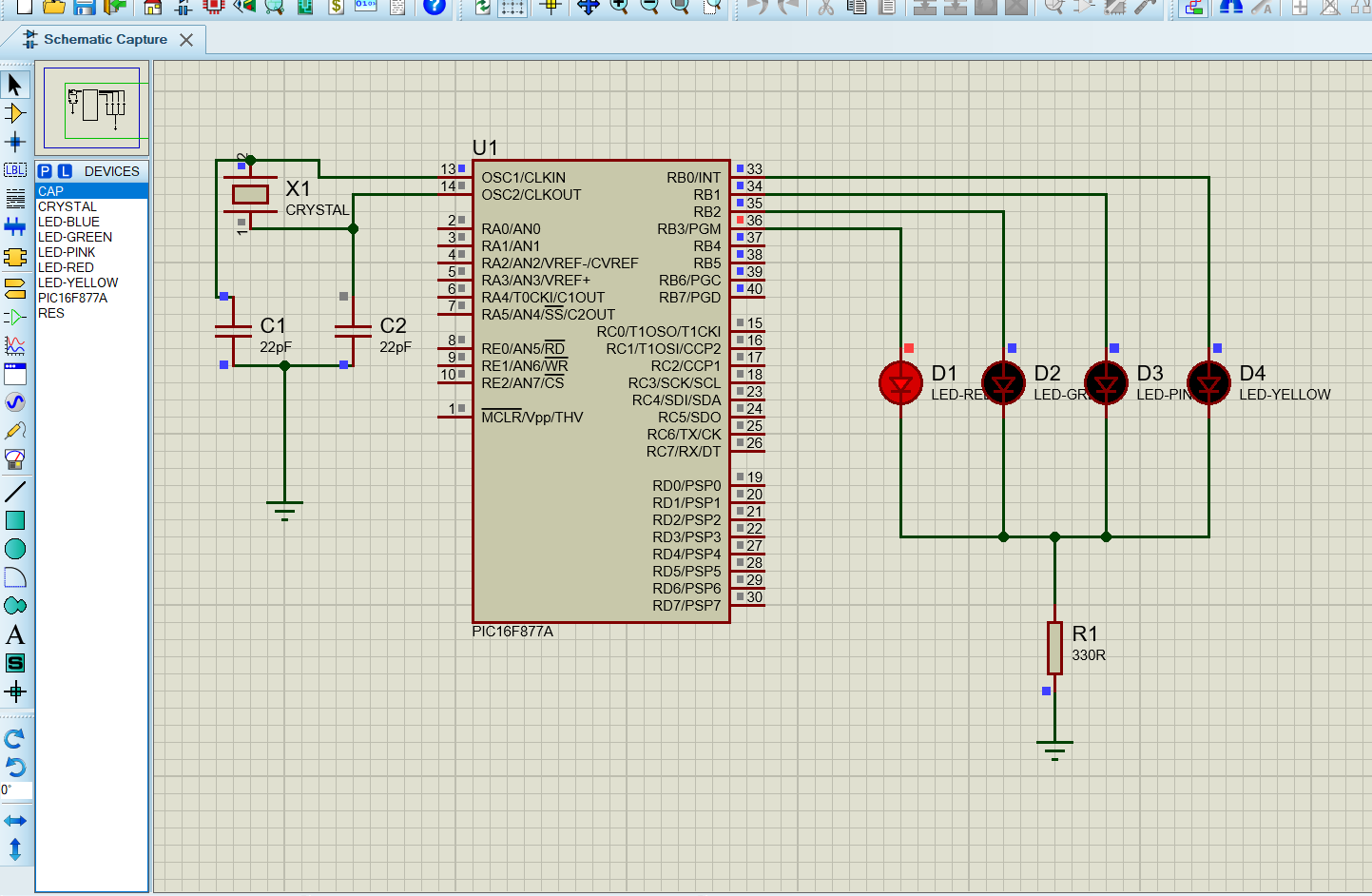
**Circuit Diagram:**



**Fig: LED Blinking**

**Code:**

int counter = 0;

void main() {

TRISB = 0x00;

PORTB = 0x00;

while(1)

{

PORTB.RB3 = 1;

Delay\_ms(300);

PORTB.RB3 = 0;

PORTB.RB2 = 1;

Delay\_ms(300);

PORTB.RB2 = 0;

PORTB.RB1 = 1;

Delay\_ms(300);

PORTB.RB1 = 0;

PORTB.RB0 = 1;

Delay\_ms(300);

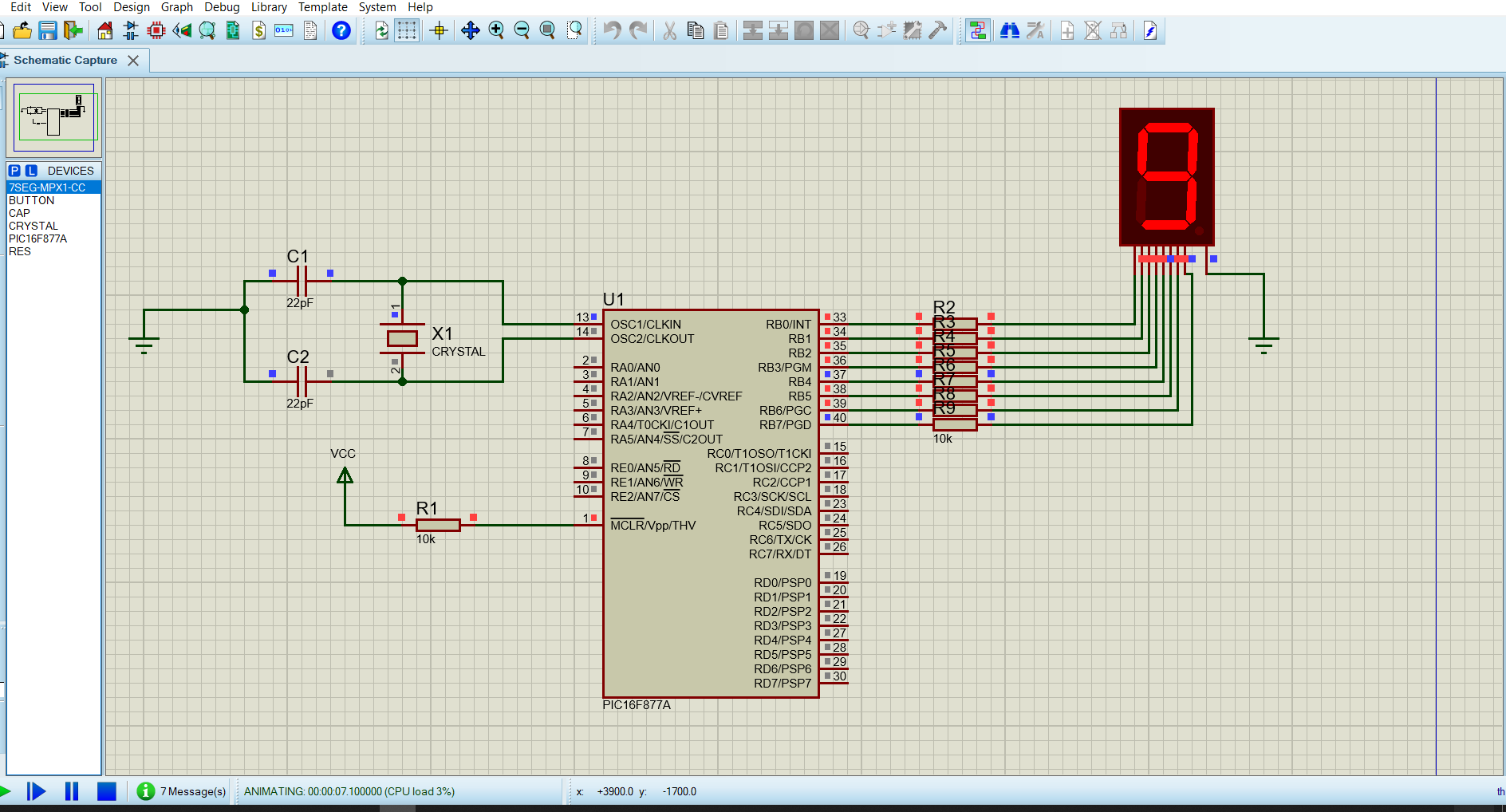
PORTB.RB0 = 0;

counter++;

}

}

**Circuit Diagram:**



**Fig: 7Segment 1Digit Display**

**Code:**

char arraycc[]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

void main() {

int i = 0;

trisb = 0x00;

portb = 0xff;

for(i=0; i<10; i++){

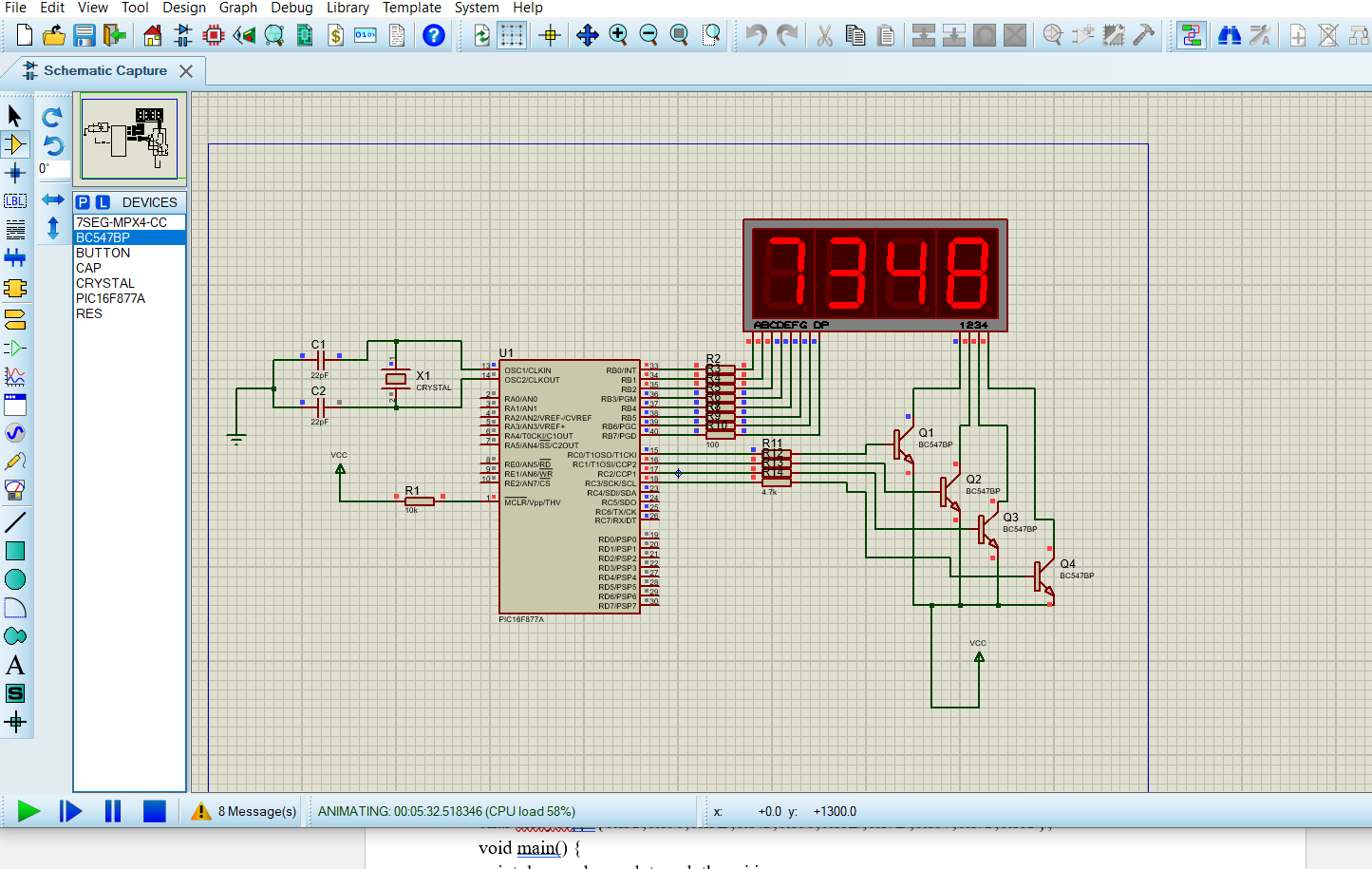
portb = arraycc[i];

delay\_ms(100);

}

}

**Circuit Diagram:**



**Fig: 7-Segment 4Digit Display**

CODE:

char arraycc[]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};

void main() {

int d\_zero,d\_one,d\_two,d\_three,i,j;

Trisb=0x00;

Trisc=0x00;

portb=0x00;

portc=0x00;

while(1)

{

for(i=0;i<=9999;i++) //counting 0-9999

{

d\_zero=i/1000; // for first digit of display

d\_one=((i/100)%10); // for second digit

d\_two=((i/10)%10); // for third digit

d\_three=i%10;

for(j=0;j<=10;j++)

{

portc.f0=0;

portb=arraycc[d\_zero];//display

delay\_ms(1);

portc.f0=1;

portc.f1=0;

portb=arraycc[d\_one];//display

delay\_ms(1);

portc.f1=1;

portc.f2=0;

portb=arraycc[d\_two];//display

delay\_ms(1);

portc.f2=1;

portc.f3=0;

portb=arraycc[d\_three];//display

delay\_ms(1);

portc.f3=1;

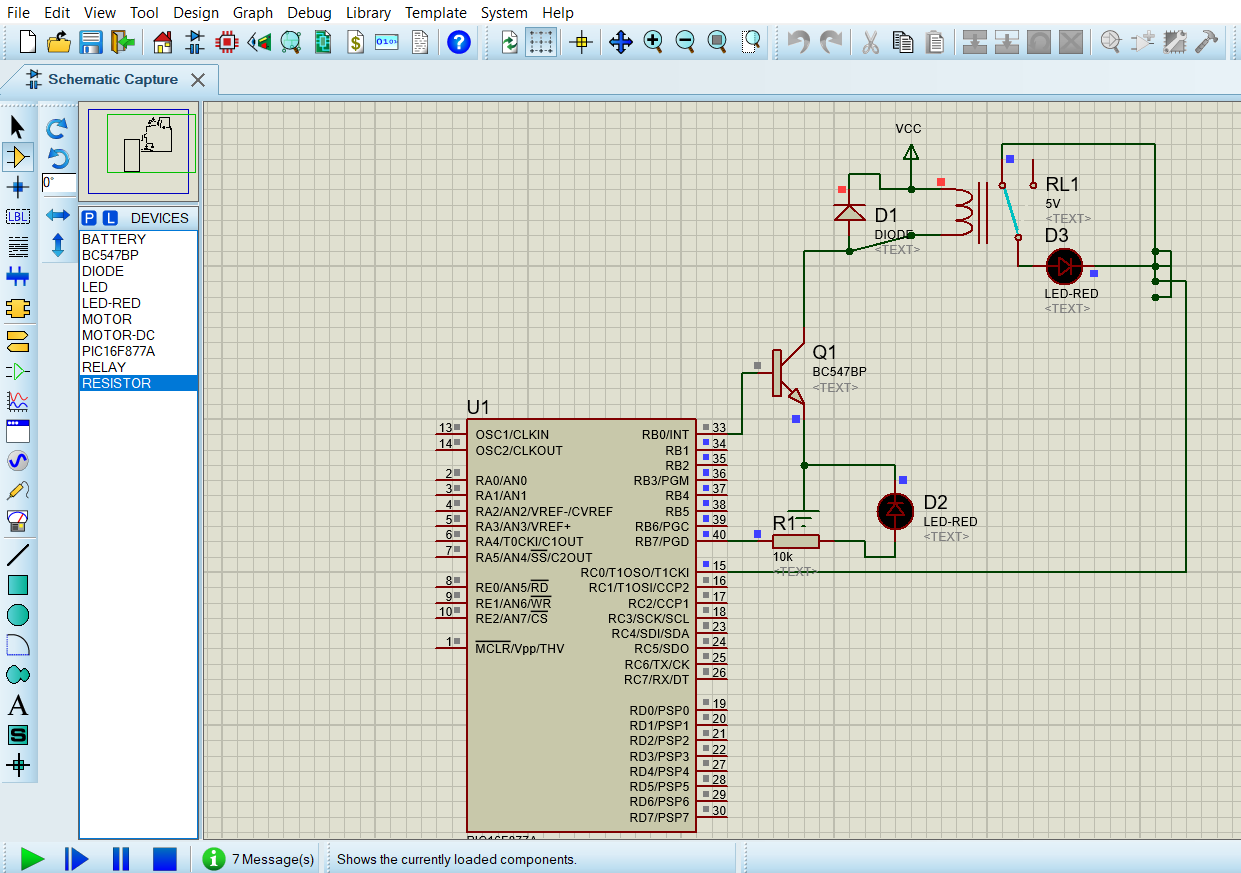
}

}

}

}

**Circuit Diagram:**



**Fig: control AC current by DC current.**

CODE:

void main() {

trisb=0;

portb=0;

while(1){

portb.f0=1;

delay\_ms(1000);

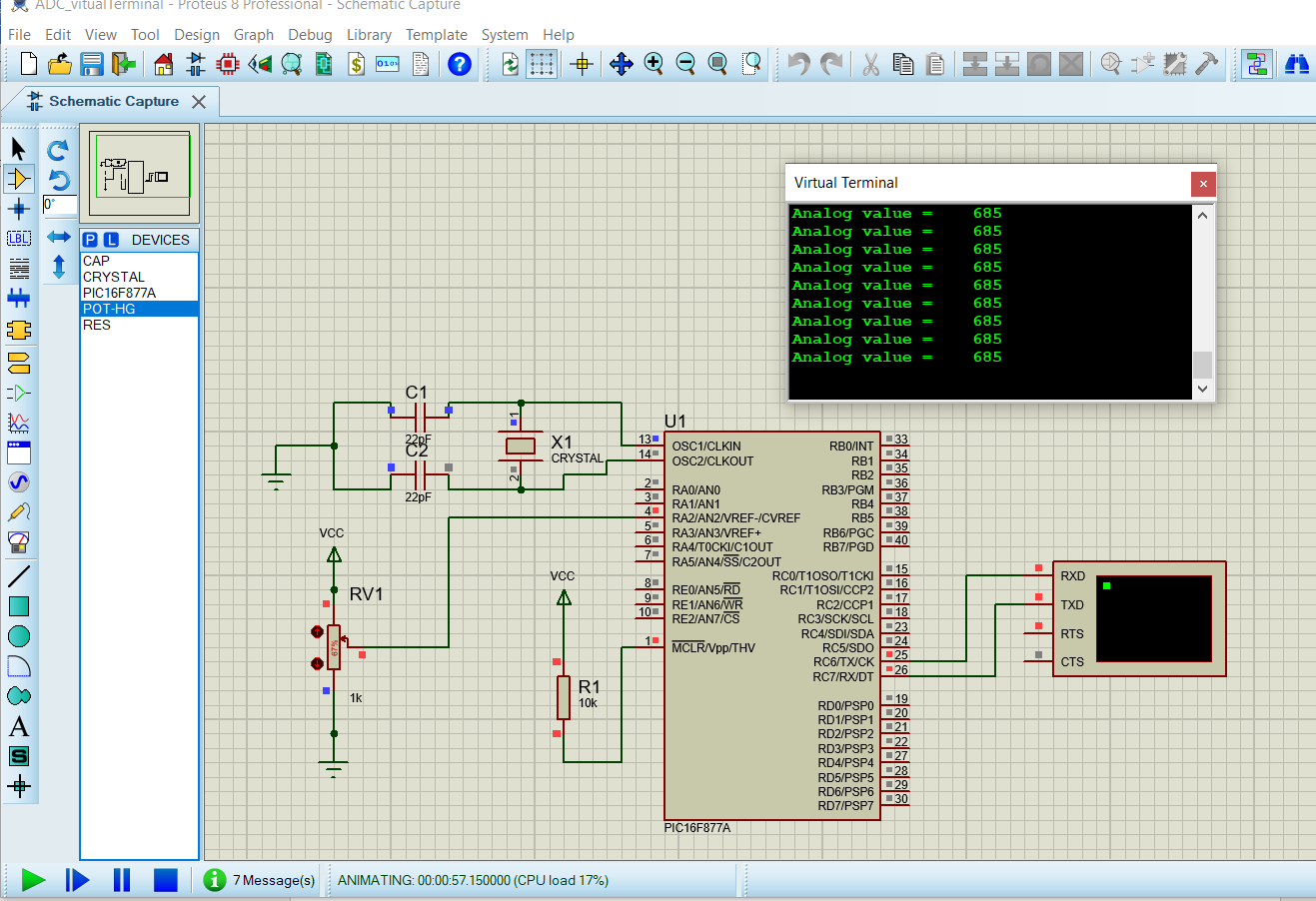
portb.f0=0;

delay\_ms(1000);

}

}

**Circuit Diagram:**



**Fig: Display ADC value in the virtual terminal.**

**CODE:**

int valADC;

char x[4];

void main(){

UART1\_Init(9600);//initialize ADC

ADC\_Init();

while(1){

valADC = ADC\_Read(2);

IntToStr(valADC,x);

UART1\_Write\_Text("Analog value = ");

UART1\_Write\_Text(x);

strcpy(x,"");

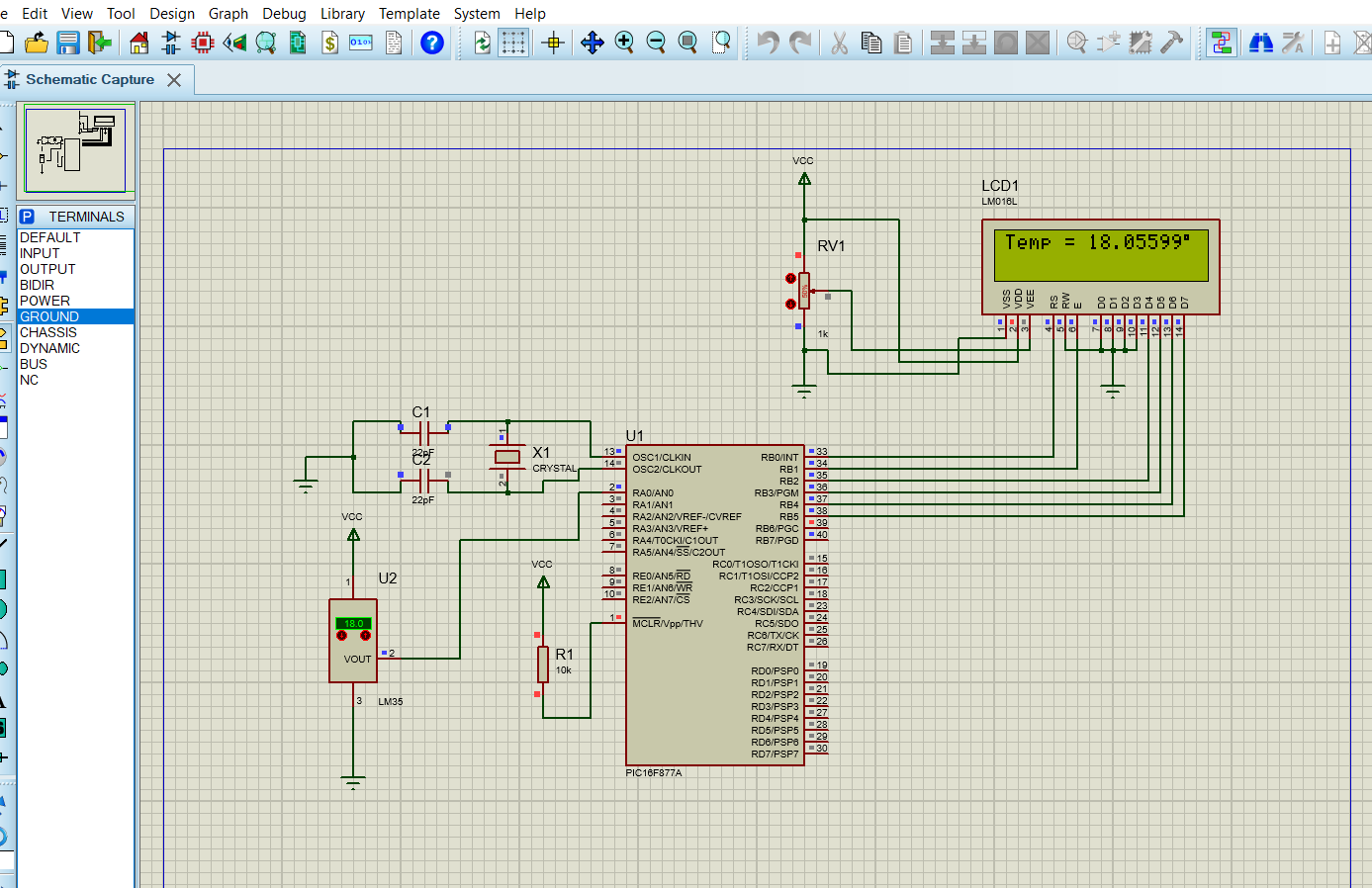
UART1\_Write(13);

Delay\_ms(1000);

}

}

**Circuit Diagram:**



**Fig: LAB-5-LM35-Temperature-Sensor-Interfacing**

**CODE:**

sbit LCD\_RS at RB0\_bit;

sbit LCD\_EN at RB1\_bit;

sbit LCD\_D4 at RB2\_bit;

sbit LCD\_D5 at RB3\_bit;

sbit LCD\_D6 at RB4\_bit;

sbit LCD\_D7 at RB5\_bit;

sbit LCD\_RS\_DIRECTION at TRISB0\_bit;

sbit LCD\_EN\_DIRECTION at TRISB1\_bit;

sbit LCD\_D4\_DIRECTION at TRISB2\_bit;

sbit LCD\_D5\_DIRECTION at TRISB3\_bit;

sbit LCD\_D6\_DIRECTION at TRISB4\_bit;

sbit LCD\_D7\_DIRECTION at TRISB5\_bit;

char display[16]="";

void main()

{

unsigned int result;

float volt,temp;

trisb=0x00;

trisa=0xff;

adcon1=0x80;

lcd\_init();

lcd\_cmd(\_lcd\_clear);

lcd\_cmd(\_LCD\_CURSOR\_OFF);

while(1)

{

result = adc\_read(0);

volt=result\*4.88;

temp=volt/10;

lcd\_out(1,1,"Temp = ");

floattostr(temp,display);

lcd\_out\_cp(display);

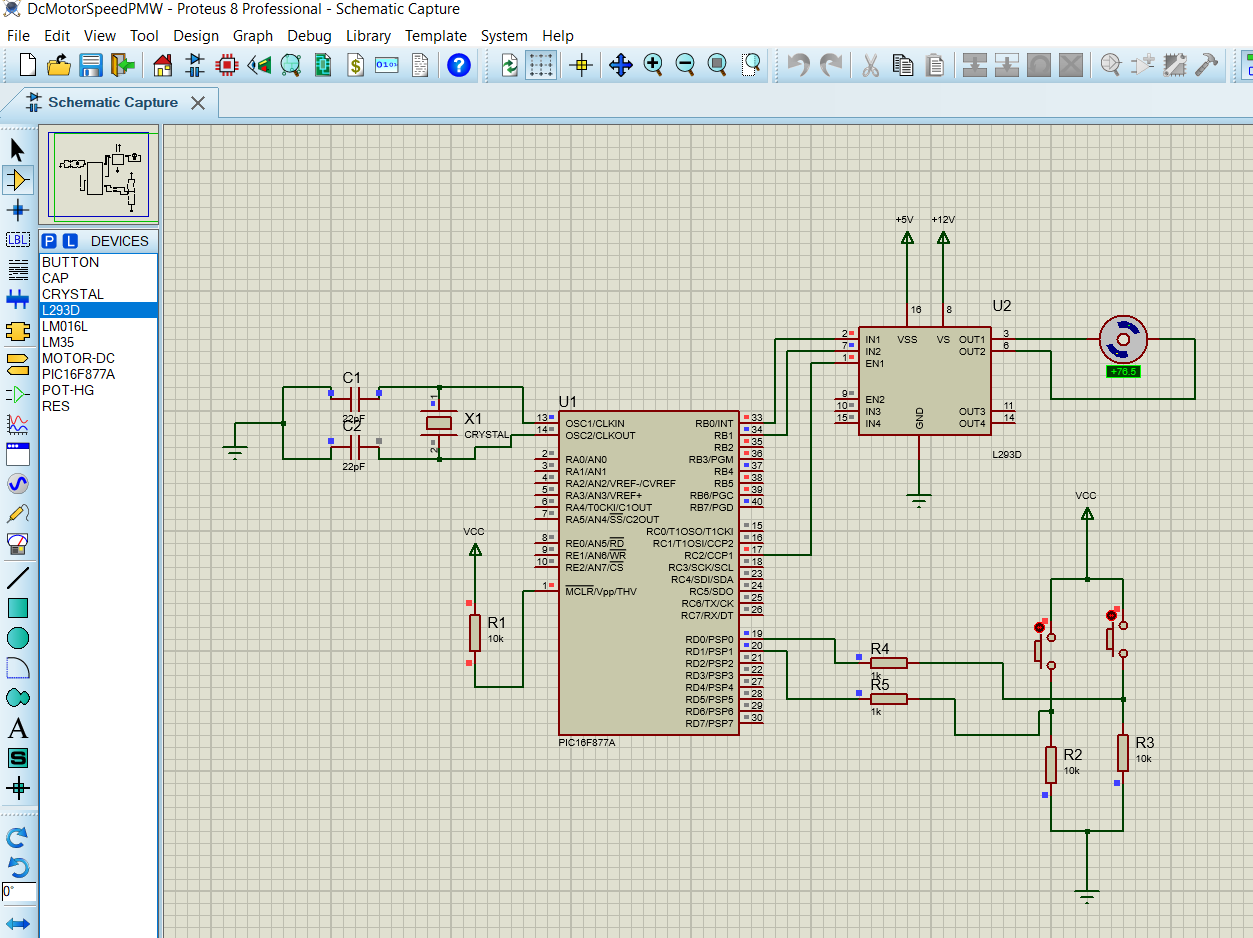
lcd\_chr(1,16,223); //print at pos(row=1,col=13) "°" =223 =0xdf

lcd\_out\_cp(" C"); //celcius

}

}

**Circuit Diagram:**



**Fig: DC Motor Speed Control PWM**

**CODE:**

void main(){

short duty = 0;

TRISD = 0xFF;

TRISB = 0x00;

PORTB.F0=0xff;

PORTB.F1=0x00;

PWM1\_Init(1000);

PWM1\_Start();

PWM1\_Set\_Duty(duty);

while (1) {

if (RD0\_bit && duty<250) {

Delay\_ms(100);

if (RD0\_bit && duty<250) {

duty = duty + 10;

PWM1\_Set\_Duty(duty);

}

}

if (RD1\_bit && duty >0) {

Delay\_ms(100);

if (RD1\_bit && duty >0) {

duty = duty - 10;

PWM1\_Set\_Duty(duty);

}

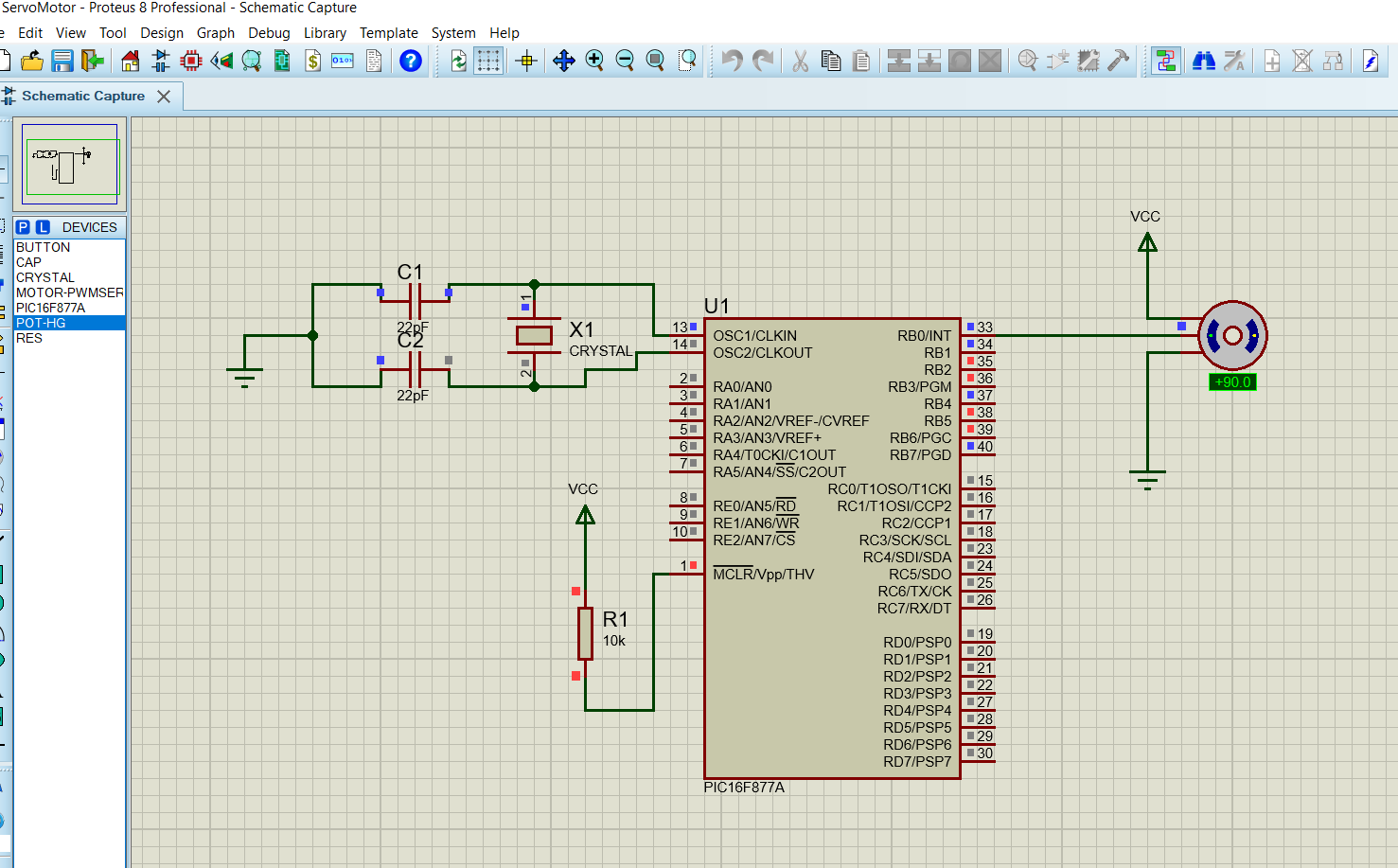
}

Delay\_ms(10);

}

}

**Circuit Diagram:**



**Fig: Interfacing Servo Motor with PIC Microcontroller**

**CODE:**

void servoRotate0()

{

unsigned int i;

for(i=0;i<50;i++)

{

PORTB.F0 = 1;

Delay\_us(800);

PORTB.F0 = 0;

Delay\_us(19200);

}

}

void servoRotate90()

{

unsigned int i;

for(i=0;i<50;i++)

{

PORTB.F0 = 1;

Delay\_us(1500);

PORTB.F0 = 0;

Delay\_us(18500);

}

}

void servoRotate180()

{

unsigned int i;

for(i=0;i<50;i++)

{

PORTB.F0 = 1;

Delay\_us(2200);

PORTB.F0 = 0;

Delay\_us(17800);

}

}

void main()

{

TRISB = 0;

do

{

servoRotate0();

Delay\_ms(2000);

servoRotate90();

Delay\_ms(2000);

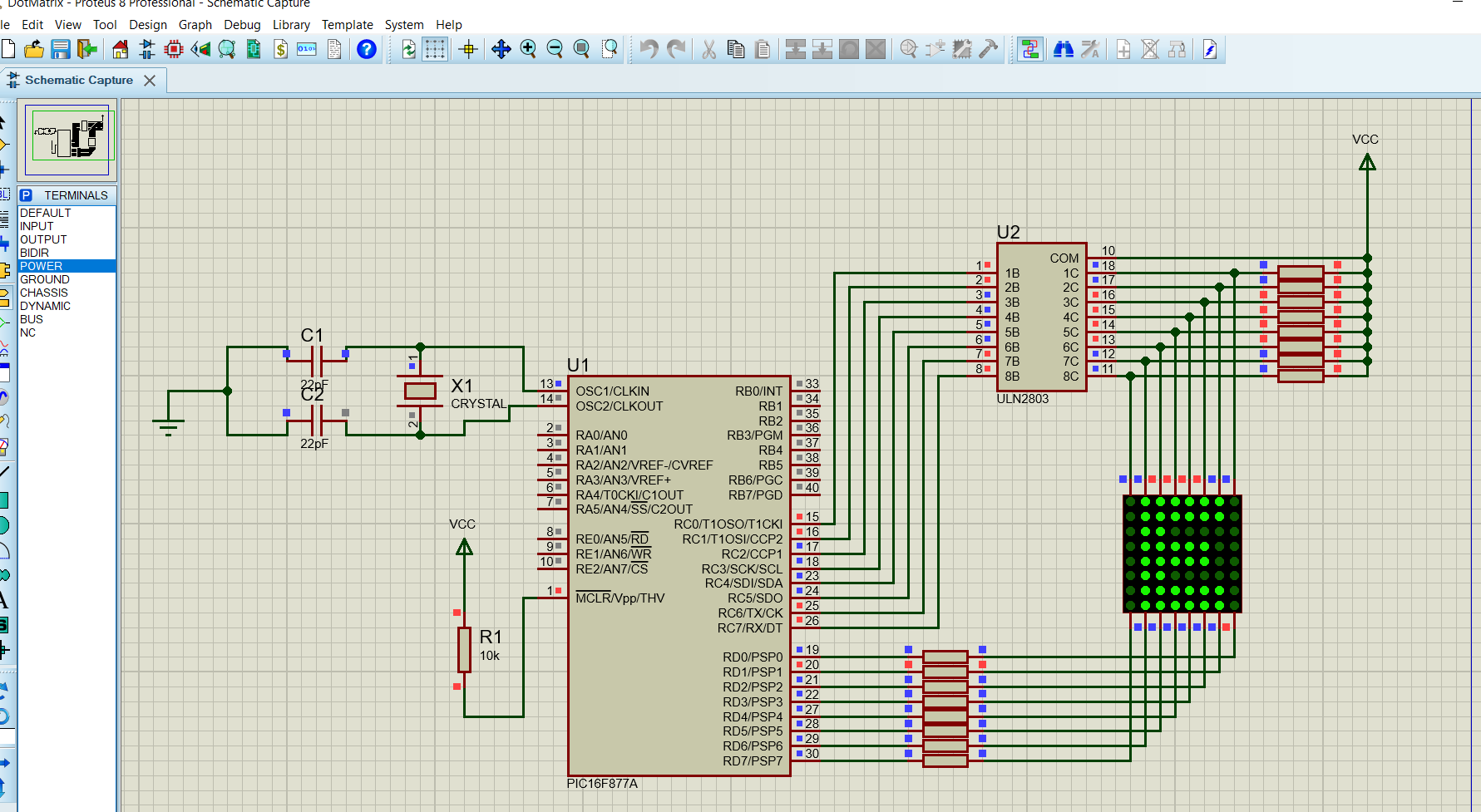
servoRotate180();

}

while(1);

}

**Circuit Diagram:**



**Fig: Dot Matrix Display Interfacing**

**CODE:**

void MSDelay(unsigned char Time)

{

unsigned char y,z;

for(y=0;y<Time;y++)

for(z=0;z<20;z++);

}

void main()

{

TRISC = 0x00;

TRISD = 0x00;

while(1)

{

PORTD = 0x80;

PORTC = 0x00;

MSDelay(10);

PORTD = 0x40;

PORTC = 0xff;

MSDelay(10);

PORTD = 0x20;

PORTC = 0xff;

MSDelay(10);

PORTD = 0x10;

PORTC = 0xdb;

MSDelay(10);

PORTD = 0x08;

PORTC = 0xdb;

MSDelay(10);

PORTD = 0x04;

PORTC = 0xdb;

MSDelay(10);

PORTD = 0x02;

PORTC = 0xc3;

MSDelay(10);

PORTD = 0x01;

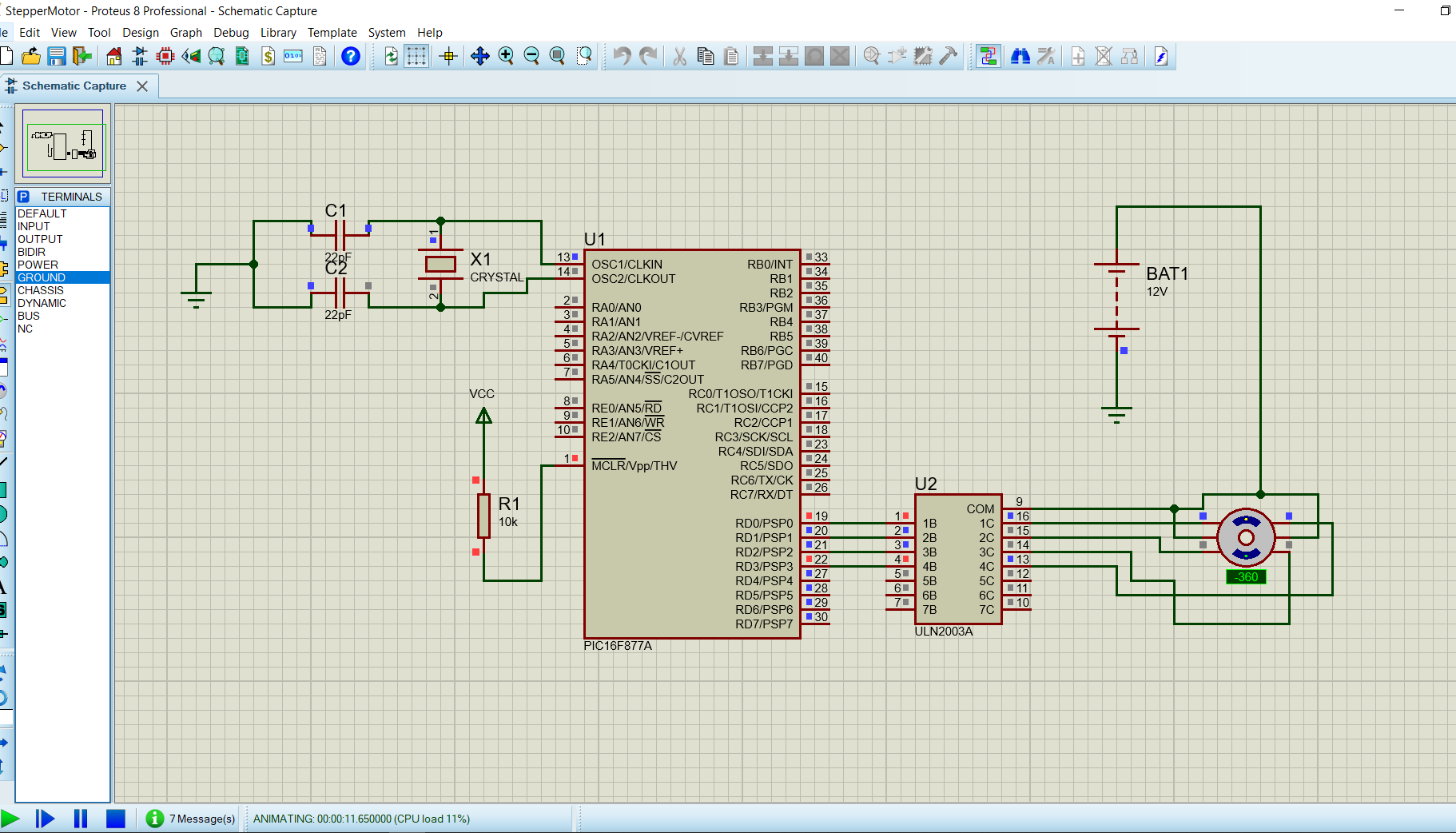
PORTC = 0x00;

MSDelay(10);

}

}

**Circuit Diagram:**



**Fig: Stepper Motor Interfacing**

**CODE:**

void main()

{

TRISD = 0b0000000;

PORTD = 0b1111111;

do

{

PORTD = 0b00000011;

Delay\_ms(500);

PORTD = 0b00000110;

Delay\_ms(500);

PORTD = 0b00001100;

Delay\_ms(500);

PORTD = 0b00001001;

Delay\_ms(500);

}

while(1);

}