

**DEVELOPMENT OF A MICRO CONTROLLER BASED SMART TEMPERATURE CONTROLLER**

**By**

**Md. Nahid Hasan**

**POST GRADUATE DIPLOMA IN INFORMATION AND COMMUNICATION  
TECHNOLOGY**

**Institute of Information and Communication Technology  
BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY  
September, 2014.**

The project report titled “Development of a Micro controller Based Smart Temperature Controller” submitted by Md. Nahid Hasan, Student ID:0412311010, Session 2012-2013 has been accepted as satisfactory in partial fulfillment of the requirement of the Post Graduate Diploma in ICT held on 16<sup>th</sup> September,2014.

### **BOARD OF EXAMINERS**

---

1. Dr. Md. Liakot Ali Professor Institute of Information and Communication Technology BUET,Dhaka-1000.	Supervisor
---	------------

---

2. Dr. Mohammad Shah Alam Assistant Professor Institute of Information and Communication Technology BUET,Dhaka-1000.	Member
---	--------

---

1. Mohammad Imam Hasan Bin Asad Lecturer Institute of Information and Communication Technology BUET,Dhaka-1000.	Member
--	--------

## CANDIDATE'S DECLARATION

It is hereby declared that this project report or any part of it has not been submitted elsewhere for the award of any degree or diploma.

---

Md. Nahid Hasan  
Student ID:0412311010  
Institute of Information and Communication Technology,  
BUET,Dhaka-1000.

Dedicated

To

My Father “Mridha Wazed Ali” and Mother “Mst. Nazmun Nahar”

## Table of Contents

---

Title	Page No.
Board of Examiners	i
Candidate's Declaration	ii
Dedication	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
List of Abbreviations	viii
Acknowledgement	ix
Abstract	x

### **Chapter 1**

#### **Introduction**

1.1 Introduction	2
1.2 Objective with Specific Aims and Possible Outcomes	3
1.3 Organization of this report	3

### **Chapter 2**

#### **Micro Controllers**

2.1 Introduction	5
2.2 Microcontroller	5
2.3 Different types of controller	7
2.3.1 On-Off Controller	7
2.3.2 Proportional Control	7
2.3.3 Integral Control	8



4.2.2 Push Button and Power Connection of Microcontroller	32
4.2.3 Main Circuit	33
4.3 Simulation and Results	34
4.4 Breadboard	35
4.5 PCB	37

## **Chapter 5 Conclusion**

5.1 Conclusion	39
5.2 Future Works	39
References	40
Appendix A Software Code written in CCS C compiler	42
Appendix B PIC header file code	45

### **List of Tables:**

Table Number	Table Caption	Page No.
Table 3.1	LCD Interfacing with Microcontroller pin.	19

## List of Figures:

Figure No	Figure Caption	Page No.
2.1	(i)The die from an Intel 8742, an 8-bit microcontroller that includes a CPU running at 12 MHz, 128 bytes of RAM, 2048 bytes of EPROM, and I/O in the same chip. (ii) 2 ATmega microcontrollers.	6
3.1	Flow chart of the Project Implementation.	13
3.2	Block diagram representation of the smart temperature controller.	14
3.3	This figure shows the pin diagram for PIC16F877A	16
3.4	Picture of LM35 Temperature sensor.	18
3.5	The 16 header pin should be soldered to the LCD first.	19
3.6	Picture of BC 548 Transistor	21
3.7	Picture of static relay.	21
3.8	HEX file (program file) import at Proteus window	23
3.9	Hex file and other simulation file list of the Design with file extension.	24
3.10	PICF877A installed in PICkit2 Programmer with external vdd connection	25
3.11	PICF877A installed in PICkit2 Programmer with external vdd connection.	26
3.12	PIC programming using MPLAB v8.70	26
4.1	Power Supply circuit diagram for the controller.	31
4.2	Vdd, GND and Push button connection of Microcontroller	32
4.3	Main Circuit Diagram	33
4.4	Main Simulated Circuit Diagram	34
4.5	Snap of the circuit on bread board	35
4.6	Snap of the circuit in power on condition	36
4.7	Final PCB Design Snapshot using Orcad Layout plus	37



## **List of Abbreviations:**

PID	Proportional Integral Derivative
uC	Microcontroller
LCD	Liquid Cristal Display
CPU	Central Processing Unit
PCB	Printed Circuit Board
DC	Direct Current
AC	Alternating Current
GND	Ground
SW	Switch
LED	Light Emitting Diode
ROM	Read Only Memory
RAM	Random Access Memory

# Acknowledgement

First of all, I would like to thank Almighty Allah SWT for Given me the strength and patience for carrying out this work and to complete this project.

I would also like to thank Professor Dr. Md. Liakot Ali, Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka,Bangladesh for his guidance, valuable suggestions and constant encouragement throughout the whole period of the work,which inspired and guided me in each and every step of the project to complete it successfully in time.

I would also like to give special thanks to Mr. Md. Imam Hasan Bin Asad, Lecturer, Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka,Bangladesh for his suggestions and help during the software programming and hardware construction part of this project work which was really a very tough part of this project.

I would also like to thank to Professor Dr. Saiful Islam and Dr. Shah Alam of Institute of Information and Communication Technology, Bangladesh University of Engineering and Technology, Dhaka,Bangladesh for their encouragement during academic classes and discussions to complete the project work.

I would like to acknowledge the efforts given by my family members especially my parents for their continuous support and inspiration, which helped me to complete the project successfully.

# Abstract

The use of microcontroller based embedded system has been drastically increased over the years. Consequently the management and monitoring job using microcontroller based embedded system has been very popular due to its reduced level of complexity and maintenance. Today smart controller for controlling temperature of heater, Boilers ,power plants, chemical plants, oil refining plants etc. are very common in industry. Manual temperature control and monitoring can cause an accident if the temperature becomes outside of the operating range. So there is a need for the development of the complete microcontroller based smart temperature controller that will overcome the limitations of manual monitoring by personnel which in turn reduces cost of productions and losses due to damages . In this work, a complete microcontroller based smart temperature controller has been developed to keep the set temperature under control and continue the production of industry. The electric circuit for the proposed system has been designed using Proteus and then it has been simulated to validate the system's performance in varying temperature conditions. Then the prototype of the system has been implemented using the bread board and tested the desired functionality. Finally the proposed system has been miniaturized using printed circuit board (PCB).The developed system can be enhanced and commercialized in future.

# **Chapter 1**

## **Introduction**

## 1.1 Introduction

Process control is an engineering discipline that deals with architectures, mechanisms and algorithms for maintaining the output of a specific process within a desired range [1-2]. It is extensively used in industry and enables mass production of consistent products from continuously operated processes such as oil refining, paper manufacturing, chemicals, power plants and many others. Process control enables automation, by which a small staff of operating personnel can operate a complex process from a central control room. There were different types of controller such as open loop control, feedback control, derivative control, integral control, PID control etc. algorithm used in the industry. However they have limitations in terms of accuracy and other performance matrix [3].

Now proportional control is one of the common control algorithms used in industry for controlling different process parameters and has been widely accepted in industrial control [4-7]. The popularity of smart controllers can be attributed partly to their robust performance in a wide range of operating conditions and partly to their functional simplicity, which allows engineers to operate them in a simple, straightforward manner. Temperature is very important parameter in the different industrial process to be control. One of the causes that will cause an accident is overheating on the equipment and the extreme temperature will cause a failure to the equipment and systems if the temperature is outside of the operating range. The damage can affect the company budget to the maintenance or replacing the damage or accident that happened. It also may affect the quality time for the company to produce the product and also affect the reputation of the company as well.

Now a days microcontroller based embedded system is being used for developing many industrial control systems due to its accuracy and high speed in computation [8]. There are scopes of research in this area. So a microcontroller based smart temperature controller has been proposed and developed in the proposed project. It can be used for the automation of temperature controlling system like oil refining, paper manufacturing, power plants, garments

industry,chemical industry,manufacturing plants, homes,or any kind of industries where they need.

## **1.2 Objective with Specific Aims and Possible Outcomes**

The objectives of the project are as follows:

- I. To develop a hardware circuit for developing microcontroller based smart temperature controller.
- II. To develop a software for the proposed system.
- III. To implement the temperature controller using the bread board.
- IV. To design and fabricate PCB for the system to miniaturize the temperature controller

The outcome of this project is a prototype of a smart temperature controller.

## **1.3 Organization of the report**

The report has been written in 5 chapters. First chapter contains the Introduction about process control industry, objectives of the project etc. Second chapter consists of different types of technology used in industry to design smart control devices with their applications, types, standards etc. Third chapter of this report contains the whole project design descriptions with snapshots and procedures. Fourth chapter is the Results and discussions part of the project. Fifth chapter is the conclusion and future works demonstration.

# **Chapter 2**

## **Microcontrollers**

## **2.1 Introduction:**

Automatic control is the application of control theory for regulation of processes without direct human intervention. In the simplest type of an automatic control loop, a controller compares a measured value of a process with a desired set value, and processes the resulting error signal to change some input to the process, in such a way that the process stays at its set point despite disturbances. This closed-loop control is an application of negative feedback to a system. By using automatic control technique we can design a lots of smart control intelligent devices for our daily life. Now a days with the invention of new technology smart control devices has been created a big market in the world.

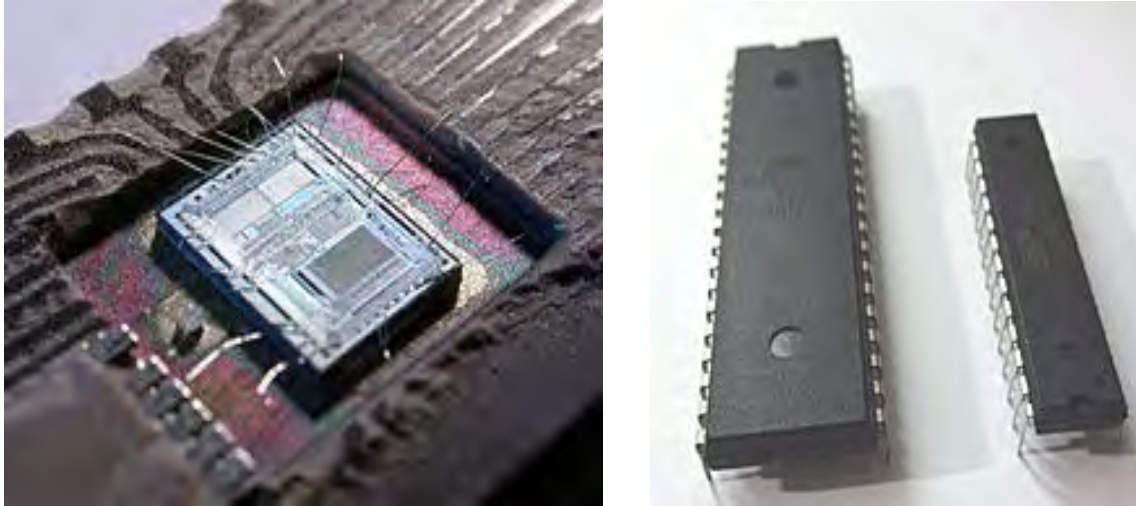
## **2.2 Microcontroller(uC):**

A microcontroller (sometimes abbreviated  $\mu\text{C}$ ,  $\text{uC}$  or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. This is a solitary chip microcomputer fabricated from VLSI fabrication. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications. So a micro controller is also known as embedded controller. The first microprocessor was the 4-bit Intel 4004 released in 1971, with the Intel 8008 and other more capable microprocessors becoming available over the next several years. However, both processors required external chips to implement a working system, raising total system cost, and making it impossible to economically computerize appliances. Today various types of microcontrollers are available in market with different word lengths such as 4bit, 8bit, 64bit and 128bit microcontrollers

Microcontrollers are used in automatically controlled products and devices, such as smart controller, automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, robots ,power tools, toys and other embedded systems. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and



input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.[12]



**Figure 2.1 :** (i)The die from an Intel 8742, an 8-bit microcontroller that includes a CPU running at 12 MHz, 128 bytes of RAM, 2048 bytes of EPROM, and I/O in the same chip. (ii) 2 ATmega microcontrollers.

Some microcontrollers may use four-bit words and operate at clock rate frequencies as low as 4 kHz, for low power consumption (single-digit milliwatts or microwatts). They will generally have the ability to retain functionality while waiting for an event such as a button press or other interrupt; power consumption while sleeping (CPU clock and most peripherals off) may be just nanowatts, making many of them well suited for long lasting battery applications. Other microcontrollers may serve performance-critical roles, where they may need to act more like a digital signal processor (DSP), with higher clock speeds and power consumption.

## **2.3 Different Types of controller**

Understanding the functions of different controllers is necessary for control engineer to find the problem in the automatic controls. An automatic controller is a device which measures the value of a variable quantity or condition and operates to correct or limit the deviation of this measured value from a selected reference. It includes both the measuring means and the controlling means.[14]

As a control engineer we should know about different controllers used and the function of different controllers and the action taken by them. We have already studied about control systems and focussed our attention on Pneumatic and Electronic Control Mechanisms. Now we will be discussing about Integral, Derivative and Proportional Controls in this article as follows:

### **2.3.1 On-Off Controller**

This is the simplest, cheapest and most reliable of controls. It is found to be most competent in handling variables associated with a batch process. Examples:

- Space temperature control in dryers and in air conditioning
- Level limits in tanks and reservoir etc

Such processes respond gradually and smoothly to the full on-off action of the controller and are defined as a "bath-tub " processes, because of large demand in contrast to supply.

### **2.3.2 Proportional Control**

This type of control mechanism is useful for controlling most variables which lack the stabilising effects of favourable storage capacity. Such "shower-bath" processes as they are more commonly

known, require that the transfer of material (fluid flow rate, gas pressure, liquid level) or energy , temperature and pressure must be kept continuously in step with the demand.

Proportional controllers are used in case of " How much error exists?". In this type of controller, it judges the error and acts accordingly.

**Output = Gain x Error**

Proportional controller output is directly proportional to the error signal so that it aims is to correct the error and attain the set point. Hence it is also defined as a control which is proportional to the deviation from set-point of the measured value.

It is represented by:

proportional controller - proportional to " $\Theta$ "

Typical examples of these are self acting **temperature controllers** and **conventional pressure reducing valves** .

### **2.3.3 Integral Controller**

Integral controllers are used in the case "How far is the error from the set value?". This types of controller judges the position of the set value and acts accordingly.

It is defined as the control action which changes at a rate proportional to the deviation, the rate of change being proportional to the deviation.

In practice this means that with measured value and set point coincident, the final control element may be in any position within the range.

It is represented by:

Integral controller - proportional to  $-\int \partial\Theta/\partial t$ .

### **2.3.4 Derivative Control**

A derivative controller is used in case of "How much time it takes to bring the system to the set-point"?. This is used wherever transfer lag problems are involved.

It is represented by:

**DERIVATIVE CONTROLLER** - proportional to  $-\partial\Theta/\partial t$

The derivative controller cannot be used independently. Either it should be used along with **proportional controller** or with **integral controller**. The set value is quickly attained by using the derivative controller.

Derivative action over-correction is good as an experienced operator will give manual over-correction before seeking a final average setting.

### **2.3.5 PID Controllers**

A proportional-integral-derivative controller (PID controller) is a control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an *error* value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the *error* by adjusting the process through use of a manipulated variable. It is mostly used in control engineering to design smart devices.

## **2.4 Sensors**

A sensor is a device that detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal; for example, a thermocouple converts temperature to an output voltage. But a mercury-in-glass thermometer is also a sensor; it converts the measured temperature into expansion and contraction of a liquid which can be read on a calibrated glass tube. All living organisms contain biological sensors with functions similar to those of the mechanical devices described. Most of these are specialized cells that are sensitive to, light, motion, temperature, magnetic fields, gravity, humidity, moisture, vibration, pressure, electrical fields, sound, and other physical aspects of the external environment [13].

## **2.5 Embedded system**

An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints. It is embedded as part of a complete device often including hardware and mechanical parts. By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs. Embedded systems control many devices in common use today. The

microprocessor based smart temperature controller is a embedded system which is used by process control industry.[11]

Modern embedded systems are often based on microcontrollers (i.e. CPUs with integrated memory or peripheral interfaces)but ordinary microprocessors (using external chips for memory and peripheral interface circuits) are also still common, especially in more complex systems. In either case, the processor(s) used may be types ranging from rather general purpose to very specialised in certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the digital signal processor (DSP).

### **2.5.1 Embedded system Applications**

Embedded systems are commonly found in consumer, cooking, industrial, automotive, medical, commercial and military applications.

Telecommunications systems employ numerous embedded systems from telephone switches for the network to cell phones at the end-user. Computer networking uses dedicated routers and network bridges to route data.

Consumer electronics include personal digital assistants (PDAs), mp3 players, mobile phones, videogame consoles, digital cameras, DVD players, GPS receivers, and printers. Household appliances, such as microwave ovens, washing machines and dishwashers, include embedded systems to provide flexibility, efficiency and features. Advanced HVAC systems use networked thermostats to more accurately and efficiently control temperature that can change by time of day and season. Home automation uses wired- and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance, etc., all of which use embedded devices for sensing and controlling.

Transportation systems from flight to automobiles increasingly use embedded systems. New airplanes contain advanced avionics such as inertial guidance systems and GPS receivers that also have considerable safety requirements. Various electric motors — brushless DC motors, induction motors and DC motors — use electric/electronic motor controllers. Automobiles, electric vehicles, and hybrid vehicles increasingly use embedded systems to maximize efficiency and reduce pollution. Other automotive safety systems include anti-lock braking system (ABS), Electronic Stability Control (ESC/ESP), traction control (TCS) and automatic four-wheel drive.

Medical equipment uses embedded systems for vital signs monitoring, electronic stethoscopes for amplifying sounds, and various medical imaging (PET, SPECT, CT, MRI) for non-invasive internal inspections

# **Chapter 3**

## **System Design**

### 3.1 Introduction

This chapter presents the techniques to implement the proposed temperature controller. Necessary hardware and software used in this project has also been described.

### 3.2 Requirement Gathering, Specification and Planning

Requirement Gathering, specification and planning are essential parts of any project and project management. During the process, different similar software and hardware are analyzed to design the controller. The Software and hardware requirements are also studied and specified in this phase. Different types of idea about the development are written up. Next is the planning phase, a plan is made to develop this circuit with requirement specification document. Figure 3.1 shows the flowcharts for implementation of a project.

The Project follows the Below **Flowchart**:

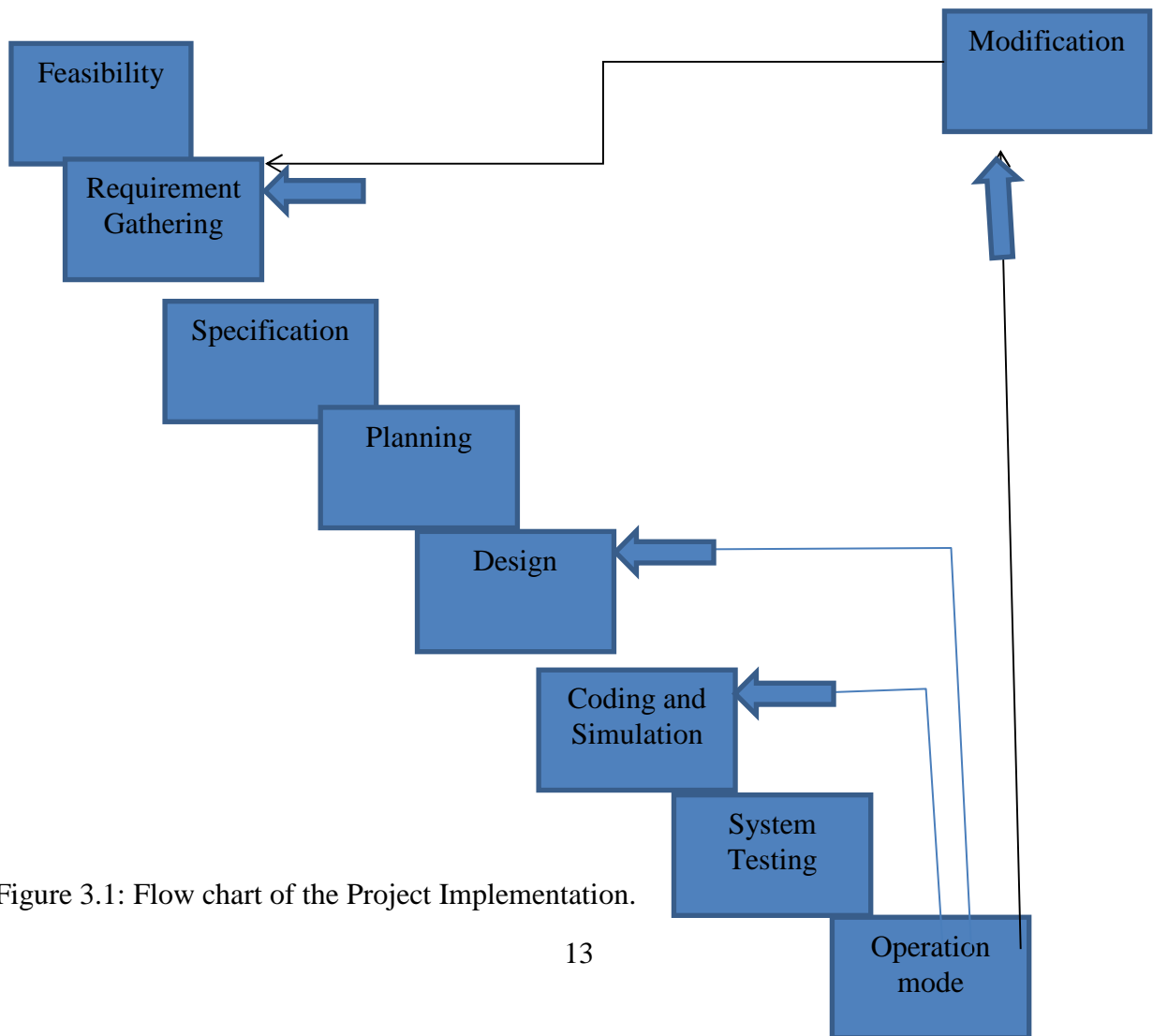


Figure 3.1: Flow chart of the Project Implementation.



### 3.3 Block diagram of the temperature controller

The block diagram of the smart temperature controller consists of several parts. The Set temperature is the input to comparator. The heater and cooler is connected with comparator via power switch so that, comparator can decide in which condition it choose to turn on /off the heater or cooler. The output of heater and cooler is connected with the total system and temperature sensor. There is a closed loop feedback between temperature sensor and comparator input.

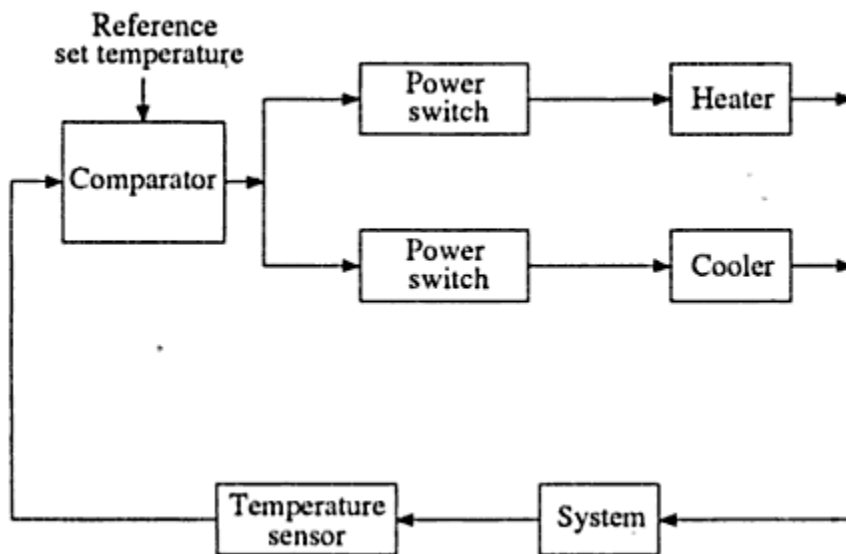


Figure 3.2 : Block diagram representation of the smart temperature controller.

### 3.4 Hardware Design Requirements List

This project will require following hardware:

- a. 1 x PIC16F877A Microcontroller
- b. 1 x LCD
- c. 1 x Buzzer
- d. 1 x Temperature Sensor (LM35)
- e. 1 x DC Brushless Fan
- f. 2 x BC 548 NPN Transistor
- g. 1 x LM7805 Voltage Regulator
- h. 1 x 6V AC Relay.
- i. Other related electronic components like capacitors, resistors, LEDs, Diodes etc

### 3.5 PIC16F876A (Microcontroller)

This powerful (200 nanosecond instruction execution) yet easy-to-program (only 35 single word instructions) CMOS FLASH-based 8-bit microcontroller packs Microchip's powerful PIC® architecture into an 40-pin package and is upwards compatible with the PIC16C5X, PIC12CXXX and PIC16C7X devices. [9]

#### **The PIC16F877A Main features:**

- 256 bytes of EEPROM data memory
- Self programming
- An ICD
- 2 Comparators
- 5 channels of 10-bit Analog-to-Digital (A/D) converter
- 2 capture/compare/PWM functions

- The synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or the 2-wire Inter-Integrated Circuit (I2C™) bus
- A Universal Asynchronous Receiver Transmitter (UART)
- Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle

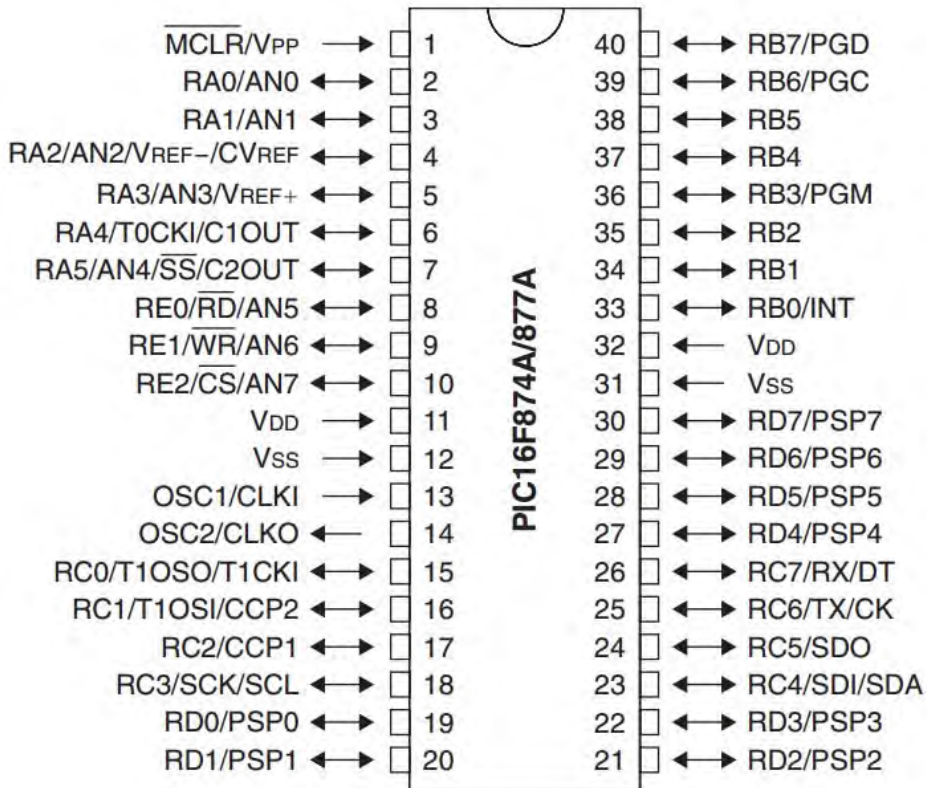


Fig 3.3: This figure shows the pin diagram for PIC16F877A. For more detail, please download the datasheet from microchip web site at: <http://www.microchip.com>

**Analog Features:**

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference

- Comparator outputs are externally accessible

### **Special Microcontroller Features:**

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins CMOS Technology:
- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

### **CMOS Technology:**

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

All of these features make it ideal for more advanced level A/D applications in automotive, industrial, appliances and consumer applications.

### 3.6 Temperature sensor (LM35)

The temperature sensor is a semiconductor device with a linear voltage-temperature relationship specified as  $10 \text{ mV}/^\circ\text{C}$ , i.e.  $V_0/T = 0.01$

Where,  $V_0$  is the sensor output voltage in volts and  $T$  is the temperature in  $^\circ\text{C}$ .

In this project, two LM35s are used for two difference temperature.  $V_s$  of the LM35s are given 5V and the  $V_{out}$  pins are connected to AN0 and AN1 (PIC16F877A) separately.

Since the sensor gain (average slope) of the LM35 is  $10 \text{ mV}/^\circ\text{C}$  and ADC has 10 bit ( $2^{10} \approx 1000$ ), so pin 5 ( $V_{ref}$ ) from PIC16F876A must be given for 1V by using the voltage divider concept. For preventing the offset, voltage for pin 5 ( $V_{ref}$ ) should be adjustable (using Preset). Fig 3.4 shows the Picture of LM35 Temperature sensor.

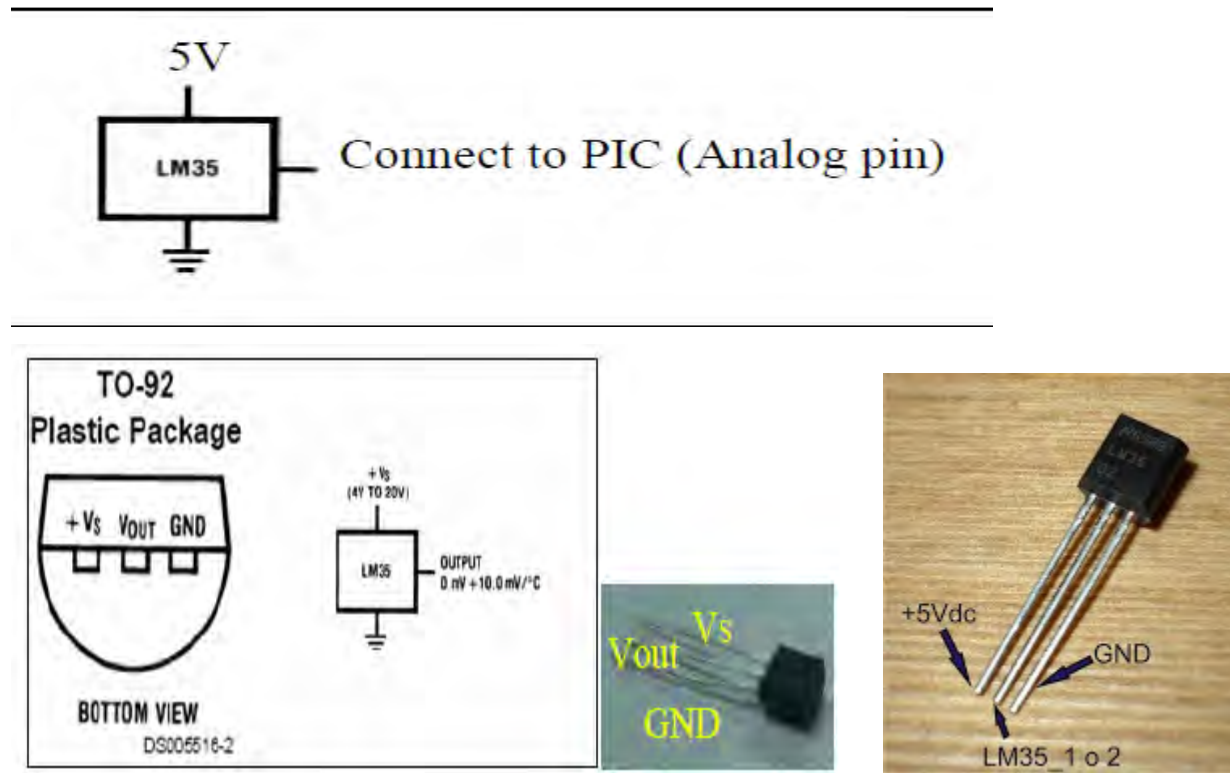


Fig 3.4: Picture of LM35 Temperature sensor.

### 3.7 Interfacing

#### **Interface PIC16F876A with LCD (2x16 character):**

LCD used in this project is JHD162A, for other type of LCD, please refer to its data sheet. Here we used the LCD as 4 bit mode to display the microcontroller instructions and the DB0-DB3 pin of LCD remain not connected.



Figure 3.5: The 16 header pin should be soldered to the LCD first.

The following table 3.1 shows the LCD (2x16 character) connection:

<b>Pin</b>	<b>Name</b>	<b>Pin function</b>	<b>Connection</b>
<b>1</b>	<b>VSS</b>	<b>Ground</b>	<b>GND</b>
<b>2</b>	<b>VCC</b>	<b>Positive supply for LCD</b>	<b>5V</b>
<b>3</b>	<b>VEE</b>	<b>Contrast adjust</b>	<b>Connected to a preset for LCD contrast adjusting</b>
<b>4</b>	<b>RS</b>	<b>Select register,select instruction or data register</b>	<b>RD1</b>
<b>5</b>	<b>R/W</b>	<b>Select read or write</b>	<b>RD2</b>
<b>6</b>	<b>E</b>	<b>Start Data read or write</b>	<b>RDO</b>
<b>7</b>	<b>DB0</b>	<b>Data bus pin</b>	<b>N/C</b>
<b>8</b>	<b>DB1</b>	<b>Data bus pin</b>	<b>N/C</b>
<b>9</b>	<b>DB2</b>	<b>Data bus pin</b>	<b>N/C</b>
<b>10</b>	<b>DB3</b>	<b>Data bus pin</b>	<b>N/C</b>
<b>11</b>	<b>DB4</b>	<b>Data bus pin</b>	<b>RD4</b>
<b>12</b>	<b>DB5</b>	<b>Data bus pin</b>	<b>RD5</b>
<b>13</b>	<b>DB6</b>	<b>Data bus pin</b>	<b>RD6</b>
<b>14</b>	<b>DB7</b>	<b>Data bus pin</b>	<b>RD7</b>
<b>15</b>	<b>LED+</b>	<b>Backlight Positive input</b>	<b>5V</b>
<b>16</b>	<b>LED-</b>	<b>Backlight Positive input</b>	<b>GND</b>

Table 3.1: LCD connection with PIC16F877A microcontroller pin

### **3.8 BC 548 NPN Transistor**

This device is designed for use as general purpose amplifiers and switches requiring collector currents to 300 mA. BC548 is used for controlling the DC brushless fan and 6V Relay with sufficient current. Here the microcontroller supplies the

required current to control the devices and perform switching. Following Figure shows the pin diagram of BC 548.

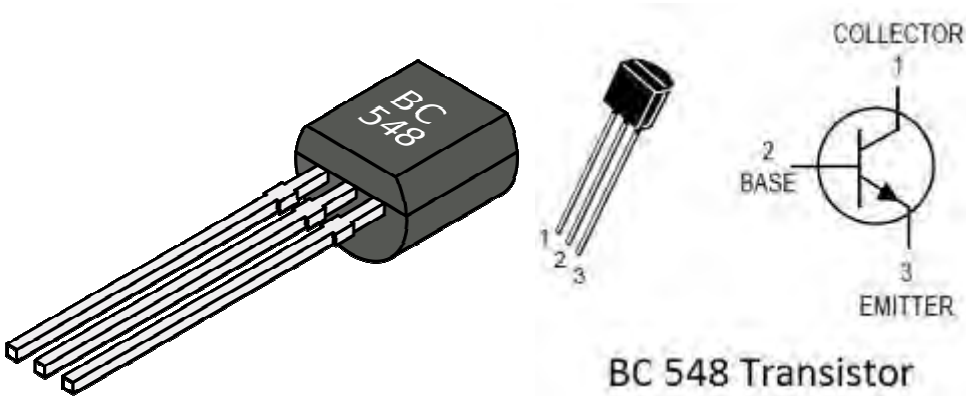


Figure 3.6: Picture of BC 548 Transistor

### 3.9 Static Relays

In my circuit I have used 6V static relay which is very necessary to switch on and off the heater. BC 548 transistor supplies the drive current to turn off and on the relay. We can use any type of AC load with this relay for certain conditions. Fig. 3.7 shows the pin connection diagram of a 6V static relay.

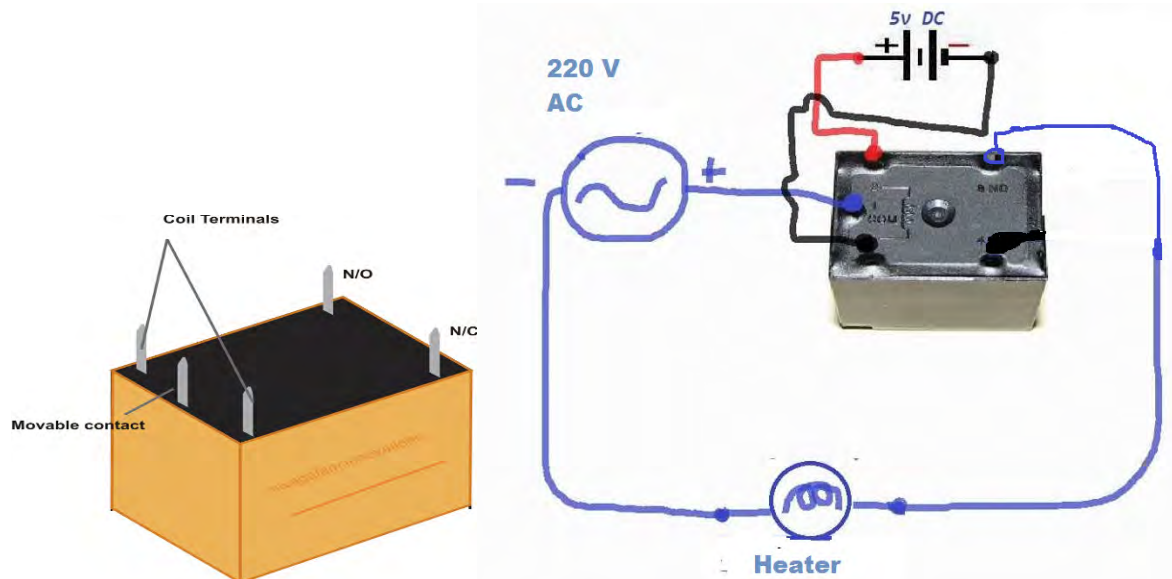


Figure 3.7: Picture of static relay.



### **3.10 Proteus**

Proteus is a Virtual System Modeling and circuit simulation application. Its a very powerful application to run microcontroller based simulation environment. The suite combines mixed mode SPICE circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller based designs. Proteus also has the ability to simulate the interaction between software running on a microcontroller and any analog or digital electronics connected to it.

### **3.11 CCS C Compiler**

CCS is an Intelligent and highly optimized C compiler contain Standard C operators and Built-in Function libraries that are specific to PIC registers, providing developers with a powerful tool for accessing device hardware features from the C language level. The CCS C compiler supports the Microchip PIC12x, PIC16x, PIC18x, and ds PIC devices. It supports everything a PIC compiler needs. Command-line compilers may be run from a DOS prompt or be integrated into MPLAB<sup>®</sup>, MPLAB<sup>®</sup> X IDE or other Microchip Third Party environments. Standard C preprocessors, operators and statements can be combined with hardware specific directives and CCS provided built-in functions and example libraries to quickly develop applications incorporating leading edge technologies such as capacitive touch, wireless and wired communication, motion and motor control and energy management.

### **3. 12 Circuit Design Description in Proteus Software**

Proteus is a very strong tool to design and simulate any complex circuit of electronics and embedded system. I have used Proteus 7.0 Professional to Design my circuit.

In this design, PIC16F877A is connected with a 16\*2 LCD display at 4 bit connection mode, a LM35 Temperature controller connected at analog input of the uC pin AN0. Two BC 548 NPN transistor is connected via 100 ohm resistor at uC pin AN1 and AN3. Q1 drives the DC fan and Q2 drives the Relay. Here, LED 1 and LED 2 represents the on off condition of two transistor. A

Buzzer is connected at RB5 port and make beep when get high logic at RB5 port. LCD is connected to RD4 to RD7 pin of microcontroller. I have showed a table of LCD connection at interfacing description table 3.1. The power and clock connection of microcontroller described at the chapter 4. At First I draw the circuit in Proteus and then imported the HEX file into microcontroller. Finally I simulated the design.

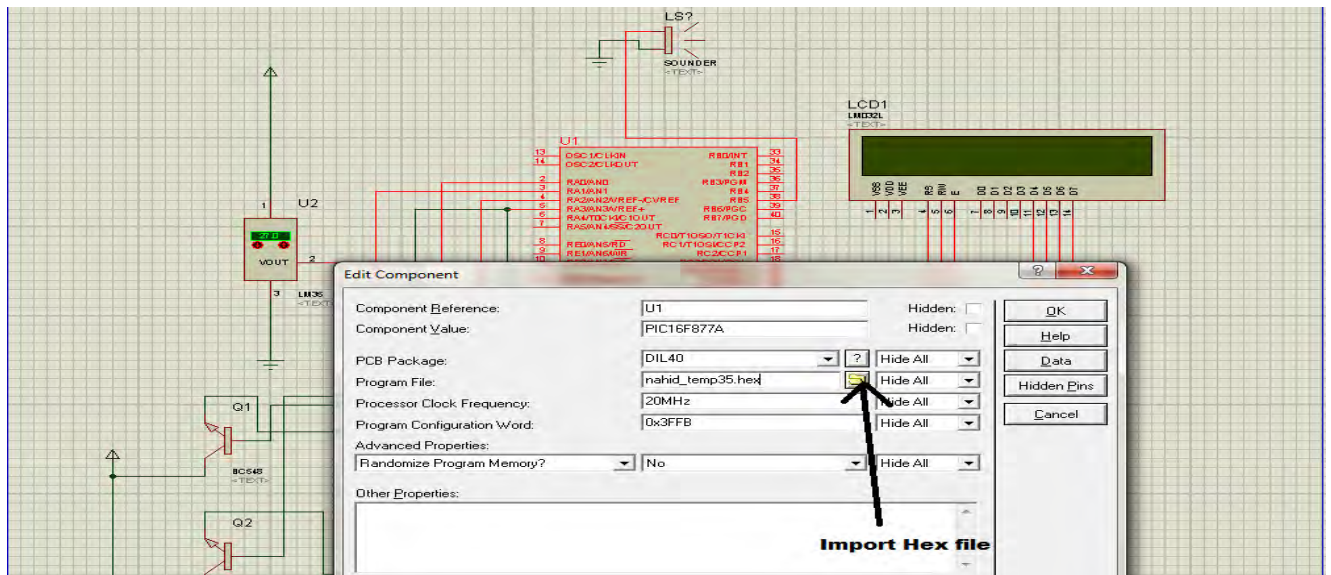


Fig 3.8 : HEX file (program file) import at Proteus window.

### 3.13 Microcontroller Programming ,coding and Hex File Generation using CCS C Compiler

I have written code for PIC16F877A microcontroller in CCS C compiler. I have used PIC wizard to create temp.h file for automatic include the header file <16F877A.h> for PIC.

After that, I have written code for my Proteus Design which is in Appendix A & Appendix B. I have saved the written code by name temp.c. Fig 3.9 shows the flowchart of the Smart temperature controller that has been used to write code in C Programming Languages.

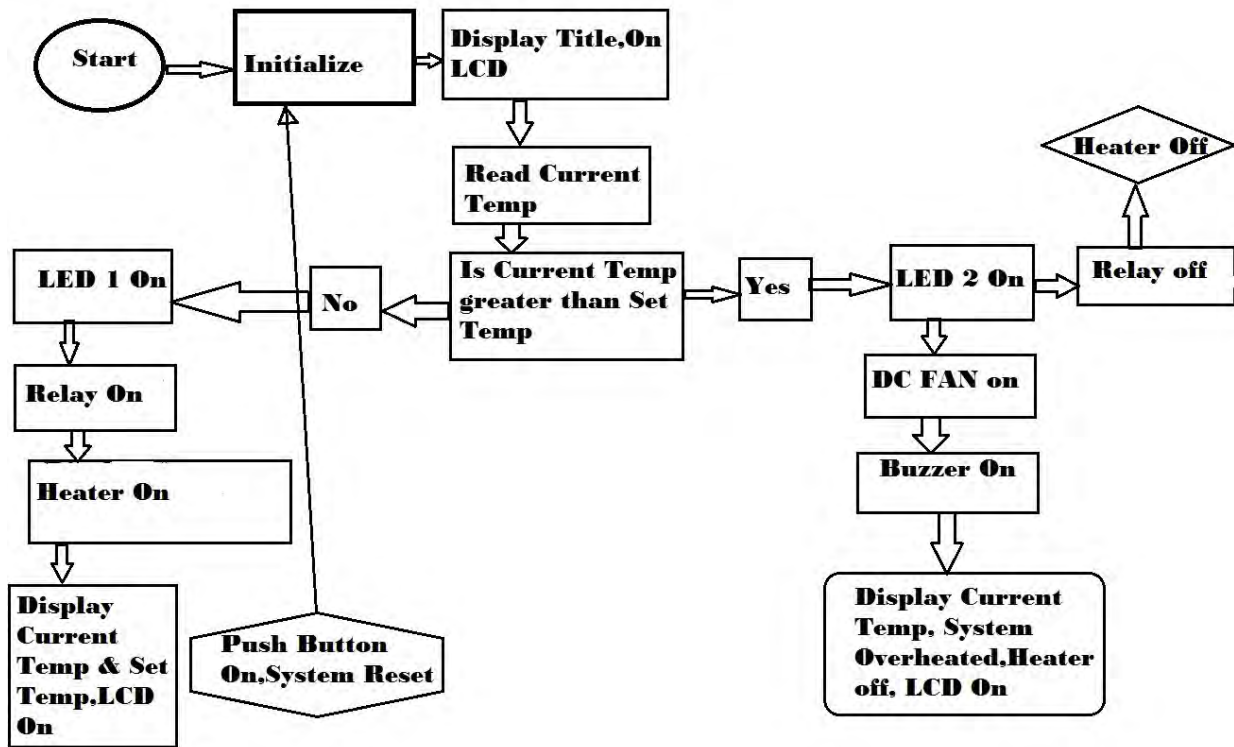


Fig 3.9: Flow Chart of the Smart temperature Controller Algorithm.

I have used delay for 20 MHz clock and also defined the port which I need to use in my design. In my code I have used the temperature maximum value as 500 degree Celsius so that higher range temperature sensor can be used according to set temperature. But In my hardware design I used LM35 DZ temperature sensor. The LM35 is rated to operate over a  $-55^{\circ}\text{C}$  to  $+150^{\circ}\text{C}$  temperature range. After completion of code , I compiled the code and generated the HEX file which will be used at our Proteus design simulation. Fig 3.10 shows the different file list of my design.

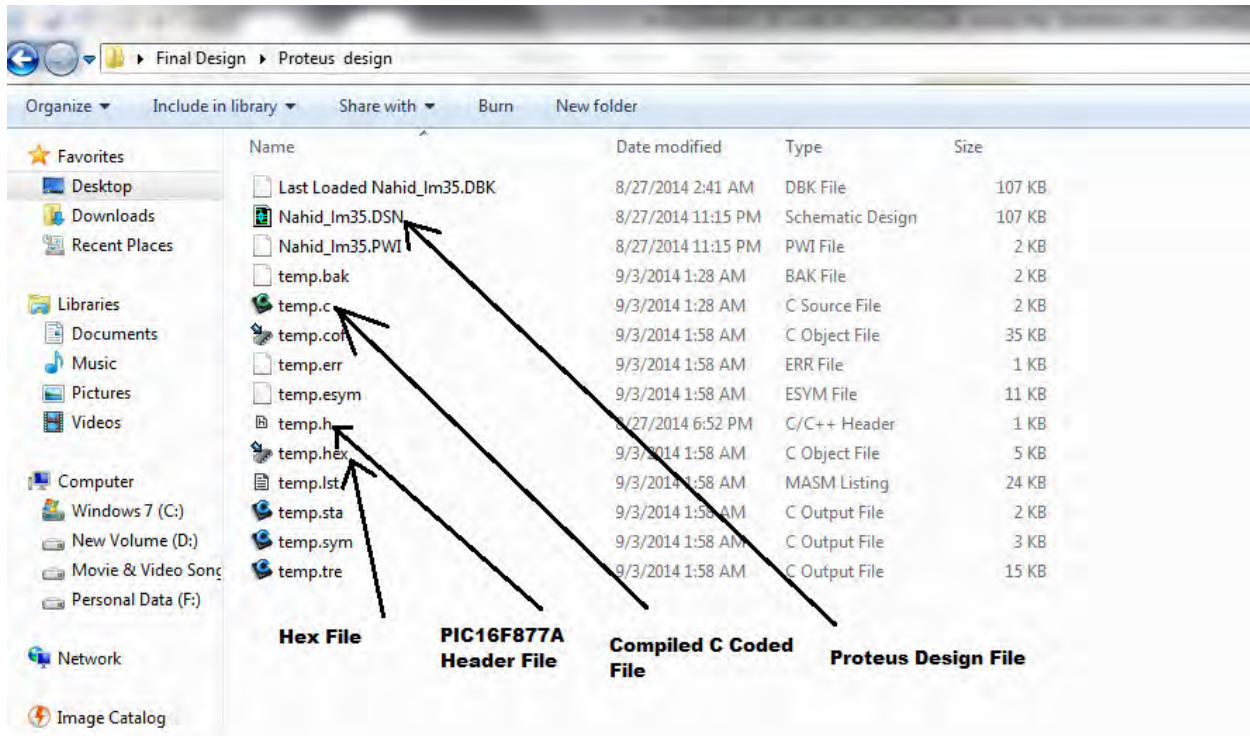


Fig 3.10: Hex file and other simulation file list of the Design with file extension.

### 3.14 Hex File Burning in PIC16F877A Using PIC Kit 2 Programmer

A PIC Kit 2 programmer has been used to burn the 40 pin PIC16F877A microcontroller. For PIC16 series external Vdd supply required to burn the chip. At first PIC has been installed into the programmer kit. Next vdd connection used from any external board. After that MPLAB 8.70 has been used to program the microcontroller by importing the hex file. Here We can also program the PIC using PIC Kit 2 programmer software. The screenshot has been taken for understanding the procedure. Fig 3.11 and 3.12 shows the Programming snapshots.

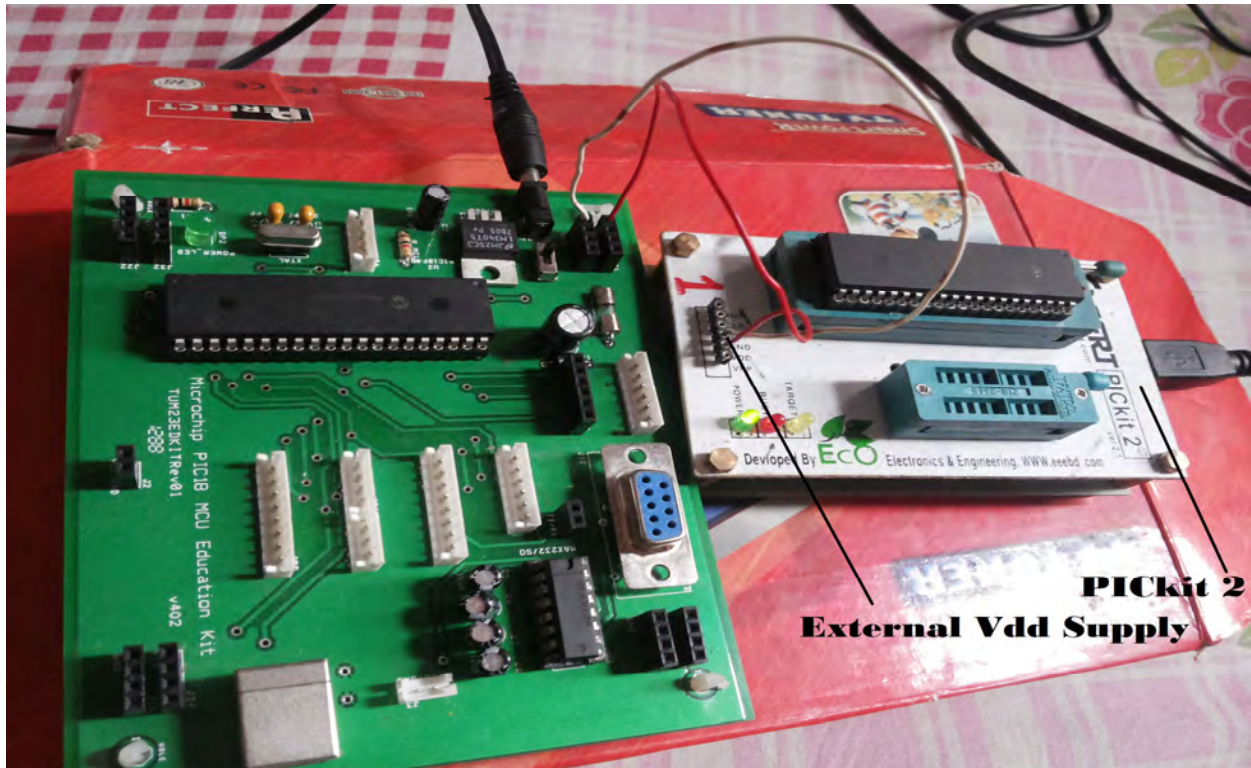


Fig 3.11: PICF877A installed in PICkit2 Programmer with external vdd connection.

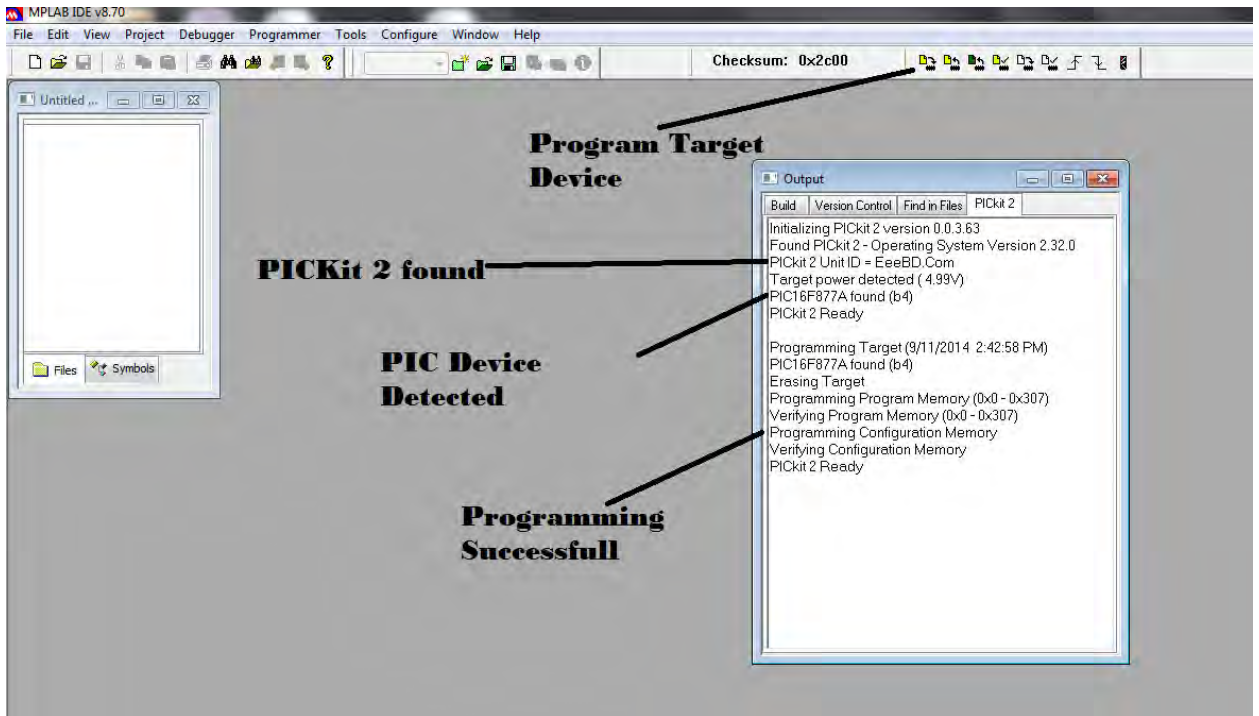


Figure 3.12: PIC programming using MPLAB v8.70

### **3.15 PCB Design**

A printed circuit board (PCB) mechanically supports and electrically connects electronic components using conductive tracks, pads and other features etched from copper sheets laminated onto a non-conductive substrate. PCBs can be single sided (one copper layer), double sided (two copper layers) or multi-layer. Conductors on different layers are connected with plated-through holes called vias. Advanced PCBs may contain components - capacitors, resistors or active devices - embedded in the substrate.[10]

Printed circuit boards are used in all but the simplest electronic products. Alternatives to PCBs include wire wrap and point-to-point construction. PCBs require the additional design effort to lay out the circuit but manufacturing and assembly can be automated. Manufacturing circuits with PCBs is cheaper and faster than with other wiring methods as components are mounted and wired with one single part. Furthermore, operator wiring errors are eliminated.

When the board has only copper connections and no embedded components it is more correctly called a printed wiring board (PWB) or etched wiring board. Although more accurate, the term printed wiring board has fallen into disuse. A PCB populated with electronic components is called a printed circuit assembly (PCA), printed circuit board assembly or PCB assembly (PCBA). The IPC preferred term for assembled boards is circuit card assembly (CCA), for assembled backplanes it is backplane assemblies. The term PCB is used informally both for bare and assembled boards.

#### **The general progression for a commercial printed circuit board design steps:**

1. Schematic capture through an electronic design automation tool.
2. Card dimensions and template are decided based on required circuitry and case of the PCB. Determine the fixed components and heat sinks if required.

3. Deciding stack layers of the PCB. 1 to 12 layers or more depending on design complexity. Ground plane and power plane are decided. Signal planes where signals are routed are in top layer as well as internal layers.
4. Line impedance determination using dielectric layer thickness, routing copper thickness and trace-width. Trace separation also taken into account in case of differential signals. Microstrip, stripline or dual stripline can be used to route signals.
5. Placement of the components. Thermal considerations and geometry are taken into account. Vias and lands are marked.
6. Routing the signal traces. For optimal EMI performance high frequency signals are routed in internal layers between power or ground planes as power planes behave as ground for AC.
7. Gerber file generation for manufacturing.

### **PCB design Procedures in Orcad**

ORCAD Layout Plus software has been used for PCB design. At first I have created new layout in orcad layout plus. Then a black screen was appeared. After that I have selected all component and place the components on design layer. Next was the wire connection part. I have connected all components by wire as per my design. Finally I have routed the complete layout. For design understanding named all the components at the PCB board.

### **3.16 Hardware Implementation**

Hardware construction is the major challenge of a system. At this stage every information of our design need to know before hardware construction. Because, in software environment we don't need to worry about every power connection and circuit elements familiarity. But in real life hardware design , this is the hard part of any major project. At first we construct the whole circuit in Bread Board and finally we will design PCB for better performance of the system.

Now a days in Bangladesh we can buy our all necessary circuit elements from Dhaka Stadium Market, Mirpur Shah Ali Market, Patuatuli Market etc. Every elements like Microcontroller, Transistor, DC Fan, LED, Diodes, capacitor, resistor, Buzzer, Relay has fixed market price at most of the places. The price does not vary a lot. So we can collect every element at a cheap price.



# **Chapter 4**

## **Results and Discussion**

## 4.1 Introduction

This chapter represents the soft design of the circuit, prototype of the temperature controller implemented using the hardware components on the breadboard and PCB. Test results of the system has also been presented.

## 4.2 Schematic Design of the Temperature controller

At this part Schematic Design of the Temperature controller has been captured. The power supply design for the whole circuit, power connection of microcontroller, main simulation circuit with schematics has been discussed as a part of the result of the project.

### 4.2.1 Power Supply

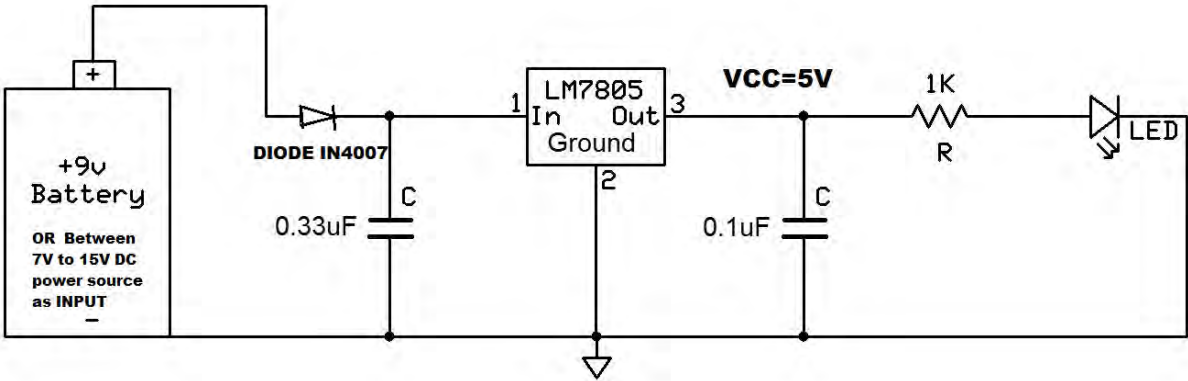


Fig 4.1: Power Supply circuit diagram for the controller.

For this project, the voltage range of power source could be given for this circuit board is between 7V and 15V. Higher input voltage will produce more heat at LM7805 voltage regulator. Typical voltage is 9V. Anyhow, LM7805 will still generate some heat at 12V. We can use a 9V Battery or AC to DC power adaptor as input for the circuit. Normally AC to DC adaptor can be plugged to 'Adaptor' type connector. LM7805 (1A maximum) will regulate the given voltage to

5V (VCC) for supplying to the PIC16F877A and pull-up the push button (input). The purpose of using diode is for circuit protection in case the polarity of the power source is incorrect. Two Capacitor is used to stabilize the voltage input and output of the LM7805. A green LED (small) as power indicator shown in the circuit diagram.

### 4.2.2 Push Button and Vdd connection of microcontroller

Here SW is the push/reset button in the system and XT is the crystal to produce clock at PIC input clock. XT may be from 4Mhz to 20 MHz crystal. I have used 20Mhz crystal in my circuit. Below is the power and push button connection of the microcontroller of my design which is not included in Proteus simulation. Fig 4.2 Shows the power and push button connection.

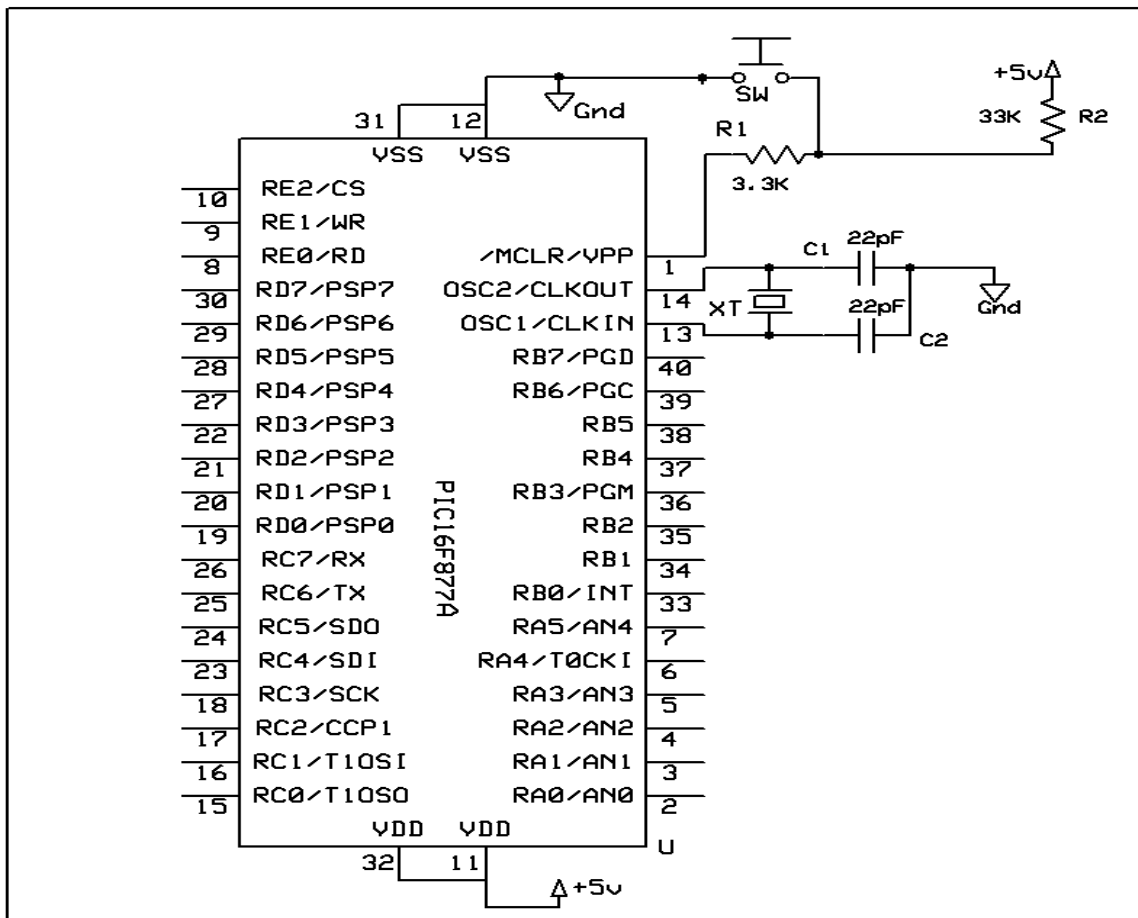


Fig 4.2: Vdd, GND and Push button connection of Microcontroller.





## 4.4 Bread board

Complete circuit circuit has been constructed in breadboard and successfully runs on the tested environment. Figure 4.5 and 4.6 shows the proposed temperature controller implemented on the bread board.

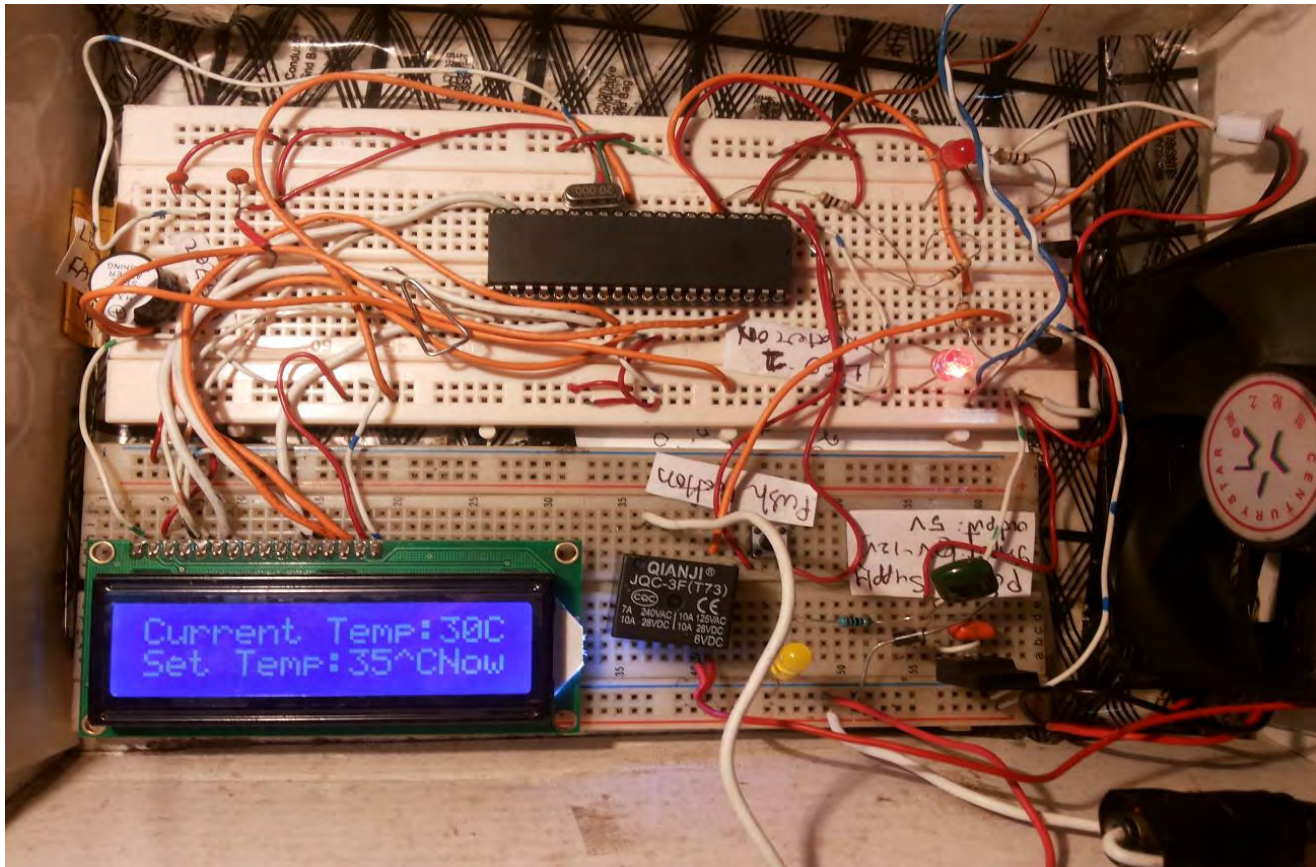


Fig 4.5: Snap of the circuit on bread board



## 4.5 PCB

I have designed PCB at Orcad Layout plus software and took the snapshot of the expected PCB design layout. Figure 4.6 shows the PCB Design of the system .

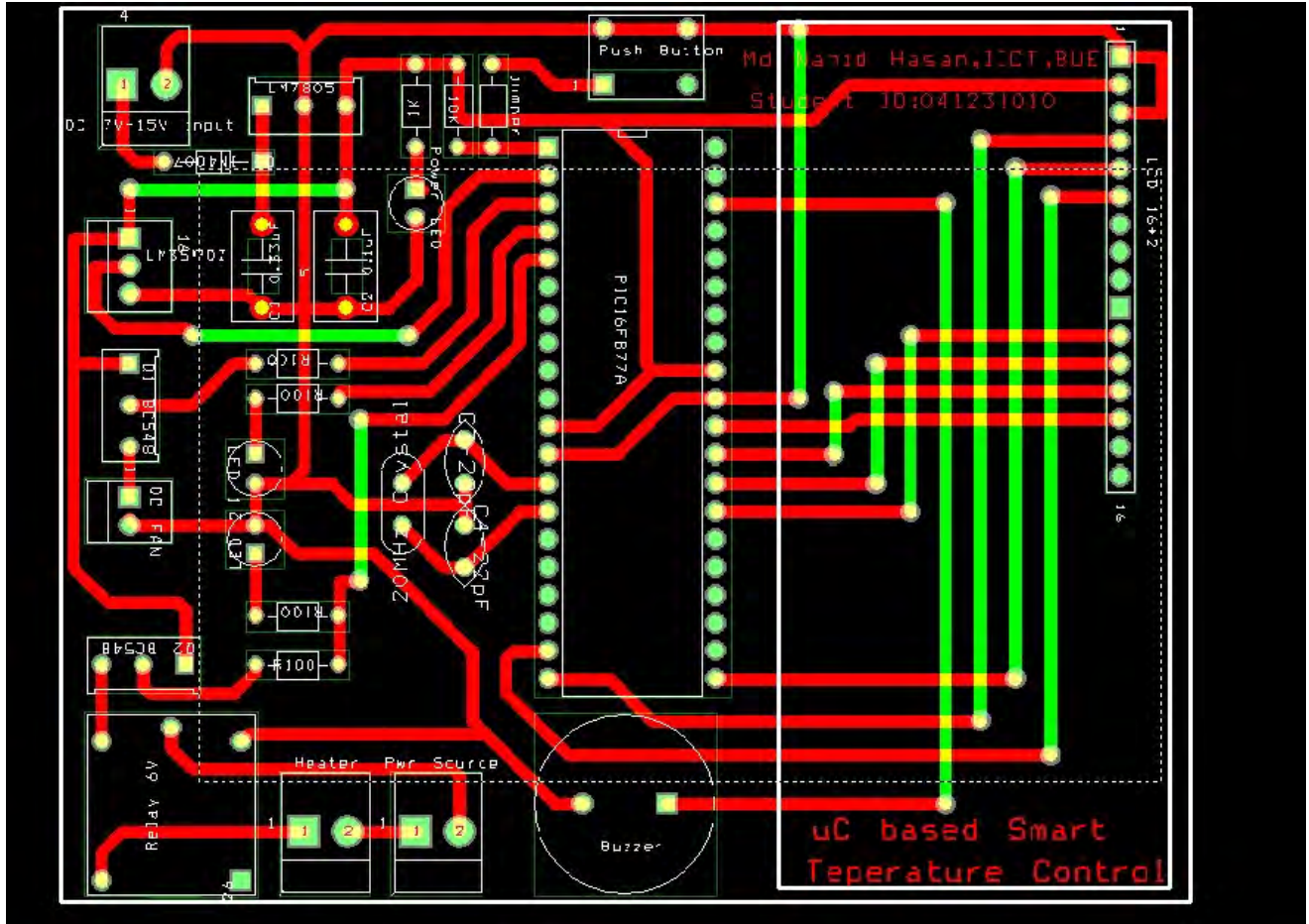


Fig 4.7: Final PCB Design Snapshot using Orcad Layout plus.



# **Chapter 5**

## **Conclusion**

## **5.1 Conclusion**

Automation is a demand in this era of information and communication technology. Especially in industrial environment it is very much necessary for continuous monitoring of different process control parameters in order to keep the quality of the production and level of production in an acceptable range. Temperature is one of such a process parameters. It has to be monitored continuously and failure of which may destroy whole production process and may cause accidents which in turn produces losses and damages. This project is an attempt for developing smart temperature controller using micro controller based system. The proposed controller has been simulated using the Proteus software to test the desired function. The desired test result leads to the implementation of the system using the necessary hardware. The system has been first implemented in the bread board where it shows the proper functionality. Then the PCB of system has been designed and fabricated and then implemented there in order to miniaturize it. The developed system can be enhanced and commercialized in future.

## **5.2 Future Works**

The author recommends the following suggestion for future research and development on the proposed system. In the current system, the set point of the temperature is fixed whereas in industry the set point is necessary to vary from process to process. So in the proposed developed system can be enhanced by making the set point programmable.

## References:

- [1] Stepopoulos, G., “Chemical Process Control: An Introduction to Theory & Practice”, Prentice Hall Of India.
- [2] Johnson, C. D., “Process Control Instrumentation Technology”, Seventh Edition, Pearson Education, New Delhi 2003.
- [3] [http://en.wikipedia.org/wiki/Process\\_control](http://en.wikipedia.org/wiki/Process_control); Last accessed on 01.05.2014
- [4] Astrom, K. J., and Hagglund, T., “PID controllers: theory, design and tuning”, USA: Instrument Society of America, 1995.
- [5] Ejiofor Virginia Ebere (PhD)<sup>1</sup>, Oladipo Onaolapo Francisca (PhD)<sup>2</sup> (2013), ‘Microcontroller based Automatic Water level Control System’, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 1, Iss. 6, ISSN 2320-9801
- [6] <http://fyptemperaturecontrolsystem.blogspot.com/> ; Last accessed on 05.02.2014.
- [7] <http://www.chemeng.queensu.ca/courses/CHEE218/projects/TemperatureControl/index.html>;  
Last accessed on 11.05.2014.
- [8] <http://www.electro-tech-online.com/threads/pid-temperature-controller-pic16f876.21513>; Last accessed on 10.06.2014.

[9] PIC data Sheet from <http://www.microchip.com>; Last accessed on 15.07.2014.

[10] [http://en.wikipedia.org/wiki/Printed\\_circuit\\_board](http://en.wikipedia.org/wiki/Printed_circuit_board); Last accessed on 25.08.2014.

[11] [http://en.wikipedia.org/wiki/Embedded\\_system](http://en.wikipedia.org/wiki/Embedded_system); Last accessed on 25.08.2014.

[12] <http://www.electronicshub.org/microcontrollers/>; Last accessed on 30.08.2014.

[13] <http://www.engineersgarage.com/articles/sensors>; Last accessed on 01.09.2014.

[14] <http://www.brighthubengineering.com/diy-electronics-devices/54829-controllers-and-their-characteristics/>; Last accessed on 01.09.2014.

## Appendix A Software Code written in CCS C compiler

```
#!/Project Title:Microcontroller Based Smart Temperature Control System
#!/Project Designed By Md.Nahid Hasan, IICT, BUET-DHAKA. Student ID:0412311010
#include "C:\Users\John\Desktop\Final Design\ temp.h"
#include <lcd.c>
#define buzzer RB5

int temperatura;

void main()
{
    setup_adc_ports(RA0_ANALOG);//LM35 Input
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_COUNTERS(RTCC_internal.rtcc_div_1);//Division Mark
    set_adc_channel(0);

    /*******

    lcd_init();
    lcd_gotoxy(1,1);
    printf(lcd_putc,"Smart Temp Controller");
    delay_ms(350);
    lcd_gotoxy(1,2);
    printf(lcd_putc,"Design by:Nahid.");
    delay_ms(2000);
    lcd_init();
```

```

lcd_init();
lcd_gotoxy(1,1);
printf(lcd_putc,"IICT, BUET.");
delay_ms(350);
lcd_gotoxy(1,2);
printf(lcd_putc,"St.ID:0412311010");
delay_ms(2000);
lcd_init();

while(true)
{
temperatura=(int)read_adc()/2;
*****Degree Celsius

lcd_gotoxy(1,1);
printf(lcd_putc,"Current Temp:");
lcd_gotoxy(14,1);
printf(lcd_putc,"%dC",Temperatura);
delay_ms(200);

***** If Temperature is greature than 35 degree Celsius
if(temperatura<=500&&temperatura>=35)
{
output_high(pin_a1);
output_high(pin_a2);
output_low(pin_a3);
output_high(pin_b5);

```

```
lcd_gotoxy(1,2);
printf(lcd_putc,"System Overheated");
delay_ms(1000);
lcd_init();
lcd_gotoxy(1,2);
printf(lcd_putc,"**Heater Off Now");
delay_ms(200);
}

//*****If Temperature less than 35 degree Celsius
else
{
output_low(pin_a1);
output_low(pin_a2);
output_high(pin_a3);
output_low(pin_b5);

lcd_gotoxy(1,2);
printf(lcd_putc,"Set Temp:35^C");
delay_ms(200);
}
}
```

## Appendix B PIC header file code.

The project file(temp.h) contains the below code:

```
#include <16F877A.h>
#device adc=10

#FUSES NOWDT           //No Watch Dog Timer
#FUSES HS              //High speed Osc (> 4mhz for PCM/PCH) (>10mhz for PCD)
#FUSES NOPUT          //No Power Up Timer
#FUSES NOPROTECT      //Code not protected from reading
#FUSES NODEBUG        //No Debug mode for ICD
#FUSES NOBROWNOUT     //No brownout reset
#FUSES NOLVP          //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES NOCPD          //No EE protection
#FUSES NOWRT          //Program memory not write protected

#use delay(clock=20000000)
```