DEVELOPMET OF EMBEDDED SYSTEM FOR TEMPERATURE MONITORING AND CONTROL

A DEISSERTATION

Submitted in the partial fulfilment of the

Requirements for the award of the degree

Or

MASTER OF TECHNOLOGY

In

ELECTRICAL ENGINEERING

(With specification in Instrumentation & signal processing)

By

MUSA HUSSEN

(1728005)

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY

ROORKEE-247667(INDIA)

JULY, 2019

DECLARATION

I hereby declare that the work carried out in this seminar titled **"Development of Embedded system for temperature monitoring and control"** is on behalf of partial fulfilment of the requirement for the award of the degree of **Master of Technology in Electrical** with specialization in **Instrumentation and Signal Processing**, Submitted to department of Electrical Engineering, Indian Institute of Technology Roorkee, India, Under the supervision and guidance of **DR. R S ANAND ,**department of Electrical Engineering, IIT Roorkee, India. I have not submitted the matter embodied in this report for the award of any other degree or diploma in this institute or any other institute.

Date: June 2019

Place: Roorkee

Place: **Roorkee** **MUSA HUSSEN**

CERTIFICATION

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Dr. R S Anand

Associate Professor

Department of Electrical Engineering

IIT Roorkee.

ACKNOWLEDGEMENT

First of all, I am indebted to the GOD ALMIGHTY for giving me an opportunity to excel in my efforts to complete this dissertation.

I wish to express my gratitude to wards my guide **DR. R S ANAND** to grant me the opportunity to work on this excellent field of research. I also thank him for being constant source of inspiration and motivation. I wish to thank him for his guidance without which I would not be able to finish this dissertation successfully.

I am also grateful to all faculty members and staff of Electrical Engineering department, IIT Roorkee.

MUSA HUSSEN

ABSTRACT

The project is about the development of an embedded system for temperature monitoring and control of a bathroom. The system is done using Atmega16/32 development board. It measures the temperature and monitors by comparing the desired temperature value which is entered from the user. 4X4 keypad in interfaced for this purpose. The user will enter the value of temperature at which the water should be heated so that the value will be compared to the actual temperature of water continuously. The values are displayed on LCD every 20 second. Liquid crystal display is interfaced to the system for displaying the measured temperature values. The user can easily watch the temperature values sensed. RTD Pt 100 is connected to PORTC of the microcontroller for sensing the value of water temperature in water bath. The resistance temperature detector (RTD) is inserted to water so that it will sense the present temperature value. Depending on the input temperature value we will have options. If the actual temperature value of water is below the desired it will continue reading. When water temperature value is equivalent to the desired temperature, it will display message off. In this case the switch will be off and we will get the water with desired temperature value.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

CHAPTER ONE

INTRODUCTION

Microcontrollers are general integrated circuit, which have microprocessors, input and output devices storage units (memories) and registers. They are programmable devices in which based on the program they will do intelligent calculations and algorithmic operations. Microcontrollers now a days are used in varies devices such as remote control devices, telephones, televisions, refrigerators, in digital camera systems, in telephone systems, washing machines, in microwaves and other embedded system devices. Based on the application, Microcontrollers can have different types of memory in order to store data and programs. One is Flash memory that will be reprogrammed many times for different applications and the other is Read only memory (RAM) which will be programmed at the company level during manufacturing. Compilers are used to compile the high level program that is written by the user and converted to a language that a computer system can understand it. This is a machine language which are stored in the form of 0's and 1's. The microcontroller will execute the program, and takes input from peripheral devices based on the program it will make some calculations and operations. This is done based on the algorithm programmed by user. The output data will be given to displaying and recording devices. Microcontrollers are categorized as eight bit, sixteen bit and 32 bit based on the data bit they have in a one execution [1]. The other main parts of an embedded system application are sensors. Sensors are devices that can detect and responds to some types of input given from the physical world. The input may be motion heat, temperature, pressure, light or any other phenomena. Temperature sensors are highly used in embedded system applications. There are a number of Temperature sensors depending on their specific applications. In this project we are using RTD (resistance temperature detector) in which its resistance varies depending on the temperature value change.

SIGNIFICANCE AND PURPOSE OF PROJECT

The main purpose of the project is design and implementation of monitoring system to monitor the water heat temperature in a bathroom. In the project the temperature sensor senses the temperature of water in a bathroom and provides an analog o/p that will be given to a controller. The microcontroller changes input value by the help of digital to analog converter and monitors the temperature limit of water temperature in a bathroom. Microcontrollers controls the amount of temperature limit based on the given fixed value which is entered by the user. Using Atmega16 microcontroller we can control peripheral devices that are attached to its input output pins and display measured values. In this project it will measure the temperature and display it on liquid crystal display (LCD). Atmel studio compiler is used to verify and compile the algorithm written by c programming language, and it will generate hex file which is loaded to the microcontroller flash memory with USBasp.

Temperature controlling and monitor using embedded system application is significant in today's technology. It's used in many applications. It's used in industry, laboratories and in medical services. In these area very significant to control the water temperature to its optimal heating stage. In this application it is very significant to preserve the minimum and maximum temperature of water in a bathroom. Based on the Minimum and maximum values the heater will be on and off. The complete embedded control system will save power by controlling the temperature limit that at which it should stop heating.

CHAPTER TWO

2.1 ATMEGA16 MICROCONTROLLER ARCHITECTURES

The ATmega16 microcontroller is eight Bit RISC (Reduced Instruction Computing) type of controller which is manufactured in Atmel Company. ATmega16 microcontrollers have 32 GPR registers of 8 bit, two 8 bit timer/counter with separate prescaler, one 16 bit timer/counter with separate prescaler, 40 pin PDIP and 131 instructions. ATmega16 have 16KB self-programmable non-volatile memory and 1 KB of sram. ATmega16 also have 512 byte of Read Only Memory and programmable read only memory (EEPROM). It has six Pulse Width modulations (P W M) channel, six channel of 10 bit analog to digital converter (ADC), Programmable Universal Synchronous Asynchronous Receiver and Transmitter (USART) interface. It have also master and slave serial peripheral interface (SPI). In this project I used 40 PDIP type of ATmega16 microcontroller which is interfaced on board [2].

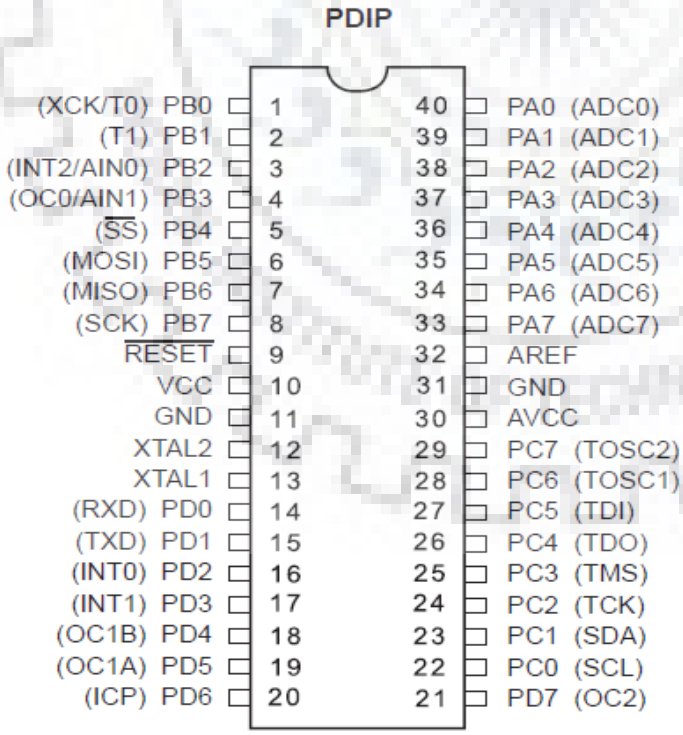


Figure 2. 1Pin diagram of ATmega16 microcontroller

VCC:

VCC is a digital voltage supply for microcontroller that is +5V.

GND:

GND is ground pin of ATmega16 microcontroller.

PORT B (PB7:PB0):

PORT B is a bi directional input output port. All ports of a microcontroller have multiple function based on the application we are going to use. PB6 pin is used for exterior clock input to ATmega16 and may be used as clock oscillator for the chip in pin1. PORTB7 is used as oscillation for pin 2. In the project we use PORTB for interfacing with liquid crystal display.

PORTC (PC5:PC0)

PORT C is eight Bit bi directional I/O port with internal pullup registers. These pins of microcontroller working as A to D converter pin input. In some areas port 4, port 5 may be applicable for two wire Serial Bus (I2C). The PORTC output buffers contain regular derive characteristics which are high sink and sources capabilities. PORT C pins are tri stated when a reset condition becoming active when the clock isn't working. When we enable JTAG interface, the pull up resistors on Port C5, Port C3 and Port C2 will be energized ever there is occurrence or reset [2].

PORT D (PD7:PD0):

PORT B is an eight Bit two directional input output port having internal pull up resistor. These are selected for every bit. The PORT D O/P buffer have same drive characteristics with high sources capabilities and sources capabilities. PORTD pins as input are externally pulled low will source current when the pull up resisters will be active. PD0 and PD1 are Universal Synchronous and Asynchronous transmitter (USART) I/O pins correspondingly.

Avcc

Avcc is the supply voltage pin for PORTA and the A to D converter (ADC). Avcc must be connected from external to Vcc, even if when A to D is not used.

Aref:

AREF is an analog reference pin in analog to digital converter.

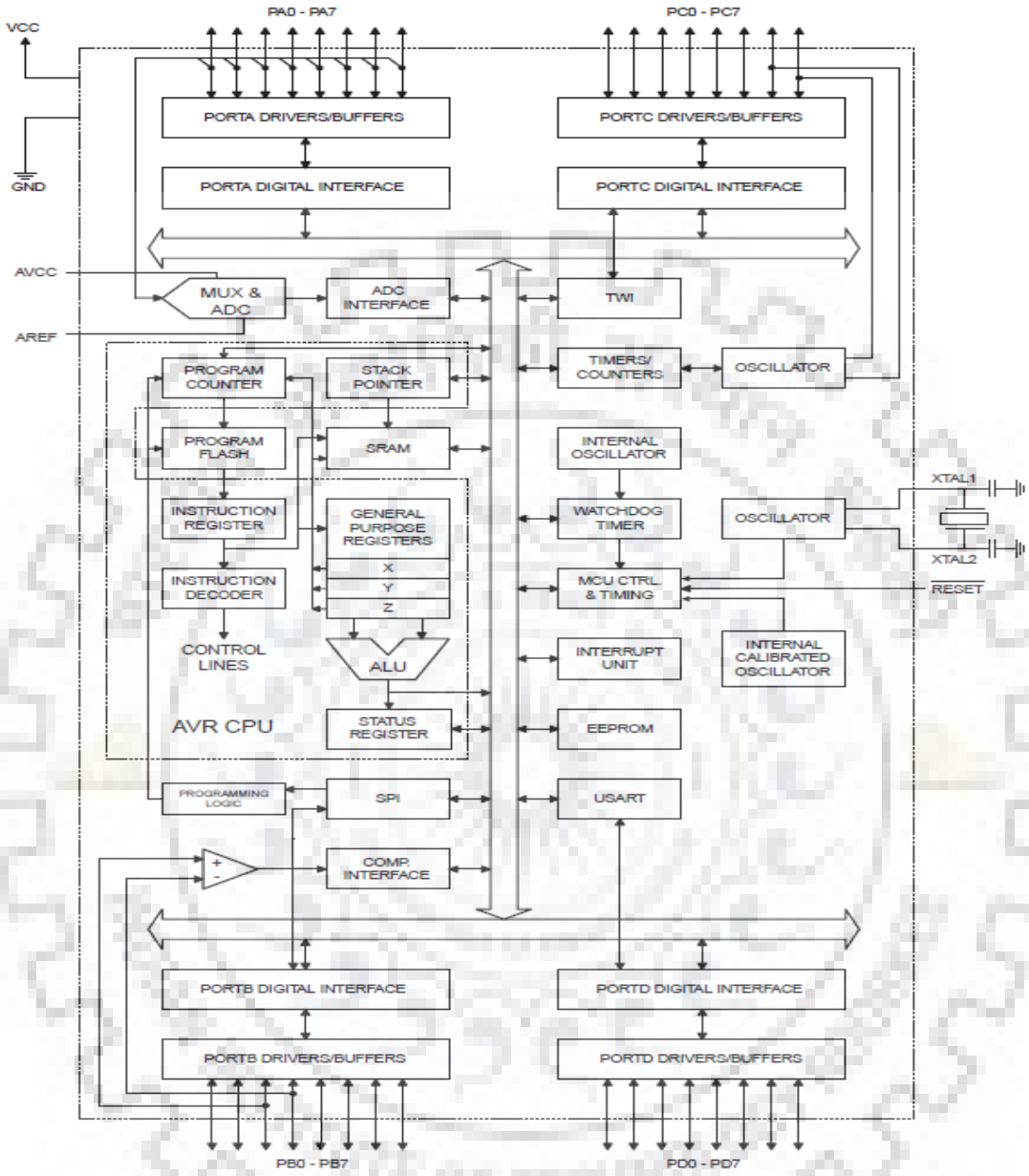


Figure 2. 2 The block diagram for ATmega16 [3]

16. BIT TIMER/COUNTER

The 16 Bit timer/counter will be given accurate program manipulation timing wave production, and for the measurement of signal timing. The more good feature of this unit are it have two independent output comparison unit, one input compare unit, input capture noise canceler, frequency generator, variable PWM, external event counter and frequency generator. The simplified Block diagram representation of 16 bit timer and counter is shown in the fig below.

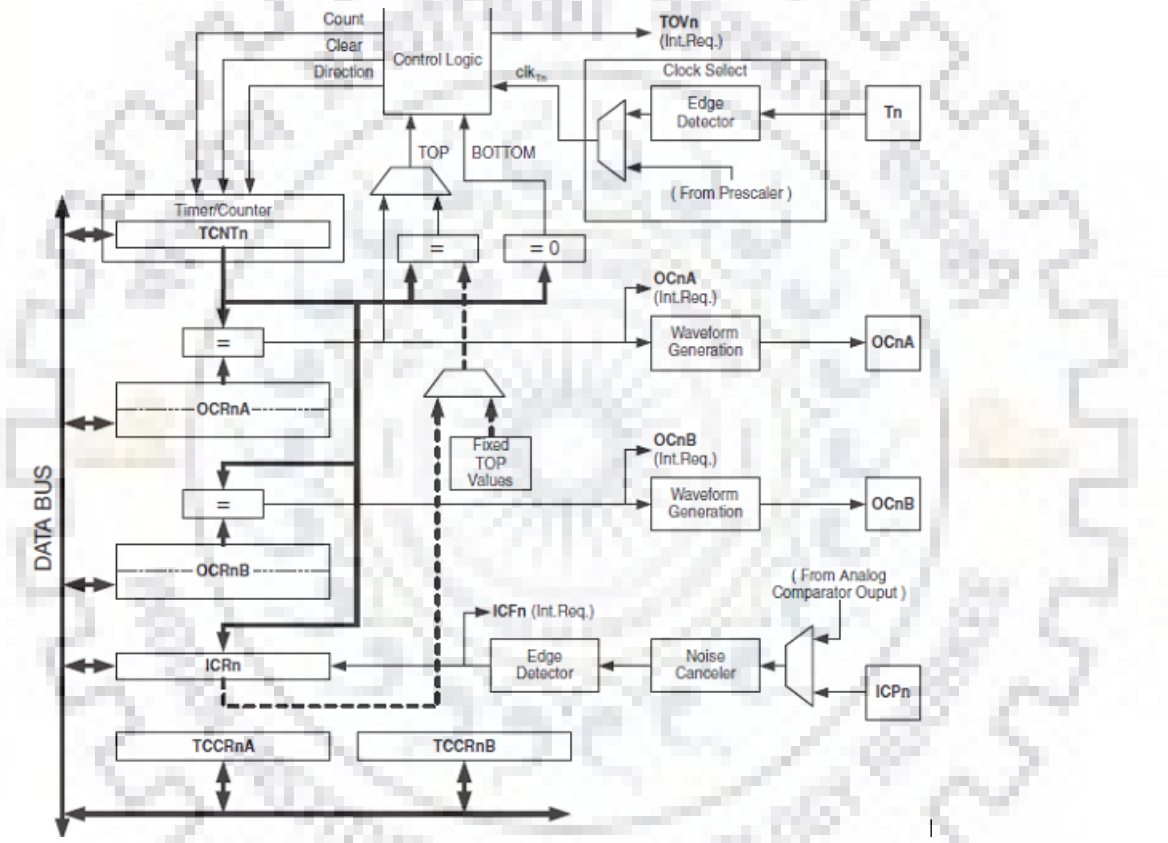


Figure 2. 3 The 16 bit timer and counter block diagram [1]

The timer/counter (TCNT1), input capture registers and output compare registers (OCR1A/B) all are sixteen bit registers. Using clock signal on pin T1 timer/counter can be clocked externally. The select clock bit (CSc12:CS0) in the Time/counter control reg B (TCCR1B) indicates which clock to use internal or externa. The buffered output compare register (OCR1A/B) are compared with the timer counter values at all the time and the wave form generator. For some applications it uses these result to generate pulse width modulation output.

Output compare unit of timer/counter

The comparator unit compares TCNT 1 and the Output compare register. When TCNT is equal OCR1 x the comparator shows a match. Then a match will set output compare flag (OCF1 x) at the next timer clock cycle. When active the flag output will produce an interrupt.

Below figure 2.3 show block diagram of O/P comparing unit.

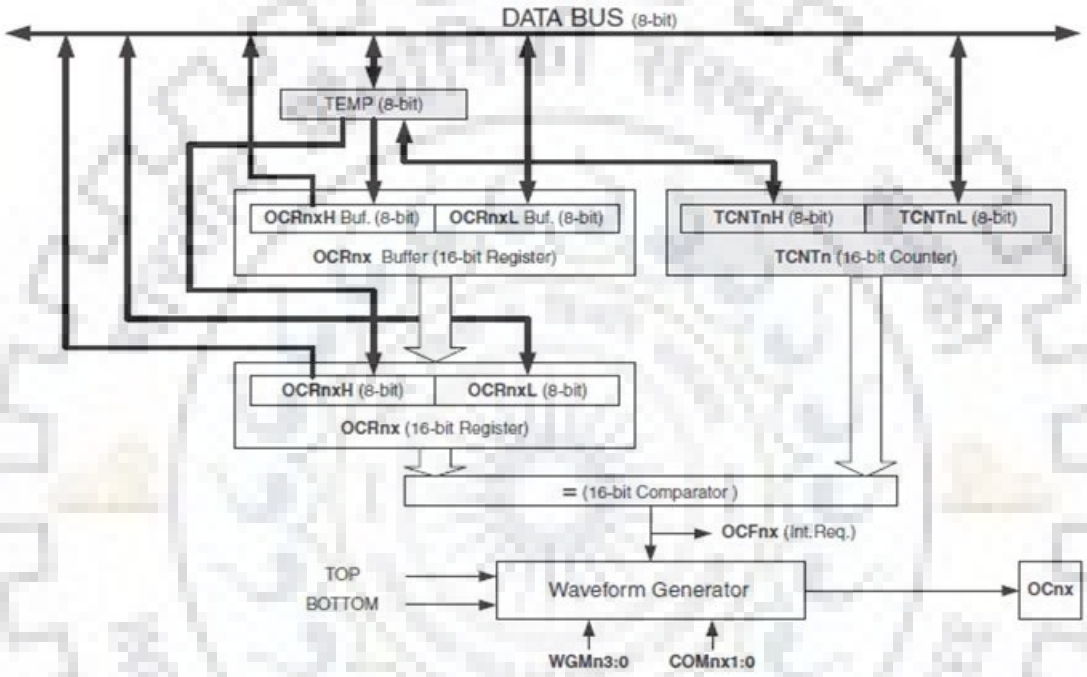


Figure 2. 4 The Output compare unit

2.3 ANALOG TO DIGITAL CONVERTER (ADC)

Atmega16 analog to digital converter (ADC) is used ten Bit sequential approximation way in order to change eight inputs from pin PORTC with analog MUX.

The analog to digital converter takes analog i/p voltage on the pin of PORTC and will be convert it to ten Bit value. These values are displayed as ADCH and ADCL in analog to digital converter registers. Here the minimum values represents ground and the maximum values represent the voltage in pin minus one LSB.

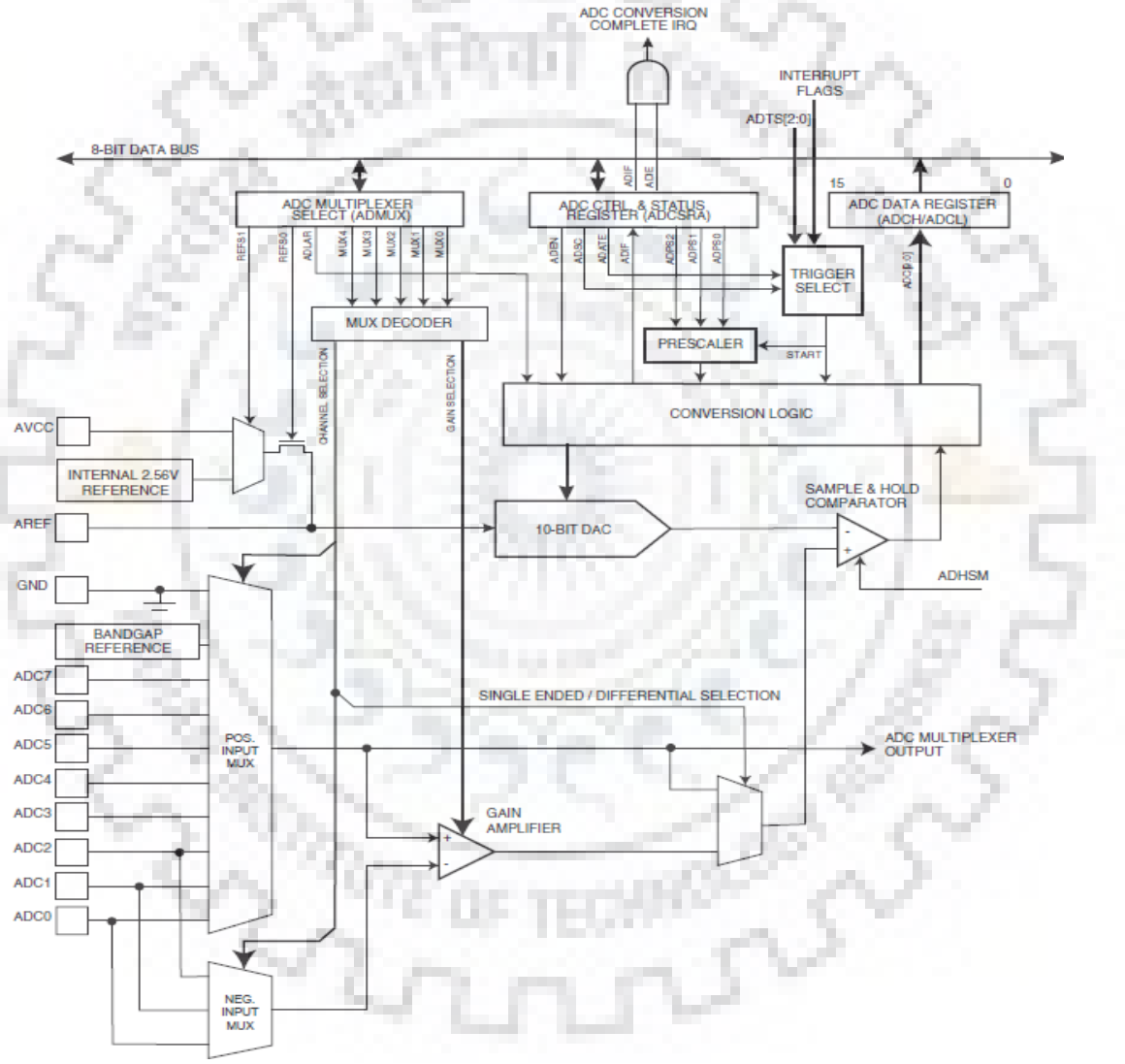


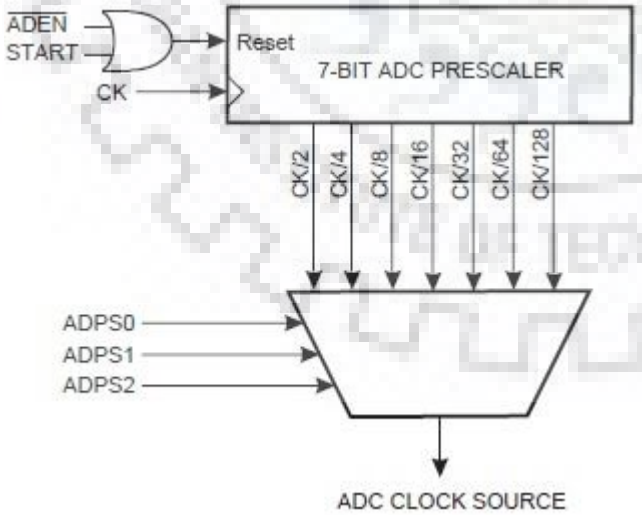
Figure 2. 5 Block diagram of ADC ATmega16 [2]

The data in the register are adjusted from right, and that ADCL will be recieved first. A VREF for A to D will be selected as 1.1V, The VCC pin that describes the conversation ranges for analog to digital converter. The result found in ADC for a single conversion is described by below equation.

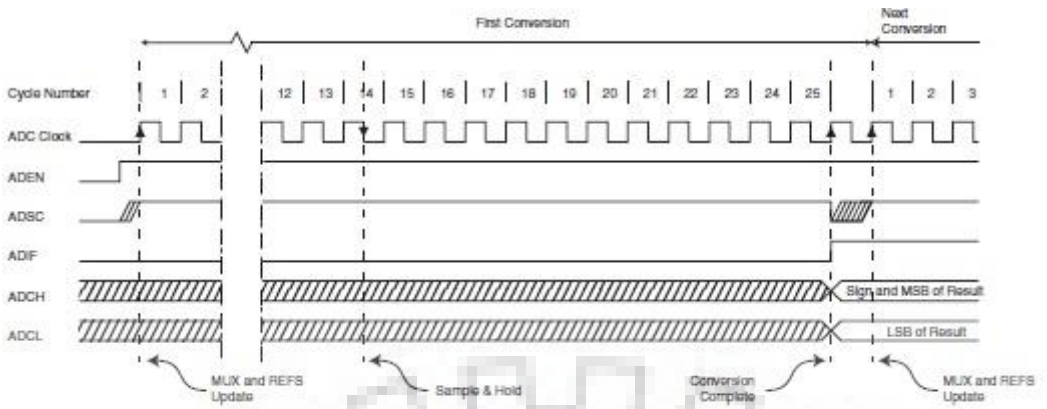
$$ADC = \frac{V_{IN} \cdot 1024}{V_{REF}}$$

2.3.1 Analog to digital converter (ADC) prescaler

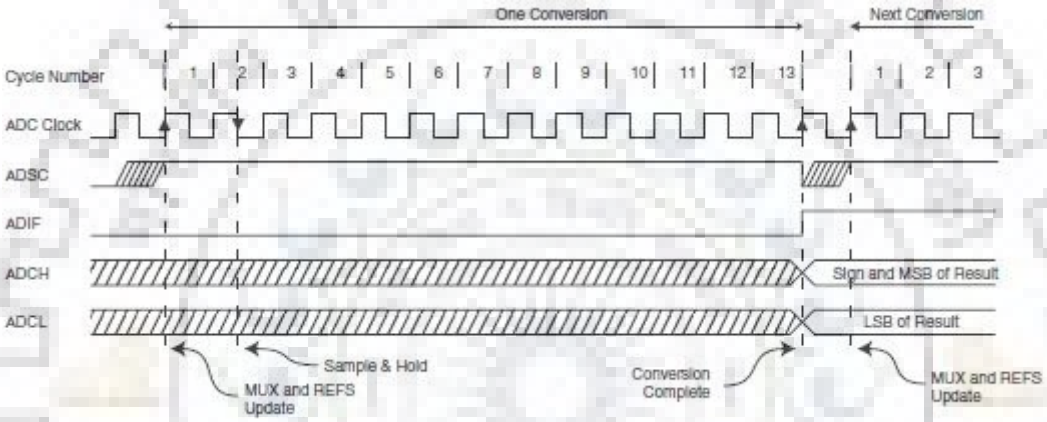
The ADC prescaler conversation will start whenever there is positive edge occurs on the selected trigger signal and ADC reset. To get desired resolution of ADC we need input a frequency of 50 to 200 KHz. As an input a prescalr will takes clock frequency of CPU. Below in figure 2.6 shows an ADC prescaler. Up to ADEN bit set to ADCRA register the prescaler start counting and will be in the running mode [2].



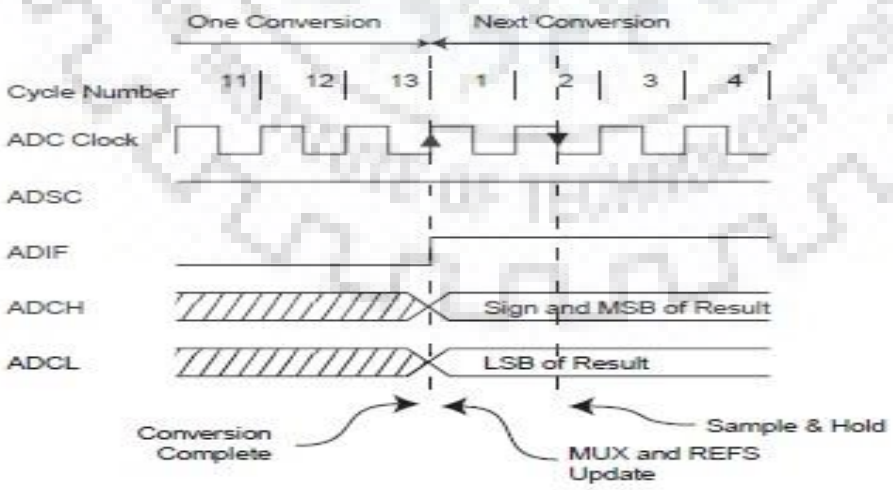
Analog to digital converter prescaler [2]



(a) First conversions

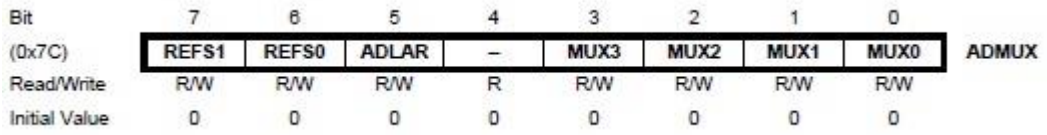


(b) Single conversions



(c) Free running conversions

2.3.2 ADC Multiplexer selection ADMUX register



Bit	7	6	5	4	3	2	1	0	
(0x7C)	REFS1	REFS0	ADLAR	–	MUX3	MUX2	MUX1	MUX0	ADMUX
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADC multiplexer select register [2].

The bit 7 to 6 – REFS1 to 0: reference of selection Bit

These bits selects the voltage reference for analog to digital converter. For this project it is 00 as AREF external voltage used.

Bit 5 - ADTE:

Whenever the bit is written to one, the trigger of analog to digital converter will be activated. A to D starts conversion on selected tiger signal of positive edge.

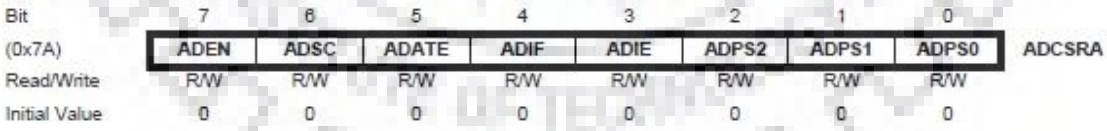
Bit 4 - ADIF: ADC interrupts flag

Bit - 4 is set whenever ADC conversion complete and data the data registers are updated their data value. When ADIE bit and I bit in SREG are set, conversion of ADC will complete and interrupt is executed [2].

Bit 3 - ADIE: ADC interrupt enable

When bit 3 is written to one and the I bit in SREG is set, conversion of ADC complete and interrupt will be activated.

2.3.3 ADCSRA – ADC control status register A



Bit	7	6	5	4	3	2	1	0	
(0x7A)	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

ADC control and status registerA

Bit 7 –ADEN: ADC Enable

When this bit is one, it will enable ADC. When it is zero, ADC will be turned off. When it is off will terminate the conversion.

CHAPTER THREE

3. SOFTWARE AND AHRDWARE REQUERMENTS

3.1 Hardware requirements and description

This project is an embedded system control of temperature of water bath. For this project I have used the following component for implementing hardware.

1) ATmega16/32 development board
2) Temperature sensor RTD Pt-100, LM35
3) On board voltage regulator
4) LCD display
5) 4X4 matrix keypad
6) USBasp programmer
7) Arduino development board
8) LED
9) Resistors, bred board, jumper wires
10) Power supply
11) Pt 100 transmitter
12) Relay

3.3.1 Temperature sensors

Resistance Temperature detector (RTD)

The resistance temperature detectors (RTD) are sensor devices that can sense the temperature and contains a resister that will changes its value of resistance when there is temperature change. These temperature sensors have been used for measuring temperature for years in industrial process and laboratories. The most typs of sensors have a particular sets of condition of temperature sensing for which they are designed.

Resistance temperature detectors (RTD) provides the following advantages

- Wide range of temperature measurement approximately -200 to 850 degree Celsius
- Better accuracy better than thermocouples
- Good interchangeability
- Stable for long time

Figure 3. 1 Resistance temperature detector Pt-100

LM35

Temperature sensor IC's are divided in to two classifications. The first is the analog temperature sensors and digital temperature sensors. Based on the output the analog temperature sensors can be categorized in to two. They are 1. Output voltage and sensors and 2. Sensors that will sense current[2]. LM35 temperature sensors has more advantage than linear temperature sensors which are calibrated in Kelvin, Because of the user is not required to subtract a large constant voltage from the output to get accurate degree centigrade scaling.

LM 35 is a precise centigrade temperature sensor which are produced by national semiconductor. Below figure shows LM 35 and the basic temperature sensor.

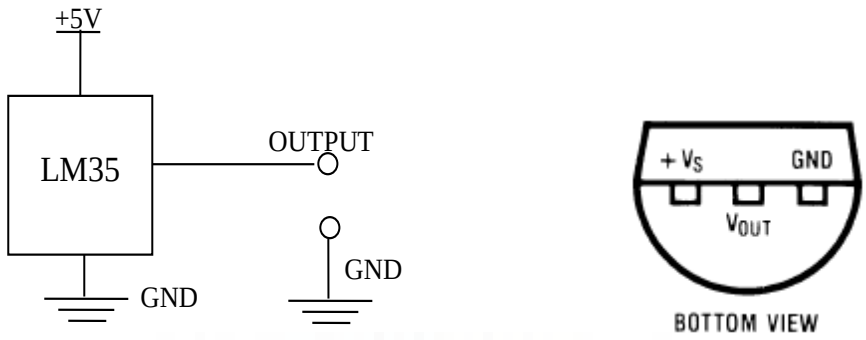


Figure 3. 2 LM 35 temperature sensor

3.1.2 Liquid Crystal Display (LCD)

Alphanumeric display used for displaying the values of temperature read from the temperature sensor and easily can be seen when you set the threshold value of operation. The liquid crystal display (LCD) can display different characters and symbols [4]. This is also used to display any character and symbols in a fixed or sliding way. In this project LCD is used to display the measured and set value of temperature of water bath. I used 16X2 and 16 pin module. Figure below is the 16 pin LCD module used in my project. These LCD is 16 characters by two line wide LCD.

Figure 3. 316 pin liquid crystal display

The LCD can be used as 8 bit or 4 bit microcontroller interface. It depends on the user. In my project I uses as 8 bit liquid crystal display.

LCD have data lines DB[7:0] for data, E(enable pin), R/W(read write) and RS – register select. These pins are used to communicate easily with the microcontroller. Below representation show a four Bit data transfer between LCD and microcontroller [6].

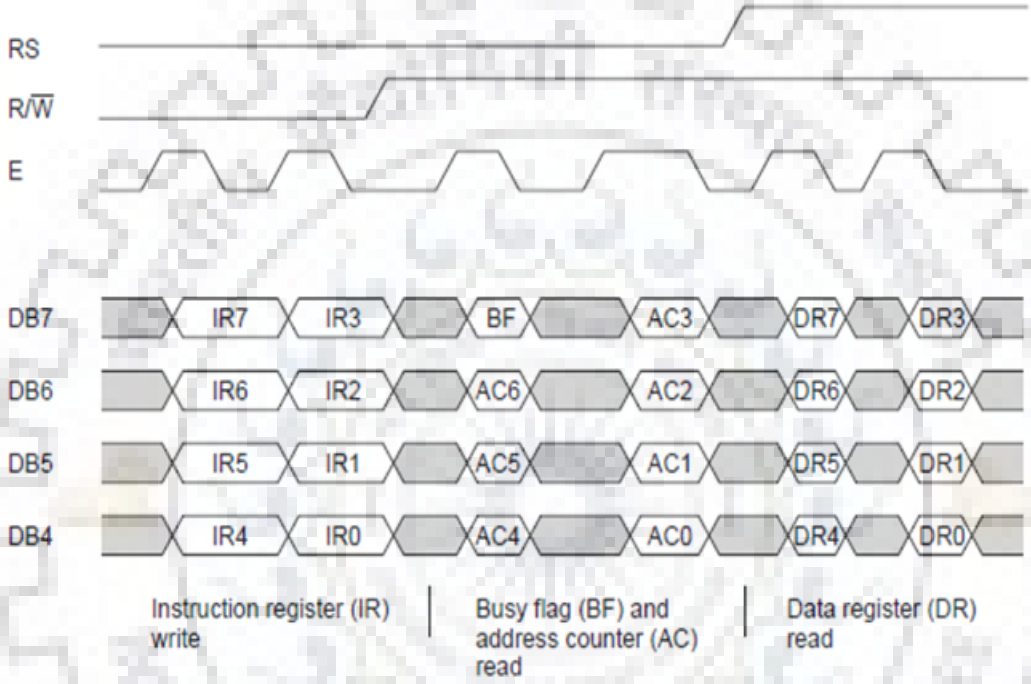


Figure 3. 4 four Bit data transfer between ATmega16 and liquid crystal display [3]

When we use a four bit data transfer only four lines are used i.e DB4-DB7 for transferring data. SO in case of 4bit data transfer we will disable data bus lines DB0 – DB3 and the transfer of data will be completed after all bits are transferred two times. For the order of data transferring four bit higher order i.e 8 bit data operation DB4 to DB7 will be transferred before the low order operations. After transferred four bits the busy flag must be checked two times.

Pin No	Function	Name
1	Ground (0v)	Ground
2	Supply voltage 5V(4.7-4.3)	VCC
3	Contrast adjustment through variable resistor	VEE
4	Selects the command register when low and selects data register when high	Register select
5	Low to write to the register, High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8 - bit data pins	Db0-DB7
8	Backlight VCC (5V)	Led+
9	Backlight ground (0V)	Led-

Tab 3.1 Pin description of 16X2 LCD

3.1.3 KEYPAD

Keypad is used for easily human interface component for many microcontroller projects. It is easy to interface to any microcontroller. The matrix keypad is used for menu selections security system projects and for data entry for embedded system.

Figure 3. 5 4X4 matrix keypad

3.1.4 Relays

Relays are electrically controlled device that can open and closes in different electrical contact. The relays deactivate or activates as switch operations of other device in the same circuits. We have two relay types' technologies which are mechanical and solid state. The mechanical relays are combinations of a switch and an inductor in which the EMF of an inductor cause to change the position. In the case of solid state relay it will accomplishes the same functioning with semiconductor devices changing impedance in order to effectively deactivate or activate a circuit open or closed.

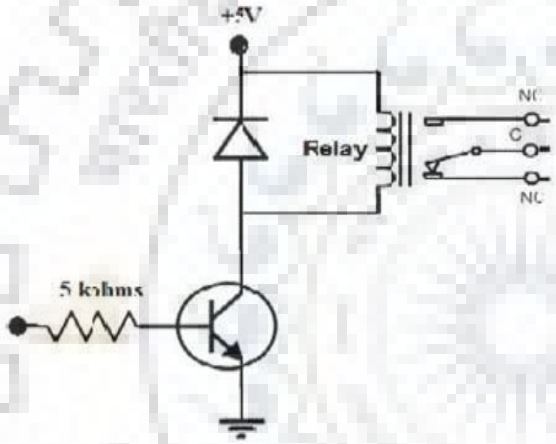


Figure 3. 6 the relay circuit

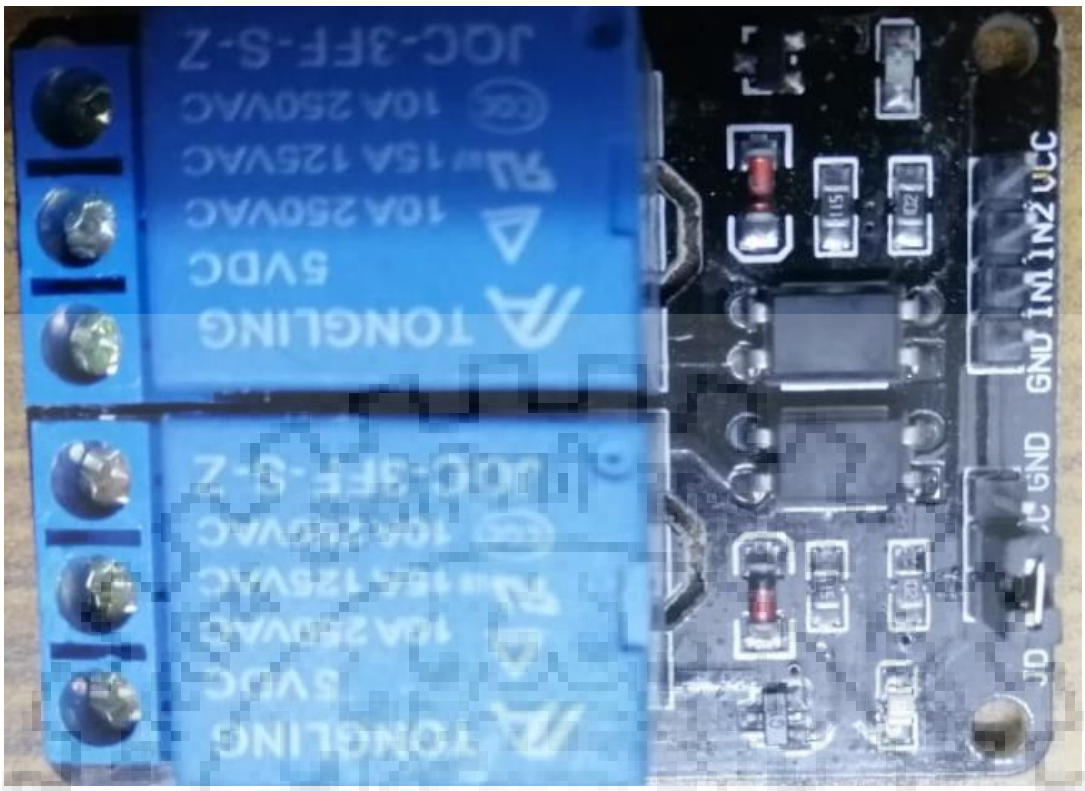


Figure 3. 7 Relay mode 2 5v DC

3.1.5 ATMEGA 16/32 development board

ATMEGA16/32 development board gives us cost effective and simple platform for prototype of project solutions. It is designed to provide connections of all pins of the microcontroller and it is very user friendly. It is made of double side PT H PCB board which will give more strength to joints and connectors to increase reliability. The ATMEGA16/32 board works on 7 V to 15 V DC or AC supply. It has the built in reverse polarity protection. The ATMEGA16/32 has RS232 interface with DB9 connector (female) based on MAX232.The heat sink inside 7805 voltage regulator is used for heat dissipation. So without over heat, it will supply 1 Ampere current continuously.

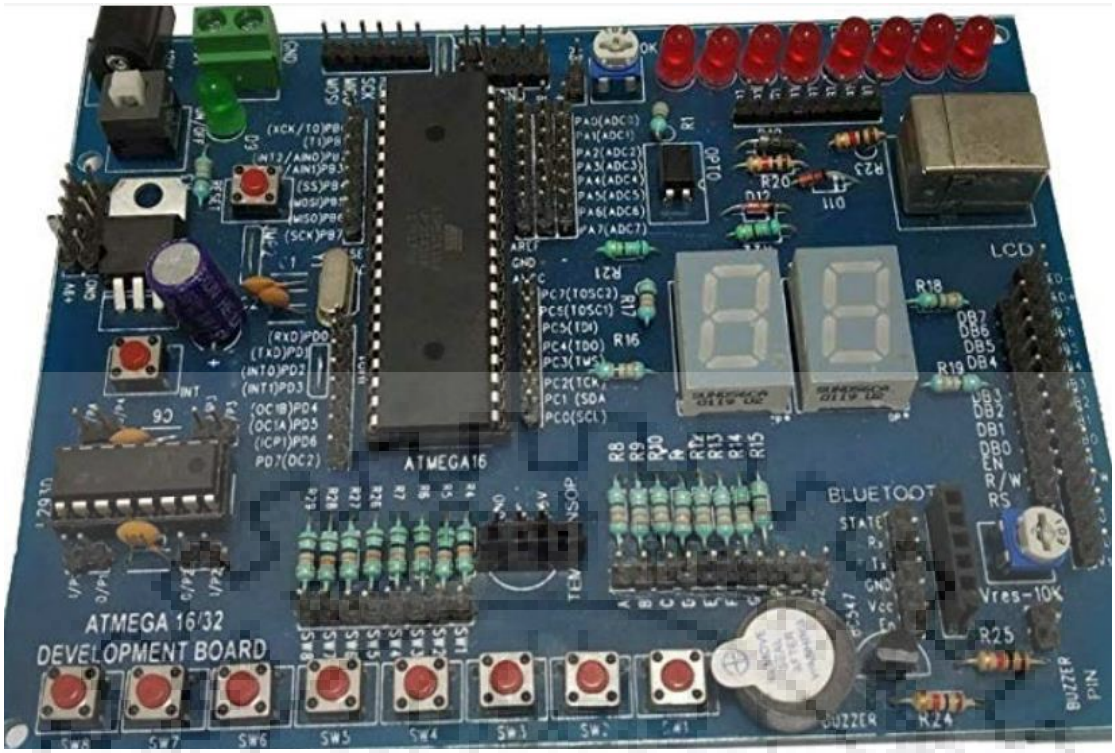


Figure 3. 8 AVR16/32 Development board

3.1.7 USBasp programmer

These is a USB in circuit programmer for ATmel AVR microcontrollers. The USBasp programmer uses a firmware only USB driver. We will not use any special USB controller. We have to connect all the pins to the AVR microcontroller board properly. After connection has done connect its USB to USB of your computer so that the build program which is hex file will be transferred to your microcontroller.

Figure 3. 9 USBasp programmer

3.2 SOFTWARE REQUIREMENT AND DESCRIPTION

3.2.1 The Atmel Studio 7

The Atmel studio software is an integrate development environment (IDE) to develop and debug any AVR microcontroller applications for projects. The Atmel studio provides very easy to use environment to write programs, to debug and easily build applications. The application program can be written using C/C++ or assembly language program. The Atmel studio have powerful visual assist extension. It have very advanced debug functionality.

Figure 3. 10 Atmel studio 7 programming IDE

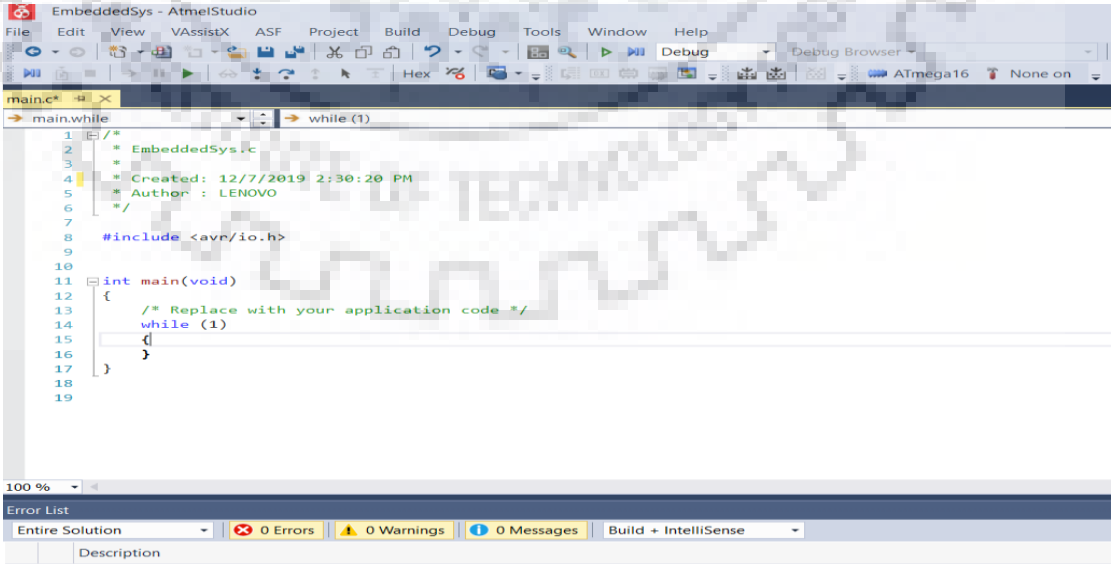


Figure 3. 11 working station of Atmel studio 7 IDE

3.2.1 Proteus simulation software

The proteus simulation software is windows application for schematic and making simulation of electronic projects. It is easy to use and build schematic of electronic project circuits. It have also printed circuit board (PCB) features for layout design. Depending on the type of project and design it can be purchased in many configurations. It is mostly used for microcontroller projects simulation.

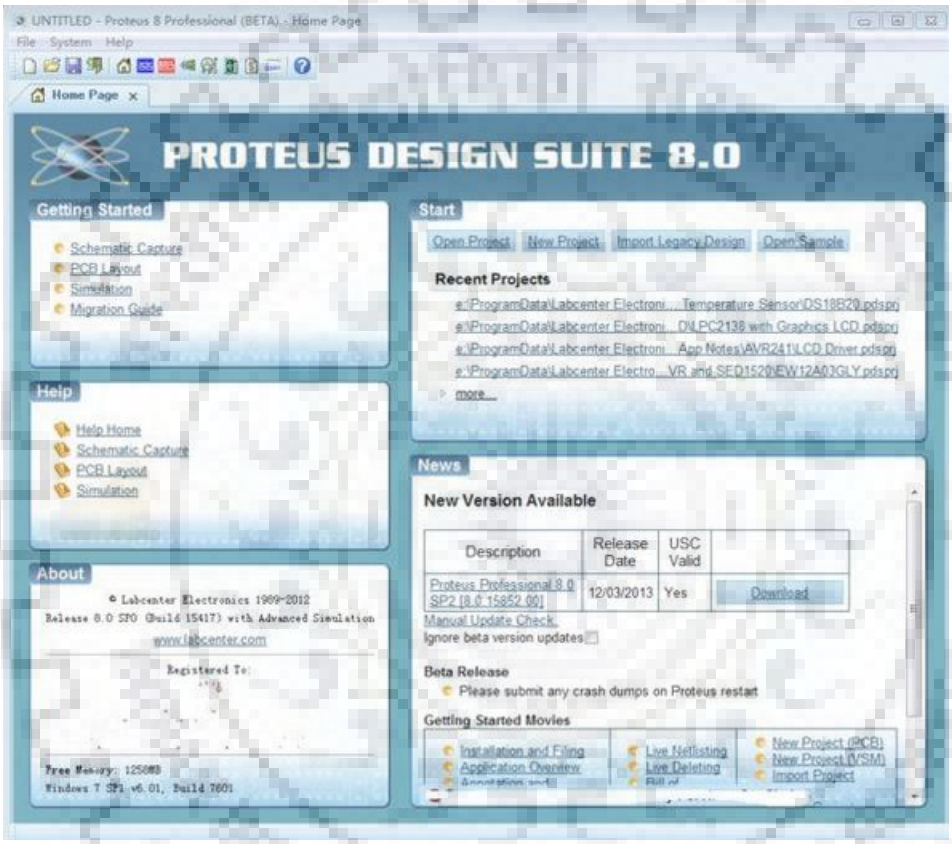


Figure 3. 12 proteus simulation software

3.2.3Arduino IDE

The Arduino IDE is an integrate development environment which provides easy platform to write C/C++ programming. It is an open source software and run on windows, Linux and Mac operating system. The IDE have its text editor for writing the programming algorithms. we can also build and debug the program before uploading to Arduino board. It have its own command window for displaying outputs.

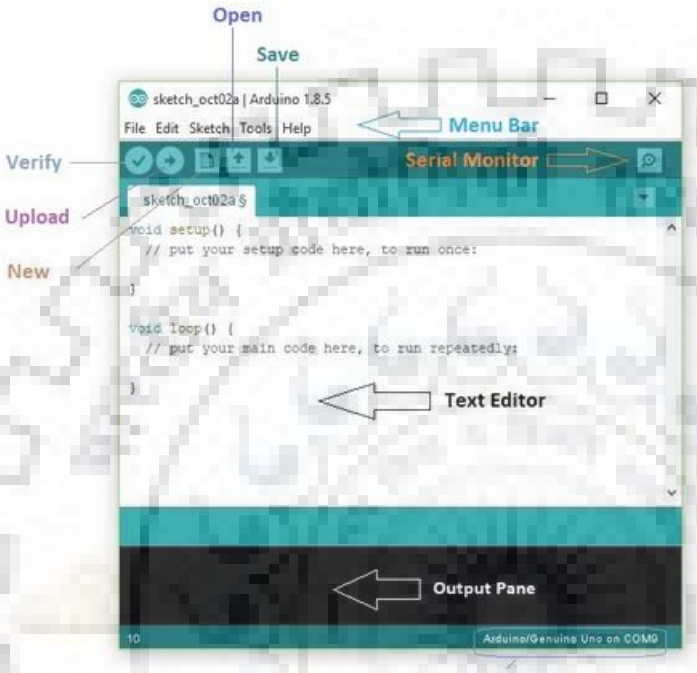


Figure 3. 13 Arduino IDE

CHAPTER FOUR

4 DESIGN AND IMPLEMENTATION

4.1PROCESS MODEL

The process start from designing the structure of system which can properly solve main problems identified. We identified all the required materials for the project. We use Arduino Uno microcontroller LM35 for proteus simulation. For prototype of these project we use Atmega16/32 development board, 4X4 keypad, RTD Pt-100 temperature sensor.

In the system we use RTD pt-100 temperature sensor which sense the temperature of water in the form of voltage. The reading voltage will be sent to microcontroller. Bsed on the program given it will process the received analog signal and converted to corresponding digital signal values. The digital value which is converted will be the equivalent temperature value. An LCD is connected to board for displaying the reading. It will display the temperature value. When the temperature becomes more than a fixed value relay becomes activated and it will switched off the heater. The program is designed to display temperature every 30 seconds.

4.1.1 DFD (Data Flow Diagram)

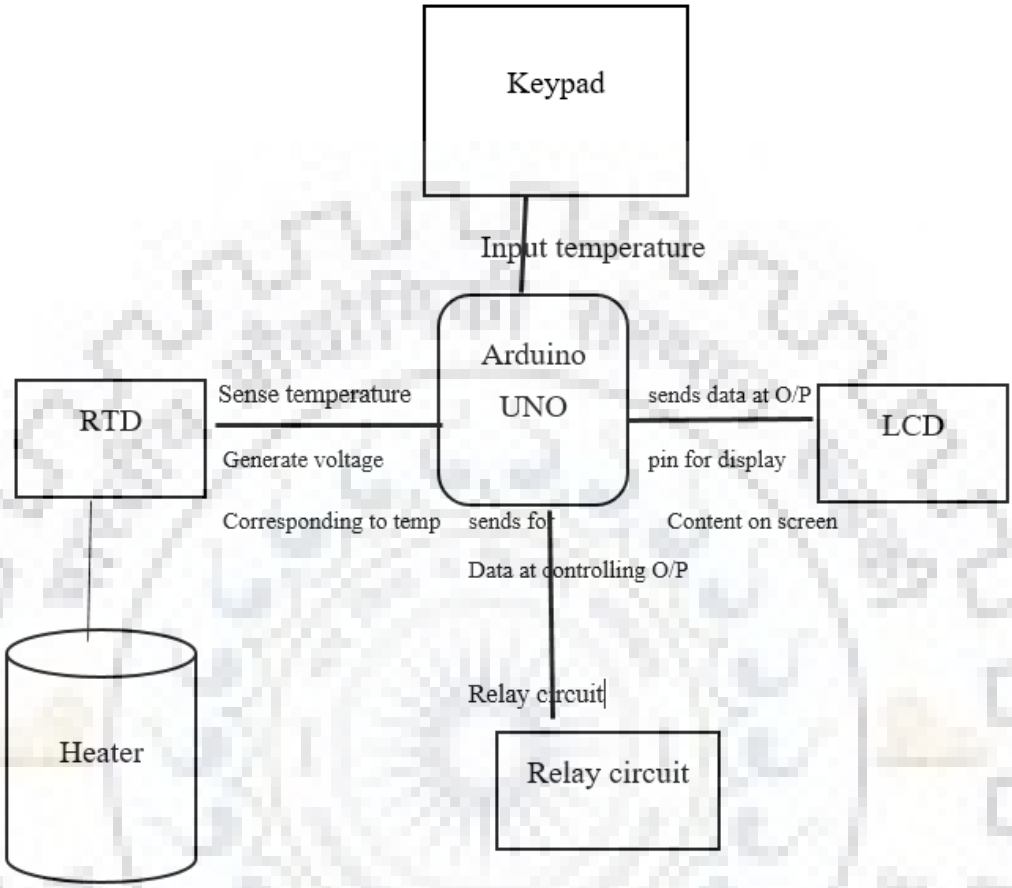


Figure 4. 1 DFD (Data Flow Diagram)

4.1.2 Software simulation

For simulation I used LM 35 temperaturee sensor IC which can generate a voltage corresponding to equivalent temperature value. These voltage is in the form of analog value. The generated analog value is fed to microcontroller unit. The voltage will be given to the analog port A0 of the Arduino Uno. The microcontroller reads analog input and converts the reading in to corresponding digital form using inbuilt A to D converter.

The A to D converter changes analog voltage level in any number between 0 t0 1023. The microcontroller uses 10 bit processing mode.

Based on the program It will multiply the digital data with the coefficient 0.488 and coverts to a particular voltage value. The output will be sent to LCD for displaying.

The Arduino also sends the control bit 0 or 1 on port six. It will compare the value of temperature and if it is lea than the desired temperature the microcontroller provides logic low level to pin 6. But whenever the reading temperature is above the desired level it will send logic high to pin 6.

Based on the logic level of digital pin 6 relay circuit will get input and according to set values devices connected to relay will turns ON/OFF. Therefore the temperature will be monitored and controlled using these method.

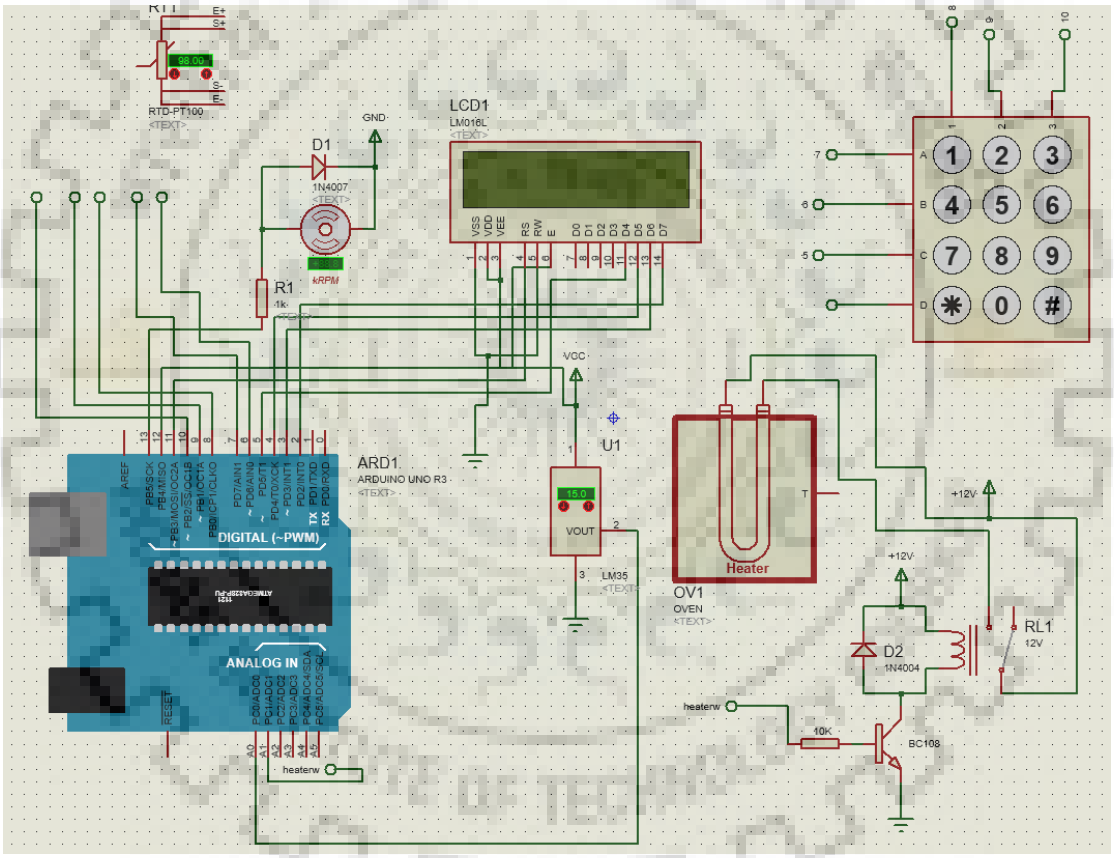


Figure 4. 2Circuit diagram of temperature monitoring and control

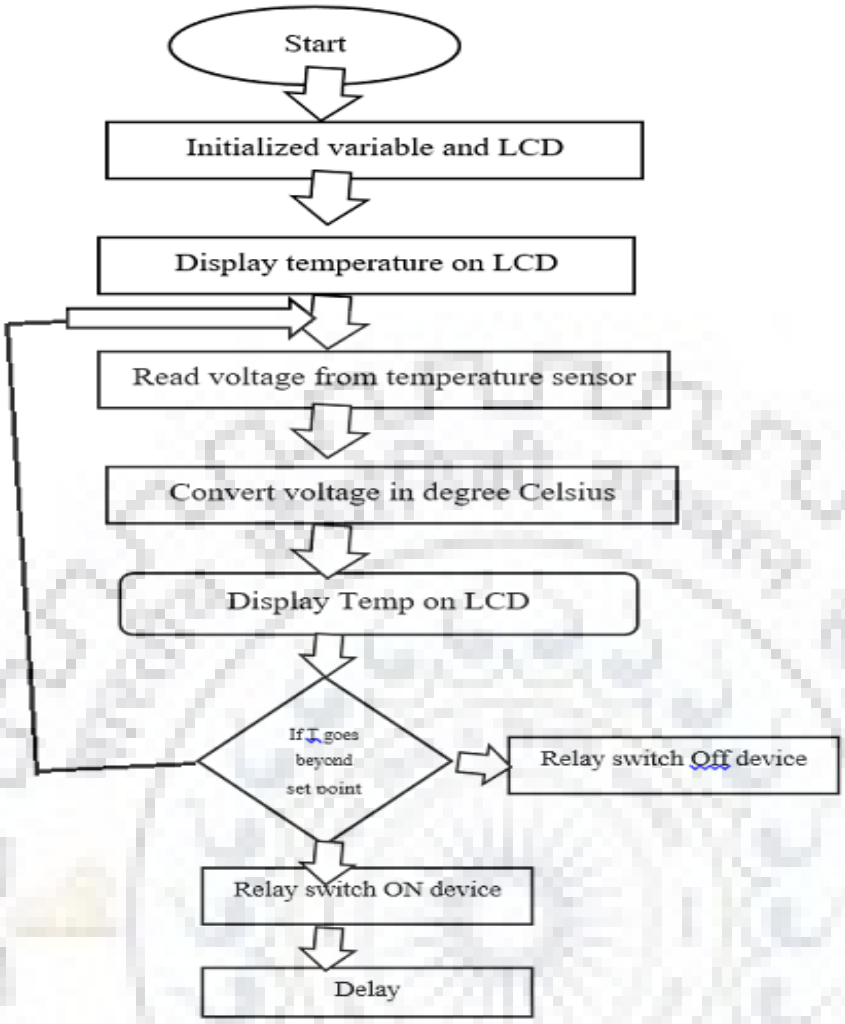


Figure 4. 3 Flow chart for simulation

4.2 HARDWARE DESIGN AND IMPLEMENTATION

4.2.1 Keypad interfacing

In the project keypad is used for entering temperature values. The desired temperature value is inserted using keypad. I 4X4 keypad which have 16 buttons. I use 8 bit mode interfacing.

The keypad in interfaced with Atmega16 microcontroller though port A.

Here is sample program for interfacing keypad

```
#define KEY_PRT     PORTA
#define KEY_DDR     DDRA
#define KEY_PIN     PINA

unsigned char keypad[4][4] = {  {'7','8','9','/'},
{'4','5','6','*'},
{'1','2','3','-'},
{' ','0','=','+'}};

unsigned char colloc, rowloc;

char keyfind()
{
```

4.2.2 LCD interfacing

The liquid crystal display (LCD) is used in this project for displaying values of temperature measured in water bath. It provides us to see the set values and all the reading temperature values. The LCD is interfaced PORT B of the microcontroller.

Here is sample code functions

```
#define  F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
#define LCD_DPRT PORTB
#define LCD_DDDR DDRB
#define LCD_RS 0
#define LCD_EN 1

void LCD_Command(unsigned char cmnd);
void LCD_Char(unsigned char data);
void LCD_Init();
void LCD_String_xy(unsigned char, unsigned char, char *);
void LCD_String(char *str);
void LCD_Clear();
```

4.2.3 Interfacing RDT Pt-100

The RTD Pt-100 temperature sensor is used to sense the temperature of water in the bath. It is connected to transmitter. One leg line of RTD is connected to PORTC 7 of Atmega16 microcontroller. The other two legs is connected to power supply. Sensor is inserted to water and start sensing the temperature of water. These value will be sent to microcontroller through PORTC.

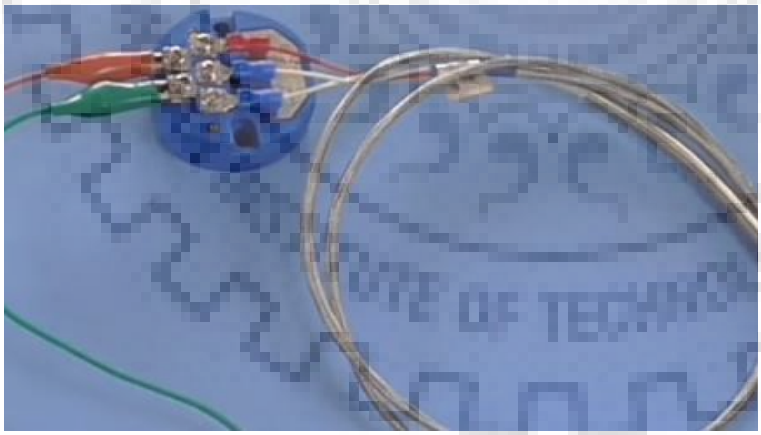


Figure 4. 4 RTD Pt – 100 Interfacing

4.3 RESULTS

After all components are interfaced properly, the following results will be noticed. On start-up there will be a welcome message for the user and it will allow to enter the amount of desired temperature. The cursor in LCD will blink. The desired Temperature value will be entered by user and will be stored in memory of microcontroller for further processing. Below figure 4.5 shows a welcome entry display.

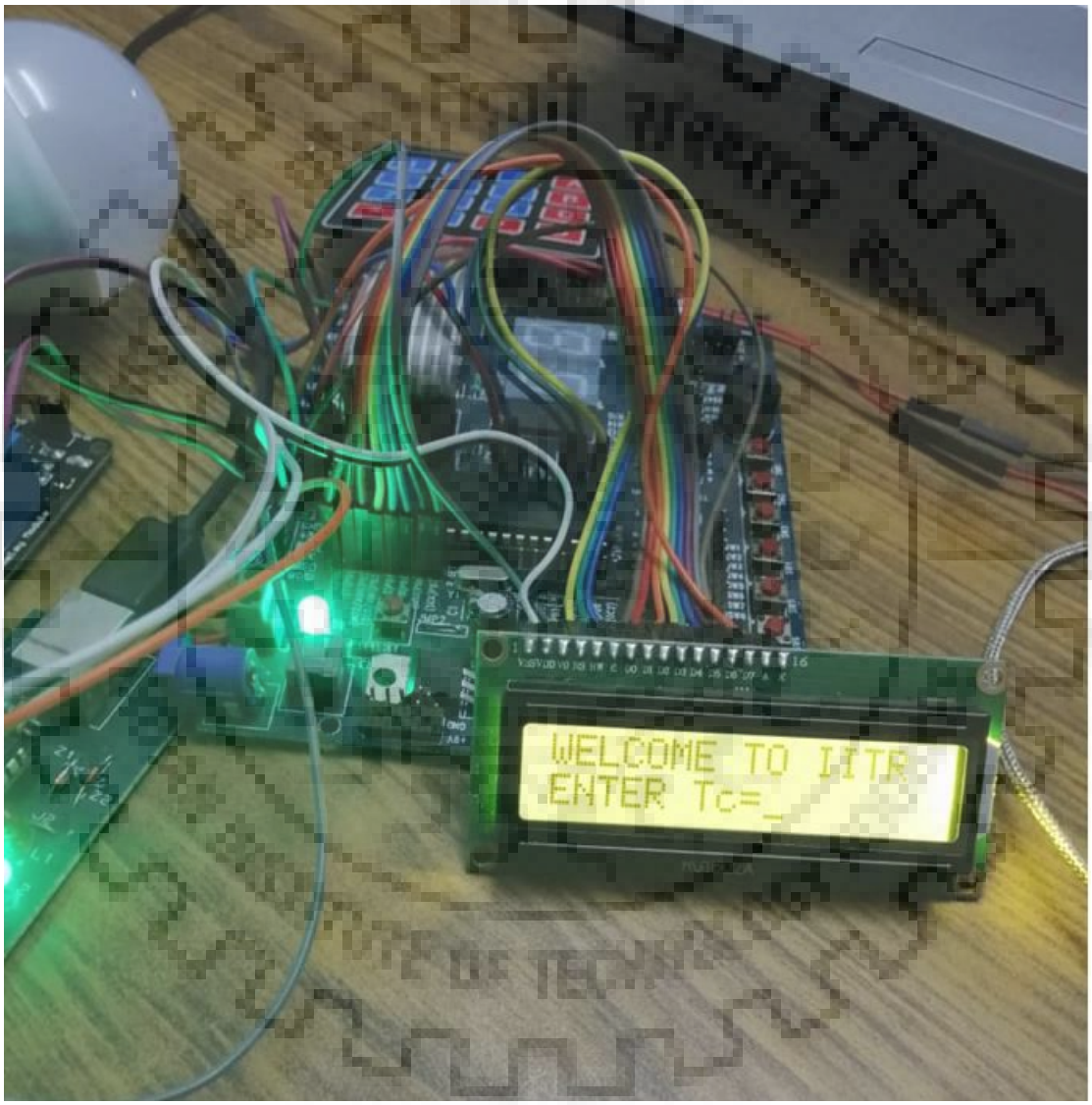


Figure 4. 5 welcome message and enter value of Tc

The next stage is reading and display the temperature of water bath using RTD and the value will be displayed to LCD. The value will be changed every 20 second whe the water heat changes its temperature value.

Here below figure 4.6 are sample measurement displays. Two measured values are displayed.

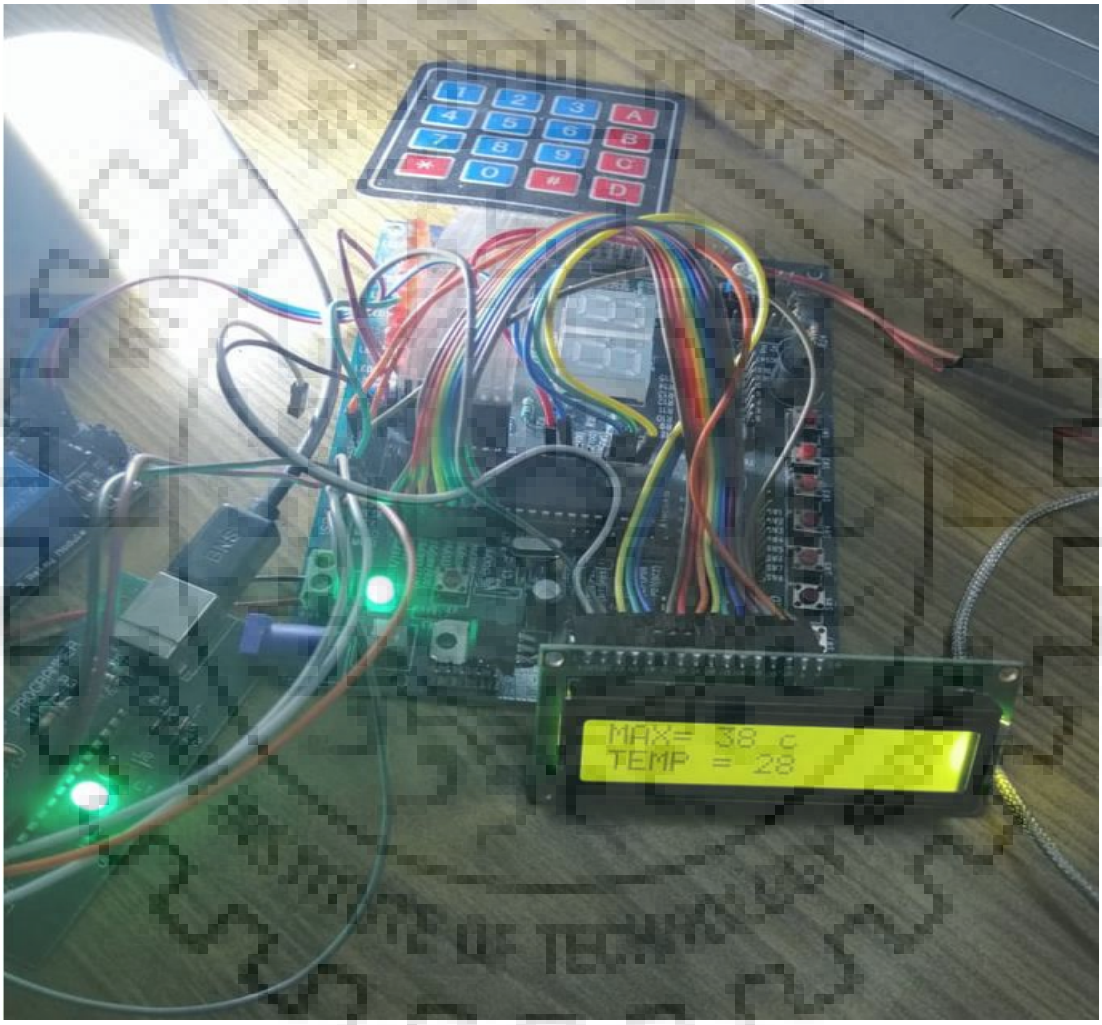


Figure 4. 6 temperature readings

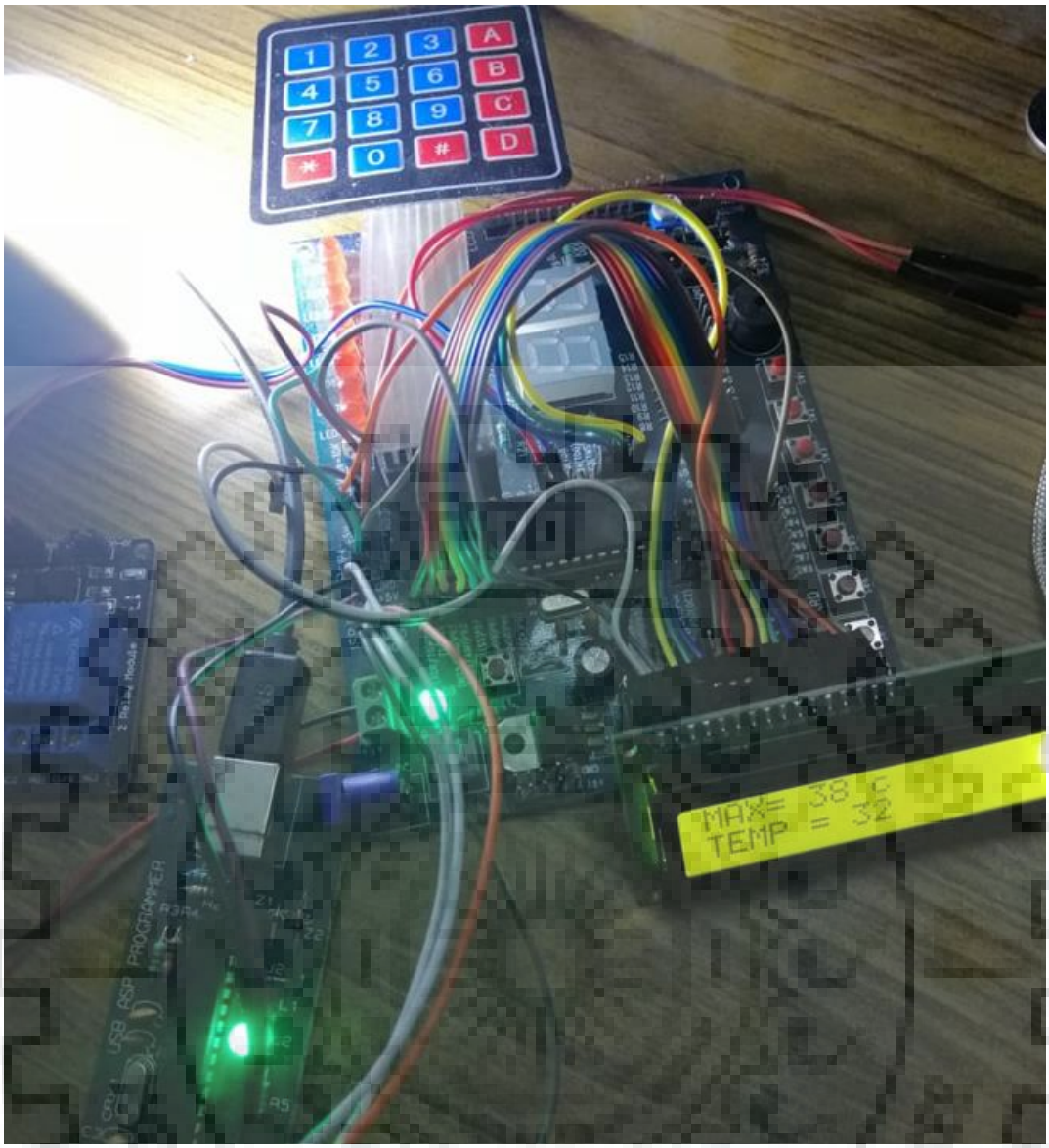


Figure 4. 7 Temperature readings

The third step is responding when the temperature becomes its desired value.

When the temperature becomes its desired value it will display the temperature reading and message of switch off. This indicates the temperature reaches its desired value.

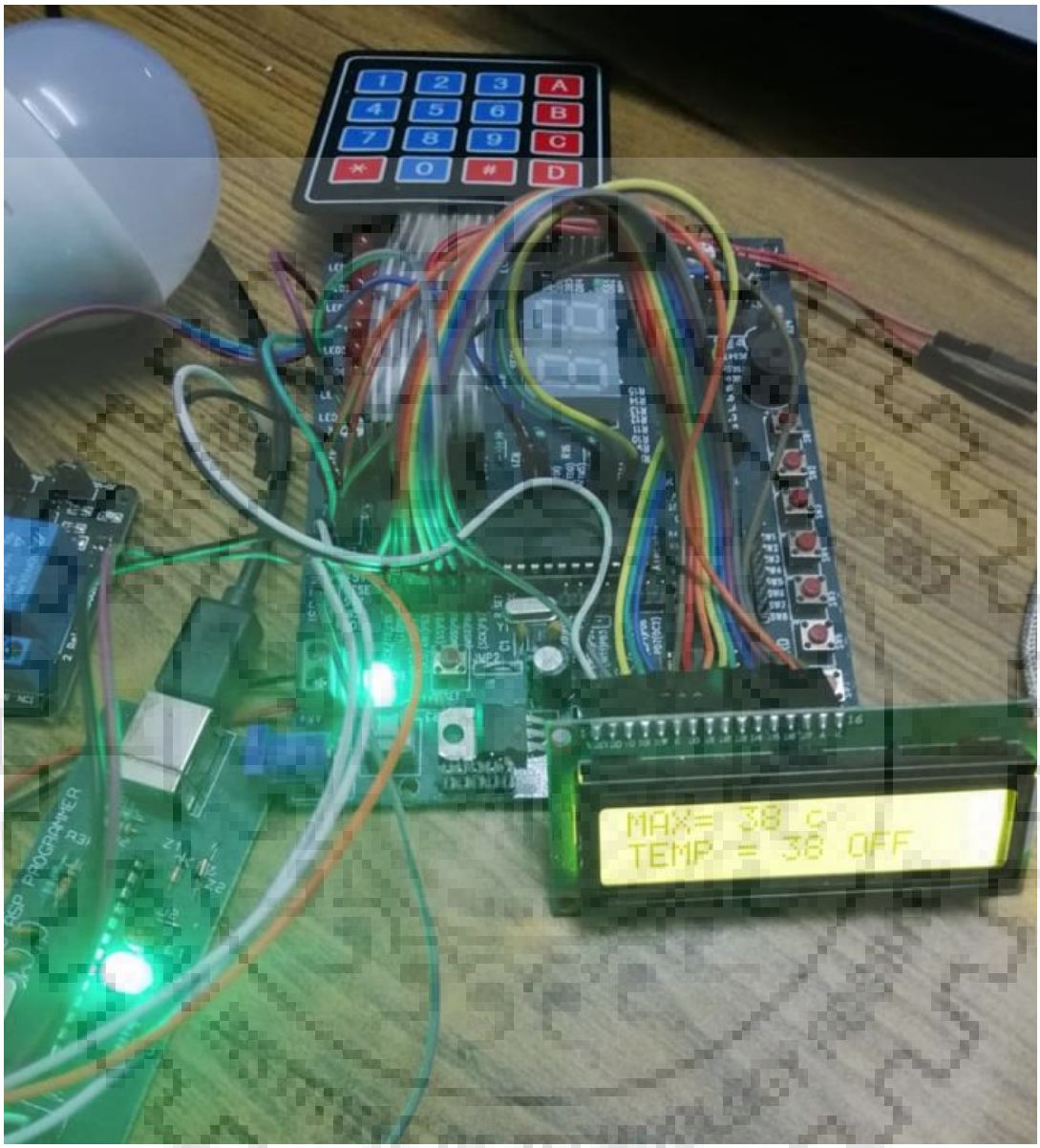


Figure 4. 8 temperature reading of desired value

CHAPTER FIVE

FUTURE WORK AND CONCLUSION

5.1 CONCLUSION

Embedded temperature monitoring system using different temperature sensor was properly designed and implemented. For the project I use RTD Pt - 100 temperature sensor to sense the water temperature. The value of temperature continuously displayed on LCD every 20 seconds. The temperature control algorithm are developed and provide efficiency and accurate measurement of temperature values. I gained knowledge of how to interface sensors in other input output devices in embedded system. Detailed Knowledge gained about microcontroller architecture and how to interface peripheral devices. Detailed knowledge of C programming with microcontroller, detailed knowledge of Atmel studio compiler. The complete working project can be used for controlling and monitor water temperature in bathroom.

5.2 FUTURE WORK

In these project more hardware and software can be included so that the system can be used for further applications. I can be modified and the measured values are transmitted through WIFI transmitters or can be send and received via internet. All the current reading can be sent through http protocol. The reading can be shown on web browsers so that anyone can see and control the process in his office. The microcontroller can be programmed to control other integrated sensors and devices.

SOURCE CODES

A) KEYPAD INTERFACING

```
#include "LCD1.h"
#include <avr/io.h>
#include <util/delay.h>


#define KEY_PRT       PORTA
#define KEY_DDR              DDRA
#define KEY_PIN              PINA

unsigned char keypad[4][4] = {      {'7','8','9','/'},
{'4','5','6','*'},
{'1','2','3','-'},
{' ','0','=','+'}};

unsigned char colloc, rowloc;

char keyfind()
{
      while(1)
      {
            KEY_DDR = 0xF0;          /* set port direction as input-output */
            KEY_PRT = 0xFF;

            do
            {
                  KEY_PRT &= 0x0F;      /* mask PORT for column read only */
                  asm("NOP");
                  colloc = (KEY_PIN & 0x0F); /* read status of column */
            }while(colloc != 0x0F);

            do
            {
                  do
                  {
                        _delay_ms(20);             /* 20ms key debounce time
*/
                        colloc = (KEY_PIN & 0x0F); /* read status of column */
                        }while(colloc == 0x0F);        /* check for any key
press */

                        _delay_ms (40);                  /* 20 ms key debounce
time */
                        colloc = (KEY_PIN & 0x0F);
                  }while(colloc == 0x0F);

                  /* now check for rows */
                  KEY_PRT = 0xEF;            /* check for pressed key in 1st row
*/
                  asm("NOP");
                  colloc = (KEY_PIN & 0x0F);
                  if(colloc != 0x0F)
                  {
                        rowloc = 0;
                        break;
                  }

                  KEY_PRT = 0xDF;              /* check for pressed key in 2nd
row */
                  asm("NOP");
                  colloc = (KEY_PIN & 0x0F);
                  if(colloc != 0x0F)
                  {
                        rowloc = 1;
```

```
                        break;
                }

                KEY_PRT = 0xBF;              /* check for pressed key in 3rd
row */
                asm("NOP");
                colloc = (KEY_PIN & 0x0F);
                if(colloc != 0x0F)
                {
                        rowloc = 2;
                        break;
                }

                KEY_PRT = 0x7F;              /* check for pressed key in 4th
row */
                asm("NOP");
                colloc = (KEY_PIN & 0x0F);
                if(colloc != 0x0F)
                {
                        rowloc = 3;
                        break;
                }
        }

        if(colloc == 0x0E)
        return(keypad[rowloc][0]);
        else if(colloc == 0x0D)
        return(keypad[rowloc][1]);
        else if(colloc == 0x0B)
        return(keypad[rowloc][2]);
        else
        return(keypad[rowloc][3]);
}

int main(void)
{
        LCD_Init();
        LCD_String_xy(1,0,"Press a key");
        while(1)
        {
                LCD_Command(0xc0);
                LCD_Char(keyfind());       /* Display which key is pressed */
        }}
```

B) LCD INTERFACING

```
void LCD_Command(unsigned char cmnd)
{
	LCD_DPRT = (LCD_DPRT & 0x0f)|(cmnd & 0xf0);		/* SEND COMMAND TO DATA
PORT */
	LCD_DPRT &= ~ (1<<LCD_RS);					/* RS = 0
FOR COMMAND */
	LCD_DPRT |= (1<<LCD_EN);					/* EN = 1
FOR H TO L PULSE */
	_delay_us(1);							/*
WAIT FOR MAKE ENABLE WIDE */
	LCD_DPRT &= ~(1<<LCD_EN);					/* EN = 0
FOR H TO L PULSE */
	_delay_us(100);
	/* WAIT FOR MAKE ENABLE WIDE */

	LCD_DPRT = (LCD_DPRT & 0x0f)|(cmnd << 4);		/* SEND COMMAND TO DATA
PORT */
	LCD_DPRT |= (1<<LCD_EN);					/* EN = 1
FOR H TO L PULSE */
	_delay_us(1);							/*
WAIT FOR MAKE ENABLE WIDE */
	LCD_DPRT &= ~(1<<LCD_EN);					/* EN = 0
FOR H TO L PULSE */
	_delay_us(2000);						/*
WAIT FOR MAKE ENABLE WIDE */
}

void LCD_Char(unsigned char data)
{
	LCD_DPRT = (LCD_DPRT & 0x0f)|(data & 0xf0);		/* SEND DATA TO DATA PORT
*/
	LCD_DPRT |= (1<<LCD_RS);					/* MAKE RS
= 1 FOR DATA */
	LCD_DPRT |= (1<<LCD_EN);					/* EN=0 FOR
H TO L PULSE */
	_delay_us(1);							/*
WAIT FOR MAKE ENABLE WIDE */
	LCD_DPRT &= ~(1<<LCD_EN);					/* EN = 0
FOR H TO L PULSE */
	_delay_us(100);
	/* WAIT FOR MAKE ENABLE WIDE */

	LCD_DPRT = (LCD_DPRT & 0x0f)|(data << 4);		/*  */
	LCD_DPRT |= (1<<LCD_EN);					/* EN=0 FOR
H TO L PULSE*/
	_delay_us(1);							/*
WAIT FOR MAKE ENABLE WIDE*/
	LCD_DPRT &= ~(1<<LCD_EN);					/* EN = 0
FOR H TO L PULSE*/
	_delay_us(2000);						/*
WAIT FOR MAKE ENABLE WIDE*/
}

void LCD_Init()
{
	LCD_DDDR = 0xFF;
	_delay_ms(20);							/*
WAIT FOR SOME TIME */
	LCD_Command(0x02);						/*
SEND $32 FOR INIT OT 0X02 */
	LCD_Command(0x28);						/*
INIT. LCD 2 LINE, 5 X 7 MATRIX */
	LCD_Command(0x0C);						/*
DISPLAY ON CURSOR ON */
```

```
        LCD_Command(0x01);                                              /*
LCD CLEAR */
        LCD_Command(0x82);                                              /*
SHIFT CURSOR TO WRITE */
}

void LCD_String_xy(unsigned char row, unsigned char pos, char *str)
{
        if (row == 1)
                LCD_Command((pos & 0x0F)|0x80);                         /* Command
of first row and required position<16 */
        else if (row == 2)
                LCD_Command((pos & 0x0F)|0xC0);                         /* Command
of Second row and required position<16 */

        LCD_String(str);                                                /*
Call LCD string function */
}

void LCD_String(char *str)
{
        int i;
        for(i=0;str[i]!=0;i++)                                          /*
Send each char of string till the NULL */
        {
                LCD_Char (str[i]);                                      /*
Call LCD data write */
        }
}

void lcd_clear()
{
        LCD_Command(0x01);
        _delay_ms(3);
        LCD_Command(0x80);
}
```

C) TEMPERATURE READING AND CONVERSION

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <string.h>
#include <stdio.h>
#include "LCD_16x2_H_file.h"

#define degree_sysmbol 0xdf

void ADC_Init(){

	DDRA = 0x00;                                                    /* Make
ADC port as input */
	ADCSRA = 0x87;                                                  /*
Enable ADC, with freq/128  */
	ADMUX = 0x40;                                                   /*
Vref: Avcc, ADC channel: 0 */
}

int ADC_Read(char channel)
{
	ADMUX = 0x40 | (channel & 0x07);                        /* set input
channel to read */
	ADCSRA |= (1<<ADSC);                                            /* Start
ADC conversion */
	while (!(ADCSRA & (1<<ADIF)));                                  /* Wait
until end of conversion by polling ADC interrupt flag */
	ADCSRA |= (1<<ADIF);                                            /* Clear
interrupt flag */
	_delay_ms(1);                                                   /*
Wait a little bit */
	return ADCW;                                                    /*
Return ADC word */
}


int main()
{
	char Temperature[10];
	float celsius;

	LCD_Init();                     /* initialize 16x2 LCD*/
	ADC_Init();                     /* initialize ADC*/

	while(1)
	{
		LCD_String_xy(1,0,"Temperature");
		celsius = (ADC_Read(0)*4.88);
		celsius = (celsius/10.00);
		sprintf(Temperature,"%d%cC  ", (int)celsius, degree_sysmbol);
/* convert integer value to ASCII string */
		LCD_String_xy(2,0,Temperature);                 /* send string data
for printing */
		_delay_ms(1000);
		memset(Temperature,0,10);
	}
	return 0;
}
```

REFERENCES

[1] Dogan Ibrahim, "Microcontroller based temperature monitoring", Elsevier science and technology books, pp 1-7, pp 129-144, September 2002.

[2] Atmel " 8 bit Microcontroller with 8KB in system programmable flash " ppl -6, 109-137, 245 -260, September 2006.

[3] Bogdan levarda and cristinn budaciu, "Design of temperature control System Using PIC18F4620", Proc. ICSTC 2010, 282-286, 2010.

[4] AGoswami, T. Bezboruah , K>C> sarma,"An embedded design For Automatic Temperature controller", International Journal of Advanced Engineering and Application, Jan 210.

[5] Limin Gai." Temperature measurement and control based on Embedded web" Computer science and information, Vol 2 No 2, may 2009.

[6] R.Szabo, A. Gontean, I. Lie, "The Oscilloscope as a digital Display" PDes 2010. 10th IFAC work shop on Programmable Device and embedded systems, 2010,pp 195-200.

[7] Yi Xianjun, liu cuimei, "Development of High precision temperature measurement system based onARM" Electronic measurement and system 9th international conference 2009, pp.1-795-1-799.

[8] Raspberry pi guide [Online]. Available

[9] R. Schmidt, U scheuermann, "Using the chip as a temperature sensor- the influence of steeplateral temperature gradient on The Vce(T)- measurement" Power Electronics and Application. EPE, 09,13th European conference, 2009, pp, 1-9.

[10] OM Prakash Verma, "Intelligent Temperature controller for water bath system" world Academy of Science, Engineering and Technology International Jornal of computer system Vol:6 No:9 2012 system," procd

[11] Wolf,W.H, -Ahrdware-software co design of embedded systems IEEEjul 1994, Pages: 967-989.