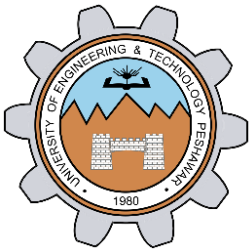
Assignment #08



Spring 2021

Microprocessor Based System Design

Submitted by: **Muhammad Kamran**

Registration No. : **18 PWCSE1737**

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."

Submitted to:

Engr. Bilal Habib

28th June , 2021

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

QUESTION:

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

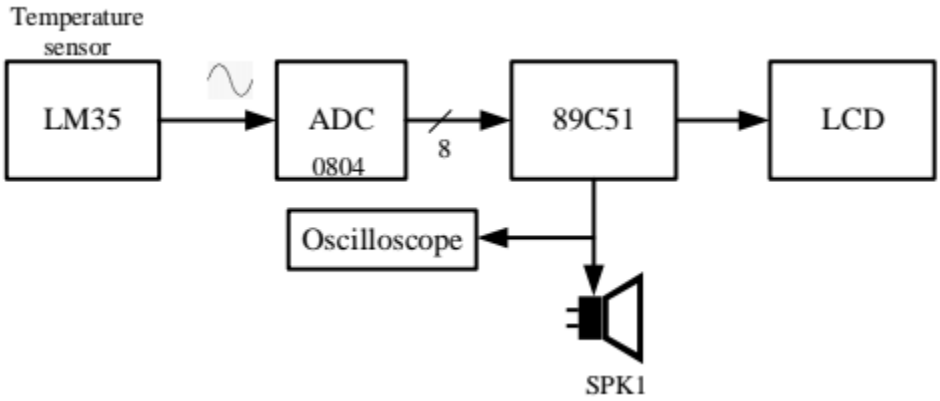


Figure 1.

If temperature > 35C or below 10C then generate a 100Hz beep using speaker/sound card in Proteus. It has a 25% duty cycle and verify it using an oscilloscope.

Sampling rate of ADC = 1K samples/sec.

Solution:

To generate a 100Hz beep using speaker having 25% duty cycle following calculation has been done.

As we know Clock frequency of 8051 microcontroller is 11.0592
So
$F = (11.0592 \times 10^{6}) / 12 = 921600Hz$
Time period for one machine cycle is:
$=1/921600=1.085 \times 10^{-6}s$
Frequency of generating beep speaker=100Hz
Duty cycle=25%
As we know that T=1/f
So T=1/100Hz
=0.01

For Active Time:

Active time = (25/100) (0.01)

Active time=0.0025 s

OFSET value for active time= 0.0025 s / $1.085x10^{-6}$ s

=2304

As we know that in Timer0 mode1 we have maximum value of 65535 so

65535-2304+1=63232

Hex value of 63232 = F700;

For Inactive Time:

Inactive time= (75/100) (0.01)

Inactive time=0.0075 s

OFSET value for inactive time= 0.0075 s / $1.085x10^{-6}$s

=6912

As we know that in Timer0 mode1 we have maximum value of 65535.

65535-6912+1=58624

Hex value of 58624=E500;

To create a digital signal having Sampling rate of ADC = 1K samples/sec following calculations has been done:

As we know that in ADC0804 the minimum time taken by ADC to convert a single sample is 110µs. To create a sampling rate of 1K sample/sec

Sampling Time Period Ts=1000µs

Sampling frequency = 1 / Ts

=1/1000 µs

= 1000 sample / s

As we know that f= 1 / (1.1RC)

Lets C=150pF

R=1 / [(1.1)($150x10^{-12}$)(1000)]

R= 6060606

Code:

```
#include <reg51.h>
#define input P1;
```

```
double newtemp,pass1,pass2,T;
sbit speaker = P3^5;
sbit rs = P3^0;
sbit rw = P3^1;
sbit e = P3^2;
sbit RD_=P3^7;
sbit WR_=P3^6;
sbit INTR=P3^4;

void Active_value_Delay()
{
TMOD=0x01;
TH0=0xF7;
TL0=0x00;
TR0=1;
while(TF0==1){
TR0=0;
TF0=0;}
}

void Delay(unsigned int value)
{
int i,j ;
for(i=0;i<value;i++)
for(j=0;j<1275;j++);
}
double ADC()
{
double temp;
RD_=1;
WR_=0;
Delay(1);
WR_=1;
while(INTR==1);
{RD_=0;
temp=input;
Delay(3);}
```

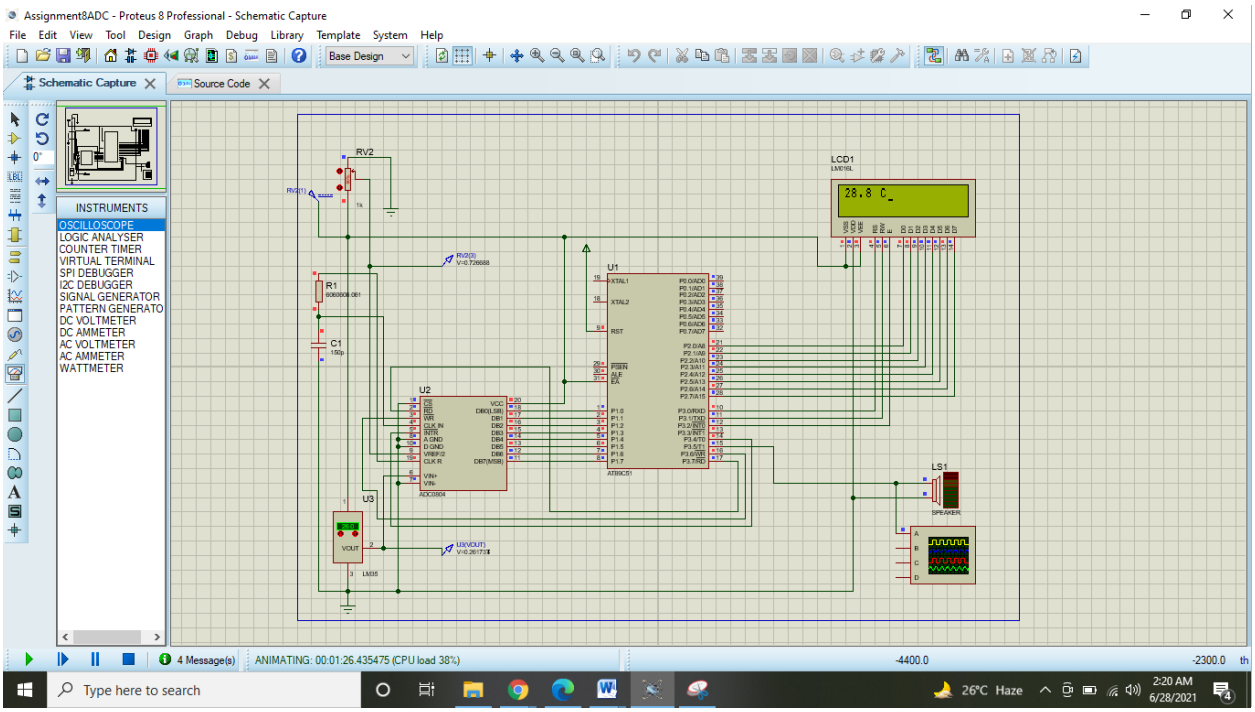
```
return temp;
}
void LCD_Command(unsigned char item)
{
P2 = item;
rs= 0;
rw=0;
e=1;
Delay(1);
e=0;
}
void LCD_Data(double item)
{
P2 = item;
rs= 1;
rw=0;
e=1;
Delay(1);
e=0;
}
void Display_Temperature(double Num)
{
unsigned char UnitDigit  = 0;
unsigned char TenthDigit = 0;
unsigned char HundDigit = 0;
unsigned char decimal=0;
int point;
point=Num*10;
HundDigit=(Num/100);
if( HundDigit != 0)
LCD_Data(HundDigit+0x30);
TenthDigit = Num - HundDigit*100;
TenthDigit = TenthDigit/10;
if (HundDigit==0 && TenthDigit==0){}
else
LCD_Data(TenthDigit+0x30);
UnitDigit = Num - HundDigit*100;
```

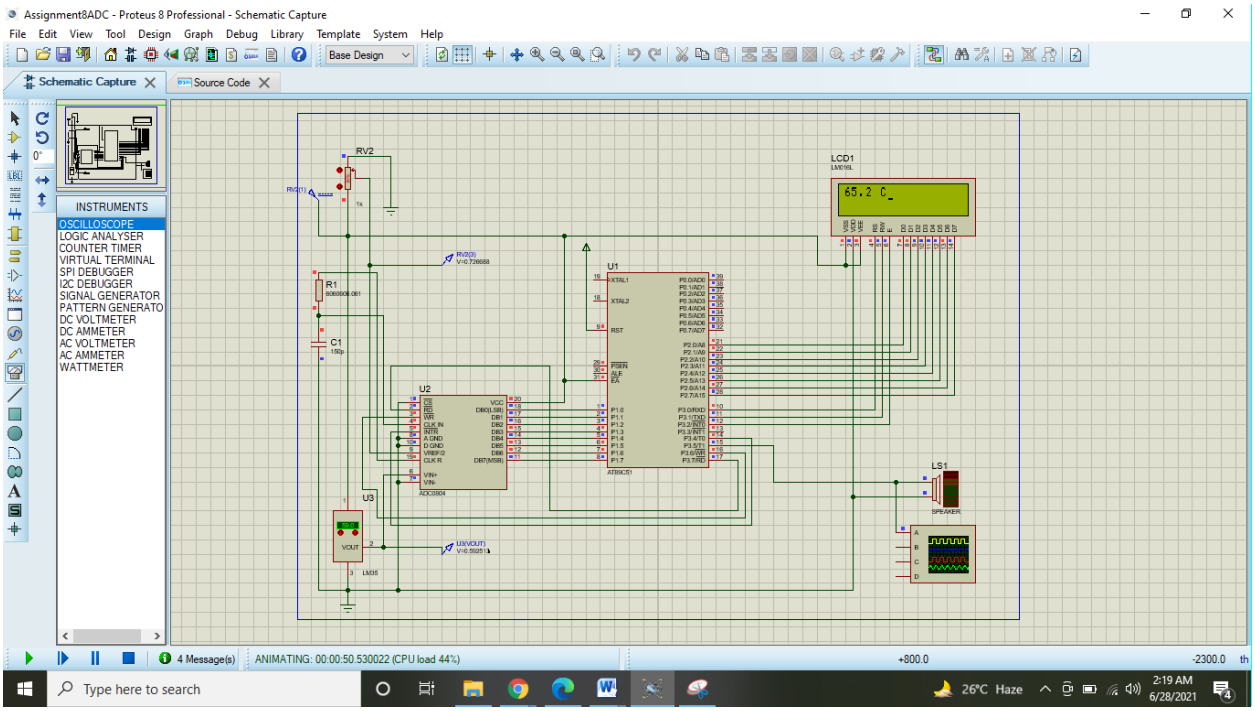
```
UnitDigit = UnitDigit - TenthDigit*10;
LCD_Data(UnitDigit+0x30);
LCD_Data('.');
decimal=(point%10);
LCD_Data (decimal+0x30);
LCD_Data(' '); LCD_Data('C');
}
void read(){
LCD_Command(0x0E);
LCD_Command(0x01);
LCD_Command(0x06);
LCD_Data('R');LCD_Data('E');LCD_Data('A');LCD_Data('D');LCD_Data('I');LCD_Data('N');LCD_Data
('G');LCD_Data(' ');
}
void main()
{
P0=0x00;
read();
while(1){
newtemp=ADC();
Delay(60);
pass1=ADC();
Delay(60);
  if (newtemp==pass1){
      break;
          }
}
while(1){
T=160;
newtemp=(((newtemp*T)/255));
if(newtemp>35 || newtemp<10)
{
  speaker=1;
  Active_value_Delay();
  speaker=0;
}
LCD_Command(0x0E);
```

```
LCD_Command(0x01);
LCD_Command(0x06);
Display_Temperature(newtemp);
Delay(300);

while(1){
newtemp=ADC();
Delay(60);
pass1=ADC();
Delay(60);
pass2=ADC();
  if (newtemp==pass1){
    if(pass1==pass2){
            break;                  }
        }
}
}}
```

Schematic:





Oscilloscope Output: