COMPUTER BASED TEMPERATURE REGULATION SYSTEM

BY

ALONGE BAMIDELE FRANK

REG. NO.: 2000/9801EE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING,

SCHOOL OF ENGINEERING AND ENGINEERING TECHNOLOGY, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA.

SEPTEMBER 2006

COMPUTER BASED TEMPERATURE REGULATION SYSTEM

BY

ALONGE B FRANK

REG. NO.: 2000/ 9801EE

A THESIS SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT FOR THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF BACHELOR OF ENGINEERING (B.ENG) IN ELECTRICAL AND COMPUTER ENGINEERING OF THE FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA.

DECLARATION

I, ALONGE B FRANK declare that this work was done by me and has never been presented elsewhere for the award of a degree. I also hereby relinquish the copyright to the Federal University of Technology, Minna, Niger State

ALONGE B FRANK
(Student Name)

(Signature and Date)

ENGR S N RUMALA
(Name of Project Supervisor)

(Signature and Date)

ENGR. M.D. ABDULLAHI
(Head of Department)

Signature and Date

(External Examiner)

Signature and Date.

ACKNOWLEDGEMENT

I want to thank the Almighty God for giving me the breath and strength to carry out the project from the beginning to its completion, also for His guidance and for giving me the patience, endurance and keeping me in good health through out my first degree in life.

The successful completion of this work was aided and guided by my supervisor ENGR. S.N. RUMALA.I therefore seize this opportunity to express my gratitude to him for his kindness in seeing to the success of this project

Further thanks to Mr. ROBBY EDEH of Electrical Electronic Engineering Department NDA (Nigerian Defense Academy) and Mr. ADEBOYE O.A.C of the Electrical and Computer Engineering Department FUT Minna for the useful criticism of the design and their contribution to the construction of the

Appreciation to my uncle Mr D.O Alonge who greatly supported me financially through out my B.Eng degree program

My profound thanks also go to Mrs Omokore MD Unilever who was there for me when the going was tough My Eldest Bother Engr. Olabisi Alonge who tailored my mentality towards self – reliance and success

I further want to thank Engr and Mrs Atoba for their helping hands in my life

My sister and her husband Mr and Mrs Elizabeth Adeshina who took over motherhood since I lost my mum

Kudos to my geez and colleagues Olakunle Tanimonure, Owolabi Babawande, Olumeyan Mikel, Ohakanu Stephen and my baby Rachael M Orok Who offered useful pieces of advice during the design and construction of the project

ABSTRACT

The project design is a COMPUTER BASED TEMPERATURE REGULATION YSTEM. This project is designed in order to provide an automated way of controlling the mperature of a given environment.

The purpose of this project is realizable via temperature sensing, signal conditioning and laying. The temperature sensor senses the current temperature of the said environment, sends it the ADC (Analogue to Digital Converter) the digitized signal is then sent via a serial interface to e computer which through the help of a programmed input (software), does the comparison and ntrol by sending a feedback signal to either a cooling element or a heating element thereby inging about the regulation of the temperature of the said environment.

A great advantage of this project is that it provides automation in temperature control d it is reliable if there can be a constant power supply to the system. It is also economical in nstruction.

TABLE OF CONTENTS

CHAPTER ONE
INTRODUCTION

1.0 MOTIVATION:

The use of computers in our day to day lives, either in business or pleasure, can not be overemphasized. One such use of computer is as a controller. The versatility of computers has enabled us to develop them to carry out many control tasks such as quality control in industries to as far as controlling space flight vessels to outer space. This has made the work of human beings so much easier in many of their tasks.

It must not be new to most people by now that we use heating systems of which a method of automatic monitoring (i.e its temperature level) is desired. Most times, people control devices such as electric kettle, a ring boiler, a furnace in a lab or other heating equipments manually. This form of control may involve observing the state of the temperature level of the material. At the same time individuals in their homes use computers for task such as data processing, accessing information, games e t.c little do they know that they could solve their problems of having to play a supervisory role over their heating equipment by using it to take over as a controller. This application requires some form of intricate linking of both devices i e the heating device and the computer. through what may be known as 'interface' circuit as they are clearly two different systems running on clearly different forms of energy to say the least. This linkage or interface allows for communication between the two systems.

1.1 OBJECTIVES

The objectives of this project are:

1. To study the processes involved in using the computer as a temperature controller
2. Show that the task is not as complex as it may seem but a simple design that can be put to use even in the laboratory by doing quite simple constructions.
3. Design and construct such a circuit that can demonstrate how such process of control using computers can be carried out.
4. To identify more ways in which computers can be used for more of such tasks in more sophisticaied ways by simple constructions.

1.2 LIMITATIONS

For reasons such as the bulk of work, time and cost, to mention but a few, the project is broken down into three parts, two of which are simulated using a power supply and indicator circuit.

1.3 THESIS OUTLINE:

This thesis sets out to present an orderly account of work done. It consists of five chapters

Chapter 1 gives an insight into the project from an introductory point of view and presents the stated objectives of the project.

Chapter 2 gives a historical review of the project and also expatiates on some of the fundamentals which are applied in carrying out the task.

Chapter 3 explains the theory of the elements that comprise the computer based temperature control system and shows how they function

Chapter 4 shows all the steps carried out in setting up the circuit elements and reports on the construction and testing of the circuit.

Chapter 5 reveals the level of accomplishment of the task and seeks to suggest further work that could improve on the systems characteristics and output.

CHAPTER TWO

LITERATURE REVIEW

2.1 A BRIEF HISTORY OF TEMPERATURE

Temperature is by far the most measured parameter. It impacts the physical, chemical and biological world in numerous ways. Yet, a full appreciation of the complexities of temperature and its measurement has been relatively slow to develop

Intuitively, people have known about temperature for a long time: fire is hot and snow is cold. Greater knowledge was gained as man attempted to work with metals through the bronze and iron ages. Some of the technological processes required a degree of control over temperature, but to control temperature you need to be able to measure what you are controlling

2.2 EARLY TEMPERATURE MEASUREMENT

Until about 260 years ago temperature measurement was very subjective. For hot metals the colour of the glow was a good indicator. For intermediate temperatures, the impact on various materials could be determined. For example does the temperature melt sulphur, lead or wax, or boil water?

In other words a number of fixed points could be defined, but there was no scale or any way to measure the temperature between these points. It is, however possible that there is a gap in the recorded history of technology in this regard as it is difficult to believe that the Egyptians, Assyrians, Greeks, Romans or Chinese did not measure temperatures in some way.

Galileo invented the first documented thermometer in about 1592. It was an air thermometer consisting of a glass bulb with a long tube attached. The tube was dipped into a cooled liquid and the bulb was warmed, expanding the air inside. As the air continued to expand, some of it escaped. When the heat was removed, the remaining air contracted causing the liquid to rise in the tube and indicating a change in temperature. This type of thermometer is sensitive, but is affected by changes in atmospheric pressure.

2.3 THE EIGHTEENTH CENTURY: CELSIUS AND FAHRENHEIT

By the early 18th century, as many as 35 different temperature scales had been devised. In 1714, Daniel Gabriel Fahrenheit invented both the mercury and the alcohol thermometer. Fahrenheit's mercury thermometer consists of a capillary tube which after being filled with mercury is heated to expand the mercury and expel the air from the tube. The tube is then sealed, leaving the mercury free to expand and contract with temperature changes. Although the mercury thermometer is not as sensitive as the air thermometer, by being sealed it is not affected by the atmospheric pressure. Mercury freezes at -39° Celsius, so it cannot be used to measure temperature below this point. Alcohol, on the other hand, freezes at -113° Celsius, allowing much lower temperatures to be measured.

At the time, thermometers were calibrated between the freezing point of salted water and the human body temperature. (Salt added to crushed wet ice produced the lowest artificially created temperatures at the time). The common Flemish thermometers of the day divided this range into twelve points. Fahrenheit further subdivided this range into ninety-six points, giving his thermometers more resolution and a temperature scale very close to today's Fahrenheit scale. (In fact there appeared to have been between 15

and 20 different temperature scales at this time, determined by nationality and application.)

Later in the 18th century, Anders Celsius realised that it would be advantageous to use more common calibration references and to divide the scale into 100 increments instead of 96. He chose to use one hundred degrees as the freezing point and zero degrees as the boiling point of water. Sensibly the scale was later reversed and the Centigrade scale was born.

2.4 THE NINETEENTH CENTURY: A PRODUCTIVE ERA

The early 1800's were very productive in the area of temperature measurement and understanding. William Thomson (later Lord Kelvin) postulated the existence of an absolute zero. Sir William Herschel, discovered that when sunlight was spread into a colour swath using a prism, he could detect an increase in temperature when moving a blackened thermometer across the spectrum of colours. Herschel found that the heating effect increased toward and beyond the red in the region we now call 'infrared'. He measured radiation effects from fires, candles and stoves, and deduced the similarity of light and radiant heat. However it was not until well into the following century that this knowledge was exploited to measure temperature.

In 1821 T J Seebeck discovered that a current could be produced by unequally heating two junctions of two dissimilar metals, the thermocouple effect. Seebeck assigned constants to each type of metal and used these constants to compute total amount of current flowing. Also in 1821, Sir Humphrey Davy discovered that all metals have a positive temperature coefficient of resistance and that platinum could be used as an excellent temperature detector (RTD). These two discoveries marked the beginning of serious electrical sensors. Gradually the scientific community learnt how to measure

temperature with greater precision. For example it was realised by Thomas Stevenson (civil engineer and father of Robert Louis Stevenson) that air temperature measurement needed to occur in a space shielded from the sun's radiation and rain. For this purpose he developed what is now known as the Stevenson Screen. It is still in wide use

The late 19th century saw the introduction of bimetallic temperature sensor. These thermometers contain no liquid but operate on the principle of unequal expansion between two metals. Since different metals expand at different rates, one metal that is bonded to another , will bend in one direction when heated and will bend in the opposite direction when cooled (hence the term Bimetallic Thermometer or BiMets) This bending motion is transmitted, by a suitable mechanical linkage, to a pointer that moves across a calibrated scale Although not as accurate as liquid in glass thermometers, BiMets are more hardy, easy to read and have a wider span, making them ideal for many industrial applications.

2.5 THE 20TH CENTURY: FURTHER DISCOVERY, REFINEMENT AND RECOGNITION

The 20th century has seen the discovery of semiconductor devices, such as the thermistor, the integrated circuit sensor, a range of non-contact sensors and also fibre-optic temperature sensors. Also, Lord Kelvin was finally rewarded for his early work in temperature measurement The increments of the Kelvin scale were changed from degrees to Kelvins. Now we no longer say "one-hundred degrees Kelvin," we instead say "one-hondred Kelvins". The "Centigrade" scale was changed to the "Celsius" scale, in honour of Anders Celsius

The 20th century also saw the refinement of the temperature scale. Temperatures can now be measured to within about 0.001°C over a wide range, although it is not a simple task. The most recent change occurred with the updating of the International

Temperature Scale in 1990 to the International Temperature Scale of 1990 (ITS-90). This document also covers the recent history of temperature standards.

2.6 HISTORY OF TEMPERATURE REGULATION

The modern history of temperature control began in the late 19th century under neonatal studies with the observation by Pierre Budin at the Paris Maternity Hospital that mortality rates decreased from 66% to 38% in infants whose birth weights were less than 2,000 g following introduction of temperature control measures. In 1957, Silverman reported that the survival of preterm infants in the first days of life was higher when the relative humidity was maintained at 80% to 90%. During this study, it was noted that the mean body temperature of infants maintained in humidified environments was significantly higher than the mean body temperature of infants in incubators with lower relative humidity (30% to 60%). This observation led to the formulation of the normothermic hypothesis, which states that the survival of preterm infants is favorably influenced by environments that maintain normal body temperature in particular, very low-birthweigh.

However, the history of computing began with analogue machine. In 1623 German scientist Wilhem Schikard invented a machine that used 11 complete and 6 incomplete sprocketed wheels that could add, and with the aid of logarithm tables, multiply and divide. French philosopher, mathematician, and physicist Blaise Pascal invented a machine in 1642 that added and subtracted, automatically carrying and borrowing digits from column to column. Pascal built 50 copies of his machine, but most served as curiosities in parlors of the wealthy. Seventeenth-century German mathematician Gottfried Leibniz designed a special gearing system to enable multiplication on Pascal's machine.

In the early 19th century French inventor Joseph-Marie Jacquard devised a specialized type of computer, a silk loom. Jacquard's loom used punched cards to program patterns that helped the loom create woven fabrics. When Jacquard died, more than 30,000 of his looms existed in Lyon. The looms are still used today, especially in the manufacture of fine furniture fabrics.

In the 1930s American mathematician Howard Aiken developed the Mark I calculating machine, which was built by IBM. This electronic calculating machine used relays and electromagnetic components to replace mechanical components. In later machines, Aiken used vacuum tubes and solid state transistors (tiny electrical switches) to manipulate the binary numbers.

Due to the limitations of analogue computers, there was a need for the invention of digital systems which help to circumvent most of the difficulties faced by analogue the first large-scale, general purpose, digital computer, ENIAC (Electronic Numerical Integrator And Computer), was built. ENIAC was initially built for the United States military to calculate the paths of artillery shells. It was later used to make calculations for nuclear weapons research, weather prediction, and wind tunnel design. ENIAC was introduced to the public in February 1946 and was used until October 1955. ENIAC was built by American physicist John W. Mauchly and American electrical engineer John Presper Eckert, Jr., at the Moore School of Electrical Engineering, University of Pennsylvania. Eckert and Mauchly successfully demonstrated ENIAC less than three years after the Army commissioned its construction.

2.7 USES OF COMPUTERS

Starting in the mid-20th century, machines—particularly computers—performed many of the bookkeeping functions that are vital to accounting systems. The widespread use of computers broadened the scope of bookkeeping, and the term *data processing* now frequently encompasses bookkeeping.

Another application of computers is realized in Spreadsheet Programming, an application program commonly used for budgets, forecasting, and other finance-related tasks. In a spreadsheet program, data and formulas to calculate those data are entered into ledgerlike forms (spreadsheets or worksheets) for analysis, tracking, planning, or "what-if" evaluations of the impacts of real or proposed changes on an economic strategy.

The most exciting of all the applications of computer is in real life control systems where computer is made to take over the manual operations of systems of which temperature control systems is one.

2.8 COMPUTER CONTROL

2.8.1 INTROUCTION

When the development of very capable and very reliable models, digital computers quickly became popular elements of industrial-plant-control problems. Computers are applied to industrial plant control problems in three ways, for supervisory or optimizing control; direct digital control and hierarchy.

2.8.1.1 supervisory or optimizing control:

In this form of control the computer operators in an external or secondary capacity, changing the set points in the primary plant control system either directly or

through manual intervention. A chemical process, for example, may take place in a furnace, the temperature off which is thermostatically regulated. For various reasons, the supervisory control system might intervene to reset the thermostat to a different level. The task of supervisory control is thus to "trim" the plant operation, hereby lowering costs or increasing production Though the overall potential for gain from supervisory control is sharply limited, a malfunction of the computer cannot adversely affect the plant

2.8.1.2 Direct digital control:

In direct digital control, a single digital computer replaces a group of single loop analogue computer replaces a group of single loop analogue controllers. Its greater computational ability makes the substitution possible, with obvious cost savings, and also permits the application of more complex advanced control techniques.

2.8.1.3 Hierarchy control:

Hierarchy control attempts to apply computers to all the plant control situations simultaneously. As such, it requires the most advanced computers and the most sophisticated automatic control devices to integrate the plant operation at every level from the top management decision to the movement of a value

When a digital computer is used as the controller then an analog to digital converter is interfaced, the input converts the analog signal from the measuring transducer. A digital to analog converter is interfaced to the output to provide the variable analog signal to control the process

2.8.2 ADVANTAGES OF COMPUTER CONTROL

The main advantage offered by the digital computer over the conventional control system described earlier is that the computer can be programmed readily to carry out wide variety of separate tasks. In addition, it s fairly easy to change the program so as to carry out a new or revised of tasks should the nature of the process change or the previously proposed prove to be inadequate for the proposed task. With digital computers this can usually be done with no change to the physical equipment of the control system. For the conventional control case, at least, some of the physical hardware apparatus of the control system must be replaced in order to achieve new functions or new implementations of them.

2.9 INTERFACING SENSORY DEVICES TO THE COMPUTER

In order to further extend the use of computer beyond the ordinary calculations, solving of different forms of equations, typing, storing information e t.c, one can use it to control other external devices. The process of connecting these external devices to a computer is known as INTERFACING and the term INTERFACE is used in this case to signify the physical device connecting the computer system to external devices to monitor physical variable. The nature of the computer and the interfacing between it and the sensory device is shown in fig 2.1

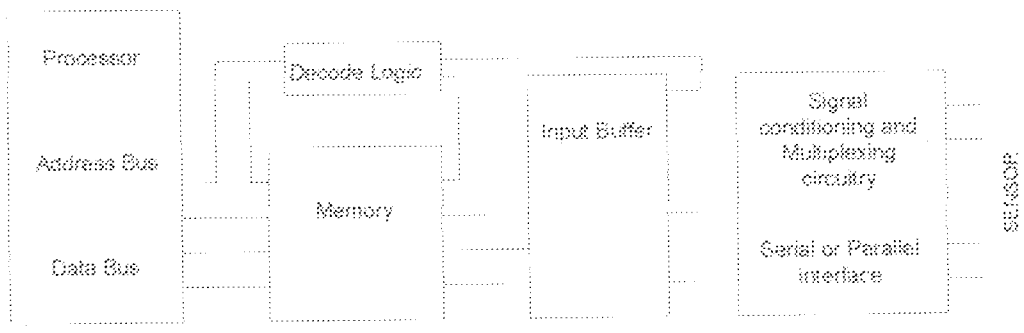


Fig 2.1 Computer sensory device interface

The interface allows the computer to control the passage of sensory information, via the interface bus, to the computer. It is comprised of

1. An input buffer
2. Multiplexing and \ or signal conditioning circuits

If an interface is required, in the shortest possible time, it is devisable to obtain system components compatible with a standard interface bus. Like the IEEE bus. Clearly the computer used would also have to be compatible with this bus and probably be supplied with routines of high level language capable of communicating with it. Applying standard interface techniques like this is especially advisable if the nature of the system is one of constant expansion or rearrangement. If the interface is likely to be relatively sample then it is cast effective to design it by one self. It may simply involve making the right connections to the computer.

The input buffer may simply be tri-state type like 74LS245 (fig 2.5) providing a path for parallel data transfer between the computer and interface bus. Fig 2.4(a) If serial data in the form of pulses is to be transferred, then some means of rendering this parallel is required. A shift register could possibly be used here fig 2.4(b)

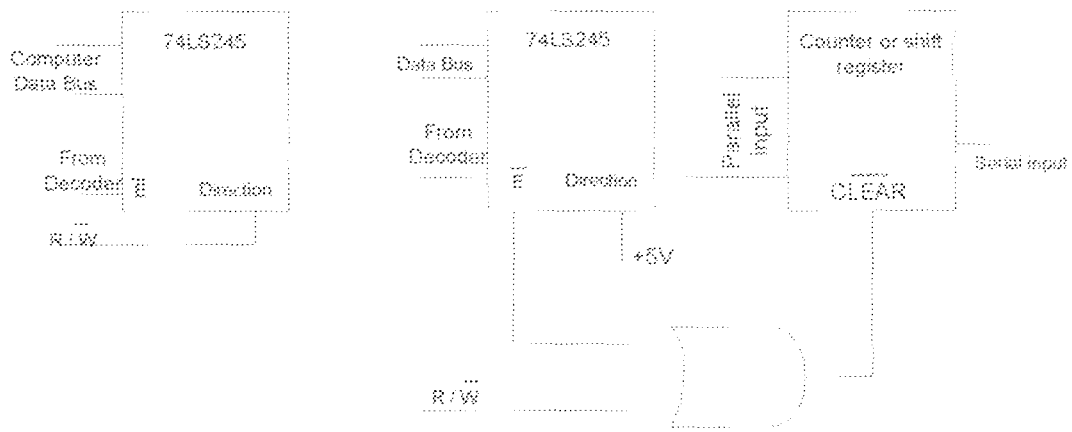


Fig 2.2 Parallel / Serial input Buffers

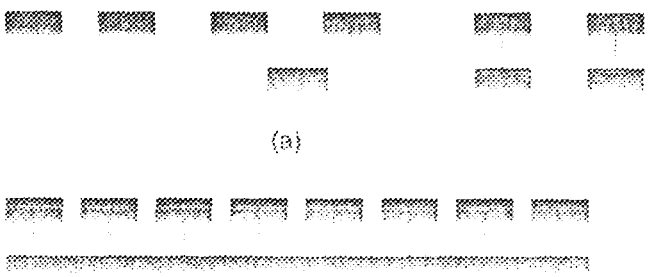
If more versatility is required at this point in the interface then one could employ an IC especially designed for the job. Like Rockwell 6522 (on board BBC microcomputer); Intel 8255, Motorola 6820 e t c. If the interface IC itself has to make decisions one could employ a single microprocessor e g Intel 8048, 8748, Rockwell 6801, Zilog Z8

These IC'S can provide parallel input or output (I/O) facilities and handle serial I/O in certain cases (e g 6522), with added features like timers and ROM on board (e g 8255). Interrupts can also be generated by these ICS relating to their current status Generally their features reflect the need for controlling data transfer into and out of the computer

2.10 COMPUTER PORTS

To make things simple, the industry has invented a set of standard interfaces called "ports" (probably they are called "ports" because they are ways to and from the outside world, by analogy with sea or airports in a country). There are basically two sorts of ports; serial and parallel. A serial sends or receives the bits of a byte at a time down a

pair of wires; a parallel down as many wires as it has bits. The parallel port can be said too have certain advantages over the serial port which could be simply understood in summary using a simple example of houses.



(a)

(b)

Fig. 2.3

If there are houses (see fig 2.3(a)), there is only one road. But as the houses are added, the number of roads needed to connect them increases faster than the number of houses. By the time there are five houses, ten roads are needed. It would not take long before an entire neighborhood was paved in asphalt.

Now picture the same neighborhood with driveways leading from each house to a common street (the bus) as shown in fig. 2.6(b). As can be seen, fewer connections are required than when all the house are connected directly to one another. The same is true in the computer.

What is actually seen when one looks at the port on the box of the computer is a 25-pin socket. To make use of it, the user (or the person who designed the peripheral you want to connect) has to know whether it is serial or parallel, whether it confirms to one of the standards –RS232 for serial, centronics or IEEE for parallel for example if the part is serial, what data feed it expects (and "Baud rate" in bits per second) and whether it does "handshaking" or expects some other protocol.

If the peripheral is some distance away - metres or hundred of miles- it becomes important to check the integrity of the data when it arrives. Inside the computer each port

is arranged to appear to the processor like memory locations, one for the data to be sent or received and one to tell which way it is going-the 'data' and 'status' bytes. This makes the work of the processor simpler. If it is sending data out of the part it just has to look occasionally at the status byte to see whether more data is needed. If it is, the program is halted and the processor writes more data to the parts. When the outside device signals that it has got enough, the status byte changes and the processor stops writing data to the port and carries on with the job in hand

2.11 USING THE PARALLEL PRINTER PORT AS AN INTERFACE

The parallel printer port is found on virtually all personal computers.

This port is often called a printer port because it is used almost exclusively with the printer. Occasionally it is called a CENTRONICS INTERFACE, named the centronics company which established the standard.

Basically the computer can only send out two things through the interface i.e data to be printed, and a special initialization signal that a printer is supposed to use to reset itself. Of course all sorts of special signals can be buried in the printer data itself. Most printers can have an elaborate set of control codes that can be sent to them in the stream of data they are to receive. But, these control codes are specific to each to printer, and they almost exclusively have to do with the formatting of the printed data (such as the selection of wide printing or underlining e.t.c).

The printing interface that is found on most computers offers a way of providing the opportunity to design, build and test interface circuits that are relatively simple, inexpensive and versatile. In addition, since the printer interface is both reasonable robust

and accessible without opening the computer case, the chances for computer damage are minimized.

2.11.1 Description of the parallel interface

The parallel printer interface on a personal computer (PC) is designed to write information consisting of text and format control of the printer and read status information such as print-complete and out-of-paper conditions from the printer. To provide the capabilities each printer interface actually consists of three independent ports- two output and one input.

The two output ports are full 8-bit (1-byte) port called the DATA port and e second 4-bit (1-nibble) port called the CONTROL port. The input is 5 bits wide and is called the STATUS port. All three ports have transistor-transistor logic (TTL), for which any voltage less than 0.8V corresponds to logic 0 and greater than 2.0V corresponds to logic 1. The most common parallel printer port connector is a female DB-25, though it is frequently necessary to use a cable with the DB-25 on the computer end and a centronics style 36-pin connector on the printer end. The connector pin definitions are given in Table 2.1.

TABLE 2.1 Connector pin definitions

DB25 pin	Centronics Pin	Register	In/Out	Bit	Name	Function
1	1	control	Out	CO*	STRODE	Signal by computer to Printer commanding that Data be read
2 – 9	2 – 9	Data	Out	D0 – D7	DATA -0 TO DATA -7	Bits of character sent to Printer
10	10	Status	In	S6	ACK	Signal by printer to Computer to indicate data received
11	11	Status	In	S7*	BUSY	Signal from printer to

						Computer to indicate Printer busy
12	12	Status	In	S5	PAPER END	Signal from printer to Computer to indicate Paper out
13	13	Status	In	S4	SELECT	
14	14	Control	Out	C1*	AUTO FEED	Signal from computer to Printer that auto line feed is required
15	32	Status	In	S3	ERROR	Signal from printer to computer that an error condition exist
16	31	Control	Out	C2	INITIATE	Signal from computer to instruct printer to reset itself
17	36	Control	Out	C3*	SLCT –IN	
18 – 25	19,21,23, 25,27,29,3 0,34					

The Bit column indicates the register bit number and sign of logic for each pin, numbering 0(least significant bit) to 7(most significant bit) in the register items marked with an asterisk(*) have their logic inverted, that is '0' bit corresponds to the pin high and '1' bit corresponds to pin low. The reason for the logic and names are buried in the early history of the personal computer. Indeed different manufacturers have different names for the functions. The pin-outs can be drawn out from the details in table 2.3 to further understand the arrangement.

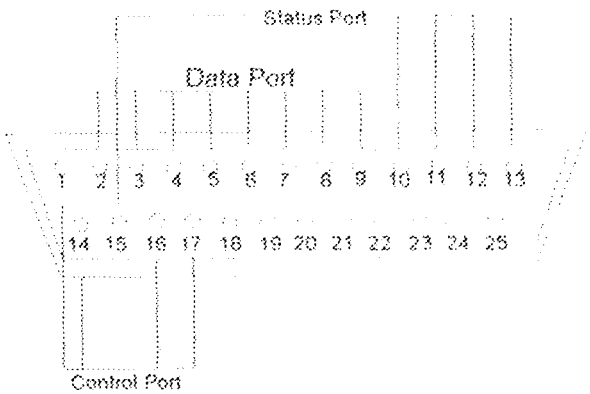


Fig 2.4 Computer parallel port pin – outs

Before deciding to use a parallel printer adaptor for control of experiments, one should confirm that there is one. The easiest way is to check the manual of the computer. Another way is to peek into the operating system to see what it thinks is there. DOS normally maintains an internal list of equipments it finds at startup.

CHAPTER THREE
THEORY OF DESIGN

3.1 BLOCK DIAGRAM

The block diagram shown in figure 3.1 shows the technique that was employed for carrying out the task. It shows a schematic diagram of temperature control of an Environment. The temperature in the Environment is converted to an analog voltage by a thermoelectric device (in this case LM35 which is an analog device). The analog voltage is converted to a digital voltage by Analog to Digital Converter (ADC).

The digital voltage is fed to a controller, which is in this case a personal computer, through an interface. This digital voltage is compared with the programmed input temperature and if there is any discrepancy (error), the controller sends out a signal to the either the heating element or cooling element, through an interface and relay in order to bring the temperature of the controlled environment to a desired value.

In other words, if the temperature of the environment exceeds the programmed input temperature, the controller sends a signal to switch the cooler ON while the heater goes OFF and if the temperature of the furnace falls below the programmed input temperature the controller sends a signal to switch the heater ON while the cooler
goesOFF.

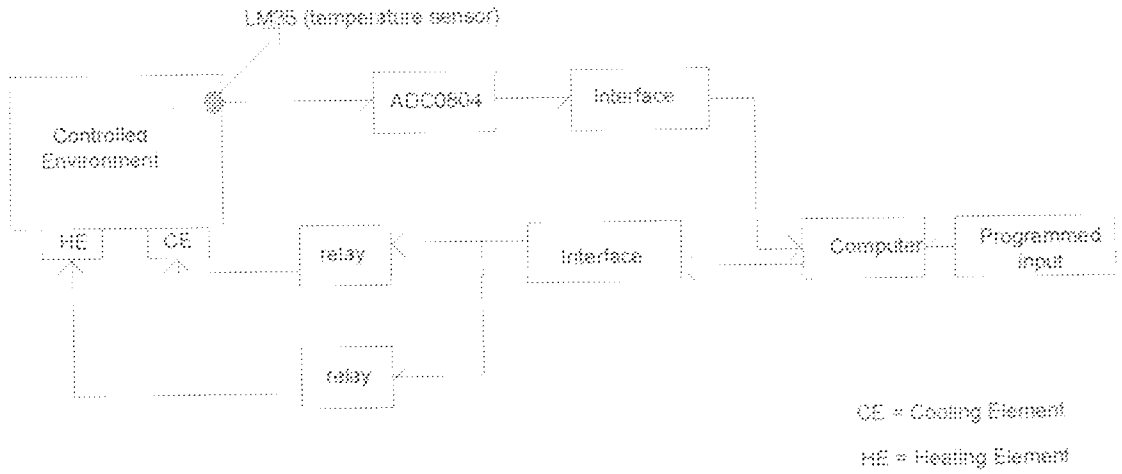


Fig. 3.1 Block Diagram of Project

The project work is designed around:

(a) An LM35 temperature sensor

(b) An ADC 0804 8-bit successive approximation register

(c) An SW74157 1-of-2 Digital Multiplexer

3.2 TRANSDUCERS

In any electric measurement or control system, all non-electrical quantities must be converted into electrical signals. The converters, called transducers, can be divided into two large groups: active and passive. An active transducer is one, which generates an output signal with no external source of energy involved. A passive transducer, however, changes one or many of its characteristic electrical properties and thus needs an external power source and signal conditioning device of appropriate complexity in order to generate a signal proportional to the variation in the measured non-electrical variable.

The relationship between the non-electric input signal and the output voltage or current generated is called the transfer characteristic of the transducer. It is

desirable, but not essential now a days, that the transfer characteristic is linear over the entire range of the input variable. Another vital characteristic is the frequency of response of the transducer and it is preferred that the transfer characteristic stays unaffected by the change in frequency over as wide a range as possible. For any real application, the knowledge of the output impedance of the transducer is of great importance as the structure of the signal-conditioning device will strongly depend on the value of that parameter.

3.2.1 TEMPERATURE TRANSDUCERS

Temperature measurement is commonly required in system monitoring applications and general experimentation for computer based systems. temperature transducers are employed to generate the necessary output in voltage form. Temperature transducers are used extensively in process industries such as chemical, food and pharmaceuticals, where control of temperature during manufacturing is important.

Four commonly used temperature transducers are the thermocouple, the resistance-temperature detector (RTD) the thermistor, and LM 35. But for the purpose of this project work, LM35 is employed.

3.2.1.1 LM35

This is a 3- terminal precision integrated circuit temperature sensor whose output voltage is nearly proportional to the celcius (centigrade) temperature. The LM35 has an advantage over linear temperature sensor calibrated in Kelvin as the user does not need to subtract a large constant voltage from its output to obtain a convenient centigrade scaling.

The LM35 does not require any external calibration to provide typical accuracies of +- 0.25C at room temperature and +- 0.75C over a full -55 to 150C temperature range.

The basic centigrade temperature circuit depicted below senses temperature between +20C to +150C.

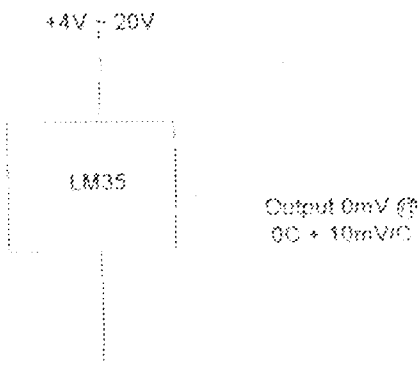


Fig 3.2 LM35 Temperature Sensor

Features

- Calibrated directly in centigrade
- Linear +10.0mV/C scale factor
- 0.5C degree accuracy at 25C
- Rated for full -55 to 150C range
- Suitable for remote applications
- Operates from 4 to 30V
- Less than 60mA current drain
- Low self- heating 0.08C in still air
- Non linearity, +-0.25C typical
- Low impedance output, 0.1 Ohms for 1mA load.

3.2.1.2 APPLICATIONS

The LM35 can be applied in the same way as other integrated – circuit temperature sensors. These applications include:

- Battery Management
- Fax Machines
- Printers
- Portable Medical Instruments
- Power Supply Modules
- Disk Drives
- Computers
- Automotive

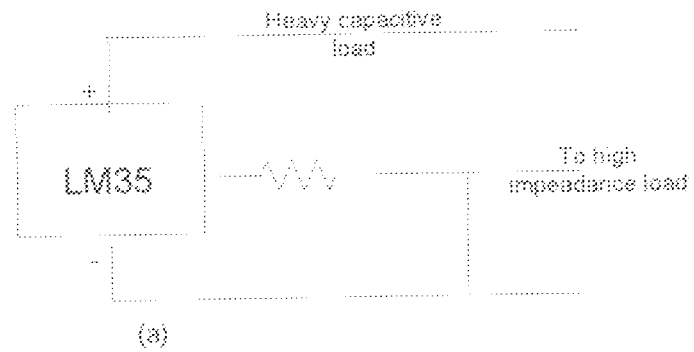
It can be glued or cemented to a surface and its temperature will be within about 0.01C of the surface temperature.

This presumes that the ambient air temperature is almost the same as the surface temperature; if the air temperature were much higher or lower than the surface temperature, the actual temperature of the LM35 dies will be at an intermediate temperature between the surface temperature and air temperature. This is especially true for the TO – 92 plastic package where the copper leads are the principal thermal path to carry heat into the device so its temperature might be closer to the air temperature than to the surface temperature.

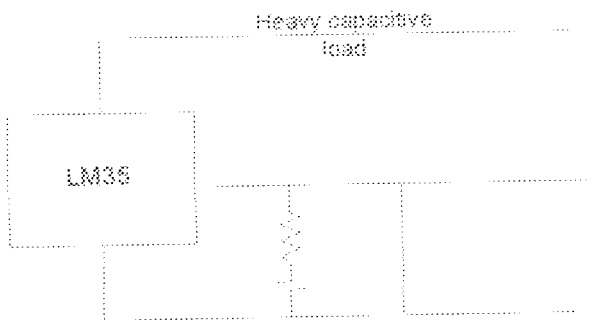
3.2.1.3 CAPACITIVE LOADS OF LM35

Like most micropower circuits, the LM35 has a limited ability to drive heavy capacitive loads. The LM35 by itself, is able to drive 50 pF without special precautions. If heavier loads are anticipated, it is easier to isolate or decouple the load

with a resistor (Fig 3.3a) or improve the tolerance of the capacitance with a series R - C damper from output to ground (Fig 3.3b)



(a)



(b)

Fig 3.3

3.3 ADC 0804 8-BIT ADC CONVERTER

This ADC0804 is a CMOS 8-bit successive approximation A/D converter using a differential potentiometric ladder. This converter is designed to allow operation with most microprocessors control buses with TRISTATE output latches directly driving the data bus. Thus this ADC appears like a memory location or I/O ports to the Microprocessor needing no interfacing logic.

Differential analogue voltage inputs allow increasing the common anode rejection and offsetting the analogue zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding smaller analogue voltage to the full 8-bits of resolution.

3.3.1 FEATURES:

- compatible with 8080 microprocessor derivative - interfacing logic needed access time is 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analogue voltage inputs
- Logic Inputs/Outputs meets both MOS and TTL voltage level specifications
- On – chip clock generator
- 0V to 5V analogue input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20 – pin DIP package
- 20 - pin molded chip carrier or small outline package

3.3.2 KEY SPECIFICATIONS

- Resolution 8 – bits
- Total error +- 1 LSB
- Conversion time 100 microseconds
- Maximum clock freq 1.46 MHz

3.3.3 FUNCTIONAL DESCRIPTION

The ADC0804 contains a circuit equivalent of the 256R network. Analogue switches are sequenced by successive approximation logic to match the analogue difference input voltage [Vin(+) – Vin(-)] to a corresponding tap on the R network.

The most significant bit is tested first and after eight comparisons (64 clock cycles), a digital 8 – bit binary code (1111 1111 = full scale) is transferred to an output latch and then an interrupt is asserted (INTR makes a high – to low transition). The device may be operated in the free running mode by connecting INTR to WR input with CS = 0. to ensure start – up under all possible conditions, an external WR pulse is required during the first power up cycle.

The device can also be operated by toggling WR input with CS= RD =0.

In this case, the output data register is updated after every conversion. WR is pulsed low, then high to start a conversion. After about 100 microseconds, the 8 – bit equivalent of the analogue input voltage is then available at the output.

A new conversion can be done by taking the WR low then high again.

3.3.4 DIGITAL CONTROL INPUTS

The digital control inputs (CS,WR,RD) meet standard TTL logic voltage levels. These signals are active low to allow an easy interface to microprocessor busses. For non – processor based applications, the CS input (pin1) can be grounded and the standard A/D start function is obtained by an active low pulse applied to the WR input (pin3), and the output enabled function is caused by an active low pulse at the RD input (pin2). Alternatively, CS and RD can be grounded and A – D conversion process controlled by the logic level on the WR input.

3.3.5 REFERENCE VOLTAGE

For every ADC device, there is always the need for a reference voltage that provides a scaling for the output binary bits as a function of the input voltage.

The output binary code as a function of the input voltage for n – bits of resolution is approximately expressed by:

Binary output = $(2^n * Vin)/Vspan$

Where Vspan = input voltage needed to produce a full scale output from the ADC, i.e all output bits set to 1.

For the ADC0804 used in the project work

Vspan = 2.56V n = 8 – bits.

Therefore

Binary output = $(2^8 \times Vin)/2.56$

$= (256 \times Vin)/2.56$

$= 100 \times Vin$

The above expression relates the output binary value as a function of the input voltage. That is for a Vin of 1.0V, the binary output = $100_{10} = 64_{16}$.

3.3.6 CLOCKING OPTIONS

The clock from the ADC can be derived from an external clock or an external RC can be added to provide self – clocking. The CLK (pin4) makes use of a schmit trigger.

For the project work, R = 10KΩ, C = 500pF

$F_{CLK} = 1/(1.1 \times RC)$

$= 1/(1.1 \times 10K \times 500p)$

$= 182KHz$

3.4 POWER SUPPLY

The project work is powered off a regulated 5V dc supply derived from a 12V ac 50Hz input obtained from the 240V 50Hz mains input by means of a 20 : 1 step-down transformer.

The 12V ac input is rectified by a bridge rectifier, smoothened by a 25V 2200microfarad capacitor and regulated by a 7805 IC to produce a ripple - free 5V dc output that powered the analogue / digital subsystems.

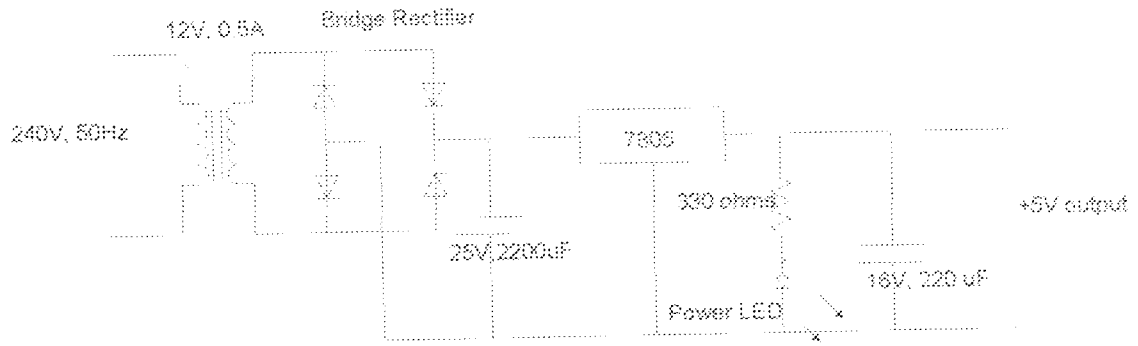


Fig 3.4 Power circuit

3.5 REFERENCE VOLTAGE GENERATOR

This is built on an LM358 dual Op – Amp configured as shown below

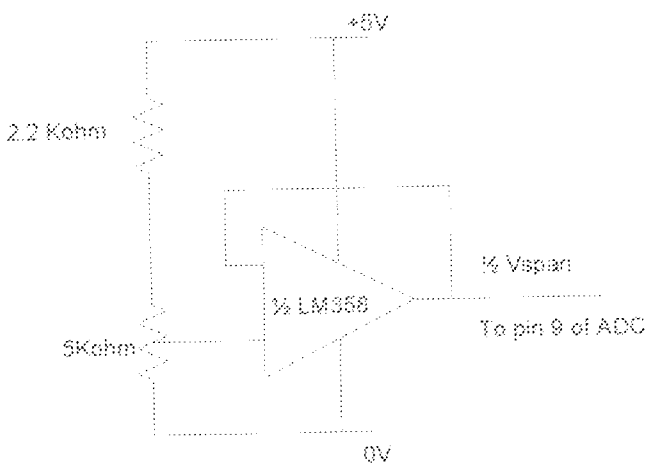


Fig 3.5 LM358 Dual Op – Amp

The LM358 provides a buffered reference voltage for the converter chip. For the ADC0804, Vref is always Vspan/2, i.e half the voltage that defines the maximum

encoded binary output. This is set to 1.28V corresponding to 2.56V full – scale. This scaling eliminate the need for any other conversion computation since the ADC0804 output is now directly the binary representation of the input signal voltage.

For example, assuming an input signal voltage of 1V on a Vfull – scale of 2.56V

2.56V = 256

1.0V = X

X= (256 x 1.0)/2.56

= 100

This corresponds to 100C ambient temperature since at 100C, the LM35 has an output voltage of 1.0V. Thus the scaling provide a linear output that directly provides the measured ambient temperature.

3.6 RELAY TIMERS

Two relays are provided to control a heating unit and a cooling unit. Since the project is intended for process temperature control, a means of accurately controlling the temperature within defined bounds is essential.

Thus either the heater or the cooler is ON, depending on the mode of operation selected for the monitoring. The relays are wired to control transistors as shown below

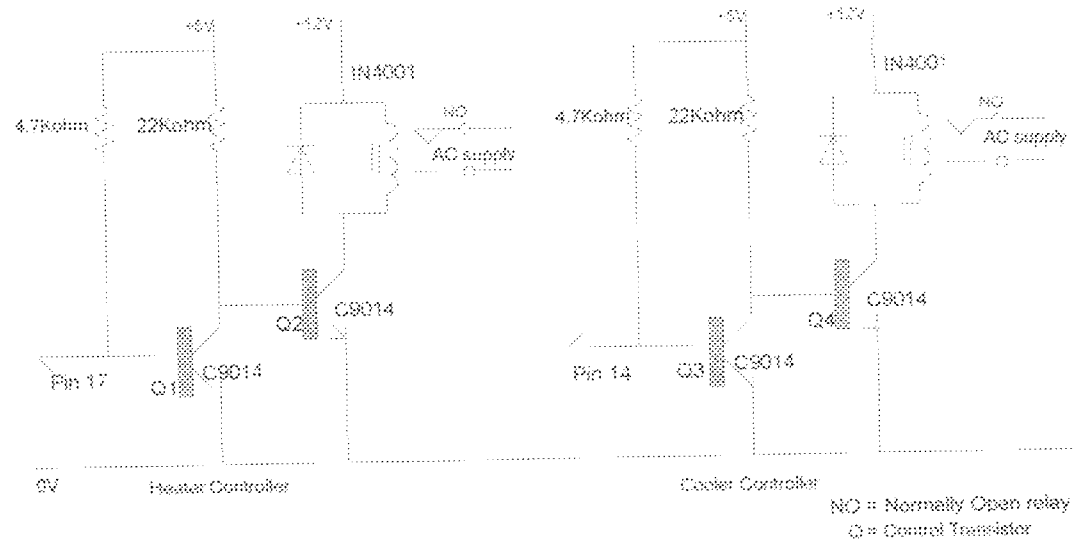


Fig 3.6 Relay circuit

Since the control register output on the IBM PC are open – collector and initialized to 1s at power up, an inverter is needed before the relay switching transistor to prevent undesirable operation. The inverters Q1 and Q3 placed before Q2 and Q4 work to prevent this.

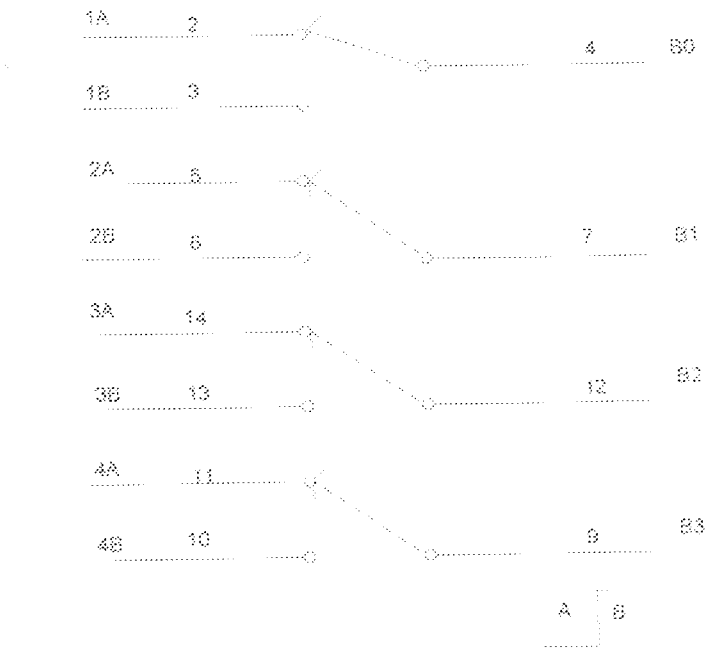
When the module is interfaced to the DB25 connector on the PC, with the control outputs initialized to 1s, the two relays are held off until the monitoring software starts issuing control signals

The heating system is controlled by bit 1 of the control register and the cooling system by bit 3.

3.7 QUAD 1 – OF – 2 MULTIPLEXER

This works on the design of a 74LS157 TTL device. This device has two 4 – bits inputs designated as 1A 2A 3A 4A and 1B 2B 3B 4B

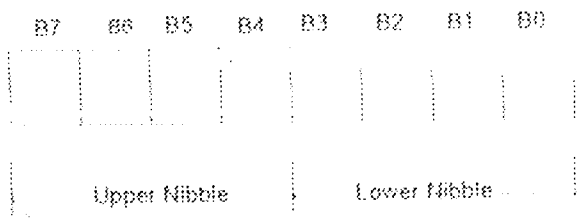
And a single 4 – bits output thus 4 – bits output can be obtained from the 8 – bits input by selecting between the 1A – 4A inputs or 1B – 4B inputs as designated below



Thus depending on the state of a control select input on pin 1, the 4 – bits inputs on 1A – 4A are passed to the output, or the 4 – bits inputs on 1B – 4B.

The quad 1 – of – 2 multiplexer was used because of the unavailability of a bidirectional printer card which would have made reading of the 8 – bits output of the ADC easier by simply reversing the parallel port data direction using bit 5 of the control register at address 0x37A.

To circumvent this difficulty, the 8 – bits digital output of the ADC is read on a high_nibble, low_nibble basis over the status register at address 0x37A. The reading is done upper nibble first (B7 – B4). The SN74157 is switched to select the four lower nibble (B3 – B0), and the controlling software reads this nibble. An 8 – bits output is formed by masking and shifting right the input bits.



The upper nibble is read over the status register and ANDed with the bit pattern 11110000 to mask out the four lower bits and XORed with 0x80 to negate the inversion on bit 7.

The lower nibble is read then ANDed with the same bit pattern and then XORed with 0x80 and shifted right four places before ORing with the upper nibble.

In the control software, the lower nibble is assigned to the variable temp_value_low_nibble

The temp_value_high_nibble = (inp(statusport)&0xF0)^0x80;

It addresses multiplexer and read again.

The temp_value_low_nibble = (inp(statusport)&0xF0)^0x80)>>4;

Then temp_value = temp_value_high_nibble + temp_value_low_nibble;

3.8 COMPUTER PROGRAMMING

A program is a set of instructions written (coded) in different languages, which a computer follows in order to accomplish a task. Examples of these programming languages include QBasic, C++, Java, Fotran e.t.c.

A computer can be programmed in both a low level language like assembly or in a high level language like Basic, C++ e.t.c. There are five major steps in high level programming.

These steps are enumerated as below

- Program Analysis
- Program Design
- Program Coding
- Testing and Debugging
- Program Documentation

Program analysis involves defining the problems involved and answering questions such as

i. Is it computer solvable?
ii. What are the program objectives?
iii. What are the desired outputs?
iv. What are the desired input data?

Program design is achieved using structured programming technique. This technique involves breaking the whole problem into program modules. An outline of program module is then written in a narrative form known as "PSEUDOCODE" (i e a form, where instructions are written step by step). The detailed sequence for each module can then be represented graphically in what is known as a flow "FLOWCHART". Flowchart will enable the programmer to see clearly, the logical flow of the instructions, how each module relate to another and how each module is linked using logic structure called "sequence or loop". Program coding is the art of writing the actual set of instructions (program) using the logical step, developed in the program design, in a language (high level) bearing in mind the following qualities.

a. The program should be structured

b. It should be reliable

c. It should be machine understandable

d. It should be able to detect common input errors

After the program has been coded in a particular programming language, it is then required to enter it into a computer where it will be test ran and debugged

Debugging is the art of subjecting a program to a trial and correction to remove common errors. There are basically four types of errors prominently notable in programming;

i. Syntax errors

ii. Logical errors

iii. Execution errors

iv. Data errors

Syntax errors occur when a programmer violates the format of a program command

E.g age = 20;

Cout<<ag<<endln;

When the above program is ran, the second line generate "syntax error" this is because the variable "ag" does not have "e" attached to it

Logical errors appear in cases where the program runs successfully but gives the wrong result. It should be noted here that the computer does not have a means of detecting logical errors.

Execution errors are caused when asking the computer to perform the task which it can not do. For example asking the computer to divide a value by zero or asking a computer to give the square root of a negative number or using a wrong variable type.

Data error concern the data supplied to the computer for the execution of the program. Wrong data is another factor that can lead to a wrong result from the computer.

3.9 PROGRAMMING THE COMPUTER FOR THE TASK

Before carrying out the task of programming the computer to control temperature must consider the choice of programming language. It must be capable of carrying out fast exchange of data to and from the computer. This is necessary as the rate of change of temperature of the external system may be very fast and as such would require a very fast control signal applied when correction is necessary. The programming language must as well be able to perform scientific calculations on such data it receives, it is desirable that it is to understand and can provide information (if possible in graphical form) to the use during actual process control

These are basically the reasons for using C++ as a choice language for the task, it is relatively faster in accessing the ports and there is also an increased readability and understandability of the functions employed in achieving the task.

The program codes can be seen in Appendix A

3.10 ENTIRE CIRCUIT DIAGRAM OF PROJECT WORK

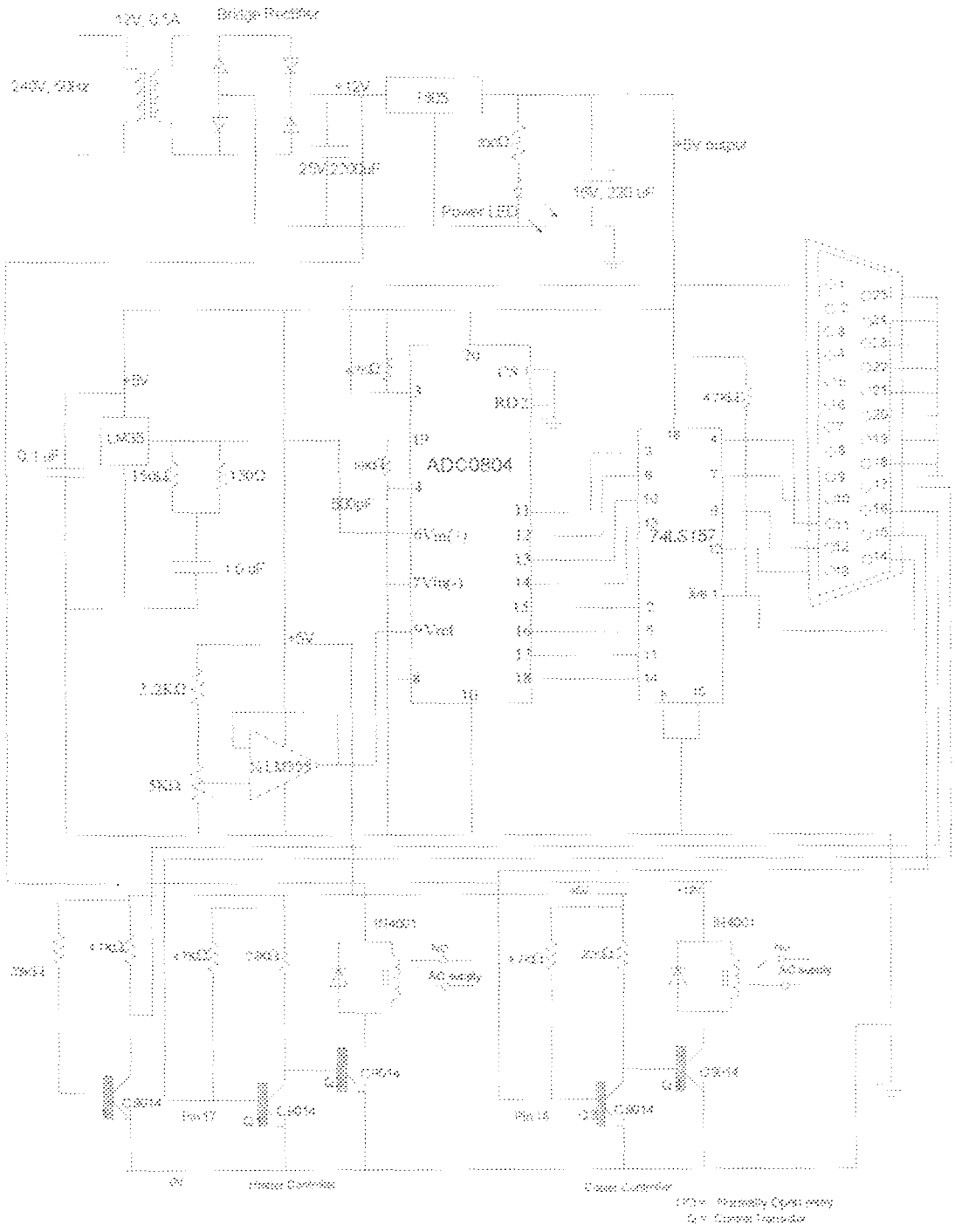


Fig 3.7 project circuit diagram

CHAPTER FOUR

4.0 CONSTRUCTION AND TESTING

Some of the stages of the circuits, like the power supply unit were first tested on breadboard. The wire links were made as short as possible to prevent unnecessary interference from external signal and all the components were ensured to be firmly connected

It was then transferred to the veroboard as designed and soft soldering was applied with the aid of some connecting wire links to form a permanent system.

All components except the temperature sensor were accommodated on the veroboard. All components were ensured to have a common ground and also distinct and different colors of wires were used to avoid confusion and to enable distinguishability

When constructing, ribbon were used to ensure an orderly arrangement of wires along the veroboard

The DB25 socket was acquired by dismantling a DB25 connecter cable and unsoldering the connecting wires. It was held onto the veroboard by the use of fabricated sheet metal joint fastened to the casing with nuts and bolts

A casing structure was constructed to house the system for adequate protection and to suit portability.

The connection of the circuit to the computer should however be made after the computer has been fully booted or else "handshaking" takes place between the computer and the circuit because the computer assumes the presence of a printer instead and as such unwanted signals are sent.

4.1 OPERATIONAL MODES OF CONTROL UNIT

The software is developed in such a way as to cater for four different operations modes in the process control design. These modes are namely

1. Temperature Reduction mode (cooler mode).
2. Temperature Elevation mode (heater mode).
3. Windows Monitoring mode (limits mode).
4. Demo Mode.

The actions performed by the software are detailed in 4.1 below

TABLE 4.1

Mode	Temp. > Preset	Temp. < Preset
R (reduction mode)	Cooling unit ON	Cooling unit OFF
E (elevation mode)	Heating unit OFF	Heating unit ON
W (window mode)	Lower preset < Temp < Upper preset	
	Cooling + Heating unit OFF	
	Temp > Upper preset	
	Cooling unit ON	
	Temp < lower preset	
D (demo mode)	Heating unit ON This mode only reads the current temperature of the particular environment in question and displays the result on the screen	

Each conversion cycle is started by issuing a 1 to bit 0 of the control register, then a 0 to the same bit position after a short delay.

4.2 PROGRAM CODE

The program code is as seen in Appendix A

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The use of computer for digital control of temperature with obvious cost savings, and also permitting the application of more complex techniques has been clearly shown in the context of this work so far. In the set up of such an application the computer serves as both the comparator and controller while the location of the substance whose heat is to be controlled serves as the plant. The temperature sensor, signal conditioning and analogue to digital converter in unison serves as the feedback element. The controlled variable is the analogue temperature while the manipulated and actuation signals are the signals to switch the heater/cooler ON or OFF and the signal from the computer through the interface respectively. The feedback signal is the signal from the analogue to digital converter through the interface that gives the status of the plant and the program input serves as the reference command as it dictates the specification of the plant.

In general the computer takes over the role of a thermostat with added advantages already discussed in chapter 2. The computer therefore exhibits a very flexible nature which when combined with external circuitry provides limitless functions that can not be over emphasized. As such we have the choice whether to exploit the such ingenuity of

our personal computers at homes, offices or even at industrial levels or to keep them at low performance or idle.

REFERENCES

[1] Microsoft Ecarta 2006 version

[2] John Billingsly

"Controlling with computers"

Portsmouth Polytechnic, 1989

[3] Thomas O Boucher

"Computer automation in manufacturing"

Department of Industrial Engineering

Rutger University, NJ

USA, 1996

[4] A Relinski

"Interfacing Transducers to the BBC Micro"

Microelectronics Educational Development Centre

UK, 1985.

[5] B L and A.K Theraja

Electrical Technology

S Chand publication

[6] C J. Fraser and J S Milne

"Microcomputer Applications in measurement system"

Dundee Institute of Technology

USA,1990

[7] David P. Beach and Roy G. Bridges

"Industrial Control Electronics"

Indiana State University

USA, 1990

[8] James Garrat

"Design and Technology"

Cambridge University

Great Britain, 1991

[9] E A Parr (ed)

"Industrial Control Handbook"

1st ED Oxford University

London, 1987.

[10] Thomas C Barte

"Digital Computer Fondamentals"

6th ED Harvard University

USA,1985.

[11] Dr Adediran Yinusa

Applied Electricity

[12] Feedback

"Computer Programming"

Basics Micamaster 960

Feedback Instrument Limited

England

[13] Engr. Musa Abdullahi

Lecture notes on Analogue electronics

ECE 314

2004

[14] Bramer and Bramer

"C for Engineers"

1991

[15] Schaum's Outline series

"Programming in C++"

1998

[16] Giorgio Rizzoni

"Principles and Applications of Electrical Engineering"

Third Edition

[17] Roger L Tokheim

"Digital Electronics"

[18] www.datasheetarchive.com.

APPENDIX A

PROGRAM CODE

```
#include <conio.h>
#include <iostream.h>
#include <windows.h>
#include <ctype.h>
#include <dos.h>
#include <stdio.h>
#include <bios.h>
static const int dataport = 0x3BC;
static const int statusport = dataport + 1;
static const int controlport = dataport + 2;
unsigned char start_conversion = 0x01;
unsigned char cooler_on = 0x02;
unsigned char heater_on = 0x08;
unsigned char cooler_off = 0x00;
unsigned char heater_off = 0x00;
unsigned char read_high_nybble = 0x04;
unsigned char read_low_nybble = 0x00;
```

```
unsigned int temporary_temp_value = 0x00;

unsigned int n,m;

float mode_0_max_temp, mode_1_min_temp, mode_2_min_temp, mode_2_max_temp;

unsigned char adc_power_ok, control_value,count;

unsigned int temp_value_high_nybble, temp_value_low_nybble,

char mode;

float temp_value;


void start_adc_get_temp_value();

void menu();

void check_adc_power();

void delay_here();

void demo_mode();

void main()

{

struct dostime_t t;

outportb(controlport, cooler_off+heater_off);

cout<<"\n            Welcome to ALONGE2006 PROCESS CONTROL DEMO
SOFTWARE\n";

_dos_gettime(&t);

printf("            You logged onto this demo at: %2d:%02d:%02d.", t.hour,
t.minute,t.second, t.hsecond);

cout<<"\n      This software allows heating/cooling control in a process environment";

cout<<"\n      A minimum temperature of 0 degrees and a maximum of 125 degrees
celcius";

cout<<"\n      Can be monitored. Resolution is 0.0001 degrees.";

cout<<"\n      Connect ADC module before running this demo or lockup will occur.";
```

```
cout<<"\n        This software is not for use outside of educational institutions",
cout<<"\n        and is the exclusive property of Engineer B  Frank  Ahonge",
cout<<"\n                                HEADHUNTAZ + Ahonge2006.\n",
for(n=0;n<1000;n++);
cout<<"\n        You can now power up the ADC \n",
check_adc_power(),
cout<<"\n                                Commencing data acquisition and process control \n";
do{
menu(),
switch(mode){
case 'R':
cout<<"\n                Cooler Mode On \n",
cout<<"\n\n\n";
do{
cout<<"\n                Enter maximum temperature for cooler mode >> ";
cin>>mode_0_max_temp;
}while(mode_0_max_temp<0 || mode_0_max_temp>127),
start_adc_get_temp_value(),
if(temp_value>mode_0_max_temp)control_value=cooler_on;
outportb(controlport, control_value),
cout<<"\n                ADC in data collection and control mode.",
cout<<"\n                Hit any key to return to main menu ",
while(!kbhit()){
delay_here(),
start_adc_get_temp_value(),
```

```
if(temp_value>mode_0_max_temp){
control_value=cooler_on;
}
else{
control_value=cooler_off;
}
outportb(controlport,control_value);
}
break;
case 'E':
cout<<"\n\n\n";
do{
cout<<"\n                Enter minimum temperature for heater mode >>: ";
cin>>mode_1_min_temp;
}while (mode_1_min_temp <0 || mode_1_min_temp > 127);
start_adc_get_temp_value();
if(temp_value<mode_1_min_temp)control_value=heater_on;
outportb(controlport, control_value);
cout<<"\n                ADC in data collection and control mode.\n";
cout<<"\n                Hit any key to return to main menu.\n";
while(!kbhit()){
delay_here();
start_adc_get_temp_value();
if(temp_value<mode_1_min_temp){
control_value=heater_on;
```

```
}
else{
control_value=heater_off;
}
outportb(controlport, control_value);
}
break;
case 'W':
cout<<"\n\n\n";
do{
cout<<"\n                Enter Window Mode Minimum Temperature >>: ";
cin>>mode_2_min_temp;
}
while (mode_2_min_temp < 0 || mode_2_min_temp>127);
do{
cout<<"\n                Enter Window Mode Maximum Temperature >>: ";
cin>>mode_2_max_temp;
}while(mode_2_max_temp <0 || mode_2_max_temp > 127 || mode_2_max_temp <=
mode_2_min_temp);
start_adc_get_temp_value();
if((temp_value>mode_2_min_temp)&&(temp_value<mode_2_max_temp))control_value
=heater_off+cooler_off;
if(temp_value<mode_2_min_temp) control_value=heater_on;
if(temp_value>mode_2_max_temp) control_value=cooler_on;
outportb(controlport, control_value);
cout<<"\n                ADC in data collection and control mode \n";
```

```
cout<<"\n                Hit any key to return to main menu \n";
while(!kbhit()){
delay_here();
start_adc_get_temp_value();
if((temp_value>mode_2_min_temp)&&(temp_value<mode_2_max_temp))control_value
=cooler_off+heater_off;
if(temp_value<mode_2_min_temp)control_value=heater_on;
if(temp_value>mode_2_max_temp)control_value=cooler_on;
 outportb(controlport, control_value);
}
break;
case 'D':
                        demo_mode();
break;
default:
cout<<"        \n\n\n\n        Data acquisition terminated \n";
cout<<"*************************************************************************
****************";
cout<<"\n          Ending ALONGF2006 DEMO CONTROL SOFTWARE \n\n\n\n";
cout<<"\n            Goodbye. Have a nice day.\n";
outportb(controlport,0x00);
break;
}
}while(mode!='X');
return;
}
```

```
void menu()

{

do{

cout<<"\n                    Main Menu:\n\n";

cout<<"                  Temperature (R)eduction Mode:\n";

cout<<"                  Temperature (E)levation Mode:\n";

cout<<"                  (W)indow Monitor Mode  \n";

cout<<"                  (D)EMO MODE\n";

cout<<"                  E(X)IT\n";

cout<<"\n       Enter ADC operating mode (R,E,W,D,X):";

cin>>mode;

mode = toupper(mode);

}while((mode!='X') && (mode!='E') && (mode!='R') && (mode!= 'W') &&
(mode!='D'));

}

void start_adc_get_temp_value()

{

outportb(controlport, control_value+start_conversion);

for(n=0;n<10;n++);

outportb(controlport, control_value);

for (n=0;n<100;n++){

for(m=0; m<1000;m++);

}

outportb(controlport, control_value+read_high_nybble);

temp_value_high_nybble = (inp(statusport)& 0xf0)^0x80;

for(n=0;n<100;n++);
```

```
outportb(controlport, control_value+read_low_nybble);
temp_value_low_nybble = ((inp(statusport)&0x f0)^0x80)>>4;
temp_value = (temp_value_low_nybble+temp_value_high_nybble)/1.7066;
}
void check_adc_power()
{
do{
adc_power_ok = (inportb(statusport)& 0x08);
} while (adc_power_ok !=0x00);
cout<<"\tADC power ok ";
return;
}
void delay_here()
{
for(n=0; n<5000; n++){
for(m=0; m<5000; m++);
}
return;
}
void demo_mode()
{
cout<<"\n\n\n";
cout<<"\t      Welcome to the DEMO MODE ";
cout<<"\n      This demo software monitors and displays ten counts of the ambient
temperature ".
cout<<"\n      around  the sensor\n";
```

```
cout << "\n    Temperature readings are diplayed on the screen.\n";
cout<<"\n       Press ENTER key to display temperature values";
delay_here();
delay_here();
cout<<"\n       Entering demo mode ";
for(count = 0; count<10; count++){
start_adc_get_temp_value();
cout<<"\n\n\nCurrent Temperature Reading:  "<< temp_value<<" degrees celsius";
getchar();
delay_here();
}
cout<<"\n       DEMO MODE EXITED";
delay_here();
delay_here();
cout<<"\n\n\n\n";
}
```