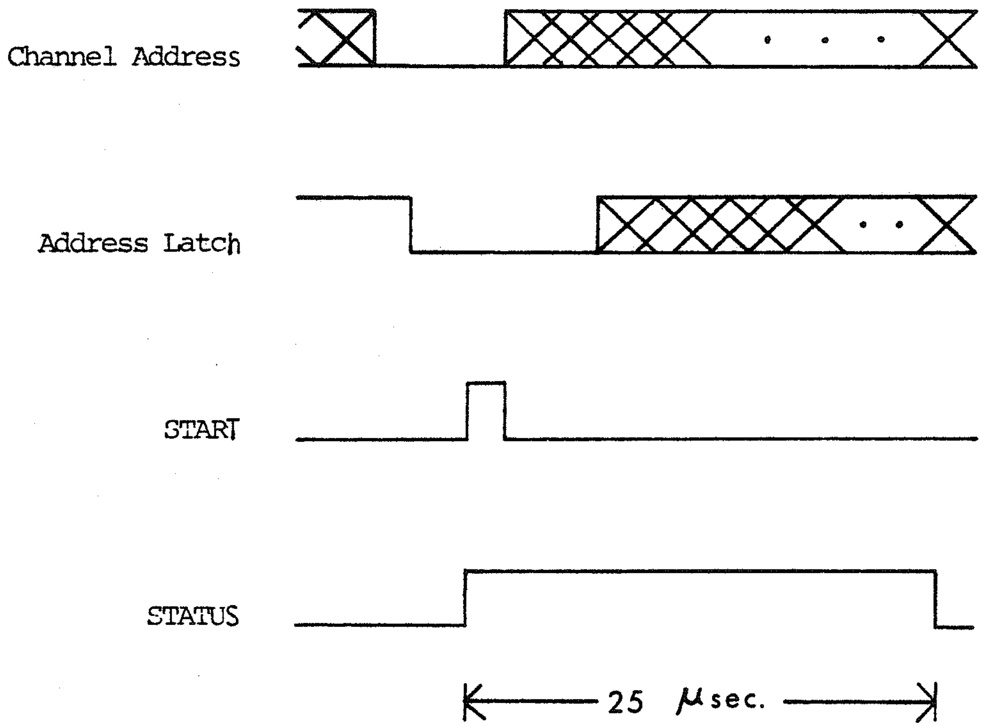

Figure 2.3. The analog board timing diagram.

To adjust the gain all probes should be placed in a 51.2°C water bath and adjust R2 until all data bits of A/D are ones.

In order to simplify trouble shooting and manual calibration, the important digital and analog signals such as START, RDY, etc., are brought out to the component side of both boards, so they can be easily probed. To simplify the manual calibration the scaling circuitry can be completely isolated from the rest of the circuit by two jumpers.

2.3. The Processor Board

The processor board has numerous duties. The most important one is to control the analog board and to interpret the raw data coming from the A/D converter. In addition, the processor board can sense the ERROR flag coming from the analog board (broken thermocouples) and notify the user by displaying an error code. The final data is transmitted to the outside world by either a serial port with variable baud rate or a parallel port.

An overall block diagram of the digital board is shown in Figure 2.4. The complete schematic of the processor board is shown in Figure 2.5.

2.3.1. The Microprocessor

The microprocessor plays the key role on the processor board. An Intel 8031 microprocessor with a 12 MHz clock is used. This single chip microcontroller has a full duplex serial port, two 16 bit timer-counters, 384 bytes of internal data memory, up to 64 K of program addressing capability in addition to 64 K of data memory addressing capabilities [MCS-51 Users Manual, 1981a]. This microcontroller eliminates the need for a number of external chips

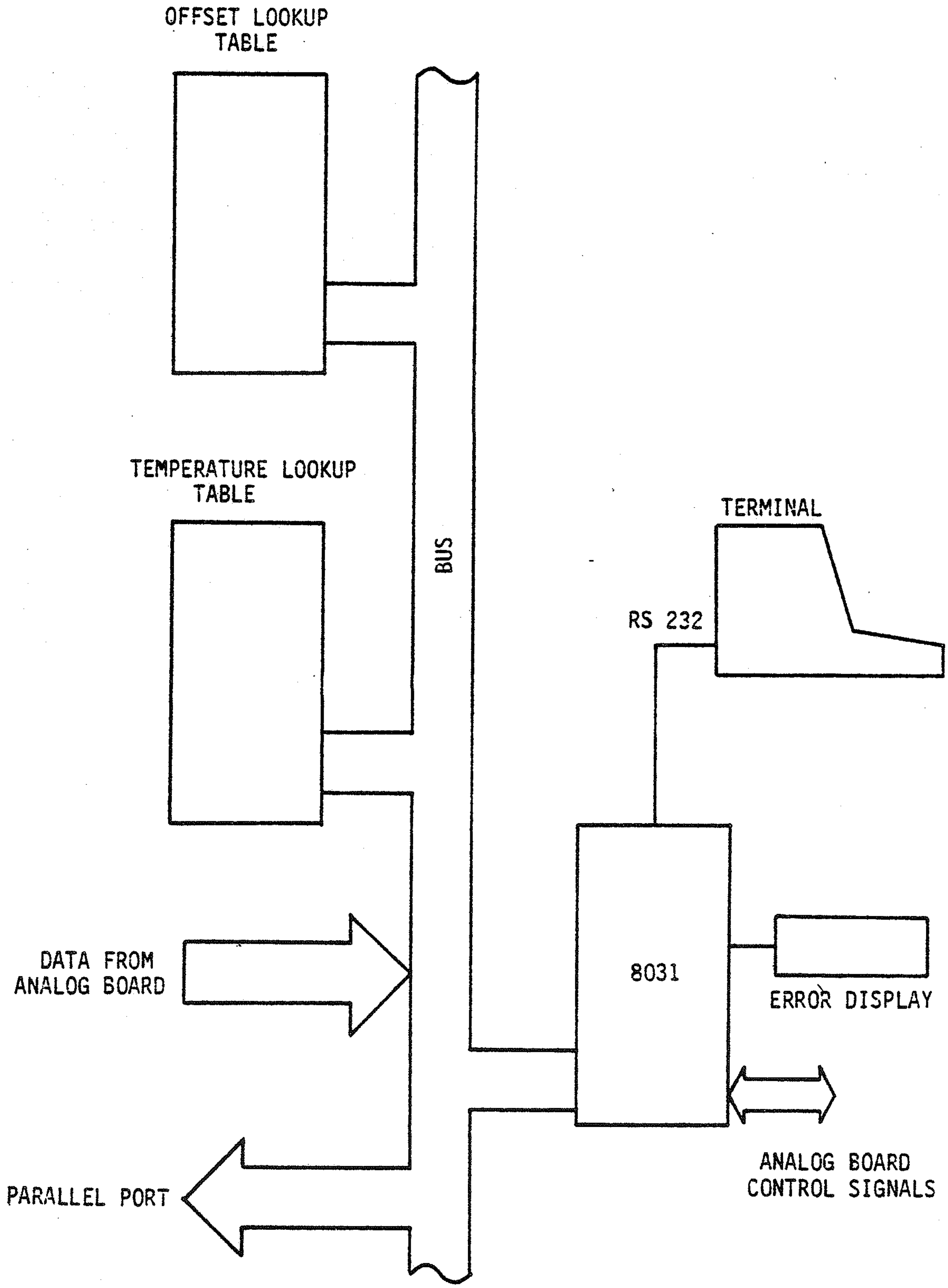

Figure 2.4. The processor board block diagram.

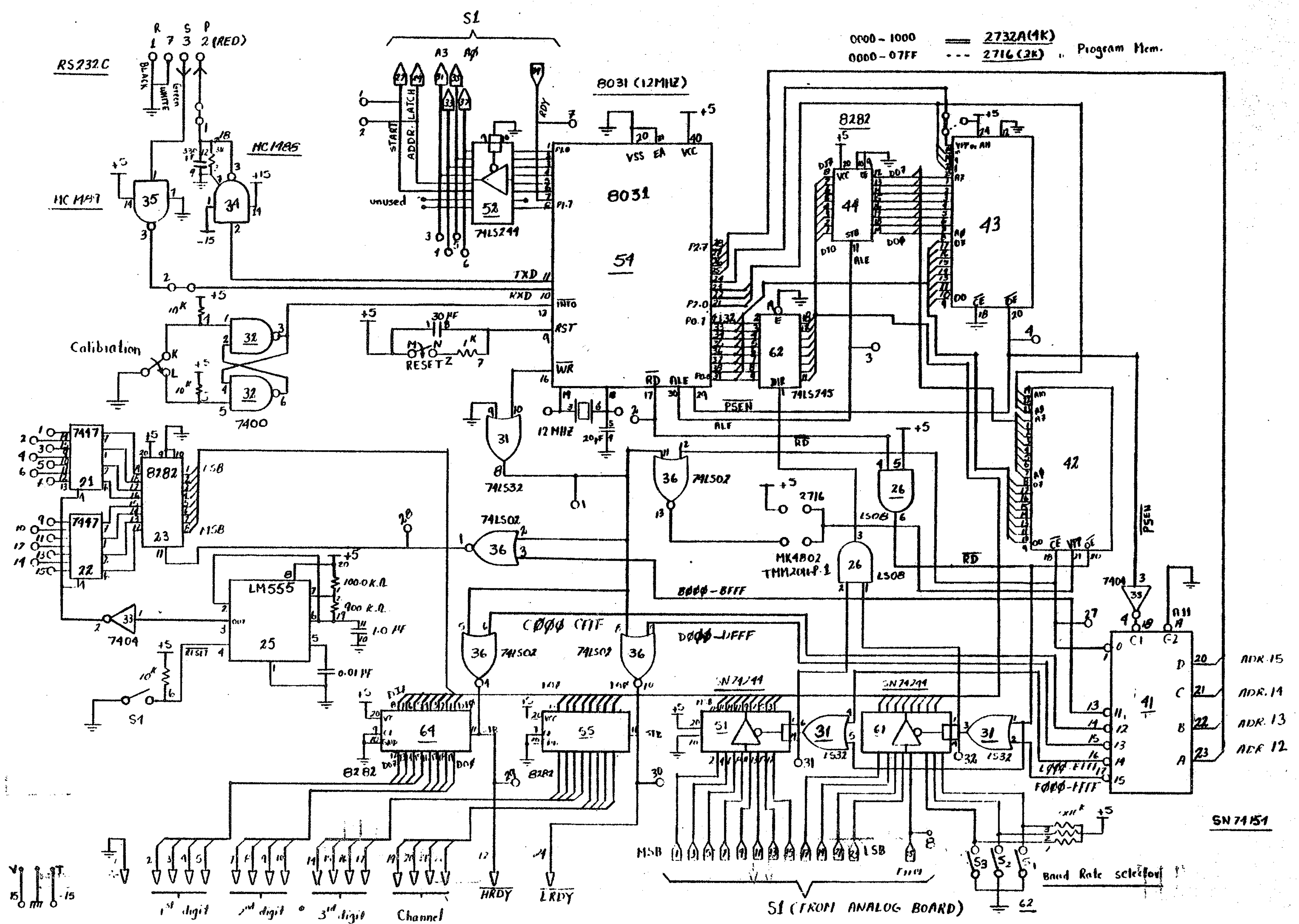

Figure 2.5. The processor board schematic.

such as external timers for baud rate generation and USART chips for serial communication and external clock generators. To go even further the program memory can be eliminated if an 8051 is used and provided that the program length would not exceed 4 K. The 8051 is just like an 8031 with 4 K of program memory on chip. Elimination of all these external chips would reduce the cost of the system significantly, because the peripheral components tend to be rather costly.

The 8031 has four 8-bit ports. Port 0 provides the multiplexed low order address and data lines while Port 2 provides the high order address lines. Port 0 and Port 2 can not be used as input ports simply because the address and data appear on them very frequently. Most of the alternate functions of Port 3 are used. The unused ones are two external clock inputs of timers and one interrupt line. Port 1 is the port that generates the signals which control and monitor the analog board. The most significant bit of this port is not used but can be used for any purpose in the future. The configuration of Port 1 is shown in Table 2.1. Notice that RDY is the only input pin in this port.

The ports in the 8031 are bidirectional. Each line in any port can be independently used as an input or an output. If a logical one is written in the latch of an I/O line, that line will be configured as an input. In order to read that input line, the pin instead of the latch must be read. In order to use a line as an output, the desired value must simply be written into the latch using "read-modify-write" instructions [Microcontroller Users Manual, 1982]. Keep in mind that system reset will configure all the ports as inputs.

Table 2.1. Port 1 Configuration.

Port pin	Signal
P1.0	A0
P1.1	A1
P1.2	A2
P1.3	AE
P1.4	ADDRESS LATCH
P1.5	START
P1.6	RDY (Input)
P1.7	UNUSED

The serial port is used for serial data communication with a terminal or a host computer. This is done through an RS 232C serial cable. Signal voltage level conversion is performed by MC1488 and MC1489 line drivers and receivers.

The serial port can operate with a series of standard baud rates. The dip switches S1, S2, and S3 select the desired baud rate; these switches are read during the initialization process. The system RESET must be pressed after the baud rate is changed. Table 2.2 contains the proper setting for each baud rate (O = open, C = closed).

Table 2.2. Baud Rate Selection Table.

S3	S2	S1	Baud Rate
C	C	C	300
C	C	O	600
C	O	C	1200
C	O	O	2400
O	C	C	4800

2.3.2. The Program Memory

In this thermometry system, program memory not only stores the system program but also contains the temperature look up table and the monitor messages as well. A standard 2732 EPROM is used for this purpose. This is a 4 K Erasable Programmable Read Only memory. For development purposes a smaller memory (2716) can also be used (jumper selectable). An 8031 with 12.0 MHz clock would require a program memory with a minimum access time of 450 nsec. It should be noted that 4096 bytes of program memory is more than sufficient for the first version software plus the look up table. The program memory can be expanded up to 64 K in future if a large program is needed.

Figure 2.6 displays the program memory map on the current system. The look up tables and the messages are intentionally stored in the very top part of the memory. This will leave a solid, one piece memory area for the actual control program. The control program begins at 100 H simply because it is a requirement of our homemade software development system. The first version of

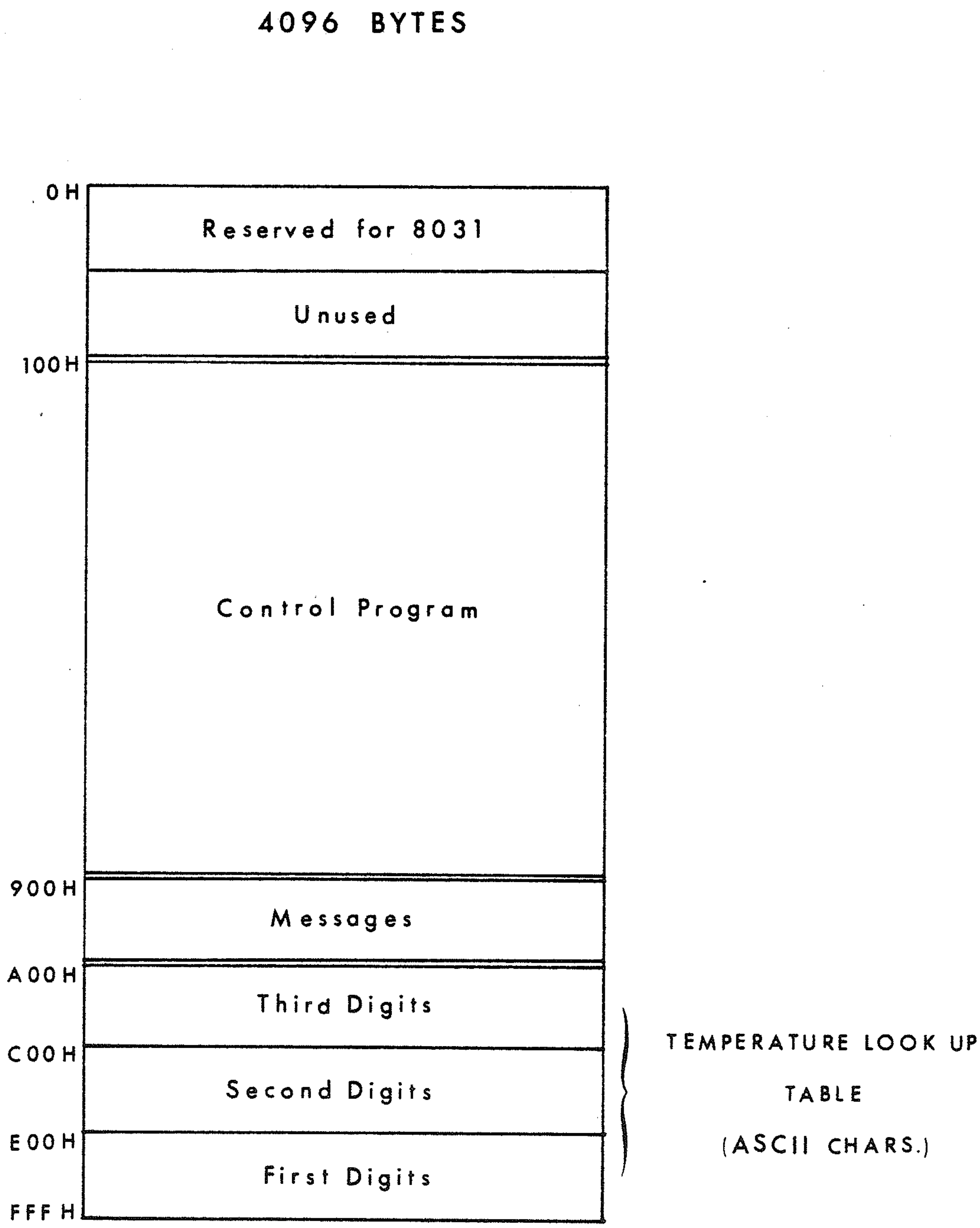

Figure 2.6. The program memory map.

the control program is thoroughly described in the listing of the control program in Appendix A.

2.3.3. The Data Memory

This system uses the data memory quite extensively. The temperature offset look up table resides in the bottom of the data memory addressing space (0 H - 7FF H). The upper parts of the data memory space are used for memory mapped input and output. More than 42 K of unused data memory is left for future expansion (perhaps for a larger offset look up table). Table 2.3 contains the address requirements of the memory mapped Input Output registers.

Table 2.3. The Address of I/O Registers.

REGISTER	ADDRESS (HEX)
DATAL	F000(_FFFF)
DATAH	E000(-EFFF)
PDATAH	D000(_DFFF)
PDATAL	C000(_CFFF)
DSPLY	B000(_BFFF)

The processor can read the data from the A/D converter, baud rate setting and ERROR flag from DATAL, and DATAH registers. PDATAH and PDATAL registers are primarily for the parallel data transmission. Finally, the DSPLY is a general purpose register, the contents of which will be displayed on two adjacent seven segment displays located on the front panel. The current system uses the displays for displaying the error codes. If a "FF H" is written into this register, all of the segments will turn off. A one hertz oscillator will cause the displays to blink, unless it is disabled by switch S4.

2.3.4. A Few Words About the Intel 8031 Microprocessor

This microprocessor is designed for control application; hence, it is called a microcontroller. It has several useful features built in that eliminates the need for a number of peripheral chips. It also has some disadvantages which are not quite obvious to the inexperienced user. These disadvantages are summarized as follows:

1. The 8031 is a pure 8 bit machine; therefore, 16 bit arithmetic cannot be done directly. This can complicate the software, especially in a 12 bit data acquisition system.
2. The instruction set lacks several very important features. For example, addition and subtraction can only be done with the accumulator. There is only one kind of conditional jump in the instruction set.
3. Data read and write are performed slowly. Prior to any data read and write, the Data Pointer Register (the only one) has to be modified. This will slow down the calibration routine program which has numerous data reads and writes.
4. Although the 8031 processor has four bidirectional ports, only two of them can be used efficiently. Ports 0 and 2 can not be used for I/O purposes because data and address information appears on them in each fetch cycle.

5. There are very few software development tools available for this processor.

In summary, it should be mentioned that the Intel 8031 is a very inexpensive and simple solution for control applications which do not require large amounts of numerical calculations.

2.4. The Noise Consideration

Noise is a phenomenon which can cause significant problems for designers of analog and digital circuitry. Computers, video monitors, power supplies, power lines, and many other electrical instruments around us are constantly generating a great deal of noise. Noise can have very unpleasant effects both on digital environment and on analog environment. A voltage transient can easily cause a software upset in a computer system. A 500 mV noise signal can easily be coupled to a low level analog signal if the circuit is not properly grounded and shielded.

Low level analog signals are very sensitive to the fas switching digital signals. High frequency TTL lines radiate noise which can be coupled to the analog signal as well as to the system ground. This can seriously endanger the integrity of the analog signals because the coupled spikes can be quite large compared to the low level analog voltages.

Noise is a type of problem that can never be eliminated completely. In this thermometry system the following considerations have been made in order to reduce the effects of noise as much as possible.

The most important remedy is to place the analog and digital circuitry physically apart. Two separate vector boards are used, one for the analog circuitry and the other for the processor board. These two boards have separate grounds. This is to prevent digital signals from flowing in the analog ground circuit and inducing spurious analog signal noise. These two grounds must be connected to each other in a single point very close to the A/D converter. This will prevent a voltage difference between two grounds in the vicinity of the A/D converter.

Although two separate boards are used, the analog board still contains some digital circuitry such as A/D converter and buffers. In order to reduce the coupling of digital noise on analog lines, coaxial cable is used for every single analog signal. The shield of the cable is grounded at both ends [Microcontroller Application Handbook, 1982a]. Coaxial cable not only provides protection against electromagnetic interference, but prevents capacitive coupling as well.

In order to filter out the possible transients on power supply buses, each package has a proper decoupling capacitor. The analog chips have two decoupling capacitors, one for high frequency and one for low frequency interference. Decoupling capacitors will be virtually useless if they are not placed properly. Therefore, they must be placed in such a way as to minimize the area of the loop formed by the capacitor and the chip [Microcontroller Applications Handbook, 1982b]. In order to protect the analog board even more, a grounded metal shield is placed behind the analog board.

Although all these considerations were helpful, the analog ground still has approximately 10 - 20 mV noise (peak to peak).

This is a low volume compared to the noise that was present before the above considerations were made.

CHAPTER III

SELF CALIBRATION

3.1. The Necessity of Calibration

In addition to the noise there are several other factors which might introduce some inaccuracy into this system. Thermocouple probes and amplifiers tend to be somewhat inaccurate. Fortunately they remain very linear. Temperature drift can cause a small amount of variation in the gain and the offset voltages of the thermocouple amplifiers. The analog multiplexer, the scaling circuitry, and the A/D converter on the other hand have extremely small temperature drift.

Measurements show that the combination of these errors can be highly significant. In other words, if a calibration scheme is not used, $0.1^{o}C$ accuracy would be an impossible claim. Fortunately, all of the components used tend to be extremely linear; this simplifies the calibration process significantly.

3.2. A Closer Look at the Analog Components

The results of experimental measurements show that AD594C (Thermocouple Amplifier) has a very linear response in a temperature range of $20^{o}C$ - $50^{o}C$. In the ideal case, this amplifier has a gain of 192.3. Let us assume that the ambient temperature change will cause a ΔG_{594} gain change and an O_{594} offset voltage. This component also has a constant calibration inaccuracy called E_{594} which is guaranteed to be less than 10 mV. In order to take the stated inaccuracies into account, Eq. (2.1) must be rewritten as follows:

$$V_{594}(mV) = [192.3 + G_{594}][V_{TC} + 0.023] + O_{594} + E_{594} \quad (3.1)$$

Keep in mind that V_{TC} is a value which is extracted from standard tables and is assumed to be precise.

The rest of the components in the analog board, are all very linear and stable. They have extremely small amounts of offset voltages and offset voltage drift levels. Any kind of inaccuracy created by these components is definitely negligible. The final analog input of A/D converter is expressed by Eq. (3.2).

$$V_{AI}\ (mV) = G_{SC}\ [192.3 + G_{594}][V_{TC} + 0.023] + O_{594} + E_{594} \quad (3.2)$$

Every single term in Eq. (3.2) is constant except V_{TC}, which varies with temperature. In other words, the inaccuracy of the components does not harm the linearity of this system.

3.3 Calibration Procedure

The basics of the calibration process are shown in Figure 3.1. The X coordinate in this figure represents the actual temperature of the probe. The Y axis represents the temperature reading of the thermometer. Therefore, the line Y = X represents a perfectly calibrated thermometry system which can precisely measure the temperature of the probe. Any other line which does not have slope of unity and zero intercept would represent an uncalibrated system.

Let us assume that an uncalibrated thermometry system can be modeled by a straight line with a slope of m and an intercept of b. This is a valid assumption in our case because it was shown in the previous section that component error will not upset the linearity of this system. The goal is to form a table for our uncalibrated system called offset table. This table contains an offset value for each temperature reading. If the offset value is algebrically added to its corresponding temperature reading, a precise reading would be the result. The vertical arrows in Figure 3.1 represent the offset values. It can be seen that each point on line 1 can be transformed to line 2 (ideal case) if a proper offset value is algebraically added to it.

To calculate the offset values, the equation of line 1 is required; this is done by taking two measurements at two temperatures, A and B. The farther away A and B become, the more accurate the calculations will be. Since the thermocouple amplifiers are most accurate at 25°C, A is selected to be 25°C. To simplify calcuations B is selected to be 35°C. Keep in mind that the selection of A and B are arbitrary, so any value between 0°C and 51.2°C can be picked. The slope and intercept of line 1 are calculated using the following equations:

$$m = (Y_B - Y_A)/10 \tag{3.3}$$

$$b = Y_A - (25).m \tag{3.4}$$

For example, if the probes are inserted in a water bath X_C degrees hot, our uncalibrated system will display Y_C instead of X_C. Since we know the characteristics of our uncalibrated system (slope and intercept are calculated using Y_A and Y_B), we can estimate X_C using Eq. (3.5).

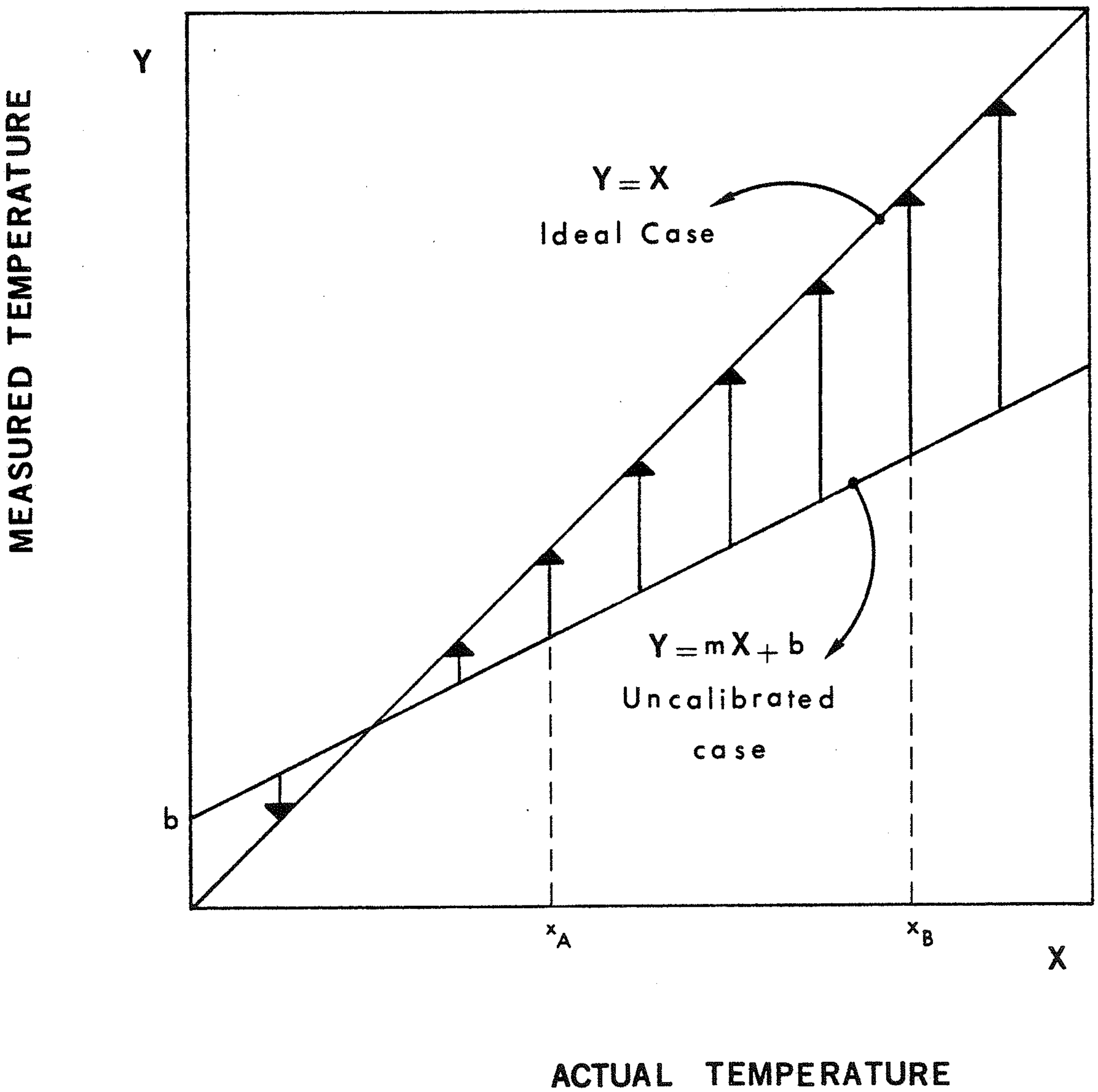

Figure 3.1. The calibration model.

$$X_C = (Y_C - b)/m \tag{3.5}$$

This equation can be used to calculate Xs for any given Y. The division in Eq. (3.5) would slow down the the processor significantly if it is performed for every sample. To avoid this, a look up table of offset values (O_Cs) for every possible temperature reading (Y_Cs) is generated during the calibration process. The offset value corresponding to Y_C is calculated using Eq. (3.6).

$$O_C = X_C - Y_C = (Y_C - b)/m - Y_C \tag{3.6}$$

The list of offset values can be stored in a Random Access Memory. The generation of this list may take a long time (few minutes), but it has to be done only once. When the list of offset values exists, then instead of using Eq. (3.5), the following equation can be used for each sample.

$$X_C = Y_C + O_C = Y_C \pm \ O_C \tag{3.7}$$

This equation has only one addition or one subtraction. The microprocessor can perform addition and subtraction much faster than division (four times faster).

Since the offset values are stored in a RAM, every time the system is turned off the contents of this memory, or offset values, will be lost. Therefore, the calibration process must be repeated every time the system is powered up. Fortunately, the development of Non Volatile Ram or Zero Power Ram can avoid this problem. MK48Z02 by MOSTEK is a 2 K x 8 Non Volatile Static RAM which has an integral lithium energy source. This static RAM unlike other RAM chips can retain its data in absence of power [MOSTEK, 1983]. Once this component is distributed, it can replace the current RAM Chip (TMM2016P-1). When this happens calibration will not be required after each power up. Instead, the program must be modified to make calibration an interrupt driven event, which means that whenever the CALIBRATE push button (on the front panel) is pressed, the processor will be interrupted and it will jump to the subroutine which does the actual calibration.

The first version of the calibration software performs this process in the following order:

1. The user is asked to place the probes in a $25^{o}C$ water bath and press return.
2. When the return is pressed the processor will sample the channels used, and will store the 9 bit readings (Y_As) for each channel in a scratch pad memory.
3. The user is asked to raise the temperature of the bath to $35^{o}C$ (or use a separate bath) and press return.
4. For a second time the processer samples the channels and stores the results.
5. Based on the stored values, the processor calculates the slope (m) and the intercept (b) for each channel using Eqs. (3.3) and (3.4).
6. Using calculated slope and intercept values, the processor calculates a sufficient number of offset values (one offset for each four temperture readings) from $0^{o}C$

to 51.2^{o}C. Each channel has its own set of offset values.

7. These values are stored in the offset look up table (2 KX8 RAM) for future reference.

Since the look up table is 2048 bytes long, each channel can only have 128 bytes of storage. Therefore, each channel can only have 128 offset values for a range of 0^{o}C - 51.2^{o}C. Considering the fact that this system has a resolution of 0.1^{o}C, four consecutive readings must share the same offset value. Calculations show that this is quite adequate for this prototype version.

The offset look up table is divided into 16 equal sections, each 128 bytes long. The offset values for channel zero reside at the very bottom of the RAM. The very top 128 bytes of RAM belongs to offset values of channel fifteen.

To find the proper offset value of a 9 bit sample, the least two significant bits of this number are ignored and the remaining 7 bits are concatinated to the 4 bit channel number. The 11 bit result is used as an address to look up the proper offset value. The byte read is a binary sign magnitude representation of the offset. In other words, the most significant bit of this offset would determine whether it is a negative or positive valve. If the most significant bit is set, then the offset value is subtracted from the 9 bit sample, otherwise it is added to it. Finally, the resulting value is used to look up the precise temperature reading from the temperature look up table. The above process is summarized in Figure 3.2.

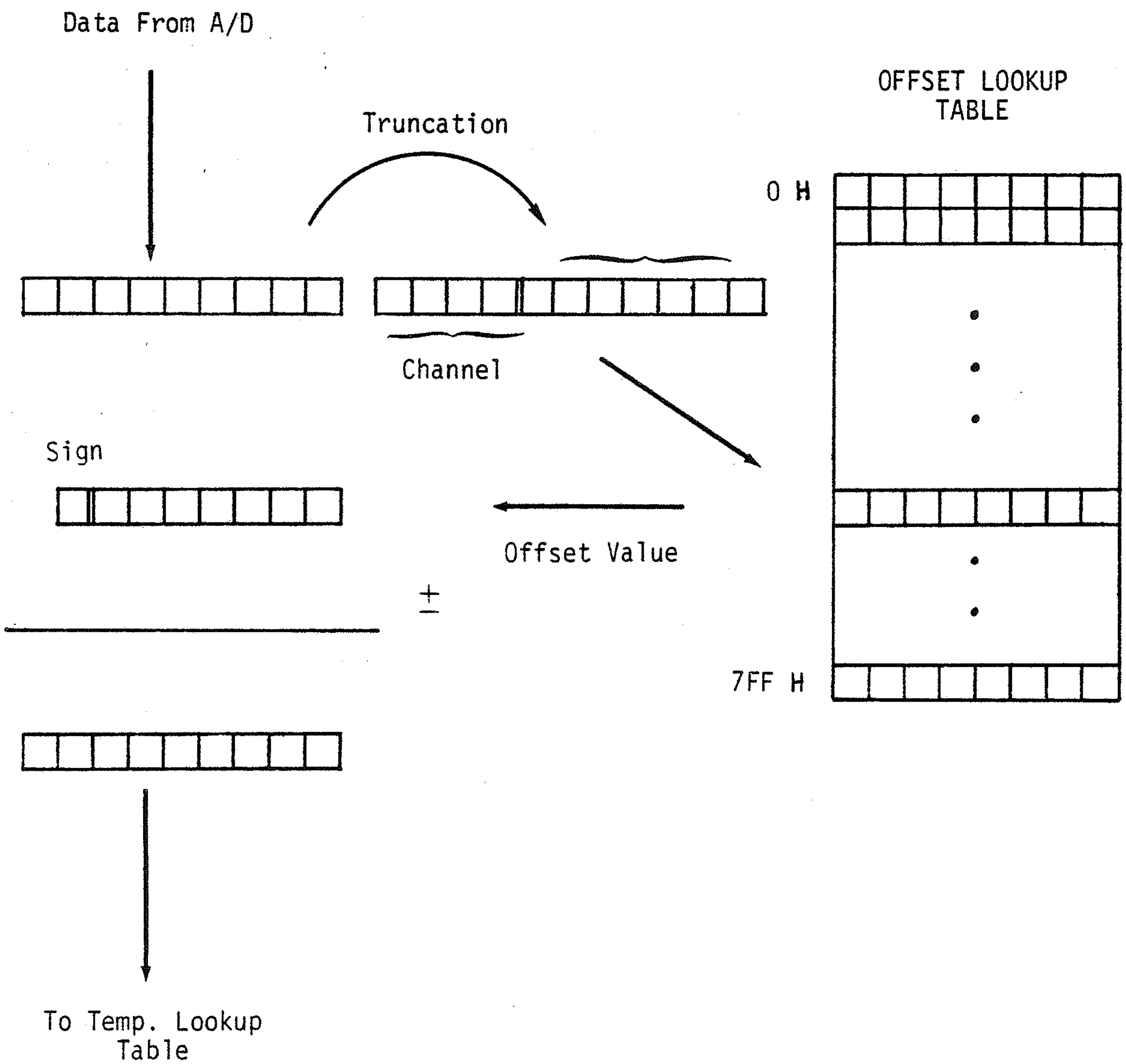

Figure 3.2. The calibration process.

It is a good idea to examine the magnitude of the offset periodically. If it becomes larger than a threshold value (specified by the user), the operator must be requested to perform a manual calibration. The first version of the calibration software does not have this feature, but a simple subroutine can easily be added to the calibration program, which would implement this feature.

CHAPTER IV

POSSIBLE IMPROVEMENT AND ENHANCEMENT

4.1. Overview

The purpose of this chapter is mainly to express several ideas which might improve the performance and reduce the production cost of this system. In order to reduce the production cost, the components should obviously be purchased in large quantities. These components are usually cheaper if the ordered quantity is more than one hundred. To improve the accuracy and quality of the temperature readings, the volume of noise must be reduced even more. This is not an easy task and would require a very careful examination of the problem.

This chapter is divided into two parts. The first portion contains some recommendations for cost reduction. The second part contains several suggestions which might improve the quality of the operation.

4.2. System Cost Reduction

The components on the analog board tend to be much more expensive than the components on the processor board. If any significant cost reduction is desired, the analog board should be the center of attention.

There are sixteen thermocouple amplifiers in the analog board. These amplifiers not only occupy a large portion of the board, but also are expensive. It might be possible to fabricate several of these amplifiers on a single die. If this system ever goes to production line, it is strongly recommended to have custom made packages instead of standard packages.

In this system, analog signal multiplexing and analog to digital conversion are performed by a complete 16 channel data acquisition system (AD 363). This product is physically placed on two separate packages; the first one has a 16 channel multiplexer plus a sample-and-hold and the second package contains a 12 bit A/D converter. The AD 363 is the most expensive part in this system. As mentioned previously, the elimination of sample-and-hold would cause an insignicant number of errors in the system. On the other hand, two least significant data bits of the A/D converter are ignored. There is no doubt in my mind that AD 363 can have a much cheaper alternative. It can be replaced by a 16 channel analog multiplexer with no sample-and-hold plus a 10 bit A/D converter.

To reduce the production cost of the processor board, an 8051 instead of 8031 can be used. This will eliminate the 4 K external program memory (2732 EPROM). It should be kept in mind that this is only possible when the control program is completely working and it is less than 4096 bytes long. If an 8051 is used, the control program will reside in a 4 K Read Only Memory inside the microprocessor package. This means that any program modification would be very costly.

4.3. Performance Improvement

In order to have accurate temperature readings, the noise on the analog signals must be reduced. There are a number of things that can improve the situation:

1. Instead of a wire-wrapped board, a printed circuit board with wide ground planes should be used. This will result

in a significant improvement. In a printed circuit the design will be more compact and the conductor lengths will be minimal.

2. Instead of sending digital signals directly to the analog board from the processor board, optical couplers can be used. This will result in a much more complete isolation of the processor board from the analog board. The analog ground and the digital ground of the processor board would not have to be connected to a single point. This will prevent the pollution of the analog ground by the digital ground.
3. Since the frequency of the noise on the analog signal is much greater than the sampling rate and the rate of the temperature change, sixteen low pass filters can be used to filter the high frequency noise on each channel. These RC filters must be placed between each thermocouple amplifier and the analog multiplexer. If the sample-and-hold is not used, a single filter, between analog multiplexer and A/D converter, can replace all these sixteen filters. In the latter case an active low pass filter with variale gain can replace the scaling circuitry and the low pass filter.
4. The noise filtering can be done digitally. To have accurate readings, several samples can be averaged by the processor. The current program calculates the average of two samples each time a reading is desired. To get better results four or sixteen samples can be averaged.

Keep in mind that an 8 bit processor can not perform double precision arithmetic efficiently; therefore, averaging multiple samples would slow down the processor significantly.

In addition to noise prevention, there are a number of other strategies which would improve the accuracy of this system. Using an accurate and sensitive thermocouple such as type-T or type-E would certainly help. In this case the output voltage of thermocouple amplifier (V594) can not be expressed by Eq. (2.1). Therefore, V594 must be interpreted differently. This would not only require a new look up table, but the manual calibration would have to be repeated as well.

CHAPTER V

SUMMARY

This thesis contains the hardware and software description of the 16 channel thermometry system. The problems which were encountered during the development phase were discussed. The calibration process also was examined in detail.

Finally, a few suggestions which might improve this thermometry unit were made.

APPENDIX

THE CONTROL PROGRAM AND THE LOOK UP TABLE

```
;*************************************************************
;*              16-CHANNEL THERMOMETRY SYSYTEM               *
;*                           BY                              *
;*                    ALFRED GHARAKHANI                      *
;*        UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN         *
;*             BIOACOUSTICS RESEARCH LABORATORY              *
;*THIS PROGRAM IS THE FIRST VERSION OF SOFTWARE WHICH WILL *
;*CONTROL THE THERMOMETRY UNIT. IT IS DIVIDED TO MANY SECT-*
;*IONS AND SUBROUTINES WHICH ARE FULLY COMMENTED.           *
;*THE  TEMPERATUE LOOK UP TABLE IS LOCATED AT THE           *
;*VERY END OF THE PROGRAM. IT IS 1.5K BYTES LONG AND WILL  *
;*RESIDE FROM 0A00 TO 0FFF.  THE READER MUST ALWAYS KEEP   *
;*IN MIND THAT THIS IS NOT A VERY FANCY PROGRAM.  IT       *
;*EXCLUDES MANY LUXURIOUS FEATURES, BUT IT CAN BE ENHANCED *
;*VERY EASILY.                                              *
;*SPECIAL THANKS TO MARY OZARKA FOR DEVELOPMENT OF CALIB-   *
;*RATION ROUTINES AND LOOK UP TABLES.                       *
;*************************************************************
;
;
;
;
;
;
;IN THE BEGINNING CONSTANTS WILL BE ASSIGNED TO PROPER
;LABELS.
ALTCH    EQU P1.4
START    EQU P1.5
RDY      EQU P1.6
BELL     EQU $07
LF       EQU $0A
BLANK    EQU $20
PRMT     EQU $3E
CR       EQU $0D
POINT    EQU $2E
PCON     EQU $87
DATAL    EQU $F000
DATAH    EQU $E000
PDATAL   EQU $D000
PDATAH   EQU $C000
DSPLY    EQU $B000
;
;
;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.
;
;
;
INIT:     ORG $100          ;START LOADING THE PROGRAM FROM
;100H.
          MOV A,#$FF        ;TURN OFF THE 7 SEGMENTS
          MOV DPTR,#DSPLY
          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).
          ORL P1,#%01010000       ;MAKE RDY AN INPUT.
          ANL P1,#%01010000
          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
;
;
          CALL BAUD         ;SET UP THE SERIAL PORT.
          MOV A,#BELL
          CALL OUTCH        ;BEEP THE TERMINAL
          MOV DPTR,#$09E5
          CALL MESS         ;PRINT "URI TERM-X"
          CALL PROMT        ;PROMT THE USER.
;
;
;THE ROUTINE MAIN SAMPLES THE DESIRED CHANNELS IN
;INCREASING ORDER STARTING WITH CHANNEL ZERO.  IF AN ERROR
;OCCURS PROGRAM WILL STOP AND AN ERROR CODE WILL BE
;DISPLAYED ON TWO SEVEN SEGMENT DISPLAYS LOCATED ON THE
;FRONT PANEL. IN CASE OF ERROR RESET MUST BE PRESSED
;IF CONTINUATION IS DESIRED.
;THIS ROUTINE WILL DESTROY THE INITIAL VALUES OF R7 AND B.
;
;
MAIN:     EQU $
          CALL CHNL         ;GET THE NUMBER OF CHANNELS DESIRED
BEGIN:    CLR A             ;START FROM CH.ZERO.
MORE:     MOV R7,A
          CALL SAMPLE
          CALL CALIB        ;LOAD ACC. WITH THE OFFSET VALUE.
          CALL TEMP         ;DISPLAY THE TEMPERATURE.
          MOV A,R7          ;INCREMENT THE CHANNEL COUNT
```

```
          INC A
          CJNE A,B,MORE
          CALL CRLF          ;START FROM A NEW LINE
          JMP BEGIN
;
;
;
CALIB:    CLR A              ;THIS IS A DUMMY SUBROUTINE.
          RET
;
;
;
;THE FOLLOWING SUBROUTINE ASKS THE USER THE NUMBER OF
;CHANNELS DESIRED. IT EXPECTS AT MOST TWO INTEGERS TO BE
;ENTERED (MIN=1 AND MAX=16). THE RESULT WILL BE PLACED
;IN THE B REG. IN HEX FORMAT.
;THIS IS A VERY PRIMITIVE ROUTINE AND WILL NOT DETECT
;INVALID INPUTS.
;THIS SUBROTINE WILL OVERWRITE THE B REGISTER.
;
;
CHNL:     MOV DPTR,#$09BF
          CALL MESS                   ;PRINT THE MESSAGE.
          CALL ECHO
          MOV B,A
          ANL B,#$0F
          CALL INCH
          CJNE A,#CR,NOCR
          JMP HERE
NOCR:     CALL OUTCH
          ANL A,#$0F
          ADD A,#$A
          MOV B,A
          CALL INCH          ;WAIT FOR CR.
          CALL CRLF
          RET
HERE:     CALL CRLF
          RET
;THE SUBROUTINE SAMPLE RECEIVES A CHANNEL NUMBER THROUGH
;THE ACC. (UNUSED BITS IN THE ACC. MUST BE CLEAR).
;THIS ROUTINE WILL SAMPLE THE REQUESTED CHANNEL TWICE.
; THE AVERAGE 12-BIT RESULT (DATA FROM A/D)
;WILL BE ROUNDED TO 9-BITS BEFORE BEING STORED IN THE DPTR.
;THE MAIN PURPOSE OF HAVING MULTIPLE SAMPLES IS TO
;DIMINISH THE EFFECT OF THE DIGITAL NOISE ON THE ANALOG
;SIGNALS.
;THIS SUBROUTINE ALSO WILL DETECT BROKEN T-COUPLES IN THE
;CHANNELS BEING SAMPLED.
;R0,R1 AND R2 WILL LOSE THEIR INITIAL VALUES.
;
```

```
;
;
;
SAMPLE:   ANL P1,#$F0      ;CLEAR THE PREVIOUS CHANNEL NUM.
          ORL P1,A         ;SEND OUT THE CH. NUM. TO AIS.
          CLR ALTCH        ;LATCH THE CH.NUM.
          SETB START
          CLR START        ;START THE A/D
;AT THIS POINT A/D WILL START ITS 25 MICROSEC. CONVERSION.
          MOV DPTR,#DATAH
          MOV R0,A         ;SAVE ACC.
          JNB RDY,$        ;WAIT UNTIL THE END OF CONVERSION
          MOVX A,@DPTR     ;LOAD ACC. WITH D1-D8.
          MOV R2,A
          MOV DPTR,#DATAL
          MOVX A,@DPTR     ;LOAD D9-D12 IN THE ACC.
          JNB ACC.3,NOERR  ;CHECK FOR ERROR
ERR:      MOV A,R0         ;THE SAMPLED CHANNEL IS OPEN.
          CALL ERROR       ;CALL ERROR HANDLING ROUTINE
NOERR:    SETB START
          CLR START        ;ACTIVATE A/D FOR THE 2ND SAMPLE
          ANL A,#$F0
          MOV R1,A
          JNB RDY,$        ;WAIT FOR THE END OF 2ND CONVERSION
          MOVX A,@DPTR     ;READ IN THE 2ND SAMPLE.
          JB ACC.3,ERR     ;CHECK FOR ERROR
          ANL A,#$F0
          ADD A,R1
          JNB ACC.7,NONE
          INC R2
NONE:     MOV DPTR,#DATAH
          MOVX A,@DPTR
          ADDC A,R2
          MOV DPL,A
          MOV DPH,#$00
          JNB CY,NOCY
          INC DPH          ;THE FINAL 9BIT RESULT IS IN DPTR.
NOCY:     SETB ALTCH
          RET
;
;
;
;THIS SUBROUTINE SHOULD BE CALLED WHENEVER AN ERROR HAS
;BEEN DETECTED. AN ERROR MESSAGE WILL BE PRINTED ON THE
;SCREEN. THE ERROR CODE WILL BE DISPLAYED BY TWO ADJACENT
```

```
;DISPLAYS ON THE FRONT PANEL.
;THE SUBROUTINE ERROR EXPECTS AN ERROR CODE IN THE ACC.
;
;
;
ERROR:    MOV DPTR,#DSPLY
          MOVX @DPTR,A
          MOV DPTR,#$09B0
          CALL MESS
          MOV A,#BELL
          CALL OUTCH        ;CREATE A BEEP
          JMP $             ;LOOP UNTIL RESET IS PRESSED.
          RET               ;A DUMMY RETURN
;
;
;
;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 .
BAUD:     MOV TCON,#%00000001
          MOV SCON,#%01010000
          MOV TMOD,#%00100000
          MOV PCON,#$80   ;DOUBLE THE BAUD RATE
          MOV DPTR,#DATAL
          MOVX A,@DPTR    ;READ THE SETTING.
          ANL A,#%00000111
          CJNE A,#$0,B600
          MOV TH1,#$30    ;300 BAUD.
          JMP GOTIT
B600:     CJNE A,#$1,B1200
          MOV TH1,#$98    ;600 BAUD
          JMP GOTIT
B1200:    CJNE A,#$2,B2400
          MOV TH1,#$CC    ;1200 BAUD.
          JMP GOTIT
B2400:    CJNE A,#$3,B4800
          MOV TH1,#$E6    ;2400 BAUD.
          JMP GOTIT
B4800:    CJNE A,#$4,B9600
          MOV TH1,#$F3    ;4800 BAUD.
          JMP GOTIT
B9600:    CJNE A,#$5,B192
          MOV TH1,#$F9    ;9600 BAUD.
          JMP GOTIT
```

```
B192:     MOV TH1,#$FD    ;19.2K BAUD.
GOTIT:    SETB TR1        ;START THE TIMER.
          SETB TI         ;SET TI FOR 1ST CHARACTER.
          RET
;OUTCH: IS A SUBROUTINE WHICH SENDS OUT THE CONTENT OF
;THE ACC. TO THE SERIAL BUFFER OF THE 8031,ACC. MUST
;CONTAIN THE 7BIT ASCII REPRESENTATION OF DESIRED
;CHARACTER.
;NOTICE THAT NO PARITY IS USED.
;VALUE OF ACC. REMAINS UNTOUCHED.
;
OUTCH:    JNB TI,$        ;WAIT UNTIL THE END OF THE PREVIOUS
;TRANSMISSION.
          CLR TI          ;GET READY FOR THE CURRENT CHAR.
          MOV SBUF,A      ;SEND THE CURRENT CHAR.
          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.
;
;
INCH:     JNB RI,$        ;WAIT FOR RECEIVE INTERRUPT
          CLR RI          ;CLEAR THE RECEIVE INTERRUPT.
          MOV A,SBUF
          ANL A,#$7F
          RET
;
;
;
;CRLF: THIS IS A SUBROUTINE WHICH SENDS OUT A CR FOLLOWED
;BY A LF.
;ACC. IS CLEARED.
;
;
CRLF:     MOV A,#CR
          CALL OUTCH
          MOV A,#LF
          CALL OUTCH
          CLR A
          RET
;
```

```
;
;
;ECHO: IS A SUBROUTINE WHICH RECEIVES CHAR FROM THE
;KEYBOARD AND SENDS IT TO THE SCREEN.ACC. WILL PRESERVE
;THE RECEIVED CHAR.
;
;
;
ECHO:     CALL INCH        ;GET THE CHAR. FROM THE KEYBOARD.
          CALL OUTCH       ;ECHO THE CHARTO THE SCREEN
          RET
;
;
;
;PROMT: IS A SUBROUTINE WHICH PRINTS A PROMT IN THE
;BEGINNING OF THE NEXT LINE.
;
;
;
PROMT:    CALL CRLF
          MOV A,#PRMT
          CALL OUTCH
          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.
;INITIAL VALUE OF R4 WILL BE DESROYED.
;
;
MESS:     CALL CRLF
          MOV R4,#$00
NEXT:     MOV A,R4           ;SAVE THE INDEX
          MOVC A,@A+DPTR
          CJNE A,#CR,AGAIN ;STOP IF CR
          JMP FINISH
AGAIN:    CALL OUTCH       ;DISPLAY THE CHAR.
          INC R4           ;INCREMENT THE INDEX.
          JMP NEXT
FINISH:   CLR A
```

```
          RET
;
;
;
;TEMP: THIS SUBROUTINE WILL PRINT THE ACTUAL CENTIGRAD
;TEMPERATURE ON THE SCREEN. EACH READING CONSISTS OF
;FIVE CHARS.THE THIRD CHAR. IS A DECIMAL POINT,AND THE
;FIFTH ONE IS A BLANK. THIS SUBROUTINE EXPECTS THE DIGITAL
;9BIT DATA IN DPTR. ACC. SHOULD CONTAIN THE OFFSET VALUE
;READ FROM CALIBRATION RAM(IF CALIBRATION IS NOT USED
;ACC. MUST CONTAIN A ZERO).THE UNUSED BITS IN DPTR MUST
;BE ZERO.
;
;
TEMP:     MOV R5,A          ;SAVE ACC. IN R5.
          ORL DPH,#%00001110
          MOVC A,@A+DPTR ;GET THE FIRST DIGIT
          CALL OUTCH
          ANL DPH,#%11111101
          MOV A,R5
          MOVC A,@A+DPTR ;GET THE 2ND DIGIT.
          CALL OUTCH
          MOV A,#POINT
          CALL OUTCH        ;PRINT A DECIMAL POINT
          ORL DPH,#%00000010
          ANL DPH,#%11111011
          MOV A,R5
          MOVC A,@A+DPTR ;GET THE LAST DIGIT
          CALL OUTCH        ;PRINT THE LAST CHAR.
          MOV A,#BLANK
          CALL OUTCH        ;PRINT A SPACE IN THE END.
          RET
;
;
;THIS SUBROUTINE WILL
;DISPLAY THE CONTENT OF A DESIRED REGITER OR RAM
;LOCATION ON FRONT PANEL DISPLAYS. THE ADDRESS OF DESIRED
;LOCATION MUST BE PLACED IN RO. THIS ROUTINE WILL WAIT FOR
;A KEY STROKE.
;NOTICE THAT THIS IS ONLY FOR DEBUGGING PURPOSES.
TEST:     MOV A,@RO
          MOV DPTR,#DSPLY
          MOVX @DPTR,A
          CALL INCH         ;WAIT FOR A KEY STROKE.
          RET
```

```
;
;
;
RECAL:    RETI              ;THIS IS FOR TEST PURPOSES.

          END
```

hex number	temperature (C)	hex number	temperature (C)
0	0.0	18	2.0
1	0.0	19	2.1
2	0.0	1a	2.2
3	0.0	1b	2.3
4	0.0	1c	2.4
5	0.1	1d	2.5
6	0.2	1e	2.7
7	0.3	1f	2.8
8	0.4	20	2.9
9	0.5	21	3.0
a	0.6	22	3.1
b	0.7	23	3.2
c	0.8	24	3.3
d	0.9	25	3.4
e	1.0	26	3.5
f	1.1	27	3.6
10	1.2	28	3.7
11	1.3	29	3.8
12	1.4	2a	3.9
13	1.5	2b	4.0
14	1.6	2c	4.1
15	1.7	2d	4.2
16	1.8	2e	4.3
17	1.9	2f	4.4

hex number	temperature (C)	hex number	temperature (C)
30	4.5	48	7.0
31	4.6	49	7.1
32	4.7	4a	7.2
33	4.8	4b	7.3
34	4.9	4c	7.4
35	5.0	4d	7.5
36	5.1	4e	7.6
37	5.2	4f	7.7
38	5.3	50	7.8
39	5.4	51	7.9
3a	5.5	52	8.0
3b	5.6	53	8.1
3c	5.7	54	8.2
3d	5.8	55	8.3
3e	6.0	56	8.4
3f	6.1	57	8.5
40	6.2	58	8.6
41	6.3	59	8.7
42	6.4	5a	8.8
43	6.5	5b	8.9
44	6.6	5c	9.0
45	6.7	5d	9.1
46	6.8	5e	9.2
47	6.9	5f	9.3

hex number	temperature (C)	hex number	temperature (C)
60	9.4	78	11.9
61	9.5	79	12.0
62	9.6	7a	12.1
63	9.7	7b	12.2
64	9.8	7c	12.3
65	9.9	7d	12.4
66	10.0	7e	12.5
67	10.1	7f	12.6
68	10.3	80	12.7
69	10.4	81	12.8
6a	10.5	82	12.9
6b	10.6	83	13.0
6c	10.7	84	13.1
6d	10.8	85	13.2
6e	10.9	86	13.3
6f	11.0	87	13.4
70	11.1	88	13.5
71	11.2	89	13.6
72	11.3	8a	13.7
73	11.4	8b	13.8
74	11.5	8c	13.9
75	11.6	8d	14.0
76	11.7	8e	14.1
77	11.8	8f	14.2

hex number	temperature (C)	hex number	temperature (C)
90	14.3	a8	16.8
91	14.4	a9	16.9
92	14.5	aa	17.0
93	14.7	ab	17.1
94	14.8	ac	17.2
95	14.9	ad	17.3
96	15.0	ae	17.4
97	15.1	af	17.5
98	15.2	b0	17.6
99	15.3	b1	17.7
9a	15.4	b2	17.8
9b	15.5	b3	17.9
9c	15.6	b4	18.0
9d	15.7	b5	18.1
9e	15.8	b6	18.2
9f	15.9	b7	18.3
a0	16.0	b8	18.4
a1	16.1	b9	18.5
a2	16.2	ba	18.6
a3	16.3	bb	18.7
a4	16.4	bc	18.8
a5	16.5	bd	18.9
a6	16.6	be	19.0
a7	16.7	bf	19.1

hex number	temperature (C)	hex number	temperature (C)
c0	19.2	d8	21.7
c1	19.3	d9	21.8
c2	19.4	da	21.9
c3	19.5	db	22.0
c4	19.6	dc	22.1
c5	19.7	dd	22.2
c6	19.8	de	22.3
c7	19.9	df	22.4
c8	20.0	e0	22.5
c9	20.1	e1	22.6
ca	20.2	e2	22.7
cb	20.3	e3	22.8
cc	20.4	e4	22.9
cd	20.5	e5	23.0
ce	20.7	e6	23.1
cf	20.8	e7	23.2
d0	20.9	e8	23.3
d1	21.0	e9	23.4
d2	21.1	ea	23.5
d3	21.2	eb	23.6
d4	21.3	ec	23.7
d5	21.4	ed	23.8
d6	21.5	ee	23.9
d7	21.6	ef	24.0

hex number	temperature (C)	hex number	temperature (C)
f0	24.1	108	26.5
f1	24.2	109	26.6
f2	24.3	10a	26.7
f3	24.4	10b	26.8
f4	24.5	10c	26.9
f5	24.6	10d	27.0
f6	24.7	10e	27.1
f7	24.8	10f	27.2
f8	24.9	110	27.3
f9	25.0	111	27.4
fa	25.1	112	27.5
fb	25.2	113	27.6
fc	25.3	114	27.7
fd	25.4	115	27.8
fe	25.5	116	27.9
ff	25.6	117	28.0
100	25.7	118	28.1
101	25.8	119	28.2
102	25.9	11a	28.3
103	26.0	11b	28.4
104	26.1	11c	28.5
105	26.2	11d	28.6
106	26.3	11e	28.7
107	26.4	11f	28.8

hex number	temperature (C)	hex number	temperature (C)
120	28.9	138	31.3
121	29.0	139	31.4
122	29.1	13a	31.5
123	29.2	13b	31.6
124	29.3	13c	31.7
125	29.4	13d	31.8
126	29.5	13e	31.9
127	29.6	13f	32.0
128	29.7	140	32.1
129	29.8	141	32.2
12a	29.9	142	32.3
12b	30.0	143	32.4
12c	30.1	144	32.5
12d	30.2	145	32.6
12e	30.3	146	32.7
12f	30.4	147	32.8
130	30.5	148	32.9
131	30.6	149	33.0
132	30.7	14a	33.1
133	30.8	14b	33.2
134	30.9	14c	33.3
135	31.0	14d	33.4
136	31.1	14e	33.5
137	31.2	14f	33.6

hex number	temperature (C)	hex number	temperature (C)
150	33.7	168	36.1
151	33.8	169	36.2
152	33.9	16a	36.3
153	34.0	16b	36.4
154	34.1	16c	36.5
155	34.2	16d	36.6
156	34.3	16e	36.7
157	34.4	16f	36.8
158	34.5	170	36.9
159	34.6	171	37.0
15a	34.7	172	37.1
15b	34.8	173	37.2
15c	34.9	174	37.3
15d	35.0	175	37.4
15e	35.1	176	37.5
15f	35.2	177	37.6
160	35.3	178	37.7
161	35.4	179	37.8
162	35.5	17a	37.9
163	35.6	17b	38.0
164	35.7	17c	38.1
165	35.8	17d	38.2
166	35.9	17e	38.3
167	36.0	17f	38.4

hex number	temperature (C)	hex number	temperature (C)
180	38.5	198	40.9
181	38.6	199	41.0
182	38.7	19a	41.1
183	38.8	19b	41.2
184	38.9	19c	41.3
185	39.0	19d	41.4
186	39.1	19e	41.5
187	39.2	19f	41.6
188	39.3	1a0	41.7
189	39.4	1a1	41.8
18a	39.5	1a2	41.9
18b	39.6	1a3	42.0
18c	39.7	1a4	42.1
18d	39.8	1a5	42.2
18e	39.9	1a6	42.3
18f	40.0	1a7	42.4
190	40.1	1a8	42.5
191	40.2	1a9	42.6
192	40.3	1aa	42.7
193	40.4	1ab	42.8
194	40.5	1ac	42.9
195	40.6	1ad	43.0
196	40.7	1ae	43.1
197	40.8	1af	43.2

hex number	temperature (C)	hex number	temperature (C)
1b0	43.3	1c8	45.7
1b1	43.4	1c9	45.8
1b2	43.5	1ca	45.9
1b3	43.6	1cb	46.0
1b4	43.7	1cc	46.1
1b5	43.8	1cd	46.2
1b6	43.9	1ce	46.3
1b7	44.0	1cf	46.4
1b8	44.1	1d0	46.5
1b9	44.2	1d1	46.6
1ba	44.3	1d2	46.7
1bb	44.4	1d3	46.8
1bc	44.5	1d4	46.9
1bd	44.6	1d5	47.0
1be	44.7	1d6	47.1
1bf	44.8	1d7	47.2
1c0	44.9	1d8	47.3
1c1	45.0	1d9	47.4
1c2	45.1	1da	47.5
1c3	45.2	1db	47.6
1c4	45.3	1dc	47.7
1c5	45.4	1dd	47.8
1c6	45.5	1de	47.9
1c7	45.6	1df	48.0

hex number	temperature (C)	hex number	temperature (C)
1e0	48.1	1f0	49.6
1e1	48.2	1f1	49.7
1e2	48.3	1f2	49.8
1e3	48.4	1f3	49.9
1e4	48.5	1f4	50.0
1e5	48.6	1f5	50.1
1e6	48.7	1f6	50.2
1e7	48.8	1f7	50.3
1e8	48.8	1f8	50.4
1e9	48.9	1f9	50.5
1ea	49.0	1fa	50.6
1eb	49.1	1fb	50.7
1ec	49.2	1fc	50.8
1ed	49.3	1fd	50.9
1ee	49.4	1fe	51.0
1ef	49.5	1ff	51.1

APPENDIX B

THE LOOK UP TABLE GENERATION PROGRAM

```
/* Generation of values for look up table */
/* Given hex 000 to 1FF, find the corresponding */
/* temperature between 0 and 51.2 C degrees */
/* Mary Ozarka */
/* July 25, 1983 */
/* Version 1 */
/* C language */

#include <stdio.h>
main()
{
        int i, j, k1, k2, resp, w, k, z;
        float g1, g2, s1, s2, inc, v1, vad, mid, m;
        float a[521], r[513], c[513], s[4096];

        printf("Default values for gains g1 and g2?\n");
        printf("(g1=192.30 and g2=9.730007)\n");
   ask: printf("type 1 (=yes) or 0 (=no)\n");
        scanf("%1d", &resp);
        if (resp == 1) {
                g1 = 192.3000000;
                g2 = 9.730007;
        } else if (resp == 0) {
                printf("value for g1?\n");
                scanf("%f", &g1);
                printf("value for g2?\n");
                scanf("%f", &g2);
        } else
                goto ask;
        printf("g1 = %6.2f   g2 = %f\n\n", g1, g2);

/* vad = [(vth + 0.023 mv) * g1] * g2 */
/* vad/5v = hex number between 000 and FFF */
/* a[i] holds the rated value of voltage (in millivolts) */
/* corresponding to temperature i/10 */
        a[0] = 0.000;
        a[10] = 0.050;
        a[20] = 0.101;
        a[30] = 0.151;
        a[40] = 0.202;
        a[50] = 0.253;
        a[60] = 0.303;
        a[70] = 0.354;
        a[80] = 0.405;
        a[90] = 0.456;
        a[100] = 0.507;
        a[110] = 0.558;
        a[120] = 0.609;
        a[130] = 0.660;
        a[140] = 0.711;
```

```
                }
                else
                        printf("%6x %14.1f", k, s[j]);
                        printf("%17x %14.1f\n\n", z, s[j+192]);
                k = k + 1;
                z = z + 1;
        }
}
```

```
a[150] = 0.762;
a[160] = 0.813;
a[170] = 0.865;
a[180] = 0.916;
a[190] = 0.967;
a[200] = 1.019;
a[210] = 1.070;
a[220] = 1.122;
a[230] = 1.174;
a[240] = 1.225;
a[250] = 1.277;
a[260] = 1.329;
a[270] = 1.381;
a[280] = 1.432;
a[290] = 1.484;
a[300] = 1.536;
a[310] = 1.588;
a[320] = 1.640;
a[330] = 1.693;
a[340] = 1.745;
a[350] = 1.797;
a[360] = 1.849;
a[370] = 1.901;
a[380] = 1.954;
a[390] = 2.006;
a[400] = 2.058;
a[410] = 2.111;
a[420] = 2.163;
a[430] = 2.216;
a[440] = 2.268;
a[450] = 2.321;
a[460] = 2.374;
a[470] = 2.426;
a[480] = 2.479;
a[490] = 2.532;
a[500] = 2.585;
a[510] = 2.638;
a[520] = 2.691;
c[0] = 0.0;
for (i = 0; i <= 512; ++i) {
/* assign linear values between the given a[i]'s */
          if (i % 10 != 0) {
                    k1 = i/10 + 1;
                    k2 = k1 - 1;
                    s1 = a[10 * k1];
                    s2 = a[10 * k2];
                    inc = (s1 - s2)/10;
                    a[i] = a[i-1] + inc;
          }
          v1 = s1 * (a[i] + 0.0230);
```

```
        vad = s2/1000 * v1;
        r[i] = 4096 * vad/5;
        /* c[i] stores the temperatures */
        if (i != 0)
                c[i] = c[i-1] + 0.10;
}

/* Find corresponding temperature for hex values */
i = 0;
for (j = 0; j <= 4095; ++j) {
        m = j;
        if (m - r[i+1] >= 0) {
                w = m - r[i+1] + 1;
                i = i + w;
        }
        mid = (r[i+1] - r[i])/2 + r[i];
        if (m < mid)
                s[j] = c[i];
        else
                s[j] = c[i+1];
}

/* Print out temperature resulting from the */
/* given hex value. */
k = 0;
z = 24;

for (j = 0; j <= 3960; j = j + 8) {
        if (j == 0) {
                printf("\n\n\n\n\n\n");
                printf("hex number  ");
                printf("temperature (C)       ");
                printf("hex number  ");
                printf("temperature (C)\n\n");
                }
        else if (j % 192 == 0) {
                printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
                printf("hex number  ");
                printf("temperature (C)       ");
                printf("hex number  ");
                printf("temperature (C)\n\n");
                k = k + 24;
                z = z + 24;
                j = j + 192;
        }
        if (k == 480)
                z = z - 8;
        if (k >= 480) {
                printf("%6x %14.1f", k, s[j]);
                printf("%17x %14.1f\n\n", z, s[j+128]);
```

REFERENCES

Omega, [1983a]. "Temperature Measurement Handbook," T-12.

Analog Devices, [1982]. "Monolithic Thermocouple Amplifier with Cold Junction Compensation," 1-8.

Omega, [1983b]. "Temperature Measurement Handbook," T-26.

MCS-51 User's Manual, [1981a]. "MSC-51 Family of Single Chip Microcomputers User's Manual," Intel Corporation, 2.1-2.5.

Microcontroller User's Manual, [1982]. Intel Corportion, 6.6-6.9.

Microcontroller Applications Handbook, [1982a]. Intel Corporation, 2-83.

Microcontroller Applications Handbook, [1982b]. Intel Corporation, 2-91.

Mostek,[1983]. "Zero Power RAM," United Technologies Mostek, 1-16.

University of Illinois at Urbana-Champaign

DEPARTMENTAL FORMAT APPROVAL

THIS IS TO CERTIFY THAT THE FORMAT AND QUALITY OF PRESENTATION OF THE THESIS SUBMITTED BY ALFRED GHARAKHANI AS ONE OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IS ACCEPTABLE TO THE DEPARTMENT OF ELECTRICAL ENGINEERING.

(Full Name of Department, Division or Unit)

2 December 1983

Date of Approval

Departmental Representative