COURSE : CSE-303 MICROPROCESSOR BASED SYSTEM DESIGN

NAME: ZAINAB KHALID

REG NO: 19 PWCSE 1743

SECTION : A

SUBMITTED TO: DR. BILAL HABIB

UNIVERSITY OF ENGINEERING AND TECHNOLOGY, PESHAWAR

DEPARTMENT OF COMPUTER SYSTEMS ENGINEERING.

TASK 7:

SENSOR INTERFACING

TASK:

In this project you are required to interface a temperature sensor to 89C51 using an ADC as shown below in figure 1.

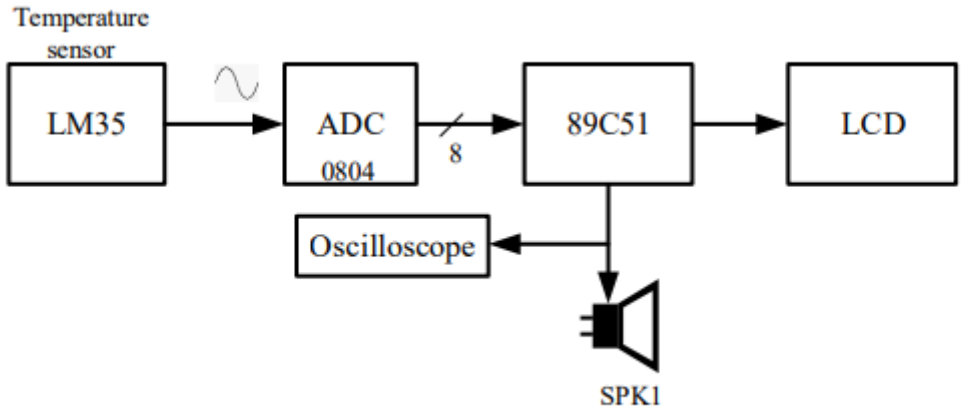


Figure 1: Sensor interfaced to Microcontroller

If temperature > 25C or below 10C then generate a 200Hz beep using speaker/sound card in Proteus. It has a 75% duty cycle and verify it using an oscilloscope. Sampling rate of ADC = 2K samples/sec

CODE:

```
#include<reg51.h>

sbit rs= P3^6; //RS pin of LCD
sbit en = P3^7; //enable pin of LC
sbit speaker= P3^4; //speaker
sbit  read = P3^0; //read pin of ADC connected to pin 3.0
sbit write = P3^1; //write pin of ADC connected to pin 3.1
sbit  intr= P3^2; //interpt pin of ADC connected to pin 3.2
unsigned char temperature;
sfr l_data=0xA0;

 // declaraing functions

void delay_fun(unsigned int); //Function of creating Delays
void write_cmd(int); //Function that sends commands to LCD
void write_data(char); //Function that writes data to LCD
```

```
void convert_fun(unsigned char); //Function that converts the Integer value to char and display it
on LCD
void Set_Timer(int,int); //Function that set the TH0 and TL0 values
void Init(void); //Function that initializes the timer values
void Start_Timer(void); //Fnction that starts Timer 0
void Stop_Timer(void); //Function to Stop Timer 0
void Ext0(void); //Function that is called after the ADC is done with conversion
void lcd_init(void); //function to give commands to lcd
 // defining the functions

 void delay_fun(unsigned int time)
 {
    unsigned int i,j;
    for(i=0;i<time;i++)
      for(j=0;j<125;j++);
 }

 void write_cmd(int z)
 {
    rs = 0; // This is command
    l_data = z; //Data transfer
    en = 1; // => E = 1
    delay_fun(150);
    en = 0; // => E = 0
    delay_fun(150);
 }

 void write_data(char t)
 {
    rs = 1; // This is data
    l_data = t; //Data transfer
    en = 1; // => E = 1
    delay_fun(150);
    en = 0; // => E = 0
    delay_fun(150);
 }

 void timer0() interrupt 1
 {
    if(speaker) //if the Speaker is ON
    {
       speaker = 0; //Turn it OFF
      Set_Timer(0xFB,0x1D); //Set Delay to 1.25msec
```

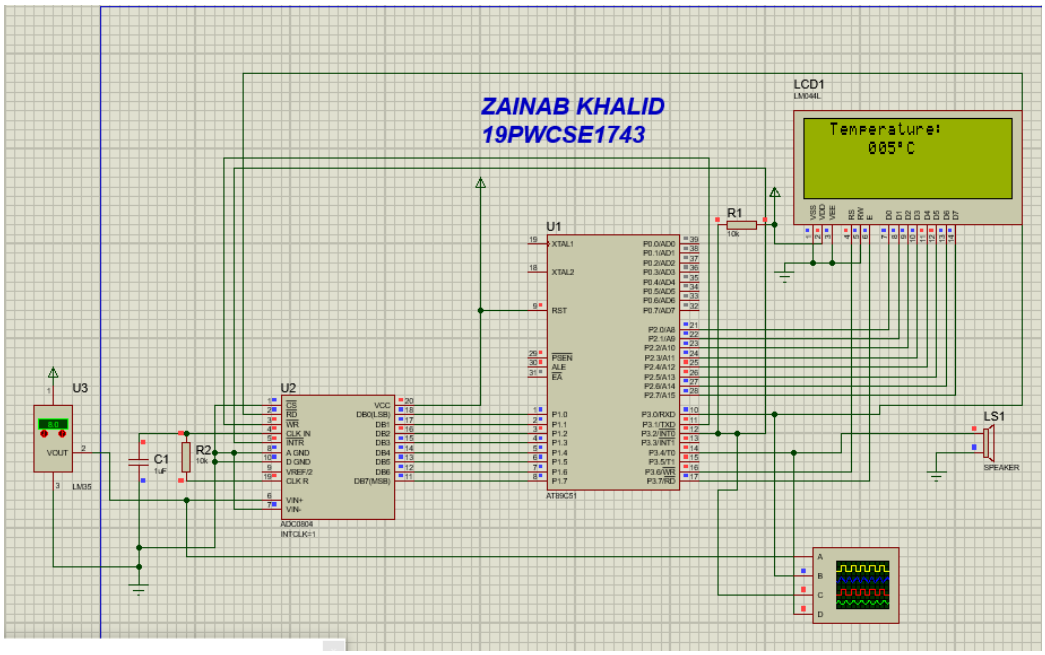
```
    }
   else //if the Speaker is OFF
    {
       speaker = 1; //Turn it ON
      Set_Timer(0xF1,0x59); //Set the delay to 3.75msec
    }

 }
void Set_Timer(int xx, int yy)
 {
    TH0 = xx; //Set the value of TH0 to xx
    TL0 = yy; //Set the value of TL0 to yy
 }

void Init()
 {
    TMOD = 0x1; //Timer 0 is Mode 1
    EA = 1; //Enable Global interrupt
    ET0 = 1; //Enable timer overflow interrupt for timer 0
    Set_Timer(0xF1,0x359); //Set the values of TH0 and TL0 for a delay of 3.75ms
 }

void Start_Timer()
 {
    TR0 = 1; //Start Timer 0;
 }

void Stop_Timer()
 {
   TR0 = 0; //Stop Timer 0
 }

void Ext0()
 {
    read = 0; //Set the RD pin of ADC from HIGH to LOW //The ADC sends the converted value
to P1
    temperature = P1; //Store the value at P1 in temperature
    convert_fun(temperature); //Display temperature on LCD

    if(temperature>25 || temperature<10) //If the is less than 10 or it is greater than 36
    {
       speaker= 1; //Turn the speaker ON
      Start_Timer(); //Start the Timer
```

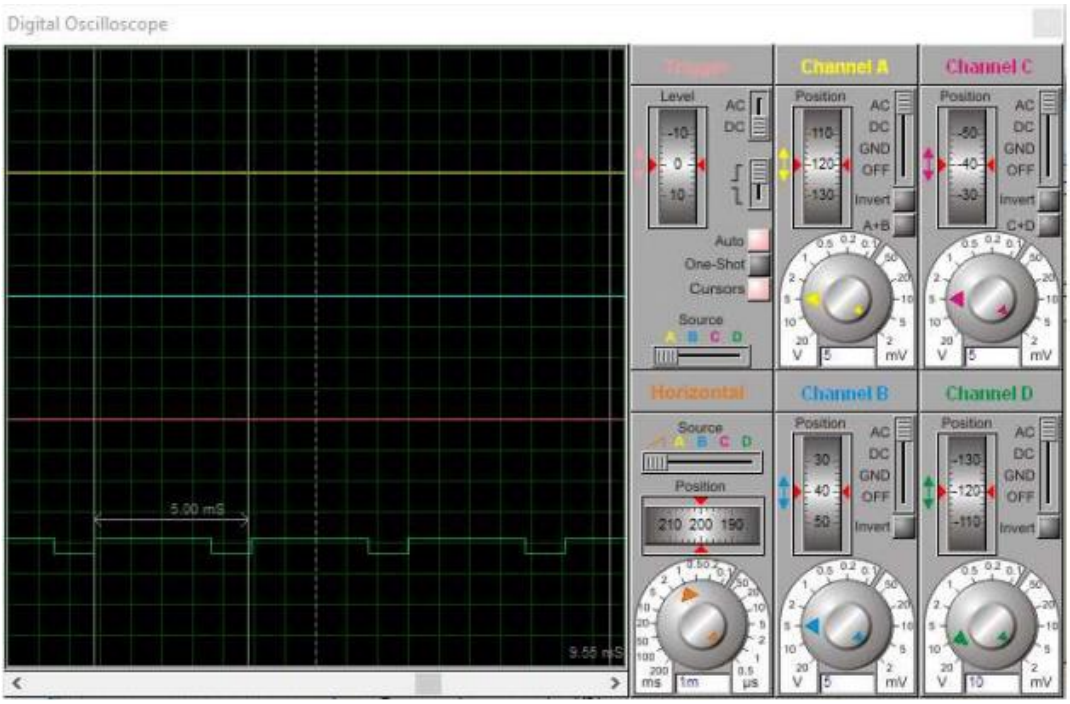
```
    }
    else //if the temperature is in-between 10 and 36
    {
      if(TR0 == 1) //if the Timer 0 is satarted
    {
      Stop_Timer(); //stop the timer
       Set_Timer(0xF1,0x59); //Set a delay of 3.75ms
    }
    speaker = 0; //Turn the Speaker OFF
    }
 }
 void convert_fun(unsigned char value)
 {
    write_cmd(0xc6); //command to set the cursor to 6th position of 2nd line on16*2 lcd
    write_data(((value/100)+48)); //Convert the hundredth place int to char and display on LCD
    write_data((((value/10)%10)+48)); //Convert the tenth place int to char and display on LCD
    write_data(((value%10)+48)); //Convert the unit place int to char and display on LCD
    write_data(0xDF); //Hex value for displaying the Degree sign
    write_data('C'); //Write C to LCD
 }

void lcd_init(void)
{
    write_cmd(0x38);
    write_cmd(0x01);
    write_cmd(0x06);
    write_cmd(0x0C);
    write_cmd(0x82);
}
//main program
void main()
{
    speaker= 0; //Turn the Speaker OFF
    P1 = 0xFF; //Set P1 as an input Port
    intr=1; //Set P3.2 as an input pin
    lcd_init();
    write_data('T');
    write_data('e');
    write_data('m');
    write_data('p');
    write_data('e');
    write_data('r');
    write_data('a');
```

```
    write_data('t');
    write_data('u');
    write_data('r');
    write_data('e');
    write_data(':');

Init();
     while (1)
    {
       read= 1; //Set the RD pin to high
      write = 0; //WR = Low
      write = 1; //Low-> high
      while(intr==1); //Wait for the ADC to Convert the given voltage
      Ext0(); //Call the Ext0 function
   }
}
```

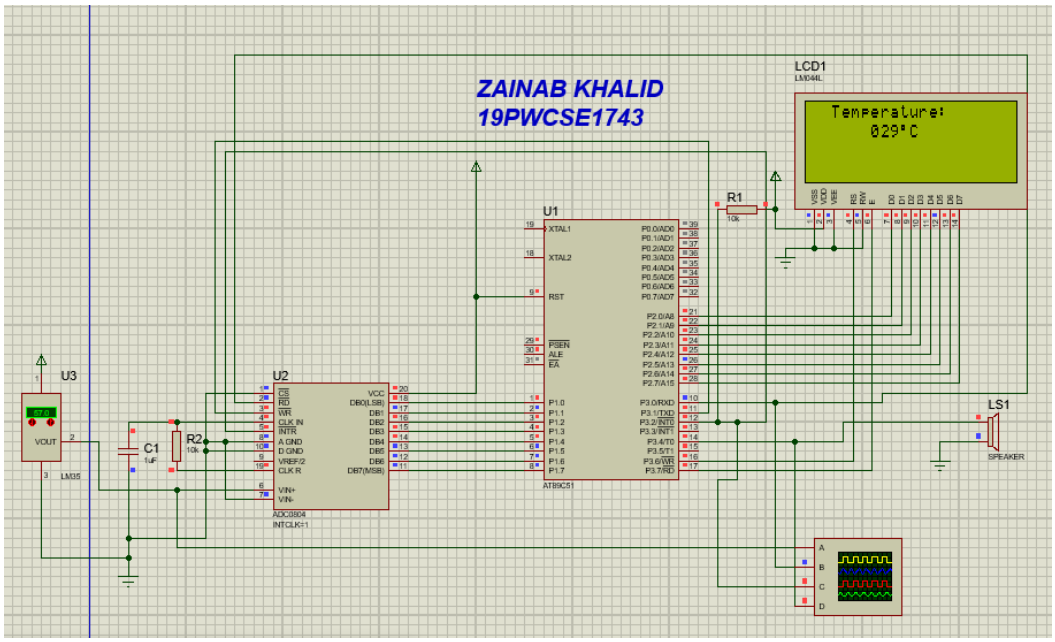
OUTPUT:

WHEN TEMPERATURE < 10:

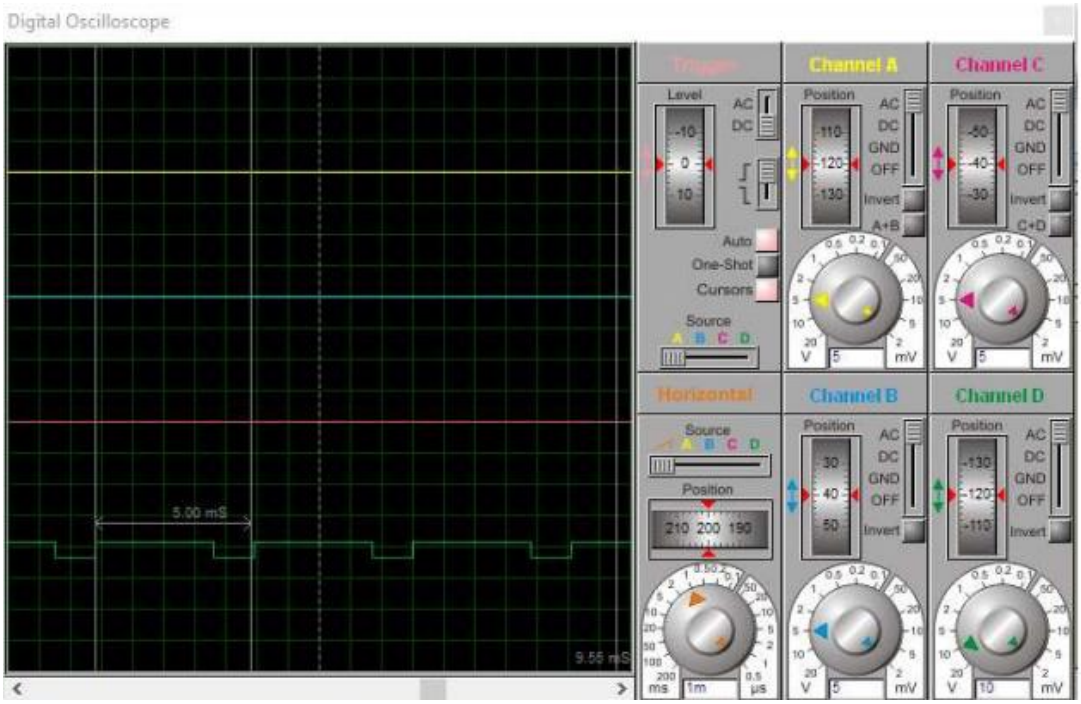




WHEN TEMPERATURE >25:



OSCILLOSCOPE:


Digital Oscilloscope
5.00 mS
9.55 mS
Trigger
Level
AC
DC
Auto
One-Shot
Cursors
Source
A B C D
Channel A
Position
AC
DC
GND
OFF
Invert
A+B
Channel C
C+D
Horizontal
Position
Channel B
Channel D