MICROPROCESSOR BASED SYSTEM DESIGN LAB

LAB 11

Spring 2021

CSE307L MBSD Lab

Submitted by: **Shah Raza**

Registration No. : **18PWCSE1658**

Class Section: **B**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Student Signature: _______________

Submitted to:

Engr. Amaad Khalil

Tuesday, August 10, 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

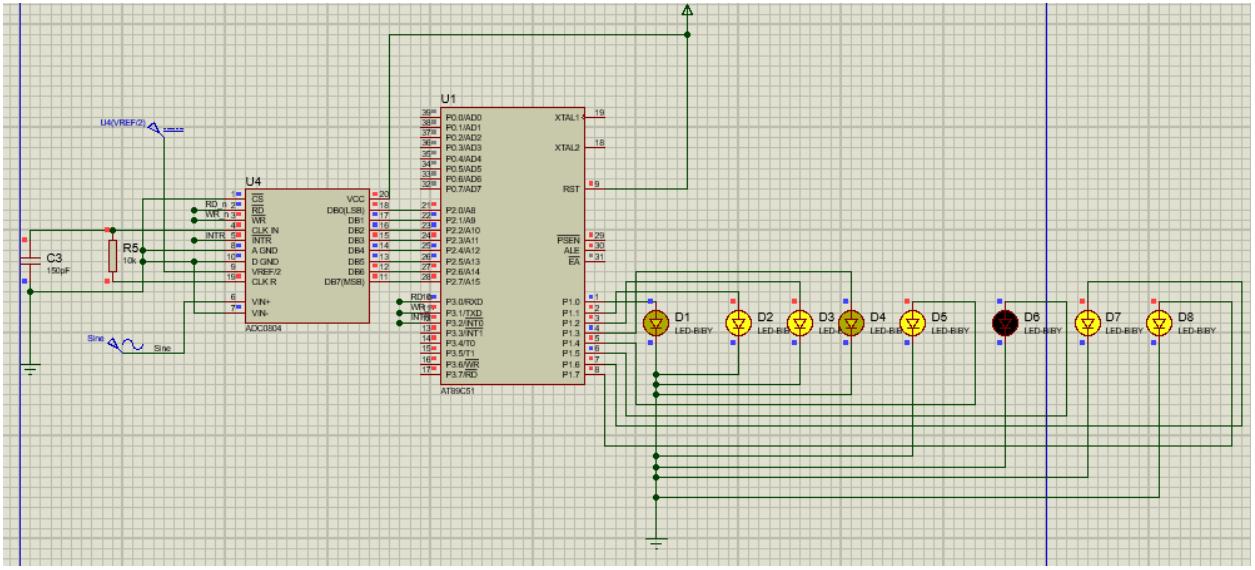
Task 1:

Interface ADC with 8051 microcontroller with 2 input Analoge signals.

Code:

```
#include <reg51.h>
#include <stdio.h>

sbit RD_n = P3^0; //P3.0 is connected to the RD pin of ADC
sbit WR_n = P3^1; //P3.1 is connected to the WR pin of ADC
sbit INTR = P3^2; //P3.2 is connected to the INTR pin of ADC

void Ext0(void); //Function that is called after the ADC is done with conversion

void main(void)
 {
    P2 = 0xFF; //Set P2 as an input Port
   P1 = 0x00; //Set P1 as an output Port
   INTR = 1; //Set P3.2 as an input pin
   while (1)
   {
       RD_n = 1; //Set the RD pin to High
      WR_n = 0;//WR = Low
      WR_n = 1;//Low-->High
      while(INTR==1); //Wait for the ADC to Convert the given voltage
      Ext0(); //Call the Ext0 function
    }
 }

void Ext0()
{
   RD_n = 0; //Set the RD pin of ADC from HIGH to LOW
   //The ADC sends the converted value to P2
   P1 = P2; //Send the value at P2 to P1
}
```

Output / Graphs / Plots / Results:



Task 2:

Interface Buzzer with ADC using 8051 microcontroller for LM35 input signal.

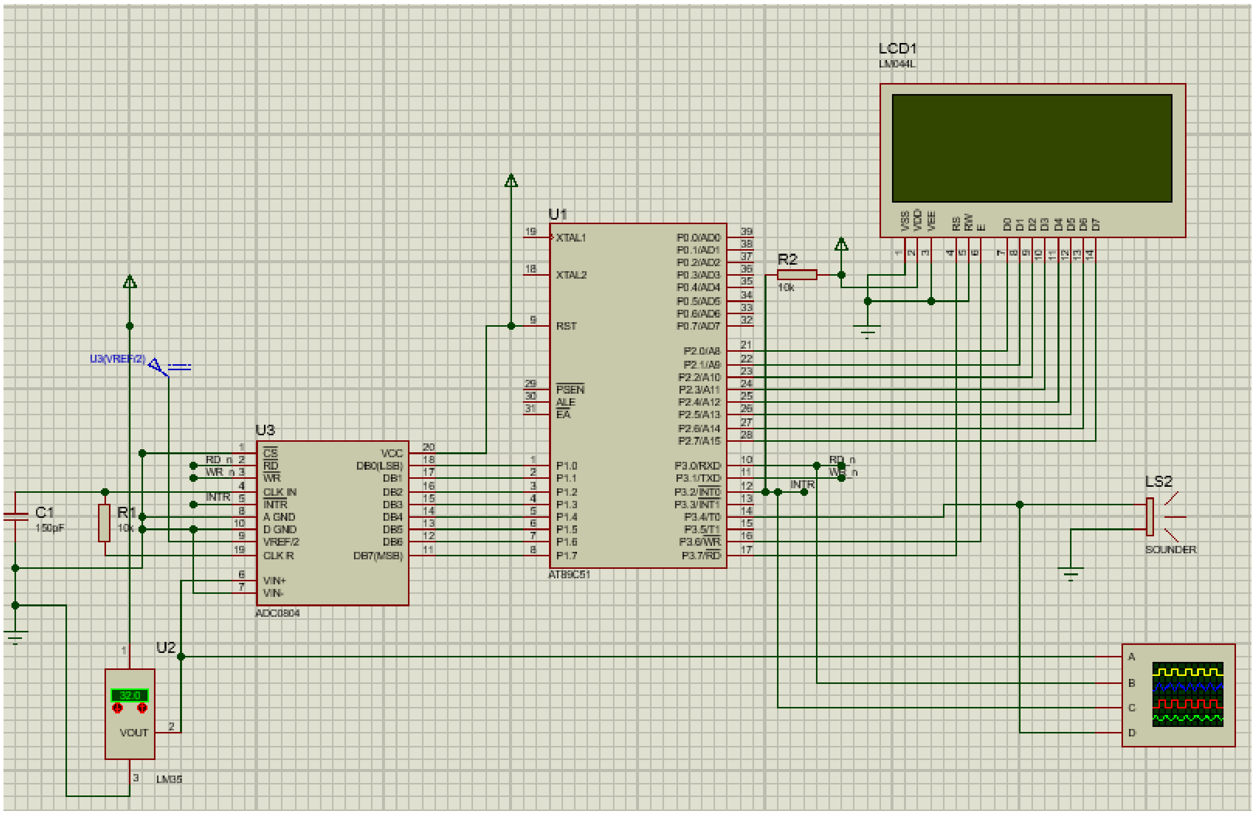
Code:

```
#include <reg51.h>
#include <stdio.h>

sbit SPK = P3^4;        //Speaker is connected to P3.4

sbit RD_n = P3^0;       //P3.0 is connected to the RD pin of ADC
sbit WR_n = P3^1;       //P3.1 is connected to the WR pin of ADC
sbit INTR = P3^2;       //P3.2 is connected to the INTR pin of ADC

sbit RS = P3^7;         //P3.7 is connected to the RS pin of LCD
sbit E = P3^6;          //P3.6 is connected to the E pin of LCD

unsigned char temperature;
unsigned char i;
unsigned char cmd[] = {0x38,0x01,0x06,0x0C,0x82};        //Command that needs to be sent to
LCD

void delay(unsigned int);       //Function of creating Delays
void writecmd(int);                     //Function that sends commands to LCD
void writedata(char);           //Function that writes data to LCD
void convert(unsigned char);  //Function that converts the Integer value to char and display it on
LCD
void SetTimer(int,int);         //Function that set the TH0 and TL0 values
void Init(void);                        //Function that initializes the timer values
void StartTimer(void);          //Fnction that starts Timer 0
void StopTimer(void);            //Function to Stop Timer 0
void Ext0(void);                 //Function that is called after the ADC is done with conversion

/*Timer 0 interrupt is called when the temperature is less that 10 or greater than 36.
It will generate a Sound of 100Hz with a Duty cycle of 25% to P3.4 */
void timer0() interrupt 1
{
  if(SPK )       //if the Speaker is ON
  {
    SPK = 0;     //Turn it OFF
    SetTimer(0xE2,0xB3);         //Set Delay to 7.5msec
  }
  else   //if the Speaker is OFF
  {
    SPK =  1;    //Turn it ON
    SetTimer(0xF6,0x3B);         //Set the delay to 2.5msec
  }
}


void main(void)
 {
   SPK = 0;      //Turn the Speaker OFF
   P1 = 0xFF;    //Set P1 as an input Port
   P2 = 0x00;    //Set P2 as an output Port
   INTR = 1;     //Set P3.2 as an input pin

   for(i = 0;i<5;i++)
   {
     writecmd(cmd[i]);           //Send the Commands to LCD
     delay(10); //Give some delay
   }

   writedata('T');
   writedata('e');
   writedata('m');
   writedata('p');
   writedata('e');
   writedata('r');
   writedata('a');
   writedata('t');
   writedata('u');
   writedata('r');
   writedata('e');
   writedata(':');
   Init();          //Initialize timer values

  while (1)
  {
     RD_n = 1; //Set the RD pin to High
     WR_n = 0;//WR = Low
     WR_n = 1;//Low-->High
     while(INTR==1);  //Wait for the ADC to Convert the given voltage
      Ext0();                //Call the Ext0 function

  }
 }

 void writedata(char t)
{
  RS = 1;            // This is data
  P2 = t;            //Data transfer
  E  = 1;            // => E = 1
  delay(150);
  E  = 0;            // => E = 0
  delay(150);
}

void writecmd(int z)
{
  RS = 0;            // This is command
  P2 = z;            //Data transfer
  E  = 1;            // => E = 1
  delay(150);
  E  = 0;            // => E = 0
  delay(150);
}

void convert(unsigned char value)
{
  writecmd(0xc6);      //command to set the cursor to 6th position of 2nd line on 16*2 lcd
  writedata(((value/100)+48));          //Convert the hundredth place int to char and display on
LCD
  writedata((((value/10)%10)+48));   //Convert the tenth place int to char and display on LCD
  writedata(((value%10)+48));          //Convert the unit place int to char and display on LCD
  writedata(0xDF);                                //Hex value for displaying the Degree sign
  writedata('C');                                 //Write C to LCD
}

void SetTimer(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
  SetTimer(0xF6,0x3B);        //Set the values of TH0 and TL0 for a delay of 2.5ms
}

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

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

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

//Ext0 is used for displaying the temperature value to LCD and generating sound at P3.4
void Ext0()
{
  RD_n = 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(temperature);         //Display temperature on LCD
  if(temperature<10 || temperature>36)         //If the is less than 10 or it is greater than 36
  {
    SPK = 1;   //Turn the speaker ON
    StartTimer();        //Start the Timer

  }
  else   //if the temperature is in-between 10 and 36
  {
    if(TR0 == 1)         //if the Timer 0 is satarted
    {
```

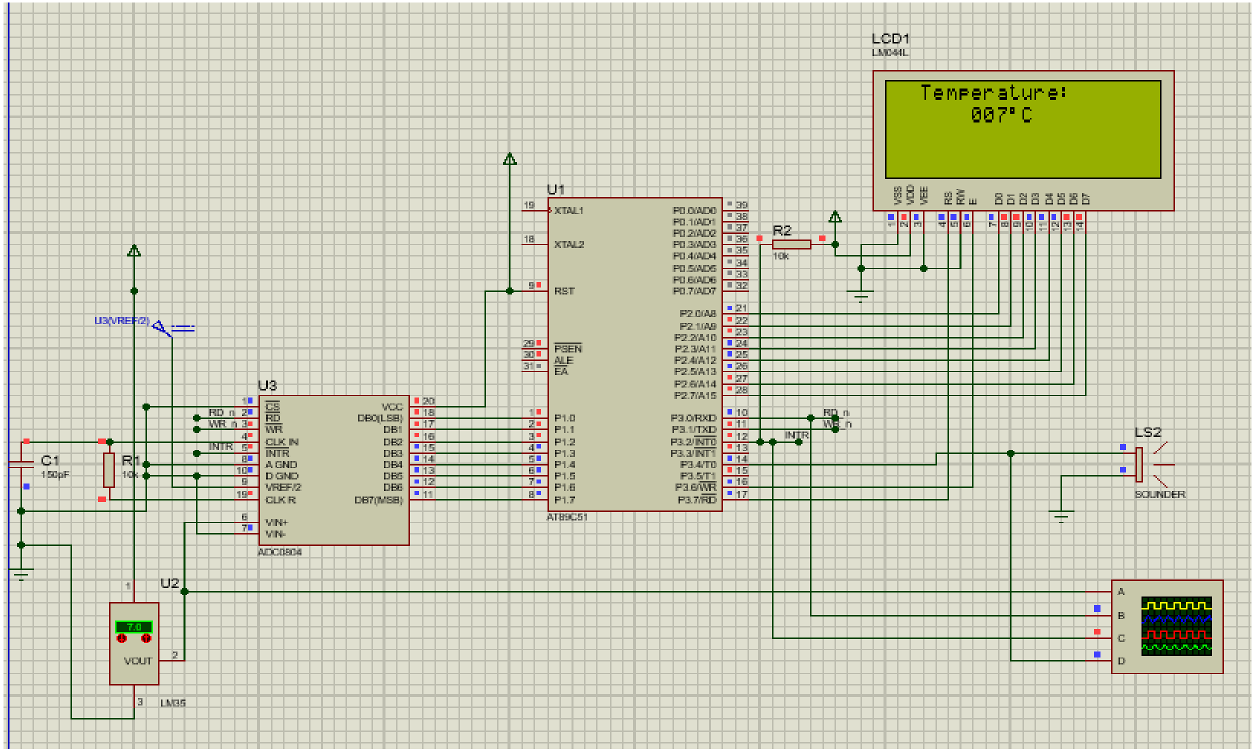
```
              StopTimer();  //stop the timer
              SetTimer(0xF6,0x3B);          //Set a delay of 2.5ms
       }
       SPK = 0;    //Turn the Speaker OFF
    }

}
```

Output / Graphs / Plots / Results:

Schematic:

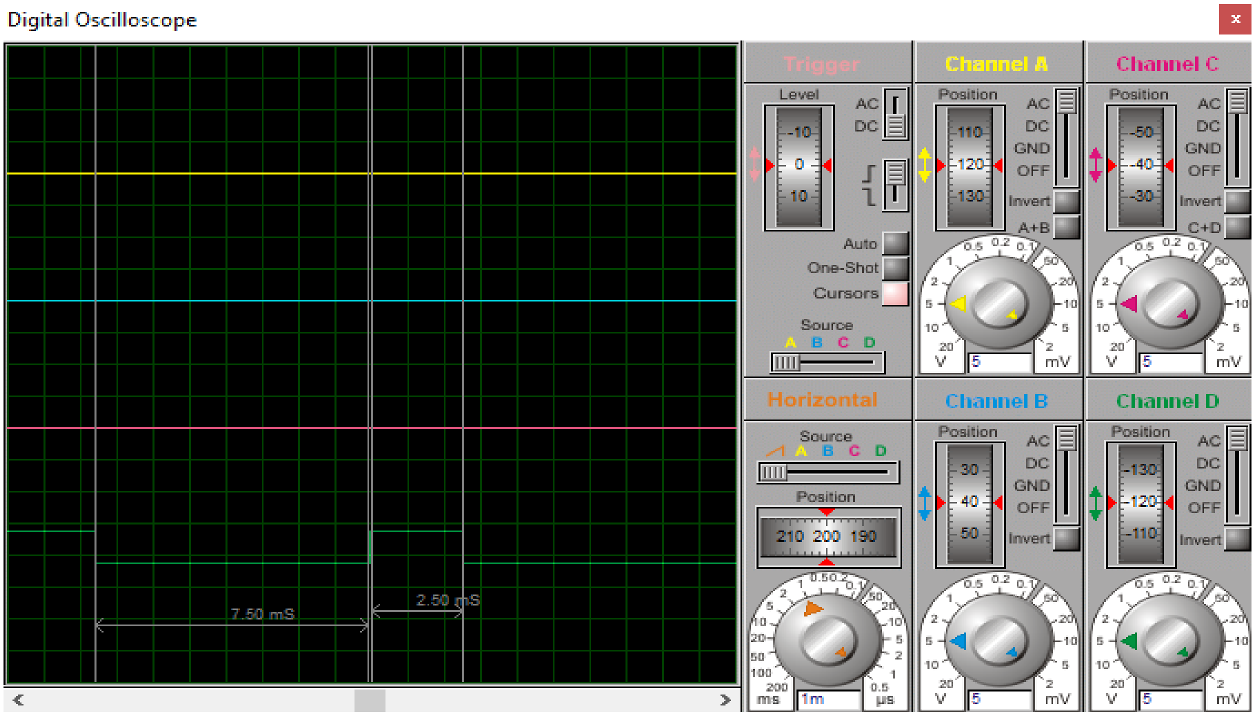


Temperature<10:

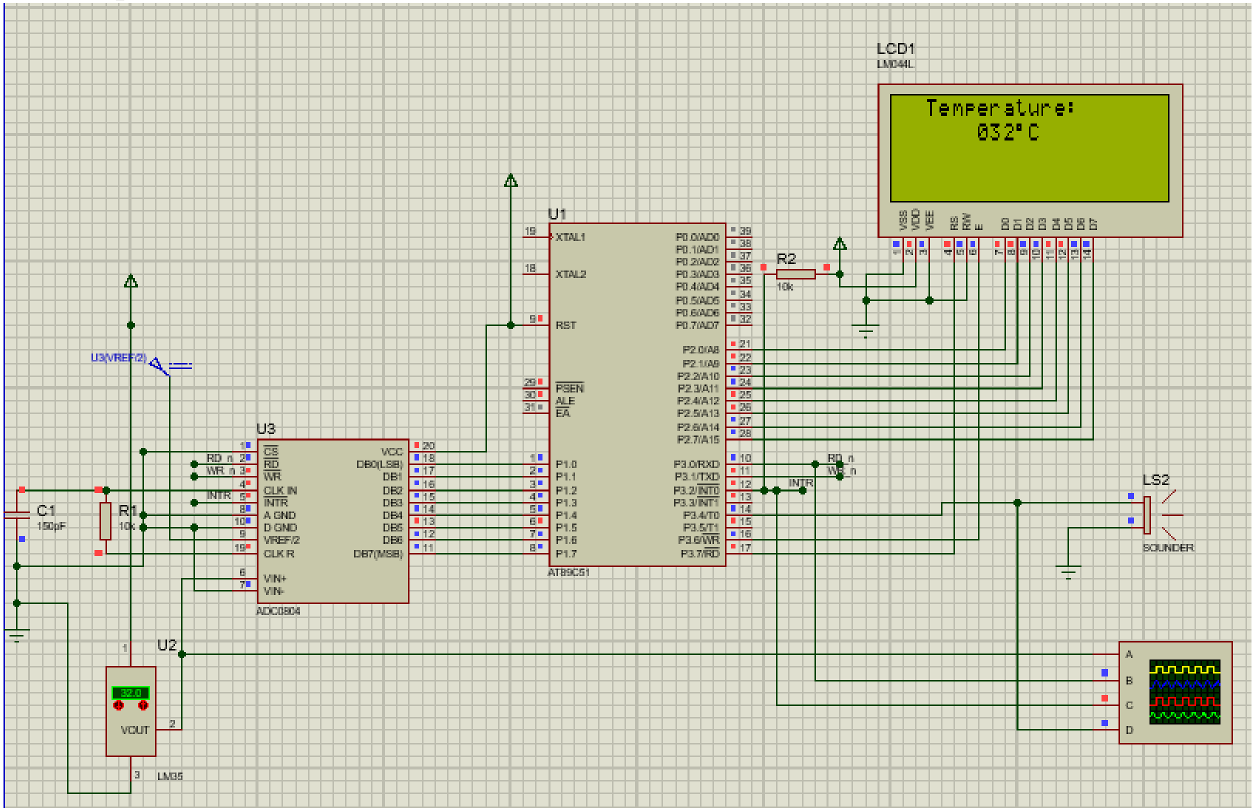

Temperature:
007°C

Oscilloscope Output:

Since the temperature is less than 10 so a sound of 100Hz with a Duty Cycle 25% will be generated.

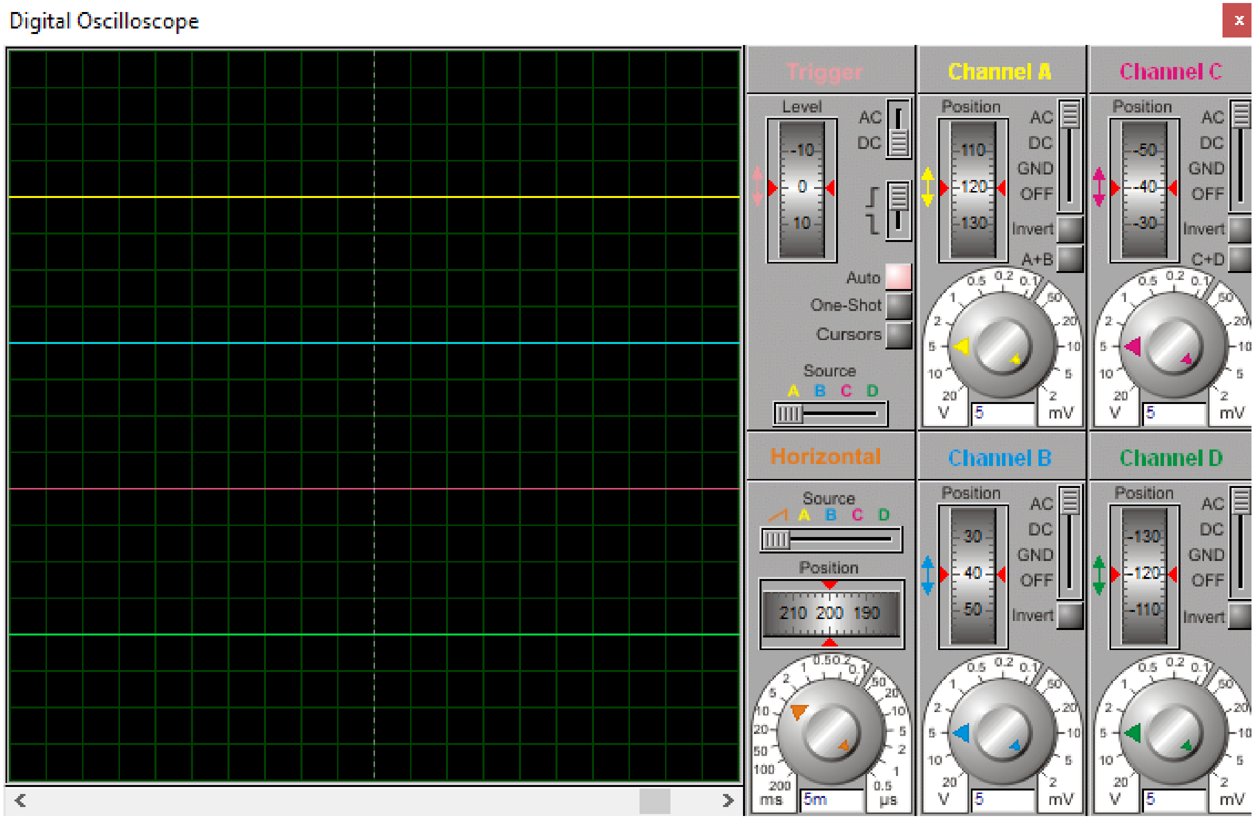


10<Temperature<36:

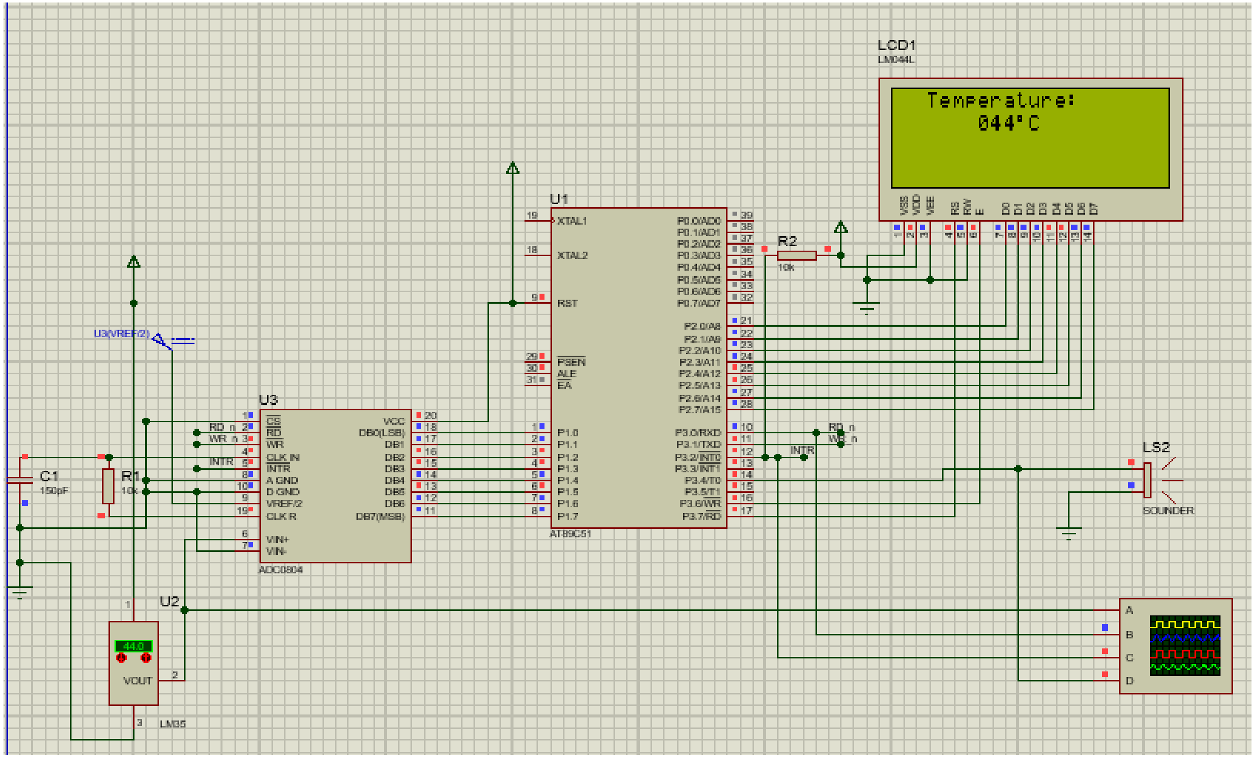


Oscilloscope Output:

No Signal should be generated in this case so a flat line.

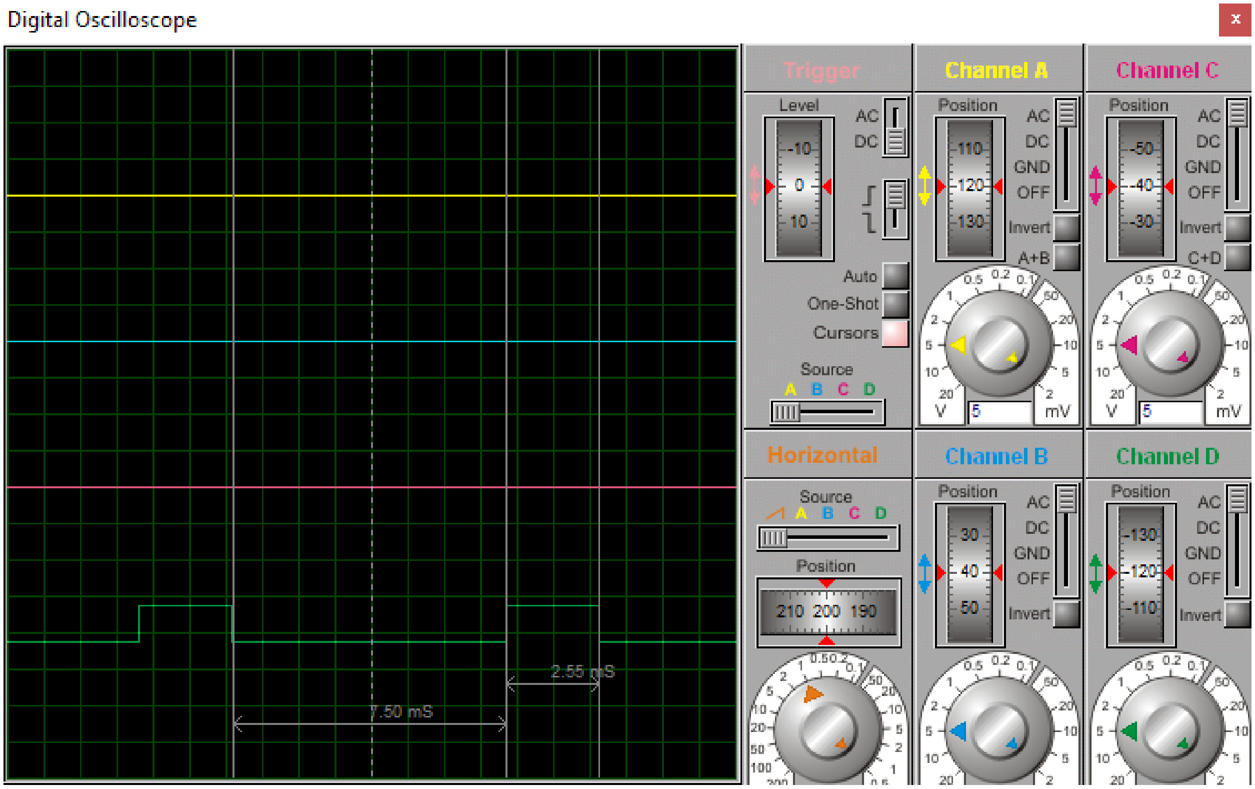


Temperature>36:



Oscilloscope Output:

Since the temperature is greater than 36 so a sound of 100Hz with a Duty Cycle 25% will be generated.


Digital Oscilloscope
x
7.50 mS
2.55 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
210 200 190
Channel B
Channel D
V
mV
5