National Library of Canada

Acquisitions and Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada

Direction des acquisitions et des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

A LOW-POWER DATA ACQUISITION SYSTEM FOR SCOLIOSIS STUDIES

BY

EDMOND HOK-MING LOU

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE.

DEPARTMENT OF ELECTRICAL ENGINEERING

EDMONTON, ALBERTA
SPRING 1993

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-82137-X

Canada

UNIVERSITY OF ALBERTA
RELEASE FORM

NAME OF AUTHOR : EDMOND HOK-MING LOU

TITLE OF THESIS: A Low-Power Data Acquisition System for Scoliosis Studies

DEGREE : Master of Science

YEARS THIS DEGREE GRANTED : 1993

PERMISSION IS HEREBY GRANTED TO THE UNIVERSITY OF ALBERTA LIBRARY TO REPRODUCE SINGLE COPIES OF THIS THESIS AND TO LEND OR SELL SUCH COPIES FOR PRIVATE, SCHOLARLY SCIENTIFIC RESEARCH PURPOSES ONLY.

THE AUTHOR RESERVES OTHER PUBLICATION RIGHTS, AND NEITHER THE THESIS NOR EXTENSIVE EXTRACTS FORM IT MAY BE PRINTED OR OTHERWISE REPRODUCED WITHOUT THE AUTHOR'S WRITTEN PERMISSION.

(Student's Signature)

7, 1/F Kam Chi Path
Ngau Chi Wan
Kowloon
Hong Kong

Date : April 20, 93

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

THE UNDERSIGNED CERTIFY THAT THEY HAVE READ, AND RECOMMEND TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH FOR ACCEPTANCE, A THESIS ENTITLED **A LOW-POWER DATA ACQUISITION SYSTEM FOR SCOLIOSIS STUDIES** SUBMITTED BY **EDMOND HOK-MING LOU** IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF **MASTER OF SCIENCE**.

N. G. Durdle

Dr. N.G. Durdle
Co-Supervisor

V J Raso

Mr. V.J. Raso
Co-Supervisor

Sun Xiaoling

Dr. X. Sun

Dr. G. Russell

Date : April 16/93

Abstract

Total contact braces are often prescribed for children with spinal deformities. A battery power microcomputer system has been developed to record the pressures between a brace and the trunk over a two week treatment period. Specially designed transducers are placed at the acting surface of each pad of the brace. Data is collected using an analog system with a maximum of 16 transducers to measure brace pressures by a MC68HC11 microcontroller. To minimize power consumption, low power CMOS circuits are used and the microcomputer and analog circuitry are switched to a low power operating mode except at the specific times when the transducer are being sampled. A programmable real time clock (RTC) is used to control the sample time and interval, and to provide an interrupt to awaken the microcomputer from the low power sleep mode. Data can be sampled with intervals ranging from 1 second to 1 hour and with measurement periods ranging from 2.5 hours to 341 days. At the end of each study period the data is transmitted to PC for analysis. The system is small, light and robust and easily carried by the patient.

Acknowledgment

I would like to thank Doug Hill for his helpful suggestions and in the preparation of this thesis. I wish to express my profound gratitude to my supervisors Dr. Nelson G. Durdle and V James Raso, for their efforts in guiding me through the process of writing and producing this thesis and thank for their patience, their guidance and their encouragement throughout my work. I would like to thank my friend Albert Lau and Selina Shum for their valuable assistance in drawing diagrams. I would like to thank my parents for their support and encouragement.

Last but not least, I wish to thank the Almighty God who gave me strength and confidence which made me completion of paper possible.

The financial support of the Department of Electrical Engineering, University of Alberta and the Glenrose Rehabilitation Hospital are gratefully acknowledged.

TABLE OF CONTENTS

CHAPTER .... PAGE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

a'	a complement or a*
A	Ampere
A/D	Analog to Digital
AS	Address Strobe
CE	Chip Enable
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
DeMux	Demultiplexer
DS	Data Strobe
E	System Clock Signal
hr	hour
Hz	Hertz
IC	Integrated Circuit
IS	Idiopathic Scoliosis
k	Kilo
LSB	Least significant bit
M	Mega
mA	Millampere
mAh	Millampere-hour
mAsec	Millampere-second
MCU	Microcontroller Unit
μ	micro
mmHg	Millimeter of mercury
min	minute
Mux	Multiplexer

NiCad	Nickel-Cadmium
OP-AMP	Operational Amplifier
Pa	Pascal
PA	Posterior-Anterior or posteroanterior
PAL	Programmable Array Logic
PC	Personal Computer
Psi	Pound-force per square inch
RAM	Random Access Memory
ROM	Read Only Memory
RTC	Real Time Clock
R/W	Read/Write
sec	second
V	Voltages

1. INTRODUCTION

Scoliosis is derived from the Greek word meaning curvature. When used in medical literature, it means a lateral curvature of the spine. A normal spine has physiological curvatures when viewed from the side. Since there is no lateral derivation when viewed anteriorly or posteriorly, scoliosis is an abnormal finding. Idiopathic scoliosis (IS) is lateral deviation of the spine for which there is no known cause and is the most common form. It occurs during the growing years and is customarily divided into three categories : infantile, juvenile and adolescent. These are classified according to the age at which the deformity is first noticed and does not necessarily coincide with the time the spinal curvature first appears.

Scoliosis which is most commonly found in adolescents is defined by the Scoliosis Research Society as a lateral spinal curvature of 10° or greater (Kehl and Morrissy, 1988) with vertebral rotation. Spinal curvature less than 10° is viewed as a variation of normal spinal anatomy. When a progressive curve is detected, aggressive treatment is indicated. Historically a brace, a type of plastic corset, has been used to support mechanically the spine when risk of progression is greatest (Asher *et al*, 1986). Curvature more than 50° usually requires surgical correction and stabilization to prevent continuing collapse. The first successful fusion of the spine was performed by Hibbs in 1914 but at that time there was no systematic method for producing correction. In 1920 a turnbuckle cast was introduced and for the first time provided a consistent means of achieving curve correction. All instrumentation systems accomplish curve correction by applying a set of forces to the spine. Curvature less than 50° are frequently treated non-operatively; the common objective is to prevent the progression of mild curves in adolescents with growth potential. This may be achieved using the combined effects of prolonged application of corrective spinal bracing and a controlled regimen of exercise (Chase *et al*, 1989).

Clinicians have few non-surgical treatment tools for children with progressive scoliosis. Among several non-operative treatments for scoliosis, brace treatment is the most widely accepted despite poor compliance and much uncertainty as to effectiveness (Houghton *et al*, 1986 and Ylikoski *et al*, 1989). Kehl and Morrissy (Kehl and Morrissy, 1988) suggested that current guidelines for initiating bracing in adolescent idiopathic scoliosis must address the relationship between curve magnitude, curve location, and the patient's skeletal maturity.

In the late 1940s, the Milwaukee brace was introduced as an orthosis that could be used to treat progressive scoliosis. In 1976, the Milwaukee brace was the most frequently used brace in the United States for non-operative treatment of IS (Andriacchi *et al*, 1976). The design and construction of the Milwaukee brace has evolved away from the initial concept with mandibular and occipital distraction. It emphasized passive correction as well as active derotation with a medially directed force. The shoulder slings, thoracic pads, and the lumbar pads of the brace can apply forces of different magnitudes, in different directions, and at different points to correct single, double, and triple curves (Andriacchi *et al*, 1976). However, the conventional Milwaukee brace has undergone many modifications (Watts, Hall and Stanish, 1977). In 1971 Hall and Miller modified the Milwaukee brace design to develop the Boston brace. The most dramatic change was the removal of the metal uprights.

The Boston brace is constructed from a prefabricated polypropylene pelvic module (Watts, Hall and Stanish, 1977). No individual plaster mold is required, and so it can save time for the patient and the orthotist. The module is trimmed into a "girdle" and shaped to the needs of an individual patient. The prefabricated module has a hard polypropylene plastic shell with a soft foam polyethylene lining providing a close fit to the pelvis and an excellent "grip" which is used as a foundation for applying forces to the spine. Figure 1-1 and Figure 1-2 show two views of a Boston brace. The symmetry of the inner surface of the orthosis is altered only by the placement of pads which make contact with chosen areas of the patient's torso. These pads provide the mechanical support for the spine, and their positions depend on the degree of rib cage rotation and the location of the curve. The biomechanical concept of the bracing system is to provide three-dimensional dynamic correction of the spine (Figure 1-3). This is achieved by using a prefabricated polypropylene module lined with polyethylene foam. The anatomic configuration of the module provides 15° of lumbar flexion, increased anterior abdominal force, fixation at the waist, and lateral torso containment (Chase *et al*, 1989).

The Charleston bending brace (Figure 1-4) is also used to treat children with scoliosis and is based on the bending moment concept described by Hibbs *et al* in 1931 (Price *et al*, 1990). In 1978, Ralph Hooper, CPO, began fabricating a side-bending orthosis for Frederick Reed, MD, of Charleston, SC. The Charleston brace is used for patients who were almost skeletally mature but continue to have curve progression. Also, it is used to hold a patient in a position of maximum side bend correction and is worn only at night time because ambulation is difficult in the bent position (Price *et al*, 1990). The brace is custom fabricated beginning with a negative plaster model of the patient. After the negative mold is completed, a positive plaster mold is fabricated. The positive

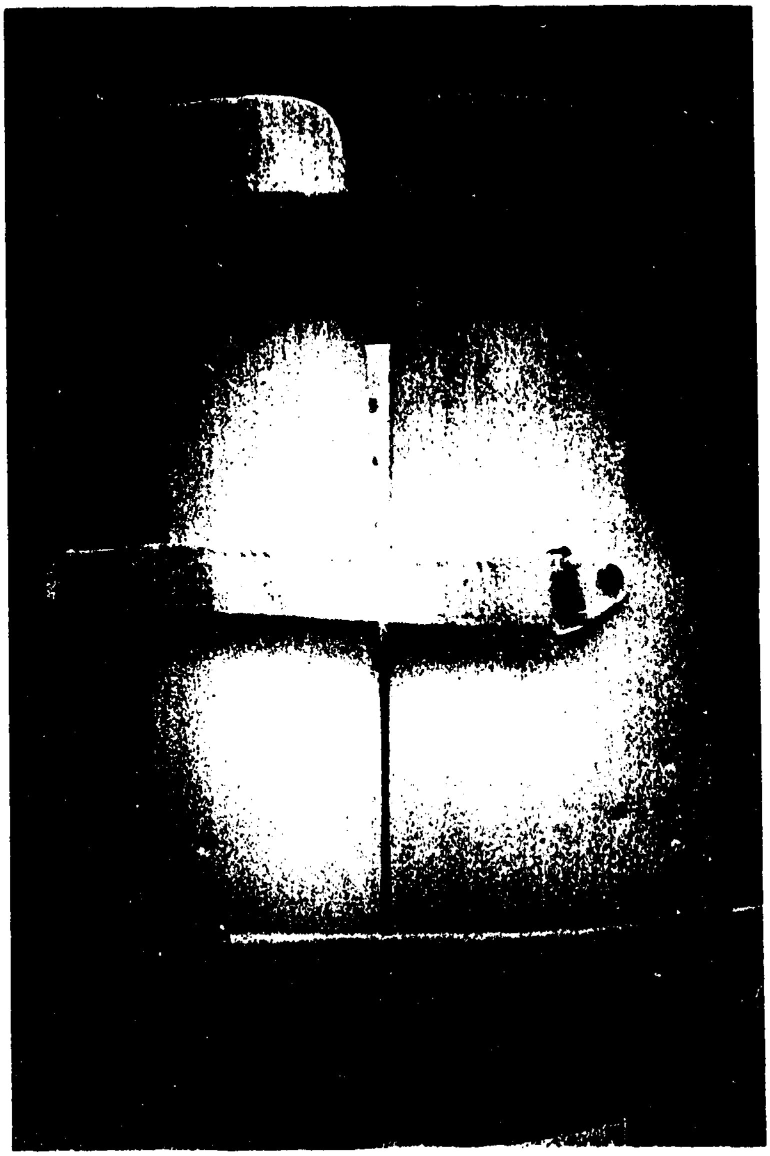


Figure 1-1 Back view of a Boston brace

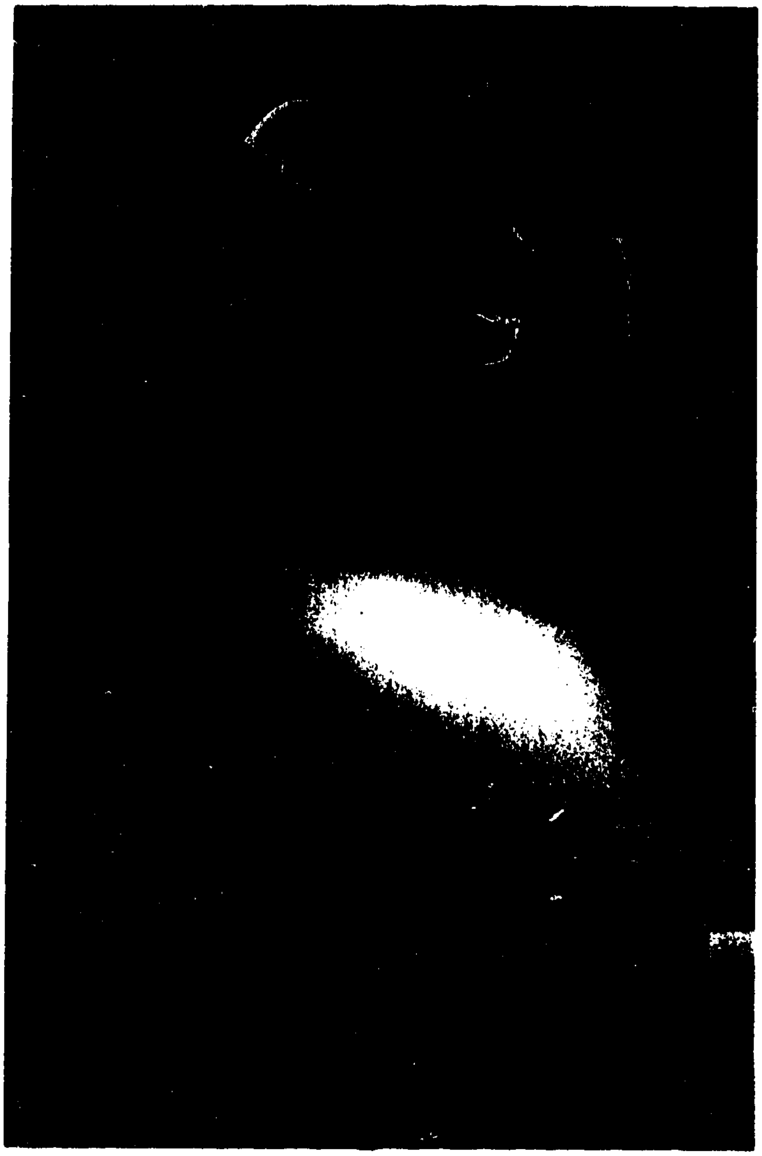


Figure 1-2 Side view of a Boston brace

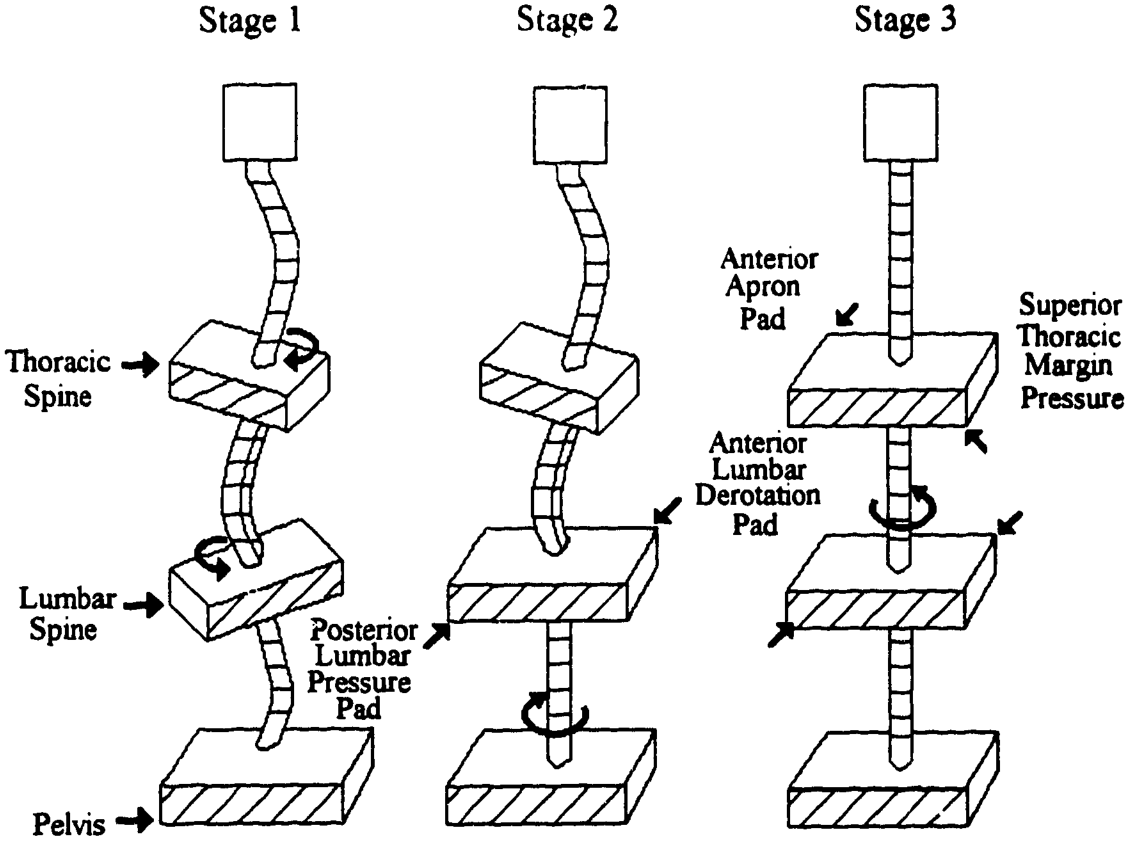


Figure 1-3 Three stages of corrective force application to the spine

Notes : Compression pads are positioned according to the degree of rotation and the apex of the curves.

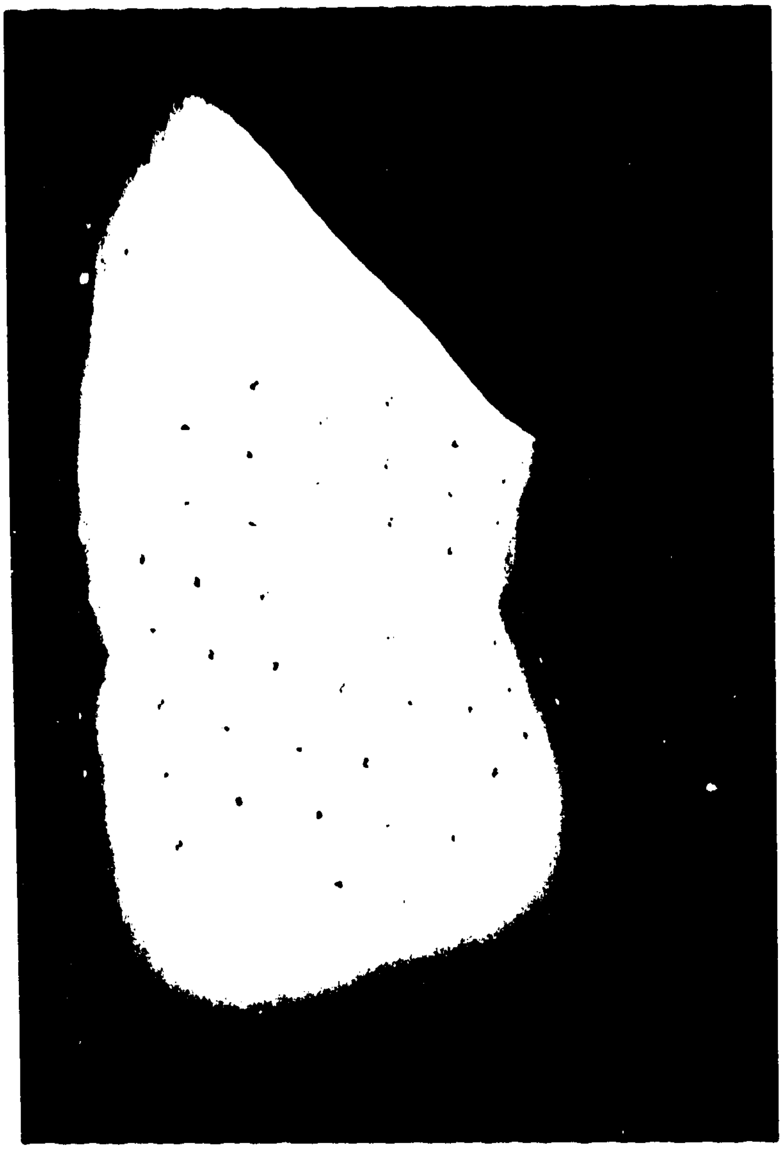


Figure 1-4 A Charleston bending brace

model is modified to meet established guidelines that will ensure maximum in-brace correction. A soft foam padding material is placed on the mold in areas of maximum pressure. After an initial adjustment period the brace is worn a minimum of eight hours each night.

The Milwaukee, Boston and Charleston braces provide mechanical support for the spine by means of pressure pads. Which brace is chosen depends on the location of the curve. The Boston brace is most effective in controlling curves in the lumbar and thoracolumbar regions whereas the Milwaukee brace is favored for the thoracic curves. The Charleston brace is used for patients who were skeletally mature but continue to have curve progression. Today, Boston and Charleston braces are favored because their low profile permits them to be hidden from view with appropriate clothing. The degree of support and the extent of corrective action depends on the location, magnitude and direction of the pressures exerted by the pressure pads relative to the spine. In the Milwaukee brace, the pads are attached to metal uprights secured in a molded bucket, but in the Boston and Charleston brace, these pads are embedded in the inner surface of a full contact polypropylene body brace. The actual corrective forces in terms of magnitude and direction developed by the brace are still not clear (Chase *et al*, 1989). The effectiveness of braces is limited by the ability of the underlying soft tissues to support and transmit the loads imposed by these pads. Although the support that the pads provide the spine has been described in the frontal plane, the three dimensional action of the brace on the spine has not been well investigated.

Weaning a patient from the brace depends upon the stability of correction achieved. Usually a patient wears a brace for at least a year on a full time basis before considering weaning. The full time program for Boston and Milwaukee brace wear is 23 hours per day. Criteria for weaning include (Moe *et al*, 1987): (1) no increase in heigh over a four month period, and (2) a Risser sign of at least 4 which means significant spi growth has ceased.

1.1 Literature Review of Brace Treatment for Scoliosis.

The area of brace treatment for children with spinal deformity has been based largely on instinct with very little hard data to support treatment decisions. As Patwardhan stated (Patwardhan *et al*, 1986) " The development of orthosis for scoliosis has been capricious, random, and the result of trial and error." In general, the extent of scoliosis present is determined radiographically according to Cobb in 1948 in a standing relaxed posteroanterior (PA) position. The "Cobb angle" is the angle between the top plate of the upper neutral vertebrae and the bottom plate of the lower neutral vertebrae. The measurement is shown in (Figure 1-5). The effectiveness of the Boston, Charleston and Milwaukee braces has been traditionally evaluated by assessing radiological changes of spine curvature in the frontal plane. Arao *et al* (Arao *et al*, 1981) suggested that the Boston brace does de-rotate the spine but they based this conclusion on an analysis of spinal alignment with the patient lying, i.e. no load on the spine, compared to conventional radiographs taken standing, i.e. with the spine supporting for full body weight. Also, by looking at in-brace curvatures, according to the Cobb's angle measurement, Willner (Willner, 1984) showed that the lateral forces are more important than the elongation forces when studying the correction of spinal deformities. The lateral push on the trunk can correct the lateral deviation as well as the rotation. Weisz *et al* (Weisz *et al*, 1989) suggested that although the Boston brace did not have a significant effect on radiological

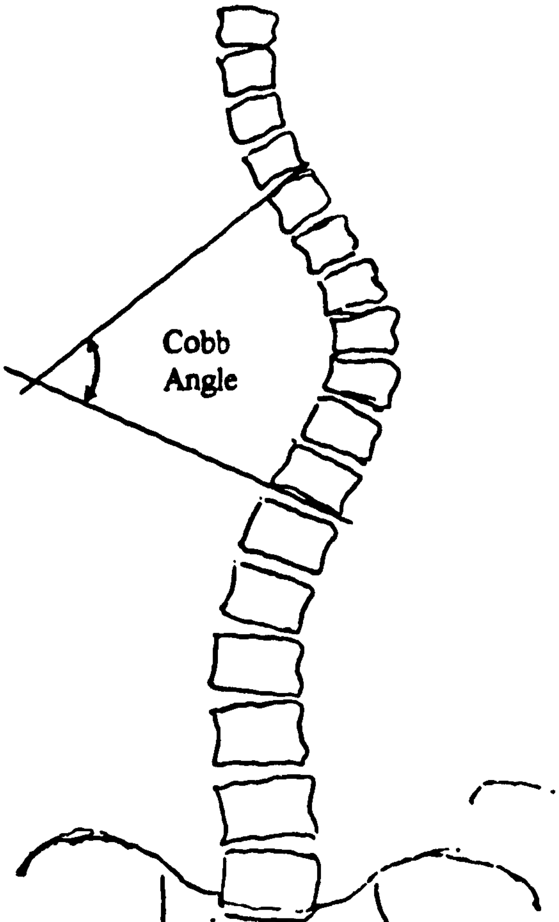


Figure 1-5 Cobb Angle Measurement

measures of deformity, it did improve trunk asymmetry significantly. Chase *et al* (Chase *et al*, 1989) studied the relationship between the brace forces and the curve correction on 14 patients, and found a negative correlation. They did not document the line of action of the pad forces nor their location relative to the spine because of the absence of suitable transducers to measure pressure directly at the interface between soft tissues and the brace. Dansereau *et al* (Dansereau *et al*, 1991) did a stereoradiographic study of 40 idiopathic scoliotic patients and showed that the Boston brace does not correct the three dimensional deformity but increased the thoracic hypokyphosis without effecting the rotation of the spine or the rib cage; even though the Cobb angle in the frontal plane was reduced. Emans (Emans *et al*, 1986) indicated that the long duration of spinal growth in younger braced patients allows both a greater chance of correction by an effective brace and a greater risk of further progression to surgery in a curve that does not respond well to bracing. To examine the biomechanical impact of Boston brace treatment for idiopathic scoliosis, Cote *et al* (Cote *et al*, 1992) used thin polymeric Force Sensing Resistors (Interlink Electronics Co., Oxford) which were mounted on a flexible tissue matrix, to measure the pressure distribution generated by the brace. However, in this study, pressures at three positions: the right side, the left side and the anterior part of the brace were measured in three steps. Therefore, the results do not reflect the pressure pattern at a specific time. Also, this method did not allow dynamic measurement on the whole interface to evaluate sitting or walking with the brace. Jiang *et al* (Jiang *et al*, 1992) used the Oxford Pressure Monitor (BADER, 1985, Talley Group Ltd., Oxford) to measure the magnitude, location and direction of pressure generated by the brace as well as forces imposed by the straps. Jiang found that there was a considerable variation in the strap tension of individuals. Some children secured the brace very aggressively and imposed high loads on their trunks; other children wore the brace loosely imposing very little pressure. These measurements were taken at a single time under laboratory conditions and may or may not be a true indicator of the pressure exerted by brace during the treatment period. Jiang found that the mean maximum pressures exerted by the brace were less than 3 kPa with low strap forces to over 7 kPa with high strap forces. The effectiveness of brace treatment seems to be related to the tightness of the straps of the brace.

2. JUSTIFICATION

Although the Boston brace is widely used for a non-surgical treatment, there has not been any well defined investigation of its three dimensional action on the spine or the variation of pad pressures over time. It is accepted that the brace pressures are the major factor in the correction of the curve, but little study has been devoted to the relationship between the pressures that the brace elicits, either passively or though muscle action and the response of the scoliotic spine. Only few clinical studies using three-dimensional reconstruction techniques (Asher *et al.*, 1987, Shinoto *et al.*, 1987) have been done to study the three-dimensional spinal morphological changes due to Boston brace wear. The location, magnitude and direction of forces exerted by the pressure pads relative to the spine are important because they influence the mechanical support and the extent of corrective action of the brace. Chase *et al* (Chase *et al*, 1989) and Jiang *et al* (Jiang *et al*, 1992) used the Oxford Pressure Monitor (Figure 2-1) to measure the pressure distribution at the interface between the Boston brace and patients' trunks. However, the monitor is heavy, so the measurement can only be recorded in the laboratory. Also, the force exerted by each strap that secures the brace is required. Chase *et al* attached a transducer (S-Sensor 70D, Orbit Controls Ltd., Cheltenham, United Kingdom) to the center of the straps and was able to record loads in the range of 0 to 250 N. Jiang used a strain gauge mounted on an aluminum insert connecting the strap with the hoop on the brace. Cote *et al* (Cote *et al*, 1992) used polymeric Force Sensor Resistors to measure the pressure distribution generated by the Boston brace. However, all the positions cannot be measured simultaneously, overlapping measurements as a result of dividing the measurements into three parts and longitudinal measurement over a large area of the surface are the weakness of this technique. The biomechanical action of the Boston brace on three-dimensional scoliotic deformities is thus still not well known. The decision to brace children with progressive scoliosis has been largely based on intuition without adequate scientific basis. The major question as to how the various brace components should be used to achieve optimum correction for different types of scoliotic curves has not been answered. An understanding of the relations between the pressures generated by wearing the brace should prove useful in the training of clinicians, in the design and modifications of the brace, and in the design of new devices for the correction of idiopathic scoliosis.

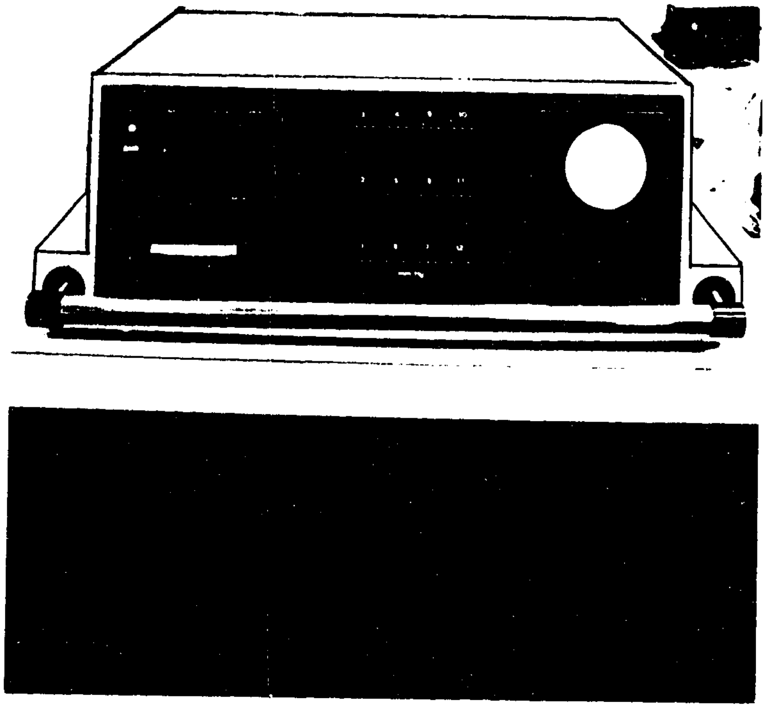


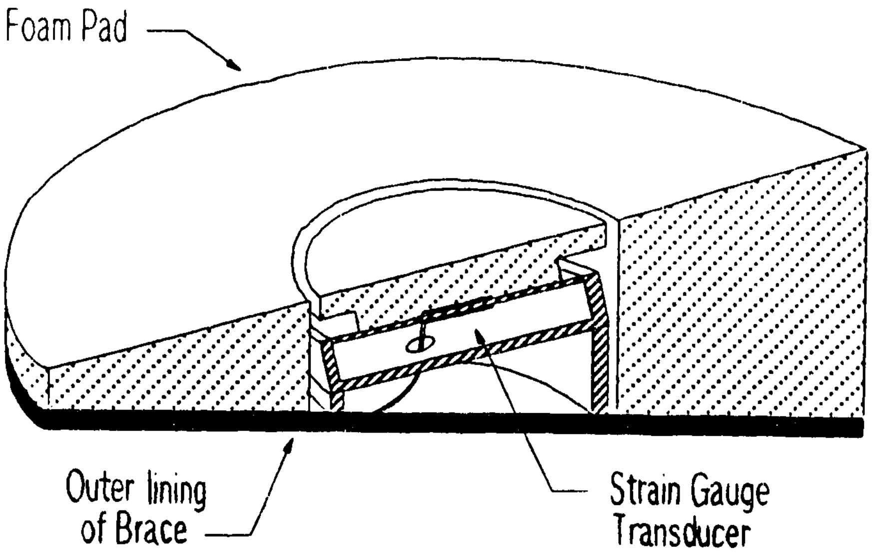
Figure 2-1 The Oxford Pressure Monitor System

3. OBJECTIVES

The objective of this work is to develop a system to measure the mechanical forces exerted by braces on children with idiopathic scoliosis. It is necessary to monitor the pressures exerted by braces over an extended study period. This thesis describes a low-power programmable data acquisition system to monitor continuously pressures and strap [illegible]es in braces used to treat children with scoliosis. Specialized pressure transc[illegible]s that can be embedded into the shell of braces are described.

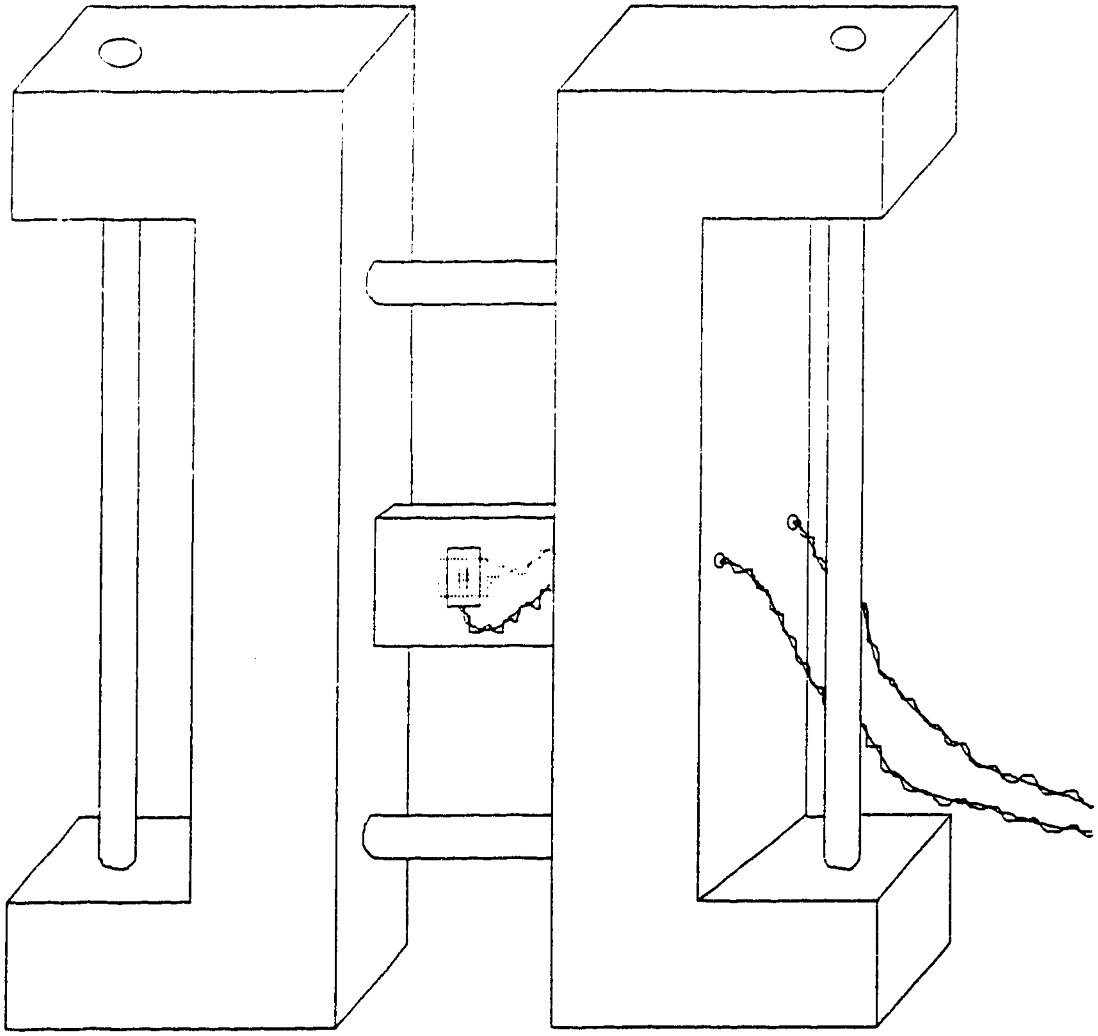
4. SYSTEM CONSTRAINTS AND REQUIREMENTS

A programmable system was designed to measure and record the pressure exerted by braces used to treat children with scoliosis. Small size, light weight and ruggedness are essential because the system must be carried by an adolescent during treatment. A compromise between the weight and the ampere-hour capacity of a battery is necessary because the battery is the heaviest component, and its capacity is directly proportional to its weight. Small size and low power consumption can be achieved by using a minimum number of CMOS ICs and turning off the power for any IC not in use. To have sufficient data for analysis, static RAM with sufficient memory for at least two weeks must be used. Length of the monitoring period was chosen as a comprise between power constraints and the necessity to have a good representation of the range of activity during brace treatment. A backup battery is provided to maintain memory in case of a power interruption. The data collection period and the sampling frequency are supplied under software control. Measurement of forces exerted by the various pads of braces requires the placement of pressure transducers at the acting surface of each pad. The pressure transducers are covered with foam to match the mechanical compliance of the pads (Figure 4-1). Transducers must be inexpensive, small, consume low power and highly sensitivite. Transducers must be temperature compensated and not interfere with the function of the brace. Since no commercially available transducers satisfy these special requirements, it was necessary to design a transducer specific for this application. Also, buckles (Figure 4-2) were designed to attach to a brace to monitor the tension in the straps of the brace. The pressure generated by the brace and the forces exerted on the straps will be measured by the same system.



Scale 3 : 1

Figure 4-1 Mounting the transducer on the brace



Scale : 2.5 : 1

Figure 4-2 Designed Buckle

5. SYSTEM DESIGN [1]

The system may be divided into two main sub-systems: (1) the programmable digital data acquisition system and (2) the transducer system and associated analog circuitry. Experiments and measurements were conducted to define optimum design parameters for each of system.

5.1 The Programmable Digital Data Acquisition System

The programmable requirement of the data acquisition system dictated the use of a microprocessor or microcontroller. The Motorola MC68HC11 CMOS eight-bit microcontroller integrated circuit was chosen for this purpose. This IC was chosen primarily because of its low power consumption and built-in A/D converter. Also, this MCU can be programmed to be in a low power (STOP) mode except at the specific time when it is acquiring data. The analog data can be converted by the A/D converter and stored in static RAM. Thus the system required both hardware and software design.

5.1.1 The Hardware Design

The block diagram of the digital system is shown in Figure 5-1. The main component of this system is the MC68HC11A1 microcontroller. It is a low power CMOS eight-bit microcontroller with highly sophisticated on-chip peripheral capabilities. The

---

[1] *A version of this chapter is being submitted for publication in Medical & Biological Engineering & Computing, 1993.*

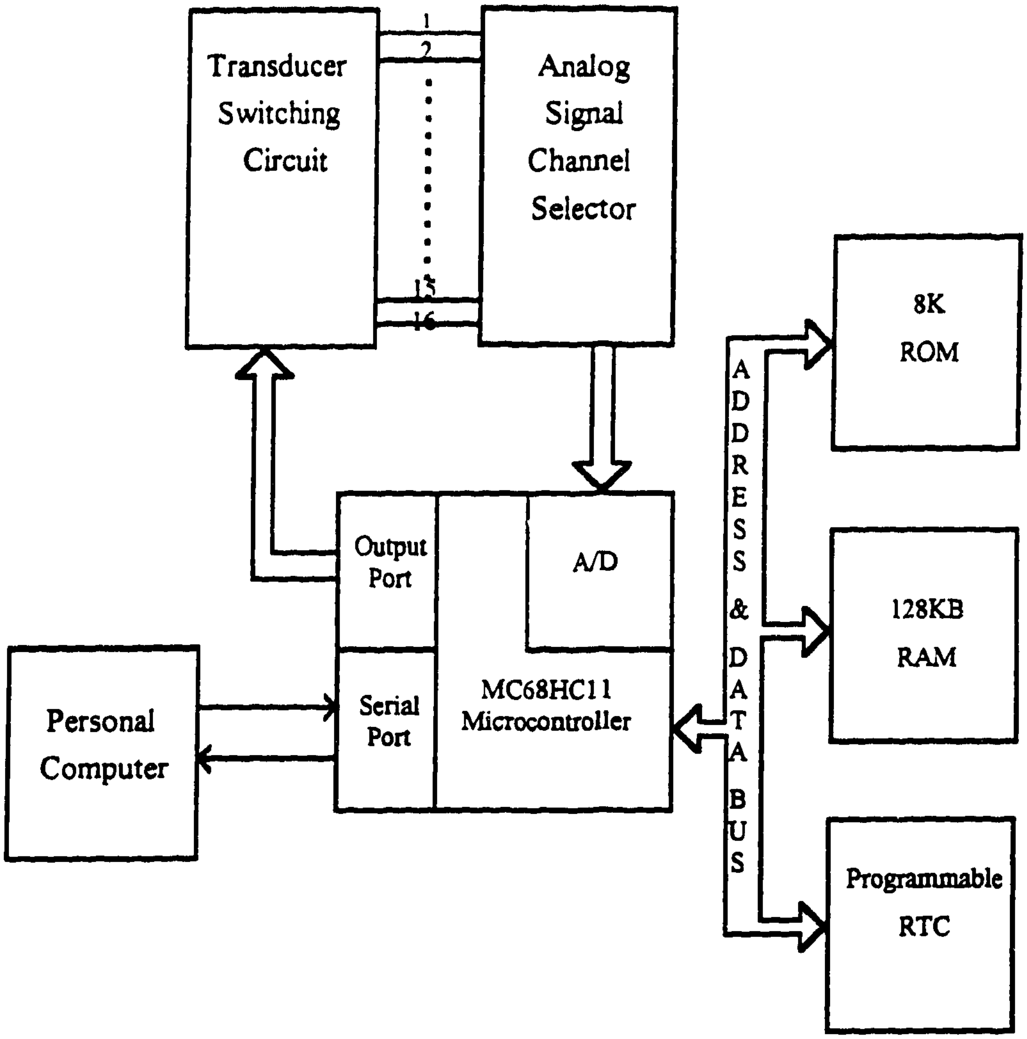
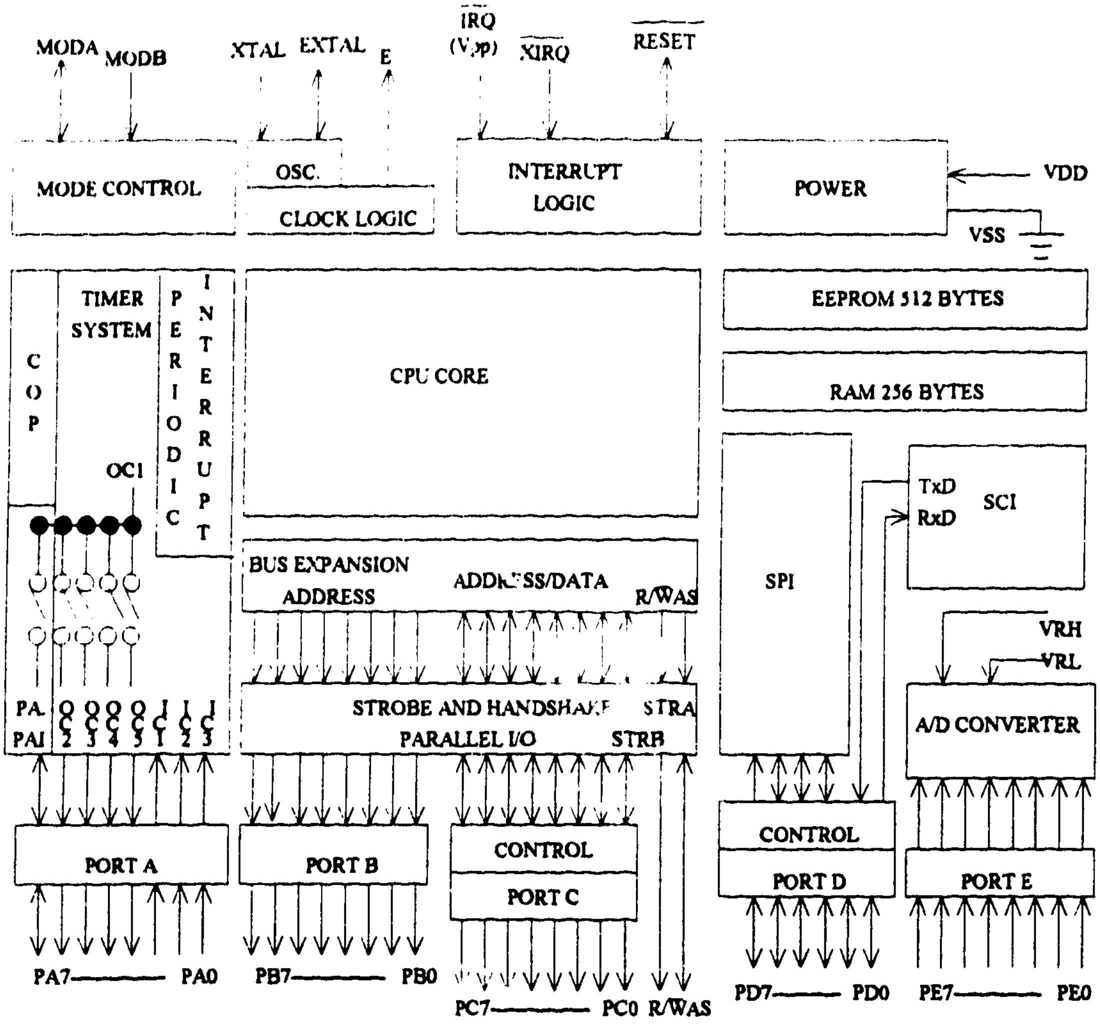


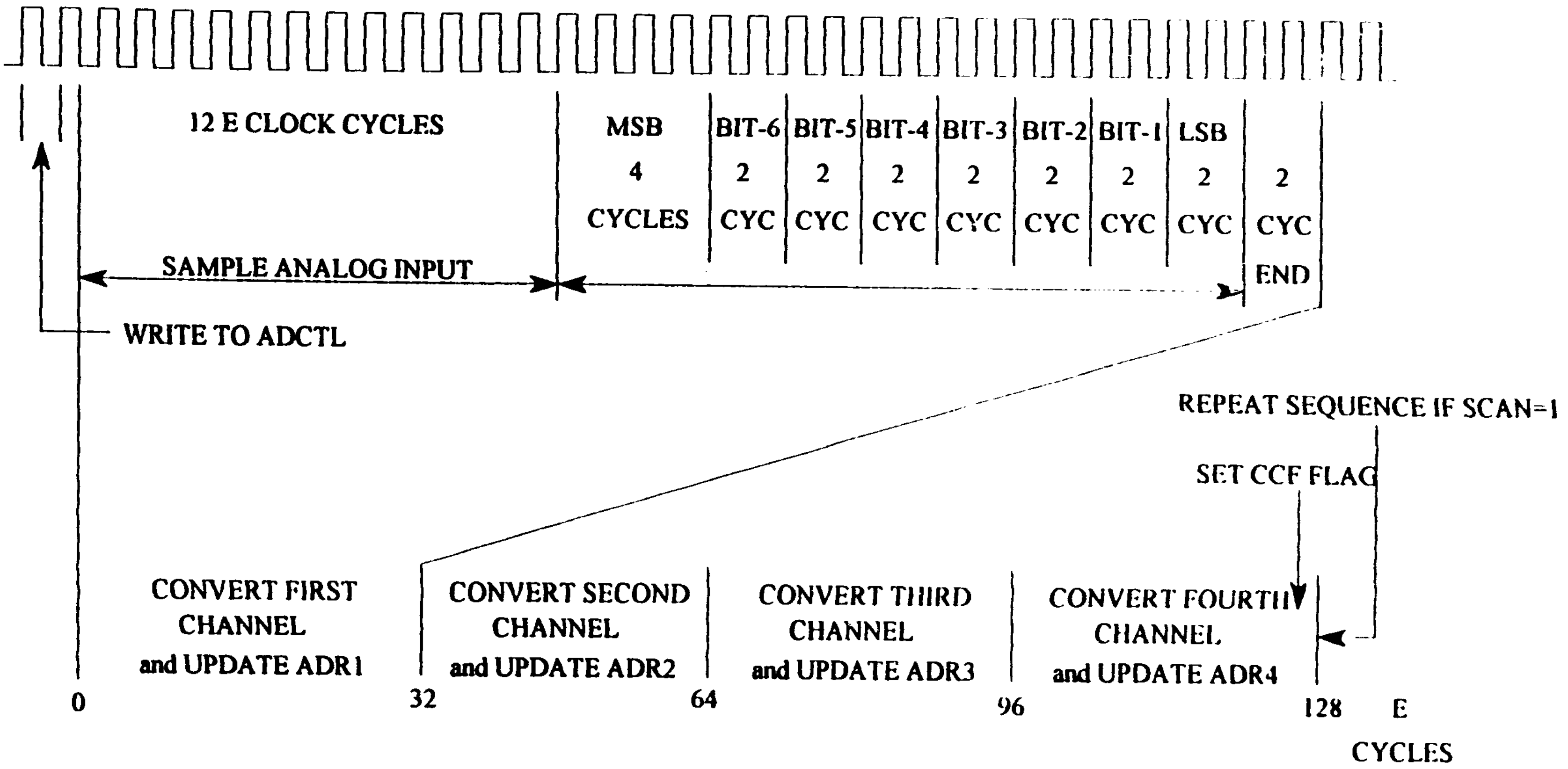
Figure 5-1 Block Diagram of the Microcomputer System

block diagram of the MCU is shown in Figure 5-2. The major on-chip peripheral functions are : an eight-channel A/D converter, an eight-bit parallel output port and a serial port. The A/D converter is used in conjunction with three analog switches to sample signals from one to sixteen force transducers. The A/D system on the MC68HC11A1 consists of a single successive-approximation A/D converter, an input multiplexer to select one of 16 channels (including eight channels associated with pins on the MCU), and sophisticated control circuitry to configure and control conversion activities. The A/D system has ± 1/2 LSB accuracy over the complete operating temperature range. Four separate result registers (ADR1 - ADR4) are included with control logic that implements automatic conversion sequences on a selected channel four times or on four channels (once each). Conversion sequences are configured to stop after one set of four conversions. Figure 5-3 shows the detailed sequence for a set of four conversions. To design a small system, all the control lines of the MCU must be used effectively. Eight lines from the parallel output port are used : four (PD3 - PD6) to control the power for the transducer circuitry, two (PA5 & PA6) as address lines to increase the address bus, one (PA4) for switching the RS232-driver-receiver on and off to save the power, and one (PA3) to select the active channels for the A/D conversions. Two lines (PD0 & PD1) from the serial port are used to communicate with any PC for programming the data acquisition system and downloading data acquired by the system. The system uses an 8 MHz crystal to generate a 2 MHz E clock. Also, the MCU operates in the expanded multiplexed operating mode to provide the capability of accessing a 64 kilobyte address space. The memory map is shown in Figure 5-4. This total address space includes the same on-chip memory addresses used for single-chip operating mode plus external peripheral and memory devices. The expansion bus is made up of port B and port C, AS and R/W* line.



PA = PULSE ACCUMULATOR

Figure 5-2 Block Diagram of the MCU



NOTES: Conversion results are builtup in the SAR and transferred into ADRx during the END period. The CCF status flag is set during the END period of the fourth conversion after a write to ADCTL. This figure assumes CSEL in the OPTION register is 0 so the E clock is acting as the conversion clock. If MULT = 0 all four conversions in the sequence are performed on the same analog channel.

Figure 5-3 A/D Conversion Sequences

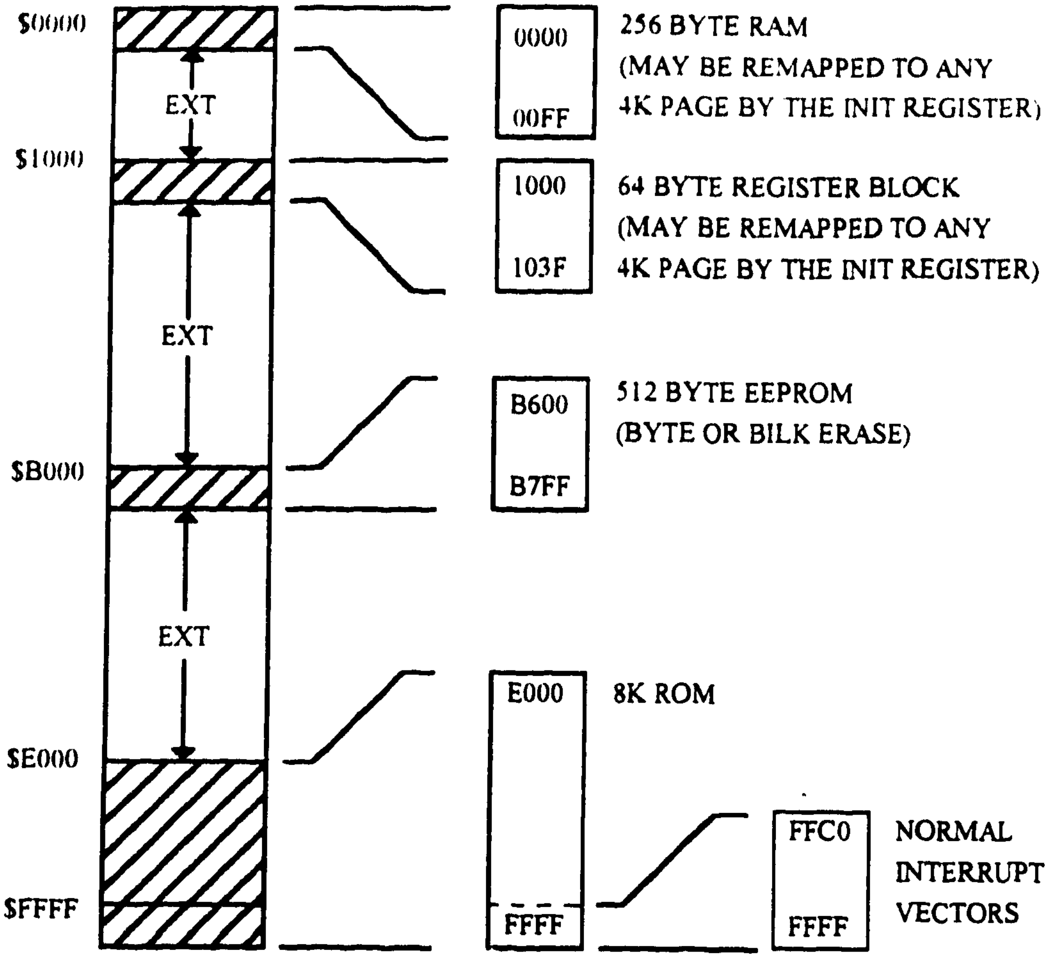


Figure 5-5 Memory Map

There are three other major components connected to the microcontroller system an 8 kilobyte CMOS ROM, a 128 kilobyte static RAM and a programmable RTC plus RAM. The logic block diagrams of the connection between these three peripheral devices and the MCU is shown in Figure 5-5, Figure 5-6 and Figure 5-7 respectively. Also, the system memory map diagram is shown in Figure 5-8.

The ROM (27C64) is a 64K-bit, ultraviolet erasable programmable read only memory. It is also a low power CMOS IC. It holds the control program for the system while the static RAM is used to store the acquired data. The program in the ROM can be erased by exposure to an ultraviolet light source. A dosage of 15 W seconds/cm^2 is required to completely erase the program in the ROM.

The static RAM (KM681000LP) is fabricated using Samsung's advanced CMOS process. It has been designed for high speed and low power applications, and is particularly well suited for battery back-up non-volatile memory application. Since the microcontroller has only 32 kilobyte left for external memory, two more address lines are required to expand the memory to 128 kilobyte. In the system, the static RAM is divided into four 32 kilobyte sections. The address buses (A0 - A14) of the static RAM are connected to the corresponding address buses of the MCU (A0 - A14). But the address buses A15 and A16 of the static RAM are connected to the two parallel output lines PA5 and PA6 respectively.

The programmable RTC with a built in static RAM (MC146818P) is a Motorola product. This device includes the unique MOTEL concept for use with both Motorola and Intel microprocessor timing cycles. However, this IC can only interface with 1MHz

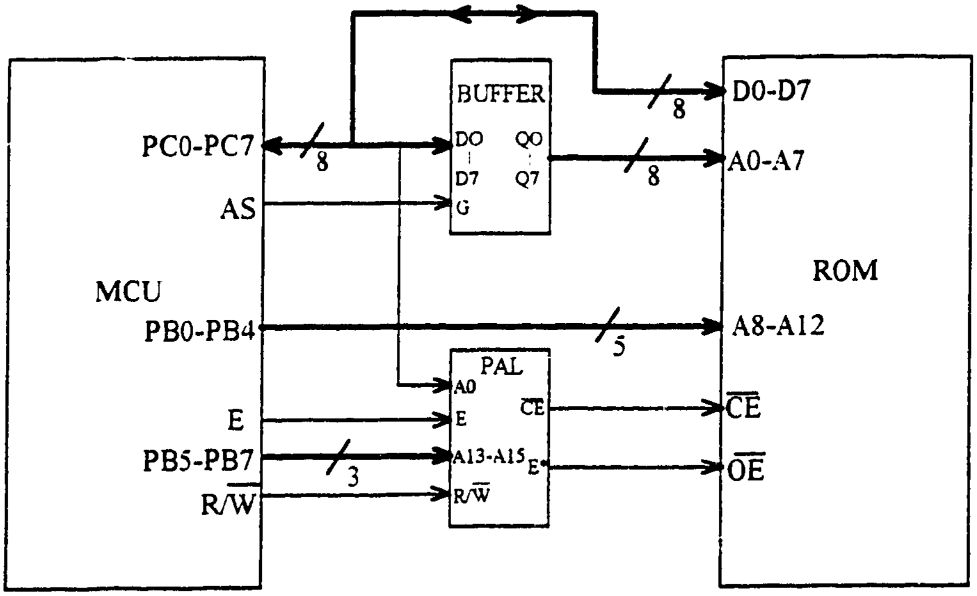


Figure 5-5 Block Diagram of the Connection between ROM and MCU

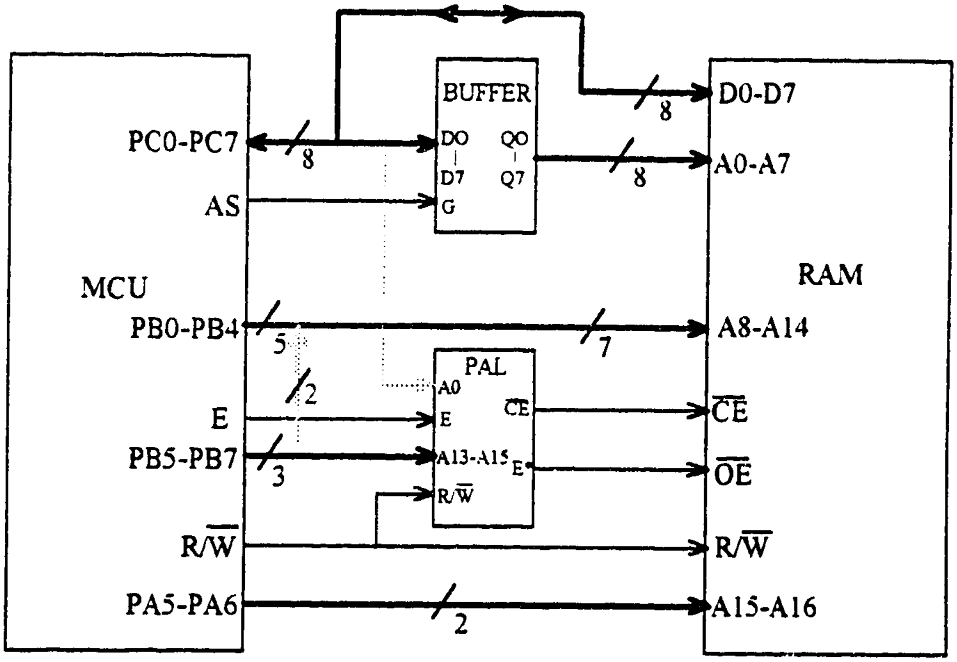


Figure 5-6 Block Diagram of the Connection between RAM and MCU

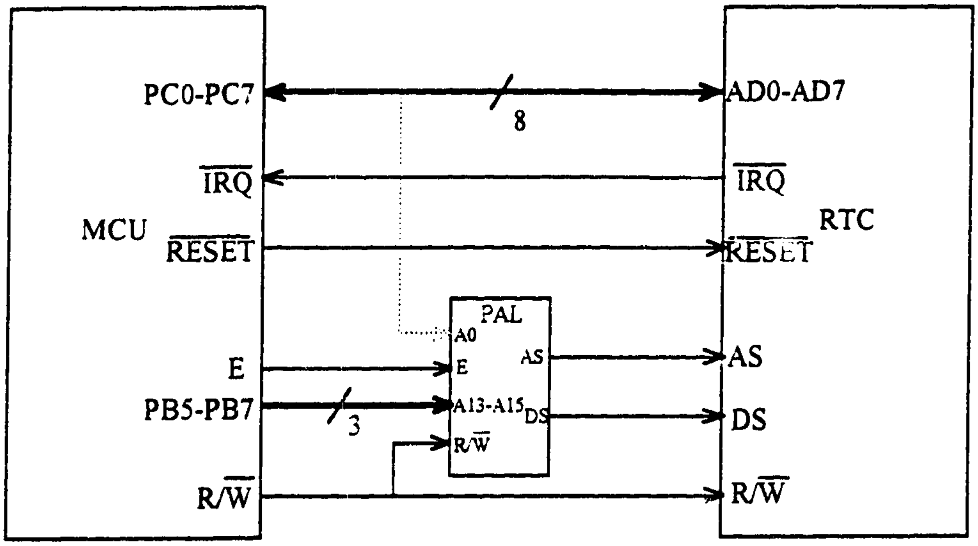


Figure 5-7 Block Diagram of the Connection between RTC and MCU

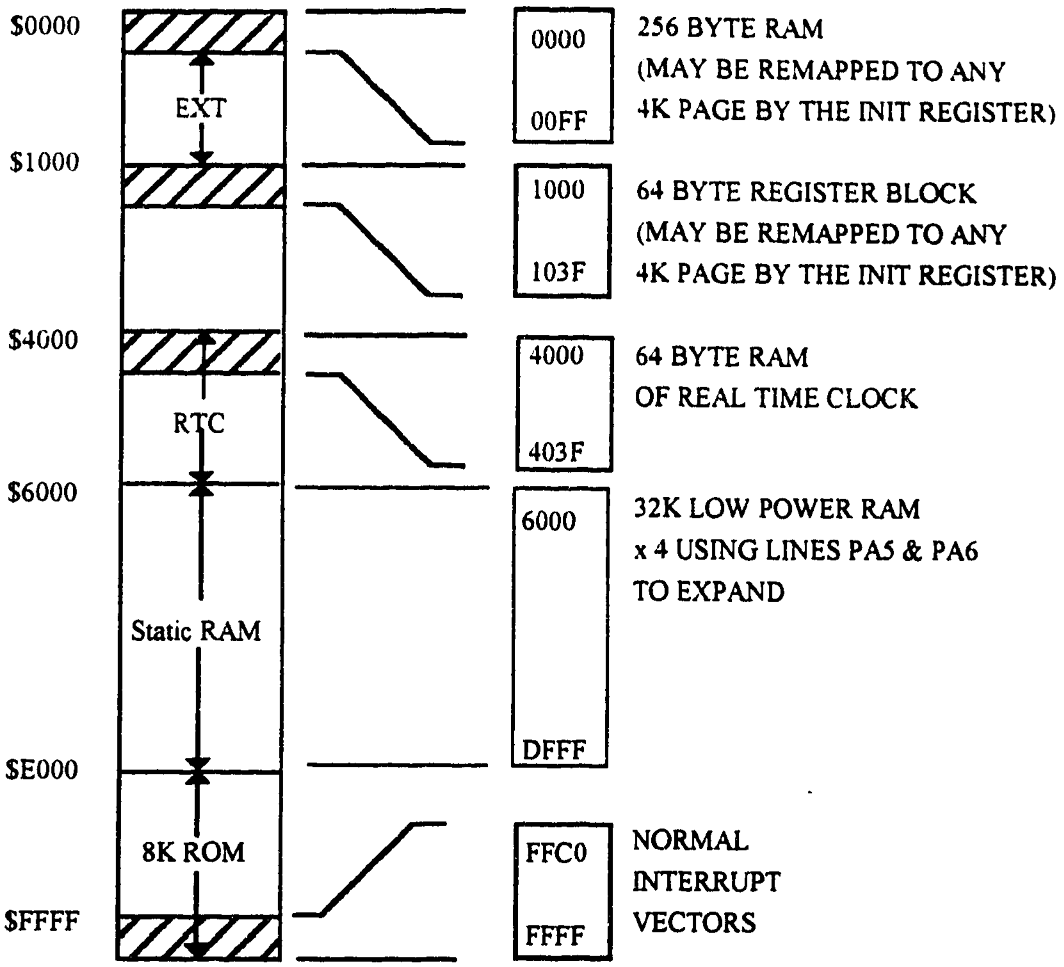


Figure 5-9 Memory of the Digital System

processor buses; therefore time delay logic is required. It combines four important features: (1) a complete time-of-day clock with alarm and a one hundred year calendar, (2) a programmable interrupt for alarm and timing functions, (3) a square wave generation circuit, and (4) 50 bytes of ultra low power static RAM. The address map of this IC is shown in Figure 5-9. The time information is stored in the built-in calendar and is updated after every sample. The four control registers A, B, C and D are used to initialize the RTC. The IC is also used to control the sample time and interval, and to provide an interrupt to the MCU. After the interrupt awakens the MCU from the STOP mode, the A/D of the MCU will convert the sample signals into digital values and store them in the 128 kilobyte static RAM. The MCU reverts to STOP mode again until the next interrupt. Also, the 50 bytes of the static RAM in this device are used to backup the data acquisition parameters such as the sample interval, the study period, the start time, the start date and the number of channels to sample before the system enters the STOP operating mode. The backup function saves the updated data in the static RAM of the RTC after each sample so that no data will be lost in event of a sudden power interruption.

Five other ICs are connected to the MCU : a PAL, a RS232 dual driver/receiver and three analog multiplexers /demultiplexers. The logic block diagram of the connections between these three chips and the MCU is shown in Figure 5-10, Figure 5-11 and Figure 5-12 respectively.

The PAL is used to replace some logic gates to reduce the system size. Five output lines such as AS, DS, E*, ROMselect and RAMselect are connected to the peripheral devices. The logic equations for AS, DS, ROMselect and RAMselect are

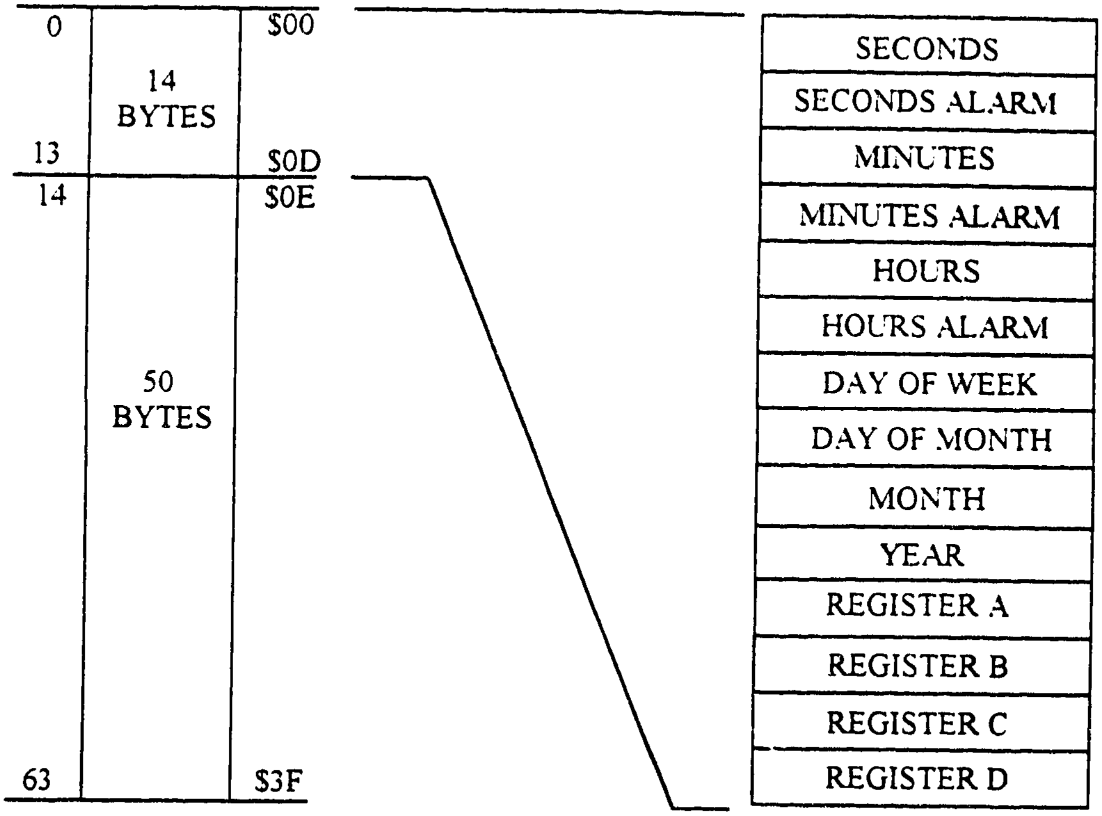


Figure 5-9 Address Map of MC146818P

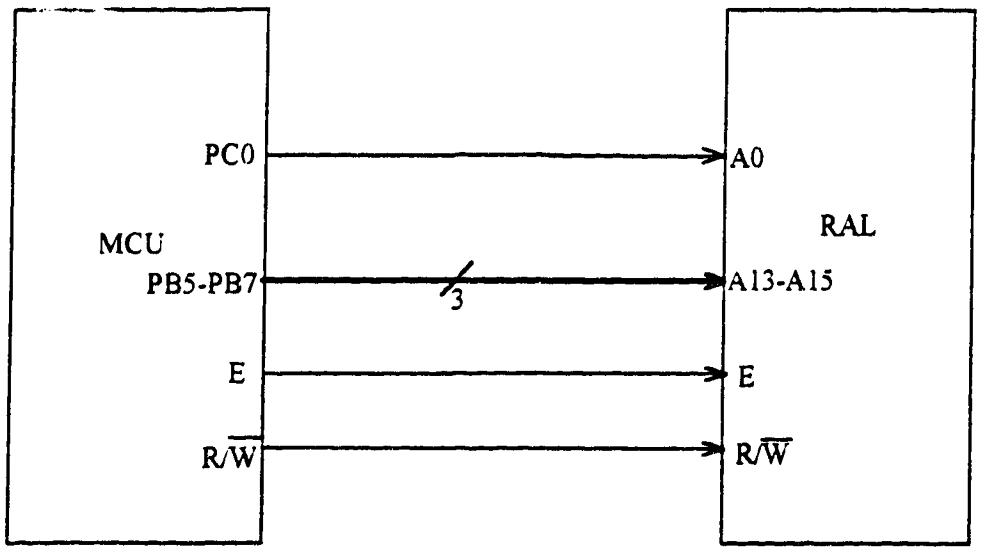


Figure 5-10 Block Diagram of the Connection between PAL and MCU

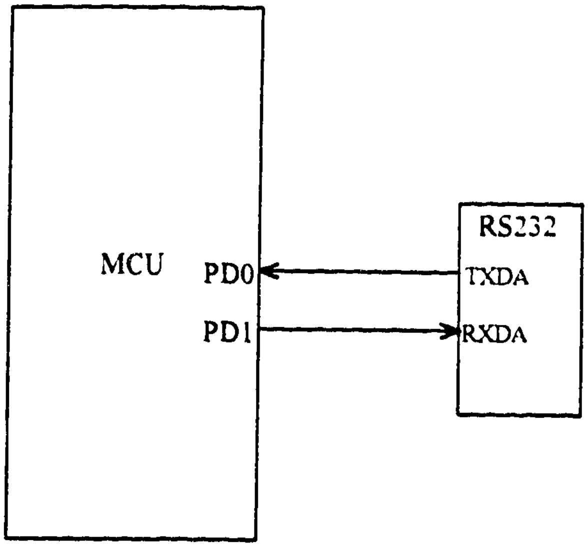


Figure 5-11 Block Diagram of the Connection between RS232 and MCU

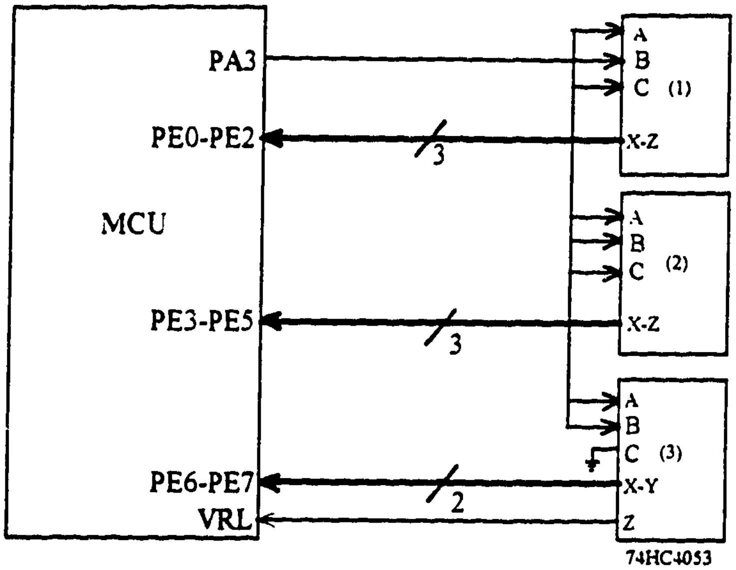


Figure 5-12 Block Diagram of the Connection between Mux/DeMux and MCU

shown in equations [1] to [4] respectively in Figure 5-13. The internal connections are shown in Figure 5-14.

The RS232 dual driver/receiver (LT1080CN) interface provides a realistic balance between CMOS levels of power dissipation and real world requirements for ruggedness. The driver outputs are fully protected against overload and can be shorted to ± 30V. The on-off pin (pin 18) of this IC is connected to the switching circuitry which is controlled by PA4, one of an eight-bit control bus from the parallel output port. A logic high on PA4 puts the device in SHUTDOWN mode, which reduces input supply current to zero and places both driver and receiver outputs into a high impedance state.

The analog multiplexers/demultiplexers (74HC4053) use silicon-gate CMOS technology to achieve fast propagation delays, low ON resistances, and low OFF leakage currents. This device is between the A/D converter of the MCU and the analog circuitry of the transducer. The channel-select input pins (A, B and C) determine which one of the analog inputs/outputs are to be enabled. The function table of this IC is shown in Figure 5-15. Since the channel-select pins are only connected to PA3, there can be only two cases for the channels selected. Either X0, Y0 and Z0 or X1, Y1 and Z1 will be selected.

There are three analog circuits in the digital system (Figure B-3) : (1) the backup power circuitry for the RAM and RTC, (2) the power on-off for the RS232 and (3) the power control circuitry for the PAL.

The backup power circuitry has two diodes and a three volt button cell. The VIN is connected directly to the power supply of the RTC and the static RAM. When the

$$A0' \cdot R'/W \cdot E = AS \quad [1]$$

$$A13' \cdot A14 \cdot A15' \cdot A0 \cdot E = DS \quad [2]$$

$$(A13 \cdot A14 \cdot A15 \cdot E)' = ROMselect \quad [3]$$

$$((E \cdot A13 \cdot A14 \cdot A15') + (E \cdot A13' \cdot A14' \cdot A15) + (E \cdot A13 \cdot A14' \cdot A15) + (E \cdot A13' \cdot A14 \cdot A15))' = RAMselect \quad [4]$$

note: '·' = AND operation
'+' = OR operation

Figure 5-13 The Logic Equations of the Output Lines

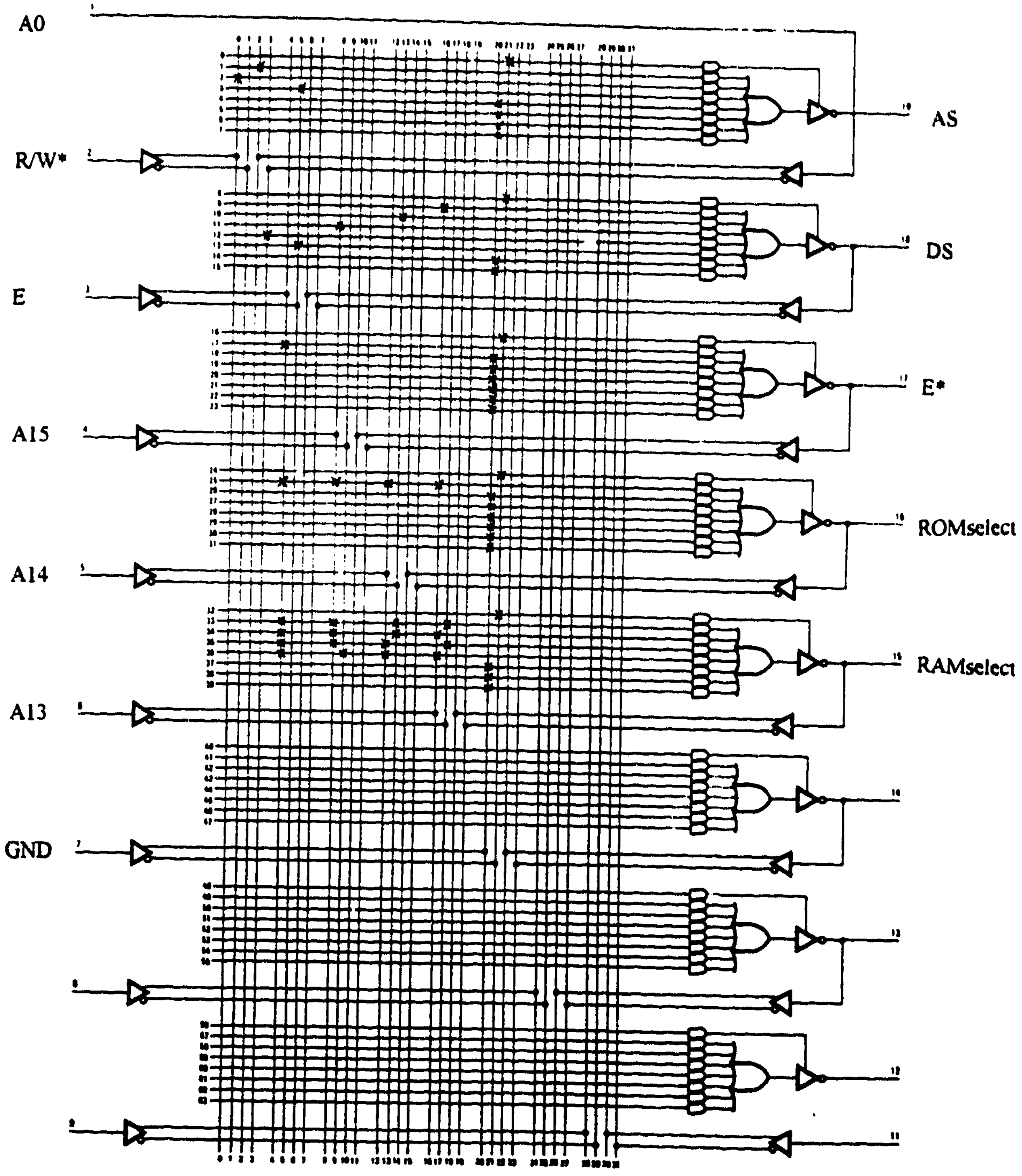


Figure 5-14 Internal Connections of PAL

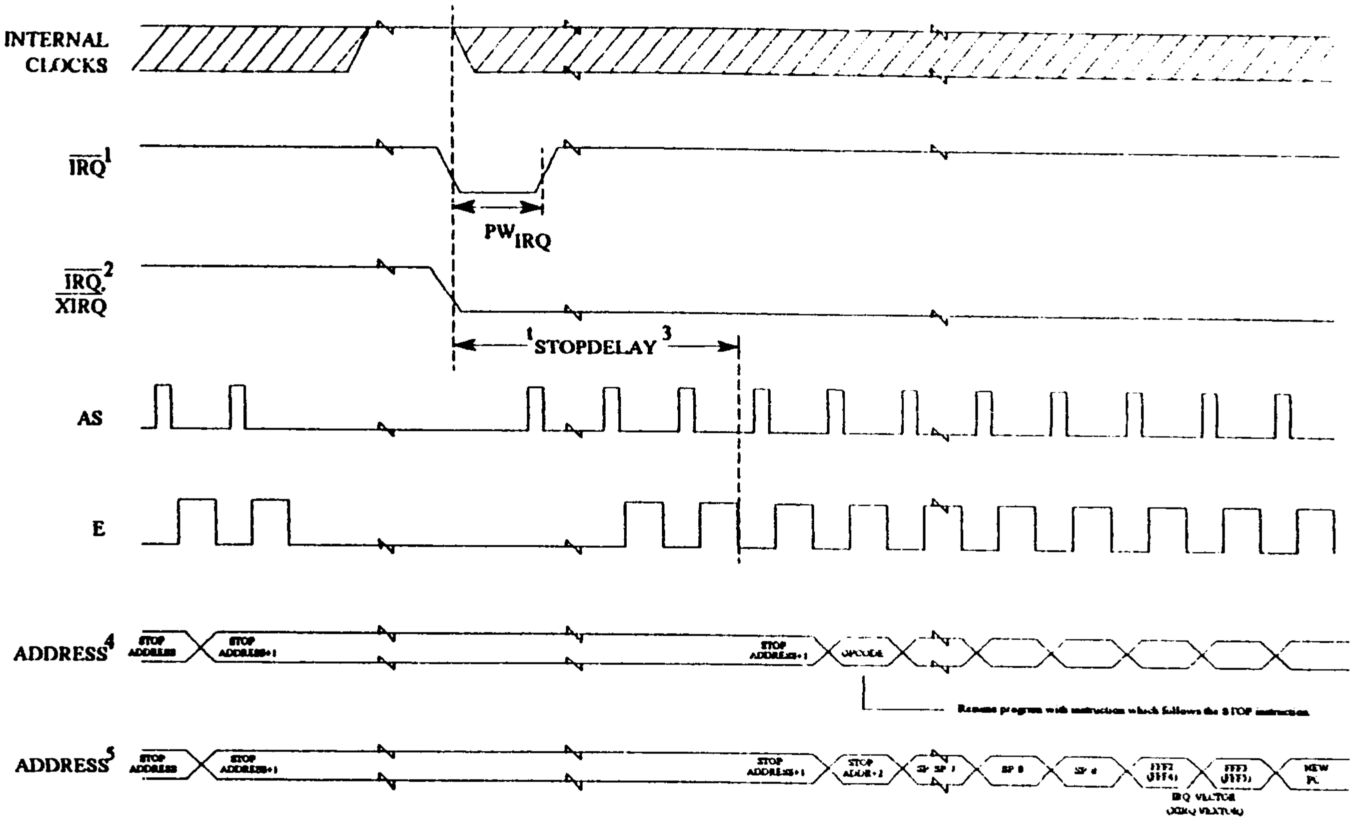
Control Inputs				On Channels		
Enable	Select					
	C	B	A			
L	L	L	L	Z0	Y0	X0
L	L	L	H	Z0	Y0	X1
L	L	H	L	Z0	Y1	X0
L	L	H	H	Z0	Y1	X1
L	H	L	L	Z1	Y0	X0
L	H	L	H	Z1	Y0	X1
L	H	H	L	Z1	Y1	X0
L	H	H	H	Z1	Y1	X1
H	X	X	X	NONE		

X = Don't Care

Figure 5-15 Function Table of 74HC4503

power is shut down, the backup power will maintains memories in both static RAM and RTC. The power on-off circuitry for the RS232 consists of three resistors and a PNP transistor. When the logic of PA4 is 0, the power for the RS232 is enabled; otherwise, it is disabled. Therefore, power can be saved by turning off the power supply for the RS232 when no communication with a PC is required. To save more power, the power control circuitry for the PAL requires that the E clock is used as the input of this circuit. When the MCU awakens from the STOP mode, a 4064 clock cycle times (i.e. 0.002 sec) delay is required to allow stabilization of the oscillator. The delay time allows the A/D circuitry to stabilize and avoids invalid results. Therefore, the E clock can be used as an input, and the PAL is turned on and stabilized before the samples are acquired. The timing diagram of the MCU after it reverts from the STOP mode is shown in Figure 5-16. Tables 5-1 shows the pin state summary for RESET and STOP.

Since the power consumption is one of the major concerns of the system design, measurement of the current consumption for different operating modes has been done. The experimental and theoretical results for current consumption are shown in the section 5.1.2.



Notes:

1. Edge sensitive $\overline{IRQ}$ pin (IRQE bit =1)
2. Edge sensitive $\overline{IRQ}$ pin (IRQE bit =0)
3. $t_{STOPDELAY}$=4064 tcyc if DLY bit=1 or tcyc if DLY=0.
4. $\overline{XIRQ}$ with X bit in CCR=1.
5. $\overline{IRQ}$ or ($\overline{XIRQ}$ with X bit in CCR=0).
6. Refer to Table 5-1 for pin states during STOP.

Figure 5-16 STOP Recovery Timing Diagram

PINS	EXPANDED MODES	
	RESET	STOP
OUTPUT ONLY		
E	ACTIVE E	0
XTAL!!!	ACTIVE	1
R/W*	1	1
PA3-PA6	0	SS
PB0-PB7	HI ADD	HI ADD
INPUT/OUTPUT		
RESET	I (0)	I
MODA/LIR	I (1)	OD (1)
MODB/V_{STBY}	I (MODES)	I (V_{STBY})
AS	ACTIVE AS	0
PA7	I	I/O
PC0-PC7	ADD/DATA	I
PD0-PD5	I	I/O
INPUT ONLY		
EXTAL	Input CLock or Connect to Crystal with XTAL	
IRQ*	Terminate Unused Inputs to V_{DD}	
XIRQ*	Terminate Unused Inputs to V_{DD}	
PA0-PA2	Terminate Unused Inputs to V_{DD} or V_{SS}	
PE0-PE7	If Not Used, External Drive Not Required	
V_{RH}-V_{RL}	If Not Used, External Drive Not Required	

SYMBOLS:

DATA = Current data present

I = Input pin, if ( ) associated then this is required input state.

I/O = Input/Output pin, state determineed by data direction register.

HI ADD = High byte of the address.

ADD/DATA = Low byte of the address multiplexed with data

OD = Open drain output, ( ) current output state.

SS = Steady state, output pin stays in current state.

!!! = XTAL is output but not normally usable for any output function beyond crystal drive.

Table 5-1 Pin State Summary for RESET and STOP

5.1.1.1 Current Consumption

Experimental Results

Range of the total current consumed = 82.75 - 84.45 mA

In Stop Mode

Range of current consumed = 0.60 - 0.80 mA

Range of the total Amp-hour consumed in two weeks = (0.6 - 0.8)mA * 24 * 14

= 201.6 - 268.8 mAh

In Sampling Mode

Range of current consumed for taking 16 channels = 59.8 - 60.8 mA

The time for taking 16 sample signals = 5.08 msec

Total no. of samples for each channel = 128 * 1024 /16

= 8192

Range of the total Amp-hour consumed = (59.8 - 60.8)* 5.08 msec * 8192

= (2488.6 - 2530.2) mAsec

= 0.6912 - 0.7028mAh

∴ range of the total Amp-hour requirement of the system = (202.3 - 269.5) mAh

Also, the maximum sampling frequency for 16 channels = 8192 / (14 * 24)

= 24 sample/hr

= 1 sample/2.5 min

If only 1 channel is chosen, the sampling frequency = 128 * 1024 / (24 * 14 * 60)

= 1 sample/6.5 sec

For a safety factor of 50 %, the maximum capacity of the power = 269.5mAh * 1.5

= 404.25 mAh

Therefore, the power capacity of the battery should be more than 400mAh.

Theoretical Results

Device	Operating	Stop Mode	Sampling Mode
68HC11A1FN	27mA	100μA	27mA
74HC373	1μA	1μA	1μA
27C64	25mA	100μA	25mA
KM681000LP	20mA	100μA	20mA
PAL16L8ACN	30mA	0	30mA
LT1080CN	22mA	0	0
MC146818P	50μA	50μA	50μA
74HC4053 x 3	6μA	6μA	6μA
OP-AMP x 8	0	0	4mA

Table 5-2 Current Consumption for each IC

When the system starts, the theoretical maximum current consumed = 124 mA

In Stop Mode

The theoretical current consumed = 357 μA

Total Amp-hour consumed in two weeks = 357 μA * 24 * 14 =120.0 mAh

In Sampling Mode

Maximum current consumed for taking 16 channels = 106.0 mA

Total Amp-hour consumed on sampling mode = 106.0 m * 5.08 m * 8192

= 4411.2 mAsec

= 1.2 mAh

Therefore, total theoretical Amp-hour requirements of the system = 120.0 +1.2 mAh

= 121.2 mAh

Range of the % Error between the theoretical and experimental result on current consumption is 66.9 - 122.3%.

In addition, battery selection has been considered carefully. Two kinds of battery have been considered, the Nickel-Cadmium rechargeable battery, and the sealed Lead-Acid rechargeable battery. Assume four AA size NiCad batteries are used. A comparison between these two batteries is given in Table 5-3.

The NiCad battery should be used instead of the Lead-Acid battery because it has more flexible dimensions and more capacity. Also, the AA size NiCad battery is more common.

	NiCad	Lead-Acid
1. Weight (grams)	200	200
2. Dimensions (in^3)	a) 4.375 x 0.625 x 1.0 b) 0.5 x 2.0 x 2.25	2.24 x 0.55 x 1.97
3. Capacity (mAh)	600	500
4. Temperature range	-20 °C to +50 °C	-60 °C to +60 °C
5. Service life charge/recharge (cycles)	500 to 1000	200 to 1000

Table 5-3 A comparison between Nicad and Lead-Acid rechargeable batteries

5.1.2 The Software Design

The MC68HC11 is able to execute all MC6800 and MC6801 instructions and allows execution of 91 new opcodes. The major functional additions include a second 16-bit index register (Y register), two types of 16-by-16 divide instructions, STOP and WAIT instructions, and bit manipulations. Figure 5-17 shows the seven CPU registers available to the programmer. The control program, written in MC68HC11 assembly language, oversees the operation of the device under interrupt control. In STOP mode, the microcontroller, ROM, RAM and RTC are in low power consumption mode while the RS232-driver-receiver, PAL, and analog circuitry consume no power. When samples are required, the programmable timer provides an interrupt to awaken the MCU. The program permits communication with any PC so that acquisition parameters may be accepted or acquired data downloaded into a file for a subsequent analysis. An user can choose to either input the parameters or output the data. If input mode is selected, the user initializes the following selections:

1) the sample interval, from 1 second to 1 day
2) the study period, from 1 day to 14 days
3) the start time
4) the start date
5) the number of channels to sample from 1 to 16

Each of these inputs should be followed by a carriage return (CR) to signify the end of input for that variable. Any software which can communicate with the RS232-serial port can be used to download the acquired data into a file for subsequent analysis. The start date and time, and the stop date and time are also be downloaded into a file for subsequent analysis.

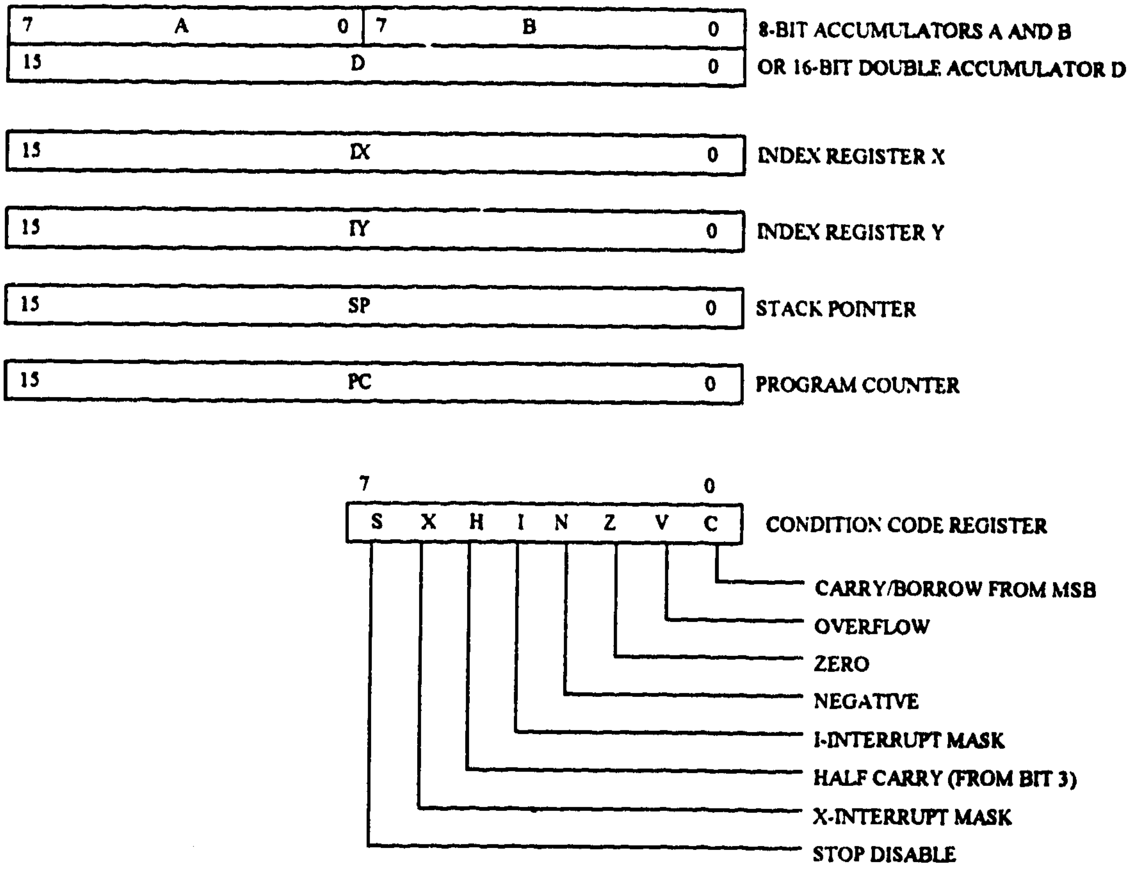


Figure 5-17 Seven CPU Registers

5.1.2.1 Program Description

After a hardware reset, the control program starts with the MAIN program. The flowchart of the MAIN routine is shown in Figure 5-18. After initialization of the stack pointer, the program calls the INIT subroutine (Figure 5-19) where the on-chip EEPROM is disabled, the microcontroller is initialized to the expanded multiplexed operating mode, the A/D converter is enabled, the baud rate is selected as 9600, both port A and D are configured as output control lines, and the transmitter and receiver are enabled. Also, some variables are reset in this stage. Then the user is allowed to choose either case 1, input acquisition parameters and take samples, by pressing the "I" key or case 2, display the acquired data and store them into a file, by pressing the "O" key. If another key is pressed, 'ERROR' is displayed and the program branches back to the choice instruction and waits for another key stroke. If no sample has been taken, even though an "O" key has been pressed, the program stays in the choice stage. After an appropriate key instruction has been received, the program invokes the requested module and executes the command. After finishing the requested module, the software returns to the MAIN routine and turns off the analog circuitry and the RS232 driver. The system remains in STOP mode awaiting an interrupt.

5.1.2.2 INPUT Module

The flowchart for the input module is shown in Figure 5-20. The routine is executed when an "I" key is pressed. The program jumps to the SET UP module to input parameters. After jumping back from the SET UP module, the power to the RS232 dual driver/receiver is turned off. The program executes the INITCLK module in which the

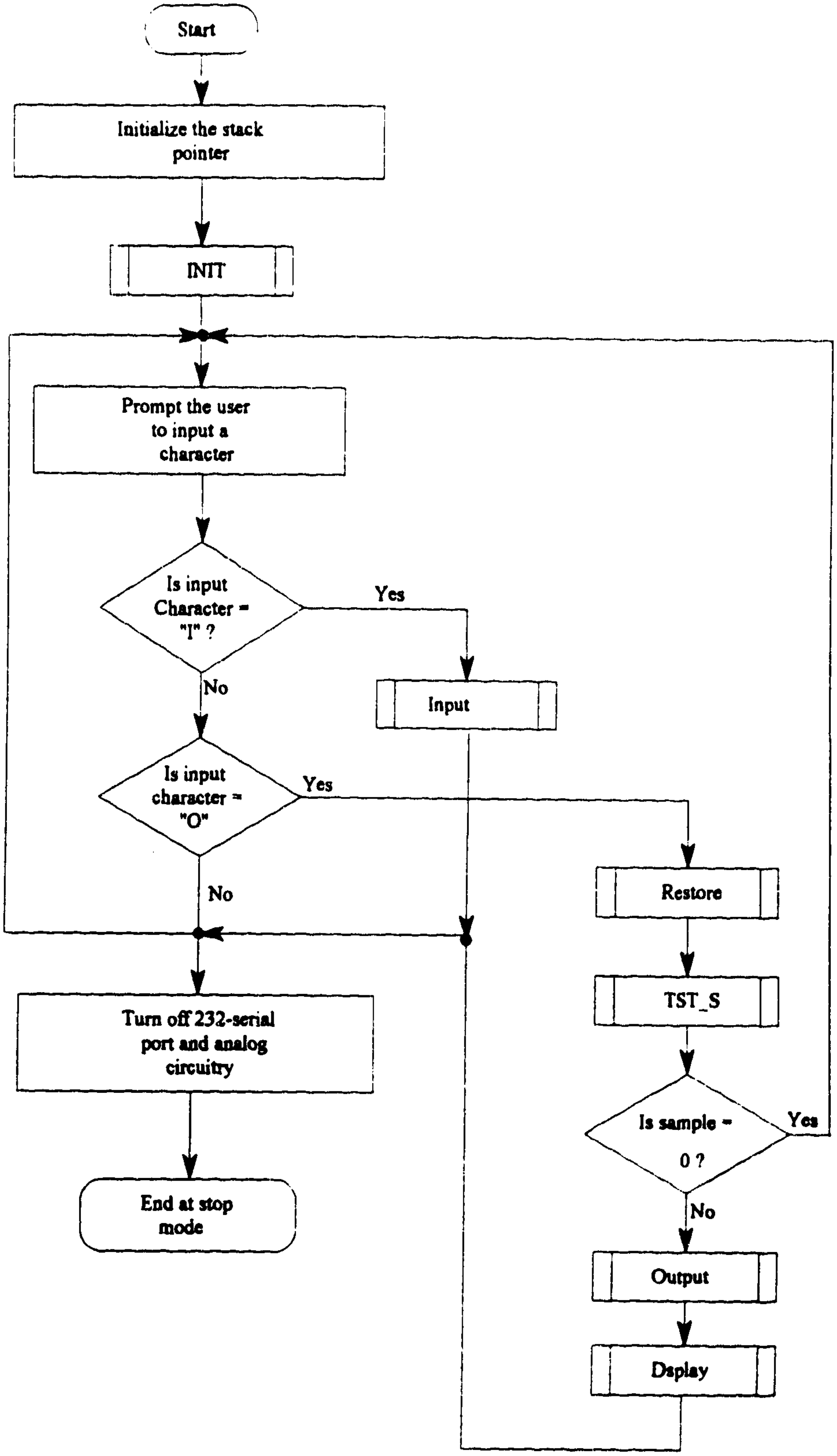


Figure 5-18 Flowchart of the Main routine

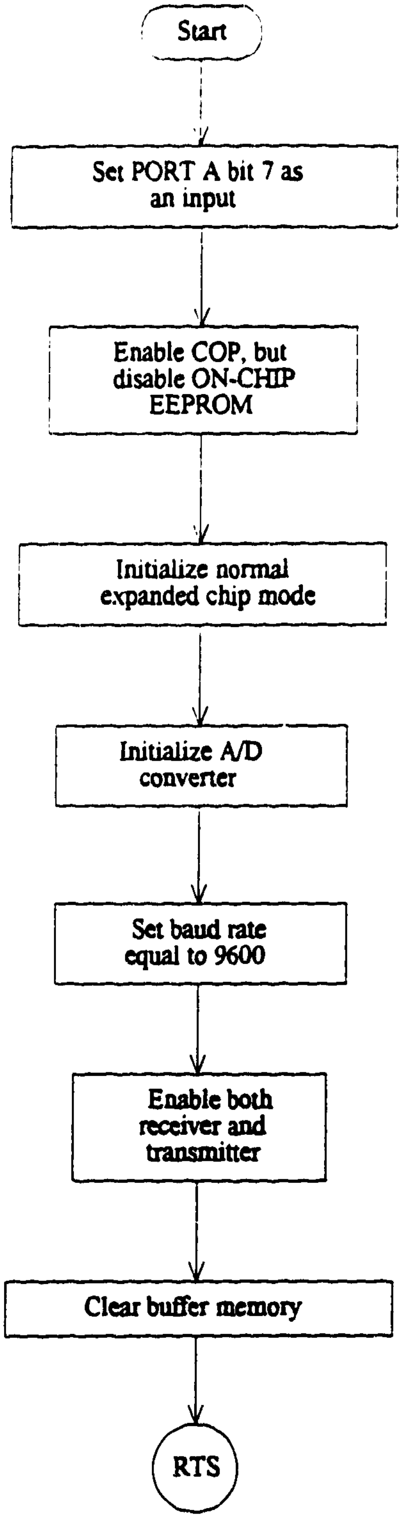


Figure 5-19 Flowchart of the INIT subroutine

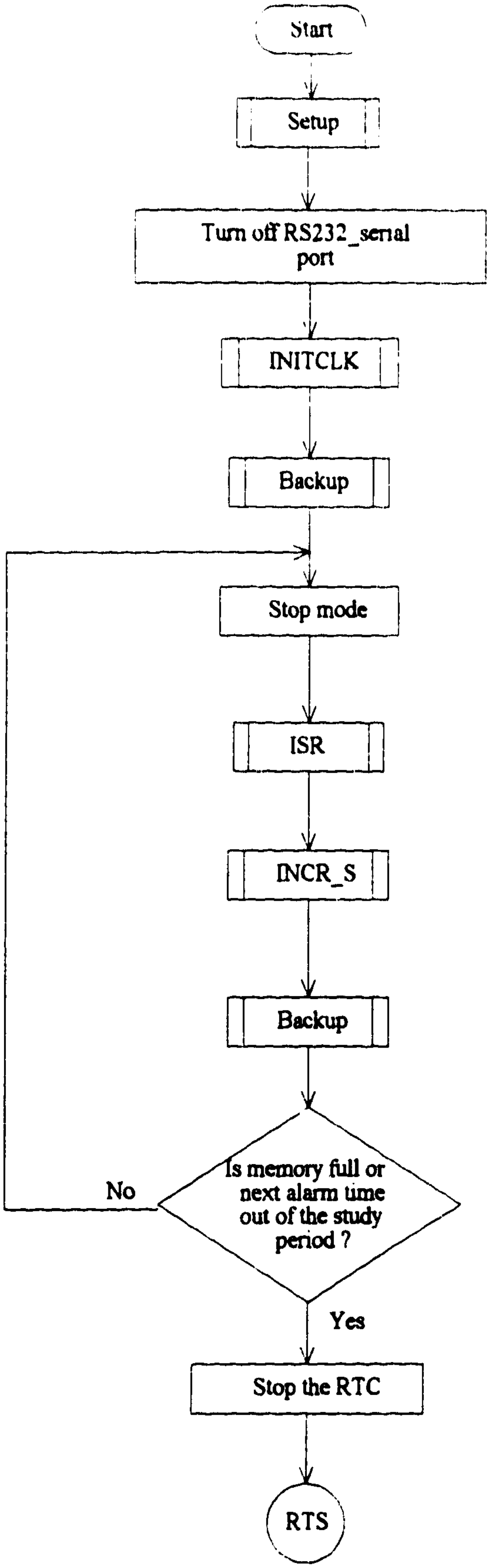


Figure 5-20 Flowchart of the INPUT module

programmable real time clock is initialized. The next alarm time will be calculated in the NXTALM module which is called under the INITCLK module. Then the clock starts running. The program returns to the INPUT module, then the BACKUP subroutine (Figure 5-21) is executed in which the first 46 bytes (containing all the input parameters) of the memory in the RAM of the microcontroller are backed up into the static RAM of the RTC. The system stays in STOP mode and waits for an interrupt. The ISR module is executed when the current time matches the alarm time. The interrupt awakens the microcontroller from STOP mode and samples are acquired. After collecting the sample data, the counter for the number of samples is increased by one. Then the BACKUP subroutine is executed again. The microcontroller sleeps and awakens repeatedly until either the memory in the static RAM is full or the next alarm time in the RTC is out of the study period range. The program then returns to MAIN routine.

5.1.2.3 SET UP Module

The flowchart of this module, in which the user inputs the acquisition parameters, sample interval, interval period, start time, start date and number of channels, is shown in Figure 5-22. These parameters are converted into decimal values and stored in the RAM of the microcontroller via the READ subroutine (Figure 5-23). Also, a flag is used to distinguish the different input : 0 for the sample interval, 1 for the study period, 2 for the start time, 3 for the start date, and 4 for the number of channels. If any input data is out of range or improper, an 'ERROR' message is displayed and the program prompts for another input. All input values are echoed to the monitor. After inputting all the required parameters, the program returns to the INPUT module.

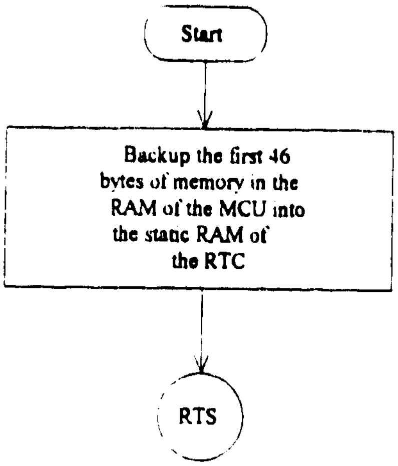


Figure 5-21 Flowchart of the BACKUP subroutine

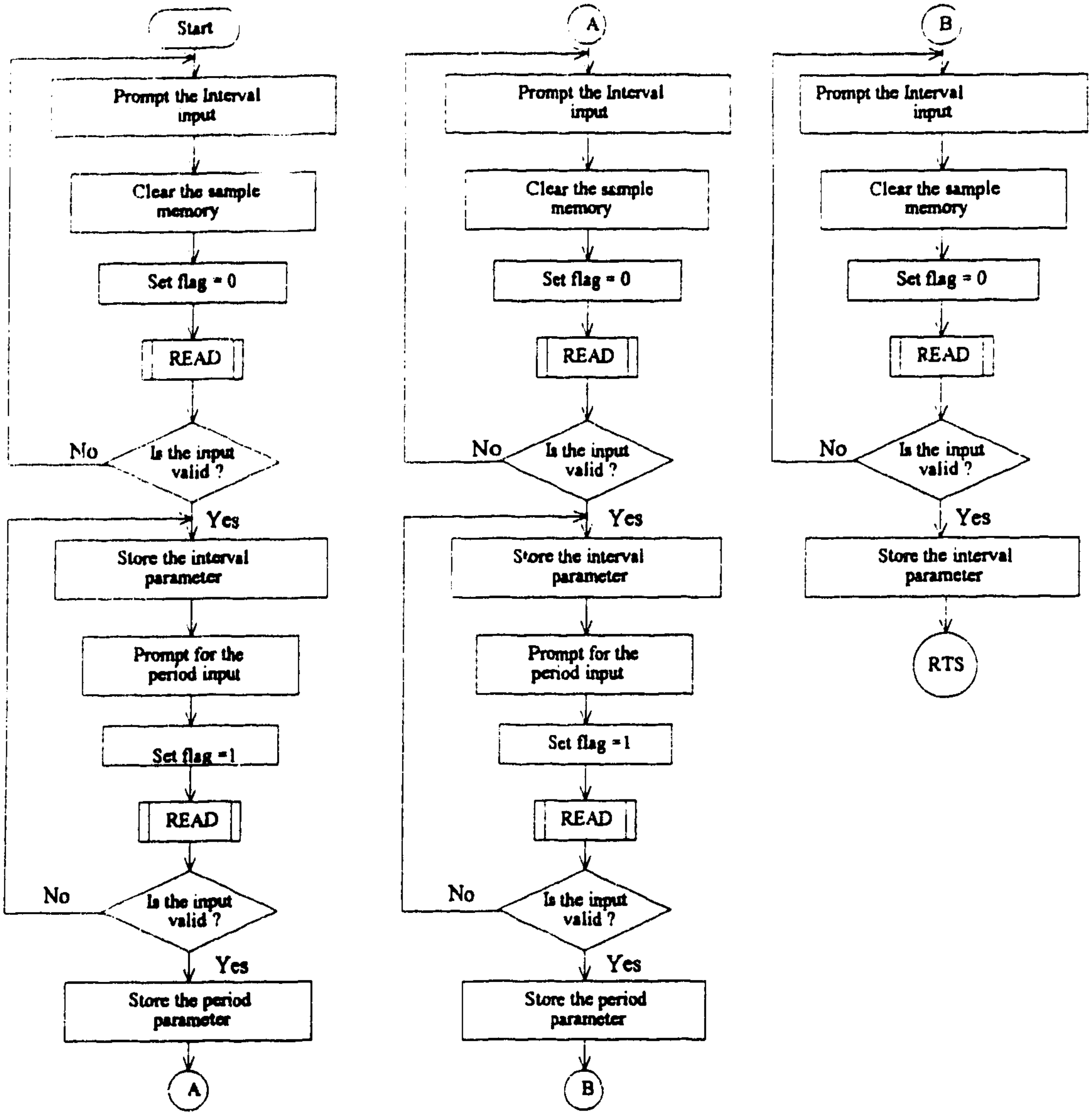


Figure 5-22 Flowchart of the SET UP module

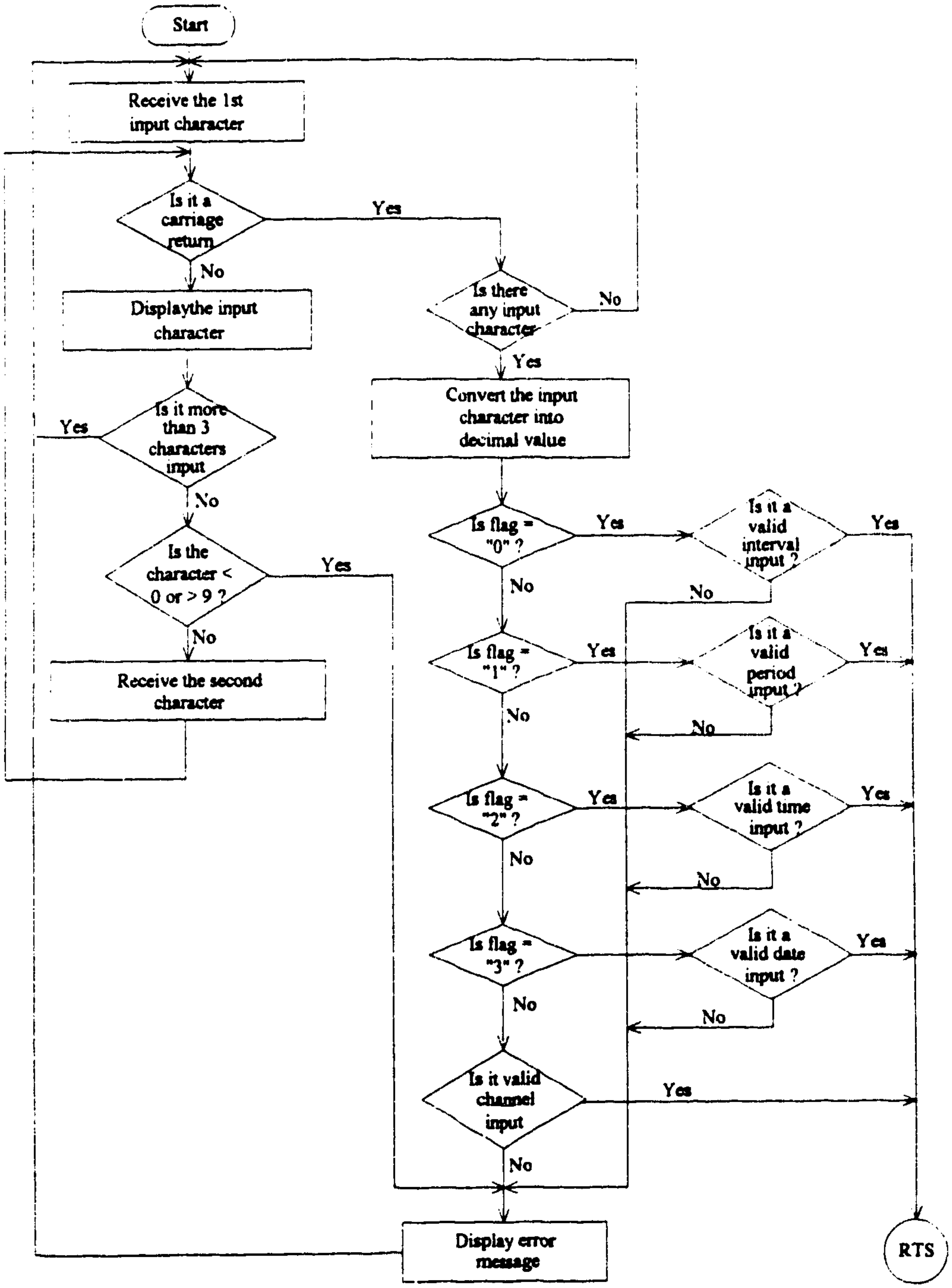


Figure 5-23 Flowchart of the READ subroutine

5.1.2.4 INITCLK Module

The flowchart for this module is shown in Figure 5-24. This routine starts by checking the power ready for the RTC. If the power for the RTC is less than 2.2 volts, this indicates invalid data in the RAM of this device. The program stays in this stage until the power supply is higher than 2.2 volts at which time the interrupt flags of the RTC are cleared. The time and calendar bytes of the RTC are then initialized, periodic interrupt, alarm interrupt, update-ended interrupt, and square wave generated are all disabled, binary data is used to update the time and calendar, the 24-hour mode format is set and the daylight savings time is enabled. The time base frequency of the RTC is then set to 32.768 kHz. The time and calendar information are initialized. The NXTALM module is called. Upon completion of the NXTALM module, the clock is turned on and the alarm interrupt is enabled.

5.1.2.5 NXTALM Module

The flowchart for this module is shown in Figure 5-25. The program starts by calculating the next alarm time of the RTC. The RTC checks the internal update cycle. The primary functions of the update cycle are to increment the second byte, check for overflow, increment the minutes byte when appropriate and so forth through the year of the century byte. The update cycle also compares each alarm byte with the corresponding time byte and issues an alarm if a match occurs. If the update cycle is in progress, no time, calendar, and alarm information are available to the program. The program will then wait until the update cycle finished. After finished the checking, the current time, calendar and

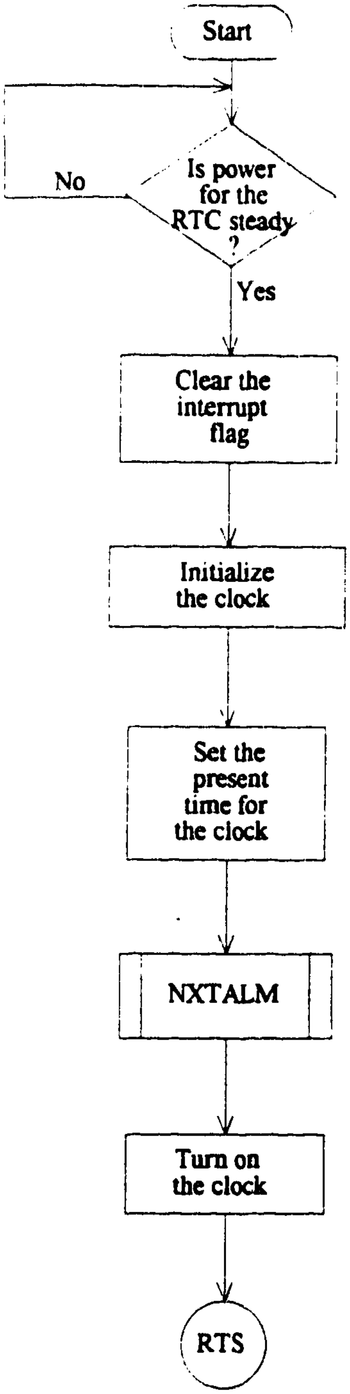


Figure 5-24 Flowchart of the INITCLK module

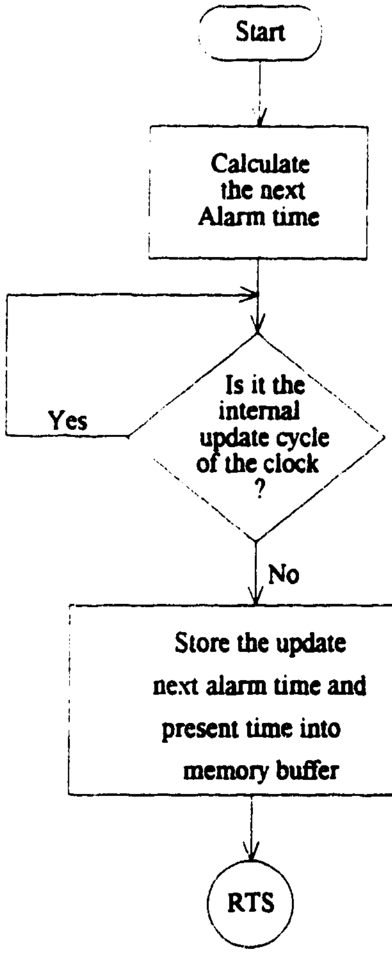


Figure 5-25 Flowchart of the NXTALM module

the alarm time are stored into the RAM of the microcontroller. The program returns to the calling module.

5.1.2.6 ISR Module

The flowchart for this module is shown in Figure 5-26. This routine is executed when the alarm time matches the current time, and the space of the static RAM is not full; otherwise, the program jumps back to the MAIN routine and the system reverts to STOP mode. When an interrupt occurs, the microcontroller is awakened and samples are acquired. The program jumps to the NXTALM module to calculate the next alarm time before taking samples. Then a time delay for the A/D power up is required, which is executed in the DELAY subroutine (Figure 5-27). ADCONV subroutine (Figure 5-28) is then executed and converted the received analog signal into the digital values and stored in the static RAM. Taking samples from 16 channels require each interrupt to last 5.08 msec. After storing the data, the program jumps to the INPUT module.

5.1.2.7 OUTPUT Module

The flowchart for this module is shown in Figure 5-29. This routine is executed when an "O" command key is pressed after the choice instruction in the MAIN routine. The start date and time, and the stop date and time are first displayed. Then DSPLAY subroutine (Figure 5-30) is executed and retrieves the sample data from the static RAM.. The data is converted into a four digit values in mV and displayed on the monitor. The number of samples equals the number of rows of data displayed. The data can be saved into a file by using any software which can communicate with the RS232 serial port. After

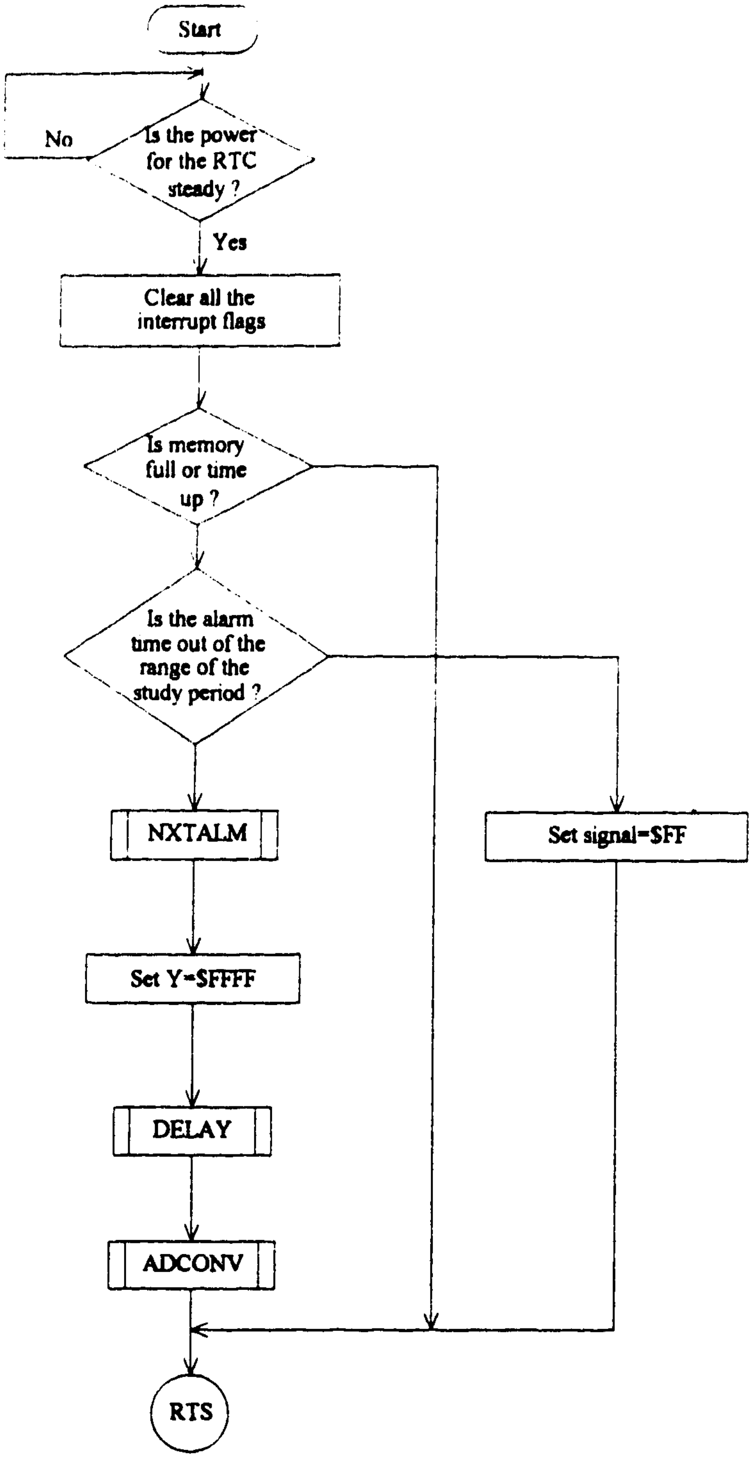


Figure 5-26 Flowchart of the ISR module

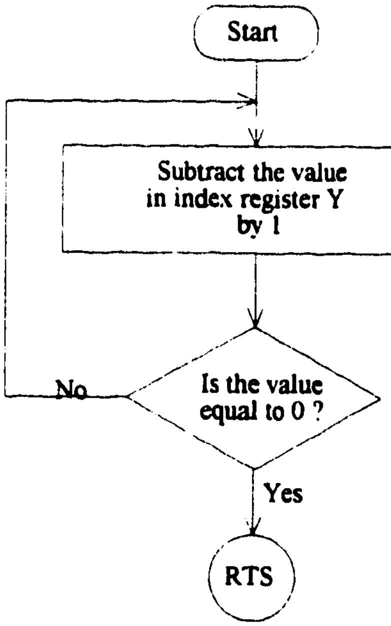


Figure 5-27 Flowchart of the DELAY subroutine

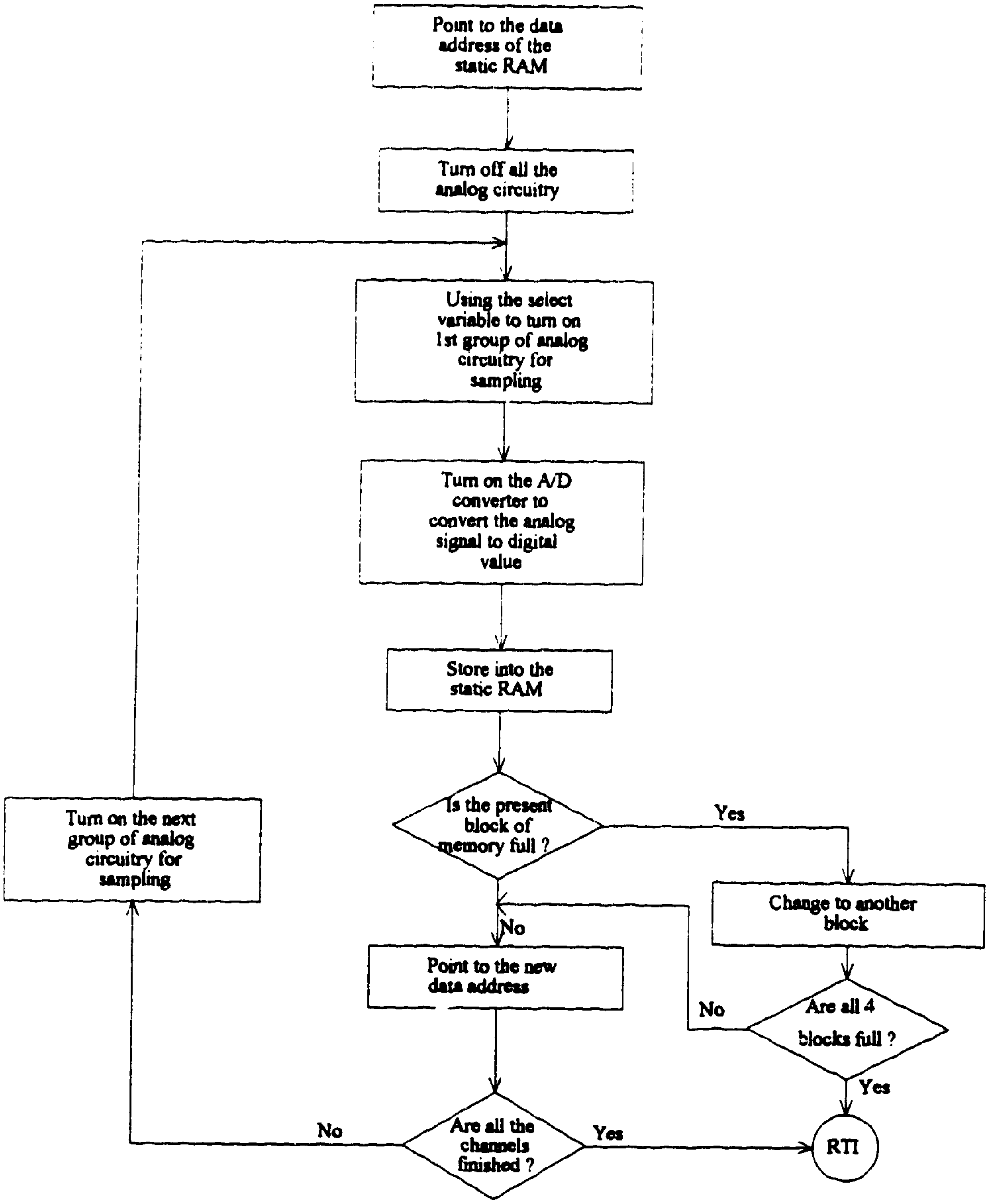


Figure 5-28 Flowchart of the ADCONV subroutine

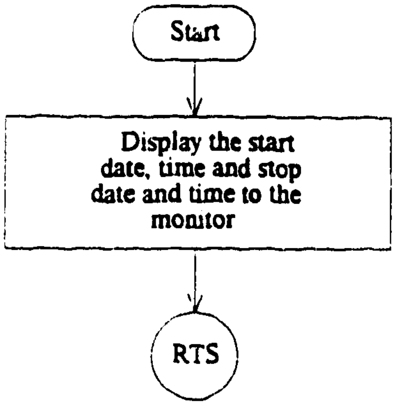


Figure 5-29 Flowchart of the OUTPUT module

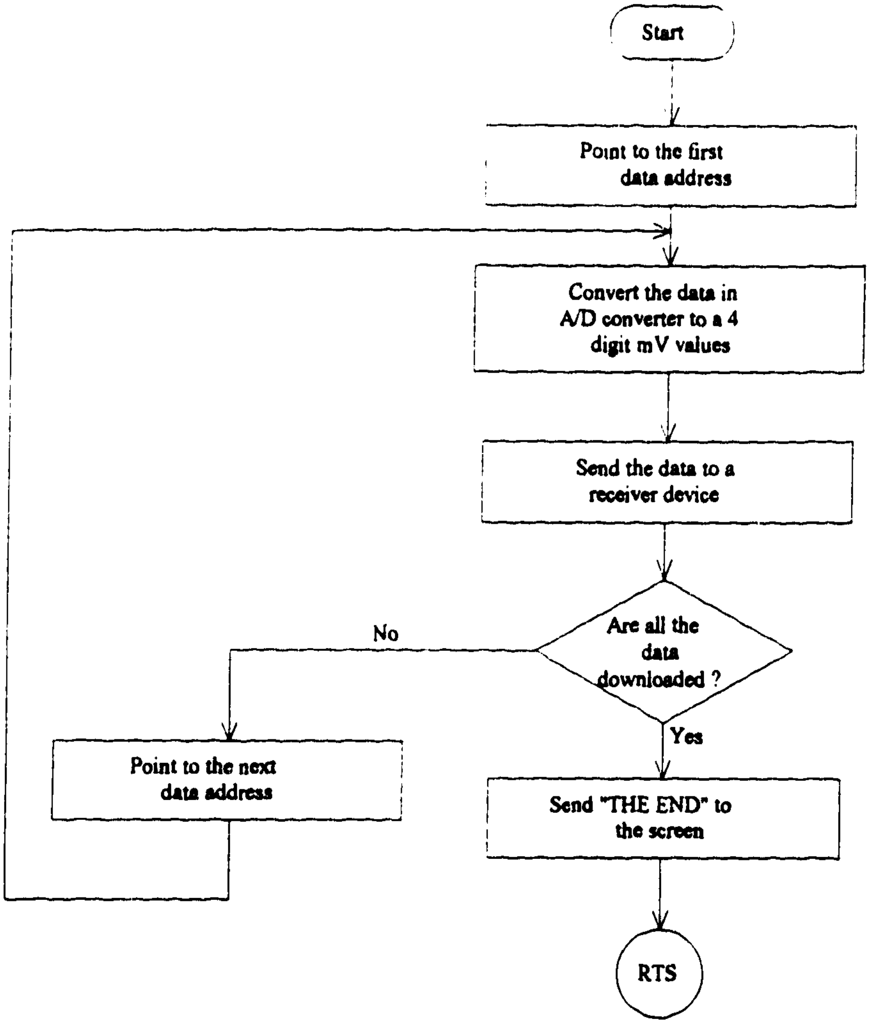


Figure 5-30 Flowchart of the DSPLAY subroutine

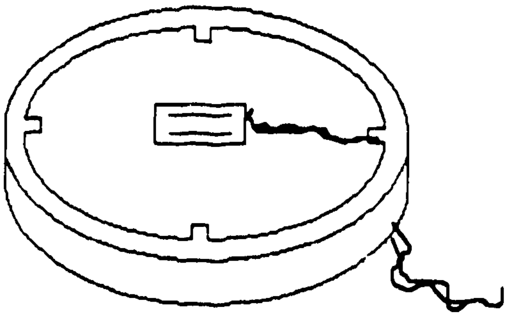
the data are transferred from the static RAM to the computer, "THE END" will be displayed on the screen. The program then returns to the MAIN routine and reverts to STOP mode.

5.2 The Transducers and Associated Analog Circuitry System

This subsystem can be divided into two parts : (1) transducer design and (2) analog circuitry design.

5.2.1 Transducer Design

Measurement of pressures exerted by the various pads of a brace requires the placement of transducers at the acting surface of each pad. The transducers must be low power devices. No suitable transducer was available commercially. Therefore, a specially designed transducer for this specific application was required. A custom design consisting of a flexible metal diaphragm, which was supported by an aluminum ring, instrumented on either side with two 120 ohm strain gauges was chosen (Figure 5-31). The use of double strain gauges provide for temperature compensation and for analog signal doubling. The typical pressure range exerted by the brace is 0 to 200 mmHg (Jiang et al, 1992). A series of experiments were conducted to evaluate the sensitivity, range and linearity of both aluminum and stainless steel transducers with various diameters and thickness. The transducers must not interfere with the normal function of the brace and must produce an electrical signal which is proportional to the pressure exerted by a region of a pad. The direction of the applied pressure measured will be determined from PA and side radiographs. Radiographic images of the transducers are digitized and the orientation of the transducers entered into a computer system for computation of forces magnitudes and directions.



Scale 4 : 1

Figure 5-31 The Designed Pressure Transducer

Notes : One notch is to let the wires pass through to the bottom.

The other three notches are used to balance the diaphragm.

The force exerted by the straps that secure the brace were measured by using the aluminum buckle shown in Figure 4-2. There are two perpendicular strain gauges on both side of the diaphragm which is about 1/32 inches thick. The purpose of using double strain gauge measurement is for temperature compensation, and the thin diaphragm provided a suitably sentivity.

5.2.2 Principles of Strain Gauge Measurement

Stress in a material cannot be measured directly, it must be computed from other measurable parameters. Therefore, the stress analyst uses measured strains in conjunction with other properties of the material to calculate the stresses for a given loading condition. There are methods of measuring strain or deformation based on various mechanical, optical, acoustical, pneumatic, and electrical phenomena.

All electrically conductive materials have a strain sensitivity - defined as the ratio of relative electrical resistance change of the conductor to the relative change in its length - and therefore can be considered as possible strain gauge materials (Window and Holister, 1982). Strain sensitivity is a dimensionless relationship expressed mathematically as :

$$F = (\Delta R/R)/(\Delta l/l) \qquad \text{Eq. 1}$$

where F = strain sensitivity = gauge factor (dimensionless)

R = initial resistance(Ω)

ΔR = change in resistance (Ω)

l = initial length of the strain gauge (inches)

Δl = change in length (inches)

Therefore, strain sensitivity is a combination of the effects of geometric changes plus a resistance change. Usually, a wheatstone bridge circuit is used for static strain measurement for electrical devices. Thus, the associated analog circuitry consists of one wheatstone bridge circuit to measure the small change of the strain gauge and amplifier circuitry to amplify the analog signal.

The transducers are supported by a circular ring and placed at the acting surface of each pad. To calculate the deviation of the flat circular plate with the uniform load applied over the concentric area of radius r_1 can be expressed as follows (OMEGA, 1987) :

$$\text{Max } \Delta = -21\, p\, r_1^2[132r^2 - 52r_1^2 \log(r/r_1) - 73r_1^2] / 1600\, E\, t^3 \qquad \text{Eq. 2}$$

where t = thickness of the plate (inches)

r = radius of the plate - the width of the supported edge (inches)

r_1 = radius of the uniform load over concentric circular area (inches)

p = uniform load on plate (psi)

E = modules of elasticity of the circular plate material (psi)

Δ = the deviation of the plate at the center (inches).

From Eq. 1, if F is given and $\Delta R/R$ is to be calculated, the value of $\Delta\, l/l$ is required. Therefore, $\Delta\, l/l$ can be expressed mathematically as :

$$\Delta\, l/l = (r^2 + \Delta^2)^{1/2} / r - 1 \qquad \text{Eq. 3}$$

5.2.3 Analog Circuitry Design

The associated analog circuitry, shown in Figure B-4, can be divided into three stages : the wheatstone bridge circuit, the differential amplifier circuit with an offset null circuit and a non inverting amplifier circuit. The quad precision operational amplifier

(LT1014CN) was chosen because both offset current and offset voltage of this IC are very low, and the amplifying gain is very large. Vsup is the supply voltage to the bridge. R_{g1} and R_{g2} are the resistances of the strain gauge. R_1 and R_2 are the resistances of the bridge completion resistors. V_{offset} and I_{offset} are the offset voltage and offset current due to the OP-AMP respectively. V_{out} is the final output voltage. R_{g2} is used for temperature compensation. When a force or pressure is applied to a transducer, the expression of Vout can be calculated from the following equations :

$$Rin = 100K // (R_1+R_2) // (R_{g1}+\Delta R_{g1}+R_{g2}-\Delta R_{g2}) \qquad \text{Eq. 4}$$

The ΔR_{g1} and ΔR_{g2} can be calculated from Eq. 1, Eq. 2, and Eq. 3 when the applied pressure is known.

Since $100K >> R_1+R_2$ and $R_{g1}+R_{g2}$, $R_1 = R_2 = R$, $R_{g1} = R_{g2} = R_g$ and $\Delta R_{g1} = \Delta R_{g2} = \Delta R_g$

$$Rin = 2R//2R_g \qquad \text{Eq. 5}$$

$$Vsup = V * Rin/[1.2K + Rin] \qquad \text{Eq. 6}$$

$$Vin = Vsup * (R_g - \Delta R_g)/2R_g \qquad \text{Eq. 7}$$

$$V_o = (6.8M/10K) * (Vin - Vref) \qquad \text{Eq. 8}$$

The maximum amplitude of the total output offset voltage due to the OP-AMP is :

$$V_{total\ offset} = 680K * I_{offset} \qquad \text{Eq. 9}$$

$$\therefore \quad Vout = V_{total\ offset} + Vo * (680K/10K) \qquad \text{Eq. 10}$$

5.3 Summary of the System Design

The hardware and software designs for the system were completed and a prototype system is being constructed. The maximum amp-hour power requirement of the system was calculated to be 269.5mAh in two weeks with 16 channels on and sampling interal was 2.5 minutes per sample, and four AA size NiCad rechargeable batteries were chosen as the power supply. The software program was written in MC6811 assembler and was burned into a 8K CMOS ROM. Strain gauge transducer system was chosen. The pressure transducers diaphragm made of stainless steel with diameter 1/2 inch was finally chosen for this application. Buckle transducers made of aluminum were designed to measure the strap forces. The analog circuitry contained a wheatstone bridge circuit, a differential amplifier with an offset null voltage and a non inverting amplifier circuit were designed.

6. PROTOTYPE VERIFICATION AND TEST RESULTS

6.1 Digital System

The digital system was first prototyped on a SK-10 board. Then, an assembly program was written on a file and burned into the ROM to verify that both the circuit and the software were correct. Critical signal waveforms, 8 MHz. oscillator, 2 MHz E clock, the signal lines PA3 to PA6 and PD2 to PD5, and the interrupt signal from the RTC, were checked by a digital oscilloscope. After verifying that the circuit was correct and the software was executing properly; a series of tests on modified circuits were done to reduce the size and power consumption of the system. Finally, a minimum number of the ICs and minimum current consumption for the system were achieved. The schematic of the designed circuit is shown in Appendix B (Figure B-1, B-2 and B-3). A printed circuit board layout (Figure B-5) was then drawn using Orcad. The part lists of the system is given in table B-1. The software listing is shown in Appendix C.

6.2 Transducer and Analog System

One channel of the analog circuit was prototyped onto another SK-10 board. Then an initial prototyped transducer with a 7/8 inch diameter aluminum diaphragm of thickness 0.03125 inch instrumented with two 120 ohm foil strain gauges (Showa Measuring Instruments Co. Ltd, Tokyo) was connected to the circuit. The analog circuit was verified by imposing a pressure, ranging from 0 to 200 mmHg, on the transducer. Since pressure equals to force divided by area. The maximum load was calculated. Then, a weight equaled to the maximum load was placed on the foam pad which was on the diaphragm An output voltage was measured, and the amplifying gain of the analog circuit was adjusted until a full range voltages, 0 to 4.5 Volts, was obtained. After verifying that the circuit and transducer worked properly, the sensitivity of the transducer was considered. A second diaphragm made of stainless steel with thickness 0.011 inches was made. Experiments were done to test the sensitivity of these two diaphragm by putting different weights on top of the foam pad which was on the diaphragm, and the corresponding output voltages were recorded. Data for both loading and unloading were obtained. The comparison of the test results between the stainless steel and aluminum is shown in Figure 6-1. From the results, stainless steel was found to be a better material for the diaphragm. After considering the sensitivity of the transducer, the size was [illegible] [illegible] design a smaller size transducer, shorter gauge length of the strain gauge was used. Diaphragms made of stainless steel with diameter of [illegible]/2 inches were made to investigate the effect of size. The sensitivity test was repeated, and similar results, both ouput voltages are linearly proportional to the applied pressure, were obtained. Figure 6-2 shows the calibration results of the transducer with 1/2 inch diameter. Finally the 1/2 inch transducer was chosen for this application because the physical site for placing the transducer could be more precisely.

The buckle transducer made of aluminum, used similar analog circuitry except that the amplifying gain was different. Calibration of the buckle was done by hanging, 1, 1.5, 2, 2.5, 3, 3.5, 5, 5.5, 6, 6.5, 7, 7.5, 8 and 8.5 kg, weight from the buckle. The test results for the buckle are shown in Figure 6-3. The strap forces on the buckle were linearly proportional to the output voltages.

Printed circuit boards for both digital and analog system were fabricated to reduce the noise and size of the system. After the printed circuit boards were fabricated, two boards were connected by a card edge connector. The prototype of the system is shown in Figure 6-4. The microcomputer, analog system and battery were housed in an aluminum box with dimensions of 16cm x 19cm x 4cm Transducer were fixed to the hard shell of the brace by using a circular arc punch to core through the foam lining. The cored out foam was put back on top of the transducers to ensure the surface was flat. Sixteen transducers were connected to two male header connectors, which were mounted on the outside surface of the brace, by using 32 AWG fine wires which were developed by the Alpha Wire Corporation (New Jersey) These wires were embedded into the foam lining. The system may be mounted on the brace with double side velcro or carried in a pouch suspended by a belt.

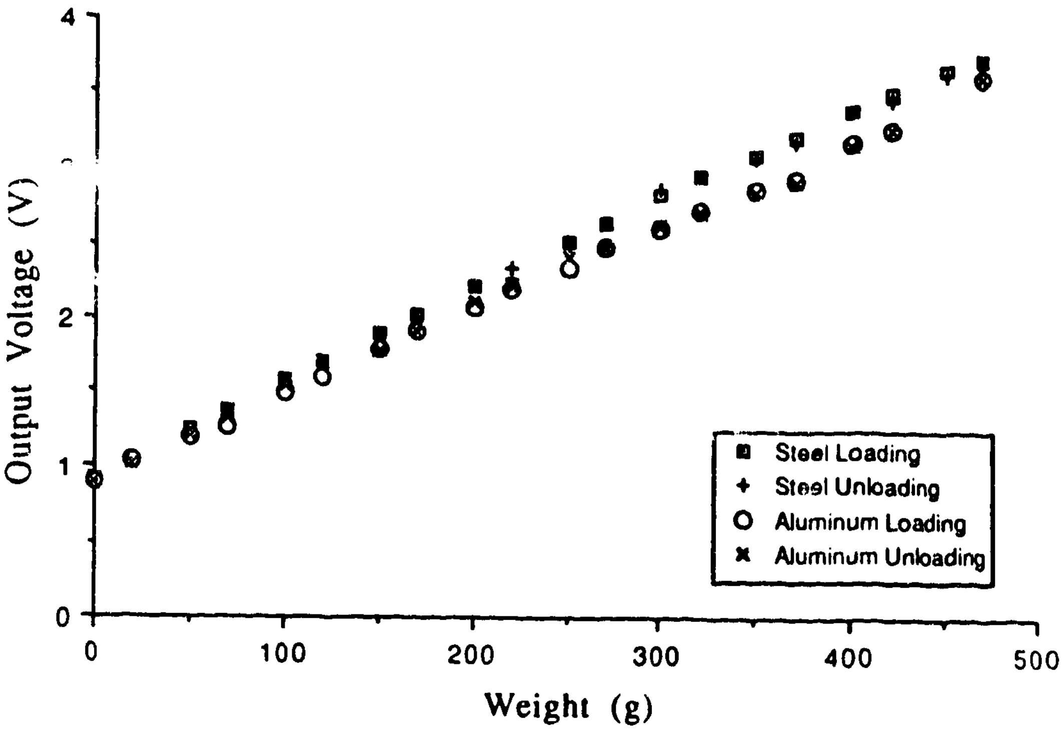


Figure 6-1 Comparsion of the Sensitivity of Aluminum and Stainless Steel

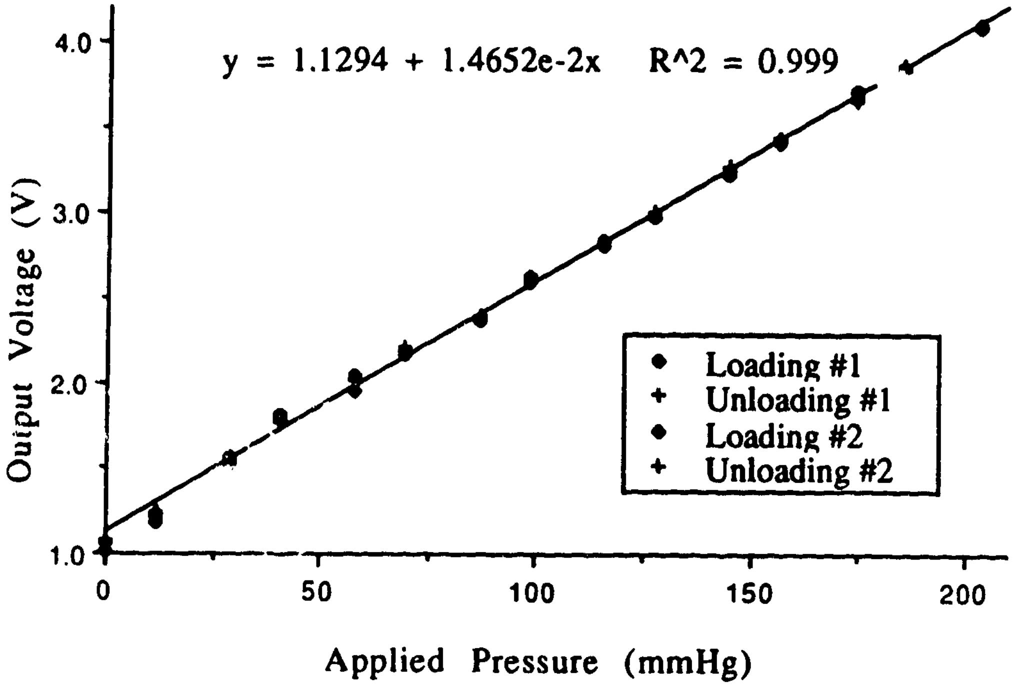


Figure 6-2 Calibration of Stainless Steel Transducer

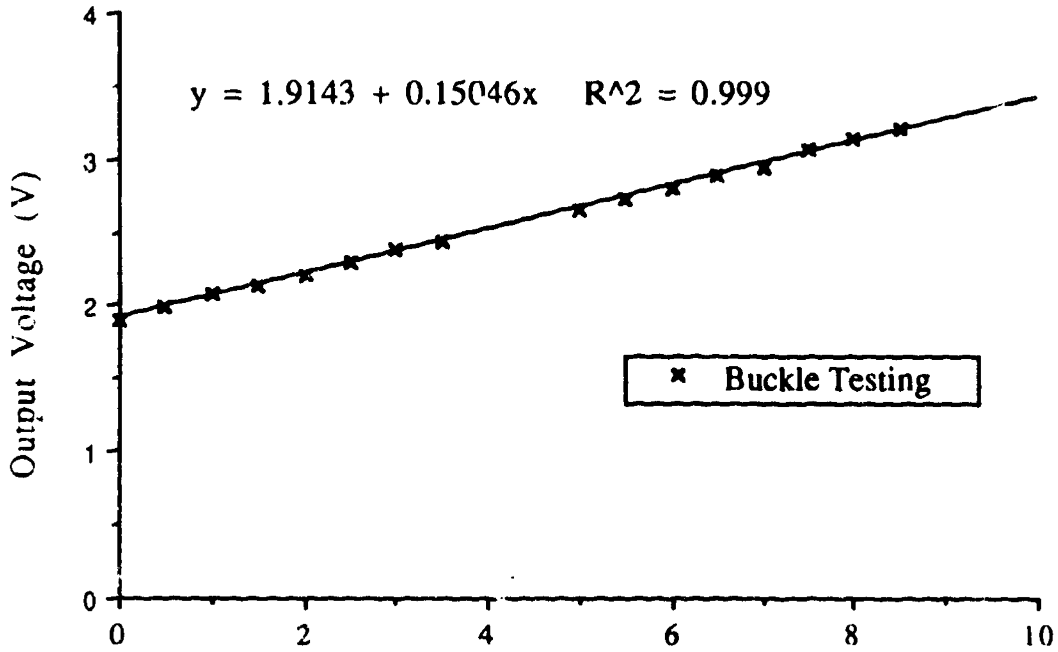


Figure 6-3 Calibration of the Buckle Transducer

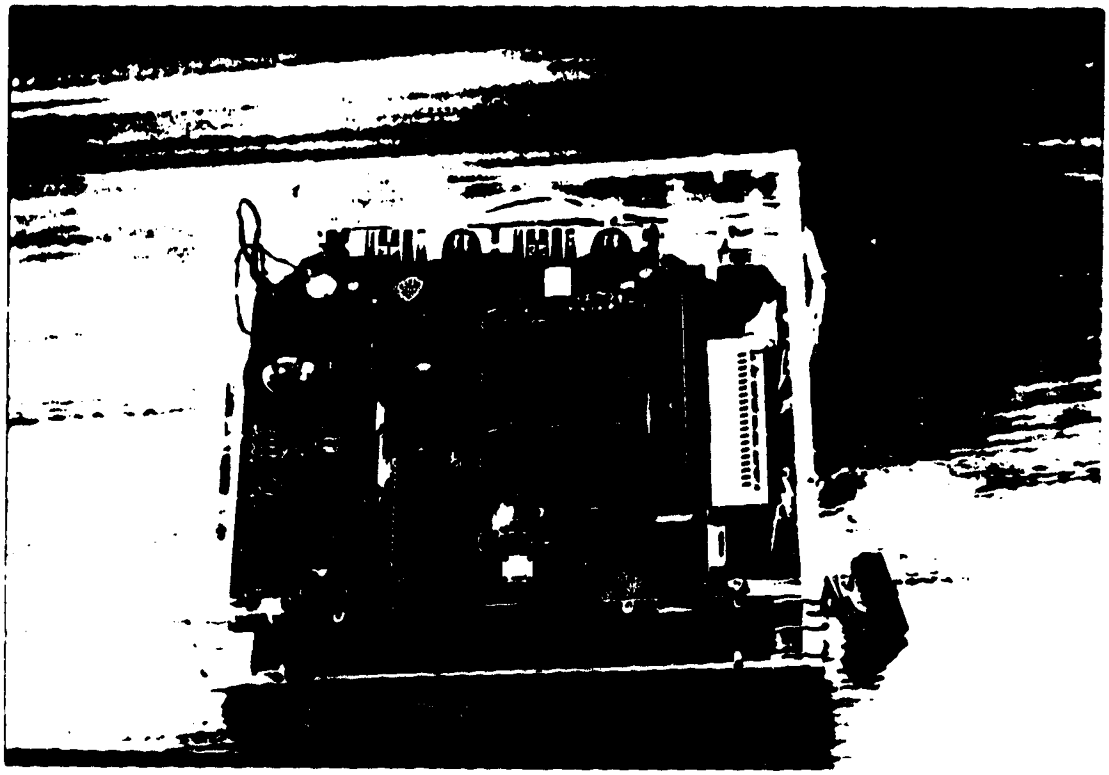
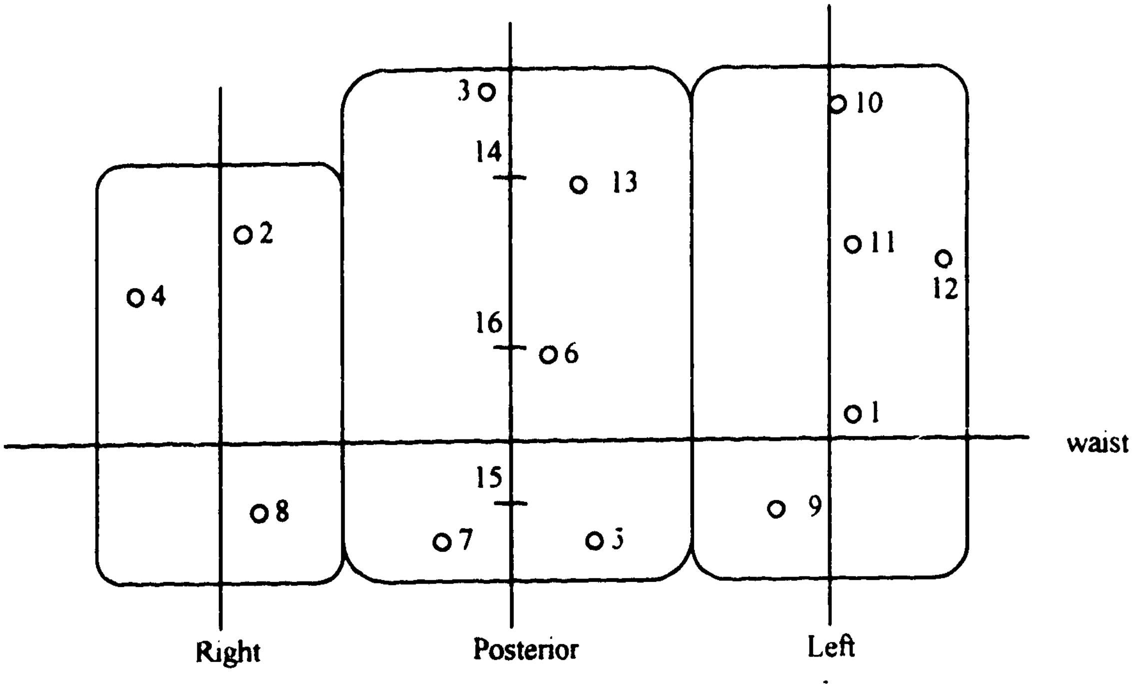


Figure 6-4 The Prototype System

6.3 Clinical Trial

The system was field tested by instrumenting a nighttime Charleston brace used by a young man (age: 18yrs). This subject had a 37^{o} right thoracic curve from T7 to T11 and was in the final phase of brace treatment. The objectives of this field tested were to prove the ruggedness and the relability of the system.

Thirteen pressure transducers were placed as follows: four on the left and right axillary lines, three on the back surface, two on the front surface and four on the buttocks region. The three straps that secured the brace were instrumented. Figure 6-5 shows the positions of the transducers in the brace. The pressure and strap forces were recorded over two days with measurements taken every minute. The data were then transferred to a Macintosh (Quadra 700) computer for analysis. Pressures at the axillary lines and forces in the straps were plotted (Figure 6-6 and Figure 6-7). The peak loads which occured when the brace was first donned was 100N at the middle strap, 80N for the bottom strap and 68N for the top strap. The maximun pressures occured at the axillary lines regions and ranged between 60 mmHg to 150 mmHg were the scoliosis correcting pressures. The mid-right and left axillary pressure variation may be due to sideway movements of the body. At the back surface, the measured pressures ranged between 34 mmHg and 83 mmHg; the pressure at the front surface ranged between 15 mmHg and 65 mmHg. The minimum pressures occured at the buttocks regions and ranged between 10 mmHg to 50 mmHg.



Channel 1-13 : Pressure transducers

Channel 14 -16: Buckle transducers

Figure 6-5 Transducers position in the Brace

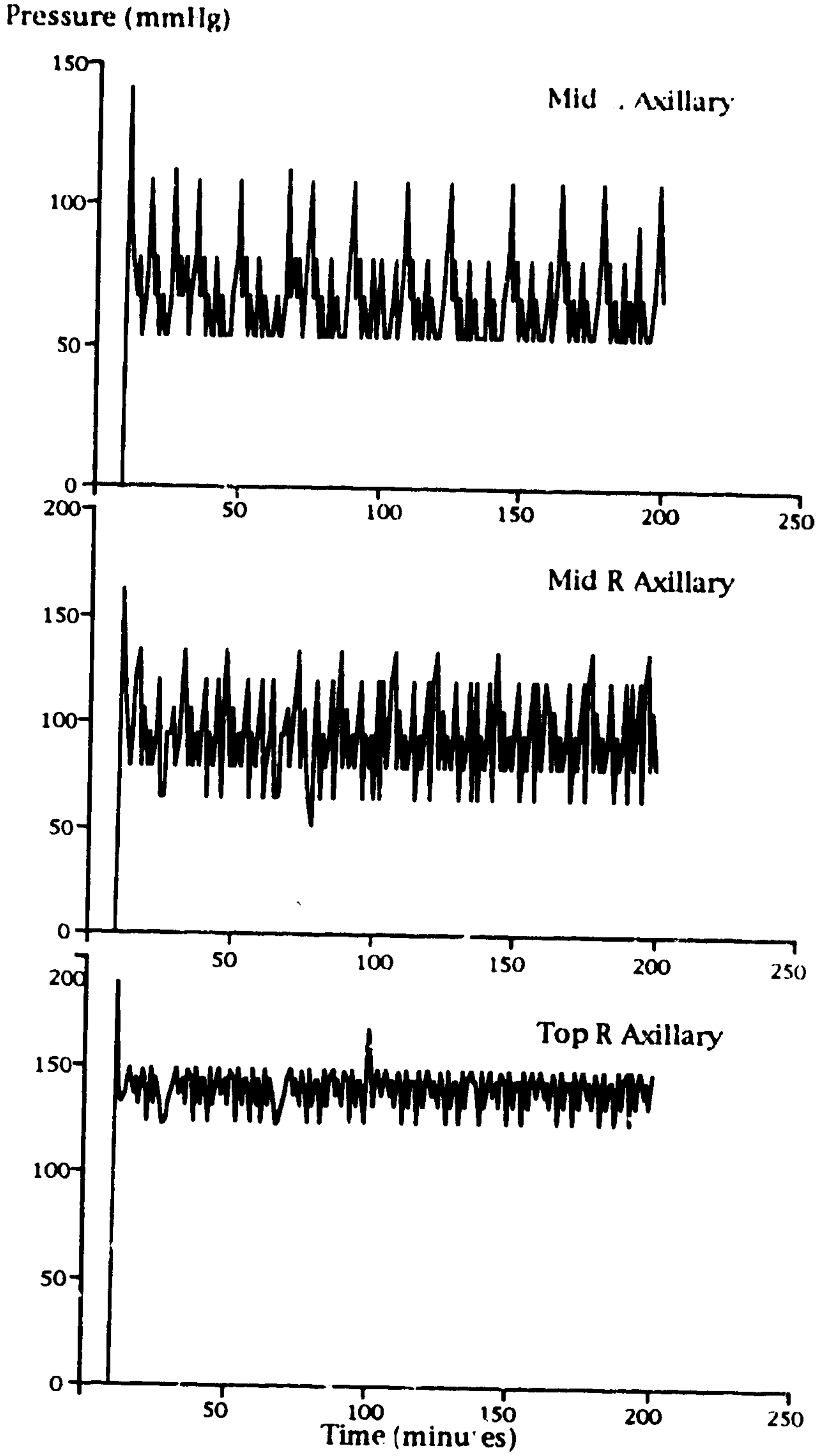


Figure 6-6 Pressures along the Axillary lines

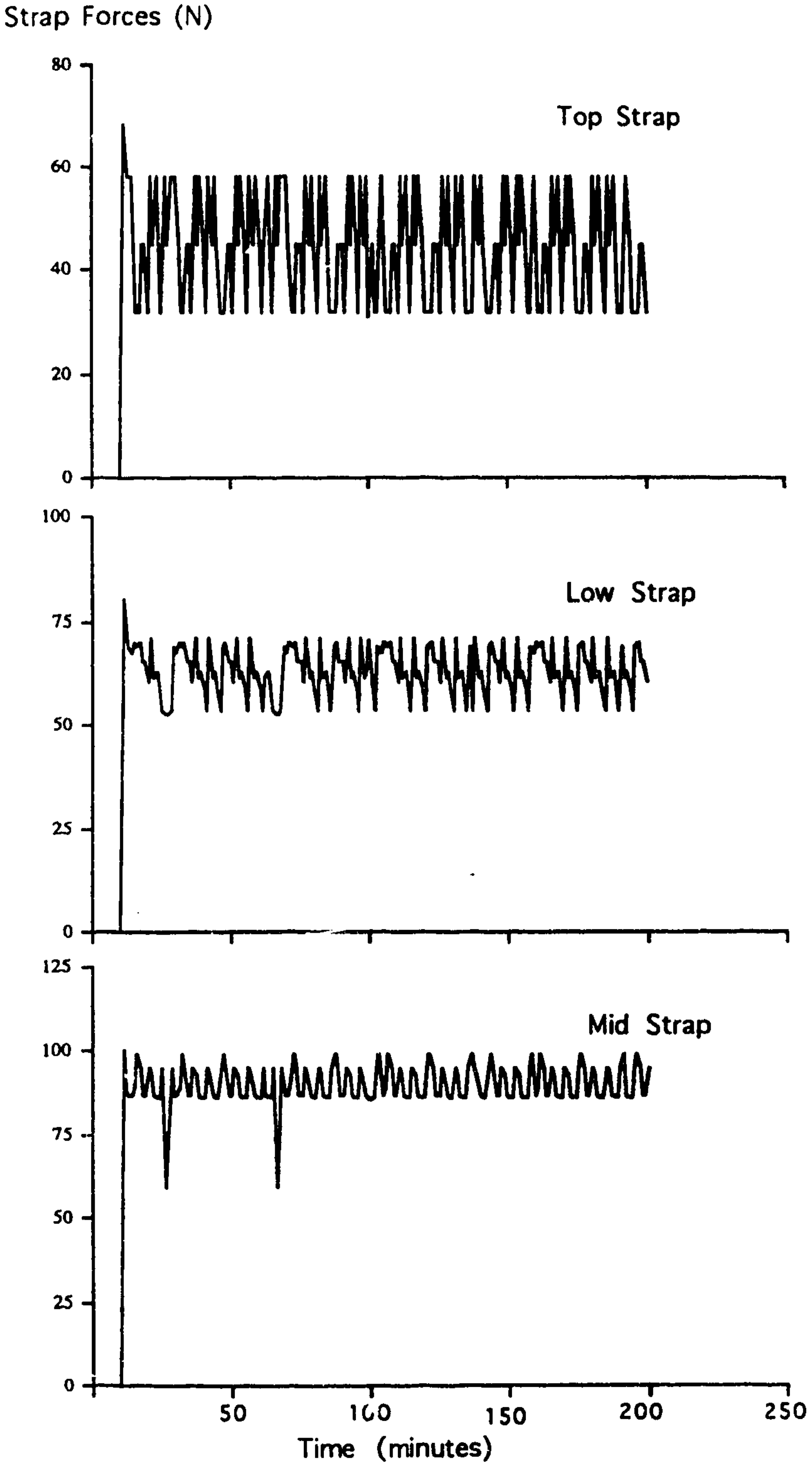


Figure 6-7 Strap Forces

7. CONCLUSION AND RECOMMENDATIONS

7.1 Conclusion

There is no system available to measure, over an extended period, the pressures exerted by braces or the strap forces in braces used to treat children with spine deformities. The intent of this research was to develop a data logg/transducer system that could be worn by a child over an extended period. The data acquisition system continuously monitors pressure transducers which measure the pressure between the brace and the trunk and buckle transducers that measure the force exerted by the straps that secure the brace. The system requires about 300 mAh to collect 8192 samples over two weeks. The low power real time clock controls the sample time and interval and provides an interrupt to the MCU so that the system can be placed in STOP mode when data is not being acquired. A built-in static RAM provides backup for all acquisition parameters. A backup battery maintains the memories in the RTC and in the 128 KB static RAM in case of a power interruption. All of the devices consume little power in STOP mode, the most significant time period. PAL is used to reduce the size. The voltage output from the transducers is linearly proportional to the applied pressure with $r^2 = 0.999$. Clinical studies of the mechanical effect of braces used in the treatment of scoliosis have begun. This information will be used to determine if the brace applies appropriate pressures to correct the scoliotic curve and will also document how well the patient adheres to instructions on how his brace should be worn. The system is not limited to use in scoliosis studies but may be used for a variety of other pressure or force measurements studies of prosthetic attachments.

7.2 Recommendations for Future Work

The system can be enhanced with improvements to both hardware and software that reduce the size, the weight, the power consumption and that provide more effective use of the memory space.

Hardware enhancements that will reduce the size, weight, and power of the system include:

(1) Replace the existing MC68HC11A1 microcontroller by MC68HC811A2 because it has a 2K EEPROM inside the chip. Therefore, the ROM chip can be eliminated, more RAM can be used and the sampling rate increased.

(2) Use Surface mount technology.

(3) Fabricate each channel of the analog system as an individual IC so that the reliability can also be increased and the noise generated from the analog system decreased.

(4) Divide the analog circuit board into two boards and installed the resistors vertically in the board to save space.

To make more effective use of the memory space, software enhancements should include such options as user selectable data channels and a user selectable sample interval.

REFERENCES :

Aaro S., Bustrom R. and Dahlborn M.: The derotating effect of the Boston brace: A comparison between computer tomography and a conventional method. **Spine** 6, pp.477-482, 1981.

Andrisechi T. P., Schultz A. B., Belytschko T. B. and Dewald R. L., Milwaukee Brace Correctionof Idiopathic Scoliosis. **The Journal of Bone and Joint Surgery** 58-A, pp.806-815, 1976.

Asher M. A. and Whitney W.H.: Orthotics for Spinal Deformity. **Orthotics Etcetera** (Eds Redford J. B., Williams and Wilkins), Baltimore, 1986.

Asher M., De Semt A. Whitney W., Onnish R. N., Cook L. and Bramble J.: Changes in Spinal Alignment Resulting from Spinal Deformity Orthoses in Patients with Adolescent Idiopathic Scoliosis. **Orthopedics Trans.** 11, pp.105-106, 1987.

Chase A. P., Bader D. L. and Houghton G. R.: The Biomechanical Effectiveness of the Boston Brace in the Management of Adolescent Idiopathic Scoliosis. **Spine** 14, pp.636-639, 1989.

Cote B., Dansereau J., Labelle H. and Reinecke S.: A Method for the Measurement of the Boston Brace Biomechanical Action on Scoliotic Deformities. **Proceedings of Int'l Symposium on 3-D Scoliotic Deformities**, (Ed Dansereau J.) Montreal, June 27-30, pp.81-88, 1992.

Dansereau J., Labelle H. and Bellefleur C.: Evaulation of the immediate effect of the Boston brace on scoliotic deformity by means of spinal and rib cage 3-D reconstruction. 37th annual meeting, Orthopaedic Research Society, p.639, 1991.

Houghton R. H., Mclnerney A. and Tew T: Monitoring True Brace Compliance, **Proceedings of the 21st. Meeting of the Scoliosis Research Society**, Hamilton, Bermuda, September, 1986.

Jiang H, Raso V. J., Hill D. L., Durdle N. G. and Moreau M: Interface Pressures in the Boston Brace Treatment for Scoliosis, **Proceedings of Int'l Symposium on 3-D Scoliotic Deformities**, (Ed Dansereau J.) Montreal, June 27-30, pp.395-399, 1992.

Kehl D. K. and Morrissy R. T.: Brace Treatment in Adolescent Idiopathic Scoliosis. **Clinical Orthopaedics and Related Research** 229, pp.34-42, 1988.

Moe J. H., Bradford D.S., Lonstein J.E., Ogilvie J.W. and Winter RB: Moe's Textbook of SCOLIOSIS and other Spinal Deformity, 2nd. edition,W.B. Saunders Company 1987.

OMEGA: Pressure and Strain, **An OMEGA Technologies Company**, pp.C28-C31, 1987.

Patwardhan A., Dvonch V., Bunch W., Vanderby R, Sartori M, Lorentz M and Meade K.. Biomechanics of the Lumbar and Thoracic Pads in orthotic treatment for idiopathic scoliosis, **Proceedings of the 21st. Meeting of the Scoliosis Research Society**, (Eds Lantz S.A. and King A. I.) Hamilton, Bermuda, September, pp.50-51, 1986.

Price C. T., Scott D. S., Reed F. E., Riddick M. F.: Nighttime Bracing for Adolescent Idiopathic Scoliosis with the Charleston Bending Brace, **Spine** 15, pp 1294-1299, 1990.

Shinoto A., Ohtsuka A. and Inoue S.:Three deminsional analysis of the effect of brace treatment on idiopathic scoliosis. **Proceedings of the 4th Int'l Symposium on Surface Topography and Spinal Deformity**, Gustav Fisher Verlag, New York, pp.113-130, 1987.

Watts H. G., Hall J. E. and Stanish W. M.: The Boston Brace System for the Treatment of Low Thoracic and Lumbar Scoliosis by the Use of a Girdle Without Superstructure. **Clinical Orthopaedics and Related Research** 126, pp.87-92, 1977.

Weisz I., Jefferson R. J., Carr A. J., Turner-Smith A. R., McInerney A.and Houghton G. R.: Back Shape in Brace Treatment of.Idiopathic Scoliosis, **Clinical Orthopedics and Related Research** 240, pp.157-163, 1989.

Willner S.: Effect of the Boston thoracic brace on the frontal and sagittal curves of the spine. **Acta Orthop Scand** 55, pp.457-460, 1984.

Window A. L., Holister G. S.: Strain Gauge Technology, **Applied Science Publishers**, pp3-5, 1982.

Ylikoski M., Peltonen J. and Poussa M.: Biological Factors and Predictability of Bracing in Adolescent Idiopathic Scoliosis, **Journal of Pediatric Orthopedics** 9, pp.680-683, 1989.

Appendix A

Operating Instructions

Download the Input Parameters

(1) Connect the 232-serial port of the data acquisition system to a PC.

(2) Execut any software program which can communicate with 232-serial port in the PC.

(3) Power up the acquisition system and push the reset button.

(4) Follow the instructions to input the parameters.

(5) Remove the connection cable between the system and PC.

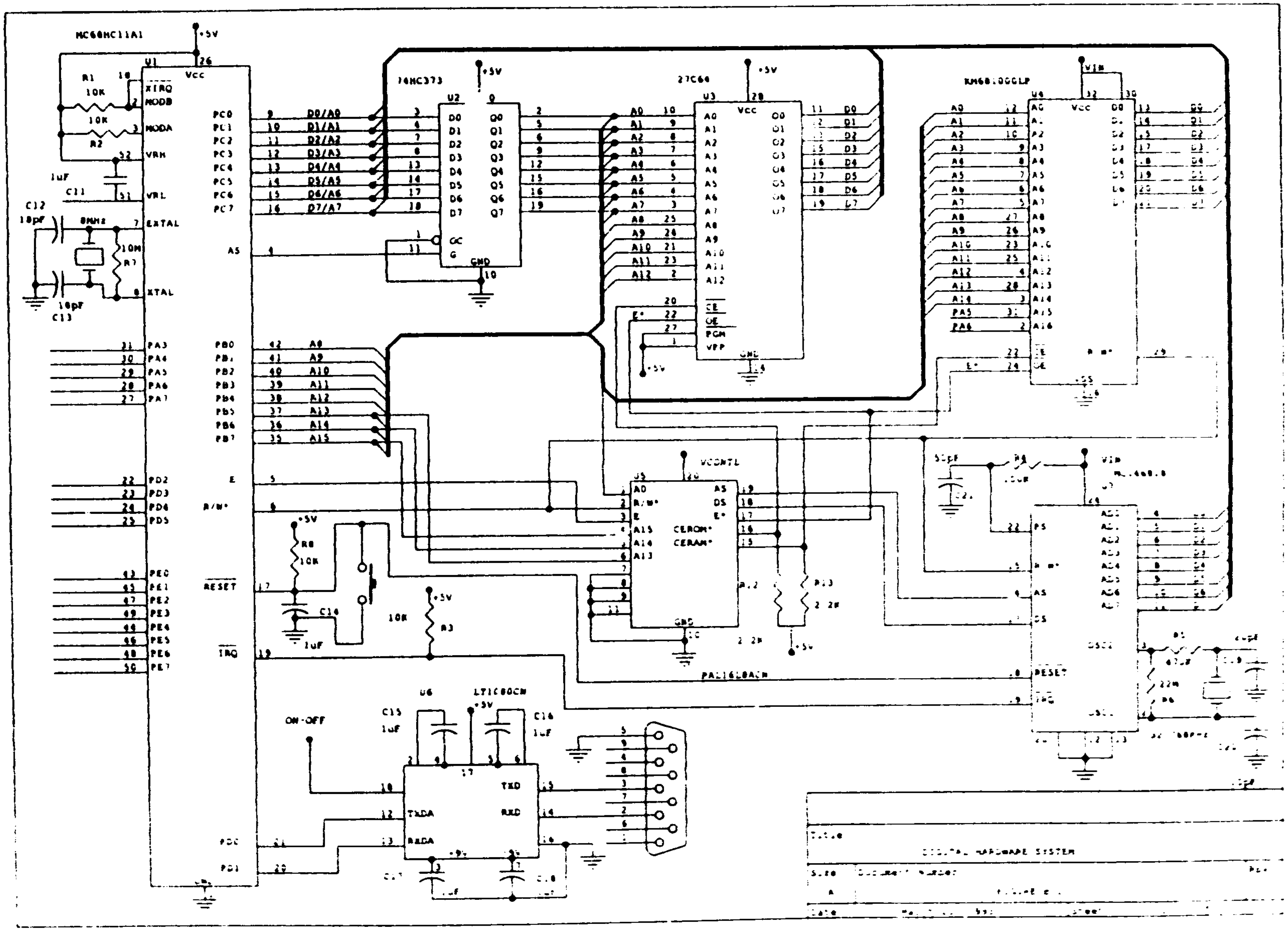
Upload the Data Acquistion Data

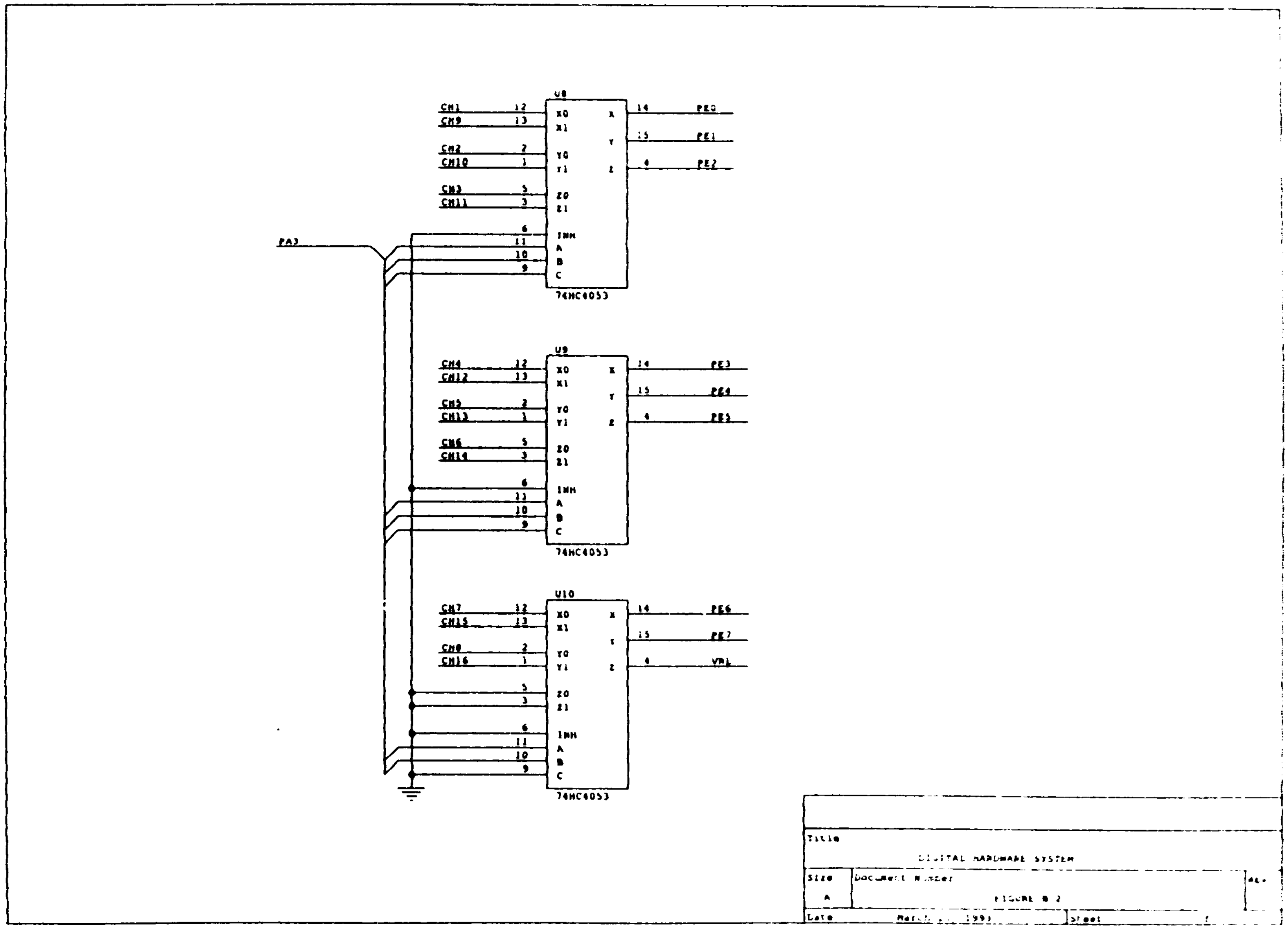
(6) First check the voltage of the power supply to the system, replace it if necessary

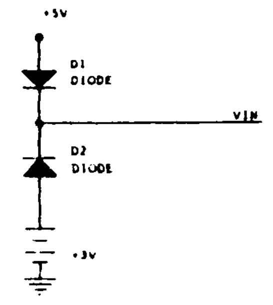
(7) Repeat procedure (1) - (5)

Appendix B

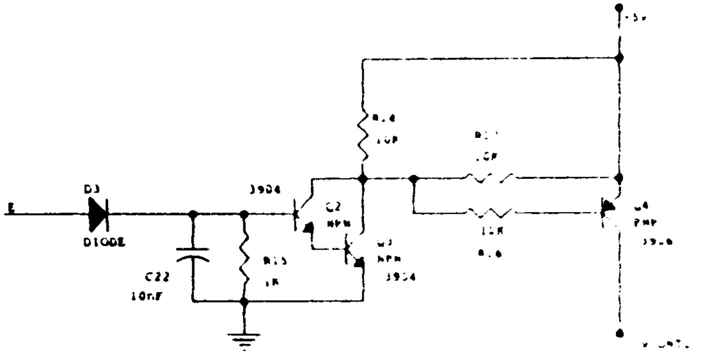
Diagrams and Parts List of the Data Acquisition System


MC68HC11A1
U1
+5V
VCC
XIRQ
MODB
MODA
VRH
VRL
EXTAL
XTAL
R1 10K
R2 10K
1uF
C11
C12 10pF
C13 10pF
8MHz
10M
R7
PC0
PC1
PC2
PC3
PC4
PC5
PC6
PC7
AS
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
E
R/W*
RESET
IRQ
PD0
PD1
PD2
PD3
PD4
PD5
PE0
PE1
PE2
PE3
PE4
PE5
PE6
PE7
D0/A0
D1/A1
D2/A2
D3/A3
D4/A4
D5/A5
D6/A6
D7/A7
A8
A9
A10
A11
A12
A13
A14
A15
74HC373
U2
GC
G
GND
27C64
U3
CE
OE
PGM
VPP
E*
HM681000LP
U4
VIN
PA5
PA6
R/W*
VCONTL
U5
A0
R/W*
E
A15
A14
A13
AS
DS
E*
CEROM*
CERAM*
PAL16L8ACN
R12
R13
2.2K
R8 10K
C14 1uF
R3 10K
ON-OFF
U6
LT1080CN
C15 1uF
C16 1uF
TXD
RXD
TXDA
RXDA
MC146818
U7
PS
AD0
AD1
AD2
AD3
AD4
AD5
AD6
AD7
OSC1
OSC2
R9
22M
32.768KHz
Title
DIGITAL HARDWARE SYSTEM
Size
Document Number
A
Rev
Date
Sheet

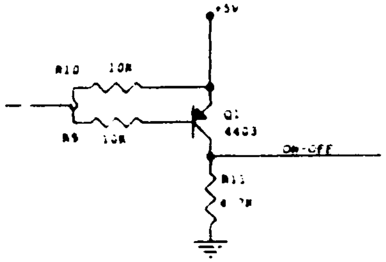

U8
U9
U10
74HC4053
CH1
CH9
CH2
CH10
CH3
CH11
CH4
CH12
CH5
CH13
CH6
CH14
CH7
CH15
CH8
CH16
X0
X1
Y0
Y1
Z0
Z1
INH
A
B
C
X
Y
Z
PE0
PE1
PE2
PE3
PE4
PE5
PE6
PE7
VRL
PA3
12
13
2
1
5
3
6
11
10
9
14
15
4
Title
DIGITAL HARDWARE SYSTEM
Size
A
Document Number
FIGURE B 2
REV
Date
March
Sheet



BACKUP POWER FOR SRAM AND RTC



POWER CONTROL CIRCUIT FOR PAL



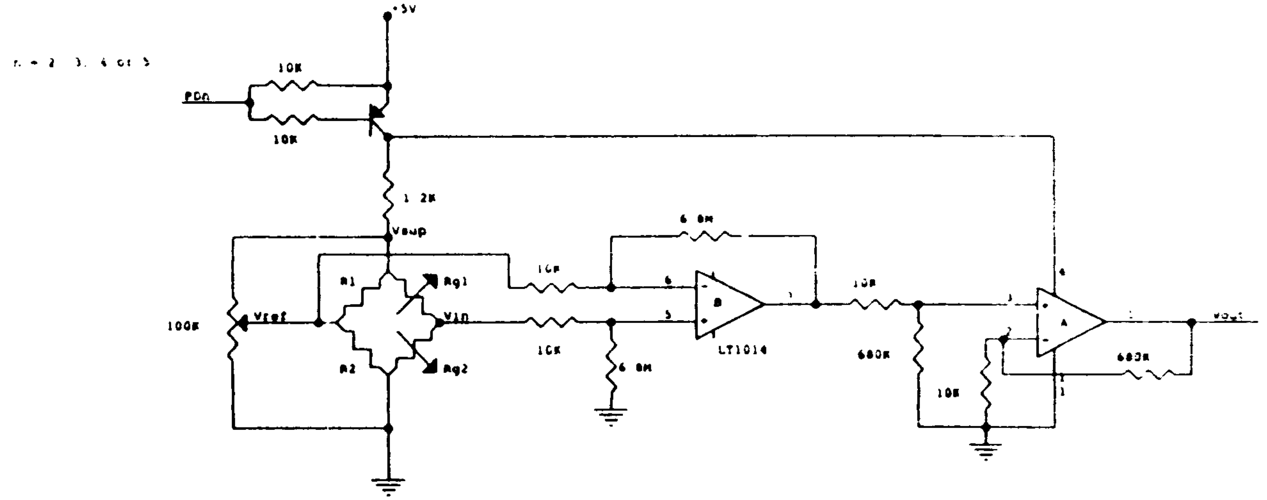
POWER CONTROL CIRCUIT FOR THE 232 DRIVER-RECEIVER

Title

POWER CONTROL CIRCUIT

Size | Document Number | REV
A | FIGURE 3.3

Date [illegible] Sheet



ANALOG CIRCUIT FOR THE TRANSDUCER SYSTEM

Title	ANALOG CIRCUIT		
Size	Document Number		REV
A	FIGURE B 4		
Date	March 20, 1991	Sheet	of

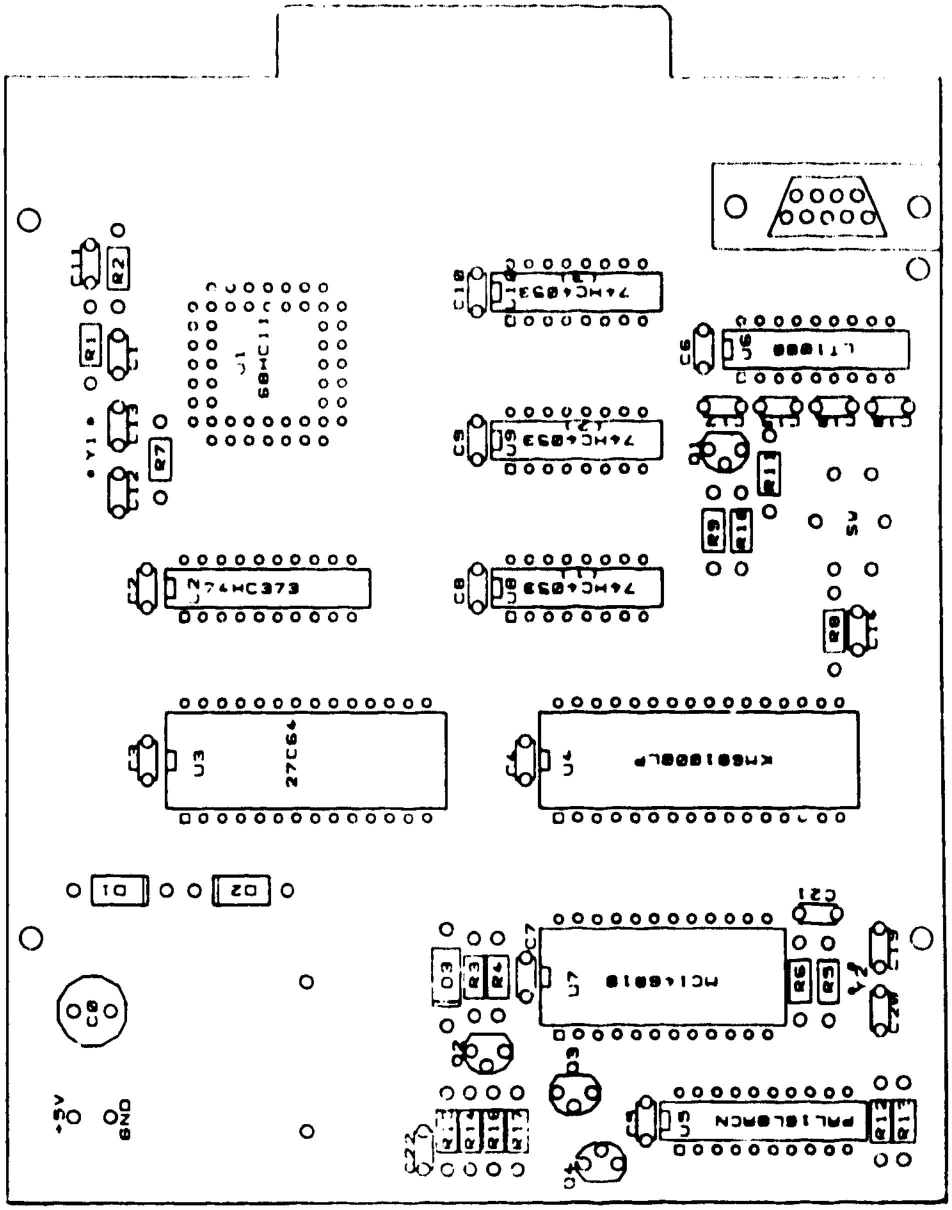


Figure B-5 The Digital Parts Location Diagram

Reference Designation	Description
C0	Capacitor, 10μF
C1 - C10	Capacitor, 0.1μF
C11, C14 - C18	Capacitor, 1μF
C12,C13	Capacitor, 18μF
C19	Capacitor, 20μF
C20	Capacitor, 10pF
C21	Capacitor, 50pF
C22	Capacitor, 10nF
D1 - D3	Diode
Q1	Transitor, PNP 4403
Q2, Q3	Transitor, PNP 3904
Q4	Transitor, PNP 3906
R1 - R3, R8 - R10, R14, R16, R17	Resistor, 10 kΩ, 5%, 1/4W
R4	Resistor, 100 kΩ, 5%, 1/4W
R5	Resistor, 470 kΩ, 5%, 1/4W
R6	Resistor, 22 MΩ, 5%, 1/4W
R7	Resistor, 10 MΩ, 5%, 1/4W
R11	Resistor, 4.7 kΩ, 5%, 1/4W
R12,R13	Resistor, 2.2 kΩ, 5%, 1/4W
R15	Resistor, 1 kΩ, 5%, 1/4W
SW	Switch, pushbutton
U1	I.C., 68HC11A1FN, MCU
U2	I.C., 74HC373, transparent latch
U3	I.C., 27C64, 8k EPROM
U4	I.C., KM681000LP, 128k RAM
U5	I.C., PAL16L8ACN, PAL
U6	I.C., LT1080CN, RS-232 driver
U7	I.C., MC146818, RTC
U8 - U10	I.C., 74HC4053, Analog Mux / DeMux.
Y1	Crystal, MCU, 8.0MHz
Y2	Crystal, RTC, 32.768KHz

Table B-1 Parts List of the Digital System

Appendix C

Software Listing

```
*******************************************************************************
*                                                                             *
* THIS PROGRAM IS WRITTEN FOR A DATA ACQUISITION SYSTEM BASED ON              *
* MC68HC11 MICROCONTROLLER SYSTEM. IT ALLOWS USER TO INPUT THE                *
* PARAMETERS, TAKES THE SAMPLES OF DATA, CONVERTS THE SAMPLED DATA            *
* INTO DIGITAL VALUES, AND DOWNLOADS INTO A FILE.                             *
*                                                                             *
*******************************************************************************

REGBS       EQU $1000           ;Start of Registers
PORTA       EQU REGBS           ;I/O Port A
PORTD       EQU REGBS+$8        ;I/O Port D
DDRD        EQU REGBS+$9        ;Data Direction for Port D
PACTL       EQU REGBS+$26       ;Pulse Accumulator Control Register
BAUD        EQU REGBS+$2B       ;SCI Baud Rate Control
SCCR1       EQU REGBS+$2C       ;SCI Control Register 1
SCCR2       EQU REGBS+$2D       ;SCI Control Register 2
SCSR        EQU REGBS+$2E       ;SCI Status Register
SCDR        EQU REGBS+$2F       ;SCI Data(Read RDR, Write TDR)
ADCTL       EQU REGBS+$30       ;A/D Control Register
ADR1        EQU REGBS+$31       ;A/D Result Register 1
OPTION      EQU REGBS+$39       ;System Configuration Options Register
HPRIO       EQU REGBS+$3C       ;Highest Priority I-Bit Int and Misc
CONFIG      EQU REGBS+$3F       ;COP, ROM, and EEPROM Enables

MEMORY      EQU $0000           ;ram address of the real time clock in MCU
RTC         EQU $4000           ;address of the real time clock
DATA        EQU $6000           ;starting address of the data
LF          EQU $0A             ;ASCII value for linefeed
CR          EQU $0D             ;ASCII value for carriage return
STEP        EQU #192            ;19.2mV
ADDELAY     EQU #$140           ;time delay for each group of sampling
VARSIZE     EQU #46             ;total variable sizes

*memory allocation for variables

            ORG $0A
COL         RMB 1               ;number of column
COUNT       RMB 1               ;count the number of days
CHANNEL     RMB 1               ;number of channels
FLAG        RMB 1               ;general flag
POS         RMB 1               ;postion
PERIOD      RMB 1               ;time period of all the samplings
SELECT      RMB 1               ;channel select
```

```
SIGNAL      RMB 1           ;signal for checking memory full
TMP         RMB 1           ;temporary memory
TEN         RMB 1           ;the ten memory
VALUE       RMB 1           ;the digital value
DATAVEC     RMB 2           ;point to the data address
DSPADDR     RMB 2           ;point to the display address
INTRVL      RMB 2           ;interval between each group of sampling
KEYBUF      RMB 3           ;key buffer variable
DATE        RMB 3           ;date memory
SAMPLE      RMB 3           ;count the number of samplings
TIME        RMB 3           ;time memory
NUMBUF      RMB 4           ;number of buffer
BUFEND      RMB 1           ;end of buffer

            ORG $E000       ;the program strats at $E000

*******************************************************************************
*                                                                             *
* MAIN : INITIALIZE THE STACK POINTER AND STARTS THE PROGRAM AFTER           *
*        RESET.                                                               *
*                                                                             *
*******************************************************************************

MAIN        LDS  #$FF       ;initialize sp after every reset
            JSR  INIT       ;jump to the INIT subroutine
CHOICE      JSR  CHOOSE     ;jump to CHOOSE subroutine
            CMPA #'I'       ;check the keyboard input
            BEQ  CASE1      ;if 'I',branch to case1
            CMPA #'O'
            BEQ  CASE2      ;if 'O',branch to case2
            JSR  ERROR      ;jump to the ERROR subroutine
            BRA  CHOICE     ;branch to the CHOICE instruction
CASE1       JSR  INPUT      ;jump to the input subroutine
            BRA  DONE       ;branch to the done subroutine
CASE2       JSR  RESTORE    ;jump to the restore subroutine
            JSR  TST_S      ;jump to the test samping subroutine
            BEQ  CHOICE     ;if no sampling, branch to the choice
            JSR  OUTPUT     ;jump to the output subroutine
            JSR  OUTCRLF
            JSR  DSPLAY     ;jump to the dsplay subroutine
DSPRDY      TST  SCSR       ;test all the data has been downloaded
            BPL  DSPRDY     ;if not, branch to DSPRDY
DONE        LDAA #%00011000 ;turn-off the analog circuitry and the 232-driver
            STAA PORTA
HOLD        CLRA            ;the program stays at the stop mode
            TAP
            STOP
            BRA  HOLD

********************* END OF MAIN **********************************
```

```
**************************************************************************
*                                                                        *
* INIT - INITIALIZATION OF REGISTERS AND MESSAGES                        *
*                                                                        *
**************************************************************************

INIT        LDAA #$80
            STAA PACTL           ;set PORTA bit 7 (PA7) as an input
            STAA PORTA           ;initialize PA6 & PA5 equal to 0
            LDAA #$0C            ;enable COP, but disable on-chip EEPROM
            STAA CONFIG
            LDAA #%00100110      ;initialize normal expanded chip mode
            STAA HPRIO
ADINIT      LDAA #$90            ;initialize the A/D converter
            STAA OPTION
COMM        LDAA #%00110000       ;set the buad rate equal to 9600
            STAA BAUD
            CLR  SCCR1
            LDAA #%00001100      ;set both receiver and transmitter enable
            STAA SCCR2
            LDAA #$FF            ;initialize port D as output control lines
            STAA DDRD
            STAA PORTD

OTHERS      LDD  #DATA           ;load the DATA address into ACCD
            STD  DATAVEC                  ;store data from ACCD into DATAVEC
            STD  DSPADDR         ;store data from ACCD into DSPADDR
            CLR  BUFEND          ;clear the BUFFEND variable
            CLR  COUNT           ;clear the COUNT variable
            CLR  COL             ;clear the COL variable
            CLR  SIGNAL          ;clear the SIGNAL variable
            RTS

**************************************************************************
*                                                                        *
* CHOOSE - DISPLAY THE START MESSAGE AND ASK FOR A KEY INPUT             *
*                                                                        *
**************************************************************************

CHOOSE      JSR  OUTCRLF         ;jump to the OUTCRLF subroutine
            JSR  OUTCRLF
            LDX  #STATMSG        ;load the starting message into IX
            JSR  OUTSTRG         ;jump to the OUTSTRG subroutine
            JSR  OUTCRLF
            LDX  #ASKMSG         ;load the asking message into IX
            JSR  OUTCHAR
```

```
*****************************************************************************
*                                                                           *
* INPUT - INPUT MODE WHERE THE SYSTEM IS TAKING THE SAMPLES                 *
*                                                                           *
*****************************************************************************

INPUT       JSR  SETUP          ;jump to the SETUP subroutine
            LDAA PORTA
            ORA  #%00010000
            STAA PORTA          ;turn off the 232-serial port
            JSR  INITCLK        ;jump to the INITCLK subroutine
            JSR  BACKUP         ;jump to the BACKUP subroutine
MLOOP       CLRA                ;clear accumulator A
            TAP                 ;transfer from ACCA to CCR
            NOP
            STOP                ;stop mode
            JSR  INCR_S         ;jump to the INCR_S subroutine
            JSR  BACKUP
            LDAA SIGNAL         ;test the SIGNAL
            CMPA #$FF
            BNE  MLOOP          ;if equal branch to MLOOP

            LDAA #$B            ;turn off the clock
            STAA RTC
            LDAA #%10000111
            STAA RTC+1
            RTS

*****************************************************************************
*                                                                           *
* SETUP - PARAMETER SET UP ROUTINE                                          *
*                                                                           *
*****************************************************************************

SETUP       CLR  SAMPLE         ;clear the no. of samples
            CLR  SAMPLE+1
            CLR  SAMPLE+2
            LDX  #INTVMSG       ;point to the interval message
            JSR  OUTSTRG        ;display the message
            CLR  FLAG           ;clear the FLAG
            CLR  POS            ;clear the POS
            JSR  READ           ;jump to the READ subroutine
            BMI  WRONG1         ;if the input is negative, branch to the WRONG1
            STAB INTRVL         ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            LDAA #':'           ; the following instuctions store the input
            JSR  OUTCHAR        ; interval into the buffer variables in mm:ss
            INC  POS            ; form
            JSR  READ           ;
            BMI  WRONG1         ;
            STAB INTRVL+1       ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            JSR  OUTCRLF
            BRA  INPUT2         ;branch to the second input
```

```
WRONG1     JSR  ERROR         ;if not proper input format, display error message
           BRA  SETUP

INPUT2     LDX  #PERIMSG      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
           JSR  OUTSTRG       ;the following instructions display the period
           LDAA #1            ;message, and store the input into PERIOD memory
           STAA FLAG          ;variable. The flag is set to 1.
           JSR  READ
           BMI  WRONG2
           STAB PERIOD
           JSR  OUTCRLF
           BRA  INPUT3
WRONG2     JSR  ERROR         ;if not proper input format, display error message
           BRA  INPUT2

INPUT3     LDX  #STARTTM      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
           JSR  OUTSTRG       ;the following instructions display the time
           LDAA #2            ;message, and store the input into TIME memory
           STAA FLAG          ;variable. The flag is set to 2.
           CLR  POS
           JSR  READ
           BMI  WRONG3
           STAB TIME
           LDAA #':'
           JSR  OUTCHAR
           INC  POS
           JSR  READ
           BMI  WRONG3
           STAB TIME+1
           CLR  TIME+2
           JSR  OUTCRLF
           BRA  INPUT4
WRONG3     JSR  ERROR         ;if not proper input format, display error message
           BRA  INPUT3

INPUT4     LDX  #STARTDT      ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
           JSR  OUTSTRG       ;the following instructions display the date
           LDAA #3            ;message, and store the input into DATE memory
           STAA FLAG          ;variable. The flag is set to 3
           CLR  POS
           JSR  READ
           BMI  WRONG4
           STAB DATE
           LDAA #'/'
           JSR  OUTCHAR
           INC  POS
           JSR  READ
           BMI  WRONG4
           STAB DATE+1
           LDAA #'/'
           JSR  OUTCHAR
           INC  POS
```

```
             JSR  READ
             BMI  WRONG4
             STAB DATE+2
             JSR  OUTCRLF
             BRA  INPUT5
WRONG4       JSR  ERROR            ;if not proper input format, display error message
             BRA  INPUT4

INPUT5       LDX  #CHNLMSG         ;load the CHNLMSG into IX
             JSR  OUTSTRG          ;output string
             LDAA #4               ;Set the flag equals to 4
             STAA FLAG
             JSR  READ             ;read the input
             BPL  SAVE
WRONG5       JSR  ERROR            ;if not proper input format, display error message
             BRA  INPUT5
SAVE         DECB
             STAB CHANNEL
             JMP  OUTCRLF

ERROR        PSHX                  ;display the error message
             JSR  OUTCRLF
             LDX  #MSG2
             JSR  OUTSTRG
             JSR  OUTCRLF
             PULX
             RTS

******************************************************************************
*                                                                            *
*                                                                            *
* CLOCK INITIALIZATION -INITIALIZE THE REAL TIME CLOCK                       *
*
******************************************************************************
*
INITCLK      LDAA #$D              ;check the power steady
             STAA RTC
             LDAA RTC+1
             TSTA
             BPL  INITCLK          ;if not branch to INITCLK

             LDAA #$C              ;clear all the inerrupt flags
             STAA RTC
             LDAA RTC+1

             LDAA #$B              ;initialize the clock but not strated yet
             STAA RTC
             LDAA #%10000111
             STAA RTC+1

             LDAA #$A              ;set the time base frequency to 32.768kHz,
             STAA RTC              ;no periodic interrupt and no square wave
             LDAA #%00100000       ;signal generate
```

```
STAA RTC+1

LDAA #9
STAA RTC
LDAA DATE           ;load the starting year into the calender
STAA RTC+1

LDAA #8
STAA RTC
LDAA DATE+1         ;load the starting month into the calender
STAA RTC+1

LDAA #7
STAA RTC
LDAA DATE+2         ;load the starting date into the calender
STAA RTC+1

LDAA TIME           ;load the ALARM hour into the calender
STAA MEMORY+5       ;store the alarm hour into MEMORY+5

LDAA #4
STAA RTC
LDAA TIME           ;load the starting hour into the calender
STAA RTC+1

LDAA TIME+1         ;load the ALARM minute into the calender
STAA MEMORY+3       ;store the alarm min into MEMORY+3

LDAA #2
STAA RTC
LDAA TIME+1         ;load the starting minute into the calender
STAA RTC+1

LDAA #0             ;load the ALARM second into the calender
STAA MEMORY+1       ;store the alarm second into MEMORY+1

LDAA #0
STAA RTC
LDAA TIME+2
STAA RTC+1          ;set the starting second as 00

JSR  NXTALM         ;jump to the NXTALM subroutine

LDAA #$B            ;turn on the clock
STAA RTC
LDAA #%00100111
STAA RTC+1
RTS
```

```
**********************************************************************
*                                                                    *
* NXTALM ROUTINE - TO CALCULATE THE NEXT ALARM TIME                  *
*                                                                    *
**********************************************************************
*
NXTALM          PSHA
                PSHB

SET_S           LDAA MEMORY+1       ;calculate the next alarm second
                ADDA INTRVL+1
                CMPA #59            ;compare with 59
                BLS  SAVE_S         ;if less than 59, branch to SAVE_S
                SUBA #60            ;else; subtract 60
                INC  MEMORY+3       ;increase the alarm minute by 1
SAVE_S          STAA MEMORY+1       ;store the next alarm second
SET_M           LDAA MEMORY+3       ;calculate the next alarm minute
                ADDA INTRVL
                CMPA #59            ;compare with 59
                BLS  SAVE_M         ;if less than 59, branch to SAVE_M
                SUBA #60            ;else; subtract 60
                INC  MEMORY+5       ;increase the alarm hour by 1
SAVE_M          STAA MEMORY+3       ;calculate the next alarm hour
SET_H           LDAA MEMORY+5
                CMPA #23            ;compare with 23
                BLS  SAVE_H         ;if less than 23, branch to SAVE_H
                SUBA #24            ;subtract 24
                INC  COUNT          ;else; increase the COUNT by 1
SAVE_H          STAA MEMORY+5       ;store the next alarm hour

UDC             LDAA #$A            ;check the internal update cycle of the clock
                STAA RTC
                LDAA RTC+1
                TSTA
                BMI  UDC            ;if yes, branch to UDC

                LDAA #0             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                STAA RTC            ;the following instructions update the
                LDAA RTC+1          ;present time ,date and alarm time into all
                STAA MEMORY         ;the MEMORY buffer address

                LDAA #1
                STAA RTC
                LDAA MEMORY+1
                STAA RTC+1

                LDAA #2
                STAA RTC
                LDAA RTC+1
                STAA MEMORY+2

                LDAA #3
```

```
                STAA RTC
                LDAA MEMORY+3
                STAA RTC+1

                LDAA #4
                STAA RTC
                LDAA RTC+1
                STAA MEMORY+4

                LDAA #5
                STAA RTC
                LDAA MEMORY+5
                STAA RTC+1

                LDAA #7
                STAA RTC
                LDAA RTC+1
                STAA MEMORY+7

                LDAA #8
                STAA RTC
                LDAA RTC+1
                STAA MEMORY+8

                LDAA #9
                STAA RTC
                LDAA RTC+1
                STAA MEMORY+9

                PULB
                PULA
                RTS

*******************************************************************************
*                                                                             *
* OUTPUT - DISPLAY THE START DATE, START TIME AND STOP DATE AND               *
*       STOP TIME ON THE SCREEN                                               *
*                                                                             *
*******************************************************************************
*
OUTPUT          LDX  #STARTDT                   ;display the startdate in yy:mm:dd
                JSR  OUTSTRG
                LDAB DATE
                JSR  TIMDSY
                LDAA #'/'
                JSR  OUTCHAR
                LDAB DATE+1
                JSR  TIMDSY
                LDAA #'/'
                JSR  OUTCHAR
                LDAB DATE+2
                JSR  TIMDSY
```

```
        JSR  OUTCRLF
        LDX  #STARTTM          ;display the start time in hh:mm:ss
        JSR  OUTSTRG
        LDAB TIME
        JSR  TIMDSY
        LDAA #':'
        JSR  OUTCHAR
        LDAB TIME+1
        JSR  TIMDSY
        LDAA #':'
        JSR  OUTCHAR
        CLRB
        JSR  TIMDSY
        JSR  OUTCRLF
        LDX  #ENDATE           ;diplay the end date in yy:mm:dd
        JSR  OUTSTRG
        LDAB MEMORY+9
        JSR  TIMDSY
        LDAA #'/'
        JSR  OUTCHAR
        LDAB MEMORY+8
        JSR  TIMDSY
        LDAA #'/'
        JSR  OUTCHAR
        LDAB MEMORY+7
        JSR  TIMDSY
        JSR  OUTCRLF
        LDX  #ENDTIME          ;display the end time in hh:mm:ss
        JSR  OUTSTRG
        LDAB MEMORY+4
        JSR  TIMDSY
        LDAA #':'
        JSR  OUTCHAR
        LDAB MEMORY+2
        JSR  TIMDSY
        LDAA #':'
        JSR  OUTCHAR
        LDAB MEMORY
TIMDSY  TBA                    ;the following instructions will convert the
        CLR  TEN               ;hex values into decimal values
CALOOP  CMPA #$A
        BLO  DSP
        SUBA #$A
        TAB
        INC  TEN
        BRA  CALOOP
        RTS

DSP     LDAA TEN
        ADDA #$30
        JSR  OUTCHAR
        TBA
```

```
            ADDA #$30
            JSR  OUTCHAR
            RTS

*************************************************************************
*                                                                       *
* DSPLAY - DISPLAY THE VOLTAGE VALUES ON THE SCREEN                     *
*                                                                       *
*************************************************************************
*
DSPLAY      LDX  DSPADDR        ;load the DSPADDR into IX
OUTDATA     CLRB
DSPLOOP     LDAA 0,X            ;load the value from address DSPADDR into A
            PSHB
            PSHX
            LDY  #NUMBUF+3      ;index Y points to NUMBUF+3 position
            LDAB #STEP          ;load the STEP constant into ACCB
            MUL                 ;ACCA x ACCB
            ADDD #5             ;add 5 into ACCD
            LDX  #10            ;load 10 into IX
            IDIV                ;(ACCD)/(IX)
            XGDX                ;exchange values in ACCD and IX
REPEAT      LDX  #10            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            IDIV                ;the following instructions convert the hex analog
            ADDD #$30           ;data into decimal digital data
            STAB 0,Y
            XGDX
            DEY
            CPY  #NUMBUF        ;check the convertion
            BNE  REPEAT         ;if not finish, continue
            ADDD #$30
            STAB 0,Y
            LDX  #NUMBUF        ;display an four digits value
            JSR  OUTSTRG
            PULX
            PULB
            INX
            CPX  #$DFFF         ;compare the output data address with the lower boundary
            BLE  DTEST          ;if less than, branch to check the display format
            LDX  #DATA          ;if greater; go to the next section of the memory
            LDAA PORTA
            ADDA #$20
            STAA PORTA
            ANDA #%01100000
            BEQ  DSPQUIT        ;if all the sections have been displayed, display quit

DTEST       CMPB #15            ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
            BNE  INCR_B         ;the following instructions will display each sampling
INCR_B      INCB                ;data at one line
            CMPB CHANNEL
            BLE  DSPLOOP
AGAIN       JSR  DEC_S
```

```
                JSR  TST_S
                BEQ  DSPQUIT
                JMP  OUTDATA
DSPQUIT         JSR  OUTCRLF            ;display THE END to the monitor
                LDX  #END2
                JSR  OUTSTRG
                RTS

***********************************************************************
*                                                                     *
* SAMPLING - THIS BLOCK INCLUDE TST_S, DEC_S AND INC_S                 *
*                                                                     *
***********************************************************************
*
TST_S           TST  SAMPLE             ;test the SAMPLE
                BNE  QTST               ;if not equal to zero, branch to QTST
                TST  SAMPLE+1           ;test the SAMPLE+1
                BNE  QTST               ;if not equal to zero, branch to QTST
                TST  SAMPLE+2           ;test the SAMPLE+2
QTST                   RTS

DEC_S           PSHA
                LDAA SAMPLE             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                SUBA #1                 ;the following instructions will subtract one
                BCC  QDEC_S1            ;from the number of samples at each time
                STAA SAMPLE             ;
                LDAA SAMPLE+1
                SUBA #1
                BCC  QDEC_S2
                STAA SAMPLE+1
                DEC  SAMPLE+2
                BRA  QDEC_S
QDEC_S1         STAA SAMPLE
                BRA  QDEC_S
QDEC_S2         STAA SAMPLE+1
QDEC_S          PULA
                RTS

INCR_S          INC  SAMPLE             ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                BVC  QINCS              ; adding 1 to the number of samples at each time
                INC  SAMPLE+1
                BVC  QINCS
                INC  SAMPLE+2
QINCS           RTS
```

```
*************************************************************************
*                                                                       *
* BACKUP - BACKUP SUBROUTINE IS USED TO BACKUP ALL THE INPUT            *
*        PARAMETERS INTO THE SRAM OF THE RTC                            *
*                                                                       *
*************************************************************************
*
BACKUP       PSHA
             PSHB
             PSHX
             LDX  #$0                 ;start from address 0 in the MCU
             LDAA #$0E
BLLOP        STAA RTC
             LDAB $0,X
             STAB RTC+1
             INCA
             INX
             CPX  #VARSIZE
             BLO  BLLOP
             PULX
             PULB
             PULA
             RTS

*************************************************************************
*                                                                       *
* RESTORE - RESTORE SUBROUTINE IS USED TO RESTORE THE VALUES IN         *
*         SRAM OF THE RTC INTO THE VARIABLES                            *
*                                                                       *
*************************************************************************
*
RESTORE      PSHA
             PSHB
             PSHX
             LDX  #$0                 ;restore the input parameters into the RAM
             LDAA #$0E                ;of the MCU
BLOOP        STAA RTC
             LDAB RTC+1
             STAB $0,X
             INCA
             INX
             CPX  #VARSIZE
             BNE  BLOOP
             PULX
             PULB
             PULA
             RTS
```

```
*******************************************************************************
*                                                                             *
* INCHAR - INPUT TO A AND ECHO ONE CHAR. LOOPS UNTIL A CHAR IS                *
*       READ                                                                  *
*                                                                             *
*******************************************************************************

INCHAR       LDAA SCSR           ;read status register
             ANDA #%00100000
             BEQ  INCHAR         ;jump if rdrf = 0
             LDAA SCDR           ;read data register
             RTS

*******************************************************************************
*                                                                             *
* OUTCHAR - OUTPUTS CHAR IN A TO SCI                                          *
*                                                                             *
*******************************************************************************
*
OUTCHAR      TST  SCSR           ;test status register
             BPL  OUTCHAR
             STAA SCDR           ;read data register
             CMPA #' '           ;if nothing, quit
             BLO  OQUIT
CH_COL       PSHB
             INC  COL            ;increase column
             LDAB COL
             CMPB #80            ;compare column with 80
             BLT  CQUIT          ;if less, quit
             CLR  COL            ;else clear column
             JSR  OUTCRLF
CQUIT        PULB
OQUIT        RTS


*******************************************************************************
*                                                                             *
* OUTCRLF - OUTPUTS A CR AND A LF                                             *
*                                                                             *
*******************************************************************************
*
OUTCRLF      CLR  COL
             PSHA
             LDAA #CR
             JSR  OUTCHAR
             LDAA #LF
             JSR  OUTCHAR
             PULA
             RTS
```

```
*******************************************************************************
*                                                                             *
* OUTSTRG - OUTPUTS A STREAM OF CHARS STARTING AT ADDRESS                     *
*      LOADED IN X                                                            *

*******************************************************************************

OUTSTRG      PSHA
OLOOP        LDAA 0,X              ; read character into A
             JSR  OUTCHAR          ;output character
             INX
             TST  0,X
             BNE  OLOOP
             LDAA #$20
             JSR  OUTCHAR
             PULA
             RTS

*******************************************************************************
*                                                                             *
*  READ - CONVERTS USER INPUT TO DECIMAL VALUE                                *
*                                                                             *
*******************************************************************************

READ         PSHX
             PSHA
             LDX  #KEYBUF
RLOOP        JSR  INCHAR           ;checking for a carriage return
             CMPA #CR              ;if yes, branch to the KEYPROS
             BEQ  KEYPROS
             JSR  OUTCHAR
             CPX  #KEYBUF+3        ;Is the input  more than three digits
             BGE  RLOOP            ;if yes, prompt for another input
             CMPA #'0'             ;else checking the input format
             BLT  NAME
             CMPA #'9'
             BGT  NAME
             STAA 0,X
             INX
             BRA  RLOOP            ;if improper input, prompt for another input
NAME         JMP  SET
KEYPROS      CPX  #KEYBUF          ;if no input; display ERROR message
             BEQ  SET
             CPX  #KEYBUF+3        ;if the input is more than 2 digitals; display ERROR message
             BGE  SET
CONVERT      LDAB KEYBUF           ;converting the input values into decimal
             SUBB #'0'
             CPX  #KEYBUF+1
             BEQ  FIRST
             LDAA #10
             MUL
             ADDB KEYBUF+1
```

```
               SUBB #'0'
FIRST          LDAA FLAG          ;check for the  FLAG values
               BEQ  TSTINTV       ;if FLAG is 0, branch to TSTINVT
               CMPA #1            ;if FLAG is 1; branch to test TSTPERI
               BEQ  TSTPERI
               CMPA #2            ;if  FLAG is 2, branch to TSTPERI
               BEQ  TSTTIME
               CMPA #3            ;if FLAG is not 3; branch to TSTCHNL
               BNE  TSTCHNL

TSTDATE        TST  POS           ;for proper date input, it must be ranged between 1 to 31
               BEQ  RQUIT
               CMPB #1
               BLT  SET
               LDAA POS
               CMPA #1
               BEQ  MONTH
               CMPB #31
               BGT  SET
               BLE  RQUIT
MONTH          CMPB #12           ;for proper month input, it must be ranged between 1 to 12
               BGT  SET
               BLE  RQUIT

TSTINTV        TST  POS           ;for proper second input, it must be ranged between 1 to 59
               BEQ  INTVL_M
               CMPB #59
               BGT  SET
               BLE  RQUIT
INTVL_M        CMPB #59           ;for proper minute input, it must be ranged between 1 to 59
               BGT  SET
               BLE RQUIT

TSTPERI        CMPB #1            ;for proper TSTPERI input, it must be ranged between 1 to 14

               BLT  SET
               CMPB #14
               BGT  SET
               BLE  RQUIT

TSTTIME        TST  POS           ;for proper hour input, it must be ranged between 1 to 23

               BEQ  HOUR
               CMPB #59
               BGT  SET
               BLE  RQUIT
HOUR           CMPB #23
               BGT  SET
               BLE  RQUIT

TSTCHNL        TSTB               for proper TSTCHNL input, it must be ranged between 1 to
16
```

```
                BEQ  SET
                CMPB #16
                BGT  SET
                BRA  RQUIT
SET             LDAB #$FF
RQUIT           TSTB
                PULA
                PULX
                RTS

**********************************************************************
*                                                                    *
*  DELAY SUBROUTINE                                                  *
*                                                                    *
**********************************************************************

DELAY           DEY
                BNE  DELAY
                RTS

**********************************************************************
*                                                                    *
*  INTERRUPT SERVICE SUBROUTINE                                      *
*                                                                    *
**********************************************************************

ISR             LDAA #$D            ;test the power ready of the RTC
                STAA RTC
                LDAA RTC+1
                TSTA
                BPL  ISR

CLRIRQ          LDAA #$C            ;clear all the interrupt flags
                STAA RTC
                LDAA RTC+1
                TSTA
                BMI  CLRIRQ

                LDAA SIGNAL         ;test the SIGNAL for memory full or time up
                CMPA #$FF
                BNE  CONT           ;if SIGNAL equals 0, branch to CONT
                RTI                 ;else; return from interrupt subroutine

CONT            LDAA COUNT          ;load the COUNT value into ACCA
                CMPA PERIOD         ;compare with the PERIOD value
                BNE  NXT            ;if not equal, branch to NXT
                LDAA MEMORY+4
                CMPA TIME           ;compare the final hour with next interval hour
                BNE  NXT            ;if not equal, branch to NXT
                LDAA MEMORY+2
                CMPA TIME+1         ;compare the final minute with next interval minute
```

```
            BHS NXT            ;if the final minute is greater, branch to NXT
            LDAA #$FF
            STAA SIGNAL        ;else increase the SIGNAL by 1
            RTI                ;return from interrupt subroutine

NXT         JSR  NXTALM        ;jump to the NXTALM subroutine

            LDY  #$FFFF
            JSR  DELAY         ;delay for AD converter set up

**********************************************************************************
*                                                                                *
* ADCONV - CONVERT THE ANALOG INPUT SIGNAL INTO DIGITAL AND STORE                 *
*      IT INTO DATAVEC                                                           *
*                                                                                *
**********************************************************************************

ADCONV      LDX  DATAVEC       ;IX points to the DATAVEC
            LDAB CHANNEL       ;load the no. of channels into ACCB
            CLR  SELECT        ;clear the SELECT
            LDAA #%11111101    ;turns off the analog circuitry
            STAA PORTD
ADLOOP1     LDAA PORTA
            ANDA #%11110000
            ORAA SELECT
            STAA PORTA
            SEC                ;set the carry
            ROL  PORTD         ;rotate left of PORTD
            LDY  #ADDELAY      ;load the value from ADDELAY into IY
            JSR  DELAY         ;jump to the DELAY subroutine
            ANDA #%00000100
            BNE  PE47          ;if PD2 equals to 1, branch to PE47
PE03        LDAA #$10
            BRA  ADST          ;branch to ADST
PE47        LDAA #$14
ADST        STAA ADCTL
ADCHK1      TST  ADCTL         ;check the A/D convert ready
            BPL  ADCHK1
            STAA ADCTL
            LDAA #3
            CMPB #3
            BHI  STO_TMP
            TBA
STO_TMP     STAA TMP
            LDY  #ADR1         ;load the values from address ADR1 into IY
ADCHK2      TST  ADCTL         ;check the A/D ready
            BPL  ADCHK2
ADLOOP2     LDAA 0,Y
            STAA 0,X
            INX
            INY
            CPX  #$DFFF        ;compare values in IX with $DFFF
```

```
          BHI  NXT32K       ;if greater than $DFFF; use next 32K memory
ADLOOP3   DEC  TMP          ;decrese TMP by 1
          BPL  ADLOOP2      ;if TMP is positive, branch to ADLOOP2
          SUBB #4           ;subtract ACCB by 4
          BMI  ADQUIT       ;if ACCB is negative, branch to ADQUIT
          LDAA SELECT       ;load SELECT values into ACCA
          ADDA #4           ;add 4 into ACCA
          STAA SELECT       ;store values in ACCA into SELECT
          BRA  ADLOOP1      ;branch to ADLOOP1
ADQUIT    STX  DATAVEC      ;load DATAVEC value into IX
          LDAA #$FF
          STAA PORTD
QUIT      RTI               ;return from interrupt

NXT32K    LDX  #DATA        ;load DATA value into IX
          LDAA PORTA
          ADDA #$20         ;add $20 to PORTA
          STAA PORTA
          ANDA #%01100000
          BNE  ADLOOP3      ;if PA5 & PA6 are not both 1, branch to ADQUIT
MFULL     LDAA #$FF
          STAA SIGNAL       ;else, increase the SIGNAL by 1
          BRA  ADQUIT       ;branch to ADQUIT

*********************** END OF THE PROGRAM ***********************

* Define constant *

STATMSG   FCC  'PRESS ENTER AFTER KEY IN ANY NUMBER'
          FCB  0
ASKMSG    FCC  'I = input parameters; O = output data :'
          FCB  0
INTVMSG   FCC  'SAMPLING INTERVAL IN (mm:ss) : '
          FCB  0
PERIMSG   FCC  'SAMPLING PERIOD IN (1-14) DAYS : '
          FCB  0
STARTTM   FCC  'START TIME (hh:mm) : '
          FCB  0
STARTDT   FCC  'START DATE (yy/mm/dd) : '
          FCB  0
CHNLMSG   FCC  'HOW MANY CHANNELS (1-16) ? '
          FCB  0
ENDTIME   FCC  'STOP TIME (hh:mm) : '
          FCB  0
ENDATE    FCC  'STOP DATE (yy/mm/dd) : '
          FCB  0
END2      FCC  'THE END'
          FCB  0
MSG2      FCC  'ERROR'
          FCB  0
```

```
            ORG  $FFF2
ISRVEC      FDB  ISR            ;interrupt service routine vector
            ORG  $FFFE
RESET       FDB  MAIN           ;reset to main
```