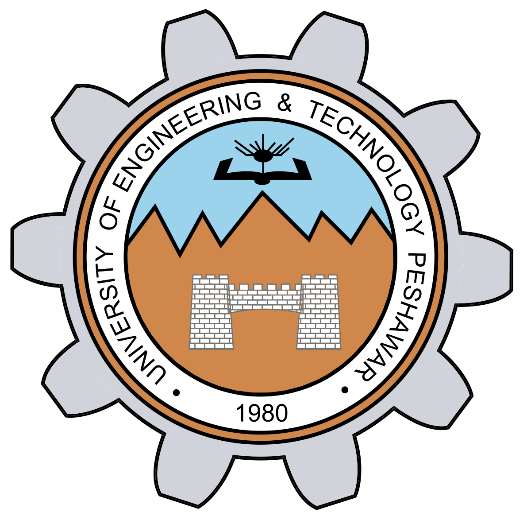
**Lab:12**

**Interfacing ADC with 8051 Microcontroller**



**MBSD Lab**

**Spring 2023**

**Submitted by:**

**Maaz Habib**

**Reg no:**

**20Pwcse1952**

“On my honour, as a student of University of Engineering and Technology Peshawar, I have neither nor received unauthorized assistance on this academic work”

**Submitted to:**

**Dr: Amaad Khalil**

**Task1: Interface ADC with 8051 microcontroller with 2 input Analogue signals.**

**Answer:**

#include <reg51.h>

#define adc\_port P1 //ADC Port

#define rd P3\_6 //Read signal P1.0

#define wr P3\_5 //Write signal P1.1

#define intr P3\_4 //INTR signal P1.3

void conv(); //Start of conversion function

void read(); //Read ADC function

unsigned char adc\_val;

int i;

void main()

{

while (1)

{ //Forever loop

conv(); //Start conversion

read(); //Read ADC

P2 = adc\_val; //Send the read value to P3

for(i=0;i<100;i++)

{

}

}

}

void conv()

{

wr = 0; //Make WR low

wr = 1; //Make WR high

while (intr); //Wait for INTR to go low

}

void read()

{

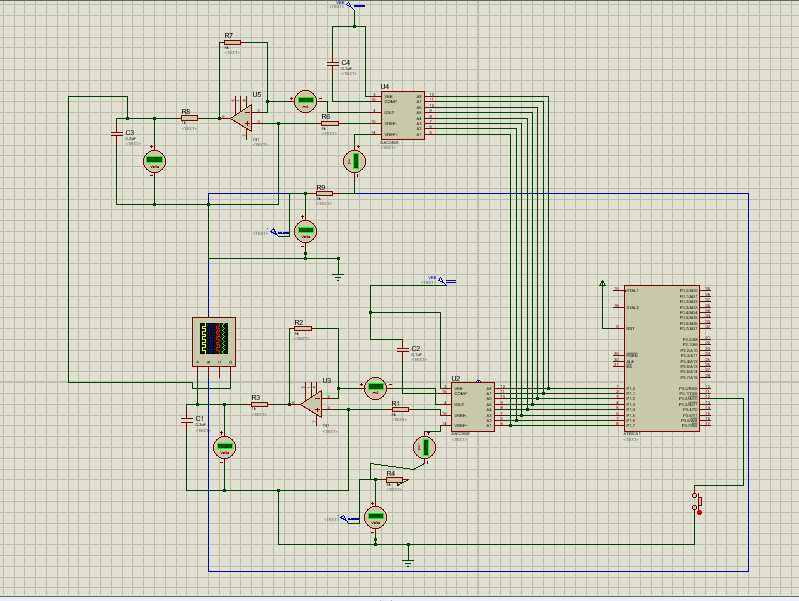
rd = 0; //Make RD low

adc\_val = adc\_port; //Read ADC port

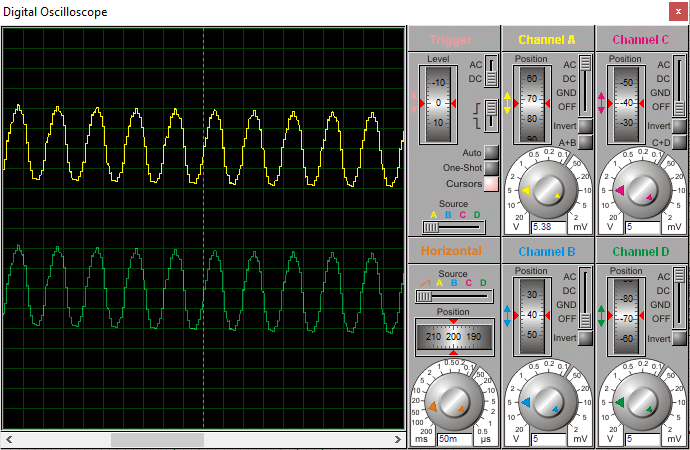
rd = 1; //Make RD high

}

**Schematic:**



**Output:**



**Task # 02**

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

**Code:**

#include <reg51.h>

#define input P1; //when converted from analog to DC goes to p1

double newtemp;

char result\_char;

int x=0;

//LCD pins

sbit reg\_select = P3^7; //register select pin

sbit enable = P3^6; //enable pin

//ADC pins

sbit rd=P3^4; //defines rd pin of ADC use for reading purposes

sbit wr=P3^5; // define wr pin of ADC use for writing purposes

sbit intr=P3^2; //defines intr pin use for sending interrupts

sbit buzzer=P3^0; //used for buzzer on off

//functions

void Delay(unsigned int Time); //Function to provide time delay in msec.

double ADC(); // Function to read the values from ADC and send to controller.

void write\_command(unsigned char item); //Function to send commands to LCD see command tables in LCD Link

void write\_data(double item); //Function to send data to LCD

void display\_temp(double num); //displays number on LCD

void read\_temp(); // Displays "temp" while controller reads from ADC

void main()

{

P0=0x00; //intialize port 0 to low use while controller reads the temperature from

while(1){ //enters in the permanent loop

buzzer=1;

read();

newtemp=adc(); //reads first value from ADC

display\_temp(newtemp); //show temperature

Delay(400);

//waits 400millisec before re-measure the value of temperature

if(newtemp>35.0 || newtemp<10.0)

{

buzzer=0; //buzzer on

Delay(3); //for 3millisec

buzzer=1; //buzzer off

Delay(7); //for 7 millisec

buzzer=0; //just to check the time;

buzzer=1;

}

else

{

buzzer=1;

}

} // end while permanent loop

} // end main loop

//functions definition

void Delay(unsigned int time) //Function to provide time delay in msec.

{

int i,j ;

for(i=0;i<time;i++)

for(j=0;j<1275;j++); //10ms

}

double ADC() // Function to read the values from ADC and send to controller.

{

double temp;

rd=1; //high to low to read from adc

wr=0; //low to high to write on adc

Delay(1);

wr=1;

while(intr==1); //interrupt is low active

rd=0;

temp=input; //getting the converted digital value

Delay(3);

return temp;

}

void write\_command(unsigned char item) //Function to send commands to LCD see

command tables in LCD Link

{

P2 = item; //Data transfer to P2

reg\_select= 0; // This is for command

enable=1; //send to high to low pulse while writing

Delay(1);

enable=0;

Delay(1);

}

void write\_data(double item) //Function to send data to LCD

{

P2 = item;

reg\_select= 1; // This is for data

enable=1; //send high to low pulse while writing

Delay(1);

enable=0;

}

void display\_temp(double number) //displays number on LCD

{

unsigned char unit\_digit = 0; //It will contain unit digit of number

unsigned char tenth\_digit = 0; //It will contain 10th position digit of number

unsigned char hunderd\_digit = 0; //It will contain 100th position digit of number

unsigned char decimal=0; //It will contain the decimal position of number

int point;

point=number\*10;

hunderd\_digit=(number/100);

if( hunderd\_digit != 0) // If it is zero, then don't display

write\_data(hunderd\_digit+0x30); // Make Character of HundDigit and then display it on LCD

tenth\_digit = number - hunderd\_digit\*100; // Findout Tenth Digit

tenth\_digit = tenth\_digit/10;

if (hunderd\_digit==0 && tenth\_digit==0){} // If it is zero, then don't display

else

write\_data(tenth\_digit+0x30); // Make Char of TenthDigit and then display it on LCD

unit\_digit = number - hunderd\_digit\*100;

unit\_digit = unit\_digit - tenth\_digit\*10;

write\_data(unit\_digit+0x30); // Make Char of UnitDigit and then display it on LCD

write\_data('.');

decimal=(point%10);

write\_data (decimal+0x30); // Make Char of Decimal Digit and then display it on LCD

write\_data(' ');

write\_data('C');

}

void read\_temp()

{ // Displays "temp" while controller reads from ADC

write\_command(0x0E); //turn display ON for cursor blinking

write\_command(0x01); //clear screen

write\_command(0x06); //increment cursor

write\_command(0x95);

write\_data('T');

write\_data('e');

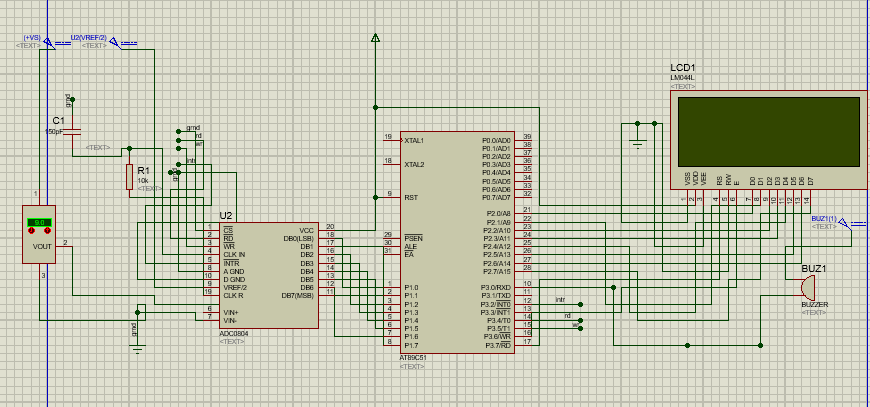
write\_data('m');

write\_data('p');

write\_data(':');

}

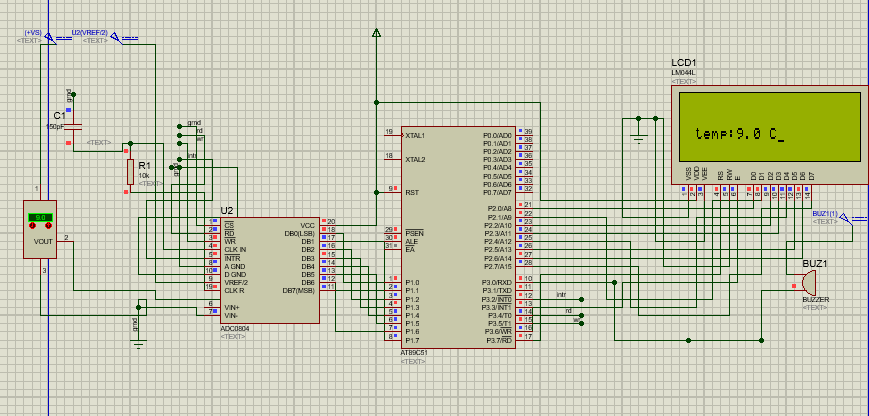
**Schematic:**



**Output:**

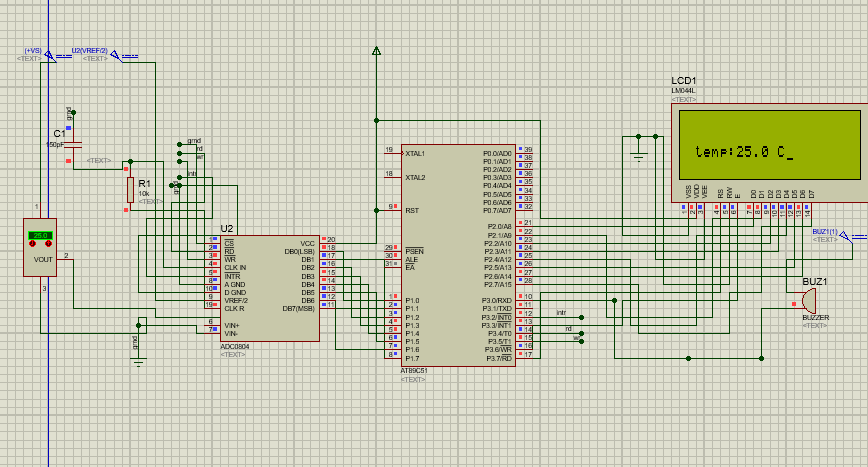
**Temp< 10 C**

* Buzzer on



**10 C<Temp< 35 C**

* Buzzer off



**Temp> 35 C**