Automation of a capacitance dilatometer using distributed control

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: [https://dspace.lboro.ac.uk/2134/13975](https://dspace.lboro.ac.uk/2134/13975)

Publisher: © K.J. Britton

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Automation Of A Capacitance Dilatometer Using Distributed Control

by

Karen Joanne Britton

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of the Loughborough University of Technology

September 1990

© by K.J. Britton 1990
for my parents
ACKNOWLEDGEMENTS

My thanks go to Professor Mike Brown for his advice and help whilst acting as my supervisor, for his continued support after leaving Loughborough University Physics Department, and for reviewing and commenting on my manuscript.

I am indebted to the Intel Corporation for providing the hardware and software design tools used on this project, and for allowing me to attend a number of Intel training courses free of charge.

I would like to express my gratitude to the following people who have helped me during the course of this research.

Bill Gabb
Bryan Dennis
John Oakley
Raj and Ramesh Pancholi
Frank Hately
Mark Snape
Steve Done
Tim Timmins
Vicky Hardman
Sharon Aylett
Rebecca Bridges

A very special 'thank you' goes to my father, Len Britton, for the many hours he has spent drawing diagrams for me.

Finally, I would like to thank my boyfriend, Jonathan Powner, for his patience and support whilst I have been writing up, and for giving me the confidence to believe in myself.
# CONTENTS

## ABSTRACT

## CHAPTER ONE - Introduction

1.1 Aim of the thesis

1.2 Explanation for the choice of method used to measure thermal expansion: Comparison of the three-terminal capacitance transducer with other experimental techniques

1.3 A summary of the recent developments in the field of micro-electronics

1.4 Structure of the thesis

## CHAPTER TWO - Theory Review

2.1 A synopsis of the thermodynamic theory of thermal expansion

2.2 Theory of the guard ring capacitance transducer

2.3 The effect of non-parallelism and surface damage on practical capacitance displacement transducers

## CHAPTER THREE - Experimental Instrumentation

3.1 Description of the cryogenic equipment

3.1.1 Operation of the cryostat

3.1.2 The design and construction of the vacuum system

3.1.3 Commissioning the instrument
3.2 Measurement of thermal expansion

3.2.1 Overview and block diagram of the measurement and control instrumentation

3.2.2 The form of the capacitance cell

3.2.3 Signal detection and filtering

3.2.4 Automation of the arms on the capacitance bridge:

3.2.4.1 Designing the mechanisation

3.2.4.2 Designing the electronic instrumentation

3.2.5 Measurement of sample temperature

3.2.6 System temperature stability

3.3 The heart of the automatic microelectronic control system

CHAPTER FOUR - System Design: The main principles

4.1 Distributed control using microcontrollers: Justification for choice of design approach

4.2 Major features of the MCS-51 microcontroller

4.3 Software development environment

4.3.1 Software programming using PL/M51

4.3.2 Series IV microcomputer development system

4.3.3 The process of program development

4.4 System automation: Overview of the multiprocessor control unit
CHAPTER FIVE - System Design: The slave control modules

5.1 The off-balance capacitance control module (OBCCON) 55

5.1.1 Circuit diagrams 55

5.1.2 Overview of the functions of OBCCON 55

5.1.3 Details of the important module tasks 58

5.1.4 Problems encountered: Diagnostic tests and module emulation 61

5.1.5 The calibration software 64

5.2 The capacitance reader and motor control module (MOTCON) 67

5.2.1 Circuit diagrams 67

5.2.2 Overview of the functions of MOTCON 67

5.2.3 Details of important module tasks 70

5.2.4 Problems encountered: Diagnostic tests and module emulation 72

5.2.5 The calibration software 75

5.3 The sensitivity control module (SENCON) 77

5.3.1 Circuit diagrams 77

5.3.2 Overview of the functions of SENCON 77

5.3.3 Details of the important module tasks 83

5.3.4 Problems encountered: Diagnostic tests and module emulation 87

5.3.5 The calibration software 89
CHAPTER SIX - System Design: The Master Control Module and the Interface to the Operator via the IBM PC

6.1 The master control module (MASCON) 91

6.1.1 Circuit diagrams 91

6.1.2 Overview of the functions of the master module 91

6.1.3 Details of important module tasks 94

6.1.4 Problems encountered: Diagnostic tests and module emulation 99

6.1.5 The calibration software 106

6.2 System communication to the operator 108

6.2.1 Hardware link to the IBM PC from the micro-system 108

6.2.2 Software for the execution of operator commands 111

6.2.3 Measurement of temperature via the IEEE interface 114

6.2.4 Data storage and analysis 115

CHAPTER SEVEN - Experimental Results and Discussion

7.1 Initial Experiments Using Epoxy Resin 117

7.1.1 Method 118

7.1.2 Results 118

7.1.3 Discussion 120
7.2 Polydiacetylene signal crystals

7.2.1 Method

7.2.2 Results

7.2.3 Discussion

7.3 Amorphous Materials

7.3.1 Theory of thermal expansion and structure of glass

7.3.2 Experimental Details

7.3.3 Lead Silicate Results

7.3.4 Vitreous Silica Results

7.3.5 Discussion

CHAPTER EIGHT - Discussion and Further Work

8.1 Advantages and disadvantages of the automation system

8.2 Discussion of system expansion

8.2.1 Using the 7 Tesla superconducting magnet

8.2.2 Optical radiation of samples within the cryostat

8.2.3 Incorporation of a He³-He⁴ dilution refrigerator

8.3 Final conclusions

APPENDIX - A: Calibration tables

B: Publication

REFERENCES
ABSTRACT

A comprehensive and novel system of automatic control has been designed and constructed using distributed intelligence techniques to control a complex item of equipment for the measurement of linear thermal expansion over the temperature range 1.5K - 300K. The system is designed to perform its own self-calibration automatically before the experiment commences.

The low temperature dilatometer developed for this research project incorporates the most sensitive length change sensor available; a three-terminal capacitance transducer. This transducer technique has been refined to resolve length changes of $10^{-5}$A and more importantly, measure length against temperature profiles to better than $10^{-2}$A. The period of time required to collect a full set of data measurements on any particular specimen was in excess of 100 hours.

Forming the heart of the automated control system is an assembly of Intel MCS-51 single chip microcontrollers connected together on a serial link consisting of a simple pair of wires. The approach has been to divide the system automation into a number of specific control tasks and to allocate a different task to each controller.

The instrument has been used as a diagnostic tool to investigate the properties of lead glasses, and in particular to study the possibility of negative thermal expansion existing at low temperatures. Thermal expansion measurements were also performed to observe the phase change within single crystals of polydiacetylene and to research into low temperature phenomena occurring within the crystal.
CHAPTER ONE - Introduction

1.1 Aim of the Thesis

The aim of the thesis is to demonstrate the application of a novel technique of distributed control, to automate a three-terminal capacitance dilatometer.

The distributed control system comprises a number of single-chip microcontrollers, which communicate via a serial link. The three-terminal capacitance dilatometer provides one of the most sensitive techniques for the measurement of linear thermal expansion.

The aim of the research was to investigate the possibilities, and practicalities, of combining high precision dilatometry, with state-of-the-art microtechnology.

The three-terminal capacitance dilatometer has been used to measure the linear thermal expansion of amorphous, and crystalline materials, from liquid helium to room temperature.

1.2 Explanation for the Choice of Method used to Measure Thermal Expansion: Comparison of the Three-Terminal Capacitance Transducer with other Experimental Techniques

Two comprehensive reviews of dilatometry techniques have been written by A R Khan\(^1\), (1982), and I J Brown\(^2\), (1982). It has therefore been considered sufficient to give a summary of their reviews. In addition to this summary an outline of the more recent changes in dilatometry techniques has been included to cover the period from 1982 - 1989.

The coefficient of linear thermal expansion is given by:

\[
\alpha = \frac{1}{T} \cdot \frac{\Delta I}{\Delta T}
\]

Where: \(\Delta T\) = change in temperature, \(\Delta I\) = change in length
At room temperature the $\alpha$ for most solids is of the order of $10^{-5} K^{-1}$. Then for a $\Delta T$ of, for example, 1K, in order to obtain $\alpha$ to within 1%, it is necessary to have a resolution in the measurement of $\Delta l/l$ of about $10^{-7}$. However, at lower temperatures $\alpha$ may be very much smaller, and so require a far more sensitive dilatometer to obtain a comparable accuracy. The $\alpha$ of materials at a temperature of about $\Theta_d/40$, ($\Theta_d$ is the Debye temperature), is of the order of $10^{-8} K^{-1}$, and therefore to obtain an accuracy of about 1%, $\Delta l/l$ must be determined to a resolution of $10^{-10}$. Thus for a sample of, for example, 50mm the measuring technique must be capable of detecting changes in length of $\Delta l \approx 5 \times 10^{-2} \AA$.

It will be appreciated that detection levels of $10^{-1} - 10^{-2} \AA$ are very much smaller than the average inter-atomic distances. This requires that the measured changes, $\Delta l$, be interpreted as the surface average of the microscopic changes, $\sigma l$, over the surface being measured. A consequence of this is that some hysteresis could be expected on thermal cycling, when the measurement depends on the mechanical linkage between surfaces of dissimilar materials.

Some of the main techniques used to measure thermal expansion are given below.

The x-ray diffraction technique measures changes in the lattice parameter, $a$, to determine the thermal expansion of a crystal. This technique usually has a resolution of $\Delta a/a \approx 10^{-5}$. Batchelder and Simmon³, (1965), have achieved a resolution of $\Delta a/a = 5 \times 10^{-6}$ using oscillating back reflection s. This resolution is not sensitive enough for the determination of $\alpha$ at low temperatures ($T < \Theta_d/10$).

Optical interferometry provides a well known, high precision technique for measuring $\alpha$. Variations of the Fizeau⁴,⁵ interferometer (1964-66) have achieved a resolution of $\Delta l/l \approx 10^{-7} - 10^{-8}$, (practical detection limit: $10^{-2} - 10^{-3}$ of a fringe). Using a stabilised laser with the Fabry-Perot multiple beam interferometer a precision of $\Delta l/l \approx 10^{-9}$ can be obtained (Jacobs⁵ et al, 1970). Precision is limited by laser suitability, and by the deformation produced when the $\alpha$ of the end plates of the sample cell are not matched in $\alpha$ to the sample.
Optical levers and amplifiers have been reviewed by Jones\textsuperscript{7}, (1961), where he specifies detection of a linear displacement of $< 1 \times 10^{-2}$\AA. However, this technique suffers from drift and hysteresis on thermal cycling, due to the mechanical linkage coupling the sample to the optical lever.

Electrical inductance dilatometers have been constructed by Sparks and Swenson\textsuperscript{8}, (1967), with a sensitivity of $1 \times 10^{-2}$\AA, but the more sensitive devices have been found to suffer from magnetic problems, (White and Collins\textsuperscript{9}, 1972).

The sensitivity of the two-terminal capacitance transducer is of the order of $\Delta l/l = 10^{-7}$. Its sensitivity is limited by the fact that the lead capacitance is included in the measured capacitance. This transducer uses an LC resonance circuit and monitors the change in resonant frequency to detect the change in capacitance. At low temperatures the resolution can be improved to $2 \times 10^{-1}$\AA by using tunnel diode oscillators (Tolkachev\textsuperscript{10}, 1975).

The three-terminal capacitance technique has become the most popular technique for measurement of $\alpha$ at low temperatures. This dilatometer is based upon a technology developed largely by Thompson\textsuperscript{11}, (1958), and employs a transformer ratio bridge for the measurement of capacitance. The main advantages of the three-terminal system are:

(1) The balance condition on the bridge is not affected by lead capacitance.

(11) Capacitance bridges were commercially available which when used with a lock-in amplifier, have sensitivities of better than $10^{-7}$pF, corresponding to a sensitivity in the picometer range (White and Collins\textsuperscript{9}, 1972, and Kroegar and Swenson\textsuperscript{12}, 1977).

In recent years the technique has reached a resolution of $1$ in $10^{10}$, and was successfully used for the detection of paramagnetic impurities in crystals, (Brown and Brown\textsuperscript{13}, 1981). The three-terminal capacitance dilatometer has been used for
this study. It utilises the same experimental technique and copper electrode system as developed and tested so successfully by Brown and Brown$^{13}$.

Ackerman and Anderson$^{14}$, (1982) developed a dilatometer for the measurement of $\alpha$ in the range $0.1 - 10K$. The dilatometer used a SQUID detector and reached a resolution of $2 \times 10^{-4}$. The sample was rigidly restrained at one end, and its temperature was varied periodically with time. The sample length varied with temperature. This in turn caused the position of a superconducting coil to vary relative to a permanent magnetic field. The magnetic flux change through the coil was detected by the SQUID detector. The output from the SQUID was fed to a digital signal averaging computer from which $\Delta l$ was obtained. The two major limitations of this technique were:

(1) Vibrations transmitted from the environment to the cryostat.

(2) A sensitivity to magnetic fields intrinsic to the SQUID system.

However Anderson$^{15}$, (1986, private communication), sees little advantage of the SQUID system over the three-terminal capacitance system, and would have initially utilised the latter technique if the highly sensitive commercial bridge had been still available.

A new method of stress free dilatometry has been developed by Neuhäuser$^{16}$ et al, (1985), to examine the structural relaxation of metallic glass samples in ribbon form. Sensitive photometric measurements are made of the width of a slit, machined into a reference plate of known material. The reference plate is partially covered by the end of the specimen ribbon which is clamped at the other end. The change in the width of the slit is a measure of the relative length changes of specimen and reference material. The resolution achieved at present is $\Delta l/l \leq 5 \times 10^{-6}$.

Advances in x-ray dilatometer techniques$^{17}$ have made possible the determination of $\alpha$ for each of the phases in single crystal
polycrystalline material, thus enabling first and second order phase transitions to be distinguished.

A dilatometer with capacitance sensor developed for the measurement of α of thin films has achieved a relative sensitivity of $5 \times 10^{-9}$ in the temperature range 300K - 1000K (Bashirov et al., 1989).

An extremely accurate and sensitive extensometer has been developed by Karlak et al., (1989), which utilises a doppler laser interferometer to measure minute thermally induced dimensional changes.

It can be seen that the three-terminal capacitance dilatometer leads the field as an extremely sensitive technique for the measurement of α at low temperature. Electronic capacitance bridges are still commercially available at a high cost, but they do not match the sensitivity of the highest precision manual capacitance bridges which were available in the early 1980's.

In FIG 1.1 a three-terminal capacitor is shown along with its equivalent circuit. The shield electrode (G) completely surrounds the other two electrodes forming a local earth shield. The direct capacitance is then the only capacitance that is defined, that is, which does not involve the lead capacitances: $C_{HG}$ and $C_{LG}$.

The three-terminal capacitor is connected across a transformer ratio bridge such that the ground capacitances, $C_{HC}$ and $C_{LG}$, only shunt the ratio arms and the bridge detector. This results in only slightly reducing the system sensitivity, with negligible effect on the detector balance condition. This is a very simple approach to the operation of the three-terminal capacitance dilatometer utilising a transformer ratio bridge. No capacitor is ideal, inductance and dissipation effects must be taken into account. The detailed operation and limitations of this technique are covered in depth by I J Brown.

5
1.3 A Summary of the Recent Developments in the Field of Microelectronics

The first microprocessor was created by the Intel Corporation in 1971. It had a 4-bit architecture; that is, its data bus and address bus, and registers were 4 bits wide. In just two decades microelectronic technology has undergone a major change. The latest microprocessors have over one million transistors on a single chip of silicon, and support a 64-bit architecture.

Microprocessor design has branched into two distinct channels:

The first is that of the general purpose microprocessor, where computing power and number crunching capabilities are of crucial importance, for example the Intel 8086 family of processors, where the 80386 has become the industry standard. This microprocessor family now provides the computing engine for 70% of the personal computer market.

The second channel is that of the embedded control processor, or microcontroller. These microcontrollers are designed to be programmed to manage specific tasks. For embedded controllers the move is towards increased functionality, with such features as a built in A/D converter, high speed I/O sub-system, on-chip data and program memory, timer/counters, and interrupt handling.

In recent years performance increases have come from two types of design enhancement:

(i) New fabrication technology and smaller circuit geometries have made possible faster system clock rates (40MHz).

(ii) The arithmetic and logic unit (ALU), originally used to enable sequential execution of mathematical instructions, has been replaced. The replacement for the ALU comes in the form of a series of dedicated logic blocks, which process the various steps of multiple instructions simultaneously. This type of "assembly-line" circuitry is known as a pipeline, and results in a reduction of the average number of clocks per instruction.
There are fundamental limitations in developing microchip technology: the speed at which an electron can propagate through the medium, and the finite width of path along which an electron can travel. If the path width is too narrow it is possible for the electrons to diffuse from one path to another. The latest design technology uses the micron CHMOS-IV process (complementary high-speed metal oxide semiconductor). There are also limitations to increasing the speed of the system clock, since the memory system cost and circuit board complexity increase disproportionately with speed.

The state-of-the-art achievements in the microelectronics industry can be appreciated by reviewing the latest microprocessors to be unveiled by the Intel Corporation.

The 1486 microprocessor was released in April, 1989, and is an upward compatible enhancement of the 8088/86 and 1386 families. This is the architecture most widely accepted in the personal computer market. It provides the highest performance available for an installed base of standard applications programs and operating systems, for the business, financial and engineering sectors. The 1486 address bus, data bus and internal registers are all 32-bits wide, enabling two words of information to be received/transmitted in one step. The processor will run at 25MHz and 33MHz clock frequencies, and carry out the execution of frequently used instructions in one clock. Processor performance is greatly improved by the use of 8K of internal data and program cache. In systems with cache memory, all data is stored in the main memory, and some selected data is duplicated in a small fast cache memory. If the desired data is in the cache, the processor can access it quickly, (a cache hit). If not, it must be fetched from main memory, (a cache miss). The average memory access time is reduced, if the cache system is organised so that the code and the data needed most frequently by the processor are in the cache. The 1486 also has an on-chip floating point unit for the execution of complex mathematical computations.

The 1860, announced in February 1989, was the world's first 64-bit general purpose microprocessor. This processor has set a new performance standard by breaking the one instruction per clock
barrier. This high performance CPU can execute up to three operations per clock by simultaneously utilizing the integer, floating-point adder, and multiplier units. Incorporated in the 1860 is a RISC (reduced instruction set computing) integer core containing about 35 instructions, most of which can be executed in a single clock cycle. The RISC core fetches both integer and floating point instructions. It executes all load, store, integer, bit and control transfer instructions. The parallel architecture has a flexible structure. This makes the 1860 processor suitable for a broad range of parallel algorithms, including traditional vector processing. The volume of raw computing power makes the 1860 able to support demanding applications, such as modelling and simulations. Integration of a 3-D graphics unit enables the processor to support interactive visualization of the results. The 1860 is optimized for performance driven applications in minicomputers, workstations, and scientific computing.

Lastly there is the 1960 family of processors (first released April 1988). The architecture of the 80960 is tuned for the specialised needs of real-time control and other embedded computing applications. It combines the best of two computer design philosophies: RISC and CISC (complex instruction set computing). The device is the highest performance processor yet developed for embedded control applications and data processing. Applications range from office automation to industrial robotics and avionics. On-chip integration includes: a built-in programmable interrupt controller with 32 priority levels, a 4 channel high speed I/O processing unit which operates in parallel with the main CPU, internal high speed data RAM, and instruction cache.

The 80960CA 32-bit processor was the world's first microprocessor to implement "superscalar" program execution design techniques. This is a technique that results in multiple instructions per clock cycle. A superscalar processor looks several instructions ahead in an instruction stream. It identifies groups of instructions that do not conflict with each other, and then begins to execute each of the instructions simultaneously.

The key to superscalar execution is to provide multiple implementations of each critical processing resource, and to
coordinate them in such a way that each stays as busy as possible. To do this the i960CA contains three full execution pipelines, each independent of the others, and each dedicated to a different class of instruction:

(i) Register to register arithmetic operations.

(ii) Coordination of all memory transfer operations.

(iii) Program control instructions (including branching and conditionals).

Over the years the actual process of designing a chip has changed dramatically. In the past it was necessary to fabricate a sample and test it to determine whether a new component was designed correctly. If errors were detected they were fixed, and the chip re-fabricated and tested again. It was often necessary to repeat this costly cycle many times. With current advanced technology it is no longer necessary to fabricate the device for the preliminary tests. Instead a model is created and a computer simulation is carried out to ensure the design is correct. The use of a computer model has made it possible to produce a sample that works perfectly the first time it is fabricated.

1.4 Structure of the Thesis

The thesis incorporates three main sections:

The first section comprises Chapter 1, 2 and 3. Chapter 1 and Chapter 2 cover the background and theory of the capacitance dilatometer technique, and recent developments in the field of microelectronics. Chapter 3 describes the development of the experimental instrumentation required for the measurement of linear thermal expansion from 1.5 - 300K.

The second section, Chapters 4, 5 and 6, comprises a detailed specification of the design for the distributed control system. Automation of the capacitance dilatometer is accomplished by an assembly of single chip microcontrollers which communicate via a
serial link. Chapter 4 covers the main system design principles. Chapter 5 and Chapter 6 give an indepth profile of each control module in the system. Chapter 6 also describes the use of an IBM PC to provide the interface between the control system and the experimental operator.

The third section, Chapters 7 and 8, describes the use of the three-terminal capacitance dilatometer from liquid helium temperatures to room temperature. Chapter 7 reviews the results obtained for the linear thermal expansion of the following materials:

(i) Epoxy resin, a flexible material which may be used to help form the support structure for superconducting magnets, (the work was sponsored by Oxford Instruments).

(ii) Single crystal polydiacetylene, a potential molecular electronic device material, (the work was carried out in collaboration with GEC Research Laboratories).

(iii) Amorphous materials, investigating the possibility of negative thermal expansion at low temperatures, (samples provided by Dr. Greenough, Department of Applied Physics, Hull University).

The final chapter, Chapter 8, discusses the advantages and disadvantages of the automated capacitance dilatometer system, the possibilities for future system expansion, and further experimental work.
THE THREE TERMINAL CAPACITOR (a) AND ITS EQUIVALENT CIRCUIT (b)
2.1 A Synopsis of the Thermodynamic Theory of Thermal Expansion

In an attempt to understand the physical properties of most solids, the model used is of a single crystal with the atoms arranged on an infinite lattice. Each atom or ion in a crystal is confined in a potential well due to the electrostatic forces of its neighbours, (FIG 2.1). The dependency of potential energy, \( U \), on the atomic separation, \( r \), cannot be calculated analytically except for a very few structures. More commonly a Taylor expansion of \( U \) in terms of \( r \) is used, and the potential energy is usually written:

\[
U(r - r_0) = C(r - r_0)^2 - D(r - r_0)^3 - E(r - r_0)^4 + \ldots \quad (1)
\]

At any temperature the atoms vibrate about their mean, or equilibrium position, \( r_0 \). There can be no linear term, \( B(r - r_0) \), since this would imply a constant force acting on each atom, in which case \( r_0 \) could not be the equilibrium co-ordinate.

Since 'C' is by far the largest of the coefficients in most materials, the approximation frequently used in solid state physics, is to set the coefficients \( D \) and above explicitly to zero. Thus, 'the restoring force acting on an atom is assumed to be directly proportional to its displacement from equilibrium'. This is the Harmonic Approximation.

In studying solids, great simplification can often be achieved by making the measurements at low temperatures. In general terms, the reason for this is that as the temperature approaches more closely to absolute zero, so do the properties of the real solid more closely agree with those of the ideal models on which their description is based. Hence, at low temperatures the Harmonic Approximation has greatest validity.

At higher temperatures the cubic \( (D) \) and the quadratic \( (E) \) term give rise to various anharmonic phenomena. The phenomena of particular interest to this work is that of thermal expansion. That is to say, if the lattice vibrations were purely harmonic in nature there would
be no thermal expansion. Another important anharmonic phenomena is phonon-phonon scattering. Thus, the study of thermal expansion can be used as a method of investigating the anharmonic forces acting in a crystal.

Thermal expansion is understood to be any dimensional change in temperature. Thermal expansion is characterised by the volumetric thermal expansion coefficient $\beta$, defined algebraically as:

$$\beta = \frac{1}{V} \left[ \frac{dV}{dT} \right]_p - \left[ \frac{d\ln V}{dT} \right]_p$$

(2)

$V$ - volume  $p$ - pressure  $T$ - temperature

However, within experimental work it is usually the linear expansivity, $\alpha$, that is measured. If the three rectangular dimensions of a solid are $L_1, L_2$, and $L_3$ then:

$$V = L_1 L_2 L_3$$

$$\frac{dV}{dT} = L_2 L_3 \frac{dL_1}{dT} + L_1 L_3 \frac{dL_2}{dT} + L_1 L_2 \frac{dL_3}{dT}$$

$$\frac{1}{V} \frac{dV}{dT} = \frac{1}{L_1} \frac{dL_1}{dT} + \frac{1}{L_2} \frac{dL_2}{dT} + \frac{1}{L_3} \frac{dL_3}{dT}$$

$$\beta = \alpha_1 + \alpha_2 + \alpha_3$$

If the solid is isotropic, as in the case of a cubic crystal then:

$$\alpha_1 = \alpha_2 = \alpha_3 \quad \longrightarrow \quad \beta = 3\alpha$$

The third law of thermodynamics can be stated as:

'The entropy of all systems and of all states of a system is zero at absolute zero.'
That is to say:

'The entropy difference \( \Delta S \) between different states of such a system at the same temperature must tend to zero as \( T \), the temperature tends to zero.'

Now Gibbs free energy is defined as:

\[
G = U - TS + pV
\]

\[
dG = dU - TdS - SdT + pdV + Vdp
\]

Where \( U \) - internal energy and \( S \) - entropy

Now \( dU = TdS - pdV \)

Therefore \( dG = -SdT + Vdp \)

So that (for an exact differential):

\[
\left[ \frac{dS}{dp} \right]_T - \left[ \frac{dV}{dT} \right]_p
\]

This is one of Maxwell's thermodynamic relations. The left hand side measures the rate of change of entropy with pressure at constant temperature. If the entropy difference due to a small isothermal pressure difference is to tend to zero as \( T \) tends to zero then:

\[
\left[ \frac{dS}{dp} \right] \text{ must tend to } 0 \text{ and } -\left[ \frac{dV}{dT} \right] \text{ must likewise vanish as } T \rightarrow 0
\]

Then from (2) it follows that \( \beta \) tends to zero as \( T \) tends to zero.

One deduction from the third law is that the thermal expansion coefficient of liquids and solids must vanish at the lowest temperatures, this is confirmed by experiment.

At the lowest temperatures the atoms stay close to the symmetric minima of their potential wells. As their thermal energy increases,
because of the anharmonic asymmetry of the wells, they spend on average more time at greater separations. An estimation can be made of the increase $\bar{x}$, in separation. Assuming a Boltzmann distribution for the lattice vibrations and restricting consideration to one dimension:

$$\bar{x} = \frac{+\infty \int_{-\infty}^{+\infty} x e^{-U(x)/kT} \, dx}{+\infty \int_{-\infty}^{+\infty} e^{-U(x)/kT} \, dx}$$

The Boltzmann distribution weights the possible values of $x$ according to their thermodynamic probability. Where $U(x)$ is from equation (1), with $x$ written for $(r - r_0)$, and $k$ is the Boltzmann constant.

For displacements, such that the anharmonic terms in the energy are small in comparison with $kT$, we may expand the integrand as:

$$\int_{-\infty}^{+\infty} x e^{-U(x)/kT} \, dx = \int_{-\infty}^{+\infty} e^{-C\alpha x^2 / kT} \left[ x + \frac{x^4 D}{kT} + \frac{x^6 E}{kT} \right] \, dx \approx \int_{-\infty}^{+\infty} e^{-C\alpha x^2 / kT} \, dx = \left[ \frac{3\pi^{1/2}}{4} \right] \cdot \left[ \frac{D}{6\gamma^2} \right] \cdot (kT)^{3/2}$$

$$\int_{-\infty}^{+\infty} e^{-U(x)/kT} \, dx = \int_{-\infty}^{+\infty} e^{-C\alpha x^2 / kT} \, dx = \left[ \frac{\pi kT}{C} \right]^{1/2}$$

Giving:

$$\bar{x} = \frac{3D}{4C^2} \cdot kT$$

Where $C$ and $D$ are coefficients in the expansion of (1).
The isothermal compressibility $\chi_T$ is defined as:

$$\chi_T = -\left[\frac{d\ln V}{dp}\right]_T$$

Also the thermal expansion can be written as:

$$\beta = \left[\frac{d\ln V}{dp}\right]_T \cdot \left[\frac{dp}{dT}\right]_V$$

Therefore

$$\frac{\beta}{\chi_T} = -\left[\frac{dp}{dT}\right]_V = -\frac{d^2 F}{dVdT} \quad (3)$$

$F$ is Helmholtz free energy.

Equation 3 relates the thermal expansion to the change in free energy of the system. The dimensional changes are such as to minimise the free energy. The Helmholtz free energy is an additive function of state and as such equation (3) includes contributions from the following:

- static lattice, lattice vibrations, conduction electrons, electric dipoles, magnetic ions, nuclear spins, and so forth.

The main concern of this work being the contribution due to lattice vibrations.

From thermodynamics the Helmholtz free energy of a system is given by:

$$F = U - TS$$

Where $U$ is the internal energy and $S$ is the entropy.

$$dF = -SdT - pdV$$

$$\left[\frac{dF}{dV}\right] = -p$$
Now $U$ and $S$ are related through the thermodynamic function:

\[ dU = TdS \quad \text{at constant } V \]

\[ \left[ \frac{dU}{dT} \right]_V = T \left[ \frac{dS}{dT} \right]_V \]

Therefore:

\[ p = -\frac{d}{dV} [U - TS]_T \]

\[ -\frac{d}{dV} \left[ U - T \int_0^T \left[ \frac{dU}{dT} \right]_V dT \right]_T \]

It can be shown for a system of harmonic oscillators that the internal energy is given by:

\[ U = U_{eq} + \frac{3N}{2} \sum_{i=1}^{3N} \hbar \omega_i + \sum_{i=1}^{3N} \hbar \omega_i \cdot \left( e^{\frac{\hbar \omega_i}{kT}} - 1 \right)^{-1} \]  

The first term is the energy due to the static lattice, the second term is the zero point energy of $3N$ three-dimensional simple harmonic oscillators. The third term is the product of three parts:

(i) $3N$ - number of oscillators in a 3-D crystal

(ii) $\hbar \omega_i$ is the energy of the phonon associated with the $i$th mode

(iii) The probability of the $i$th mode being occupied -

\[ f(\omega_i) = \left( e^{\frac{\hbar \omega_i}{kT}} - 1 \right)^{-1} \]

The third term is the temperature dependent term of the lattice vibrations.
Substituting for $U$ in the expression for $p$ and differentiating with respect to $T$:

$$\frac{dp}{dT} = - \sum_{i=0}^{3N} \left[ - \frac{d\omega_i}{dV} \cdot \frac{d}{dT} \left( e^{\frac{\hbar\omega_i}{kT}} - 1 \right)^{-1} \right]$$

(6)

Now since:

$$\frac{\beta}{\chi T} = \left[ \frac{dp}{dT} \right]_V$$

It follows that:

$$\frac{\beta}{\chi T} = - \sum_{i=0}^{3N} \left[ \frac{d\omega_i}{dV} \cdot \frac{d}{dT} \left( e^{\frac{\hbar\omega_i}{kT}} - 1 \right)^{-1} \right]$$

(7)

The coefficient of thermal expansion, $\beta$, is more often written in terms of a quantity $\gamma$, called the Grüneisen constant, which characterises the way the lattice vibrational frequencies, $\omega$, change with volume $V$. Defining the Grüneisen parameter $\gamma_i$ of the $i$th mode:

$$\gamma_i = - \frac{d\omega_i}{dV} \cdot \frac{V}{\omega_i}$$

Let $x = \frac{\hbar\omega_i}{kT}$ then substituting in (7):

$$\frac{\beta}{\chi T} = k \sum_{i=0}^{3N} \left[ \frac{1}{V} \cdot \frac{x^2 e^x}{(e^x - 1)^2} \gamma_i \right]$$

Now let the contribution of the $i$th mode to the specific heat, $c_i$, be

$$c_i = k \sum_{i=0}^{3N} \left[ \frac{x^2 e^x}{(e^x - 1)^2} \right]$$

$$\frac{\beta}{\chi T} = \frac{1}{V} \frac{\sum \gamma_i c_i}{\sum c_i}$$
\[ \beta \frac{\gamma}{\chi_T} = \frac{c_V}{V} \quad (8) \]

Where \( \gamma = \frac{\Sigma \gamma_1 c_1}{\Sigma c_1} \) and \( c_V = \Sigma c_1 \)

(8) is known as the Grüneisen relationship which is found to be experimentally true for many solids over a wide range above about 200K.

The manner in which \( \gamma \) varies with temperature will give us an indication as to how the frequency components of the lattice vibrations vary with temperature. For most frequencies the value, \( \omega_1 \), decreases as the volume is increased, so the \( \gamma_1 \) and hence \( \gamma \) are positive; thus giving a positive thermal expansion, \( \beta \). Values for the low frequency modes of the individual \( \gamma_1 \) can be obtained, for isotopic materials, from the measured pressure derivatives of the elastic stiffness constants.

The microscopic theory of thermal expansion derives from the work of Mie\(^1\), (1903), and Grüneisen\(^2\), (1912). Grüneisen, using a quantum Einstein model characterised by a single vibrational frequency \( \omega \), derived from interatomic forces, identified the Grüneisen parameter as:

\[ \gamma = - \frac{d \ln \omega}{d \ln V} \]

The Grüneisen constant characterises the anharmonicity of the crystal lattice potential. For a purely harmonic crystal \( \gamma = 0 \). Since little is known about the anharmonic contributions in a particular crystal lattice, the thermal expansion is usually studied using the 'Quasi-Harmonic Model'. This model treats the lattice vibrations as purely harmonic, but assumes the frequencies of vibration to be volume dependent. In this approximation the lattice vibrational entropy is the sum of the separate contributions (Leibfried and Ludwig\(^3\), 1961, and Baron and Klein\(^4\), 1962).

The thermal expansion phenomenon can provide us with information about those properties of a solid which cannot be obtained by studying specific heat. Measurements of thermal expansion give
direct access to the Grüneisen constant. A study of this constant gives a direct measure of the anharmonicity of vibrations of atoms in a solid. Using equation (8) \( \gamma \) can be calculated from a knowledge of the experimentally measured quantities: \( \beta, C_V, \chi_T \), and \( V \).

2.2 Theory of the Kelvin Guard Ring Capacitance Transducer

In its simplest form the equation for an ideal parallel plate capacitor is given as:

\[
C = \frac{A \varepsilon}{g} \tag{9}
\]

- \( C \) is the capacitance between the two plates
- \( A \) is the area of one of the plates
- \( g \) is the separation (gap) between the plates
- \( \varepsilon \) is the permittivity of the medium between the plates

This equation assumes a uniform electric field between the plates in a direction perpendicular to the plate surface. For an ideal capacitor, assuming an infinite plate geometry, this would be true. However, for a practical capacitor, account must be taken of fringing effects at the edges of the plates. At the edges of the plates there is an increase in charge density which leads to 'bowing' or 'lengthening' of the lines of force, (FIG 2.2). Consequently the electric field between the plates is non-uniform.

The effect of fringing can be minimised by using a three-terminal Kelvin Guard Ring capacitor. This capacitor consists of an extensive flat electrode separated by a small distance from a second smaller circular electrode. The smaller electrode is surrounded by a coplanar extensive flat electrode which forms the guard ring, (FIG 2.3). Even with a guard ring there is still some distortion of the electric field near the edges of the plates, and it is necessary to make some correction to equation (9). In essence this correction increases the effective area of the central electrode from \( A \) to \( A + \Delta A \). Where \( \Delta A \) is the area of an additional strip extending over half of the width, \( w \), of the gap between the central electrode and the guard ring. The correction to equation (9) was derived by Maxwell5.
\[ C = \frac{\pi r^2 \varepsilon}{g} \cdot (1 + \eta) \]  

(10)

\[ \eta = \frac{w}{r} \cdot (1 + \frac{w}{2r}) \cdot \left(1 + \frac{0.22w}{g}\right)^{-1} \]

and \( r \) is the radius of the central electrode.

The Maxwell equation (10) assumes that the space between the guard and the central electrode is a narrow deep gap without a dielectric filling (i.e. a vacuum). To improve the mechanical stability of a practical capacitance dilatometer the gap between the electrode and the guard ring may be filled with an epoxy such as stycast 2850FT. This has the effect of increasing the capacitance of the slit. Brown and Bulleid\(^7\), (1978), have suggested a first order correction to equation (10) to account for this difference by including the relative permittivity, \( \varepsilon_s \), of the stycast in the last term, giving the following relationship:

\[ C = \frac{\pi r^2 \varepsilon}{g} \cdot \left[ 1 + \frac{w}{r} \left(1 + \frac{w}{2r}\right) \cdot \left(1 + \frac{0.22w}{g\varepsilon_s}\right)^{-1} \right] \]  

(11)

2.3 The Effect of Non-Parallelism and Surface Damage on Practical Capacitance Displacement Transducers

The three-terminal capacitor has been used as a device to monitor displacements for the last 30 years: White\(^6\), (1961), and Collins and White\(^8\), (1964); investigating magnetic critical phenomena, Donaldson and Lanchester\(^9\), (1968). More recently the technique has been improved to achieve a sensitivity of \(10^{-10} m\). Brown and Brown\(^10\), (1981), used the technique for the detection of paramagnetic impurities in crystals. Heerens and Vermeulen\(^11\), (1975), have carried out work on the theoretical behaviour of guard ring capacitors with ideal geometry. Gladwin and Wolfe\(^12\), (1975), carried out experimental measurements of the linearity of capacitance transducers under ideal conditions. However it was not until 1978 that a paper was produced detailing the behaviour of the device in a 'non-ideal' situation where the electrodes may not be quite
parallel, and where the surface of the electrodes may be damaged.

Brown and Bulleid⁷, (1978), used a Michelson interferometer to investigate the effects of surface damage and tilt on a capacitance transducer. FIG 2.4 shows the results for the surface damage experiment. The graph compares data for polished parallel plates with shot blasted parallel plates. Because the departures from linearity due to surface damage are very small the graph shows the gradient of the reciprocal capacitance, \( \frac{d(1/C)}{dg} \), with separation \( g \). For the perfect capacitor the gradient would remain constant. The shot blasting produced a visually rough surface with undulations measured at \( \pm12\mu m \). It can be seen that the effects due to surface damage are small, and indeed are only important for plate separations \(<100\mu m \). However, the effect of residual tilt on one of the plates has a marked effect on the linearity of the guard ring capacitor transducer. Khan¹⁰ et al, (1980), extended the work of Brown and Bulleld⁷, (1978), to include the detailed effect of tilt on the linearity of capacitance transducers. Their results have been summarized in FIG 2.5. Again the point to point gradients of the reciprocal capacitance against plate separation have been plotted. It can be seen that at electrode separations of less than 100\( \mu m \) there is a pronounced effect upon the linearity of these transducers:

- tilt = 5' produces a 10% error at 120\( \mu m \)
- tilt = 5' produces a 20% error at 100\( \mu m \)

(considering electrodes 25mm in diameter as used with the Michelson interferometer).

Equations (10) and (11) have also been plotted in FIG 2.5. It can be seen that both equations demonstrate good agreement with the experimental results for the range 150\( \mu m \) - 300\( \mu m \). It can therefore be seen that both these equations adequately describe the behaviour of the guard ring capacitor, and can confidently be used to calculate the electrode separation during an experiment.

For the capacitance transducer to be linear, and to be described by a simple equation such as (10), I J Brown¹³ concluded that a number of criteria must first be met.
(i) \[ \frac{dC}{dg} \sim \frac{1}{g^2} \]

Hence for high sensitivity the electrode separation \((g)\) must be as small as possible.

(ii) To minimise the effect of surface damage and tilt, assuming a 9mm electrode radius as used with the three terminal capacitance dilatometer.

Electrode separation > 200\(\mu\)m

(iii) The area of the inner electrode should be large to enable sufficient accuracy to be obtained using a bridge method of capacitance measurement.

(iv) \(r_0 \gg r_1\)

The radius of the extensive electrode should be large in comparison to the radius of the inner electrode.

(v) The slit between the electrode and the guard ring \(w\), should be narrow, as assumed by Maxwell in deriving (10). Decreasing \(w\) effectively increases the capacitance between the inner lower electrode and the guard ring (at earth potential). When using the transducer with a bridge detector method increasing this capacitance has the effect of shunting the bridge detector, thus decreasing the instruments sensitivity. Therefore there is a limit to which \(w\) can be decreased.

The following practical solution was found by I J Brown\(^8\) to be both linear and sufficiently sensitive:

- (r) Radius of the inner electrode \(\approx 9\)mm
- (r\(_0\)) Radius of extensive electrode \(\approx 13\)mm
- (\(\omega\)) Width of the gap, \(w\) \(\approx 0.05\)mm
- (g) Separation of the electrodes \(\approx 250 - 350\mu\)m

This electrode arrangement was kept throughout the current experiments, as detailed in Chapter 7 of this thesis.
(FIG. 2-1) THE POTENTIAL 'WELL' OCCUPIED BY AN ATOM IN A CRYSTAL, ILLUSTRATING THE MODELLING OF POTENTIAL ENERGY.
(FIG 2.2) LINES OF FORCE BETWEEN TWO CAPACITANCE PLATES

(FIG 2.3) CROSS-SECTION OF A KELVIN GUARD-RING CAPACITOR
(FIG. 2.4) EFFECT OF SURFACE DAMAGE ON THE LINEARITY OF CAPACITANCE TRANSDUCERS

(FIG. 2.5) EFFECT OF TILT ON THE LINEARITY OF CAPACITANCE TRANSDUCERS
3.1 Description of the Cryogenic Equipment

3.1.1 Operation of the Cryostat

A cryostat was designed and developed for use with a low temperature capacitance dilatometer. Oxford Instruments custom built the cryostat which was purchased with a grant from the Paul Instrument Fund. The cryostat resides in a strong metal cage, positioned on a two ton concrete block. This block sits on a hard core pillar (4m x 4m), which extends down to the bedrock, and is isolated from the rest of the building. In this way the cryostat was decoupled from mechanical vibrations due to its surroundings. Both the control electronics and the cryostat were placed on this plinth to encourage vibrational stability.

The cryostat consists of a dewar and variable temperature sample system, used in conjunction with a 7 Tesla superconducting solenoid, (FIG3.1). The magnet is supported from a top plate by three stainless steel rods, and is positioned vertically within the helium bath of the cryostat. A photograph of the top of the cryostat is shown in PHOTO 3.1.

The outer dewar of this cryostat has a 50 litre capacity. Liquid nitrogen, (boiling point = 77.4 K), is used as the coolant within the outer dewar. The inner dewar can be cooled using liquid helium, (boiling point = 4.2K), and will be referred to as the helium bath. To obtain the base temperature of 1.5K it is necessary to pump on the surface of the liquid helium to reduce its vapour pressure. Details of this will be given later in this section.

A variable temperature insert (VTI) with its own outer vacuum jacket resides within the helium bath, (FIG3.2). The VTI works on the principle of allowing a controlled flow of helium, from the helium bath, through a needle valve and into the VTI chamber. This space is pumpable via a large rotary pump (900 litres/minute). The helium bath is kept close to atmospheric pressure. A gas flow pump on the VTI drives the helium flow by creating a pressure difference between
the bath and the return line.

Temperature control can be achieved using a 330Ω heater, which is located at the bottom of the VTI; adjacent to the heater is a rhodium/iron (Rh/Fe) thermometer. For the temperature range 300K - 4.2K, coarse temperature control is achieved by setting the needle valve, and fine control is achieved by regulating the power to the heater on the VTI heat exchanger. The temperature range 4.2K - 1.5K is obtained by reducing the vapour pressure of the liquid helium after it has passed through the needle valve. Again the temperature can be controlled via the electronic feedback circuit.

The inner-most chamber of the cryostat consists of a complicated sample rod, (FIG 3.3). This demountable sample enclosure screws into a square cut thread in the base of the VTI. The thread ensures good thermal contact with the low temperature point of the insert and minimises vibrations. A calibrated Rh/Fe sensor and a capacitance sensor are mounted on the experimental flange. The sample temperature is controlled directly by a 330Ω heater on the sample cell. The electrical wiring for the thermometers, for the heater, and for experimental use runs up the sample cell pumping line to an electrical fitting at the top bung. The wires are thermally anchored at various points to minimise conduction to the vacuum cell. The sample access tube is 50mm in diameter.

As the system has a large thermal capacity there are two alternative control positions. The first control position, as mentioned above, consists of a calibrated 27Ω Rh/Fe sensor, a CS-400-GR capacitance sensor, and a 30Ω heater on the sample rod. These devices are mounted on the experimental flange to which the experimental apparatus is bolted, and therefore give a direct indication of sample temperature. The Rh/Fe sensor is sensitive to magnetic fields, but is very stable during thermal cycling. However the capacitance sensor is a sensitive thermometer with zero magnetic field effect, but there is some drift on thermal cycling. Therefore before commencing an experiment involving magnetic fields the capacitance sensor must be calibrated against the Rh/Fe sensor in zero magnetic field. The size of the magnetoresistive effect was not a consideration during this set of experiments to measure linear thermal expansion since the experiments were carried out in zero
magnetic field. The second control position comprises a second Rh/Fe sensor and 300 heater. The sensor and heater are located on the base flange of the VTI, in close proximity to the source of the cooling power, (the Joule/Thompson expansion at the entry of the capillary into the dynamic space). The associated 3120 control unit allows temperature control via either of the two Rh/Fe thermometers or the capacitance sensor.

Another feature of this cryostat consists of a set of mylar windows which enable the experimental sample to be optically irradiated from an external source. Care must be taken as mylar film is slightly porous to helium gas above 80K.

Above 4.2K the cryostat has two modes of operation:

GRAVITY FEED - This is the easier mode of operation, but the flow of liquid helium is limited, and therefore the cooling power is also limited. The insert and port are first evacuated with the needle valve closed, and then vented to an atmosphere of helium gas. This process is executed to minimise the chances of a blockage when the needle valve is first opened. The exhaust port is connected to a helium recovery system, and then the needle valve is opened to give the required flow and cooling power. Once the required temperature has been attained, the flow is restricted until the temperature settles at a value below the required level. The exact temperature value may then be set on the temperature controller, and the heater used to bring the insert to the required value.

PUMPED FEED - In this case the insert exhaust port is connected to a rotary pump with a minimum pumping speed of 450 litres/minute. The insert is evacuated with the needle valve closed, and then the needle valve is opened to give the required cooling power. The means of temperature control are the same for both modes of operation.

To obtain temperatures below 4.2K a rotary pump with a 900 litre/minute capacity is required. Temperatures below 4.2K are obtained by pumping against the needle valve, thereby reducing the vapour pressure and thus the helium temperature. Again there are two modes of operation - continuous and single shot. The continuous mode is as for above 4.2K except the needle valve is merely cracked open
with the pump running. Temperature is controlled via the temperature control unit. This mode provides a base temperature of 2.5K - 2.0K depending on the heat input from the sample and support rod. The single shot mode involves filling the insert pumping port with helium, closing the needle valve, and then pumping on the helium to reduce its vapour pressure, and thus reduce the temperature. This allows the ultimate base temperature of 1.5K to be reached. A detailed account of the cryostat operation can be found in reference 1.

3.1.2 The Design and Construction of the Vacuum System

The vacuum requirements for the new cryostat were carefully analysed, (FIG 3.4 and PHOTO 3.4). The vacuum system was designed and constructed to give maximum versatility. For example, should one pumping system fail, a second system can be brought into operation. The system was built with copper pipes, (diameter: 15mm and 28mm), which link to the cryostat via flexible stainless steel couplings. System valves consisted of a combination of butterfly and baffle valves. The cryostat could be totally isolated from the vacuum system using the main valves, which are located on the top flange of the cryostat. It was essential to isolate the cryostat when left overnight during an experimental run. An automatic alarm and pump switch-off procedure was also in operation should the diffusion pump temperature have exceeded 30°C.

A large helium-sealed 2-stage rotary pump (R3), capacity = 1333 litres/minute, provides the means to pump the VTI, and reduce the vapour pressure above the liquid helium. To achieve vacuums of the order of $10^{-5} - 10^{-6}$ mbar, the usual combination of an oil vapour diffusion pump, backed by a mechanical rotary pump, were utilised. Two such sub-systems were built: D1 backed by R1, D2 backed by R2. Condensable vapours produced by back-streaming were removed by using a cold liquid nitrogen trap with each diffusion pump. Baffle valves placed above the diffusion pumps also aided in the prevention of system contamination, due to back-streaming.

Pirani gauges (P1, P2, P13, P14) enabled measurement of backing and roughing pressures in the range 1atm - 10^{-3} mbar. For the measurement of low pressures in the range $10^{-2} - 10^{-7}$ mbar, Penning gauges (Pe1,
Pe2, Pe3) were used. A Budenberg meter, M1, gives an indication of the pressure above the helium bath (range: 0 - 96cmHg). The meter, M2, gives an indication of the pressure within the VTI, (range:0 - 1000mbar).

The high vacuum for the specimen cell could be achieved using pumps D2 and R2. The high vacuum chamber surrounding the VTI could be pumped using D1 and R1. Alternatively the pumping power of these two systems can be combined by opening valve 3. The vacuum chamber surrounding the main nitrogen bath is evacuated using these pumps. During an experiment the main bath is filled with liquid nitrogen, this causes the vacuum chamber surrounding the main bath to cryopump, giving an order of magnitude improvement on the vacuum pressure (10^{-4}mbar $\rightarrow$ 10^{-5}mbar). It is then possible to seal off this vacuum chamber. The helium bath may also be serviced by these pumps. This is of particular importance when flushing the system with helium gas. Helium gas can enter the helium bath by connecting a gas cylinder to the pressure vent and opening valve 11.

The sole purpose of rotary pump R3 is to pump the VTI to reduce the vapour pressure of liquid helium within the can, and hence cause the temperature to decrease. Helium can be bled into the VTI from the helium bath via a needle valve. Pump R3 may be by-passed by opening valve 20. Gas may be vented by opening valve 17 or collected using the helium recovery system (open valve 18.)

The helium recovery system consists of a set of four weather balloons, which recover gas boiling off from the helium bath and from the VTI (when under gravity feed). Two balloons are connected to the vacuum system to collect helium gas, whilst the other two are connected to the compressor which evacuates the balloons and compresses the helium into gas cylinders. The roles of the two sets of balloons are then reversed by the opening and closing of valves on the recovery unit. Care was taken to obtain balloons which offered little resistance to the expanding gas. Failure to do this would result in a build-up of pressure within the cryostat. The system boil-off rate may be monitored by using the flow meter. During cool-down it is necessary to maintain a slight over-pressure of helium gas in the VTI. Gas may be fed directly into the VTI from a cylinder, or from the balloons if they are first filled with
helium gas. The over-pressure is indicated on M2 and also by the inflation of a small rubber bladder connected to the exhaust port of the VTI.

3.1.3 Commissioning the Instrument

On cooling the cryostat to liquid nitrogen temperatures, gauge Pel indicated a major leak in the high vacuum chamber. Initial steps were taken to check that the mylar windows were correctly in place, and that the vacuum system itself was not the cause of the leak. The VTI was removed from the cryostat, the VTI vacuum chamber was evacuated, and a mass spectrometer was placed on the vacuum line. The VTI was then leak tested using a small cylinder of helium gas. A serious leak was discovered on the stainless steel collar of the VTI. An attempt was made to seal the leak using a silver-bearing 'soft solder'-type alloy and super flux. This held well at room temperatures, but the leak reappeared at low temperatures.

On the subsequent cooling of the cryostat a second severe gas leak was observed on the vacuum for the main bath. It was found that in this case the outer rubber O-ring on the top flange was not producing a good seal. The O-ring was not cracked, but its cross-sectional area was found to be inadequate for the groove in which it sat. The O-ring was replaced with a thicker one, and this cured the leak.

In order to repair the first leak it was necessary to return the whole cryostat to Oxford Instruments where the stainless steel collar on the VTI was re-welded.

Whilst repairing the cryostat a number of small changes were made to improve the system. In order to cool the cryostat the helium bath must initially be filled with liquid nitrogen. This liquid must then be flushed out of the top of the cryostat by pressurising the helium bath with clean helium gas. In order to do this a steel tube is inserted into the helium bath, and located in a small funnel towards the bottom of the bath. In practice it was found to be extremely difficult to locate this funnel. This problem was overcome by extending the funnel using a short length of stainless steel tube. The second change was the addition of a heater adjacent to the
needle valve, to prevent the needle valve from becoming blocked with frozen moisture. Power to the heater was controlled externally.

As liquid nitrogen evaporated from the system the exiting cold gas was causing the pipes and outer valves to freeze. To prevent the gas from causing a valve failure, right-angled sections of pipe were used to direct the gas away from the cryostat. Lastly, the copper pipe work was extended from the nitrogen filling vent on the cryostat to simplify the transfer of liquid nitrogen into the system.

3.2 Measurement of Thermal Expansion

3.2.1 Overview and Block Diagram of the Measurement and Control Instrumentation

To determine the thermal expansion of the sample using the three-terminal capacitance dilatometer it is necessary to monitor the change in capacitance of the sample cell and monitor the sample temperature. A diagram specifying the measurement and control instrumentation is shown in FIG 3.5. The diagram indicates the signal detection system, the bridge re-balancing system, and a number of purpose built interface circuits to enable automatic control of the instrumentation. At the centre of the automatic system is the multiprocessor control unit. To appreciate the functionality of the control unit it is necessary to understand how combinations of logical zeros and ones are utilised by the interface and feedback circuits to control the measurement instrumentation. The remainder of this chapter details the component parts of this block diagram.

3.2.2 The Form of the Capacitance Cell

The copper capacitance dilatometer cell was designed and constructed by M. Brown and I. Brown; full details can be found in reference 2. For this reason only a brief description shall be given here.

The cell is made of high-conductivity oxygen-free (HCOF) copper, for which the thermal expansion data is well documented. For data analysis the expansion data of Kroegar and Swendon\(^3\), (1977), has
been used; being expressed as a power series polynomial expansion in temperature. To obtain the actual expansion of the specimen, corrections are made for the radial expansion of the lower potential electrode and axial expansion of the annular cell, by using this data.

Cylindrical samples are best suited to the cell shown in FIG 3.6 where the sample expansion is measured relative to that of the HCOF copper. The dilatometer consists of an annular cell with copper plates bolted onto the top and bottom. The upper plate comprises the guard-ring of a Kelvin guard-ring capacitor. Located coplanar with this guard-ring is the low potential capacitance electrode (2). The gap between electrode (2) and the guard-ring has been filled with stycast 2850 FT (with catalyst 24LV) epoxy resin. Electrode (1), the high potential electrode, is placed as a snug "top-hat" on the sample. To minimise the effects of surface damage and tilt, the electrode separation of (1) and (2) should not be smaller than 200μm (assuming a 9mm electrode radius). This was determined by Brown and Bulleid⁴, (1978), in an investigation into the effect upon the linearity of this transducer using a Michelson interferometer. All the copper components were hand lapped and polished to give optically flat surfaces to improve parallelism between the electrodes.

The technique for mounting the specimen within the cell is shown in (FIG 3.7). The support consists of a flat copper plate and central thin walled cylinder. Half of the cylinder was detachable thereby forming a split ring. To secure the sample it was placed within the split ring, which was in turn clamped by a copper clip. To further increase the mechanical stability of the support, a small amount of GE varnish was placed around the lip of the split ring and sample. This arrangement was found to give reliable and reproducible results, without contributing any anomalous expansions to the measurement.

3.2.3 Signal Detection and Filtering

Measurement of the three-terminal capacitance of the sample cell is via a CR1616 precision bridge. The unknown capacitance is compared with variable standard capacitors using a variable ratio arm, see
FIG 3.8. This transformer ratio arm bridge can detect changes of 0.1aF ($10^{-19}$F) with an accuracy of a few parts per million. The core permeability is particularly unaffected by age and temperature. The flux can be confined to the core with very little flux leakage, such that the ratio of the open circuit voltage induced in the two secondary windings exactly equals the ratio of the number of turns ($N_s/N_w$). This ratio can be varied by a series of taps along the secondaries.

The GR1616 uses several fixed standard capacitors in combination with a single decade divider. For the simplified bridge shown in FIG 3.8 the balance condition can be given simply as:

$$\frac{C_X}{C_S} = \frac{V_s/V_X} = \frac{N_s}{N_X}$$

However, changes in temperature do affect the capacitance standards, so they are located within a thermally isolated box with a time-constant of 6 hours. The bridge precision is dependent upon the ambient temperature at which the bridge is maintained, and so the bridge should be stable at this temperature for 36 hours before attempting any measurements.

The bridge detection electronics were isolated inside a metal cabinet maintained at 24.5°C using a temperature controller (see section 3.2.6). For maximum shielding the sample cell is connected to the bridge by doubly screened cable giving separate earths $E_1$ and $E_2$. The mains earth supply is $E_2$ and $E_1$ is a specially constructed earth. The inner screens of the cables from the bridge, the phase sensitive detectors (PSDs), the coherent filters, the reference unit, and the low noise amplifier are connected together at one point, this is $E_1$. It consists of a 2m length of perforated copper tube buried in sand within a 12ft hole drilled into the bedrock. Good electrical conductance is ensured by frequent watering.

As the sample expands or contracts within the sample cell so the off-balance capacitance signal from the bridge changes. This voltage signal passes through a low noise amplifier, coherent filters and PSDs. The voltage is then sampled by an integrating 12 bit A/D converter (see 5.2). A separate Seromex power supply is used to power the detection instrumentation. A General Radio CO oscillator
1310B is used to energise the primary of the bridge transformer at a working frequency of 1.1KHz. The oscillator signal is amplified by a 100W amplifier before reaching the bridge. The original signal is also fed through a buffer to a reference unit. This reference unit was used to offset any phase shift produced by the amplifiers and bridge transformer circuitry. The unit also acts as a phase splitter producing two square wave signals in phase quadrature; one of these signals being in phase with the reference signal from the phase shifter. These two square wave signals provide reference signals for the PSDs.

Because of the extreme sensitivity the detection system is isolated within a Faraday cage. A second aluminium cage (5mm thick) surrounds the bridge to isolate it from audio frequency interference. To alleviate the problem of earth-loops, the bridge and detector were isolated from each other and from the mains earth, and it was ensured that the separate earthing for each instrument came together at only one point.

The low noise amplifier has a variable gain facility between 30dB and 100dB, and provides the inputs to the two phase detection networks. A resistor and relay interface circuit enables computer control and simulation of the different amplifier gains, (FIG 3.9). For sensitive low temperature measurements gains of 90dB are required, but for rapidly changing capacitance values it is necessary to decrease the system gain to prevent the need for constant bridge rebalancing.

Two detection networks are incorporated so that both the resistive and reactive components of the signal can be recovered separately. The coherent filters (built in-house), act as narrow band pass filters (bandwidth 10Hz - 2KHz) with fixed gains of 40dB. The output from each filter passes to a separate PSD. An external time-constant facility enables the PSD dealing with the capacitance (reactive) component to vary the time-constant between 15 and 90 seconds, (FIG 3.9). Again this facility can be computer controlled (see 5.3), and is necessary to deal with noise spikes (use a long time-constant) or to have a quick response time (use a short time-constant).
As indicated the reference unit provides a reference signal to the filters and PSDs, and in practice the adjustment of the reference signal phase is significant to the operation of the detection system. The aim is to tune the two detection networks to the resistive and reactive components of the bridge output voltage signal by adjusting the phase. This is achieved by off-setting the bridge by say 1fF and adjusting the phase shifter to give a maximum deflection on the PSD monitoring the reactive part of the signal. The process is then repeated with decreasing capacitance offsets at higher amplifier gains. A maximum signal on the capacitive component corresponds to a minimum signal on the resistive component. When in tune a change in the bridge capacitance will have little or no effect on the resistive part of the signal. Likewise changing the bridge conductance should not affect the reactive part of the signal. This can be used as a test to see that the phase has been adjusted correctly.

3.2.4 Automation of the Arms on the Capacitance Bridge

3.2.4.1 Designing the Mechanisation

The capacitance standards inside the capacitance bridge may be set and changed manually by moving the lever arms at the front of the bridge. The actual capacitance set is reflected by a digital display on the bridge. Each arm controls one capacitance decade, for example: the 1 fF arm can change the capacitance by -1fF through to +10fF. This is shown by a single display digit changing from -1, 0, 1, 2 ... through to 10 (or X). To automate the process of rebalancing the capacitance bridge it is necessary to achieve mechanical control of these lever arms.

Mechanical control was achieved through a series of drive rods and bevel gears connected to stepper motors, (PHOTO 3.10). Most sample capacitances are of the order of picofarads, and for highly accurate work control of the 1aF arm is required. Hence 9 of the arms were mechanised covering the range 1aF to 100pF.

The lever arms were located along the bottom of the 30cm high capacitance bridge, at a distance of 20mm apart, and their direction of travel was along an arc of a circle. These three factors
increased the complexity of the mechanisation task. Stainless steel rods (45cm), which were threaded over part of their length, connected between the bridge arms and an array of bevel gears located in an aluminium plate at the top of the bridge. The rods were secured at the bottom by a 1cm thick aluminium base plate. This plate helped considerably in keeping the rods aligned. It was also necessary to stagger the rods and gears in two stages to alleviate the problem of lack of space between the lever arms. PHOTO 3.11 shows the capacitance bridge and associated electronics.

Each pair of bevel gears were inter-locked at right-angles. The outer diameter of the gear was 1.035\"; these were the smallest gears of this type available. A larger gear would have given smoother control, but lack of space between the arms prohibited this. The gears work on a 1:1 ratio and each steel gear has 40 teeth.

A stepper motor was used to turn each pair of gears. The shaft of each motor was connected to the gear by a steel rod and brass coupling. The stepper motors (manufactured by Astrosyn, type: 23PM-C301) have a step angle of 1.8 degrees and provide a holding torque of 4Kg-cm. Movement of the gears via the motors enabled the steel rods to turn on their own axis. A short threaded brass tube provided the coupling between the brass lever arm and the threaded steel rod, (FIG 3.12 a). A 1cm slit in the brass lever arm, and a screw threaded through the brass coupling and the brass lever arm, gave flexibility to this link. This flexibility enabled the lever arm to move freely on the slightly curved path over the whole range of its travel. As the rod rotates clockwise so the brass coupling travels up the rod carrying the lever arm with it, and conversely, if the motor causes the rod to rotate anti-clockwise the brass coupling travels down the rod and brings the lever arm down, (FIG 3.12 b).

The capacitance value set on the bridge can be read directly from the digital display, or electrically via a 50-way binary coded decimal (b.c.d) output at the back of the bridge. The capacitance bridge can be rebalanced to a new capacitance setting by controlling the movement of the right motors in the selected direction, and by monitoring the changing capacitance value via the 50-way b.c.d output. Although the stepper motors provide accurate and powerful
control of the arms it was not possible to equate a change of 1 capacitance digit with a specific number of steps of the motor shaft. The reason for this, as mentioned earlier, is that the movement between the digits is not linear; the arms move on a slight arc. In addition this method would not allow for any slippage within the gears when moving between digits.

A rubber wedge was placed between the base plate on which the motors and gears were secured and the top of the rack containing the bridge and additional instrumentation. The rubber wedge helped to damp down vibrations between the bridge and the motors.

3.2.4.2 Designing the Electronic Instrumentation

To enable the arms on the capacitance bridge to be computer controlled an interface circuit was designed and built (FIG 3.13) to drive and control the stepper motors. This Auto/Manual Bridge Interface circuit (A.M.B.I.) allowed for both manual and computer control of the arms on the capacitance bridge.

The control system requires high performance stepper motors capable of high torques and fast stepping rates. By utilising a stepper motor controller integrated circuit (IC), employing electronic logic and solid state switching devices, this can be achieved. The IC used for this application is the SAA1027 (IC3). A feature of this device is that it requires only clock, direction and mode input signals. The phase sequence required to drive the stepper motors is generated inside the IC and output on pins 6, 8, 9, 11 thus simplifying the microprocessor's control burden. The clock input (pin/5) is derived from two monostable chips (74121, IC1 and IC2). By varying the potentiometers, VR1 and VR2, the input pulse frequency can be changed between 90Hz and 1KHz. Each clock pulse causes the motor to step once (1.8°). Two hundred pulses are required for one revolution. By experiment it has been determined that on average five revolutions of the motor shaft are required to move the lever arm on the bridge by 1 placement digit. It follows that for a pulse rate of 1KHz, 200 steps/revolution, the bridge arm would cause a change of 1 digit/second. For a pulse rate of 90Hz there would be a digit change approximately every 11 seconds. The potentiometers were set to give a rate of 1 digit change every 2 seconds.
A manual wafer switch on the front panel of A.M.B.I. selects between manual and computer control of the bridge arms. Pins 2 and 3 of the drive chip select whether to start/stop the motor, or move the arms up or down. With the wafer switch selecting computer control, two binary inputs from the microelectronic computer system control pins 2 and 3. Alternatively high/low inputs are obtained using relays 10 and 11, and manual push-button control.

The IC provides the correct sequence of pulses to energise the motor windings and cause the shaft to rotate. The sequence of pulses from the drive chip, switch on and off four power transistors (T1 to T4) which drive the windings of the stepper motor. Power resistors are used to decrease the electrical time-constant of the circuit (L/R) and thus improve motor performance at higher stepping rates.

\[
\text{Total inductance/phase} = L \quad \text{Total resistance/phase} = R = R_1 + R_M
\]

- \(R_1\) - series resistance
- \(R_M\) - motor winding resistance/phase = 0.33Ω

If \(R_1 = 0\) :

\[
\frac{R_M}{R_M + R_1} = 1 \quad \rightarrow \quad \text{Time-Constant} = \frac{L}{R}
\]

If \(R_1 = R_M\) :

\[
\frac{R_M}{R_M + R_1} = \frac{1}{2} \quad \rightarrow \quad \text{Time-Constant} = \frac{L}{2R} \quad \text{(improved torque)}
\]

Now \(R_1\) is limited by the equation:

\[
R_1 = \frac{(V_S - V_M)}{I}
\]

- \(V_S\) - supply voltage = 17V
- \(V_M\) - rated motor voltage = 1.3V
- \(I\) - rated current/phase = 3.9 A

Hence:

\[
R_1 = \frac{(17 - 1.3)}{3.9} \approx 4Ω
\]

Thus:

\[
\frac{R_M}{R_M + R_1} = \frac{0.33}{4.33} \approx \frac{1}{13} \quad \rightarrow \quad \text{Time-Constant} = \frac{L}{13R}
\]

The actual \(R_1\) used was 4.4Ω (2 x 2.2Ω, 25W power resistors) in series with each power transistor emitter. To prevent a large voltage drop across the power transistors when the motors are off relay 12 has been included. When the relay is de-energised 17 Volts appear across the 22Ω, 50 W power resistor, R24, instead of across the power transistors T1 to T4.
Only one motor is activated at one time, therefore a set of nine 4-pole relays have been employed to switch in any one of the nine motors. These relays must be capable of withstanding 4 amps of current. A light emitting diode (l.e.d) is associated with each relay to indicate which motor has been engaged. The l.e.d.s can be viewed on the front panel of A.M.B.I. With the wafer switch in manual position, the select switch can activate any one of the nine relays. Having set the wafer switch to computer control, four binary inputs applied to the 74154 (IC8) decoder chip can be used to select any one of the nine motors. The relays (R1 to R9) are turned on by a logic 0, the outputs of the decoder chip are therefore fed into 7404 (IC6, IC7) inverter ICs to provide the active low signal. Darlington drivers (8247, IC5 and IC4) act as the drive buffers for the relays.

Three supply voltages are required to power the A.M.B.I. circuit. A +5 Volt supply is used by most of the logic chips in the circuit; however a +12 Volt supply is needed for the stepper motor drive chip, the darlington drivers and the relays. Using a +12 Volt, 1 Amp transformer and two voltage regulators these two requirements can be met. The stepper motors themselves run off a +17 Volt power supply, and in this case a more substantial 100 VA, 8 Amp torroidal transformer has been used with a 25 Amp bridge rectifier. Power was supplied to the motors by three substantial 12 core screened cables which could withstand several amps.

The A.M.B.I. Instrument has been mounted within the rack of electronics containing the capacitance bridge. To dissipate heat, the power resistors were attached to the perforated lid of A.M.B.I. and a small fan was also incorporated inside the instrument.

3.2.5 Measurement of Sample Temperature

The sample temperature can be measured accurately from 1.5K - 100K using a germanium resistance thermometer (type: CR1000, No. 4034, Cal. No. CCC731). The thermometer is screwed to the base of the copper capacitance cell to give good thermal contact. This thermometer requires a range of currents to be supplied from 1µA at 1.5K to 1mA at 100K. These currents are stipulated by the manufacturer to ensure that the thermometer acts in a calibrated manner, and are necessary to prevent excessive heating at low temperatures.
The currents are provided by a constant current source, (FIG 3.14). This current source has been built in-house, and currents may be set manually or by the computer system. A +9 Volt lead-acid accumulator has been used to improve current stability. The electronic circuit has been carefully designed to provide a constant current of \( I \pm 1\% \). Using the current source the polarity can be reversed to enable the voltage reading to be thermally averaged.

The voltage drop across the germanium sensor is measured using a Keithley nanovoltmeter. This measured voltage is returned to the computer system via an IEEE interface. Knowing both the current through the thermometer, and the voltage drop across the thermometer, the resistance can be calculated. Then using calibration data the temperature of the sensor can be evaluated.

The other temperature sensors within the cryostat have been outlined in section 3.1.1. One Rh/Fe sensor and one capacitance sensor are located at the bottom of the sample rod, and adjacent to them is a 30Ω heater. A second Rh/Fe sensor and associated 30Ω heater are positioned at the bottom of the VTI. All three sensors can measure temperature in the range 1.5K - 300K. Using the given calibration (Appendix A) an overall accuracy of ±0.01K below 27K and ±0.02K up to 400K can be obtained for the Rh/Fe thermometers. The capacitance sensor is calibrated against the Rh/Fe thermometer resulting in an accuracy of ~0.03K.

A 3120 temperature controller, purchased from Oxford Instruments, can interface with any one of these three sensors. The instrument is designed to measure and display the system temperature, and can maintain the system at some required temperature by controlling the power to the heater. The 3120 temperature controller compares the measured temperature with the set temperature, and generates an error signal proportional to the temperature difference. This error signal is used as the input to a Three-Term Controller. The Controller regulates the power sent to the heater, and works using proportional, integral and derivative action. The 3120 temperature controller can be controlled manually, using the front panel, or automatically by the computer system. As with the Keithley nanovoltmeter the 3120 temperature controller communicates to the computer system via an IEEE interface. Under steady state conditions
the function of the heater is to compensate exactly for the heat loss from the system. Control can be accomplished easily for an ideal uniform system: uniform heat loss, uniform heat distribution and small thermal mass sensor. In reality all systems have to adopt a compromise. For this cryostat, the heater and sensor located on the bottom of the sample rod are in close proximity, resulting in rapid system response at the expense of accuracy. In practice it was found that the controller could maintain the measured temperature to within ±0.02K of the set temperature.

To summarize, the sample temperature can be measured using:

1. Germanium resistance thermometer, range: 1.5K - 100K. This sensor gives the most precise reading being located on the capacitance cell itself. By carrying out cubic spline interpolation using GINO graphics routines, in conjunction with the given calibration table for the Germanium thermometer (Appendix A), an overall accuracy of ±40mK can be obtained.

2. Rhodium/Iron resistance thermometer, range 1.5K - 300K. The accuracy of the calibration data is given as ±3mK below 30K and ±5mK above 30K (see Appendix A). This sensor is positioned on the sample rod.

3. Capacitance thermometer, range 1.5K - 300K. This sensor is calibrated against the Rh/Fe thermometer positioned on the sample rod, and is utilised in the presence of a magnetic field.

3.2.6 System Temperature Stability

The bulk of the interfacing electronics is located in one rack positioned on the "vibration free" area of the laboratory. Within this rack is the capacitance bridge and oscillator, the signal detection system, the stepper motor drive unit (A.M.B.I.) and the system power supply. For accurate capacitance measurements it is imperative that the capacitance bridge be kept at a constant temperature, and that the bridge be stable at this temperature for a period of at least 36 hours prior to commencing the experiment.
System temperature stability has been achieved using a proportional temperature controller (PTC) and associated fan heater. The ramp voltage on the PTC is set to achieve the required system temperature by means of a potentiometer. If the controlling voltage (CV), which is related to the actual system temperature, falls below the bottom of the ramp the heater is turned on. If CV is above the top of the ramp the heater is turned off. For CVs lying between the top and bottom of the ramp, power is supplied to the heater in short bursts (FIG 3.15). The temperature controller is located in the rack, and the fan heater has been placed at the back of the rack below the bridge. A thermistor fixed to the back plane of the capacitance bridge, has been used as the temperature sensor. The system temperature has been set to a value of 24.5°C. When the temperature is low the heater is turned on. When the system reaches the desired temperature just sufficient power is supplied to the heater to make up for the system losses. This results in accurate (±0.25K) temperature control (FIG 3.16).

Obvious problems arise in maintaining the whole rack of instruments at a constant temperature and care must be taken to see that 'hot spots' do not arise. Several mercury thermometers and a second thermistor positioned on the bridge have been used to monitor the rack temperature. At different points throughout the bridge variations of ±1°C have been observed. The main concern however is that the bridge itself is maintained at a constant temperature in order that the capacitance standards remain stable. Variations of only ±0.5°C have been observed over the bridge area. The author was also plagued with the unusual problem of a good English summer with temperatures rising to 33°C. The remedy to this problem was to wait until cooler weather was forecast before beginning the experiment.

3.3 The Heart of the Automatic Microelectronic Control System

A full explanation of the constituent parts of the microelectronic control unit is given in Chapters 4, 5 and 6. The interfacing electronics and instrumentation used in the measurement of thermal expansion, over the temperature range 1.5K - 300K, has been detailed in this chapter. Overleaf is a summary of the main functional requirements of the microelectronic control unit, for the
experimental determination of linear thermal expansion.

1. Control of the stepper motors to rebalance the capacitance bridge.

2. Ability to read the current capacitance display from the 50-way b.c.d output at the back of the bridge.

3. Ability to read the off-balance voltage from the bridge via the signal detection system, and evaluate the equivalent off-balance capacitance.

4. Ability to calculate the on-balance capacitance of the capacitance cell from the display capacitance and the off-balance capacitance.

5. Control of the gain of the low-noise amplifier to change the system sensitivity when required.

6. Control of the time-constant of the phase sensitive detector to reduce system response to noise spikes on the signal.

7. Ability to determine sample temperature from 1.5K - 100K using the germanium resistance thermometer, (voltage drop being determined by the Keithley nanovoltmeter and the current being set by the constant current source).

8. Ability to determine sample temperature from 1.5K - 300K using the Rh/Fe resistance thermometer. This measurement being returned to the system via the 3120 temperature controller and IEEE interface.

9. Ability to perform an automatic self-calibration of the instrument before commencing an experiment.

10. Ability to store a large quantity of experimental data ready for data analysis.

11. Ability to interface to the system operator via a visual display unit and keyboard.
FIG 3.1
SKETCH OF THE CRYOSTAT SYSTEM ARRANGEMENT.

- Sample rod evacuation valve.
- Electrical connectors
- Sliding seal on sample rod.
- Dynamic pumping port for VT insert.
- Main bath exhaust port.
- Sample rod pumping line.
- Outer vacuum case.
- Magnet support system.
- VT. pick up tube.
- Nitrogen vessel.
- 7 Tesla compensated solenoid
- Axial window set.

Helium max level:

Min level:

\[ \frac{L}{20''} \]

\[ \frac{L}{16''} \]
(PHOTO 3.1) THE TOP OF THE CROYOSTAT SHOWING THE ASSOCIATED VACUUM LINES AND VALVES
FIG 3:2
SKETCH OF DYNAMIC VARIABLE TEMPERATURE INSERT (VTI)
Electrical and vacuum connections

"O" Ring sealed top fitting

Knurled top plate.

Sliding seal.

Sample rod pumping line.

Wiring loom.

\( \sim 80 \text{ K Contact screw.} \)

\( \sim 30 \text{ K Contact cone.} \)

Indium seal.

Demountable vacuum case.

Copper support rod.

Thermal contact.

Experimental space.

Dynamic flow from V.T. Insert.

Temperature sensor mount + heater.

Axial window.
VACUUM SYSTEM FOR THE CRYOSTAT: LAYOUT, FIG 3.4

TO BALLOONS AND HELIUM RECOVERY SYSTEM
PUMPING LINE FOR VACUUM ON MAIN DEWAR
HIGH VACUUM FOR THE VARIABLE TEMPERATURE INSERT
HELUM BATH

VARIABLE TEMPERATURE INSERT

HIGH VACUUM FOR SAMPLE ROD
TRANSFER TUBE VACUUM JACKET

FLOW METER
12

13 PRESSURE VENT

14

15

16

17 VENT

18

19

20

PUMP SWITCH FOR R3

BAFFLE VALVE

LIQUID N2 TRAP

DIFFUSION PUMP D1

BACKING PRESSURE

FORE LINE TRAP

HELUM SEALED ROTARY PUMP R3

ROTARY PUMP R1

ROTARY PUMP R2

PUMP Pe3 PENNING
Pe3

R1 D1 R2 D2 PUMP SWITCHES
Pe1 Pe2 Pe3 Pe4

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20
(PHOTO 3.4) VACUUM SYSTEM FOR THE CROYOSTAT
FIG 3.5 BLOCK DIAGRAM OF THE SIGNAL DETECTION AND CONTROL INSTRUMENTATION.
COPPER CAPACITANCE DILATOMETER CELL (FIG 3-6)

TECHNIQUE FOR MOUNTING SPECIMEN (FIG 3-7)
FIG 3-8
ELEMENTARY CAPACITANCE BRIDGE
WITH TRANSFORMER RATIO ARMS
RELAY NETWORK FOR CHANGING THE GAIN OF THE LOW NOISE AMPLIFIER AND TIME-CONSTANT OF THE P.S.D. (FIG 3.9)
(PHOTO 3.10) THE STEPPER MOTORS AND BEVEL GEARS FOR THE MECHANICAL CONTROL OF THE CAPACITANCE BRIDGE ARMS
(PHOTO 3.11) THE CAPACITANCE BRIDGE AND ASSOCIATED ELECTRONICS
Curved path travelled by the lever arm

Rotation of steel rod.

Vertical movement of brass coupling

Brass lever arm
1 cm slot

Threaded steel rod
Brass coupling

(passes through hole in coupling & slot in lever arm)

(FIG 3.12) COUPLING BETWEEN THE BRASS LEVER ARM AND THE THREADED STEEL ROD
THE AUTO/MANUAL BRIDGE INTERFACE CIRCUIT (FIG 3.13)

[Diagram of the circuit with various components and connections labeled.]
PRECISION CURRENT SOURCE 1μA-10μA-100μA-1mA (FIG 3.14)
PROPORTIONAL TEMPERATURE CONTROL (FIG 3-15)

Controlling voltage (related to temperature)

Ramp voltage

Current applied to load.

Load (heater) turned on at zero voltage points.

Bursts of current are applied to the load when the controlling voltage lies between the top and bottom of the ramp voltage.

EFFECTS OF PROPORTIONAL CONTROL COMPARED TO ON/OFF CONTROL (FIG 3-16)

Temperature

Hysteresis

Time

On/off control

Proportional control
4.1 Distributed Control Using Microcontrollers: Justification for the Choice of Design Approach

In automating the experimental tasks involved in the measurement of linear thermal expansion, the approach that has been adopted is that of 'distributed control using microcontrollers'. The system automation is sub-divided into a number of specific control functions, and a single microcontroller is assigned to each function. This multiple processor implementation gives a modular solution, and allows the system to be expanded in a simple fashion. Each processor controls and makes measurements on a localised function, providing feedback information as necessary. As the microcontrollers are working concurrently, system tasks may be carried out simultaneously, alleviating the problem of task prioritisation which would occur in a single processor or personal computer system.

Distributed control systems are easier to maintain and debug. The allocation of specific areas of control to individual processors simplifies the finding and isolation of system faults. With a single processor system the effect of one fault can mean total system failure; with a multiprocessor system a fault occurring within one processor may be isolated, and will not necessarily result in catastrophic system failure.

The microcontroller used for this system design is the Intel 8751. The 8751 is a member of the MCS-51 family of Intel chips; it is the EPROM (Erasable Programmable Read Only Memory) version of the 8051 microcontroller. This single chip controller has a number of special features as summarized in section 4.2. With a multiprocessor system, a set of these special features is available to each section of the automation system. Of particular importance are the large number of bit selectable Input/Output (I/O) lines, (32 I/O lines are available on each 8051). The system design requires a large number of I/O lines for experimental measurements and instrument control. If a personal computer were to be used the hardware expansion and multiplexing of the I/O ports would become complex, as would the
software needed to 'mask-out' specific port lines from the byte addressable ports to enable single line control.

The distributed control system has been designed with one 8751 as the master controller and three slave 8751 processors. Communication between the master and slave processors is via a serial link. This well tested communication link consists of a simple pair of wires. One line is used for transmission of serial data and the other is used for receiving serial data. This communication link has an extremely high noise immunity for low baud rates (2.4Kbaud) over a distance of several meters. The 8751 has its own built-in serial port, configuring the serial port for a given data rate and protocol is easily implemented in software (see section 4.2).

Distributed control is well suited to the automation of a complex experimental system where mechanical control, experimental measurements, and controlled feedback are all required. When the system is expanded the utilisation of a serial link and a master/slave communication protocol will mean that the additional task may be assigned to a new slave processor, which can then be simply connected onto the existing serial link.

4.2 Major Features of the 8051 Microcontroller

The Intel MCS-51 microcontroller was selected as the computing engine for each module in the automation system. One of the main advantages of the 8051 is that it is a complete computer on a single chip of silicon, incorporating the central processing unit, memory, and input/output. This controller has many other important attributes which favour its use, these are outlined below, and summarized in FIG 4.1.

8-Bit CPU

The central processing unit (CPU) has an 8-bit architecture. An 8-bit architecture normally refers to the size of the internal registers being 8-bits wide or byte size. The implications of this are that most CPU operations or calculations act on bytes of information. However, a major advantage of the 8051 is that most of
its registers are also bit addressable, so that individual bits of information can be accessed and operated on easily.

**Internal Port Lines**

Four internal 8-bit ports provide 32 input/output lines for communication with the outside world. Under software control these lines are bi-directional and bit addressable. All Port3 pins have an alternate function as listed below:

- **P3.0** RXD: serial input port
- **P3.1** TXD: serial output port
- **P3.2** INT0: external interrupt 0
- **P3.3** INT1: external interrupt 1
- **P3.4** T0: timer/counter 0 external input
- **P3.5** T1: timer/counter 1 external input
- **P3.6** WR: external data memory write strobe
- **P3.7** RD: external data memory read strobe

**Memory Organisation**

The 8051 has its own internal data memory and program memory space. The data memory consists of 128 bytes of "on-chip RAM", plus a number of special function registers, and in addition there is the option of accessing 64Kbytes of external RAM memory. Up to 64Kbytes of program memory can be accessed by the microcontroller, the lower 4Kbytes may reside on-chip. To access the internal program memory the EA pin must be tied high. External memory can be addressed via Port2 and Port0. Data memory can use either a 16-bit address or an 8-bit address. The external read and write control lines are provided by the alternate functions of pins P3.7 and P3.6. A fetch from external memory always uses a 16-bit address, and the "program store enable pin" (PSEN) is used as the read memory strobe.

**Timer/Counters and Timer Generator**

Two 16-bit timer/counters are available for use in four different modes of operation. These modes can be initialised by selecting the appropriate bits in the special function registers TMOD.
(timer/counter mode control register) and TCON (timer/counter control register). The timer/counters can be used to determine pulse widths, measure time intervals or initiate events. They allow up to a maximum interval of 65,536 instruction cycles (over 65ms) with a 1μs resolution, assuming the microcontroller is using a 12MHz crystal oscillator. A longer delay may easily be obtained using software. One important mode of operation is the use of timer 1 as a baud rate generator for the 8051's serial port. The baud rate is the number of bits of data transmitted per second. Timer 1 can be used to select the speed at which information is to be communicated via the serial port.

The 8051 has its own internal clock generator. This clock generator provides the internal clock signals to the processor. All that is required externally is a crystal oscillator and two capacitors. The internal clock signal defines the processor's internal phases, states and machine cycles.

Interrupt Structure

There are five interrupt sources on the 8051. Two are from the external interrupt pins INT0 and INT1; these can be either level or transition activated depending on bits ITO and IT1 in the special function register TCON. Each timer/counter can cause an interrupt to occur when its 16-bit register overflows. The fifth source of interrupt comes from the 8051's serial port when the transmission or reception of a serial data byte is complete.

Each interrupt source may be independently enabled or disabled, and each may be classified as high or low priority. A high priority interrupt source can interrupt a low priority service routine. If required, an interrupt source can be disabled to allow polling - continuous testing of the interrupt bit to see if a specific event has occurred.

For each interrupt source a particular program memory address has been reserved starting at 0003H and continuing at eight byte intervals. When an interrupt occurs the CPU automatically executes an internal subroutine call to the corresponding address. At this address there is usually a 'sign-post' indicating to the CPU the
start address of the user's interrupt service subroutine. The CPU jumps to this routine, executes the instructions contained there, and then returns to the main program at the point where it was interrupted.

**Serial Port**

A very attractive feature of the 8051 is a full duplex serial port; it is capable of simultaneously receiving and transmitting serial information. It is this feature that has been utilised in the multiprocessor system design detailed in this thesis. The receive and transmit registers are both accessed through the special function register called SBUF, although SBUF physically consists of two separate registers. The port can operate in one of four modes which incorporate variable baud rates and transmission of up to 11 bits of information at one time, (9 data bits, 1 start bit and 1 stop bit). The mode that has been adopted for this system design is that of 'multiprocessor communication' and, as such, it deserves a special mention at this point.

If the serial port is set up for mode 2, the baud rate can be varied under the control of timer 1, and 9 serial bits of information are received plus the start and stop bits. The 9th data bit received is stored in bit RB8 of the serial port control register (SCON). By initially setting bit SM2 of SCON the serial port will be programmed so that when the stop bit of a serial data byte is received, the serial port interrupt will only be activated if RB8 = 1, that is if the 9th bit received is a '1'. If bit RB8 = 0 the serial port interrupt will not be activated even though a serial data byte has been received.

This particular feature can be used in multiprocessor systems where one 8051 is set up as the master processor and the other 8051s are set up as the slave processors. Each controller has its own unique address and when the master wishes to transmit a block of data to one of the slave processors it sends out an address byte on the serial port. An address byte differs from a data byte in that the 9th bit sent is a '1' in an address byte and '0' in a data byte. With their SM2 bits set to '1' the slave controllers can only receive serial address bytes; they will ignore all data bytes which
have the 9th data bit set to '0'. However all slaves will be
interrupted by an address byte, each slave will read and examine the
byte to see if it is being addressed by the master. The addressed
slave will clear its SM2 bit and prepare to receive data bytes from
the master. All the other slaves will leave their SM2 bits set,
carry on with their own specific tasks, and ignore any data bytes
sent by the master. The master processor can then send serial data
bytes to the addressed slave, (FIG 4.2).

Boolean Processor

The 8051 is particularly important in that it is a single chip
controller designed to perform both large "number crunching"
activities and "bit-manipulation". This makes it a suitable choice
for a number of tasks ranging from pure computation to control
applications.

It is the "bit-manipulation" capability of the boolean processor
that will now be briefly discussed. A number of instructions within
the 8051 instruction set operate on single bit (Boolean) variables.
Individual bits may be set or cleared, or tested as true or false.
Conditional jumps may be performed within a program depending on the
state of a specific bit. Single bits may be moved from one
addressable bit to another. Complex logic operations (AND and OR)
can also be applied as specific bit operations. This greatly
simplifies the software, relieving the need to mask out particular
bits within a byte, which do not need to be tested or altered when
carrying out an operation.

The 8051 internal RAM address space is set up such that there are
128 addressable bits available to the user between bit address 00H
and 7FH. Bit addresses 80H to F7H define bits within the
controller's special function registers and as indicated earlier
these can be both bit and byte addressable.
4.3 Software Development Environment

4.3.1 Software Programming Using PL/M51

The 8751 microcontrollers within this system have been programmed using the high level language PL/M51. A high level language more closely resembles the human thought process than a low level assembly language does, and as such it is easier to read and understand. It was designed by the Intel Corporation to meet a wide range of requirements in computer systems and applications. PL/M is used for programming various families of Intel processors with the option of converting PL/M code from one processor to another, thus making it portable.

The language has many advantages, it is a block structured language which enforces structured programming. If a program is well structured it becomes easier to maintain and debug. This flexible language has the facility to clearly define the scope of user variables, so that variables may be declared local to a particular procedure or global to a whole program module. PL/M51 is a general purpose language, it does not have the constraints of FORTRAN which is primarily suited to scientific applications, nor is it restricted to mainly business tasks like COBOL. It is a richly featured language which allows direct access to the microcontroller's functions and I/O lines.

Every PL/M statement forms part of at least one block which commences with a 'DO' statement and finishes with an 'END' statement. The in between statements in the block make up the processes and data definitions. These blocks can be nested one inside the other. Block structured programs are most clearly represented by block structured flowcharts. The approach has been to use Nassi-Schneidermann diagrams\(^1\). For comparison and clarity the software overview flowcharts given in Chapters 5 and 6 have been shown in both the block structured and the sequential flowchart notation.
4.3.2 Series IV Microcomputer Development System

A good way to develop programs for use in dedicated applications is with a comprehensive development system. The Series IV is a real microcomputer system, complete with monitor, keyboard, dual floppy disk drive, 32MByte Winchester hard disk, and its own operating system called ISIS. It provides the host environment for running a wide variety of hardware and software development tools.

The development system supports software for text and program editing using the versatile text editor, Aedit. Compilers for the popular high level languages PL/M, PASCAL, FORTRAN and C are supported along with a variety of macro-assemblers covering several families of Intel processors. The Series IV interacts with a range of in-circuit emulators. These powerful emulation tools enable debugging of the prototype microcomputer system hardware and software concurrently. Thus in a number of ways the user-friendly, menu-driven Series IV greatly eases the burden of the system design, development and debugging.

4.3.3 The Process of Program Development

The first step is to completely define the design problem that is to be solved. Having clearly defined the problem, a proposed solution can be outlined in terms of the software and relevant hardware. At this stage a more detailed design of the hardware can be tackled.

The next step is to design the software for the system. For the reasons explained earlier (see 4.3.1), the programming language chosen for this task is PL/M51. The best way to deal with a large software problem is to break it down into smaller tasks or modules. The software modules are then sub-divided further into individual program procedures (or sub-routines). PL/M51 is ideally suited to this approach as it is a block structured language.

Having defined the program modules and procedures, the software can be coded into PL/M51. It is at this point that the Series IV development tools come into play. The text editor, Aedit, allows the program code to be entered and stored as a file in the memory of the development system. The PL/M51 code is now ready for translation.
into machine code using the PL/M51 compiler. The resulting relocatable object code can be linked together with other program modules and useful library files using the RL51 linker and locator software. At the end of this development process an absolute object module file is produced ready for testing.

For detailed and extensive software and hardware debugging the ICE51 in-circuit emulator has been used. This extremely powerful development tool is dedicated to the 8751 microcontroller and has two modes of operation.

The first is a stand-alone mode for software execution and debugging. The emulator hardware includes a crystal power accessory which provides the clock pulses and power to the emulation processor whilst operating in this mode. The emulation processor allows both real-time and single step execution of a program's object code. The ICE51 simulates the environment of an 8751 processor providing 8K of RAM for user code. The ICE51 allows the user to set breakpoints at specified program positions so that real-time execution can be halted at points of interest, and the contents of memory locations and internal registers examined or altered. The emulator also incorporates symbolic addressing.

During emulation ICE51 maintains a trace buffer which can collect data on up to 1000 frames (that is a maximum of 250 instructions). The trace buffer also collects I/O port values. Thus a section of code can be emulated, and then the trace buffer examined to observe that the code has executed correctly. The trace buffer instructions are displayed in assembly language mnemonics, not as PL/M code.

When the prototype system hardware is available the ICE51 can be used in the second mode of operation. In this mode the emulator interacts with the user hardware through a buffer box and cable. The cable terminates with a 40 pin plug which fits into the 8751 socket position on the user hardware. In this mode the emulator plug physically replaces the 8751 chip in the user system and power and clock are provided by the user hardware. All the mode 1 facilities are available in mode 2. In addition the emulator is now able to access the user's external memory (if present), interrogate port pins and simulate signals to test external interfacing electronics.
connected to the system.

The aim of the diagnostic testing is to prove that the system design is fully functional in all of its executable modes of operation. To further this aim Macro blocks can be created using the ICE51. These Macros consist of sequences of emulator commands brought together in a block, to extend and automate sections of the production testing.

Finally when rigorous testing and repetition of the process of program development has produced a working hardware and software system design the absolute object code can be programmed into the 8751 EPROM chip. For this step the Series IV is used in conjunction with the Intel PROM-Programmer. The obvious advantage of using the 8751, as opposed to the 8051, is that should errors occur on running the prototype system the program can be erased from the 8751, using ultra-violet light, and a revised version can then be programmed in.

4.4 System Automation: Overview of the multiprocessor Control Unit

A schematic overview of the distributed control system is shown in FIG 4.3. It should be noted that the hardware modules indicated refer to a complete electronic control card associated with a particular function (of which the 8751, as the computing engine, forms an integral part). The experimental tasks requiring automation have been clearly specified in section 3.3. The modular solution to the system design will now be given, outlining the automation tasks allocated to each module and the interconnections between them.

The Off-Balance Voltage Reader and Conversion Module (OBCCON) reads in the analogue off-balance voltage (OBV) signal from the capacitance bridge via a 12 bit A/D converter. This signal has already passed through a substantial detection and filter system consisting of a low noise amplifier, coherent filters and phase sensitive detectors (PSD). Knowing the present gain setting of the low noise amplifier, and using internally stored calibration data, the slave 8751 for OBCCON is able to evaluate the equivalent off-balance capacitance corresponding to the OBV reading from the bridge. This information can then be made available to the motor control module when requested.
The Motor Control Module (MOTCON) is controlled by a slave 8751 processor. This module reads in the capacitance display setting from the capacitance bridge. It also re-balances the capacitance bridge by controlling the movement of the mechanised arms which are used to change the capacitance display setting. This module receives the present off-balance capacitance reading from OBCCON. Data transfer between the two slaves is via a parallel port and is initiated by an external interrupt from MOTCON. The handshake lines are used to ensure successful data transfer. From the current bridge display and the off-balance capacitance, the 8751 motor control processor can calculate the actual capacitance of the sample cell.

The Sensitivity Control Module (SENCON) is controlled by a third slave 8751 processor. This module monitors the OBV from the bridge and can interrupt the other two modules when it is necessary to re-balance the capacitance bridge, or change the system sensitivity by altering the gain of the low noise amplifier. The criteria for making these changes will be discussed in detail in Chapter 5. Information transfer between the modules is via a parallel port, initiated by an external interrupt and completed using handshake signals.

The Master Control Module (MASCON) is controlled by the 8751 master processor. This processor carries out all its communications with the slave processors via the serial link. Each slave processor has its own unique address, the master is able to send or request information from a particular slave by first sending out the slave's address on the serial link. The exact protocol for the multiprocessor communication is explained in section 4.2.1. The master control module may request the present on-balance capacitance from MOTCON or the off-balance capacitance from OBCCON or the present gain or time-constant setting from SENCON. MASCON may over-ride the system automation executed by the slave processors and command changes to system parameters.

The system runs in two modes of operation: calibration mode or experimental mode. Within the calibration mode the system carries out its own self-calibration in preparation for an experimental run. It is the master control module that will initialise the slave modules into the calibration mode. Whilst in this mode OBCCON
Interrupts MOTCON requesting that a specified capacitance value be set on the bridge, and OBCCON interrupts SENCON requesting that a particular gain setting be output to the low-noise amplifier. With the knowledge of the actual capacitance and gain settings OBCCON can read in the equivalent OBV and store the value in the appropriate position in the calibration table. In this mode the data transfer and interrupt direction is from OBCCON to MOTCON and SENCON. Having completed the system calibration successfully the master can initiate the slaves into the experimental mode of operation, and the collection of experimental data can begin.

There is still the need to transfer the experimental information to the system operator for storage and data analysis. This is achieved by transferring the data to an IBM personal computer (IBM 286 PC) from the master control module along a second serial link. The IBM PC's hard disk enables mass storage of data. A single experiment can involve the collection of 70Kbytes of data over a period of 100 hours.

In order to calculate the linear thermal expansion of a specimen, as well as the capacitance data used to determine the change in length of the sample, it is necessary to monitor the sample temperature. Initially it was planned to have a fifth module monitoring the sample temperature, however this proved unnecessary when a simpler solution became available. The sample temperature was monitored using two resistance thermometers (see 3.2.4.). The voltage drop across each thermometer was measured by sensitive instruments both of which have their own IEEE interface. It is now relatively cheap to purchase an easily installed IEEE interface card for the IBM PC, and so temperature measurements were read into the computer directly. In the case of the germanium resistance thermometer it was necessary to control the thermometer current. This was achieved using a bespoke constant current source which was controlled directly by the IBM PC.

The automation of a system designed to measure linear thermal expansion using a three-terminal capacitance dilatometer has been broadly outlined above, and will be explained in detail in the following two chapters. The only communication between the master and slave modules, and the master module and the IBM PC, is via a
serial link. One may then question as to why the three slave modules are connected directly together via parallel ports and interrupts, when all the information to be exchanged between the slaves could have been sent via the master on the serial link. An explanation for adopting this approach is given below.

The three slaves form a functional unit which monitors and controls the capacitance measurements. If the system were to be expanded this sub-module would in all probability remain unaltered by additional features. For example, the addition of a super-conducting magnet control module could be regarded as a separate sub-module in its own right, and there would be no need for communication between the slave modules in either of the sub-modules. If the master processor was initially to co-ordinate all action, as the number of slaves increased so would the utilisation of the bandwidth of the bus. This would provide a limit on the maximum possible expansion of the system and involve complicated re-programming of the master for each new expansion.
PORT 0 & PORT 2 ARE USED AS THE ADDRESS & DATA BUS LINES WHEN ACCESSING EXTERNAL DATA OR PROGRAM MEMORY.

PORT 0 & PORT 2 ARE USED AS THE ADDRESS & DATA BUS LINES WHEN ACCESSING EXTERNAL DATA OR PROGRAM MEMORY.

12MHz

+5V

CHIP RESET ON POWER UP

POWER SUPPLY

PORT 3 PINS HAVE TWO FUNCTIONS

FIG 4.1 SPECIAL FUNCTIONS OF THE MCS-51 MICROCONTROLLER
FIG 4.2 SERIAL INFORMATION

ADDRESS BYTE

START BIT

8 DATA BITS

DATA BYTE

9th DATA BIT

STOP BIT
INPUT CAPACITANCE DISPLAY DIGITS FROM THE BRIDGE

SELECT AND CONTROL MOTOR ARMS

COMMUNICATION WITH THE SYSTEM OPERATOR

INPUT VOLTAGE DROP ACROSS Rf RESISTANCE THERMOMETER

INPUT VOLTAGE DROP ACROSS Ge RESISTANCE THERMOMETER

SELECT CURRENT FOR Ge THERMOMETER

FIG 4.3 A SCHEMATIC OVERVIEW OF THE DISTRIBUTED CONTROL SYSTEM.
CHAPTER FIVE - System Design: The Slave Control Modules

5.1 The Off-Balance Capacitance Control Module (OBCCON)

5.1.1 Circuit Diagrams and Photograph of OBCCON

The first figure, FIG 5.1, is of the electrical circuit for the OBCCON module. The second figure, FIG 5.2, shows the module's connections to the other system controllers. A photograph of the printed circuit board (PCB) is shown in PHOTO 5.1.

5.1.2 Overview of the Functions of OBCCON

The function of this module is to read in the off-balance voltage from the capacitance bridge via the signal detection and filter system, and to convert this voltage into the equivalent off-balance capacitance that it represents. OBCCON achieves this by scanning internally stored calibration data that has been obtained during the auto-calibration of the instrument (see 5.1.5). Two flowchart representations of the main software programming module are given in FIG 5.3 and FIG 5.4. For detailed explanations of individual software procedures refer to the separate 'Program Listings' booklet.

On power-up a number of program parameters and internal registers of the 8751 are initialised. The controller then waits within a program loop for either the experiment to commence or for a calibration to be requested. Both of these actions are caused by the master controller (MASCON) interrupting the slave on the Serial Port Interrupt. If a calibration is requested the slave will call a procedure named SELF_CALIB (see 5.2.5), and return to this loop on completion of the calibration. If the master is requesting that the experiment commences, OBCCON will drop out of this initial loop and enter the main program 'DO' loop, labelled 'COMMENCE'. During the experiment OBCCON continually carries out two main functions:

The first task involves accessing OBV data from the 12-bit A/D converter; this is executed by the READ_A_D procedure.
The second task takes the current OBV data read in and calculates the off-balance capacitance that it represents. This is executed by the OBV_TO_OBC procedure. For this function OBCCON needs to know which sensitivity setting the low-noise amplifier is on (i.e. 30dB, 50dB, 70dB or 90dB), the absolute value of the OBV, and its sign (positive or negative). The controller can then selectively scan through the capacitance calibration data stored in external RAM memory to evaluate by how much the capacitance bridge is off-balance.

Apart from these procedures which are continually running, three interrupt procedures are in operation. The SERIAL_PORT interrupt procedure is vectored to whenever a serial data byte is received from the master or transmitted to the master. Having addressed the OBCCON slave (see multiprocessor communication, 4.2), the master sends out a single command byte to tell the slave what action to take. The master may request a calibration, start or stop the experiment, request the latest off-balance capacitance data, or ask for the calibration data to be transferred (copied) to the IBM PC so that it can be viewed by the operator. The serial port interrupt has been set to high priority and can therefore interrupt lower priority interrupts. It is important that the master is able to take control of the slave at any point during the program, and if necessary cause the experiment to halt.

External interrupt 0 is used by the motor controller (MOTCON) to request the latest OBC data. The data is sent via PortB of the external RAM and I/O chip on the OBCCON module. Four bits of the port are used to indicate the number of capacitance digits the bridge is off-balance, and four bits are used to indicate which bridge arm the data refers to.

For example:

PortB = 04H: 0-decade 0, the 1aF arm; 4 = 4 digits off-balance
i.e. the bridge OBC = 4aF

PortB = 42H: 4-decade 4, the 10fF arm; 2 = 2 digits off-balance
i.e. the bridge OBC = 20fF

An extra port line is necessary to indicate whether the OBC is
positive or negative i.e. +20fF or -20fF off-balance. Having interrupted OBCCON the OBC data is handshaked across to MOTCON using the DATA$REQUEST and DATA$GRANTED port lines. Use is made of a fully-locked asynchronous bus (FIG 5.5), to guard against data errors. Having output the OBC data OBCCON sends a finished message (0FAH) on PortB to MOTCON to indicate the data transfer is complete. This task is executed by the interrupt procedure called OBC_REQUEST. External interrupt 0 is enabled during the main program module and is set as a low priority interrupt.

As stated earlier in this chapter, in order to know which calibration data is to be scanned through, OBCCON needs to know which gain the low-noise amplifier is set to. The sensitivity controller (SENCON) controls the gain of the low-noise amplifier, and informs OBCCON of any changes that are going to be made to the system's sensitivity. SENCON causes an interrupt on external interrupt 1 of OBCCON. The gain information is transferred via the low nibble of PortA on the external RAM and I/O chip. The data is handshaked using the strobe and buffer-full lines also on the external 8156 chip (see 5.1.2.). This task is executed by the interrupt procedure SENSITIVITY_CHANGE.

The 8156 RAM and I/O chip on the OBCCON board expands the I/O capacity of the module and provides storage space for the calibration data. The 8751 is limited to 128 bytes of on-board RAM which is needed for internal program variables and the stack. To store the calibration data 200 data bytes are required; the 8156 has 256 bytes of RAM memory available for this purpose.

Finally, when the master commands the experiment to finish, OBCCON drops out of the COMMENCE loop, sends an 'EXPERIMENT_FINISHED' message to the master via the serial link, and then disables all interrupts. In reality the program does not come to a complete halt, but starts back at the beginning of the outer program loop labelled 'MAIN_PROGRAM'.
5.1.3. Details of Important Module Tasks

1 Reading data from the A/D converter

The OBV from the bridge is obtained from the output of the phase sensitive detector (PSD). The signal is transferred to the OBCCON module via a thin coaxial cable. A special edge connector incorporating two gold plated coaxial connections at either end (as well as three rows of pins) enables the signal to be fed into the analogue input of a 12-bit A/D converter. This converter forms part of the OBCCON module and interfaces directly with the 8751 controller. The converter\(^1\), (ICL7109), is the integrating type which provides high accuracy with low noise and low drift. This high accuracy is obtained at the expense of speed; the converter can only operate at up to 30 conversions per second. However since a high speed conversion is not a requirement of the system, but accuracy is, this type of converter was chosen in preference to the successive approximation type.

The chip requires both positive and negative 5V supplies; these can be obtained from a single +5V power supply by using a DC/DC converter chip. The A/D chip works by integrating the input voltage over a fixed number of counts. The output of the integrator is then de-integrated to zero at a fixed rate. Hence the time taken for the output to return to zero (represented by the number of clock periods counted) is proportional to the input signal.

The 12-bit A/D converter provides a resolution of \(\pm 4096\) bits. The analogue input range can be set by adjusting the potentiometer (VR1-FIG 5.1). For this system the range has been set to \(\pm 3\)V. With the low-noise amplifier on the lowest gain of 30dB the maximum OBC signal that can be detected is \(\pm 30\)fF (\(\approx \pm 3\)V OBV). For capacitance arms in the range 1fF to 1aF an OBV of 0.1V is approximately equivalent to 1 digit off-balance, (TABLE 5.1).

For the 1aF to 1fF arm then:

\[
1 \text{ digit off-balance} = 0.1V = \frac{4096}{30} = 136 \text{ bits}
\]

With this resolution an error of two bits will have a small effect.
(1.5% error on the least significant capacitance digit) when calculating the equivalent OBC.

By taking its RUN/HOLD input high the converter will continuously perform conversion cycles. The LBEN (low byte enable) input and the HBEN (high byte enable) input pins can then be controlled by the processor to access the 12-bit digital output, polarity and overrange information. During a conversion the STATUS output line goes high at the beginning of the signal integrate phase, and goes low one half of a clock period after fresh data has been stored on the output latches. The 8751 makes use of the STATUS pin as a data valid flag to decide when to access data; the controller waits until a conversion has just been completed before accessing the digital outputs. To conserve port pins both the low and high bytes of information are read via Port2 on the 8751. A single output pin and an inverting logic gate can be used to enable either the low or high data from the A/D.

Both OBCCON and SENCON access data from the A/D converter. To prevent both slaves from simultaneously trying to access the A/D converter, the IN_USE line indicates to OBCCON when SENCON wishes to read the A/D converter. In this case SENCON will take priority over OBCCON in reading the A/D converter, (see 5.3.2).

Details of the READ_A_D software procedure are given by the flowchart FIG 5.6. This procedure is utilised by both the main program module and by the calibration program module. The reason for this was to reduce the program byte count. The MCS-51 has only 4K of program memory and consequently memory space needs to be used frugally.

II Accessing data from SENCON via the 8156 RAM I/O and timer chip

OBCCON uses Port0 to access data from the 8156 via the multiplexed address/data bus. To read the information from the external ports the IO/\bar{M} line on the 8156 must be high, and the ports set to input, output or handshake mode by sending a control byte to the control register on the RAM chip. PortA is used as an output port during instrument calibration when OBCCON tells SENCON what gain to set on the low-noise amplifier, and PortA is used as an input port whilst
in experimental mode when SENCON notifies OBCCON of changes to the system sensitivity. In both cases the buffer-full (BF), strobe (STB), and interrupt line (INTR) on the RAM chip take care of handshaking the information.

For the strobed output mode (FIG 5.7) OBCCON writes the gain data to PortA. The write line (WR) going low causes INTR to go low. The interrupts on the 8156 are active high, whilst the interrupts on the 8751 are active low. Therefore an inverting logic gate has been placed between INTR on the 8156 and INTI on the 8751. OBCCON polls the interrupt pins to check that data has been transferred. When the write line returns high the BF line goes high to indicate to SENCON that data is available to be read. OBCCON then causes an external interrupt on SENCON to tell it that data is available. SENCON responds by reading in the data, and then taking the strobe line low. The 8156 will then take the BF line low, and SENCON responds by taking the strobe line high. Once the strobe line is high again the 8156 takes INTR low.

For the strobed input mode (FIG 5.8), SENCON places the gain data on PortA and then strobes the 8156 chip. The 8156 responds by reading the data into the buffer and taking the BF line high. SENCON, on seeing that the data has been read by the 8156 (i.e. BF = 1), takes the strobe line high again. This in turn causes an interrupt from the 8156 to OBCCON (i.e. INTR \(\rightarrow 1\) and INTI \(\rightarrow 0\)). OBCCON responds to the external interrupt by vectoring to an interrupt procedure called SENSITIVITY_CHANGE which reads in the data via the address/data bus. The read strobe causes INTR and the buffer-full line to return low.

III Utilising the calibration table to obtain the off-balance capacitance

Calibration data is stored for both positive and negative OBCs in the range \(\pm 30\)FF to \(\pm 1\)aF. As indicated by TABLE 5.1 different gain settings are used when obtaining data for different capacitance arms. Data is stored for \(\pm 1 - \pm 10\) digits off-balance for all arms except the 10FF arm where the range is \(\pm 1 - \pm 3 (\pm 30\)FF\) digits off-balance. When the slave inputs an OBV, a comparison is made with the OBV values stored in the calibration table. By knowing both the
sign of the OBV and the current gain setting, the number of calibration values to be compared with OBV is decreased to ten.

Having decided which set of calibration values are to be used for comparison, OBCCON calls the select procedure. This procedure compares the OBV with each of the ten calibration values, starting with 1 digit off-balance, until OBV is found to be less than or equal to the calibration value. When a match has been found a flag is set to halt further comparisons, and the corresponding OBC is then stored. This OBC value will then be available to be read by MOTCON. OBCCON will then read in a new OBV and the process of updating the OBC will be repeated.

If the sample capacitance is changing rapidly the system will be on the lowest gain (30dB) and an OBC of up to 30fF can be evaluated. Calibration values for 10fF, 20fF and 30fF have been stored, intermediate values such as 14fF, 22fF are not. These values are calculated by assuming a linear change in OBV between 10fF - 20fF and 20fF - 30fF. This is an approximation, but the occasions when the capacitance changes this rapidly are rare; in practical terms this compromise is an acceptable operational arrangement.

5.1.4 Problems Encountered: Diagnostic Tests and Module Emulation

In the initial design of the OBCCON module an 8-bit successive approximation A/D converter was used. Its advantages were: ease of interfacing to the processor, fast conversion times (9μs), a single +5V supply, and bipolar input ranges. The converter could also work in continuous conversion mode by introducing a delay between the output busy line, and the input write line which triggers the next conversion. This delay was provided by a dual monostable multivibrator which provided a short convert pulse (6.2μs) and a long delay pulse (5.2ms). The convert pulse must be in excess of 550ns, and the delay pulse must be long enough for OBCCON to have time to read the digital output before the next conversion. The converter features a linearity of ½ least significant bit and a resolution of 1 in 256 bits.

The circuit worked well; however the resolution was found to be insufficient for the required task. In the bipolar mode the
converter had a range of ±128 bits. For the lower capacitance arms in the range 1aF -1fF:

\[
128 \text{ bits} = 10 \text{ digits off-balance} \\
or \quad 13 \text{ bits} = 1 \text{ digit off-balance}
\]

The noise pick-up on the converter was the order of 1-2 bits, so that the percentage error was of the order of 7.5 - 15%. Despite the converter's ease of use it was decided that this error was unacceptably high. The OBCCON module was re-designed using a 12-bit A/D converter. In investigating 12-bit A/Ds the virtues of the integrating converter were discovered. The integrating converter, although far slower (30 conversions/second), provided high resolution and high noise immunity (see 5.1.2)

II: OBCCON required memory and I/O expansion. This was obtained by the addition of an 8156 chip. The chip is accessed using an 8-bit address (256 memory bytes). Using ASM51 assembler the 8751 can access external data memory using an 8 or 16 bit address bus utilising Port0 (low byte) and Port2 (high byte) of the controller.

However the program modules were written in PL/M51, and this programming language only allows data to be accessed using a 16-bit address. This posed a problem as Port2 (bits 8 -15 of the address) was needed for other I/O requirements. The problem could have been overcome by writing a macro in ASM51 and combining it with the PL/M51 modules; however a more easily implemented solution was found.

Ports on the 8751 are bi-directional. To set a port to act as an input port all bits must be set to '1' internally so that they may be pulled high or low by an external source. One 8-bit port on the 8751 was required to read in data from the 12-bit A/D (the port lines are multiplexed, alternatively reading low and high data from the converter). This port will act as an input port throughout the experiment and calibration. It was decided to use Port2 to read in the data from the A/D. Port2 is also used to output the high byte of the 16-bit address when accessing external memory. By setting external addresses to start at FF00H and go through to FFFFH it was ensured that Port2 was always set to '1' internally. Port0 was connected to the address/data bus of the 8156 enabling addresses from 00H - FFH to be selected. Hence, whenever an external memory read or write occurs, only '1's are written to Port2. This leaves

62
Port 2 acting as an input port when data from the A/D is read in. In effect the software accesses external data bytes using a 16-bit address, but in fact only the low byte of the address is actually used. This solution was found to work well in practice.

III: In testing the OBCCON hardware and software it was necessary to simulate the inputs from the other control modules. This was achieved using logic switching inputs and I.e.d. outputs. Problems occurred in accessing information from external Port A (used to transfer the gain value). Port A is used as both an input and an output port. To cover both possibilities an 'open circuit' position was added to the high or low inputs of the logic switches. When in the open circuit position the port pins could be used as an output port and their status read on a set of I.e.d's. A further refinement was to add switch debounce circuitry to the logic switching devices in order to prevent the controller from interpreting one logic change as several hundred. This had been an initial source of error when real-time emulation was attempted.

IV: Another switch problem was that of switching between accessing external data memory, and accessing the external ports on the 8156. This is controlled by the IO/M line (input/output or memory select). When an interrupt occurs care must be taken to first save the original status of this line. The original value of the line can then be restored at the end of the interrupt procedure, thus preventing unwanted changes to this pin.

V: Calibration values are stored as an array within the RAM chip. Before these values can be used successfully within the experiment they must undergo a conversion (see 5.1.5). This results in there being two sets of calibration values: the original set and the converted set. There is not enough memory space in the RAM to store both sets of calibration data. Therefore the original set of data is transferred to the IBM PC for safe storage before carrying out the data conversion, and only the converted data remains in the RAM chip for use during the experiment.

VI: Diagnostic testing of the OBCCON was an exhaustive business as each capacitance value had to be set on the bridge manually before the OBV value could be read in and stored in the calibration table
by OBCCON. Having obtained a good set of calibration data, and having tested that the calibration module was executing correctly, it was possible simply to type the calibration data into the storage array for use in the main module and by-pass the lengthy calibration task.

A short macro was written to run under the ICE51. The macro displayed OBV and OBC values at different points in the program whilst the main module ran continuously. The author could then continuously change the capacitance settings on the bridge, and observe that OBCCON responded correctly producing an accurate off-balance capacitance for the off-balance voltage read in.

VII: A number of test macros and programs were written as part of the emulation and verification process. For example the READ_A_D code was initially executed as a stand alone module. A variable analogue voltage was applied to the input of the A/D. The program module was adapted to read the input 100 times and store the value in an array. These array values could then be examined for discrepancies and accuracy. It was ensured the sign and overrange facilities worked correctly, and the range was adjusted to read a maximum input of ±3V.

5.1.5 The Calibration Software

All three slaves are put into calibration mode by a command from the master controller. Thereafter the calibration task is controlled by OBCCON, and monitored by the master. When carrying out a calibration the capacitance bridge must be set to 'cal' position. If the capacitance decades are then all set to zero a null off-balance voltage will be read from the bridge indicating that the bridge is on-balance.

The master sends a command to OBCCON which causes the SELF_CALIB procedure to be called. SELF_CALIB resides in a separate module called the CALIBRATE$MODULE. Two flowcharts are given in FIG 5.9 and FIG 5.10

During calibration OBCCON selects each OBC in turn starting with positive OBC values in the range +30ff - +1aF, and then moving on to
negative values in the range $-30\mu F$ to $-1\mu F$. OBCCON selects the appropriate gain setting for a particular arm, and then causes an external interrupt on SENCON to request that it sets the amplifier to the specified gain. When SENCON confirms that this has been done OBCCON causes an external interrupt on MOTCON to request that it sets a specified capacitance value on the bridge. OBCCON then waits until MOTCON completes the task of moving the arms on the capacitance bridge to achieve the desired off-balance capacitance. OBCCON can now carry out the task of reading in the off-balance voltage from the A/D converter and then placing it in the correct position in the table.

At the successful completion of the collection of calibration data for each arm, OBCCON sends a message back to the master so that the master can monitor the progress of the calibration task. When all the calibration data has been stored in OBCCON's RAM it tells SENCON to reset the gain to 30dB (the least sensitive position). OBCCON then outputs a finished message to SENCON and to MOTCON to tell them that their part in the calibration task is complete. OBCCON then notifies the master that the calibration data has now all been collected. At this point OBCCON must wait until the master is ready to read the calibration data from the slave and copy the data on to the hard disk of the IBM PC.

The master sends a command to OBCCON which calls a procedure to transfer the calibration data along the serial link to the master controller. As an error check, when a block of calibration data is transmitted two check-sum bytes are sent at the end of the block. These bytes are calculated by adding all of the transmitted bytes together and sending the total as the check-sum. If an error does occur in transmitting the data the master can command OBCCON to send the data again. Only when the data has been successfully transmitted will the master clear a WAIT flag which allows OBCCON to continue with the next part of the calibration procedure. The calibration data is then, in turn, transferred from the master controller to the IBM PC.

In the next part of the calibration OBCCON calls a procedure named CONVERT_TABLE_STORE. The purpose of this procedure is to convert the data to the form in which it will be used during the experiment.
Initially the table contains calibration values corresponding to whole digits off-balance (1, 2, ..., 10), but if an OBV is read in which corresponds to 2.4 digits off-balance there needs to be some method of deciding whether to choose 2 or 3 as the answer. It is for this reason that the half-way calibration values are calculated from the original data and stored in the table, i.e. the calibration data now corresponds to 0.5, 1.5, ..., 9.5 digits off-balance. For example if the OBV read in is greater than 7.5 digits but less than 8.5 digits the calculated value will be 8 digits off-balance. During this procedure the limiting calibration points are also evaluated for each arm corresponding to ±0.5 digits off-balance (minimum) and ±9.5 digits off-balance (maximum). These limiting calibration points are utilised by SENCON (see 5.3). When the data conversion is complete OBCCON sends a message to the master: CONVERT_CAL_COMPLETE, and then waits for the master's response. The master will respond by commanding the transfer of the converted calibration data. As before check-sum information is included at the end of the transmitted data block. When a successful transfer is complete the master clears the WAIT flag and OBCCON proceeds to the final part of the calibration task.

In the final part of the task OBCCON sends a message to the master stating that the limiting calibration data is ready for transfer. Again the master will command the transfer procedure to be called, and the limiting data will be passed to the master controller. On leaving the calibration module OBCCON resets PortA to act as an input port which is its normal mode of operation during the running of an experiment. It is important that the interrupt lines (INT$SENSE and INT$MOTOR) are left high; these lines are only in operation during the calibration tasks. Having completed the calibration task OBCCON will leave the CALIBRATE$MODULE and return to the MAIN$MODULE whence the SELF-CALIB procedure was called. The CALIBRATE$MODULE is complex and utilises the READ_A_D procedure from the main module; this was necessary to reduce the size of the program code.
(PHOTO 5.1) THE OFF-BALANCE CAPACITANCE CONTROL MODULE: OBCCON
Circuit Design for the Off-Balance Capacitance Controller Module Showing Board Components & Pin-Out to Edge Connector (Fig 5.1)
CIRCUIT DESIGN FOR THE OFF-BALANCE CAPACITANCE CONTROLLER SHOWING CONNECTIONS TO THE OTHER CONTROLLERS (FIG 5.2)

MASTER 8751

GAIN DATA INPUT
BUFFERS FULL
STROBE

OBC
DIGITAL ADDRESS

MOTCON 8751

P0.6 P2.6

RESET RD WR ALE AD0
RAM I/O PC1

&8156 [OBCCON CARD]
P0.8-P0.7 P3.7

DATA GRANTED

P2.0 P2.7

10 11
RXD TXD

10
P0.6

11
TXD RXD

RST

+5V

EA OFF-BALANCE CAPACITANCE

P0.0 P0.7 8751

P1.1 P1.2

ADDRESS/DATA 12 BIT A/D DATA, OVER-RANGE AND SIGN BITS

P1.0 12 BIT

STATUS

BIT 1 IN HIGH

BIT 8 A/D

BIT 9 OVER-RANGE

BIT 10 IN HIGH

BIT 11 SIGN

BIT 12 CONVERTER STATUS

BUFFER 7109

BUFFER (IC3)

EXAP 7V

VCC 5V

P2.6 P2.7 P0.8-P0.7

P2.0 35

GAIN VOLTAGE INPUT

GAIN DATA OUTPUT

BUFFER FULL

STROBE

P3.4 INTO

INTB

8751

PC1

PC2 (IC2) PC0

P80-PB7 PCL

1K

1K

DATA REQUEST 16

SIGN BIT 8 28

12 CHIP IN USE

ADDRESS/DATA

[SENCON CARD] RAM I/O & TIMER

8156

C1 C2

30pF

30pF
START

INITIALISE REGISTERS AND PARAMETERS

HAS THE EXPERIMENT COMMENCED?

NO

HAS A CALIBRATION BEEN REQUESTED?

YES

CARRY OUT CALIBRATION

RE-INITIALISE PARAMETERS

TELL MASTER: "EXPERIMENT COMMENCED"

HAS THE EXPERIMENT FINISHED?

NO

READ IN THE LATEST OFF-
BALANCE VOLTAGE FROM THE
CAPACITANCE SENSOR VIA THE 12
BIT A/D CONVERTER

HAS THE EXPERIMENT FINISHED?

YES

TELL MASTER: "EXPERIMENT FINISHED"

DISABLE ALL INTERRUPTS

NO

CONVERT DIGITAL OFF-BALANCE
VOLTAGE TO AN OFF-BALANCE
VALUE BASED ON STORED CALIBRATION DATA
OBCCON: FLOWCHART FOR THE MAIN SOFTWARE MODULE (FIG 5.4)

MAIN PROGRAM: Do while one (i.e. go forever)

Reset external memory chip (8156), clear BYTE$SENT
(this bit is set when a serial data byte is sent).
Select external input/output ports (IO/M = 1).
Output a command byte to the command status register to
 initialise the external I/O ports: PORTA=I/P, PORTB=O/P,
 use PORTC in handshake mode, enable PORTA interrupt.
Initialise interrupt line to SENCON and MOTCON.
Set OBCCON's own external interrupt lines high.
Select low level triggered interrupts on OBCCON.
Set timer 1 to auto-reload mode and turn timer 1 on.
Set up timer 1 counter with auto-reload value to
 give a baud rate of 3125 bits/second (timer 1 is used
 by the serial port as a baud rate generator).
Set serial port to high priority, and all other
 interrupts to low priority.
Set up serial port register to select 9 bit UART
 mode with variable data rate, set SM2 to disable
 reception of serial data bytes (i.e. 9th bit = 0),
 REN is set to enable serial reception.
Initialise handshake lines to MOTCON: set DATA$REQUEST
 ready for input. Clear DATA$GRANTED ready for output.
Set status pin to act as an input (controlled by A/D).
Clear TB8 (9th data bit transmitted) to indicate serial
 byte sent is a data byte, not a command byte.
Clear EXPT$IS$RUNNING bit to prevent the experiment from
 commencing. This bit is set by a command from the master
Clear CAL BIT to prevent the calibration procedure from
 running. This bit is set by a command from the master.
Initialise interrupt enable register: enable serial port
 interrupts and global enable bit, disable external
 interrupts and timer interrupts.

Do while the experiment has not commenced

<table>
<thead>
<tr>
<th>No</th>
<th>Has the master requested a calibration?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Carry out the calibration of the instrument by calling the SELF_CALIB procedure.</td>
<td></td>
</tr>
</tbody>
</table>

Select external memory (IO/M = 0), set sensitivity value
to 30db (lowest gain), set off-balance capacitance value
to not valid, set mode of operation to experimental,
send message to master saying 'experiment commenced'.
Enable external interrupts.

COMMENCE: Do while the experiment is running

Read in and store the latest 12-bit off-balance
voltage from the A/D converter (call READ_A_D).

<table>
<thead>
<tr>
<th>No</th>
<th>Is the experiment still running?</th>
<th>Yes</th>
</tr>
</thead>
</table>
|    | Convert the off-balance voltage (OBV) to an
   off-balance capacitance value (OBC) using
   internal calibration data (call OBV_TO_OBC). |

Tell master that the experiment has finished.
Disable all interrupts.
DATA REQUEST

"SEND ME DATA"

DATA AVAILABLE

"HERE IT IS"

DATA GRANTED

"I'VE GOT IT"

"I SEE YOU'VE GOT IT"

FULLY-LOCKED ASYNCHRONOUS BUS (FIG 5.5)
OBCCON: FLOWCHART FOR THE READ A/D PROCEDURE (FIG 5.6)

Set AGAIN to 1

FIRST: Do while AGAIN = 1

<table>
<thead>
<tr>
<th>No</th>
<th>Is the mode experimental?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Call initialise to set limits on size of OBV read in</td>
<td></td>
</tr>
</tbody>
</table>

Set STATUS bit for input (indicates whether A/D is busy). Set PORT2 for input to read in digital output from A/D. Clear OVER_RANGE, COUNT, DIGIT_STORE, & SIGN_SUM variables

AVERAGE: Do while count is less than 16

<table>
<thead>
<tr>
<th>Y</th>
<th>Does SENCON wish to access the A/D?</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Do</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Globally disable all interrupts.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wait whilst the A/D is not busy converting.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wait whilst the A/D is busy converting.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enable the high byte of data to be read from the A/D</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Store the 4 most signif. bits of data in high STORE.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Store the sign and overflow data information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enable low byte to be read &amp; store in low STORE.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Globally enable interrupts again.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>Is the mode experimental?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>No</td>
<td>Is stored data between 0.5 &amp; 9.5 digits?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WITHIN_ACCEPT_LIMITS=NO</td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Has there been an overflow?</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>(MODE=CAL) OR (MODE=EXPERIMENTAL &amp; OK)?</td>
</tr>
<tr>
<td></td>
<td>Add data to DIGIT_STORE, increment count.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>Is the sign of the data positive?</td>
</tr>
<tr>
<td></td>
<td>Add one to SIGN$SUM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>Has the gain changed or is experiment to finish?</td>
</tr>
<tr>
<td></td>
<td>ERROR$FLAG is set true, &amp; COUNT = 16 to exit loop</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>Is the experiment still running?</td>
</tr>
</tbody>
</table>

AGAIN = φ

| No | Has data changed sign or is ERROR$FLAG true? | Yes |
|        | AGAIN=0 leave FIRST loop | AGAIN=1 stay in FIRST loop |

Find the average reading by dividing DIGIT_STORE by 16

<table>
<thead>
<tr>
<th>No</th>
<th>Is the remainder &gt; 8</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Add one to the average result (average OBV value).</td>
<td></td>
</tr>
</tbody>
</table>
INDICATION OF THE SIZE OF VOLTAGE PRODUCED FOR A GIVEN OBC (TABLE 5.1)

<table>
<thead>
<tr>
<th>Gain</th>
<th>Arm</th>
<th>Voltage equivalent for 1 digit off balance</th>
<th>Voltage equivalent for 10 digits off-b</th>
</tr>
</thead>
<tbody>
<tr>
<td>90dB</td>
<td>1aF</td>
<td>0.15V</td>
<td>1.5V</td>
</tr>
<tr>
<td>70dB</td>
<td>10aF</td>
<td>0.1V</td>
<td>1.0V</td>
</tr>
<tr>
<td>50dB</td>
<td>100aF</td>
<td>0.1V</td>
<td>1.0V</td>
</tr>
<tr>
<td>30dB</td>
<td>1fF</td>
<td>0.1V</td>
<td>1.0V</td>
</tr>
<tr>
<td>'30dB</td>
<td>10fF</td>
<td>1.0V</td>
<td>Max=30V=30fF</td>
</tr>
</tbody>
</table>

N.B. GIVEN VOLTAGES ARE ONLY ROUGH APPROXIMATIONS

8156 HANDSHAKE TIMING: STROBED OUTPUT MODE (FIG 5.7)

8156 HANDSHAKE TIMING: STROBED INPUT MODE (FIG 5.8)
SELECTING GAIN SETTINGS FOR THE LOW-NOISE AMPLIFIER  
(TABLE 5·2)

<table>
<thead>
<tr>
<th>Gain</th>
<th>Relay number</th>
<th>Binary bit pattern</th>
<th>Port pins on gain ports</th>
<th>D-type pin no</th>
<th>Ribbon cable colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 dB</td>
<td>7</td>
<td>80H</td>
<td>100000000</td>
<td>1</td>
<td>BLACK</td>
</tr>
<tr>
<td>40 dB</td>
<td>6</td>
<td>40H</td>
<td>010000000</td>
<td>2</td>
<td>BROWN</td>
</tr>
<tr>
<td>50 dB</td>
<td>5</td>
<td>20H</td>
<td>001000000</td>
<td>3</td>
<td>RED</td>
</tr>
<tr>
<td>60 dB</td>
<td>4</td>
<td>10H</td>
<td>000100000</td>
<td>4</td>
<td>ORANGE</td>
</tr>
<tr>
<td>70 dB</td>
<td>3</td>
<td>08H</td>
<td>000010000</td>
<td>5</td>
<td>YELLOW (Y)</td>
</tr>
<tr>
<td>80 dB</td>
<td>3,2</td>
<td>0CH</td>
<td>000011000</td>
<td>5,6</td>
<td>Y+ GREEN</td>
</tr>
<tr>
<td>90 dB</td>
<td>3,1</td>
<td>0AH</td>
<td>000010100</td>
<td>5,7</td>
<td>Y+ BLUE</td>
</tr>
<tr>
<td>100 dB</td>
<td>3,0</td>
<td>09H</td>
<td>000010010</td>
<td>5,8</td>
<td>Y+ VIOLET</td>
</tr>
</tbody>
</table>

SELECTING TIME-CONSTANT FOR THE PHASE SENSITIVE DETECTOR (TABLE 5·3)

<table>
<thead>
<tr>
<th>Time constant</th>
<th>Port pin</th>
<th>Capacitance value</th>
<th>Time constant</th>
<th>Mode of use</th>
<th>D-type pin no</th>
<th>Ribbon cable colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>1</td>
<td>6 μF</td>
<td>90 sec</td>
<td>Experimental calibration</td>
<td>11</td>
<td>GREY</td>
</tr>
<tr>
<td>short</td>
<td>0</td>
<td>1 μF</td>
<td>15 sec</td>
<td>Experimental calibration</td>
<td>11</td>
<td>GREY</td>
</tr>
</tbody>
</table>

POWER CONNECTIONS FROM SENCON MODULE TO RESISTOR/RELAY BOX (TABLE 5·4)

<table>
<thead>
<tr>
<th>Voltage</th>
<th>Mode of use</th>
<th>D-type pin no</th>
<th>Ribbon cable colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 V</td>
<td>Ground connection for time-constant</td>
<td>12</td>
<td>WHITE</td>
</tr>
<tr>
<td>0 V</td>
<td>Power for resistor relays</td>
<td>9</td>
<td>GREEN</td>
</tr>
<tr>
<td>+5 V</td>
<td>Power for resistor relays</td>
<td>10</td>
<td>YELLOW</td>
</tr>
</tbody>
</table>
OBCCON: FLOWCHART FOR THE CALIBRATION SOFTWARE MODULE (FIG. 5.9)

START

CLEAR CAL. SET. ACES. EXTERNAL MEMORY
A REFERENCE TONE FROM CALIBRATION DATA
SET MODE TO HANDSHAKE MODE, INITIALIZE HANDSHAKE LINES

SET MODE TO CALIBRATION, FINISHED = NO

OP 'CALIBRATION BEGINS' TO MASTER

CHECK BYTE HAS BEEN SENT

INITIALIZE VARIABLES FOR CALIBRATION

OP GAIN OF YAB TO SENCON

IS MODE SET TO CALIBRATION

NO

OP CALIBRATION VALUE TO MOTOON

READ IN EQUIVALENT O.B.V

STORE CALIBRATION DATA & DECIDE WHETHER TO CHANGE THE SENSITIVITY

IS A SENSITIVITY CHANGE REQUIRED

YES

OP SELECTED GAIN TO SENCON

NO

IS THE CALIBRATION COMPLETE

YES

SET MODE TO END_CALIBRATION

NO

SET SENSITIVITY STORE & OBCC TO 'FINISHED'

OP 'FINISHED MESSAGE TO MOTOON
START

A

OP 'FINISHED' MESSAGE TO SENCON

OP FINISHED MESSAGE TO MASTER

CHECK BYTE HAS BEEN SENT

SET WAIT (TO BE CLEARED BY MASTER)

OP 'CALIBRATION COMPLETE' MESSAGE TO MASTER

CHECK BYTE HAS BEEN SENT

HAS THE WAIT FLAG BEEN CLEARED?

NO

YES

RESET THE WAIT FLAG
MODIFY CALIBRATION DATA

OP 'CONVERT, CALIBRATION COMPLETE' TO MASTER

CHECK BYTE HAS BEEN SENT

WAIT = 0

NO

YES

RESET WAIT_FLAG TO 1

OP 'LIMITING, CAL' MESSAGE TO MASTER

CHECK BYTE HAS BEEN SENT

WAIT = 0

NO

YES

SELECT EXTERNAL PORTS (IOM = 1)
RESET PORT A TO ACT AS AN INPUT PORT

IS THE EXTERNAL INTERRUPT 1
LINE = 0?

YES

NO

RESET WAIT FLAG & ENABLE OPTERRUPT LINES TO NOT ENABLE A SENCON ARE HIGH

END
OBCCON: FLOWCHART FOR THE CALIBRATION MODULE (FIG 5.10)

Do

Clear CAL_BIT which caused the SELF_CALIB procedure to be called & clear IO/M line to select external memory.

Empty the calibration storage array.

Set IO/M to select external ports. Set PORTA to output. Initialise handshake lines, set FINISHED$CALIBRATION to NO Clear BYTE$SENT. Set MODE to CALIBRATION.

Send 'CALIBRATION BEGINS' message to the master.

Initialise the first calibration digit of \( +10\mu F \) to be set on the bridge: ARM = START_DECade (10\( \mu F \) arm), DIGIT = 1, CONTROL = POSITIVE. Select lowest gain of 30dB.

Tell SENCON to set gain value to SENSITIVITY$STORE (30dB)

CAL: Do while mode of operation is set to calibration.

Output calibration digit(s) to MOTCON to set on bridge.

Read in the equivalent off-balance voltage (OBV).

Store OBV in the calibration data table.

Is a sensitivity (gain) change necessary?  
- Yes: Tell SENCON to change to the next gain setting.
- No: Has the calibration finished?  
  - Yes: Set MODE to END_CALIBRATION

Load SENSITIVITY$STORE with FINISHED message for SENCON.
Load OBC value with FINISHED message for MOTCON.
Output FINISHED CALIBRATION message to MOTCON.
Output FINISHED CALIBRATION message to SENCON.
Output FINISHED message to the master (MASCON).
Set WAIT flag ready to transfer the calibration data.
Output CALIBRATION COMPLETE message to the master.

Wait till master has read the calibration data (WAIT = 0)
Reset WAIT flag ready for transfer of converted cal. data.

Execute CONVERT_TABLE_STORE procedure to transform calibration data ready for use during the experiment.
Output CONVERT_CAL_COMPLETE message to master.

Wait till master has read the converted cal. data (WAIT=0)
Reset WAIT flag ready for transfer of limiting cal. data.
Output LIMITING_CAL data available message to master.

Wait till master has read the limiting cal. data (WAIT=0).
Set IO/M to select external PORTs, reset PORTA to input. Wait for ext. interrupt line to go high on resetting PORTA

Set WAIT line & interrupt lines (INT$MOTOR, INT$SEN) high before returning to the main module. Clear IEB0 and IEB1 to ensure there are no external interrupts pending.
5.2 The Capacitance Reader and Motor Control Module (MOTCON)

5.2.1 Circuit Diagrams and Photograph of MOTCON

The first figure, FIG 5.11, is of the electrical circuit for the MOTCON module. The second figure, FIG 5.12 shows the module connections to the other system controllers. A photograph of the PCB is shown in PHOTO 5.2.

5.2.2 Overview of the Functions of MOTCON

The main function of the motor controller is to calculate the on-balance bridge capacitance corresponding to the actual unknown capacitance of the sample cell, and to rebalance the capacitance bridge by moving the mechanical arms as requested. To achieve this objective MOTCON needs to read in, or calculate, the following information: the current capacitance setting on the bridge, the current off-balance capacitance (OBC) from OBCCON, the current on-balance capacitance (calculated from OBC and the current setting on the bridge), and to recognise an interrupt from SENCON which requests a bridge rebalance. Flowchart representations of the main software programming module are given in FIG 5.13 and FIG 5.14

As with all of the controllers a number of parameters and internal registers are initialised on power-up and reset. The slaves are reset by a pulse on their reset line from the master controller (MASCON). MOTCON then waits (as do the other slaves, see 5.1.2) for the master to begin the experiment or to start a calibration.

When the experiment commences MOTCON's first task is to read in the capacitance setting on the bridge via the BCD capacitance output connector (FIG 5.15). This is accomplished by the READ$BCD procedure. All nine arms are read in (range: 100pF - 1aF). Four bits are needed to represent each capacitance digit as a binary number from -1 to X (an 'X' represents 10 on the display). To represent -1 the binary value 1101 (OBH) is used by the bridge. To facilitate the reading of all 36 bits (9 arms X 4 bits) of information the MOTCON module (see FIG 5.11) has three I/O expander chips (Intel 8243). Any one of the three can be accessed by the slave controller enabling its chip select pin. Each expander has four 4-bit ports. Using three
of the ports the first I/O expander (IC2) latches information from capacitance arms: 100pF, 100fF, and 100aF. I/O expander IC3 latches information from the capacitance arms: 10pF, 10fF, 10aF, and I/O expander IC4 latches information from capacitance arms 1pF, 1fF and 1aF. The slave reads in each capacitance digit in turn by selecting one of the I/O expanders, and then outputting and latching the address of one of the external ports. The slave may then reset the address lines to the expander high and input the corresponding capacitance data for that particular arm. Each arm value is read twice and the values are compared. If an error occurs a second attempt is made to read the digit. If the values are still different an error bit is set which may be accessed and interpreted by the master. The capacitance display digits are stored in an array called NAME.

The MOTCON slave will now delay for 10 seconds before causing an external interrupt on OBCCON to request the latest off-balance capacitance. The delay time determines the frequency with which the on-balance capacitance is determined and updated. This delay time can be altered by the operator via a command from the master.

Before reading in the new OBC data the READ$OBC procedure zeros the storage array called CAPS$DIGITS. Data is read in on Port0, four bits indicate the value of the capacitance digit, four bits state the capacitance arm or decade. A ninth bit indicates whether the OBC is positive or negative (see 5.1.2). Each OBC digit is handshaked across using the DATA$REQUEST and DATA$GRANTED lines. MOTCON will continue to request digits until OBCCON sends a 'finished' message, or an 'OBC invalid' message instead of an OBC data byte.

Having obtained the capacitance bridge setting and the current OBC, MOTCON makes a decision as to whether or not to calculate the on-balance capacitance setting, i.e. the actual capacitance of the sample cell. If the OBC digits are valid MOTCON calls the COMPUTE$NEW$BCD procedure, if they are invalid MOTCON sets an error flag to indicate this information to the Master.

The COMPUTE$NEW$BCD procedure will either add or subtract the OBC stored in CAPS$DIGITS from the current display capacitance stored in NAME array. The result, which is the on-balance capacitance, is then
stored in the array called NEW$CAP. If the sign of OBC is positive then a positive capacitance must be added to the present display value to rebalance the bridge. If the sign of the OBC read from OBCCON is negative then the OBC value must be subtracted from the present display value to rebalance the bridge.

When MOTCON has calculated an updated on-balance capacitance it checks to see whether SENCON has requested that the bridge be physically rebalanced. If a rebalance is not required MOTCON repeats the process of reading the display, reading the current OBC and calculating the on-balance capacitance. If SENCON has decided that a bridge rebalance should occur it would have interrupted MOTCON on external interrupt 1 causing the bridge rebalance bit to be set to 'REQUESTED'. MOTCON responds to this request by calling the MOTOR_CONTROL procedure. This complex procedure compares each of the capacitance digits set on the bridge, with each of the new on-balance capacitance digits stored in the NEW$CAP array. If the two corresponding digits are different MOTCON turns on the correct motor to change the display digit to the new value. Details of this important module task are given in 5.2.3. When the bridge has been rebalanced MOTCON clears the bridge rebalancing flags to indicate to SENCON that the task is complete.

MOTCON continues to loop round the main module calculating the sample capacitance each time, and responding to bridge rebalance requests from SENCON.

As stated initially, the most important task carried out by MOTCON is the calculation of the on-balance capacitance which will be equal to the sample capacitance setting when the bridge is on-balance. It is this reading that must be stored along with the sample temperature to be used in calculating the thermal expansion of the sample. This on-balance capacitance (stored in NEW$CAP array) may be requested at any time by a command from the master controller via the serial port interrupt. The master commands the OUT_CAP_VALUE procedure to be called which transfers each on-balance capacitance digit on the serial link (see 5.2.3). The serial port is set as a high priority interrupt and so may occur at any time provided the interrupts are enabled. It is therefore important to turn off the power to the motors whilst a serial interrupt procedure is being
executed. This will prevent errors from occurring should the MOTOR_CONTROL procedure be currently in operation.

A command from the master causes the experiment to finish; this is true for all slave controllers. MOTCON drops out of the DO-loop, named 'COMMENCE', sends a message to the master to indicate that the experiment has finished, and then disables interrupts. MOTCON then starts back at the beginning of the outer program loop labelled 'MAIN_PROGRAM'.

5.2.3 Details of Important Module Tasks

I The Motor Control Procedure

MOTCON controls the motor arms on the bridge through the Auto/Manual Bridge Interface unit (A.M.B.I.)—see 3.2.4. The low nibble of Port2 on the slave controller is used to output a binary code to A.M.B.I. to select the correct motor (e.g. 0000 selects the motor controlling the 1af arm, 1000 selects the motor controlling the 100pF arm). Bit 4 of Port2 selects the motor direction (up or down), bit 5 of Port2 selects the power to the motor (start or stop). Rebalancing of the capacitance bridge is achieved through the MOTOR_CONTROL procedure (FIG 5.16). This procedure is used by both the main module during the experiment and the calibration module. The reason for this is to conserve program memory space and to simplify the programming task (i.e. only one motor control procedure to be written and tested instead of two). During the experiment the MOTOR_CONTROL procedure will only be called if an external interrupt from SENCON has set the BRIDGE$REBALANCING bit to request a rebalance. MOTCON then takes bit P3.5 on the slave controller (the BRIDGE$IS$BALANCING line) high to indicate to SENCON that a rebalance is underway. At the completion of the MOTOR_CONTROL procedure MOTCON takes the BRIDGE$IS$BALANCING line low to inform SENCON that the task is now complete.

The MOTOR_CONTROL procedure begins by setting a bit to inform the master that a rebalance is taking place. This bit will be cleared next time through the software loop after a new on-balance capacitance has been calculated. MOTCON will then look in turn at
each capacitance digit on the bridge, and compare it with the on-balance capacitance digit. If the two digits are not equal, MOTCON calculates in which direction the motor arm must be moved to set the display digit to the on-balance value, and then turns on the power to the motor.

Whilst the motor is turned on, and the moving arm is causing the displayed capacitance digit to change, MOTCON continually reads in the display digit value via the I/O expanders, and halts the motor when the display digit is equal to the on-balance digit. To prevent error values from being saved as the binary output flicks from one digit to the next, the procedure reads a bridge value three times with a delay of 0.1 seconds between each reading, and only accepts a value as being 'true' if these three readings are the same. Having rebalanced the first arm on the bridge MOTCON goes on to compare the next display digit and on-balance capacitance digit, and repeats the procedure until the bridge display has been reset to the new on-balance capacitance. At this point the motor select code is set to '1111'. This motor does not exist, and so if the motor power is turned on by accident the motors will not respond. Finally, the bridge rebalancing bits are cleared to indicate to SENCON that the rebalance is complete.

Two extra sections of code were incorporated in the procedure to overcome a hardware problem, and to deal with a potential mechanical problem; these are detailed in section 5.2.4.

II Transferring the Capacitance Data to the Master

The on-balance capacitance is continuously computed throughout the experiment. This capacitance corresponds to the capacitance of the sample cell, and from this measurement, having allowed for any expansion of the copper making up the electrodes and often the body of the cell, the thermal expansion of the sample within the cell can be determined.

The master controller can access this data by sending a command to the motor controller via the serial port. The OUT_CAP_VALUE procedure will be called which sends the capacitance digits to the master one byte at a time. The digits are stored in the NEW$CAP
array. The first byte contains the digit for the 1aF arm, the ninth
array byte contains the on-balance digit for the 100pF arm. A tenth
byte of information is included containing error information which
can be interpreted by the operator in the following manner:-

Tenth data byte sent - 0000 0000 - Capacitance result is error free.

0001 0000 - Invalid capacitance display digits read from the bridge.

0010 0000 - Invalid off-balance capacitance read from OBCCON.

0100 0000 - Bridge rebalance in progress, present capacitance data
may not therefore have been recently updated.

1000 0000 - Program has just commenced and valid data is not yet
available.

0111 0000 - Indicates three of the above error conditions present.

If the capacitance data is transferred successfully an eleventh byte
is sent to the master which contains the check-sum information (i.e.
the sum of the first ten array bytes transmitted) to be used by the
master to check for data errors. Before sending each byte of data
MOTCON waits for a 'SEND_VALUE' message from the master. If this
message is not received, or if MOTCON has to wait longer than 5
seconds for a reply, then the slave controller assumes an error has
occurred with the data transmission. An error flag is set which
enables the slave to exit from the OUT_CAP_VALUE procedure and to
return to the main program without sending any further data bytes.

5.2.4. Problems Encountered: Diagnostic Tests and Module Emulation

1 Calculation of the On-Balance Capacitance

The on-balance capacitance digits are calculated by adding or
subtracting the OBC digits stored in the CAPS$DIGITS array from the
BCD display digits stored in the NAME array. The result is placed in
NEW$CAP array. During an experiment acceptable display digits range
from 0 - 9, (-1 and X being considered as error values).
If the OBC is positive its value is added to the display capacitance byte-wise across the array, starting with the least significant byte corresponding to the 1aF arm. If on addition there is a carry, a carry flag is set. The carry is then included in the addition of the next two array bytes (e.g. carry-out from addition of the 10aF digits is added to the addition of the 100aF digits). This carry flag is called 'CARRY' to distinguish it from the MCS-51's own internal carry flag.

When the sign of the OBC is negative care must be taken in subtracting the CAPS$DIGITS array from the NAME array bytes. Again the carry technique can be used, but in this case the carry flag is set by the software if there is a borrow required on the subtraction of the previous two digits (i.e. if the result of the subtraction is less than zero a borrow is required from the next significant digit).

The problem that occurs is that PL/M does not have negative numbers, thus an inequality such as 'IF RESULT < 0' cannot be used.

For example, for a byte operation in PL/M:

\[
\begin{array}{c}
0000 0000 \\
0000 0001 - \\
1111 1111 \\
\hline
\end{array}
\]

A simple solution to the problem was found to be effective. The largest digit stored is 9, thus the largest negative number that the calculation procedure may have to deal with is \(0 - 9 = -9\). Therefore adding 9 to both sides of any software test involving an inequality will prevent an error from occurring and produce a correct boolean result.

II Rebalancing The Capacitance Bridge

The digit range on the bridge is from \(-1, 0, 1, \ldots, 8, 9, X\). The X (\(+10\)) position is only used by the operator when checking the standard bridge capacitance against a known external standard capacitor. The \(-1\) position is only used by the automation system during a calibration when calibration data for both positive and negative OBCs is collected. Hence, during an experiment if MOTCON reads an X or a \(-1\) from the bcd output an error has occurred. Continued motion of the motor may damage the capacitance bridge as there will be no
further travel in the arm. This error situation is highly unlikely, but may occur should a bridge arm overstep its mark and miss the digit at which it was to stop.

The MOTOR_CONTROL software will deal with this problem by stopping the power to the motor and then changing the motor direction. If the bridge digit = X then the motor direction is set to down. If the bridge digit = -1 and the resulting display is not to be -1 (i.e. this is not a calibration) then the motor direction is set to up. The motor power is turned on again and a delay of 0.5 seconds is allowed before the display digits are re-read; this gives the motor arms time to start to move away from the error position.

A second software problem, which was difficult to track down and isolate, was the production of an incorrect reading at the same position on several arms on the bridge. As a bridge arm changes from one position to the next so the binary bits on the bridge output flick on and off. However, it was found that for a short period of time between binary digits 2 (0010) and 3 (0011) the value 1011 (or 0BH) was read from the bridge. At no time between digits 2 and 3 should this value occur. It was confirmed that this incorrect reading was coming directly from the bridge itself, and not from the control electronics, by viewing the output from the bridge on a logic analyser. As the bridge electronics could not be investigated this problem was overcome by using a software patch to mask the most significant bit of the nibble between digits 1 and 5.

e.g. DIGIT = 1011, MASK = 0111 —> RESULT = DIGIT AND MASK = 0011

It was necessary to do this as the code 1011 is used by the bridge to represent the digit -1, and as previously explained the MOTOR_CONTROL procedure regards a digit of -1 as an error during the experiment, and changes the motor direction to up. Thus if the motor arm was decreasing from 3 to 1 and an incorrect reading of 1011 was read between digits 3 and 2 the motor arm would be set to up. The arm would become trapped within a loop between X and 2 and never reach its final destination.

A further complication, which was overcome by using a number of test procedures with variable input parameters, was that of timing. At
what speed should the motor arms be changing between digits? How long a delay should there be between re-reading a BCD bridge digit?

Transient values may be present on the bridge being output for a fraction of a second whilst a digit change occurs. To avoid these transient values, three readings are taken at 0.1 second intervals, and the result is only accepted if all three readings are the same. This approach will work well providing the arm is travelling slowly enough. If each digit change takes 0.5 seconds, and it takes a minimum of 0.3 seconds to detect that digit change, a number may be easily missed. A compromise of 2 seconds per digit change was found to work well. Having detected a digit, a delay of about 0.1 seconds was required before turning off the motor power to allow the capacitance digit to be clearly in view on the visual capacitance display for the benefit of the operator.

5.2.5 The Calibration Software

MOTCON is placed in the calibration mode of operation by a command from the master which causes the SELF-CALIB procedure to be called from the CALIBRATE$MODULE. Flowcharts for the calibration module are given in FIG 5.17 and FIG 5.18. During calibration MOTCON enables external interrupt 1 as an edge triggered interrupt. The interrupt pin is controlled by OBCCON. As described in (5.1.4) OBCCON will interrupt MOTCON when there are calibration digits to be set on the bridge. When an interrupt occurs the CAL$INT procedure is vectored to, here the INT$BIT$SET variable is set to 1. After initialisation MOTCON remains within a DO-loop labeled CAL whilst the mode is set to calibration. Each time round the loop, MOTCON tests the INT$BIT$SET bit to see if OBCCON wants calibration digits set on the bridge.

If OBCCON is outputting a positive OBC on the bridge, only one digit needs to be set by MOTCON. However, when negative OBBCS are set on the bridge two digits need to be set, one of which is always -1.

For example:

Positive bridge setting of $+10\text{fF}$ - 0 0 0 0 1 0 0 0 0 aF  
Negative bridge setting of $-10\text{fF}$ - 0 0 0 -1 9 0 0 0 0 aF
In the calibration mode the sign line from OBCCON is used to indicate whether the calibration data consists of one or two digits. On recognising that OBCCON has calibration digits to send, MOTCON responds by first reading the current BCD bridge display utilising the READ$BCD procedure from the main module. The digits are stored in an array called NAME. The OUTPUT array which will contain the calibration information is then zeroed. This happens each time new calibration data is to be set on the bridge. MOTCON now reads in the data from Port0 using the DATA$GRANTED and DATA$REQUEST handshake lines. The handshake process follows that used for the READ$OBC procedure in the main module. The digits are stored in an array. The position of each digit within the array corresponds to its arm and decade value.

Once data has been transferred between OBCCON and MOTCON the NAME array contains the current capacitance setting as displayed on the bridge, and the OUTPUT array contains the calibration digits which are to be the new capacitance setting on the bridge. The procedure called MOTOR_CONTROL is then called to update the bridge display. There is a delay of one minute after updating the display, to give the filter and detection system time to respond to the change in capacitance and for oscillations to die away. This is necessary because the PSD is set to a long time-constant during a calibration which slows down the system response time. MOTCON then completes the second half of the data handshake with OBCCON, and clears the INT$BIT$SET bit ready for next time. MOTCON will continue to execute the 'DO-loop' labeled CAL until OBCCON interrupts the program again requesting a new value to be set on the bridge.

At the end of the calibration OBCCON sends a 'finished' message to MOTCON instead of a calibration digit. On receiving this message MOTCON sets the mode to END_CALIBRATION and leaves the CAL loop. External interrupt 0 is disabled, the slave exits the SELF_CALIB procedure, and returns to the main program module. In comparison with the OBCCON module, MOTCON's calibration task is straightforward: the slave simply sets the calibration data on the bridge at OBCCON's request.
(PHOTO 5.2) THE CAPACITANCE READER AND MOTOR CONTROL MODULE: MOTCON
CIRCUIT DESIGN FOR THE MOTOR CONTROLLER MODULE SHOWING CONNECTIONS TO OTHER CONTROLLERS (FIG 5.12)
MOTCON: FLOWCHART FOR THE MAIN SOFTWARE MODULE (FIG. 13)

START

INITIALISE REGISTERS AND PARAMETERS

HAS THE EXPERIMENT COMMENCED?

YES

NO

HAS A CALIBRATION BEEN REQUESTED?

YES

NO

SEND EXPERIMENT COMMENCED MESSAGE TO MASTER & ENABLE INTERRUPT FROM SENCN

HAS THE EXPERIMENT FINISHED?

YES

NO

DISABLE ALL INTERRUPTS

TELL MASTER "EXPERIMENT FINISHED"

INPUT THE CURRENT CAPACITANCE

DISPLAY SETTING FROM THE BRIDGE

DELAY FOR 10 SECONDS BEFORE REQUESTING THE OFF-BALANCE CAPACITANCE

INPUT THE CURRENT OFF-BALANCE CAPACITANCE

DIGITS FROM OMCON

IS THE CURRENT OFF-BALANCE CAPACITANCE VALID?

YES

NO

CONTROL THE MOTOR ARMS ON THE BRIDGE TO RE-BALANCE TO LATEST OBC

DELAY FOR 15 SECONDS TO ALLOW THE BRIDGE TO STABILISE

COMPUTE WHAT THE LATEST ON-BALANCE CAPACITANCE SHOULD BE

SET BIT TO INDICATE TO MASTER THAT OBC IS NOT VALID

HAS SENCN REQUESTED THAT THE BRIDGE SHOULD BE REBALANCED?

YES

NO
**MAIN PROGRAM**: Do while '1' (i.e. go forever)

Initialise delay parameters, ensure power to the motors off. Initialise handshake lines to OBCCON: DATA$GRANTED is set to act as an input line, DATA$REQUEST is cleared to act as an output line. Set interrupt line out from MOTCON to OBCCON (READY$FOR$DATA). Initialise the bridge rebalancing bits. Set up timer 1 to auto-reload mode and turn timer 1 on. Set up timer 1 counter with auto-reload value to give a baud rate of 3125 bit/second, (timer 1 is used by the serial port as a baud rate generator).

Set serial port to high priority and other interrupts to low priority. Set up serial port register to select 9 bit UART mode with variable data rate, set SM2 to disable reception of serial data bytes (i.e. 9th bit 0), REN is set to enable serial reception.

Zero the arrays used to store the display capacitance, off-balance capacitance, and on-balance capacitance. Enable serial port interrupt. Clear EXPT$IS$RUNNING bit to prevent the experiment from commencing. Clear CAL bit to prevent the calibration procedure from running. Set bit to indicate to master that data has yet to be collected. Enable interrupts.

Do while the experiment has not commenced.

<table>
<thead>
<tr>
<th>No</th>
<th>Has the master requested a calibration?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Carry out calibration (call SELF_CALIB)</td>
<td></td>
</tr>
</tbody>
</table>

Send a message to the master: 'experiment has commenced'

Enable external interrupt 1 so that SENCON can interrupt OBCCON and cause a bridge rebalance.

**COMMENCE**: Do while the experiment is running.

Read in current display capacitance from bridge

Delay for 10 seconds before requesting off-balance capacitance from OBCCON (master can change delay).

Interrupt OBCCON, and input current off-balance capacitance (OBC) digits from OBCCON.

<table>
<thead>
<tr>
<th>No</th>
<th>Is the current OBC valid?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set bit to show the master that OBC is not valid.</td>
<td>Compute on-balance capacitance by adding/subtracting OBC from the current capacitance display.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No</th>
<th>Has SENCON requested a bridge rebalance?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Control motor arms to rebalance the capacitance bridge i.e. set bridge to computed on-balance capacitance value. Delay for 30 seconds after rebalancing to allow for the system to 'settle'.</td>
<td>Tell master the experiment has finished. Disable all interrupts.</td>
</tr>
</tbody>
</table>
FIG 5-15
TERMINAL IDENTIFICATION AT B.C.D CAPACITANCE OUTPUT
CONNECTOR (A J13, REAR PANEL OF BRIDGE.)
Set bit to indicate to master: bridge rebalance in progress

MOT$CON$LOOP: Do I = 0 to 8 (compare each pair of digits).

Y Is the capacitance display digit = on balance digit? N

  Do
      Output code to select motor arm number I
  N

      Is on balance digit > capacitance display digit Y

    Select motor arm down         Select motor arm up

    Turn on power to the selected motor.

  Do while on balance digit <> capacitance display digit

      Do T = 0 to 2 (read in the display digit 3 times).
      Output code to select b.c.d. digit to be read in.
      Latch address code on expander chip (PROG = 0)
      Set low nibble of PORT1 to read in b.c.d. digit.

      N Is the capacitance display digit between 1 & 5 Y

    Input b.c.d. digit & store in SAVE array at index T.
    Input b.c.d. digit & mask the m.s.b. of the nibble to prevent an error.
    Disable chip select & reset PROG on expander chip.
    Delay 0.1s to prevent transient readings being saved

  No Are the 3 display digit readings the same? Yes

    Save display digit at index I of the digit array.

    Is the saved digit 10 or is the saved digit -1 & the MODE not calibration Y

      Do
          Turn off the power to the motor arm.
      N

          Is the saved digit 10? Yes

          Select motor arm up         Select motor arm down

          Delay 1s, turn on power, delay 0.5s

        Delay 0.1s to ensure new digit is in place and then turn off power to the motor.

    N

Select motor arm "F" as protection (motor does not exist).
Clear BRIDGE$REBALANCING$BIT as bridge has been rebalanced.
Clear BRIDGE$IS$BALANCING$BIT to tell SENCON rebalance ends
Disable all interrupts during initialisation. Set PORT0 & SIGN to read in calibration digits from OBCCON. Clear INT$BIT$SET -set when OBCCON has cal. digits to send. Set MODE to CALIBRATION. Re-enable interrupts.

CAL: Do while the MODE of operation is set to calibration.

Has OBCCON cal. digits to send (INT$BIT$SET = 1)?

Y

Do

Read current capacitance display from bridge & store.

Zero OUTPUT array which will contain calibration data

Do R=0 to 1

Request the calibration digit from OBCCON.

Wait till data has been sent (i.e. DATA$GRANTED).

Store calibration value (both digit and decade).

Store sign bit: is this the +ve or -ve calibration

No

Has OBCCON finished the calibration?

Yes

Do

Separate decade and digit information

Store digit in OUTPUT array according to the decade position.

Set R=2 to drop out of the inner loop.

No

Is there 1 cal. digit to I/P?

Yes

Set R=2

N

Is the 1st cal. digit?

Y

Finish handshake: send data received & wait for confirm.

Output the calibration capacitance digits to be set on the bridge, (call MOTOR_CONTROL procedure).

Delay for 1 minute to allow system to settle.

Finish handshake: send data received message and wait for acknowledgement from OBCCON.

Clear INT$BIT$SET ready for the next interrupt.

N

Clear CAL BIT before returning to the main program to prevent SELF_CALIB from being recalled.

Disable external interrupt 1.
5.3 The Sensitivity Control Module (SENCON)

5.3.1 Circuit Diagrams and Photograph of SENCON

The first figure is of the electrical circuit for the SENCON module (FIG 5.19) and the second figure shows the module's connections to other system controllers (FIG 5.20). A photograph of the PCB is shown in PHOTO 5.3.

5.3.2 Overview of the Functions of SENCON

The function of the third slave, SENCON, is to monitor the rate of change of the off-balance voltage from the capacitance bridge. SENCON accesses the OBV by sharing the 12-bit A/D converter with OBCCON. SENCON samples the OBV every 30 seconds. When SENCON has obtained 10 readings it analyses the data using a least squares fit procedure to calculate the gradient and the intercept. The controller utilises this information to predict when a bridge re-balance will be necessary, and to decide whether to change the system sensitivity. If the gradient is too sharp, SENCON will decrease the sensitivity of the low-noise amplifier by 20dB. If the gradient is too shallow, SENCON will increase the sensitivity by 20dB. Before performing this task SENCON informs OBCCON of the change. Should SENCON evaluate that the next bridge re-balance will occur in less than five minutes, it will interrupt MOTCON to request a bridge rebalance, and then inform OBCCON of the rebalance. The controller can also change the time-constant of the PSD from long (90 seconds) to short (15 seconds) at the master's request. Two flowchart representations of the main software programming module (MAIN$MODULE) are given in FIG 5.21 and FIG 5.22.

The decisions to rebalance the capacitance bridge and to change the sensitivity are made by the STORE_DATA_AND_ANALYSE procedure using the current gradient and intercept information, and the stored limiting calibration data. The range of the A/D is ±4096 bits, from on-balance to off-balance (±3V). For the arm settings 1aF-1fF, 10 digits off-balance is equivalent to 1 Volt off-balance. For the 10fF arm, 1 digit off-balance (10fF) is equivalent to 1 Volt off-balance. The maximum off-balance setting is ±30fF (±3Volts) on the lowest gain of 30dB. Consider then, during the experiment, the...
detection system must be sensitive enough to detect the small capacitance changes, but not so sensitive as to send the detection instrument from on-balance to off-balance within a few minutes, making continual re-balancing necessary. Therefore an optimum compromise must be obtained. Having carried out a number of manual experiments investigating the change of capacitance with temperature and time, and in view of the sensitivity of the detection electronics, the author decided on an 'ideal' state. The ideal would be for a rate of change of OEV requiring the bridge to rebalance once every hour. The acceptable limits would lie between an upper gradient whereby a bridge re-balance would occur once every 30 minutes, and a lower limiting gradient whereby a re-balance would occur once every 90 minutes.

At the end of the calibration task the master sends SENCON the limiting calibration data corresponding to ±9.5 digits off-balance for each of the capacitance arms 1aF-1ff and for ±3 digits (±30fF) for the 10fF arm. For the lower arms, if the capacitance approaches or exceeds 9.5 digits off-balance it is necessary to re-balance the bridge.

Now:  
10 digits off-balance = 1 Volt off-balance  
9.5 digits off-balance = 0.95 Volts off-balance

The range on the A/D, see FIG 5.23, has been set such that:

4096 bits = 3 Volts
Thus:  
1297 bits = 0.95 Volts
Therefore:  
1297 bits = 9.5 digits off-balance

For the ideal state: 1 bridge re-balance every 60 minutes  
i.e. from on-balance to off-balance in 60 minutes

Ideal rate of change: 1297 bits/hour = 0.36 bits/second
Upper limiting gradient: 1297 bits/30 minutes = 0.72 bits/second
Lower limiting gradient: 1297 bits/90 minutes = 0.24 bits/second
The least squares fit procedure will calculate the gradient of the best fit line through the data points. If the calculated gradient lies between 0.72 - 0.24 bits/second then SENCON will not change the system sensitivity. For a gradient greater than 0.72 bits/second SENCON will decrease the system gain by 20dB (minimum setting 30dB). For a gradient less than 0.24 bits/second SENCON will increase the system gain by 20dB up to a maximum setting of 90dB.

Using the limiting calibration data point SENCON extrapolates the best fit line to calculate at what time the OBV will be expected to exceed the limiting value (9.5 digits off-balance). The EVALUATED_TIME_USING procedure achieves this by using the equation for a straight line:

\[ Y = mX + c \]

\( m \) = gradient, \( c \) = intercept, \( Y \) = limiting calibration data point, \( X \) = time at which OBV will exceed this limiting value assuming the gradient remains constant.

Each set of readings takes 4.5 minutes to obtain (30 seconds between each of the 10 readings). Therefore if a bridge rebalance is estimated as being necessary within the next 5 minutes, SENCON interrupts MOTCON to request a bridge re-balance before taking the next set of readings. Should the capacitance be changing very rapidly with the gain already on its lowest value of 30dB the limits can be increased from ±9.5 fF off-balance to ±30fF off-balance to decrease the frequency of bridge re-balancing. Care must be taken to use the correct limit: if the gradient is positive the limit corresponding to +9.5 digits is used, if the gradient is negative the limit corresponding to -9.5 digits is used. To simplify the controller's calculations the number range from the A/D is shifted by the slave, see FIG 5.24.

From the A/D output
Positive voltage range: 0 - 4095 000H - FFFH polarity bit = 0
Negative voltage range: 0 - 4095 000H - FFFH polarity bit = 1

Inside the controller
Positive voltage range: 4096 - 8191 1000H - 1FFFH
Negative voltage range: 0 - 4095 0000H - 0FFFH
Although the 8751 has its own internal timer/counters these are suited to the measurement and production of short delays of the order of microseconds or milliseconds. Hence the use of timer 1 as a baudrate generator for the serial port on each controller. For SENCON a time delay of 30 seconds is also required between taking each reading of the OBV. This delay was achieved via hardware rather than the software. A precision timer (ZN1034) and a monostable chip (74121) were used to provide a 1 second trigger pulse on external interrupt 1 of SENCON every 30 seconds. The precision timer produces a delay of 28.88 ± 0.05 seconds which activates a trigger pulse of 1.14 ± 0.05 seconds from the monostable to INTI giving a total delay of 30.02 ± 0.10 seconds. Exact timing and the errors involved were investigated by setting up the delay circuit on a separate breadboard and using an electronic counter to monitor the delay times. Capacitance values were varied to obtain the exact time delay.

Each time an interrupt comes in on INTI the program vectors to the READ_A_D procedure. The A/D converter is located on the OBCCON module, but is shared by SENCON and OBCCON. SENCON only requires access to it once every 30 seconds. When SENCON wishes to access the A/D it informs OBCCON by taking the IN_USE line low (PB2 on the external RAM chip). Each time OBCCON enters its READ_A_D PROCEDURE it polls this line before accessing the A/D. To ensure that OBCCON has had time to complete any current use of the converter and to recognise that the IN_USE line is low SENCON waits for 0.06 seconds before accessing the A/D. This wait time was determined by using the ICES1 to measure the time taken by OBCCON to execute the A/D accessing code. Interrupts are disabled during execution of the A/D accessing code to prevent changes to the wait time. The maximum time taken to read the A/D once was 0.04 seconds. Setting the wait time to 0.06 seconds ensured that SENCON did not try to read the A/D at the same time as OBCCON.

The sensitivity of the low-noise amplifier on the signal detection and filter system can be changed by switching in external resistors into the amplifier's gain circuit. The exact values of the resistors required to produce the amplifier gains 30dB - 100dB were determined by A.R.Khan. The author has rebuilt the resistor/relay network. Under SENCON's control this network selects specific resistors to achieve the chosen gain value. A circuit diagram of the
resistor/relay network can be found in FIG 3.9. Binary bit patterns output on external PortA (GAIN$PORT$A) of the 8156 RAM chip are input to the D-type connector on the box to turn the relays on and off and thus select the correct gain position, (TABLE 5.2). The slave can select any gain between 30dB and 100dB, but the microcontrol system only uses the gain settings: 30dB, 50dB, 70dB, and 90dB. The binary patterns are stored within the slave's ROM in an array called GAIN_ARRAY. It is then a simple matter to output the selected array value on GAIN$PORT$A to change the system sensitivity.

Before changing the gain, SENCON informs OBCCON of the new value. SENCON uses the low nibble of PortI to send the gain data to the external RAM chip on the OBCCON module. As explained in (5.1.3), data is handshaked safely to the RAM chip using the BUFFER$FULL and STROBE lines which are connected respectively to P3.5 and P1.6 on SENCON. Having informed OBCCON of what the new gain value will be, SENCON makes OBCCON wait whilst it sets the new value via the relay box. It then completes the handshake and allows OBCCON to continue taking readings using the new gain setting.

On completion of the STORE_DATA_AND_ANALYSE procedure SENCON may decide that a bridge re-balance is necessary or that a gain change is necessary. In both cases the capacitance bridge must be re-balanced. SENCON interrupts MOTCON on INTI using its INT$MOTOR line (P1.4). MOTCON responds by taking the BRIDGE$IS$BALANCING line (P1.5) high whilst it controls the motor arms to bring the OBV back towards zero. SENCON waits until MOTCON indicates that the re-balance is complete by taking the BRIDGE$IS$BALANCING line low again. This is all achieved by the BRIDGE_REBALANCING procedure.

SENCON also controls the external time-constant for the PSD. There are two choices of time-constant either "long" or "short". In general long time-constants are used during calibration and short time-constants are used during the experiment. However the Master can command SENCON to switch from one to the other by calling the CHANGE_TIME_CONSTANT procedure. The external time-constant circuit includes two capacitors: 1µF giving a 15 second time-constant and 6µF giving a 90 second time-constant. The capacitors reside in the resistor/relay box and are switched using a double pole reed relay. Pin PBl on the external RAM selects between a short (PBl=0) or long
(PB1 =1) time-constant (TABLE 5.3). Power connections for the resistor/relay box are given in TABLE 5.4.

As with the other slaves the master sends commands to SENCON via the serial port. A command value is received as a number in the range 0 to 7. SENCON then carries out the specific action associated with the command number. This is achieved in software by using a 'DO-CASE' block to select one of a number of cases, or in this instance one of a number of commands:

<table>
<thead>
<tr>
<th>COMMAND</th>
<th>EFFECT</th>
<th>RESULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CAL_BIT -1</td>
<td>Calibration begins</td>
</tr>
<tr>
<td>1</td>
<td>CALL CHANGE_TIME_CONSTANT</td>
<td>Master selects time-constant</td>
</tr>
<tr>
<td>2</td>
<td>EXPT$IS$RUNNING -1</td>
<td>Experiment commences</td>
</tr>
<tr>
<td>3</td>
<td>EXPT$IS$RUNNING - 0</td>
<td>Experiment finishes</td>
</tr>
<tr>
<td>4</td>
<td>CALL LIMITS</td>
<td>Limiting calibration data read from the Master</td>
</tr>
<tr>
<td>5</td>
<td>CALL CLEAR_OVERRIDE</td>
<td>SENCON resumes control of the system sensitivity</td>
</tr>
<tr>
<td>6</td>
<td>CALL INITIALISE_CHANGE_GAIN</td>
<td>Master takes control of the system sensitivity</td>
</tr>
<tr>
<td>7</td>
<td>CALL GAIN_VALUE_TO_MASTER</td>
<td>Tell Master the current gain</td>
</tr>
</tbody>
</table>

To summarize, SENCON continually monitors the rate of change of the off-balance voltage. Using this information SENCON makes decisions as to when to re-balance the capacitance bridge or when to change the system sensitivity. SENCON initialises a re-balance by interrupting MOTCON. SENCON changes the gain by switching different resistor values into the low noise amplifier, and informs OBCCON of the change. The PSD time-constant is also controlled by the sensitivity controller (SENCON).
5.3.3 Details of Important Module Tasks

I Least Squares Fit Procedure

This procedure evaluates the best line fit taking the 10 points in the A-D_DATA$ARRAY as the Y coordinates and using the constant variables in the X_VALUE array as the X coordinates. The X axis refers to time in units of 30 seconds. The Y axis refers to the off-balance voltage from the A/D converter. This procedure evaluates the gradient and the intercept on the Y axis for the best line fit.

The calculations involve multi-byte arithmetic, and so it has been necessary to write multiply, subtract and divide procedures which handle three and four byte numbers. As stated earlier ten readings are taken with data being sampled every 30 seconds. Consequently the X values are known and remain constant. This enables any sums within the calculation which only involve X variables to be substituted by a constant number in the equation. Finally to limit the X_VALUE array to single bytes the time variables have been divided by 30 giving:

<table>
<thead>
<tr>
<th>Time (seconds)</th>
<th>X_VALUE (x_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>90</td>
<td>3</td>
</tr>
<tr>
<td>120</td>
<td>4</td>
</tr>
<tr>
<td>150</td>
<td>5</td>
</tr>
<tr>
<td>180</td>
<td>6</td>
</tr>
<tr>
<td>210</td>
<td>7</td>
</tr>
<tr>
<td>240</td>
<td>8</td>
</tr>
<tr>
<td>270</td>
<td>9</td>
</tr>
</tbody>
</table>

To develop a least squares fit procedure the following algorithms must be coded:

\[
\text{GRADIENT} = \frac{(XY) - XY}{(X^2) - X^2} \quad \text{INTERCEPT} = \frac{(X^2)Y - X(XY)}{(X^2) - X^2}
\]

Where:

\[
(XY) = \frac{1}{n} \sum_{i=1}^{n} x_i y_i \quad X = \frac{1}{n} \sum_{i=1}^{n} x_i \quad Y = \frac{1}{n} \sum_{i=1}^{n} y_i
\]

\[
(X^2) = \frac{1}{n} \sum_{i=1}^{n} x_i^2 \quad x^2 = X \times X \quad XY = X \times Y
\]

n - number of data points (= 10)

By multiplying denominator and numerator by \(n^2\), and equating the following:

\[
\text{SUMS1} = \{x_i y_i\} \quad \text{SUMS2} = \{y_i\} \quad (x_i - 45) \quad ((x_i)^2 - 2025) \quad (x_i x_i) = 285
\]
GRADIENT = 10 * SUM$1 - 45 * SUM$2

\[ (10 \times 285) - 2025 \]

INTERCEPT = (285 * SUM$2) - (45 * SUM$1)

\[ (10 \times 285) - 2025 \]

Simplifying:

GRADIENT = (2 * SUM$1) - (9 * SUM$2)

165

INTERCEPT = (19 * SUM$2) - (3 * SUM$1)

55

The solution to this problem is flowcharted in FIG 5.25 where the following intermediate sums are calculated:

\[
\text{GRADIENT} = \text{SUM$3} - \text{SUM$4} \quad \text{SUM$5}; \quad \text{GRADIENT'} = 10 \times \text{SUM$5} - \text{SUM$6}
\]

\[
\text{ONE\_SIX\_FIVE} \quad 165 \quad 165 \quad 165
\]

\[
\text{INTERCEPT} = \text{SUM$7} - \text{SUM$8} \quad \text{SUM$9}; \quad \text{INTERCEPT'} = 10 \times \text{SUM$9} - \text{SUM$10}
\]

\[
\text{FIFTY\_FIVE} \quad 55 \quad 55 \quad 55
\]

The gradient and intercept are multiplied by ten before carrying out the final division to provide an extra significant digit in the result. In the event of a zero gradient being calculated the result is increased by 1. This will not effect any decisions made by SENCON, but will prevent any errors due to division by zero during the STORE\_DATA\_AND\_ANALYSE procedure.
III Multiply Procedure

This procedure multiplies a single byte (OP1) by a multibyte array (OP2), and places the result in the OP3 array (FIG 5.26). The OP2 array may be 2, 3, or 4 bytes big, this information is passed as a parameter to the procedure. The procedure works using the method of long multiplication by taking the least significant byte of information first, OP2(0), and multiplying by OP1. Any carry from the first multiplication is added in to the second multiplication, and so on working towards the most significant digit. If there is a carry out from the final multiplication (OP1 X OP2(3)), the result is placed in the most significant byte of the result array, OP3(4).

III Subtract Procedure

This procedure (FIG 5.27) calculates the difference between two multi-byte numbers stored in arrays OP1 and OP2 and places the result in the OP3 array. The procedure also calculates whether the result of the subtraction produced a positive or a negative result. This information is stored in the global variable RESULT$NEGATIVE, and is used during the least squares fit procedure to determine whether the gradient is negative or positive. The importance of the information is in determining which calibration data limit to use in estimating the time to the next bridge re-balance:

gradient = positive, the positive off-balance limit is used;
gradient = negative, use the negative off-balance limit.

The size of the input arrays is indicated to the subtract procedure by an input parameter. First the procedure evaluates whether the result of the subtraction will be positive or negative. This is done by comparing successive bytes starting with the most significant bytes say OP1(3) and OP2(3). If OP2(3) > OP1(3) the result is negative, if OP2(3) < OP1(3) the result is positive, if OP2(3) = OP1(3) compare the next two significant bytes (OP2(2) and OP1(2)). If all bytes are equal then the result is zero and considered positive. For negative results the array order is switched i.e. OP2-OP1, so that the subtraction produces a positive difference. The subtraction is carried out starting with the least significant bytes and working towards the most significant bytes. If a 'borrow' is required a carry flag is set in order to 'pay-back'
the 'borrow' to the next significant byte. For example:

**DECIMAL SUBTRACTION**

<table>
<thead>
<tr>
<th>27</th>
<th>(borrow 10 and add to 7, then pay it back to 1, 1 goes to 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- 19</td>
<td></td>
</tr>
<tr>
<td>-- 08</td>
<td></td>
</tr>
</tbody>
</table>

**MULTI-BYTE SUBTRACTION**

<table>
<thead>
<tr>
<th>FA 1B</th>
<th>(borrow 100H and add to 1BH, then pay it back to E2H, E2H goes to E3H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E2 1D</td>
<td></td>
</tr>
<tr>
<td>17 FE</td>
<td></td>
</tr>
</tbody>
</table>

Now in PL/M51 a byte operation such as 1BH - 1DH gives the result FEH (the 2s complement value for -2), and the actual result of the operation on borrowing 100H gives: 11BH - 1DH = FEH. Thus no alteration need be made to the result of the single byte subtract operation even if a 'borrow' is employed.

The procedure utilises its own carry flag rather than use the MCS-51's internal flags. To produce efficient code from PL/M51 source statements the compiler performs extensive optimisation of the machine code. This means that the exact sequence of machine code produced to implement a given sequence of PL/M51 source statements cannot be predicted. Consequently the state of the hardware flags cannot be predicted for any given point in the program.

**IV Divide Procedure**

The divide procedure uses the method of multiple subtraction to divide the SUM array by the DIVISOR array placing the result in the RESULT array. Obviously the divide procedure would become very time consuming if it consisted of thousands of multiple subtractions carried out using the subtract procedure. To increase the speed of the code a number of mini-divisors are created within the procedure: 1024, 256 and 16 times the original divisor. These mini-divisors are obtained by shifting the original several bits to the left to effect a multiplication by a power of two. The divide procedure carries out multiple subtraction using each mini-divisor in turn. Each result is stored in an accumulative total before finally using the original divisor on the remainder of the SUM. This method reduces the time taken for code execution from minutes to seconds. Details of the procedure are given in a flowchart (FIG 5.28 and FIG 5.29).
5.3.4 Problems Encountered: Diagnostic Tests and Module Emulation

I: Initially a pre-developed maths package was investigated to carry out the arithmetic operations. This was rejected on two counts:

a) The package, written in ASM51, required specific RAM bytes to be reserved for use by the software package. However because the 8751 is limited to 128 bytes of on-chip user RAM this would seriously impede further software developments.

b) Careful analysis of the code listing for the divide subroutine within the package revealed that the routine did not function correctly.

By choosing to create the procedures the author was able to make extensive use of external data memory to store the byte arrays, thus releasing the internal RAM for other purposes. More importantly it could then be ensured that the procedures functioned correctly.

II: Testing of the LEAST_SQUARES procedure was carried out by comparing the results of the procedure with those obtained from a proven least squares fit package used on the Pet computer. Sets of XY coordinates were input into both programs and the resulting gradients and intercepts compared.

III: The software modules for SENCON were found to exceed the 4KByte limit on internal program memory. A considerable amount of care and ingenuity were required to reduce the byte count. Short sections of repetitive code were formed into procedures and any unnecessary re-initialisations were removed. Alternative methods of writing sections of code were developed which required fewer bytes of memory. This optimisation was achieved by investigating the ASM51 code produced by the compiler from the original PL/M51 source code. It was then clear which procedures used a large byte count and where savings could be made. Care was taken to monitor the level of procedure nesting. Each time a procedure is called its return address must be stored on the stack. For nested procedures each return address is stacked "one on top of the other". For interrupt procedures a minimum of 11 bytes are added to the stack to store the return address and the contents of the internal working registers of
the slave controller. Interrupts on the slave controllers are prioritised; the serial port interrupt can interrupt an external interrupt service routine. In this case we have a minimum of 22 bytes to be stored on the stack plus an extra 2 bytes for each nested procedure. The stack is formed from the internal RAM of the 8751 growing from high addresses (07FH) towards lower addresses (00H). Internal RAM is also used to store data variables starting from address 00H. Thus if the stack becomes too large it will overwrite the data variables which will result in chaos. The code can be investigated using the ICE51 to discover the maximum level of nesting and the number of stack bytes required at different points in the program.

IV: The limited availability of program and data memory was found to cause several problems and potential errors. Use of the upgraded version of the 8051, the 8052, was considered. This controller has double the program and data memory space of the 8051. However the in-circuit emulation and prom-programming tools were not available for use on this chip.

V: On using the resistor/relay network for changing the gains of the low-noise amplifier, it was found that switching was not always successful. It was found that with the original circuit a logic high from the non-inverting buffer could drop to 2 or 3V. This was not sufficient to switch the relay on. This problem was overcome by reversing the relay contacts so that a logic zero switched the relay on, and by then using an inverting buffer to provide a maximum output current of 24mA. The inputs to the buffer chips were provided by binary outputs from the OBCCON module. Using this arrangement the voltage drop was measured at 4.7V which was well in excess of the 3.4V required to switch on the relays.

VI: The external time-constants for the PSD were choosen by carrying out a series of experiments to examine the response of the system to a variety of time-constant capacitors. A longer time-constant slows down the systems responses to sample capacitance changes, but also smooths out response to noise by not responding rapidly to noise fluctuations. It is important that the initial calibration data stored for use in the experiment be accurate, for this reason a long time-constant was employed during
auto-calibration. Again a good compromise must be obtained, if 100 calibration data points are to be stored it would not be practical to have a time-constant of 5 minutes. Using a 90 second time-constant reduces the time taken for the calibration task to 2.5 hours and provides the system with stable low-noise signals. A short time-constant of 15 seconds was used during the experiment this allowed the system to respond quickly to capacitance changes, but still provided low noise levels on the 30dB - 70dB gain settings. For the most sensitive setting of 90dB the long time-constant may be most appropriate. However there is the facility for the operator to select either time-constant, and to change this value at any time during the course of the experiment.

5.3.5 The Calibration Software

SENCON's part in the calibration is small. SENCON is placed in the calibration mode by a command from the master controller (MASCON). This results in the SELF_CALIB procedure being called from the CALIBRATE$MODULE (FIG 5.30 and FIG 5.31).

As indicated in (5.1.5) OBCCON oversees the bulk of the calibration operation, and interrupts SENCON each time a sensitivity change is required. The interrupt comes in to SENCON on external interrupt 0. On receiving the interrupt SENCON vectors to the GAIN_FROM_OBCCON procedure which inputs the new gain setting on the low nibble of Port1. The data is handshaked in using the BUFFER$FULL (P3.5) and STROBE (P1.6) lines. The INTERRUPT$RECEIVED flag is set at the end of this procedure and the slave returns to the SELF_CALIB procedure.

At the start of the SELF_CALIB procedure SENCON sets the time-constant to "long", enables external interrupt 0 and sends a message to the master to indicate the start of the calibration. The slave then continually loops until the INTERRUPT$RECEIVED flag is set to 'YES', whereupon the slave enters a small 'DO-LOOP' which takes the new gain value that has been read in by the interrupt procedure and outputs it to the resistor/relay network to change the sensitivity of the low-noise amplifier. Having made the gain change there is a delay of one minute to allow the signal detection system to respond to the new gain setting. SENCON then outputs a
'continue' message to OBCCON to inform it that the gain value has been changed and that OBCCON can now continue reading in calibration data. SENCON then clears the INTERRUPT$RECEIVED flag before leaving the loop.

Throughout the calibration SENCON's task is to respond to interrupts from OBCCON to change the system gain. At the end of the calibration OBCCON interrupts SENCON and sends a 'finished' message instead of gain data. SENCON responds to the finished message by setting the mode of operation to END_CALIBRATION, and leaving the INTERRUPT$RECEIVED flag clear. On returning to the SELF_CALIB procedure SENCON will then be able to exit the main 'DO WHILE MODE - CALIBRATION' loop, disable external interrupt 0, and leave the calibration module.

The CALIBRATE$MODULE utilises the OUT_TO_GAIN_BOX, DELAY and HANDSHAKE procedures from the main program to preserve program memory space and to simplify de-bugging by using previously tested code from the main module.
(PHOTO 5.3) THE SENSITIVITY CONTROL MODULE: SENCON
CIRCUIT DESIGN FOR SENSITIVITY CONTROLLER MODULE SHOWING CONNECTIONS TO THE OTHER CONTROLLERS (FIG 5.20).
MAIN PROGRAM: Do while 1 (i.e. go forever)

Set IO/M line to select external ports. Set GAIN$PORT$A as an output port (output gain setting for the low noise amp. to relay box). Set TIME$PORT$B as an output port (output short or long timeconstant to the PSD). Initialise external interrupt lines high. Set timer 1 to auto-reload mode and turn timer on. Set interrupt 0 as a low level triggered interrupt, and interrupt 1 as edge triggered. Load counter register with auto-reload value. Initialise BRIDGE$IS$BALANCING handshake lines to MOTCON, initialise DATA$GRANTED and DATA$REQUEST handshake lines to OBCCON. Clear initialise bit (set when master takes control of the system gain). Set serial port to 9 bit UART mode & variable data rate. Set SM2 to disable reception of serial data bytes, set REN to enable serial reception, set serial port to high priority interrupt and enable. Clear EXPT$IS$RUNNING bit to prevent the experiment from commencing. Clear CAL$BIT to prevent the calibration procedure from running. Initialise BYTE$SENT to 0. Initialise GAIN_CHANGE and REBALANCE bits to 0, neither are necessary. Clear the OVER_LOAD and FAST_SPEED bits. Set the timeconstant to long (90 seconds), and initialise the gain to 30dB (the lowest setting). Set STATUS line, CHIP_IN_USE line, and BYTE_ENABLE line ready for the A/D converter.

Do while the experiment has not commenced

<table>
<thead>
<tr>
<th>No</th>
<th>Has the master requested a calibration?</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carry out the calibration of the instrument by calling the SELF_CALIB procedure.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Send 'experiment commenced' message to master. Reconfirm gain as 30dB & timeconstant as long. Delay for 1 minute.

COMENCE: Do while the experiment is running

Read in and store the current off-balance voltage (OBV) from the bridge via the 12-bit A/D converter.

<table>
<thead>
<tr>
<th>Yes</th>
<th>Has there been a voltage overflow?</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decrease the gain setting on the low noise amp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Read 10 averaged OBVs at 30sec. intervals</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluate gradient &amp; intercept using LS fit</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Is a gain change necessary?</td>
<td>Yes</td>
</tr>
<tr>
<td>Tell MOTCON to re-balance the capacitance bridge.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Tell MOTCON to re-balance. Change system gain by 20dB (inc. or dec. as required)</td>
<td></td>
</tr>
<tr>
<td>Do while master has control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read OBV from 12-bit A/D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tell master "experiment finished" and disable interrupts
GRAPH INDICATES THE RATE OF CHANGE OF BITS CORRESPONDING TO THE OFF-BALANCE VOLTAGE (FIG 523)

Bits (Output from A/D converter)

Limiting reading 1305 bits = 9.5 digits off-balance

Maximum gradient

Ideal gradient

Minimum gradient

Time at which rebalance is necessary

Readings used to calculate gradient

Evaluated time to bridge rebalance = 27 x 30 seconds = 13.5 min

Time

(seconds x 30)
RANGE OF OFF-BALANCE VALUES MANIPULATED WITHIN THE MICROPROCESSOR

FIG 5-24

Bits off-balance

(+3V) 1FFFH

Maximum positive value

+9.5 digits off-balance

(0V) 1000H

On-balance

-9.5 digits off-balance

(-3V) 0000H

Maximum negative value

Time
SENCON: FLOWCHART FOR THE LEAST_SQUARES_FIT_PROCEDURE (FIG 5.25)

Initialise variables: ONE_SIX_FIVE = 165, FIFTY_FIVE = 55. Clear store variables, clear NEG_GRADIENT bit, clear carry flags. Zero the sum arrays to be used to store intermediate results during the procedure.

| Evaluate: $(XY)_n = \sum x_i y_i$ store result in SUM$1$ array |
| Evaluate: $(Y)_n = \sum y_i$ store result in SUM$2$ array |
| Multiply: $2 \cdot SUM$1 store result in SUM$3$ array |
| Multiply: $9 \cdot SUM$2 store result in SUM$4$ array |
| Subtract: $SUM$3 - $SUM$4 store result in SUM$5$ array |

| Is the result of the subtraction negative? |
| No (i.e. is SUM$5$ NEGATIVE) | Yes |

| NEG_GRADIENT = NO | NEG_GRADIENT = YES |
| Multiply: $19 \cdot SUM$2 store result in SUM$7$ array |
| Multiply: $3 \cdot SUM$1 store result in SUM$8$ array |
| Subtract: $SUM$7 - $SUM$8 store result in SUM$9$ array |
| Multiply: $10 \cdot SUM$5 store result in SUM$6$ array |
| Multiply: $10 \cdot SUM$9 store result in SUM$10$ array |
| Divide: $SUM$6 / ONE_SIX_FIVE store result as the GRADIENT |
| Divide: $SUM$10 / FIFTY_FIVE store result as the INTERCEPT |

| No | Is the GRADIENT zero? |
| Yes |

Set GRADIENT to 1 to prevent a calculation error due to division by zero.
SENCON: FLOWCHART FOR THE MULTIPLY PROCEDURE (FIG 5.26)

<table>
<thead>
<tr>
<th>Zero array OP3 which will contain result of multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESULT(0) = OP1 * OP2(0)</td>
</tr>
<tr>
<td>RESULT(1) = [OP1 * OP2(1)] + high byte of RESULT(0) word</td>
</tr>
<tr>
<td>OP3(0) = low byte of word variable RESULT(0)</td>
</tr>
<tr>
<td>OP3(1) = low byte of word variable RESULT(1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Is the OP2 array 2 bytes big?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>OP3(2) = high byte of RESULT(1)</td>
</tr>
<tr>
<td>RESULT(2) = [OP1 * OP2(2)] + high byte of RESULT(1)</td>
</tr>
<tr>
<td>OP3(2) = low byte of RESULT(2)</td>
</tr>
<tr>
<td>Yes</td>
</tr>
<tr>
<td>Is the OP2 array 3 bytes big?</td>
</tr>
<tr>
<td>No</td>
</tr>
<tr>
<td>OP3(3) = high byte of RESULT(2)</td>
</tr>
<tr>
<td>RESULT(3) = [OP1 * OP2(3)] + high byte of RESULT(2)</td>
</tr>
<tr>
<td>OP3(3) = low byte of RESULT(3)</td>
</tr>
<tr>
<td>OP3(4) = high byte of RESULT(3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4 BYTES</th>
<th>3 BYTES</th>
<th>2 BYTES</th>
<th>1 BYTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP2(3)</td>
<td>OP2(2)</td>
<td>OP2(1)</td>
<td>OP2(0)</td>
</tr>
<tr>
<td>OP1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OP3(4)</td>
<td>OP3(3)</td>
<td>OP3(2)</td>
</tr>
</tbody>
</table>

The procedure uses the method of long multiplication to multiply OP1 by the OP2 array. The result is stored on the OP3 array.
SENCON: FLOWCHART FOR THE SUBTRACT PROCEDURE (FIG 5.27)

Zero the OP3 array which will contain the result of the subtraction.

Clear the carry flags: MY_CARRY_FLAG = 0, CARRY$SAVE = 0
Set RESULT$NEGATIVE to NO.
Initialise the array index: J = U - 1 (U = number of bytes)

Do I = 0 to (U-1) where U is no. of bytes in input arrays, (compare successive bytes to check for a negative result).

Yes

RESULT$NEGATIVE = NO

Set I = (U-1) to drop out of the loop now that it has been established that the result of the subtraction will be negative.

RESULT$NEGATIVE = YES

Set I = U-1 to leave loop as the result of the subtraction will be positive.

No

Do I = 0 to (U-1) where U is no. of bytes in input arrays.

Yes

Do: leave byte order to get a +ve result on subtraction

SUB$A = OP1(I)

SUB$B = OP2(I)

No

Is SUB$A < (SUB$A + CARRY$SAVE) ?

Yes

MY_CARRY_FLAG = 0

SUB$A = OP1(I)

SUB$B = OP2(I)

MY_CARRY_FLAG = 1

Carry out subtraction of bytes:

OP3(I) = SUB$A - (SUB$B + CARRY$SAVE)

No

Is MY_CARRY_FLAG = 1 ?

Yes

CARRY$SAVE = 0

CARRY$SAVE = 1

<table>
<thead>
<tr>
<th>4 BYTES</th>
<th>3 BYTES</th>
<th>2 BYTES</th>
<th>1 BYTE</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP1(3)</td>
<td>OP1(2)</td>
<td>OP1(1)</td>
<td>OP1(0)</td>
</tr>
<tr>
<td>OP2(3)</td>
<td>OP2(2)</td>
<td>OP2(1)</td>
<td>OP2(0)</td>
</tr>
<tr>
<td>OP3(3)</td>
<td>OP3(2)</td>
<td>OP3(1)</td>
<td>OP3(0)</td>
</tr>
</tbody>
</table>
RESULT = 0, CONTINUE = 1, S=R (no. of bytes in SUM array)

Do T = 0 to 4

| COPY_TO(T) = 0 | Empty the COPY_TO array |
| SUM_DOWN(T) = SUM(T) | Copy contents of SUM array |
| NEW_DIVISOR(T) = 0 | Empty NEW_DIVISOR array |

COPY_TO(0) = DIVISOR(0) Make a copy of the divisor

Multiply the copy by 1024 & store in NEW_DIVISOR array.

Call SMALL_DIVIDE - Calculates number of times the NEW_DIVISOR goes into the SUM_DOWN array.

Multiply the copy by 256 & store in NEW_DIVISOR array.

Call SMALL_DIVIDE - Calculate the number of times the NEW_DIVISOR goes into the remainder left in SUM_DOWN array

Multiply the copy by 16 & store in NEW_DIVISOR array.

Call SMALL_DIVIDE

BOBBY: Do while CONTINUE

Subtract SUM_DOWN from COPY_TO, store result in OUTPUT

| No | Is the result of the subtraction negative? | Yes |
|    |                                            |

Bo: Do

RESULT = RESULT + 1

Do T=0 to 4

Copy the result of the subtraction into SUM_DOWN from OUTPUT.

CONTINUE = 0 to leave loop

Divide copy of divisor 2

CONTINUE = 0 to leave loop

RESULT = RESULT + 1

Do while RESULT$NEGATIVE = NO (i.e subtraction is not –ve)

Subtract: SUM_DOWN array - NEW_DIVISOR array
Store the result in the OUTPUT array.

| No | RESULT$NEGATIVE = NO? | Yes |
|    |                        |

Do

RESULT = RESULT + 1

Do T=0 to 4

Copy OUTPUT array into SUM_DOWN array
SENCON: FLOWCHART FOR THE CALIBRATION SOFTWARE MODULE (FIG5.30)

START

SET INTERRUPT RECEIVED TO "NO"
SET MODE TO CALIBRATION
SELECT LEDG END TO CONSTANT
SET BRIDGE B BALANCE LINE FOR INPUT
ENABLE EXTERNAL INTERRUPT (REQUEST OR CON TO REQUEST A GAIN CHANGE)

SEND MESSAGE TO MASTER STATION:
CALIBRATION BEGINS

CHECK THAT MESSAGE HAS BEEN TRANSMITTED

IS MODE SET TO CALIBRATION

NO

YES

NO

HAS AN INTERRUPT BEEN RECEIVED FROM ORCON?

YES

OUTPUT NEW GAIN SETTING TO LOW-
NOISE AMPLIFIER VIA THE RELAY BOX

DELAY FOR ONE MINUTE AFTER CHANGING
THE GAIN SETTING ON THE AMPLIFIER

COMPLETE HANDSHAKE WITH ORCON TO
INDICATE GAIN VALUE HAS BEEN CHANGED AS REQUESTED

CLEAR INTERRUPT RECEIVED FLAG WHICH
WAS SET BY ORCON INTERRUPTING SENCON

COMPLETE HANDSHAKE WITH ORCON TO
INDICATE A "FINISHED" MESSAGE WAS RECEIVED

DISABLE EXTERNAL INTERRUPT

END
Set INTERRUPTS$RECEIVED flag to NO.
Set MODE to CALIBRATION.
Select long TIME CONSTANT for the calibration.
Set BRIDGE$IS$BALANCING line to act as an input line.
Enable external INTERRUPT 0 (used by OBCCON during calibration to indicate a change in gain setting is needed.

Send 'CALIBRATION BEGINS' message to the master.

Check message has been transmitted.

Do while mode of operation is set to CALIBRATION

<table>
<thead>
<tr>
<th>No</th>
<th>Has a new gain value been received from OBCCON</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Do</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Output the new gain setting to the low-noise amplifier via the relay box.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Delay for one minute after setting the new gain.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Complete handshake with OBCCON to indicate gain value has been changed to the requested setting.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Clear the INTERRUPTS$RECEIVED flag which was set when OBCCON interrupted SENCON with the new gain.</td>
<td></td>
</tr>
</tbody>
</table>

The byte received from OBCCON was a FINISHED MESSAGE not a gain value as the MODE is no longer set to CALIBRATION. Now complete handshake with OBCCON to indicate that the finished message has been received and acted upon.

Disable external INTERRUPT 0
This interrupt is only enabled during a calibration.
CHAPTER SIX - System Design: The Master Control Module and the Interface to the Operator via the IBM PC

6.1 The Master Control Module (MASCON)

6.1.1 Circuit Diagrams and Photograph of MASCON

The first figure (FIG 6.1) is of the electrical circuit for the MASCON module. The second figure (FIG 6.2) shows the module's connections to the other system controllers. A photograph of the printed circuit board (PCB) is shown in PHOTO 6.1.

6.1.2 Overview of the Functions of MASCON

The master control module, (MASCON), interfaces between the slave controllers and the operator who is entering commands on an IBM 286 PC. It is MASCON's job to communicate with, and send commands to, each of the slave controllers (MOTCON, OBCCON and SENCON). MASCON communicates with the slaves on a serial link. The master controller receives each command from the IBM PC; this is analogous to a middle manager who receives orders from the director (the operator of the IBM PC). MASCON commands the appropriate slave (employee) to carry out the directed task, and then monitors the execution of the task. Following execution MASCON reports back to the operator of the IBM PC (director) with the results of the completed task. Flowchart representations of the main software programming module are given in FIG 6.3 and FIG 6.4.

The main hardware components of the MASCON module (FIG 6.1 and FIG 6.2) are the 8751 master controller, the 6402 UART (Universal Asynchronous Receiver Transmitter), the RS232 converter, and the over-voltage protection circuit (common to all hardware modules). Communication to the slaves is through the master's own serial link. Each slave can be reset individually by the master, and the master itself resets on power-up by means of an 'RC' reset circuit. To enable communication to the IBM PC, a UART and an RS232 converter have been introduced. Two separate 8-bit ports, Port 1 and Port 2, on the master are used to transfer parallel data bytes to and from the UART. The UART converts between parallel and serial data, and
transfers data to the IBM PC on its own serial link. The RS232 communication channel on the IBM PC works on a ±12 Volt logic levels, whilst the output from the UART provides ±5 Volt logic. A special RS232 converter chip (IC6) requiring a single 5 Volt supply converts between voltage levels. IC6 achieves this conversion by using internal capacitors to store charge. The UART requires that data is clocked into its registers at 16 times the required data transfer rate. A miniature crystal oscillator (EXO-3) provides the UART with a clock frequency of 76.8 KHz which gives a data transfer rate of 4800 baud. A serial data byte consists of 8 data bits, 1 stop bit and no parity. Transfer is synchronised with the IBM PC at a rate of 4800 baud.

During initialisation MASCON sets interrupt 0 to high priority. An interrupt received on this pin signals that the UART has received a command byte from the IBM PC. The serial port on the master is set to the variable data rate/9-bit UART mode. Timer 1 is used as a baud rate generator for the serial port; the baud rate is set to 3125 bits/second. The status flags on the UART are reset by MASCON taking the master reset line (MR) high. The data received reset pin (DRR) is set high to act as an input line to the master controller. When the UART receives a command byte DRR goes low and must be reset by the master after reading each byte. After parameter initialisation the master calls the 'RESET' procedure to reset each slave controller in turn. Resetting the slaves allows each of them to carry out their own parameter initialisation. They will then remain within a 'DO WHILE' loop waiting for the master to signal the start of an experiment or calibration.

When the IBM PC sends a command to the master via the UART, the UART signals to the master using interrupt 0. Upon interrupt 0 the procedure named IBM is called, (FIG 6.5). A five byte array called TABLE has been set to handle the storing of several command bytes being sent in succession. A pointer is used to index into the table. This pointer is incremented each time a new command byte is received and then decremented when the command is executed. The command codes range from 1 - 12. Command code '11' (0BH) causes the experiment to start, and command code '12' (0CH) requests a calibration. Both of these commands should only be received at the beginning of program execution, and require only that the bit 'expt$is$running' or the
bit 'cal_bit set by the interrupt procedure. On returning to the main program the effect of setting expt is running will cause the experiment to commence and the effect of setting cal_bit will cause the calibration procedure, SELF_CALIB to be called (see 6.1.5).

During the running of the experiment the IBM PC may send other commands, these are received and stored in the array called TABLE by the procedure called IBM. This procedure indicates to the main program that a command byte has been received by setting the 'command_byte_received' bit. Once the command has been executed by the main program the command_byte_received bit is cleared.

When the command to start the experiment is received the procedure START_EXPERIMENT is called. The master addresses each slave in turn and then commands them to set the experiment running. Because of the interconnections between the slaves OBCCON must be initialized first, followed by SENCON, and then MOTCON. Should a problem occur during initialization a set-up error is reported back to the IBM. A similar procedure named STOP_EXPERIMENT is called when the master commands the experiment to stop. In this instance the 'stop' command will not take effect immediately; the slaves will complete any task that is in progress before stopping the experiment. An immediate halt can be obtained by calling the 'RESET' procedure.

As explained, the IBM interrupt procedure sets the 'command_byte_received' bit when a command is received from the IBM PC. This bit is polled continuously during the main program and if set, the procedure 'DO_CASE_COMMAND' is called. The table containing the stored commands is received as an input parameter to this procedure. The DO_CASE_COMMAND procedure (FIG 6.6) empties the table of command bytes, resets the pointer to zero, and then proceeds to execute the commands consecutively. Each command is separated into a high and low nibble. The high nibble contains the command code, and the low nibble may contain additional information needed to execute the command (e.g. the value of a new sensitivity setting). Each command byte causes a separate procedure to be called which executes the command. These will be reviewed in section (6.1.3).

To set up communications with a particular slave the procedure CHAT_TO_SLAVE is called. This procedure sends out an address byte on the serial link to the particular slave that the master wishes to
talk to, and then waits for the reply "I am listening" from the slave. On receiving this reply the procedure outputs a command on the serial link. This procedure has two input parameters enabling the master to give any command to any slave. The master waits up to five seconds to receive a reply from the slave. If no reply is received, or an incorrect reply is received, the master reports an error to the IBM PC.

In summary the master resets each slave and then awaits the start of an experiment or calibration, which is initialised by a command from the IBM PC. Any command byte that is received by the IBM PC is sent via the UART. The UART will signal that a command byte has been received by causing an interrupt on INTO. The master stores the command and then communicates it to the appropriate slave controller. The master monitors the progress of these commands and reports any errors in their execution back to the operator of the IBM PC. On receiving the STOP command (command code 01H) the master halts each slave and then notifies the operator of successful or unsuccessful completion. After this the master delays for 20mS to ensure that data transfer to the IBM PC is complete. It then returns to the start of the main program and re-runs the initialisation process.

6.1.3 Details of Important Module Tasks

The important tasks outlined here correspond to commands received from the operator via the IBM PC. Each task is accessed through the DO_CASE_COMMAND procedure (FIG 6.6) which selects the instruction to be executed or the procedure to be called.

**Command 0**

This command value is not used as ASCII OH represents a null character which produces an error message on the IBM.

**Command 1 : 'expt$is$running = 0'**

This instruction causes the experiment to be stopped. On clearing the bit expt$is$running the master is able to exit the COMMENCE loop and execute the STOP_EXPERIMENT procedure which halts each slave processor in turn.
Command 2: 'CALL OBC TO MASTER'

The master controller addresses OBCCON and commands that it sends the latest off-balance capacitance (OBC) value. This will consist of four bytes of information: two are OBC digits, the third is the sign of the OBC, and the fourth is the checksum of the first three bytes. In most cases there is only one valid OBC digit and the sign. However, if the capacitance of the sample cell is changing very rapidly with temperature there may be two valid OBC digits. In this particular case the gain setting is 30dB and the selected capacitance arm is 10ff (i.e. range ±10ff - 30ff off-balance).

If an error occurs during data transfer, or the checksum value does not agree with the sum of the data received, the master transmits an error message to the IBM PC. If no error has occurred, the OBC information is converted to the ASCII equivalent and transmitted to the IBM PC.

Command 3: 'CALL CLEAR OVER RIDE'

It is possible for the operator to control the sensitivity of the system by setting an over-ride flag. This procedure returns control of the system gain to SENCON by clearing the over-ride flag. The master sends a command byte to SENCON along the serial link, to clear the over-ride flag and then waits for SENCON to echo the command byte back. If the command byte is returned corrupted, the master sends a 'CONFIRM_ERROR_DETECTED' message to the slave and the IBM PC. If the echoed command is correct a 'CONFIRM_NO_ERROR_DETECTED' message is sent to SENCON and the IBM PC.

Command 4: 'CALL CHANGE DELAY'

The high nibble of this command contains the information to select the delay parameters from an array called DELAY_VALUE. These parameters are then passed to the MOTCON slave. The slave uses the values to change the time delay between accessing and updating the latest OBC value from OBCCON. This command facility is available to give the operator flexibility to control the selection of capacitance data.
The command byte and delay parameters are echoed back from the slave to the master and a similar error checking procedure to command 3 is carried out.

Command 5: 'CALL CHANGE TIME CONSTANT'

On receiving this command the master addresses SENCON, and then demands that the slave change the time-constant. The time-constant may be long (90 seconds), or short (15 seconds). A long time-constant makes the off-balance voltage (OBV) reading less sensitive to noise fluctuations and should be used when the system gain is 70 dB or above. SENCON will select the correct time-constant for the current gain setting, but the operator may choose to over-ride both system gain and time-constant control.

As with all of the command procedures, the master sends the new time-constant setting to the slave, and then waits for SENCON to echo the value back. If an error occurs with this or any other command, an error message is sent to the IBM PC detailing which slave and which command are causing the problem. If the command is executed successfully a 'CONFIRM_NO_ERROR_DETECTED' message is sent.

Command 6: 'CALL CHANGE GAIN'

The high nibble of this command contains the information to select a new gain setting for the system. In normal operation SENCON selects and controls the system sensitivity. However, the operator can over-ride the system control by sending this command to the master via the IBM PC.

The master transmits the command and then the gain value to SENCON, and then waits for SENCON to echo them both back correctly. SENCON sets an internal over-ride flag which prevents it from making further changes to the gain. If an error occurs in transmitting the data, and SENCON does not execute the command successfully, then an error message is returned to the IBM PC.

This command gives the operator the ability to take a complete set of data readings on one particular sensitivity setting, provided that the off-balance voltage reading remains within range. If the
operator chooses to select a high gain when the sample capacitance is changing rapidly, the bridge will need to be frequently rebalanced. Similarly, if the operator chooses to select a low gain when the sample capacitance is changing slowly, sensitivity will be lost and the data results will be less accurate. For this reason the operator must exercise discretion in choosing to over-ride the gain setting that has been selected by SENCON.

To return control of the system sensitivity to SENCON the operator must send command 3 to clear the over-ride flag.

Command 7 : 'CALL_SEND_GAIN'

This command requests that SENCON tells the master what the current gain setting is on the system. Each command procedure follows a similar pattern; the SEND_GAIN procedure is no exception. First the CHAT_TO_SLAVE procedure is called which sends the command to the chosen slave, SENCON, and the master then waits for SENCON to echo the command back correctly. The master then knows that the slave has received the command and is ready to respond. The current gain setting is requested and received. The gain value is echoed back to the slave as an error check. If the slave returns a 'CONFIRM_NO_ERROR_DETECTED' message the received gain setting is sent to the IBM PC. If a problem occurs at any point during the execution of the command, the master sends an error message to the IBM PC. In each case this error message consists of coded information indicating which slave and which command are causing an error to occur. The operator may then decide what action to take.

Command 8

This command value is not currently in use.

Command 9 : 'CALL_READ_CAP_FROM_SLAVE_OUT_UART'

This command is used to access the latest on-balance capacitance reading from the MOTCON slave. The procedure READ_CAP_FROM_SLAVE_OUT_UART is called which requests, and reads in the capacitance data and then transmits the data to the IBM PC. A flowchart of the procedure is given in FIG 6.7.
The master addresses MOTCON and requests the latest capacitance data. If MOTCON does not reply within 5 seconds an error is reported to the IBM PC. Because the serial port interrupt has been set to the highest priority interrupt in each of the slaves they should respond immediately to an interrupt from the master. Thus 5 seconds will give the slaves ample time to respond.

If MOTCON is ready to send the capacitance data, the master disables the serial port interrupt, and then requests each data byte in turn. After requesting a data byte the master polls the received interrupt flag (RI), and stores the received data byte in the capacitance data array. Nine capacitance data bytes are transmitted corresponding to the decades 1aF - 100pF. The tenth byte received contains an information code which can be interpreted by the IBM PC (see 5.2.3). The eleventh byte is the checksum. If a problem occurs at any point during data transfer, or if the received checksum does not equal the master's own summation of the data, an error is reported to the IBM PC.

If the data transfer has been executed successfully each capacitance data byte is converted to the equivalent ASCII code and then transmitted via the UART to the IBM PC. By performing the conversion to ASCII codes the data can be more easily manipulated and stored by the IBM PC. If the data transfer has been unsuccessful the ASCII code for 'X' is loaded into each byte of the capacitance array and this information is sent to notify the operator of an error.

Command 10 : (0AH)

This command value is not used as ASCII 0AH is interpreted as a 'line feed' character, and the IBM PC treats a 'line feed' as an 'end of string character' when transmitting serial data.

Command 11 : (0BH)

As detailed in (6.1.2) this command is used to start the experiment running and is therefore only used at the beginning of the experiment. Should the operator send this command during the running of the experiment the DO_CASE_COMMAND procedure is called, but no action is taken.
Command 12 : (OCH)

This command is used to start the experimental calibration, and like command 11 it is only used at the beginning of the experiment. Again, should the operator send this command during the running of the experiment, no action will be taken by the master.

For each of the above commands requested by the IBM PC and executed by the master, if an error occurs the master reports it to the IBM PC, giving information as to which slave and which command has caused the problem. Furthermore, the master will also send an error message to the slave involved with the execution of the command.

The procedure SEND_UART is a simple five line procedure which outputs a single byte of information to the UART and ensures that the byte is latched into the transmitter register. Whenever the master transmits a data byte to the IBM PC this procedure is called. The master waits until the UART's transmit buffer register is empty and then outputs the byte on Port 2. The master then strobes the transmit buffer register load pin, to request that the data byte is transferred to the transmitter register. The UART converts the data byte from parallel to serial data before transmitting it to the IBM PC.

6.1.4 Problems Encountered : Diagnostic Tests and Module Emulation

I. Master Emulation

The master controller module (MASCON) was the last module to be designed and built. This module interfaces between the IBM PC and the slave controller modules. It is unlike the slave modules, which spend the majority of their time inputting and outputting logic signals, which monitor and control the real world instrumentation. MASCON spends all of its time in communication with either the slave modules or the IBM PC. Because of this fact, this module was the most difficult to test in isolation. Up to this point each slave module had been tested using an ICE51 (In-Circuit Emulator for the 8751 microcontroller). Using the emulator it was possible to simulate the receiving of a command byte from the master on the
serial port of the slave, and to then monitor the subsequent action taken by the slave. As the system design and testing neared completion it was necessary to adopt a different approach in carrying out the emulation of MASCON. This approach demanded that the slave software and hardware was functioning correctly, and that the slave modules could be used with confidence. System testing of the four modules is shown in PHOTO 6.2.

The approach taken was to emulate MASCON and to link and test it with each slave module in turn. Each slave module contained an 8751 eprom chip into which the slave software had been blown. Each slave module could be reset via MASCON. OBCCON was the first slave module to be linked to MASCON. Logic interface circuits were used to simulate the inputs coming from, and monitor the outputs going to, the other two slave modules. On linking MASCON and OBCCON, testing highlighted the modifications that needed to be made to the main software module and to the calibration software module.

It was a fairly simple task to erase and blow an eprom chip in order to make modifications to the slave software. The program in the eprom could be erased by subjecting the chip to ultraviolet light for about 20 minutes. It was then possible to download the modified program from the Series IV development system to the Prom-Programmer on to which the eprom chip had been secured. Blowing the eprom took the order of minutes to execute and verify. However, testing was hampered by the fact that a number of faulty eproms were found in the batch being used. Initially it was thought that there was an error in the software, but by blowing the same program into each of the eproms it was discovered that a number of the chips functioned correctly and that a number of the chips were faulty.

Having successfully tested MASCON with each slave module in turn, the next step was to test MASCON and two slave modules (OBCCON and MOTCON), and then to test MASCON and three slave modules (OBCCON, MOTCON, and SENCON). A complete calibration must be executed before the main module software can be tested. Because OBCCON was no longer being emulated, it was not possible to take the easy option of loading temporary calibration test data into the RAM of the OBCCON module. Details of MASCON calibration software are given in 6.1.5. It was also necessary at this stage to link the master module.
(MASCON) to the IBM PC so that the calibration data read in by the slave could be examined by the operator and stored on the hard disk of the IBM PC. Again the details of data transfer and storage are given in 6.2.2 and 6.2.3.

II Error Handling

A major problem and design consideration of any automated system is its response to potential errors and safety critical situations. For example:

'What happens if part of the system does not respond as expected or crashes?'

'During data transfer, is the information received checked for possible transmission errors?'

A 'system lockup', that is the trapping of the software in some infinite loop, must be avoided. The system should have some means of error recovery and should be able, if possible, to report back reasons for system failure.

Some computer systems cannot tolerate software faults (or bugs). They must be of high integrity; responding in an ordered fashion in all circumstances. Such software is designated 'safety critical'. It is not sufficient to test the software dynamically and to build in error checking procedures. The code must be verified and validated using a proven static analysis tool. Examples of such software needs are the protection system for a nuclear power station and the auto-pilot control on an aircraft.

The computer system for the automation of the capacitance dilatometer is not safety critical, but the system does need to be highly reliable. As such there does need to be a reasonable degree of sophistication built into the error handling procedures within the software. However, should the system be expanded to include the automated control of the superconducting magnet within the cryostat, then one might well deem the software to be safety critical. The level of liquid coolant in the cryostat and the temperature must then be monitored to avoid unwanted quenching of the magnet. Should
the magnet cease to be superconducting a safety critical software system must know how to respond.

In building error checking into the software the system response time needs to be considered. For a real time system, where data is read and updated every few milliseconds, the time allocated to error handling procedures is limited. However for this system, where the response time is the order of seconds, there is ample time to execute error handling procedures.

Where several data bytes have been transferred from slave to master and/or master to IBM PC, the addition of a checksum byte (or word) has been used as a means of error checking the data. A checksum byte, as the name implies, is the sum of the data bytes being transferred. For example, when OBCCON is ready to transfer the calibration data it sums the data bytes and stores the result in the checksum word. The checksum is the last two bytes of data to be transmitted in any transaction. When the calibraton data is finally received by the IBM PC it performs a check by comparing its own checksum with that transmitted. If the two checksums differ the IBM PC will request that the data be transferred again. A maximum of four attempts to read the data are allowed. The checksum may take different forms: the data may be logically ANDed or ORed together. It was decided to take the simple approach of adding each byte together into a running total which was then sent as the checksum. Obviously errors may have occurred in the data transfer which leave the checksum containing the same value as an error free transfer; hence one error may be masked by another error. No error check can be fool proof, since the check itself is open to errors.

Another common method of error checking is the use of a parity bit. This is usually a 9th data bit transferred in addition to the 8 data bits of the byte. The 9th bit is used to carry out either even or odd parity checking. For even parity, the 9th bit is set such that the number of 1's in the 9 bits transferred is even. For odd parity error checking, the 9th bit is set such that the number of 1's transferred in each 9 bits is odd. This system relies on the fact that the machine receiving the data is set up to receive and check bytes with the same set parity. On the master/slave system, the 9th data bit transferred with each byte on the serial data link, was
used to indicate whether the byte was a command byte (9th bit = 1) or a data byte (9th bit = 0). Thus in this multiprocessor mode, parity checking of data transferred between the master and slave modules was not possible, but parity checking between the UART on the master module and the IBM PC was possible.

The procedure called IBM handles the receiving of command bytes from the IBM PC. A table has been set up to handle the possibility of the operator sending several command bytes from the IBM PC in quick succession. The table is able to store up to 5 commands and MASCON will execute each of these commands in sequence. Use of the table prevents potential errors due to loss of command bytes. The command byte received will be echoed back to the IBM PC for confirmation: 'Is this the command you want me to execute?' The IBM PC will reply with one of two messages: CONFIRM_NO_ERROR_DETECTED, go ahead and store this command for execution, or CONFIRM_ERROR_DETECTED, do not store the command byte you have received as an error has occurred in the data transfer. A similar approach is adopted for the transfer of single data byte between the master and slave modules. In the case of a single data byte, instead of using a checksum, the byte is echoed back to the slave, or to the master, for confirmation.

As mentioned in (6.1.2) the procedure CHAT_TO_SLAVE plays an important role in accessing the slave that the master wishes to talk to. Whenever a command is to be sent to a slave this procedure is called. As such it was necessary to build careful error checks into this procedure. The master sends out an address byte to the slave and waits for a reply. The master waits until the slave is actually listening and ready to receive a command byte before sending the command. This ensures that the communication channel between master and slave has been correctly set up. Should the slave not reply within 5 seconds, the master assumes an error has occurred and will drop out of the CHAT_TO_SLAVE procedure and report the error to the IBM PC, indicating which slave and which particular command caused the problem. In this way the master will not become trapped in a loop waiting forever for the slave to reply. Similarly, should the slave reply with any message other than 'I am listening' the master will not continue, but will leave the CHAT_TO_SLAVE procedure and again report a problem to the IBM PC. By acting in this manner, account is taken of potential errors in command byte transfer and/or
data byte transfer. The operator is also made aware of these problems via the IBM PC and is able to request the command again, or act on the error information received.

Error checks are also made within the slave software. For example the master may address the slave, the slave may reply 'I am listening', but then not receive a command from the master. In this case the slave will drop out of the serial port interrupt procedure, after waiting for 5 seconds, and then return to whatever it was doing prior to the interrupt. Similarly if the slave receives a command that it does not recognise, it will not attempt to execute any command, but will return to its main program. In this way the system is not corrupted and the master is able to make another attempt at speaking to the slave.

The BASICA software being used on the IBM PC normally interprets data as ASCII codes. Thus when it receives the hex value '0DH' it interprets this as a 'carriage return', which would signal the end of the string of data. Other ASCII codes which are interpreted as end of string characters are: OAH = 'line feed', 20H = 'space', and 2CH = 'comma'. The hex value 00H = 'null' will cause an error message on the IBM. When transferring the calibration data to the IBM from MASCON this caused problems. Every time MASCON sent a data byte, which happened to be one of the above codes, the BASICA software interpreted it as an 'end of string'. To avoid this problem when sending calibration data from MASCON, the serial port on the IBM PC was read directly, rather than by the BASICA I/O port command. Details of this are given in the next two sections. Apart from the calibration data, all data/command bytes transferred between MASCON and the IBM PC were converted into ASCII codes. Each ASCII byte transmitted to the IBM PC was followed by '0DH' - a carriage return byte to signal an 'end of string'. The conversion was achieved by having an array of ASCII codes and by using the hex value to select the required element from the array. Only 16 array bytes were needed for the values 00H to 0FH. As these ASCII array values were read, but not written by the master, they were stored in ROM memory rather than using up the limited RAM memory space.

The numbers 00H to 09H are used to indicate the value of the capacitance digits, and the codes OAH to OFH are used to indicate
Information codes sent with the capacitance data (5.2.3). Originally it had been intended that the on-balance capacitance digits from MOTCON should be sent in packed b.c.d. format (each digit takes 4 bits, therefore 2 digits per byte). However, because the data is interpreted as ASCII codes it was easier to send one digit per byte rather than use the BASICA software to unpack the bytes, which would make the task unnecessarily complex. The capacitance data consisted of: nine capacitance data bytes, one information byte, and one checksum byte. Each code received from the slave was converted to ASCII by the master and then transmitted to the IBM PC via the UART. A carriage return byte followed each byte transmitted. If during the execution of the READ_CAP_FROM_SLAVE_OUT_UART procedure an error had occurred the master would replace the capacitance data bytes with the ASCII code for 'X'. Thus the operator could easily determine which capacitance data values were erroneous.

In carrying out error checking a balance must be achieved. Obviously error handling is necessary in all automated systems, but exhaustive and time consuming checking could result in hindering, rather than improving, the accuracy and performance of the system.

III Stack Problems

MASCON is the only module that does not have any external RAM, and as such its own internal RAM memory is precious, being limited to 128 bytes. On testing the software spurious errors occurred. It was discovered that MASCON was running out of stack space, which was causing the return addresses on the stack to be lost, this resulted in chaos.

The top of the stack is at address 80H and it was observed that the RAM space upto 6FH was being used for storage of both local and global variables, and the 8751 registers. Thus only 17 bytes of RAM remained for use as the stack. Each time a return address is stored on the stack 2 bytes are used. Each time an interrupt procedure is called there is an 11 byte penalty to the stack. An interrupt procedure stores the return address, the interrupt handler address, and a number of the processors registers are also pushed on the stack. It can easily be seen that if a low priority interrupt were to be executing when a high priority interrupt occurred then 22
bytes of stack would be needed. As MASCON has only 17 bytes there was a problem.

Rather than adding extra RAM memory to the module, which would involve changing the module design, it was decided to consolidate the existing use of RAM memory in an attempt to free some more stack space. This was achieved by making the capacitance array, and limiting calibration data array, local to the procedures in which they are used instead of global to the whole module. The command byte table was also reduced from originally 10 to 5 bytes. By using this approach the stack size was increased to 32 bytes (i.e. 60H to 80H). This was ample stack space: low levels of interrupt (2 x 11 bytes) and 5 levels of nested procedures (5 x 2 bytes) could occur simultaneously without a problem.

6.1.5 The Calibration Software

MASCON's task is to monitor the calibration of the capacitance dilatometer, and to transfer the calibration data to the IBM PC. Flowcharts which show an overview of the software for the calibration module are shown in FIG 6.8 and FIG 6.9.

The procedure SELF_CALIB is called when the IBM PC sends a command byte to MASCON to request that a calibration takes place. During the calibration external interrupt 0 is disabled. When executing the main software module this interrupt is used to indicate that a command byte has been received. However, during the calibration process it is necessary to poll this interrupt pin, and prevent the IBM interrupt procedure from being called.

MASCON uses the START_CALIBRATION procedure to tell each slave in turn to start the calibration process. If the slave successfully initialises its own calibration software it replies to MASCON with the message 'CALIBRATION_BEGINS'. MASCON then goes on to talk to the next slave. If at any point communication breaks down then then MASCON sends a 'PROBLEM' message to the IBM PC. OBCCON is the last slave to be put into calibration mode, this is because OBCCON controls the calibration and requires MOTCON and SENCON to be initialised first and ready to respond to its requests.
The progress of the calibration process is monitored by MASCON. This information is relayed to the IBM PC so that the operator can check that the calibration data is being stored correctly by OBCCON. A message is sent to MASCON from OBCCON at the completion of each stage of the calibration. There is a message corresponding to each capacitance arm, for both the collection of positive and negative calibration data. When OBCCON has completed the collection of calibration data it sends a final 'FINISHED_CALIBRATION' message.

The next step is for OBCCON to send a copy of the calibration data to MASCON. MASCON does not have its own external RAM memory and must pass each calibration data byte on to the IBM PC as it is received. The procedure 'READ_SLAVE_TRANS_UART performs this task. Each calibration byte is asked for in turn by sending the request 'SEND_ME_DATA'. The calibration byte is then output immediately to the UART. The last two bytes received are the two checksum error bytes which are used by the IBM PC to check for errors in the transmission.

Having sent the data to the IBM PC, MASCON waits within a procedure called MONITOR for confirmation that the IBM PC's own error checksum is the same as that transmitted by OBCCON, at the end of the calibration data block. If all is well the IBM PC accepts the data as valid and sends a 'CONFIRM_NO_ERROR_DETECTED' message to MASCON. In the calibration mode of operation interrupt 0 is disabled. Therefore MASCON polls this interrupt flag during the CONFIRM procedure. When the IBM PC confirms success, MASCON tells OBCCON to clear a 'wait' flag, this enables OBCCON to continue with the next stage in the calibration process. If an error has occurred MASCON drops out of the CONFIRM procedure and does not clear OBCCON's wait flag, but returns to make another attempt at reading the calibration data. A maximum of four attempts have been allowed after which MASCON will try to continue with the next part of the calibration monitoring process.

Once the calibration data has been successfully transferred MASCON allows OBCCON to carry out the data conversion, which is the next step in the calibration (see 5.1.5). Meanwhile MASCON waits to receive a 'CONVERT_CAL_COMPLETE' message from OBCCON. Again, if this message is not received MASCON notifies the IBM PC of the problem by
sending a 'HASSLE' message. The same procedure, READ_SLAVE_TRANS_UART, is used to transmit the converted calibration data. The CONFIRM procedure is called once more to check that the IBM PC has received the data, and has confirmed the checksum as correct.

At this point MASCON will tell OBCCON to continue with the final part of the calibration, which is the transfer of the limiting calibration data to MASCON. In turn, MASCON transfers this data to both the IBM PC and SENCON. The limiting calibration data is used by SENCON to determine when to request a bridge rebalance. Having received the limits from OBCCON, MASCON transmits them to SENCON via the serial link. MASCON can make a maximum of four attempts at sending the limiting data to SENCON. After an unsuccessful attempt the message 'SENCON_TRANSFER_UNSUCCESSFUL' is sent to the IBM PC, and after a successful attempt the message 'SENCON_TRANSFER_SUCCESSFUL' is sent to the IBM PC. The procedure TRANS_LIMITS_SENCON is used to transmit the data to the slave.

When the limiting calibration data has been transmitted to SENCON the calibration process comes to an end. MASCON indicates this to the IBM PC by sending the message 'CALIBRATION_ENDS'. Before leaving the calibration module, MASCON re-enables interrupt 0. MASCON then exits the calibration module and returns to the main software module to await further commands from the IBM PC.

6.2 System Communication to the Operator

6.2.1 Hardware Link to the IBM PC from the Microsystem

A serial/parallel adapter card was plugged into a system board expansion slot of the IBM PC. The card provides a parallel port, and a serial port, for communication to the external world. The serial port is classified as an RS-232C port. Communication to MASCON is via this serial port. The hardware data transfer lines between the IBM and MASCON were as shown overleaf.
receive data pin 2
transmit data pin 3
signal ground pin 5

The 'request to send' (pin 7) and 'clear to send' (pin 8) signals on the RS-232C were shorted together so that whenever the IBM PC pulsed a 'request to send' on the line this signal was fed back on to the 'clear to send' line, and read by the IBM PC. This reply was interpreted as the external device declaring that it was 'clear to send' data.

RS-232C does not use 0V and 5V logic levels. It works using logic levels of -12V (logic low) and 12V (logic high). To overcome this problem a line driver/receiver chip was used to interface between the IBM PC and the MASCON module. The chip works on a single 5V supply and will convert ±12V signals to TTL logic and vice versa. In practice the chip provides output signals of ±8V from 0V and 5V inputs, but this was sufficient to drive the serial port on the IBM PC.

The master controller's own serial port was being used for multiprocessor communications between the slaves. Therefore it was necessary to introduce a Universal Asynchronous Receiver/Transmitter (UART) to convert between parallel and serial data on the MASCON module. Two ports on the master controller were utilised to transmit and receive parallel data to and from the UART.

The data format for the serial port on the IBM PC is as follows: the UART on the serial/parallel adapter card automatically inserts a start bit (logic 0), then a maximum of 8 data bits, followed by the parity bit (if programmed to do so), and then the stop bit or bits (1, 1.5, 2), depending on the command in the line control register. The author's intention was to select both the UART on the MASCON module and the UART on the adapter card, to be set up to send and receive data in the following format:

1 start bit; 8 data bits; 1 even parity bit; 1 stop bit

Although as far as the hardware is concerned this set up was perfectly acceptable, a problem occurred in setting up the
communications software. The software has been written in BASICA (Advanced BASIC). This language uses the 'OPEN "COM1:"' command to allocate a buffer for input and output. The communications adapter is opened as a file, and all input and output statements valid for disk files are also valid for communications. However the syntax of this command is such that only 7 data bits and 1 parity bit can be transmitted or received, or alternatively 8 data bits and no parity bit. Because the numerical calibration data is being sent as hex codes in the range 00H - FFH, the use of all 8 data bits in the byte is required. Consequently it was necessary to set the UART and the adapter card to expect no parity bit. The baud rate on the IBM PC was set to 4800, to match the rate of data transmission from the UART.

The operator was able to interact with the system by selecting options from the system menu displayed on the VDU screen. Each option can be selected via a function key (F1 - F10), or a control key. When the program is executed the menu is displayed automatically. The operator is able to display the menu choices at any time by entering \tM (control M). Section (6.1.2) detailed the effects of selecting each particular function key. PHOTO 6.3 displays the system menu as seen by the operator. When a function key is pressed a system interrupt occurs causing the IBM PC to vector to a software subroutine which will execute the task associated with the function key. This will normally involve sending a command to MASCON, transmitted via the serial port.

When the IBM PC receives data on its serial port an interrupt will occur, and the program enters a routine to read in the data string (which will be terminated by the carriage return character). During the calibration procedure this serial interrupt is disabled, and the IBM PC polls the serial receive data buffer waiting for calibration data bytes. Using this technique slows down the reception of serial data, but allows the IBM PC to read and store single bytes of data which may be any hex code in the range 00H - FFH. When the communications port interrupt is disabled an 'end of string' code such as '0DH' will not be interpreted by the IBM PC.
6.2.2 Software for the Execution of Operator Commands

The system software running on the IBM PC has been written in BASICA (Advanced BASIC). The language enables easy access to the machine's communications ports and function keys. Code can be written quickly and effectively. The language allows the user to work simultaneously at a low level: reading and writing to the machine's hardware registers directly, and at a high level: function key options are simple to redefine and data can be written to the hard disk with a single line of code. BASICA was already installed on the machine and proved to be an uncomplicated language to learn and use.

Within the computer industry the move is towards "user friendly" programming languages. That is, languages which enable complex tasks to be accomplished with a few lines of code. These languages are referred to as '4GLs' (4th Generation Languages), and are primarily restricted to data base processes. For these applications large quantities of data are stored on records within files. The language enables this information to be accessed swiftly and selectively in a variety of ways. The user is able to manipulate the data and print reports as required. 4GLs produce a friendly environment to interface to the user. This environment comes in the form of screens and menus with selectable options, and help-files. This level of sophistication was not required for the current application; BASICA was found to suit and fulfil all the programming requirements.

An overview of the system software is shown in FIG 6.10. The software is straightforward, and has the task of interfacing between the operator and the system of microcontrollers. Having initialised arrays and error messages, the program redefines the function keys to give the operator a number of system options, such as: starting an experiment or displaying the latest on balance capacitance (see PHOTO 6.3). Linked with each command is a subroutine. Once a key is made active the system will respond to the pressing of that key by vectoring to the associated subroutine, and executing the commands contained there. The serial communications port is set up and enabled using the following two lines of code:

```
COM(1) ON
OPEN "COM1 : 4800,n,8.cs.ds.cd" FOR INPUT AS #1
```
An automatic timer is also set up which enables the capacitance data to be read at fixed intervals from MASCON.

During the calibration procedure messages are displayed on the VDU screen as the system completes each part of the process. These message strings are initialised at the start of the program, e.g.

\[ A(0) = '10aF arm positive calibration complete' \]

The software displays the system menu on the screen at the start of the program and then waits in a 'WHILE WEND' loop for operator requests. If the option 'Q' (quit from the program) is selected the variable QUIT is set to zero enabling the program to exit from the WHILE loop, and then to exit from the program.

The execution of each command is dealt with in the same way. The called subroutine will print messages to the screen to notify the operator of any further information that they need to enter. For example, on pressing 'F7' the operator selects to change the system gain (or sensitivity) setting. The subroutine associated with F7 will display the following message:

'To change the gain setting of the low noise amplifier and thus change the sensitivity of the capacitance bridge press the key of your choice. PLEASE NOTE THIS WILL OVER-RIDE SENCON.'

<table>
<thead>
<tr>
<th>SELECT KEY</th>
<th>GAIN (in dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>7</td>
<td>70</td>
</tr>
<tr>
<td>9</td>
<td>90</td>
</tr>
</tbody>
</table>

Enter a numerical key to indicate your choice.'

When the key is entered, the information will be sent with the command code to MASCON. As explained in (6.1.3) the command information is placed in the low nibble of the byte sent. Any additional information needed by MASCON in order to execute the command is placed in the high nibble of the byte sent. MASCON is
then able to extract this information.

A generalised flowchart representing the execution of a command is shown in FIG 6.11. When a function key is pressed the subroutine first stops the IBM timer. The timer periodically causes an interrupt which activates the request for the latest on-balance capacitance data from MASCON. Obviously it would cause confusion, if this were to occur whilst the IBM PC was in the process of executing another command. For the same reason all other function keys are temporarily disabled on entering a command subroutine. The IBM PC outputs the command byte to the transmit buffer of the serial port, and the program waits until the byte has been transmitted and then echoed back. If an error occurs in sending the command byte an error flag is set which in turn causes an error message to be printed on the VDU screen. If the byte is successfully transmitted the IBM PC awaits the outcome of its execution. Once the command has been executed MASCON notifies the IBM PC of its completion. On successful completion a favourable message such as: "the gain setting has been changed to choice number 5" is displayed on the screen, otherwise the message: "error detected, please try again" is written to the screen.

When the experiment is started the software requests the name of the file in which data is to be stored during the experiment. Similarly, if a calibration is requested, the program asks for the names of three files to store data: the calibration data; the converted calibration data; the limiting calibration data. This data can be displayed on completion of a calibration by using the option tL (control L) to list the calibration data in tables on the screen. It is wise for the operator to check the validity of the calibration data before commencing an experiment and taking data readings. The calibration section of the program provides the operator with clear messages as to the progress of the calibration procedure. These messages will aid the operator in finding out at what point the system failed, should an error have occurred.

Originally MASCON was set up to periodically send the latest on balance capacitance data to the IBM PC via the serial link. The IBM PC would then receive and store this data. The operator could select to change the times between data readings. This approach had a
potential problem: the operator may press a function key in the middle of a transfer of capacitance data to the IBM PC. To prevent this conflict the timer on the IBM PC was employed to cause the IBM PC to request the capacitance data periodically. During the execution of a command the timer is disabled, and at the end of the subroutine the timer is enabled again. This solution prevented the IBM PC from trying to talk to MASCON whilst MASCON was trying to talk to the IBM PC.

6.2.3 Measurement of Temperature via the IEEE Interface

As detailed in section 3.2.5 the sample temperature can be measured using a germanium resistance thermometer in the range 1.5K - 100K. The voltage drop across the resistance thermometer is measured using a Keithley nanovoltmeter. The voltmeter has an IEEE interface. A second temperature sensor positioned on the sample rod of the cryostat measures temperature in the range 1.5K - 300K. This second sensor is a Rhodium/Iron resistance thermometer. The 3120 temperature controller also has an IEEE interface. The voltmeter and the temperature controller can each be identified by their own unique address.

The IBM PC can communicate with these two instruments via an IEEE communications interface card. By transmitting the address of the instrument it wishes to talk to, data can be selectively read from either the voltmeter or the temperature controller.

The germanium thermometer requires a range of currents to be supplied at different temperatures (see section 3.2.5). These currents are supplied by a constant current source and can be selected by sending a binary code to the constant current source from an output port on the IBM PC. Two lines enable any one of four codes:

1mA : 11 ; 100μA : 10 ; 10μA : 01 ; 1μA : 00

A third port line reverses the polarity of the current through the sensor:

forward : 0 ; reverse : 1
A fourth port line provides a common signal ground. By knowing the current through the sensor, and the voltage drop across the sensor via the IEEE Interface, the IBM PC can calculate the resistance of the thermometer. Data calibration points of resistance versus temperature were stored in a file on the hard disk. The computer could then sort through the file using its calculated resistance values to find the corresponding temperature. Calibration points were stored every 50mK for the temperature range 1.5K - 100K. This detailed table was derived from the original calibration table in Appendix A. Cubic spline interpolation was carried out on the original calibration values, and data points were then taken from the fitted curve.

The temperature controller enables the temperature of the Rhodium/Iron sensor to be read directly via the IEEE. The commands for reading and writing to the 3120 temperature controller are found in the Oxford Instruments manual. As well as simply reading the temperature from the controller the IBM PC is able to control the system temperature by selecting a set temperature and monitoring power to the heater to keep the system at that temperature. This facility is available and easily implemented, but was not put into operation by the author. However, having the facility to control the temperature enables the operator to repeat a set of measurements for a small temperature range, or to monitor one particular temperature over a period of time.

Each time the latest on-balance capacitance data is read from MASCON a subroutine is called to measure the system temperature. The temperature data is then stored with the corresponding capacitance value.

6.2.4 Data Storage and Analysis

During an experiment the capacitance and temperature data is stored on the hard disk of the IBM PC. This bulk storage medium is able to store all the experimental data from a five day run (= 70Kb). For the initial experimental work carried out by Khan² and Brown³ the data was stored on paper by having a line printer dedicated to the experiment. This often caused problems such as: paper jamming within the printer, running out of paper, and faded printer ribbon.
producing unreadable output. These problems resulted in loss of data. Having the data stored directly on hard disk simplifies the collection and manipulation of results tremendously. Data can easily be summarized and then transferred to the University's main frame computer via Kermit (a software protocol enabling the transfer of data between a PC and the main frame).

Once the experimental data has been transferred to the main frame, the Fortran data analysis program\(^3\) can be used to represent the data graphically, and convert the data to thermal expansion versus temperature. The graph plotting and fitting routines on the main frame enable the analyst to fit a curve to the measured data points. The gradient of the curve fitted to the graph of the change in gap between the electrodes versus temperature is then used to calculate the thermal expansion versus temperature curve. Obtaining a good data fit is time consuming, and it requires an experienced analyst to achieve good results across the whole temperature range.

Traditionally PCs do not have the computing power nor the software application packages to accomplish the graph plotting tasks required. The algorithms involved are quite complex and require powerful number crunching facilities. To obtain real-time good quality graphics would also prove difficult. However the situation is changing with the arrival of the IBM 486 machine with its increased processing power and cache memory.

An alternative approach would be to carry out the data storage and analysis on a workstation. The cost of workstations is falling rapidly, and their areas of application are expanding into the PC arena. SUN in particular are pushing to take a share of the PC market. Workstations provide excellent screen graphics and ample computing power. A workstation would relieve the need to analyse the data on the main frame, and thereby localise the data collection and analysis within the same system.
(PHOTO 6.1) THE MASTER CONTROL MODULE: MASCON
CIRCUIT DESIGN FOR THE MASTER CONTROLLER SHOWING BOARD COMPONENTS AND PIN-OUT TO EDGE CONNECTOR (FIG 6.1).

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS232 Converter</td>
<td>Connects to IBM PC</td>
</tr>
<tr>
<td>UART 6402 (IC2)</td>
<td>Baud rate generator</td>
</tr>
<tr>
<td>MASTER CONTROLLER 8751 (IC1)</td>
<td>System controller</td>
</tr>
<tr>
<td>OVER VOLTAGE PROTECTION CIRCUIT</td>
<td>Protects against voltage spikes</td>
</tr>
<tr>
<td>POWER IN</td>
<td>Supplies power to the circuit</td>
</tr>
</tbody>
</table>

- **RS34.23 (IC4)**: Over voltage protection element
- **RS34.23 (IC4)**: Power supply for the circuit
- **R3, R4, R5, R6**: Resistors for voltage regulation
- **C6**: Capacitor for smoothing power supply
- **Q1**: Transistor for current amplification
- **R7, R8, R9**: Resistors for biasing
- **C1, C2, C3**: Capacitors for filtering
- **XTAL1, XTAL2**: Crystal oscillators
- **C30pF, C30pF**: Capacitors for frequency stabilization
- **+5V, GND**: Power supply and ground connections
- **TRC, TRR1, TRR2**: Transistor circuits
- **MR, INT0**: Input/output pins for controller
- **TXD, RXD**: Serial data input/output pins
- **B5, B6, B7**: Reset control pins
- **R1, R2, R3**: Resistors for current limiting
- **C1, C2**: Capacitors for filtering
- **+5V, GND**: Power supply and ground connections
- **IC6, IC1, IC2**: Integrated circuits for various functions
- **TX: RX**: Serial link to slaves
- **TXD, RXD**: Serial data input/output pins

The diagram shows the interconnected components and their respective functions in the circuit design for the master controller.
CIRCUIT DESIGN FOR THE MASTER CONTROLLER SHOWING CONNECTIONS TO THE OTHER CONTROLLERS (FIG 6.2).

The diagram shows the connections between the IBM AT PC and other controllers. The schematic includes various components such as ICs, resistors, capacitors, and connectors, indicating the flow of signals and power through the system. The circuit design is designed to facilitate communication between the master controller and the slaves, possibly through a serial link to IBM.

Key components include:
- **IBM AT PC**
- **RS232 Converter**
- **8751 Microcontroller**
- **Baud Rate Generator**
- **8402 UART**
- **Slave Controllers**

The connections are detailed with specific pin numbers and voltages, ensuring proper signal transfer and operation within the system.
START

INITIALISE REGISTERS AND PARAMETERS
RESET EACH SLAVE CONTROLLER IN TURN

HAS THE IBM COMMANDED THE EXPERIMENT TO START?

NO

HAS THE IBM REQUESTED A CALIBRATION?

NO

YES

RESET 'COMMAND RECEIVED' VARIABLES
CARRY OUT THE CALIBRATION

RESET 'COMMAND RECEIVED' VARIABLES

START THE EXPERIMENT RUNNING ON EACH SLAVE IN TURN

YES

HAS AN ERROR OCCURRED DURING SET-UP?

NO

IS EXPERIMENT STILL RUNNING?

YES

NO

HAS A COMMAND BEEN RECEIVED FROM THE IBM?

YES

EXECUTE THE SELECTED COMMAND

RESET THE 'COMMAND RECEIVED' VARIABLES

NO

STOP THE EXPERIMENT RUNNING ON EACH SLAVE IN TURN

DELAY FOR 20ms.
**MASCON: FLOWCHART FOR THE MAIN SOFTWARE MODULE (FIG 6.4)**

**MAIN PROGRAM:** Do while 1 (i.e. go forever)

Initialise interrupt lines to prevent a false external interrupt. Set external interrupt 0 to high priority, this interrupt occurs when the IBM PC has a command byte for MASCON. Select 9-bit UART mode with variable data rate for the Serial Port. Clear SM2 to enable reception of serial data bytes (9th bit = 0). Set REN to enable serial data reception. Set DATA_RECEIVED_RESET as an input line. It is used by the UART to indicate that it has a command byte to transmit from the IBM PC. Set and clear the MASTER RESET pin to clear UART status flags.

Clear EXPT$IS$RUNNING: the experiment is not running.
Clear CAL_BIT: a calibration has not been requested.
Clear BYTE$SENT: a serial data byte has not been sent.
Clear BYTE$RECEIVED: data byte has not been received.
Set up TIMER 1 with the auto-reload value to give a baud rate of 3125 bits/second for the Serial Port. Select low level triggered interrupts.
Clear DO_BYTE. used to store serial data bytes from slaves.
Set POINTER-0 to point to the 1st byte of TABLE array, used to store command codes.

Reset each slave controller in turn.

Do while the experiment has not commenced.

<table>
<thead>
<tr>
<th>No</th>
<th>Has the IBM PC requested a calibration?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do</td>
<td>Decrement the pointer by 1 &amp; clear COMMAND BYTE RECEIVED bit now that the command has been received to start the calibration.</td>
</tr>
<tr>
<td></td>
<td>Carry out system calibration (call SELF_CALIB).</td>
</tr>
</tbody>
</table>

Decrement the pointer by 1 & clear COMMAND BYTE RECEIVED bit now that the command has been received from the IBM PC to start the experiment.

Start the experiment going on each slave in turn.

Yes | Has an error occurred in starting the experiment?

| Do | COMMENCE: Do while the experiment is running.

| N | Execute the selected command by calling the DO_CASE(.COMMAND) procedure.

| Y | Clear the COMMAND BYTE RECEIVED bit now that the command byte has been executed.

Stop the experiment running on each slave in turn.

Delay for 20mS, this gives the UART time to transfer the last data byte before being reset during initialisation at the start of the program.
**MASCON: FLOWCHART FOR THE IBM INTERRUPT PROCEDURE (FIG 6.5)**

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read &amp; store the command byte from the IBM (read via UART).</td>
<td>Toggle the DATA RECEIVED_RESET (DRR) pin to indicate to the UART that the command byte has been read.</td>
</tr>
<tr>
<td>Do while INT0=0: wait for the interrupt line to go high this is</td>
<td>caused by DATA RECEIVED going low when DRR is reset.</td>
</tr>
<tr>
<td>Do while TBRE=0: wait till UART transmit buffer is empty.</td>
<td>Echo command byte back to the IBM PC via the UART.</td>
</tr>
<tr>
<td>Strobe TBR (TRANSMIT BUFFER REGISTER) to request data be transferred</td>
<td>Output a carriage return byte to the UART to signal the end of the message to the IBM PC.</td>
</tr>
<tr>
<td>to the transmit register of the UART.</td>
<td>Strobe TBR to request data transfer to transmit register.</td>
</tr>
<tr>
<td>Do while INT0=1: wait till IBM sends a confirmation byte.</td>
<td>Read &amp; store confirmation byte from the IBM (read via UART)</td>
</tr>
<tr>
<td>Toggle DRR pin to indicate that the byte has been read.</td>
<td>Clear any pending interrupts on INT0 which may have been caused during polling.</td>
</tr>
<tr>
<td>Do while INT0=0: wait till interrupt level returns high,</td>
<td>No Confirmation byte states 'no error detected'? Yes</td>
</tr>
<tr>
<td>this is caused by the DR pin going low when DRR is reset.</td>
<td><strong>Do:</strong></td>
</tr>
<tr>
<td>Store the received command byte at position 'POINTER' within the</td>
<td>Set COMMAND_BYTE_RECEIVED to 'YES'.</td>
</tr>
<tr>
<td>TABLE array.</td>
<td>N Has the IBM commanded the experiment to start? Y</td>
</tr>
<tr>
<td>Set EXPT$IS$RUNNING bit to '1'.</td>
<td>Y Has the IBM commanded the calibration to start?</td>
</tr>
<tr>
<td>Set the CAL$BIT to '1'.</td>
<td>Increment POINTER by 1 to point to the next available byte in the TABLE array.</td>
</tr>
</tbody>
</table>
The 'F3' key has been pressed on the IBM, this selects the command to send the latest off-balance capacitance (OBC) to the IBM. OBC consists of 3 bytes: 1 sign byte and 2 OBC digits.

The 'F4' key has been pressed on the IBM, this selects the command to change the system sensitivity to SENCON.

The 'F5' key has been pressed on the IBM, this selects the command to start the current gain setting from SENCON.

The 'F9' key has been pressed IBM allows the delay time between each update of the capacitance value to be changed.

If key 'F9' is pressed IBM allows the delay time between the on-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the off-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the on-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the off-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the on-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the off-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the on-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the off-balance capacitance to be changed.

If key 'F9' is pressed IBM allows the delay time between the on-balance capacitance to be changed.
Zero the ERROR$FLAG and the CHECK. SUM byte.

Address MOTCON & request the latest on-balance capacitance data to be sent to the master, (call CHAT TO SLAVE(4,1)).

Wait up to 5 seconds for MOTCON to reply.

No Is MOTCON ready to send the capacitance data?  Yes

Set ERROR flag to 1 i.e there has been an error

Do Clear DO BYTE used to store serial data from MOTCON. Disable Serial Port interrupt.

Do I=0 to 10: I/P capacitance digits, checksum, code

Load serial buffer with a request to MOTCON to send a data byte.

Transmit the data byte (call TRANS).

Wait up to 5 seconds for MOTCON to reply.

No Has a byte been received (RI=1)?  Yes

Set I=10 to leave loop.  Store byte in the capacitance data array.

Clear the received interrupt flag (RI=0).

Re-enable the Serial Port interrupt.

Do I=0 to 9

Sum the capacitance digits & information code byte
Store the result in a checksum byte.

Yes (MASCON's checksum = MOTCON's checksum)?  No

Set ERROR$FLAG to '1'

Do I = 0 to 9

Convert the capacitance digits to ASCII, ready to send to the IBM PC.

Send the information code byte to the IBM PC and delay for 20mS to allow time for the byte to transfer.

Do I = 0 to 8

Output each of the nine capacitance digits (1aF through 100pF decades) to the IBM PC.

Output 'CARRIAGE RETURN' to the IBM PC to signal the end of the string of capacitance digits.
(PHOTO 6.2) TESTING THE MICROELECTRONIC CONTROL SYSTEM
FLOWCHART FOR THE CALIBRATION SOFTWARE MODULE

START

1. Disable external interrupt 0
2. Initialize cal-bit and do-byte to 0

3. Command each slave to start calibration and monitor feedback information from OBCCON

4. Wait for the 'calibration complete' message from OBCCON. Delay for 10ms.

5. Is the calibration complete?
   - Yes: Send a message to the IBM, stating that the calibration is complete. Delay for 10ms.
   - No: Send the calibration data to the IBM. If the data transfer is successful, set U=3.

6. Loop counter variable U is set to zero

7. Is the loop counter greater than 3 (U>3)?
   - Yes: Increment U
   - No: Read the calibration data from OBCCON

8. Wait for the 'convert calibration data complete' message from OBCCON. Delay for 10ms.

9. Is the data conversion complete?
   - Yes: Send a message to the IBM, stating that the converted calibration data is ready. Delay for 10ms.
   - No: Send a message to the IBM, stating there is a problem. Delay for 10ms.
INCREMENT U

LOOP COUNTER VARIABLE 'U' IS SET TO ZERO

IS LOOP COUNTER > 3 ?

READ THE CONVERTED CALIBRATION DATA FROM OBCCON

SEND THE CONVERTED CALIBRATION DATA TO THE IBM. IF THE TRANSFER IS SUCCESSFUL SET LOOP COUNTER U=3

WAIT FOR THE LIMITING CALIBRATION DATA AVAILABLE MESSAGE FROM OBCCON. DELAY FOR 10 ms.

IS THE LIMITING CALIBRATION DATA AVAILABLE ?

SEND A MESSAGE TO THE IBM STATING THAT A PROBLEM HAS OCCURRED DELAY FOR 10 ms.

YES

SEND A MESSAGE TO THE IBM STATING THAT THE LIMITING CALIBRATION DATA IS READY. DELAY FOR 10 ms.

LOOP COUNTER VARIABLE 'U' IS SET TO ZERO

IS THE LOOP COUNTER (U-3) ?

READ THE LIMITING CALIBRATION DATA FROM OBCCON.

SEND THE LIMITING CALIBRATION DATA TO THE IBM. IF THE TRANSFER IS SUCCESSFUL SET U = 3. DELAY FOR 10 ms.

SEND THE LIMITING CALIBRATION DATA TO SENCON, DELAY FOR 10 ms.

SEND MESSAGE TO THE IBM STATING THAT CALIBRATION IS COMPLETE

ENABLE EXTERNAL INTERRUPT 0

END
Enable external interrupt 0 so that the pin can be polled during the calibration. Clear the CAL_BIT which caused the SELF_CALIB procedure to be called. Clear DO_BYTE ready to receive a serial data byte from the slave controllers.

Command each slave in turn to start the calibration process.

Monitor the calibration feedback information from OBCCON.

Wait to receive a calibration complete message from OBCCON.

Delay for 10mS to ensure UART has transferred data to IBM.

<table>
<thead>
<tr>
<th>N</th>
<th>Does the received message say calibration complete?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Clear DO_BYTE containing message from OBCCON. Send 'calibration complete' message to the IBM PC.</td>
</tr>
<tr>
<td></td>
<td>Delay for 10mS</td>
</tr>
</tbody>
</table>

**Do** $U=0 \to 3$

- Read the calibration data and the checksum bytes from OBCCON and send them to the IBM PC.
- Wait for confirmation. If transfer successful set $U$ to 3

Wait to receive a 'converted cal. data available' message.

<table>
<thead>
<tr>
<th>N</th>
<th>Is the converted calibration data available?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Send a message to OBCCON. Send 'converted calibration data available' message to the IBM PC.</td>
</tr>
<tr>
<td></td>
<td>Delay for 10mS</td>
</tr>
</tbody>
</table>

**Do** $U=0 \to 3$

- Read the converted calibration data and checksum bytes from OBCCON, and send them to the IBM PC.
- Wait for confirmation. If transfer successful set $U$ to 3

Wait to receive a 'limiting cal. data available' message.

<table>
<thead>
<tr>
<th>N</th>
<th>Is the limiting calibration data available?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>Send a message to OBCCON. Send 'limiting calibration data available' message to the IBM PC.</td>
</tr>
<tr>
<td></td>
<td>Delay for 10mS</td>
</tr>
</tbody>
</table>

**Do** $U=0 \to 3$

- Read the limiting calibration data and checksum bytes from OBCCON and send them to the IBM PC.
- Wait for confirmation. If transfer successful set $U$ to 3

Send the limiting calibration data to SENCON. Delay 10mS. Send message to IBM stating: 'CALIBRATION COMPLETE'. Re-enable external interrupt 0 ready for the experimental mode of operation when the IBM PC sends a command byte.
START

SET VARIABLE QUIT TO 1

INITIALISE PARAMETERS
INITIALISE ARRAYS

SET UP ERROR MESSAGES
AS STRINGS

DEFINE FUNCTION KEYS
AND POSITIONS OF
ASSOCIATED SUBROUTINES
TO EXECUTE COMMANDS

SET UP SERIAL PORT
FOR COMMUNICATIONS WITH
THE MASTER CONTROLLER
AND ENABLE CHANNEL

SET UP CALIBRATION
MESSAGES AS STRINGS

DECLARE CALIBRATION
ARRAYS TO STORE DATA

DISPLAY MENU OF USER
FUNCTIONS ON SCREEN
(GOSUB 17000)

HAS QUIT BEEN SET TO ZERO
TO END PROGRAM

YES

END

NO
### System Menu Display on the IBM PC

<table>
<thead>
<tr>
<th>Key</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>start the experiment running</td>
</tr>
<tr>
<td>F2</td>
<td>stop the experiment running</td>
</tr>
<tr>
<td>F3</td>
<td>read and display the latest off-balance capacitance</td>
</tr>
<tr>
<td>F4</td>
<td>return control of the amplifier gain to the system</td>
</tr>
<tr>
<td>F5</td>
<td>change the delay time between updating the c.b.c.</td>
</tr>
<tr>
<td>F6</td>
<td>change the time-constant for the phase detector</td>
</tr>
<tr>
<td>F7</td>
<td>change the gain on the amplifier (over-rides system)</td>
</tr>
<tr>
<td>F8</td>
<td>display the current gain setting for the amplifier</td>
</tr>
<tr>
<td>F9</td>
<td>change delay between updating current capacitance</td>
</tr>
<tr>
<td>F10</td>
<td>display the current on balance capacitance value</td>
</tr>
<tr>
<td>^C</td>
<td>(control C) start the calibration running</td>
</tr>
<tr>
<td>^Q</td>
<td>(control Q) quit from the basic program</td>
</tr>
<tr>
<td>^M</td>
<td>(control M) displays this menu</td>
</tr>
<tr>
<td>^L</td>
<td>(control L) lists the calibration data</td>
</tr>
</tbody>
</table>

![System Menu Display](image)

(Photograph 6.3) THE SYSTEM MENU DISPLAY ON THE IBM PC
GENERALIZED FLOW CHART TO REPRESENT COMMAND EXECUTION SEQUENCE ACTIVATED ON PRESSING A FUNCTION KEY (FIG 6.11)

START

STOP TIMER, TURN KEY(0) OFF, CLEAR VDU SCREEN

LOAD THE COMMAND NUMBER INTO X

OUTPUT THE COMMAND NUMBER TO THE MASTER (GOSUB 18000)

HAS AN ERROR OCCURRED IN TRANSMITTING THE COMMAND?

NO

WAIT TO RECEIVE MASTER'S REPLY TO THE COMMAND

YES

STORE REPLY FROM MASTER

HAS THE COMMAND BEEN EXECUTED SUCCESSFULLY?

NO

PRINT AN ERROR MESSAGE TO THE SCREEN

PRINT A 'COMMAND EXECUTED SUCCESSFULLY' MESSAGE

YES

PRINT AN ERROR MESSAGE TO SCREEN AND TURN ON TIMER

TURN KEY(0) ON AGAIN

RETURN
7.1 Initial Experiments Using Epoxy Resin

The initial experiments were carried out using two samples of epoxy resin. The author was commissioned by Oxford Instruments (who supplied the samples) to investigate the thermal expansion of the epoxy resin over the temperature range 5K - 300K. Oxford Instruments use the material to support superconducting magnets within their cryostats. They needed to determine any change in position of the magnet due to expansion or contraction of the resin. If the position of the magnet changes then the position of the magnetic field will change. This could result in an experimental sample originally placed within the homogeneous part of the magnetic field, being within a non-homogeneous part of the field as the temperature falls.

These preliminary experiments enabled the author to familiarise herself with the experimental instrumentation, low temperature techniques, and vacuum equipment. Problems and difficulties encountered at this stage were then taken into consideration in the design of the distributed automated control system.

One of the difficult aspects of the experiment is the analysis of the data. An analysis program was previously developed by I.J. Brown\(^1\), (1981). The program uses \(\beta\)-cubed splines to fit a curve to the experimental data points. The program has been successfully used to fit data over small temperature regions (\(\approx10K\)); the degree of difficulty is substantially increased when a good data fit is required over the full temperature range (5K-300K). The program displays the experimental data as: 'change in the size of the gap between the copper electrodes against temperature'. Having made corrections for the expansion of the capacitance cell, this data corresponds to the change in length of the sample with temperature. In the normal mode of operation corrections are made for the radial expansion of the low potential electrode (see electrode 2 on FIG 3.6), and for the axial expansion of the annular cell (if the cell is made of copper). These corrections were made using the expansion data for copper as detailed by Kroegar and Swenson\(^2\), (1977). This expansion data is expressed as a power series.
polynomial expansion in temperature and is immediately applicable to
analysis using a digital computer. The program differentiates the
curve fitted to the experimental data points to produce a graph of
linear thermal expansion versus temperature.

7.1.1 Method

The form of the capacitance dilatometer cell used for the experiment
was as given in (3.2.2). The two samples used were cylindrical rods
of identical length enabling the same capacitance dilatometer cell
and electrode system to be used for both experiments. The following
experimental details and parameters apply to both sample I and
sample II.

Radius of low potential electrode (2) = 9.165 x 10^{-3}m
Width of stycast gap between electrode and guard ring = 6.5 x 10^{-5}m
Epoxy resin specimen length = 5.000\pm0.001 x 10^{-2}m

Sample I:
Width of the gap between electrodes at 295.0K = 4.39673 x 10^{-4}m

Sample II:
Width of the gap between electrodes at 296.7K = 4.27696 x 10^{-4}m

The width of the gap is calculated by a computer program named GAP
(developed by I.J. Brown\(^1\)) which deduces the result using the above
given data. For the optimum solution a radius of 9mm was selected
for the low potential electrode, and the initial gap must be greater
than 2 x 10^{-4}m to avoid erroneous effects due to non-parallelism and
surface roughness of the capacitance plates. The reasons behind
selecting these criteria are given in reference 3 and 1.

7.1.2 Results

Sample I:

The sample was cooled to liquid helium temperatures and results were
obtained for the range 5K - 290K. The graph in FIG 7.1 displays the
change in sample gap between the electrodes for the temperature
range 5K - 290K. Details of the results for the low temperature
range can be seen in FIG 7.2. Using I.J. Brown's analysis program\textsuperscript{1} \(\beta\)-cubed spline curves were fitted to the data points. The fitted curves are indicated by the solid line in both graphs. To aid and improve the accuracy of the fitted curves 'knots' have been incorporated. A 'knot' is indicated by a dashed vertical line on the graph. Each section of data between two 'knots' has been fitted by a single \(\beta\)-cubed spline curve. The first and second differentials of the curve either side of the 'knot' match exactly at the knot boundary.

The solid fitted line is then used by the analysis program to produce the thermal expansion versus temperature results shown in FIG 7.3. For the high temperature region (>235K) the thermal expansion results begin to curve downwards. This is believed to be a consequence of the fitting procedure. Alternative 'fits' to the same data produced an upward tilting curve for this temperature range. In discussions with Dr. Greenough of the Applied Physics Department, Hull University, it was discovered that other workers using \(\beta\)-cubed spline fitting encountered similar problems in attempting to 'clamp' the end of the fitted curve.

The results indicate a small coefficient of linear thermal expansion for the range 5K-50K:

\[ 0.5 \times 10^{-7} \text{K}^{-1} < \alpha < 4 \times 10^{-7} \text{K}^{-1} \]

The value of \(\alpha\) rises steeply between 50K and 170K. Above 170K the value of \(\alpha\) begins to level off giving a maximum value of \(\alpha = 6.8 \times 10^{-6} \text{K}^{-1}\) at 230K.

**Sample II:**

The change in gap for the temperature range 5K-30K is shown in FIG 7.4. A good fit was obtained by weighting the points selectively and placing a 'knot' at 20K. Placing a knot ensures that the first and second derivatives of the function are constant across the knot boundary. The change in gap for the range 5K-290K is shown in FIG 7.5 The graphs of the corresponding \(\alpha\) are given in FIG 7.6 and FIG 7.7 The thermal expansion coefficient remains small and positive for the range 5K-20K: \(2 \times 10^{-8} \text{K}^{-1} < \alpha < 29 \times 10^{-8} \text{K}^{-1}\). Above 20K \(\alpha\)
rises steeply and then begins to level off around 150K, reaching a maximum of $\alpha = 18.6 \times 10^{-6}$ at 290K.

7.1.3 Discussion

Both epoxy resin samples display similar expansion curves: small coefficients of thermal expansion at low temperatures, followed by a steep rise in $\alpha$, culminating in leveling off of the thermal expansion towards room temperature (290K). However, $\alpha$ for sample II, at 290K, is nearly three times that of sample I. A rapid increase in the expansion coefficient is observed above 50K for sample I, whilst for sample II the increase commences as low as 20K.

Little more can be said about the results since the details of composition and specification of manufacture are unknown due to commercial considerations. The author wishes to thank Oxford Instruments for granting permission for the publication of these results.

7.2 Polydiacetylene Single Crystal

Previous researchers have studied the thermal expansion of a wide variety of semi-crystalline or amorphous materials. However comparatively little information is available on the thermal expansion behaviour of polymer single crystalline materials. An initial experiment was carried out to study the thermal expansion of polydiacetylene bis (p-toluene sulphonate) of 2,4 hexadlyne - 1,6 dial (PDA-TS) over the temperature range 4.2K-300K. Polydiacetylene was of particular interest because of its potential use as a device material within molecular electronics. GEC Research Laboratories were investigating the possibility of creating a molecular binary counter by utilising the polydiacetylene structure. Large single crystals of polydiacetylene with a high degree of perfection were obtained by solid state polymerisation of the monomer material. The structure of the polydiacetylene chain is shown in FIG 7.8. The specimen PDA-TS crystals used in this experiment were prepared by N.J. Poole, and loaned to us by M.N. Wybourne, both of GEC Research Laboratories. At high temperatures, each unit cell contains two polymer chains extending in the crystallographic 'b' direction,
(FIG 7.9); then at 195K, PDA-TS undergoes a second order phase transition. Below this transition temperature the space group remains the same, but the unit cell is doubled in size and contains two different species of the polymer chain. This structural phase transition is caused by an 8-degree rotation of the massive side groups attached to the polymer backbone. These side groups are oriented along the (102) planes of the high temperature structure, with adjacent rows having opposite directions of rotation.

7.2.1 Method

A special version of the capacitance dilatometer cell, (type B), was built to suit the thermal expansion measurements of small pieces of crystal (FIG 7.10). Three identical pieces of PDA-TS were placed symmetrically between the electrode plates (1 and 2) as 'spacers'. As a consequence of placing the crystal pieces as spacers any expansion of the crystal will directly result in an increase in the electrode displacement and a subsequent decrease in the capacitance of the dilatometer cell. This is in direct contrast to the previous experiment which used a cell of type A, whereby expansion of the epoxy resin caused the electrodes to move closer together resulting in an increase in capacitance. To allow for this arrangement the computed results obtained on analysis of the crystal data have been inverted, and the inversion results then plotted to give graphs of linear thermal expansion coefficient against temperature.

Experimental numerical data:

Gap between the electrodes at 297.45K = 5.75976 x 10^-4 m
Polydiacetylene specimen length at 297.45K = 5.75976 x 10^-4 m

Using the three-terminal capacitance system, specimen length changes in the direction perpendicular to the carbon backbone, (known as the direction a*), were measured for the temperature range 4.2K - 300K. This data was then analysed to obtain the linear coefficient of thermal expansion for the single crystal polydiacetylene in the a* direction, which should comprise a linear combination of the temperature dependence for the lattice parameters a and c.
7.2.2 Results

The change in gap between the electrodes corresponds directly to change in thickness of the crystal with temperature. The change in crystal thickness over the temperature range 160K - 230K can be seen in FIG 7.11; this graph clearly depicts the phase transition at 195K. Distinct minima can be observed, (FIG 7.12): the first centres on 195K reflecting the known phase change involving torsional rotation of the side groups; the second occurs at 45K corresponding to a negative coefficient of thermal expansion at lower temperatures. It is interesting to note that a maxima occurs around 159K - this however is the phase transition temperature for the monomer crystal.

The thermal expansion is indicated as being positive above 195K (FIG 7.13) with a value of $\alpha = 4.3 \pm 0.1 \times 10^{-5} K^{-1}$ at 280K.

A small interesting feature was observed at 35K, (FIG 7.14), which appears as a small anomaly. This may have been caused by low lying optical modes freezing out. Clearly further work is necessary to clarify these results.

7.2.3 Discussion

The value for $\alpha_a^*$ obtained at 280K is consistent with other values of $\alpha_a$ and $\alpha_c$ as published by other workers$^6,7$. PDA-TS is unusual in exhibiting a positive coefficient of expansion at room temperature which was observed in agreement with Batchelder$^8$. Negative expansion at room temperature is a common observation in polydiacetylenes with PDA-TS being the exception$^9$. Baughman$^{10}$ has argued that thermally excited torsional motions of the chain contribute significantly to the observed negative thermal expansion coefficients in polydiacetylene. On the other hand Batchelder has suggested that these torsional motions are of secondary importance which would seem to be the case for PDA-TS. Also in agreement with Batchelder, a negative thermal expansion was observed at lower temperatures, below 50K. Batchelder observed negative thermal expansion below 70K, but his low temperature evaluations are based on only four data points$^8$.

At 195K the phase change occurs involving torsional rotation of the
polymer side groups and, as a result of the different nuclearation centres, domains are formed. The domain walls act as the boundaries between regions where the side group rotations are in the opposite sense. At these boundaries there will be a re-orientation of the structure, akin to a martensitic transformation, and therefore morphological changes will occur at the surface. These are expected to manifest themselves as ridges which would give rise to the apparent increase in thermal expansion.

The massive side groups attached to the polymer backbone give rise to optical phonons with energies compatible to those of the acoustic phonons responsible for heat transport. In PDA-TS there is experimental evidence for several optical modes involving motion of the polymer side groups which have energies sufficiently low to interact strongly with acoustic phonons. Work by Wybourne et al. has lead to the suggestion that the feature at 35K may be a low lying optical mode. Far-infrared spectroscopy has shown the existence of a number of low energy modes down to ~15cm\(^{-1}\) in PDA-TS. Recent neutron scattering data for TS monomer have shown the existence of optical modes down to energies ~3cm\(^{-1}\); these should be similar in characteristic for both the monomer and the polymer since modes primarily involving motion of the side groups should not be strongly affected by the formation of the polymer chain. Present available data shows strong evidence of coupling between acoustic and optical phonons.

It is helpful at this stage to see if the previous work, using infra-red spectroscopy and neutron scattering techniques, is consistent with the thermal expansion anomaly at T=35K. To determine this an estimate of the range of energy level splitting required to produce a Schottky anomaly at 35K is made. If the Grünesien parameters for the system are equal, then it is expected that the temperature position of the anomaly will be the same for both the thermal expansion curve and the specific heat capacity curve.

For a graph of specific heat against \(x\), (where: \(x = \frac{T}{\Delta}, T = \) temperature in Kelvin, \(\Delta = \) energy level splitting), the Schottky anomaly will occur at: \(x \approx 0.5\)

In this case an anomaly has been observed at T= 35K, and so
assuming equal Grüneisen parameters: \( \Delta = \frac{T}{x} \)

\[
\Delta = \frac{35}{0.5} = 70
\]

Boltzmann constant \( k = 1.381 \times 10^{-23} \text{JK}^{-1} \)

\[
\Delta = 70 \times k \text{ Joules}
\]

1eV = \( 1.61 \times 10^{-19} \) Joules

Range of energy level splitting: \( \Delta \approx 6 \text{meV} \)

This result is consistent with the work carried out by Wybourne et al using far-infrared and neutron scattering techniques.

7.3 Amorphous Materials

7.3.1 Theory of Thermal Expansion and Structure of Glass

I The Two Level System

A generally accepted theoretical model, which appears to explain the experimental observations on dis-ordered solids, is that of the two level system (TLS). The model proposes the existence of a broad energy band of localised excitations which may be strongly coupled to phonons. These excitations are described as having one ground state and one excited state; these two states being well separated from states of higher energy.

FIG 7.15 shows two potential energy minima being separated by a large potential barrier. The origin of the two level system may be due to the tunnelling of some entity or configuration between the two potential energy minima. However the microscopic existence of the two level system has not been established for any amorphous material.

It has been suggested that the tunnelling nature of the two level system may be investigated by a measurement of the TLS contribution.
to thermal expansion. This suggestion arose from observing the effect of impurities within crystals on the thermal expansion coefficient of the crystal: this coefficient can become very large if the impurities re-orient by tunnelling.

The Grüneisen Parameter, \((\gamma)\), for these types of materials may also be large in magnitude \((\approx 300)\). Thus, if tunnelling does give rise to the two level system in amorphous solids, one might expect the \(\gamma\) of glasses to be anomalously large.

The linear coefficient of expansion for vitreous silica is negative at all temperatures below 150K \((\text{Hahn}^{17})\). However, not until temperatures below 4K does the Grüneisen parameter become large \((\gamma > 4)\) relative to that expected from thermal phonons \((\text{White}^{18})\). This result coincides with the view that the large negative \(\gamma\) at lower temperatures is associated with the localised two level system.

**II Proposed Structure of Vitreous Silica**

The structure of vitreous silica is not known with any certainty. The following three models have been proposed based on existing experimental evidence, and may allow some insight into understanding disordered solids. A detailed review is given by Ford\(^{19}\), and is summarized below.

**Random Network Theory**: Proposed by Zachariasen\(^{20}\) in 1935. It was suggested that 'glass consists of a continuous assembly of atoms in which there is no unit cell repeated regularly, but where there are definite atomic coordinates and interatomic distances'.

This model is supported by X-ray diffraction results by Warren\(^{21}\) where a pattern of diffuse rings was observed. The pattern obtained for an amorphous material is that of diffuse rings, whilst sharp rings are obtained for a polycrystalline material.

**The Crystallite Theory**: The X-ray diffraction results can also be interpreted in terms of the crystallite theory. This theory proposes small regions of crystalline silica packed together in random orientation. The suggestion is that the broad rings arise from cristobalite crystals, a crystalline form of silica, so small that
they broaden the sharp lines obtained for cristobalite. The theory proposes that it should be possible to calculate the average size of the cristobalite crystals in the vitreous silica from the powder pattern. If this theory is assumed, an average size of $8 \text{Å}$ is obtained. Since within cristobalite the edge of the unit cell is $7 \text{Å}$, it follows that the crystals, within vitreous silica, are approximately one unit cell in size.

The x-ray diffraction results, with the exception of the information obtained on small angles of scatter, support both the Random Network Theory and the Crystallite Theory. There is an absence of small angle scatter within vitreous silica, which implies an unbroken system of bonding without boundaries or voids. Thus a view is obtained of single cristobalite crystals, and beyond the single crystals continuous bonds in all directions.

Warren\textsuperscript{21} made use of the Radial Distribution Function (RDF) obtained from X-ray scatter by Fourier analysis, to investigate the structure of vitreous silica. This technique which is generally applicable to liquids, amorphous solids, and crystalline solids, makes no prior assumptions as to the structure of the material under investigation. From the X-ray data the proposed structure was this: 'each silicon atom is tetrahedrally surrounded by four oxygen ions at $1.62 \text{Å}$, and each oxygen ion to be shared between two silica ions'. The Si-O-Si bond was thought to be almost linear: $(150^\circ-180^\circ)$, with the orientation of the two connected tetrahedra being quite random about the Si-O-Si bond.

The Penta-gonal Dodecahedral Model: The proposed theory was that the structure of vitreous silica was that of a packed jumble of polyhedra. The only polyhedra that was found to fit geometrical considerations, density calculations, and diffraction results was that of Penta-gonal Dodecahedral (PD). Taking this view it is thought that the edges of the polyhedra are the Si-Si distances, and the oxygens lie on these edges between the silicones, assuming the Si-Si bond angle to be $180^\circ$. The angle between any two faces on the PD is $116^\circ$. Since an angle of $120^\circ$ is required to completely fill the space the PD must be distorted slightly to achieve that objective. It is also thought that the pentagonal rings within the PD model are puckered (i.e. not plane), this would be in keeping
with the three main crystalline forms of silica which are made up of six-sided puckered rings.

III The Thermal Expansion of Silica and Lead Silica

Experimental results of Hahn and Kirby\textsuperscript{17} have shown silica to have a small linear coefficient of thermal expansion ($\alpha \approx 0.5 \times 10^{-6} K^{-1}$ at a temperature of 300K). The coefficient is seen to go negative around 180K. Low temperature results by White\textsuperscript{18} indicate an $\gamma$ of -40 to -50 for temperatures below 3K. This large and negative Grüneisen parameter is indicative of a tunnelling process.

It is believed that low frequency, shear vibrational modes are responsible for the negative expansion of silica at low temperatures\textsuperscript{19,22}. The most likely transverse vibration is that of the oxygen atoms perpendicular to the Si-O-Si bond. These oxygen atoms are referred to as "bridging oxygens" as they form a bridge between two silicon atoms. It is thought that when an oxygen atom is displaced from its equilibrium position, and there are no near neighbours to exert restoring forces, the effect will be to pull the atoms closer together, thereby contracting the network. In order for this to happen it is necessary for the shear modulus to be low and for there to be room for the atoms to move in this way.

Lead silicate can be formed with over 80 mol % PbO. From density calculations it appears that initially lead enters the silicon network substitutionally without destroying the silicon-oxygen chains. The lead atom, having a high mass, is thought to have a damping effect on this transverse vibration of the oxygen atoms leading to a decrease, or removal of, the negative expansion coefficient.

7.3.2 Experimental Details

This short study into the thermal expansion of vitreous silica and lead silicate was instigated by N. Ford\textsuperscript{19} and R.D. Greenough of the Applied Physics department, University of Hull. Ford's experimental results indicated the possibility of negative thermal expansion at low temperatures (below $\approx 25K$). Although the results inferred this to be the case, the interferometer technique used by Ford lacked
accuracy at low temperatures. The results below 20K may be in error by as much as 75%. It was hoped that the use of capacitance dilatometry by the author, on the same samples, would help to clarify the situation.

The two samples used in this study were a pure silica sample (to be used as a reference sample), and a lead silicate sample: PbO - weight 66.5%, SiO₂ - weight 33.5%. Both samples were the same geometric shape: hollow cylinders. The silica sample was of length 25.39mm, and the lead silicate sample was 19.31mm in length. Measurement of the linear thermal expansion was made in the range 4.2K - 300K for both samples. The expansion of each sample was measured relative to that of copper using a copper capacitance cell (see 3.2.2). This proved to be a difficult task in the case of silica since its expansion coefficient (\(\alpha \approx 0.5 \times 10^{-6}K^{-1}\)) is very small relative to that of copper at room temperature.

7.3.3 Lead Silicate Results

The linear thermal expansion coefficient (\(\alpha\)) for the lead silicate sample is shown graphically in FIG 7.16. At a temperature of 280K the linear thermal expansion coefficient is of the order of 11.4 \(\times 10^{-6}K^{-1}\). A sharp decrease in \(\alpha\) is observed at around 80K, with \(\alpha\) decreasing almost linearly with temperature between 80K and 40K. At temperatures below 30K \(\alpha\) tends towards zero expansion; that is no negative expansion is indicated. However on closer inspection (FIG 7.17) it is apparent that \(\alpha\) appears to go slightly negative below \(7K\). There are too few data points below \(7K\) to say with any certainty that a negative \(\alpha\) exists. Since this negative coefficient is so small (\(\approx 5 \times 10^{-8}K^{-1}\)) it may well be a consequence of the \(\beta\)-spline fitting.

Ford's results\(^{19}\) (referred to as the Hull results) indicate an \(\alpha\) of \(7 \times 10^{-6}K^{-1}\) at 280K compared with the present results of 11.4 \(\times 10^{-6}K^{-1}\) at 280K. Both sets of results fall rapidly at low temperatures (<60K). However whilst the present results tend towards zero expansion at 0K, the Hull results when extrapolated indicate \(\alpha\) going sharply negative below 22K. There is agreement over the temperature range 30K-60K. As mentioned earlier in this chapter, the capacitance dilatometer had been previously used to investigate
relative change in the thermal expansion coefficient at low temperatures, (e.g. the production and investigation of anomalies below 10K), where the measurement of absolute expansion was not of crucial importance. It was therefore decided that the expansion of a pure silica sample should be measured to investigate any inaccuracies, or systematic errors, which may have been introduced into the experimental system by way of reference material.

7.3.4 Vitreous Silica Results

The thermal expansion of silica over the temperature range 4.2K - 300K is shown in FIG 7.18. For comparison the results obtained by other authors19,23,24 for the expansion of silica have been included. G.K. White's results are shown for the temperature range 3-30K.

White23 found variations of between 10 - 15% for the coefficient of expansion at low temperatures by examining various silica samples with different fictive temperatures, (the temperature at which the glass was quenched).

The present results compare well below 60K (FIG 7.19). At 60K the present results deviate sharply from that of other authors. To clarify this situation the results for the lead silicate sample have also been included on FIG 7.18 and FIG 7.19. In both cases the significant deviation occurs at about 60K.

It was considered possible that:

(i) A systematic error was present in the experiment.

(ii) The thermal expansion data for copper used to compute the thermal expansion of other samples was a contributing factor to the discrepancy at higher temperatures.

(iii) The measurement of α for silica above 30K was not suited to the dilatometry technique. Where the sample expansion is small, compared to that of copper, and where consequently corrections to data are significant.

To determine the thermal expansion coefficient of silica using a copper capacitance cell consider FIG 7.20.
$g$ = gap between the electrodes
$l_c$ = length of the copper cell
$l_s$ = length of the sample
$l_e$ = length of copper electrode
$\alpha$ = thermal expansion of silica

\[ g = l_c - (l_s + l_e) \] at some temperature $T$

Now
\[ T = T' + \Delta T \quad \text{where } T' = \text{room temperature} \]
\[ l = l' + \Delta l \quad \text{where } l' = \text{length at room temperature} \]

\[ \alpha = \frac{\Delta l}{\Delta T} \times \frac{1}{l} \]

\[ g = (l_c' + \Delta l_c) - (l_s' + \Delta l_s) - (l_e' + \Delta l_e) \]

\[ g = l_c'(1 + \alpha_c \Delta T) - l_s'(1 + \alpha_s \Delta T) - l_e'(1 + \alpha_c \Delta T) \]

\[ g = (l_c' - l_s' - l_e') + \Delta T(\alpha_c(1 + \alpha_c \Delta T)) - \alpha_s l_s' \]

\[ g = g' + \Delta T(\alpha_c(1 + \alpha_c \Delta T)) - \alpha_s l_s' \]

Now $g - g' = \Delta g$:

\[ \frac{\Delta g}{\Delta T} = \alpha_c(1 + \alpha_c \Delta T) - \alpha_s l_s' \]

Therefore:

\[ \alpha_s = \frac{(l_c' - l_e')}{l_s'} \cdot \alpha_c - \frac{\Delta g}{\Delta T} \cdot \frac{1}{l_s'} \]

\[ \alpha_s = K \cdot \alpha_c - \frac{G}{l_s'} \quad (1) \]

$K$ = constant = ratio of (copper length - copper electrode length) to sample length at $T'$ (room temperature)

$G$ = gradient of graph of electrode gap against temperature

Assuming the experimental results are correct, and the sample lengths have been measured accurately, equation (1) implies the discrepancy lies in the values used for $\alpha_c$, the thermal expansion coefficient for copper.
The lead silicate experiment was carried out in April 1985 using the original cryostat with glass dewars. The silica experiment was performed in July 1988 using the new cryostat commissioned from Oxford Instruments. Two separate copper cells were built out of the same material to accommodate the two samples of different lengths, both cells utilised the same copper electrode system. Both sets of results show this divergence at 60K. These factors lead the author to believe that the error may lie within the thermal expansion data used for copper. Earlier experimental work, carried out at low temperature, where the expansion of copper is small, would therefore not have been significantly disturbed by such effects, and hence would not have been observed at that time.

The data used for the linear thermal expansion of copper was that of Kroegar and Swenson\(^2\). They used an absolute dilatometer based, on a three-terminal parallel plate capacitor design, to make their measurements. Many authors have measured the thermal expansion of copper and the results are well documented. \(^{23,24,25}\)

Perlera and Graham\(^26\) used an optical lever dilatometer to measure the thermal expansion of five copper samples in order to ascertain if specimen dependent effects existed in the expansion of nominally pure copper. Their results showed that no two copper samples were in agreement to within several times the scatter of the measurements (±2%). The largest change (20%) was caused by the annealing of very pure copper in an oxidising atmosphere. The results were also compared with an N.B.S. copper standard (measured by the University of Iowa Group). A maximum disagreement of 10% near T = 6K was observed. Perlera and Graham suggest that an impurity dislocation interaction may be the cause of the anomalous expansion. Their results indicate that an additional contribution to the thermal expansion arises in the interacting of the impurities and some other defect, which depends somewhat on grain size. The most likely imperfections to have strong interaction with the impurities as well as macroscopic features are the dislocations. For the sort of specimens studied in their work the dislocation density was thought to be of the order of \(10^{-7} - 10^{-8}\) lines/cm, or about \(10^{-7}\) lines/atom. The impurity level is \(10^{-5}/\)atom which is sufficient to load the average line with at least 100 impurity atoms.
McLean et al measured the expansion of copper below 30K using a variable transformer technique. Their results agreed with those of White and Collins to within ±5% at low temperatures, and to within ±2% above 20K.

These anomalies/discrepancies in the copper expansion data, as observed by other workers, lead the author to suggest that the expansion of the copper used to make the capacitance cell, may be somewhat different from the copper expansion data of Kroegar and Swenson. Previous experimental work using the capacitance dilatometer had been carried out at very low temperatures (< 20K), where the expansion of copper is virtually zero, and significantly smaller than the thermal expansion of the sample being measured. Therefore any discrepancies in the copper expansion data would not have had any noticeable effect on the experimental results.

The thermal expansion of pure silica is small and the values well documented. Using the polynomials of Hahn and Kirby (which fit their experimental data) the expansion of silica was calculated for points up to room temperature. These values were then substituted in equation (1). The values of G and K were substituted from our experimental results for silica. G is the gradient of the graph of electrode gap (Δg) against temperature (T), and K is the ratio of (length of copper cell - length of copper electrode) to the specimen length. By carrying out back substitution in equation (1) the thermal expansion of copper was recalculated, and the results have been plotted in FIG 7.21. For comparison, the copper data of Kroegar and Swenson has been plotted on the same graph. It can be seen that for temperatures up to 70K the two sets of copper data agree within experimental error. Beyond this temperature the data diverges. The maximum disagreement being of the order of 8%. If these adjusted copper values indicate the true expansion of the copper cell, it should be possible to use them to recalculate the thermal expansion of the lead silicate sample. This exercise has been carried out and the results plotted in FIG 7.22. For comparison the expansion data from Hull University, for the same lead silicate sample, has also been plotted. It can be seen that close agreement is now obtained between the two sets of data up to a
temperature of 150K. Beyond this temperature the data results diverge, but even at these higher temperatures there is a considerable improvement in the correlation obtained between the two sets of lead silicate data.

For a closer inspection, the thermal expansion of lead silicate up to 120K is shown in FIG 7.23. There is extremely close agreement between the two sets of data down to 30K. Below 30K the Loughborough data follows a gentle curve towards zero expansion at 0K, whilst the Hull data takes a sharper negative plunge. Because of the close agreement above 30K, and the unreliability of the Hull results below 30K, the author believes that the thermal expansion of lead silicate does in fact remain positive even at low temperatures.

7.3.5 Discussion

It had been hoped that the thermal expansion of a second lead silicate sample could have been measured. This would further validate the suggestion that the data discrepancy is due to the expansion data used for copper. The author's main task was the realisation of a distributed control system to measure linear thermal expansion using capacitance dilatometry. This ambitious task limited the time available for the further investigation of the thermal expansion results.

The lead silicate results for this sample show no firm evidence of negative thermal expansion at low temperatures. These results are in agreement with Ford's\textsuperscript{19} for the temperature range 30K - 150K. The low temperature silica results are in agreement with White's\textsuperscript{22}. The negative expansion coefficient of pure silica at low temperatures is believed to be due to the transverse vibration of the oxygen atom in the Si-O-Si chain. Inclusion of lead within the structure acts by restricting this transverse vibration. This results in a marked decrease in the negative expansion coefficient at low temperatures. It would seem logical to conclude that as the percentage of lead within the silica structure is increased so further damping of the transverse mode will occur, with the result that a positive expansion coefficient is observed at low temperatures for the lead silicate sample.
FIG. 7.1
CHANGE OF DILATOMETER ELECTRODE SEPARATION FOR EPOXY RESIN SAMPLE 1,
RANGE: 5K - 290K.
FIG 7.2
CHANGE OF DILATOMETER ELECTRODE SEPARATION FOR EPOXY RESIN SAMPLE I, RANGE: 5K-25K.
FIG 7.3
LINEAR THERMAL EXPANSION OF EPOXY RESIN
SAMPLE I, RANGE: 5K-290K.
FIG 7.4
CHANGE OF DILATOMETER ELECTRODE SEPARATION
FOR EPOXY RESIN SAMPLE II, RANGE: 5K-30K
\[ \Delta \text{GAP} \times 10^5 \text{Å} \]

**FIG 7.5**

CHANGE OF DILATOMETER ELECTRODE SEPARATION FOR EPOXY RESIN SAMPLE II, RANGE: 5K-290K

TEMPERATURE (K)
FIG 7.6
LINEAR THERMAL EXPANSION OF EPOXY RESIN
SAMPLE II, RANGE: 5K - 25K
FIG. 7.7
LINEAR THERMAL EXPANSION OF EPOXY RESIN
SAMPLE II, RANGE: 5 K - 290 K.
(FIG 7-8) STRUCTURE OF THE POLYDIACETYLENE CHAIN

R is the side group in the toluene sulphonate polymer

\[ R : \text{C}_6\text{H}_5-O-\text{SO}_2-\text{O}-\text{CH}_3 \]

(FIG 7-9) EACH UNIT CELL CONTAINS TWO POLYMER CHAINS

IN THE CRYSTALLOGRAPHIC 'b' DIRECTION

(FIG 7-10) CAPACITANCE DILATOMETER CELL (TYPE B)
FIG. 7.11
CHANGE IN THICKNESS OF THE POLYDIACETYLENE CRYSTAL OVER THE PHASE TRANSITION TEMPERATURE.
FIG 7.12
CHANGE IN THICKNESS OF THE
POLYDIACETYLENE CRYSTAL
RANGE: 4.2K-300K

$\Delta \text{GAP} \ (\times 10^3 \text{ Å})$

TEMPERATURE (K)
FIG. 7.13
LINEAR THERMAL EXPANSION OF POLYDIACETYLENE SINGLE CRYSTAL IN THE a* DIRECTION, RANGE: 4.2K-300K
FIG. 7.14
CHANGE IN THICKNESS OF THE POLYDIACETYLENE CRYSTAL INDICATING A SMALL ANOMALY AT 35K.
POTENTIAL ENERGY (E) OF THE SYSTEM AS A FUNCTION OF A GENERALISED COORDINATE X.

(FIG 7.15)
FIG. 7.16
LINEAR THERMAL EXPANSION OF LEAD SILICATE,
RANGE: 4.5 K - 290 K.
FIG. 7.17
LOW TEMPERATURE LINER THERMAL EXPANSION OF LEAD SILICATE
RANGE: 4.5 K - 20 K.
LINEAR THERMAL EXPANSION OF SILICA AND LEAD SILICATE, RANGE: 0.5K–290K

- Loughborough, Lead Silicate Results
- Hull, Lead Silicate Results
- Loughborough, Silica Results
- White, Hahn, Silica Results
(FIG. 7.19) LINEAR THERMAL EXPANSION OF SILICA AND LEAD SILICATE, RANGE: 4.5K-110K

- LOUGHBOROUGH, LEAD SILICATE RESULTS
- HULL, LEAD SILICATE RESULTS
- LOUGHBOROUGH, SILICA RESULTS
- WHITE, HAHN, SILICA RESULTS
(FIG. 7-20) MEASUREMENT OF LINEAR THERMAL EXPANSION
ADJUSTED COPPER VALUES
+ DATA FROM KROEGAR & SWENSON

(FIG 7.21) LINEAR THERMAL EXPANSION COEFFICIENT FOR COPPER.
(FIG. 7.22) LINEAR THERMAL EXPANSION COEFFICIENT FOR LEAD SILICATE

EXP. COEF. ($10^{-6}$K$^{-1}$)

+ LOUGHBOROUGH DATA

○ HULL DATA

TEMPERATURE (K)
(FIG. 7·23) LINEAR THERMAL EXPANSION COEFFICIENT FOR LEAD SILICATE

EXP. COEF. ($\times 10^{-6} K^{-1}$)

- LOUGHBOROUGH DATA
- HULL DATA

TEMPERATURE (K)
CHAPTER EIGHT - Discussion and Further Work

8.1 Advantages and Disadvantages of the Automation System

An advantage of the modular system, which utilises distributed control, is the ease of system expansion. Additional slave controllers can be independently connected on to the serial link without the need for modifications to the existing slave modules. To enable the master controller to communicate with the new slave, it would be necessary to build in additional software routines. However, these modifications would not involve radical changes to the current software, but would be additions to the existing software structure within the master control module (MASCON).

The present system uses the Intel 8751 8-bit microcontroller. However, should a more sophisticated controller be required in expanding the system, the Intel 8796 (eprom version of the 8096) could be used. The Intel 8796 is a powerful 16-bit microcontroller with additional features such as a 10-bit A/D converter, a pulse-width modulated output, high speed I/O subsystem, and a full duplex serial port. The advantage of using the 8796 is that it can operate in the same mode of multiprocessor communication as the 8751 (see 4.2), and communicate via the duplex serial port.

The modular design approach results in system segregation. Consequently if one slave control module should fail, the result may not necessarily be total system failure (see 6.1.4). Because of the modular design it is possible for the other controllers to continue to function. This will be advantageous to the system operator trying to identify where the failure has occurred. If a correct response can still be obtained from the functioning slaves, the problem can be isolated to the faulty slave.

A further advantage to using this modular approach is that should a module fail, it can be programmed to fail in a "safe" way. That is, should an error occur, the module can activate an error handling routine to stop processing, and shut the module down in a controlled manner.
A considerable degree of intelligence has been incorporated into the microcontrol system enabling it to make valid responses to experimental data changes. For example:

- **SENCON** analyses of the rate of change of capacitance with temperature in order to select the sensitivity setting for the system (see 5.3.3).

- **OBCCON** is able to distinguish between valid and invalid readings of off-balance voltage from the A/D converter (see FIG 5.6).

- **MOTCON** returns error information to **MASCON** along with the latest on-balance capacitance reading.

From a practical point of view the large number of I/O lines available on each 8751 make it a good choice for this application of distributed control. There are many logic signals to be received from, and transmitted to, the instrumentation associated with the three-terminal capacitance dilatometer. For example, there is a 50-way output on the back of the capacitance bridge, from which the bridge capacitance setting must be read. Khan developed the prototype automation system using a DAI Personal Computer, and found it necessary to resort to a considerable degree of multiplexing in order to read and write the necessary information.

Considerable problems were encountered in designing the mechanical linkage for the manual capacitance bridge (see 3.2.4). This could be seen as a disadvantage of this approach to automation. However, the alternative was to use an electronic capacitance bridge with inadequate sensitivity.

The creation of a working microelectronic system requires a range of development tools. In this particular case a Series IV development system, a Prom-Programmer, and an In-Circuit Emulator for the 8751 were required to aid in the design and testing of the system. The obvious disadvantage is the high cost of acquiring these tools. This was not a problem in this instance, as these items were donated to the Physics Department by the Intel Corporation.

The design of the microelectronic distributed control system...
required the researcher to have an understanding of several areas of expertise. Hardware and software experience were needed. A knowledge of the interfacing instrumentation was also required for this application. In addition to this it was necessary to have an understanding and appreciation of the physics incorporated in this extremely sensitive three-terminal capacitance technique.

8.2 Discussion of System Expansion

8.2.1 Using the 7 Tesla Superconducting Magnet

Built into the cryostat is a 7 Tesla superconducting magnet (see 3.1.1) with a homogeneity of 1% in a cylinder 5cm x 5cm long. The associated electronics facilitate the sweeping of the magnetic field between 0 and 7 Tesla. Alternatively, the field may be kept constant whilst the temperature of the specimen within the bore of the magnet is varied.

The superconducting magnet was incorporated in the original cryostat design in order that the linear thermal expansion \( \alpha \), of dielectric crystals could be studied under an applied magnetic field. I J Brown\(^2\) has demonstrated the existence of Schottky-type anomalies in the low temperature \( \alpha \) of dielectric crystals, produced by the presence of a small concentration of strongly coupled paramagnetic impurity ions. The application of an applied magnetic field should cause a temperature shift of the position of the \( \alpha \) Schottky anomaly. The information obtained from this experimental work can be used to investigate the energy level structures of these materials.

The system could also be used to study the magnetic susceptibility of materials as a function of applied magnetic field or temperature. This information can be used to determine the ordering of magnetic moments within a material, and to deduce the energy level structure for that particular system. The system could also be adapted to investigate other thermal properties, such as specific heat, within a magnetic field.

The microelectronic control system can be expanded to include a slave 8751 for the automated control of the superconducting magnet.
The controller would also, simultaneously, monitor the level of liquid helium within the cryostat. In this way quenching of the magnet, due to evaporation of coolant below a critical level, can be prevented. If the level should fall below a critical level the controller would respond by switching off the 60A current to the magnet, hence avoiding a potentially dangerous situation. This again emphasises the advantage of a modular design in providing a 'fail-safe' system.

8.2.2 Optical Radiation of Samples within the Cryostat

A series of parallel mylar windows have been incorporated into the cryostat. The windows enable a light beam to strike a specimen within the inner sample chamber. There is a short distance (~ 20 mm) between the bottom of the cryostat, and the top of the concrete block on which the metal cage of the cryostat resides. Thus any beam of light would need to be redirected by means of a precise, mechanical, adjustable mirror system.

The reason for the inclusion of the set of windows was to make possible the optical irradiation of samples whilst in situ within the cryostat. Previous workers (Brown and Brown, 1980) have carried out low temperature measurements of linear thermal expansion $\alpha$, on gamma ($\gamma$) irradiated ruby ($\text{Al}_2\text{O}_3$:Cr) using a three-terminal capacitance dilatometer. Their results for $\alpha$ indicated a positive Schottky-type anomaly at $\approx 3.9K$, that was not present in the data for pure $\text{Al}_2\text{O}_3$. This contribution to $\alpha$ they attributed to the presence of Cr$^{2+}$ (produced by the $\gamma$ radiation) exhibiting a large positive magnetic Grüneisen coefficient as predicted by a dynamic Jahn-Teller model. The amount of Cr$^{2+}$ present in the Al$\text{O}_3$ can be increased by $\gamma$ radiation, or decreased by irradiating with ultra-violet (UV) light at 365nm (Brown and Robach, 1978).

The ruby sample should be cooled to liquid nitrogen temperatures, and then removed from the cryostat in order to $\gamma$ irradiate the sample. In order to prevent self-annealing the sample should remain immersed in liquid nitrogen during irradiation. The ruby sample can then be returned to the cryostat, and further cooled to liquid helium temperatures. The sample can now be irradiated, via the windows, in a controlled manner, using UV light to remove Cr$^{2+}$. 

137
Removal of the sample from the cryostat to carry out optical irradiation, and then re-assembly of the sample, can result in effects due to hysteresis (Brown and Brown\textsuperscript{5}, White\textsuperscript{6}). It is for this reason that care should be taken to keep the sample at liquid nitrogen temperatures during \( \gamma \) irradiation.

Previous workers (Brown and Brown\textsuperscript{3}) specified that their results were near the present limit of their measurement technique (T \( \geq \) 3.5K), and would benefit from the use of a lower temperature system. This would enable them to view the 'tail' of the observed Schottky anomaly going towards absolute zero. The cryogenic system specified within this thesis (see 3.1) has a base temperature of 1.5K, and would therefore be suitable to carry out further work on ruby at the lower temperatures.

### 8.2.3 Incorporation of a He\textsubscript{3}-He\textsubscript{4} Dilution Refrigerator

At present the cryostat is limited to a base temperature of 1.5K. The system can be modified to reach lower temperatures in the milli-kelvin range. This can be achieved by removing the variable temperature insert (see 3.1.1), and replacing it with a He\textsubscript{3}-He\textsubscript{4} dilution refrigerator.

The operation of a dilution refrigerator can be described by analogy to the cooling produced on evaporation of a liquid into the vapour phase. Below 0.87K (\( T_{\text{c}} \) point) a mixture of He\textsubscript{3}-He\textsubscript{4} spontaneously separates into two phases: one concentrated in He\textsubscript{3}, and the other dilute in He\textsubscript{3} (with the He\textsubscript{4} being predominantly in the superfluid state). Because of its lower density He\textsubscript{3} floats on top of He\textsubscript{4}. A concentrated phase of He\textsubscript{3} is analogous to the liquid phase. A dilute phase of He\textsubscript{3} atoms which are dispersed amongst the He\textsubscript{4} is similar to the vapour phase. Cooling is produced when He\textsubscript{3} atoms from the concentrated phase cross the phase boundary into the dilute phase. This is then analogous to cooling by evaporation where molecules absorb the latent heat of vaporization to escape into the vapor phase. By using a vacuum pump the He\textsubscript{3} can be continuously removed from the system, thereby achieving cooling.

Unlike conventional evaporation refrigerators (cryostats) in which the vapour pressure rapidly approaches zero as the temperature is
reduced, the transport of $\text{He}^3$ across the phase boundary will continue because of the 6.4% solubility of $\text{He}^3$ in $\text{He}^4$ even in the limit of absolute zero. However, the heat of quasi-vapourization of $\text{He}^3$ vanishes proportionally to $T^2$, where $T$ is the temperature. As a result the heat removal rate from the $\text{He}^3$-$\text{He}^4$ mixing chamber vanishes as $T^2$. This limits the final base temperature of the dilution refrigerator, (typically $\approx 3\text{mK}$).

The dilution refrigerator provides the means by which temperatures in the milli-kelvin range can be obtained. This temperature range would be particularly useful for the investigation of the physics of heavy fermions.

8.3 Final Conclusions

The automation of a three-terminal capacitance dilatometer has been successfully achieved by using distributed control techniques. The automation consists of an assembly of single-chip microcontrollers connected on a serial link. This research has involved the design of a novel and intricate control system, based on multiprocessor communication between intelligent master and slave controllers. The complexities of the system automation, and the problems encountered in achieving a practical working instrument have been described herein. Further, the possibilities for system expansion have been considered and reviewed.

The three-terminal capacitance dilatometer has been used to measure the linear thermal expansion of several materials.

Preliminary experiments have been carried out on two epoxy resin samples.

From the linear thermal expansion results the phase change occurring at 195K in single crystal polydiacetylene has been clearly observed. Also, this sensitive capacitance technique has been able to detect a small anomaly at 35K. This anomaly may be due to a low lying optical mode.
Lastly, the linear thermal expansion of pure silica and lead silicate has been measured, and the results have been discussed.

The work involved in the design and testing of the automation system limited the time available for further experimental work. Clearly, further work on the linear thermal expansion of lead silicate at different compositions is required, to complement the results obtained here. A more detailed study is necessary, in order to confirm the initial conclusion that there is no firm evidence of negative thermal expansion in the lead silicate sample at low temperatures.
CALIBRATION NUMBER 14321

THE TEMPERATURES WERE OBTAINED FROM RHODIUM-IRON AND PLATINUM RESISTANCE THERMOMETERS WHOSE OWN CALIBRATIONS ARE ACCURATE TO PLUS OR MINUS 0.003K OF EPT-76 UP TO 30K, AND PLUS OR MINUS 0.005K OF THE IPTS-68 ABOVE.

\[ \Delta T = Z_{\text{REF}} - Z \]

WHERE \( Z_{\text{REF}} \) IS OBTAINED FROM THE 1 27 OHM RHODIUM-IRON REFERENCE SCALE 1.5 TO 300K (1983) ISSUED BY CRYOGENIC CALIBRATIONS LTD. \( Z \) HAS BEEN CALCULATED USING TWO FUNCTIONS:

BELOW 27K: \[ Z = \frac{(R - 2.32785)}{25.42389} \]

ABOVE 27K: \[ Z = \frac{(R - 2.37233)}{21.63820} \]

THIS CALIBRATION USED IN CONJUNCTION WITH THE REFERENCE TABLES SHOULD LEAD TO AN OVERALL ACCURACY OF PLUS OR MINUS 0.01K BELOW 27K AND PLUS OR MINUS 0.02K ABOVE.

CHECKED...
### APPENDIX A

**CRYOGENIC CALIBRATIONS LTD**

**PITCHCOTT, Nr. AYLESBURY, BUCKS., HP22 4HT, ENGLAND. TEL. 029-664-269**

**CALIBRATION NUMBER 14321**

**THERMOMETER SPECIFICATION: C.C.LTD. 27 OHM RHODIUM-IRON.**

**NO 466**

<table>
<thead>
<tr>
<th>TEMPERATURE (KELVIN)</th>
<th>RESISTANCE (OHMS)</th>
<th>DELTA ZX10<strong>6</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.501</td>
<td>1.92343</td>
<td>167</td>
</tr>
<tr>
<td>1.615</td>
<td>1.94267</td>
<td>151</td>
</tr>
<tr>
<td>2.010</td>
<td>2.00733</td>
<td>115</td>
</tr>
<tr>
<td>2.789</td>
<td>2.12819</td>
<td>63</td>
</tr>
<tr>
<td>3.217</td>
<td>2.19074</td>
<td>46</td>
</tr>
<tr>
<td>4.016</td>
<td>2.30095</td>
<td>5</td>
</tr>
<tr>
<td>4.226</td>
<td>2.32842</td>
<td>-0</td>
</tr>
<tr>
<td>5.395</td>
<td>2.47121</td>
<td>-44</td>
</tr>
<tr>
<td>7.030</td>
<td>2.64504</td>
<td>-73</td>
</tr>
<tr>
<td>8.020</td>
<td>2.73832</td>
<td>-82</td>
</tr>
<tr>
<td>10.026</td>
<td>2.90506</td>
<td>-95</td>
</tr>
<tr>
<td>13.022</td>
<td>3.11159</td>
<td>-95</td>
</tr>
<tr>
<td>16.034</td>
<td>3.28265</td>
<td>-83</td>
</tr>
<tr>
<td>20.026</td>
<td>3.47187</td>
<td>-61</td>
</tr>
<tr>
<td>25.129</td>
<td>3.67869</td>
<td>-27</td>
</tr>
<tr>
<td>27.447</td>
<td>3.76701</td>
<td>0</td>
</tr>
<tr>
<td>30.235</td>
<td>3.87384</td>
<td>-102</td>
</tr>
<tr>
<td>34.925</td>
<td>4.06295</td>
<td>-269</td>
</tr>
<tr>
<td>40.175</td>
<td>4.30085</td>
<td>-451</td>
</tr>
<tr>
<td>50.238</td>
<td>4.86425</td>
<td>-789</td>
</tr>
<tr>
<td>60.293</td>
<td>5.57658</td>
<td>-1090</td>
</tr>
<tr>
<td>70.299</td>
<td>6.41268</td>
<td>-1338</td>
</tr>
<tr>
<td>80.493</td>
<td>7.36130</td>
<td>-1512</td>
</tr>
<tr>
<td>90.209</td>
<td>8.32650</td>
<td>-1615</td>
</tr>
<tr>
<td>100.311</td>
<td>9.36809</td>
<td>-1676</td>
</tr>
<tr>
<td>125.957</td>
<td>12.07352</td>
<td>-1656</td>
</tr>
<tr>
<td>150.451</td>
<td>14.64594</td>
<td>-1509</td>
</tr>
<tr>
<td>175.437</td>
<td>17.22402</td>
<td>-1273</td>
</tr>
<tr>
<td>200.597</td>
<td>19.77741</td>
<td>-997</td>
</tr>
<tr>
<td>225.383</td>
<td>22.26232</td>
<td>-676</td>
</tr>
<tr>
<td>250.487</td>
<td>24.76107</td>
<td>-333</td>
</tr>
<tr>
<td>272.990</td>
<td>26.99375</td>
<td>-5</td>
</tr>
<tr>
<td>273.236</td>
<td>27.01814</td>
<td>0</td>
</tr>
<tr>
<td>300.496</td>
<td>29.72270</td>
<td>426</td>
</tr>
</tbody>
</table>
### Thermometer Specification

**CryoCal Inc., Type CR1000, No. 4034**

<table>
<thead>
<tr>
<th>Temperature (Kelvin)</th>
<th>Resistance (ohms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.497</td>
<td>13,950</td>
</tr>
<tr>
<td>1.588</td>
<td>11,762</td>
</tr>
<tr>
<td>1.799</td>
<td>8,314</td>
</tr>
<tr>
<td>2.022</td>
<td>6,107</td>
</tr>
<tr>
<td>2.201</td>
<td>4,935</td>
</tr>
<tr>
<td>2.375</td>
<td>4,106.1</td>
</tr>
<tr>
<td>2.571</td>
<td>3,415.3</td>
</tr>
<tr>
<td>2.815</td>
<td>2,790.3</td>
</tr>
<tr>
<td>3.037</td>
<td>2,369.7</td>
</tr>
<tr>
<td>3.241</td>
<td>2,068.1</td>
</tr>
<tr>
<td>3.413</td>
<td>1,858.8</td>
</tr>
<tr>
<td>3.640</td>
<td>1,631.1</td>
</tr>
<tr>
<td>3.810</td>
<td>1,489.3</td>
</tr>
<tr>
<td>3.992</td>
<td>1,358.0</td>
</tr>
<tr>
<td>4.216</td>
<td>1,219.7</td>
</tr>
<tr>
<td>4.417</td>
<td>1,114.5</td>
</tr>
<tr>
<td>4.590</td>
<td>1,033.7</td>
</tr>
<tr>
<td>5.026</td>
<td>866.5</td>
</tr>
<tr>
<td>5.401</td>
<td>752.5</td>
</tr>
<tr>
<td>5.774</td>
<td>659.0</td>
</tr>
<tr>
<td>6.214</td>
<td>567.44</td>
</tr>
<tr>
<td>6.669</td>
<td>489.82</td>
</tr>
<tr>
<td>7.054</td>
<td>435.11</td>
</tr>
<tr>
<td>7.516</td>
<td>379.76</td>
</tr>
<tr>
<td>8.032</td>
<td>328.43</td>
</tr>
<tr>
<td>8.544</td>
<td>286.39</td>
</tr>
<tr>
<td>9.054</td>
<td>251.57</td>
</tr>
<tr>
<td>9.501</td>
<td>225.72</td>
</tr>
<tr>
<td>10.004</td>
<td>200.99</td>
</tr>
<tr>
<td>10.522</td>
<td>179.47</td>
</tr>
<tr>
<td>11.013</td>
<td>162.09</td>
</tr>
<tr>
<td>12.019</td>
<td>133.67</td>
</tr>
<tr>
<td>13.022</td>
<td>112.52</td>
</tr>
<tr>
<td>14.019</td>
<td>96.419</td>
</tr>
<tr>
<td>14.998</td>
<td>84.021</td>
</tr>
</tbody>
</table>
### Temperature (Kelvin) vs. Resistance (ohms)

<table>
<thead>
<tr>
<th>Temperature (Kelvin)</th>
<th>Resistance (ohms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.050</td>
<td>73.457</td>
</tr>
<tr>
<td>17.043</td>
<td>65.391</td>
</tr>
<tr>
<td>18.075</td>
<td>58.498</td>
</tr>
<tr>
<td>18.992</td>
<td>53.357</td>
</tr>
<tr>
<td>20.095</td>
<td>48.125</td>
</tr>
<tr>
<td>22.452</td>
<td>39.444</td>
</tr>
<tr>
<td>25.247</td>
<td>32.119</td>
</tr>
<tr>
<td>27.371</td>
<td>27.952</td>
</tr>
<tr>
<td>30.067</td>
<td>23.8326</td>
</tr>
<tr>
<td>32.122</td>
<td>21.3368</td>
</tr>
<tr>
<td>35.458</td>
<td>18.1374</td>
</tr>
<tr>
<td>37.804</td>
<td>16.3606</td>
</tr>
<tr>
<td>40.223</td>
<td>14.8339</td>
</tr>
<tr>
<td>45.285</td>
<td>12.3733</td>
</tr>
<tr>
<td>50.359</td>
<td>10.5952</td>
</tr>
<tr>
<td>55.010</td>
<td>9.3686</td>
</tr>
<tr>
<td>60.429</td>
<td>8.2746</td>
</tr>
<tr>
<td>65.235</td>
<td>7.5204</td>
</tr>
<tr>
<td>70.495</td>
<td>6.8639</td>
</tr>
<tr>
<td>75.409</td>
<td>6.3708</td>
</tr>
<tr>
<td>80.350</td>
<td>5.9645</td>
</tr>
<tr>
<td>85.678</td>
<td>5.6046</td>
</tr>
<tr>
<td>90.240</td>
<td>5.3481</td>
</tr>
<tr>
<td>95.487</td>
<td>5.1002</td>
</tr>
<tr>
<td>100.188</td>
<td>4.9136</td>
</tr>
<tr>
<td>125.681</td>
<td>4.2864</td>
</tr>
</tbody>
</table>

The temperatures were obtained from rhodium-iron and platinum resistance thermometers whose own calibrations are accurate to ± 0.003K of the IPTS-76 up to 30K, and ± 0.005K of the IPTS-68 above.

This calibration should be accurate to ± 0.006K up to 30K, and ± 0.008K or ± 0.0005 ohms above.

Checked.
DIRECT THERMAL EXPANSION MEASUREMENTS OF A POLYDIACETYLENE SINGLE-
CRYSTAL PERPENDICULAR TO THE CARBON CHAIN DIRECTION

K J Britton and M A Brown
Physics Department, University of Technology, Loughborough, U K.
and
M N Wybourne
G E C Research Laboratories, Hirst Research Centre, Wembley, U.K.

ABSTRACT

The thermal expansion of polydiacetylene-bis (toluene sulphonate) has been measured in the temperature range 4.2 to 300K using a prototype precision three-terminal capacitance dilatometer. The change in sample dimension in a direction perpendicular to the carbon back-bone was measured to better than 1 part in $10^5$ in an attempt to use thermal expansion to observe any material phase transitions and any low-energy optical mode structure. The expansion coefficient was $4.3 \times 10^5 \, \text{K}^{-1}$ at 280K which is consistent with earlier measurements using X-ray techniques in a similar crystallographic direction. A distinct structure was seen in the data centred at 195K, which corresponds to a known phase change involving torsional rotation of the side groups, and a low temperature feature was observed at 35K, which may be caused by low-lying optical phonon modes.

1. INTRODUCTION

In this paper we report an initial experiment using a prototype three-terminal capacitance dilatometer to study the thermal expansion of polydiacetylene bis (p-toluene sulphonate) of 2,4 hexadiyne-1, 6 dial (PDA-TS) over the temperature range 4.2-300K. Large single crystals of polydiacetylenes with a high degree of perfection can be obtained by the solid state polymerisation of the monomer material\textsuperscript{1)}, Figure 1. At high temperatures each unit cell contains two polymer chains extending in the crystallographic b direction, Figure 2, while at 195K PDA-TS undergoes a second order phase transition below which the space group remains the same but the unit cell is doubled in size.
and contains two species of polymer chain. This structural phase transition is caused by an 8 degree rotation of side groups oriented along the (102) planes of the high temperature structure with adjacent rows having opposite directions of rotation²).

2. EXPERIMENTAL

Three identical pieces of PDA-TS were placed symmetrically between two electrode plates as 'spacers' (Figure 3). The specimen length changes were found using electrodes 1 and 2 in a three-terminal capacitance system³) and the results are shown in Figure 4.

The specimen length at 4.2K = 0.573nm and αₐₑ at 280K = 4.3 x 10⁻⁵K⁻¹.
3. CONCLUSIONS

The thermal expansion was measured in the \(a^*\) direction and should comprise a linear combination of the temperature dependence for the lattice parameters \(a\) and \(c\). The value for \(\alpha_{a^*}\) obtained at 280K was \(4.3 \times 10^{-5} \text{K}^{-1}\), which is consistent with other values for \(\alpha_a\) and \(\alpha_c\) published in the literature\(^4\)). However, between \(~140\text{K}\) and \(195\text{K}\) there is an increase in crystal thickness which must originate from some other mechanism. At 195K the phase change occurs involving torsional rotation of the polymer sidegroups and, as a result of different nucleation centres, domains are formed; the domain walls being the boundaries between regions where the sidegroup rotations are in the opposite sense. At these boundaries there will be a re-orientation of the structure akin to a martensitic transformation and, therefore, morphological changes will occur at the surface\(^6\)). These are expected to manifest themselves as ridges which would give rise to the apparent increase in thermal expansion. From the data we estimate the height of these ridges \(~250\text{nm}\). Finally, as shown in the insert to Figure 4, a small feature occurs at \(~35\text{K}\), which may be caused by low-lying optical modes freezing out, although we have not been able to investigate this further. Further work is clearly necessary on this system and a new cryostat is under construction which will enable us to study this feature in detail.

4. REFERENCES

APPENDIX C

\[ p = -\frac{d}{dV} \left[ U - T \frac{1}{T} \int_0^T \left[ \frac{dU}{dT} \right]_V \right] \tag{4} \]

\[ U = U_{eq} + \frac{1}{\Sigma} 3N \sum_{i=1}^{\infty} \hbar \omega_i + 3N \sum_{i=1}^{\infty} \hbar \omega_i \cdot \left( e^{\hbar \omega_i / kT} - 1 \right)^{-1} \]

Substituting for \( U \) in (4) and differentiating with respect to \( T \):

\[ \left[ \frac{dp}{dT} \right]_V = -\frac{d^2 U}{dVdT} + \frac{T}{T} \int_0^T \left[ \left( \frac{dU}{dT} \right)_V \cdot dT + \frac{d^2 U}{dVdT} \right] \]

\[ \left[ \frac{dp}{dT} \right] = kT \left[ \frac{1}{kT} \int_0^T \frac{1}{\hbar \omega_i} \left[ \frac{d}{dV} \left( \frac{\hbar \omega_i}{kT} - 1 \right)^{-1} \right] \right] \cdot dT \]

Let \( x = \hbar \omega \quad y = kT \)

\[ \left[ \frac{dp}{dT} \right] = kT \int_0^{rac{1}{y}} \frac{1}{y} \int_1^{\frac{1}{y}} \left[ \frac{d}{dV} \left( \frac{e^{x / y} - 1}{y} \right)^{-1} \right] \cdot dy \]

Using the result (*) below:

\[ -k \cdot 3N \sum_{i=1}^{\infty} \frac{d \hbar \omega_i}{dV} \cdot \frac{kT}{0} \int_0^{rac{1}{y}} \frac{d}{dy} \left( \frac{e^{x / y} - 1}{y} \right)^{-1} \cdot dy \]

\[ -k \cdot 3N \sum_{i=1}^{\infty} \frac{d \hbar \omega_i}{dV} \cdot \frac{d}{dy} \left( \frac{e^{x / y} - 1}{y} \right)^{-1} \]

\[ -3N \sum_{i=1}^{\infty} \frac{d \hbar \omega_i}{dV} \cdot \frac{d}{dT} \left( \frac{e^{x / y} / kT - 1}{y} \right)^{-1} \]

\[ \frac{d}{dx} (f(y/x)) = -\frac{y}{x} \cdot \frac{d}{dy} (f(y/x)) \]

\[ \frac{d}{dx} \left( \frac{x \cdot d(f(y/x))}{y} \right) = \frac{1}{y} \cdot \frac{d}{dy} (f(y/x)) + \frac{x}{y} \cdot \frac{d}{dy} (f(y/x)) + \frac{-x}{y} \cdot \frac{d}{dy} (f(y/x)) \]

\[ \frac{d}{dx} \left( \frac{x \cdot df(y/x)}{dy} \right) = -\frac{d^2 f(y/x)}{dy^2} \quad (*) \]
APPENDIX D

Operation of the Capacitance Bridge

A complete specification of the '1616 Precision Capacitance Bridge', and details of the theory, installation and operation can be found in the Instruction Manual, (General Radio Company 1971).

The 1616 capacitance bridge is designed for the precise measurement of capacitors and capacitance standards. The system measures either 3-terminal or 2-terminal capacitors. The transformer ratio arm circuitry of the bridge assures that 3-terminal measurements can be made accurately, even in the presence of large capacitance to ground. A wide range of capacitance can be measured, extending from the resolution limit of 0.1aF ($10^{-7}$pF) to a maximum of 10μF, with internal standards, or further with external standards.

The ratio arms of the bridge are transformer windings tapped on the standard side in decimal steps (-1, 0, 1, 2, ... 9, X), and on the unknown side in decade steps (x100, x10, x1). Separate fixed capacitance standards are used, with values ranging in decade steps from 1aF to 100nF. This combination of internal standards and transformer ratios makes possible the wide measurement range of 1 to $10^{14}$. The advantage of transformer ratio arms in a bridge are that accuracies within a few parts per million are possible over a wide range of integral values, even for ratios as high as 1000 to 1, and that these ratios are almost unaffected by age, temperature and voltage.

There are different ways of balancing a simple transformer ratio capacitance bridge. The circuitry of the 1616 bridge is based on the method whereby a single decade divider is used in combination with multiple fixed capacitors. Consider a 100 turn secondary transformer winding, tapped every 10 turns to provide 10V increments. If a 100pF capacitor is then connected to the 70V tap and a 10pF capacitor to the 20V tap the resulting detector current balances that of a 72pF "unknown" capacitance connected to 100V. The bridge can be given 6
figure resolution, for example, through the use of 6 fixed capacitors in decade steps from 100pF to 0.001pF, each of which can be connected to any one of the taps on the transformer. The bridge ratio can also be altered by the use of taps on the unknown side of the transformer to select the voltage applied to the capacitance being measured.

It is economically reasonable to construct the relatively few fixed capacitance standards to have the necessary stability and accuracy for such a bridge; that is one that will measure with 0.01% accuracy over 6 decades of capacitance and 3 decades of frequency. At low frequencies a limit is imposed on sensitivity by the maximum voltage obtainable from the transformer, since for a given core, the voltage at saturation is proportional to the frequency. At high frequencies there is a decrease in accuracy resulting from the decrease in core permeability with frequency, from the increased loading of the transformer by its self-capacitance as well as the bridge capacitance and of course, from the usual residual capacitances and inductances in the bridge wiring components.

The circuitry of the 1616 bridge comprises 12 internal capacitance standards each of which is calibrated at a cardinal value, and each is switched to an appropriate tap on the ratio transformer as the bridge is brought into balance. The 12 standards cover the capacitance range from 100nF to an effective 1aF, one for each multiple of 10. The "high" side of each standard capacitor is connected to the front-panel-selected tap on the standards side of the transformer secondary. The "low" side of each standard is permanently connected to the same summing point which drives the detector. However, the 2 largest capacitance standards can be disconnected from the summing point by dropping the C MAX lever switch on the front of the bridge. Thus, stray capacitance shunting the detector can be reduced during measurement of extremely small values of increments of the unknown capacitance.

Each of the first 6 CAPACITANCE lever switches indicates values in tenths of the value of its associated standard. Thus, the lever with a -1, 0, 1, 2, ... 9, XpF readout actually switches the 10pF standard capacitor. The maximum current through each of these standard capacitors (when the corresponding readout is X, i.e. ten) obtains
when its HIGH side is connected to the transformer secondary at the 200th turn, 180° out of phase from the unknown capacitor connection. The intermediate taps are 20 turns apart. The zero position is a ground connection. The -1 position is a connection to a 20-turn tap on the "unknown" side of the transformer secondary.

In order to provide greater stability in the set of capacitance standards it is preferable to use a moderate valued capacitor connected to fewer turns than an extremely small valued one connected to the usual number of turns. Therefore a set of taps is brought out from the ratio transformer at intervals of 2 turns (to a maximum of 20 turns). Each capacitor that connects to this set of taps is 10 times as large as it would have to be if it were connected to the usual (20-turn-per-step) taps. The last 6 "C" lever switches span 2 turns per step. The 6 associated standard capacitors are each 10 times as large as their effective values.

Therefore, each of the last 6 capacitance lever switches indicates values in hundredths of the value of its associated standard. Thus the lever with a -1, 0, 1, 2, ... 9, XaF readout actually switches the 100aF standard capacitor. For these, of course, the -1 position is a connection to a 2-turn tap on the "unknown" side of the transformer.

Physically, the transformer is manufactured with a separate winding of 22 turns, tapped every 2 turns for the small C standards. A set of 20 windings of 20 turns each serves the large C standards and the "unknown", by a series combination that acts as a centre-grounded 400-turn secondary. Except for the primary (200 turns) all these windings are very intimately coupled by multifilar construction.

Each C standard is calibrated to the desired value by means of a trimmer. All 12 are located behind the locked door on the front panel, where they are labeled with the effective values of the corresponding standards. The set of 12 capacitance standards can be calibrated quickly and accurately by a series of comparison balances starting with a single external standard capacitor of almost any size within the range of the bridge. Since 6-figure resolution of the bridge permits comparison with a precision of 1ppm down to 0.1pF, the
accuracy of the calibration is usually determined by the accuracy of the standard.

Only one external standard is required, most conveniently a 3-terminal 100pF standard. Using a test frequency of 1KHz, the accurate, internal 0.1 transformer ratio can be used to ensure accurate decade ratios of the internal standards. The -1 position of any capacitance balance switch connects the corresponding internal capacitor to a 20-turn tap on the "unknown" side of the ratio transformer. This capacitor can be compared with the next lower decade capacitor, which is connected to the 200-turn winding on the standard side when the corresponding lever is set to the X position. Any adjustments required can then be made with one of the trimmers accessible beneath a hinged cover on the front panel. Such checks or recalibrations of the bridge need not be made often, the capacitors are constructed to be so stable that after calibration they may be expected to change less than 10ppm per year in normal use.
Chapter One

10. Tolkachev A M, Alexandrovski I, Kuchnev V I, Cryogenics, 15, p.547
15. Anderson D A, (1986), Personal communication to Brown M A


Chapter Two


Chapter Three


Chapter Four


Chapter Five

1. Intersil Data Sheet: 'ICL7109 12 Bit Binary A/D Converter for Microprocessor Interfaces


Chapter Six


Chapter Seven


8. Batchelder D N, (1976), J. Polymer Science, 14, pp.1235-40,


Chapter Eight


