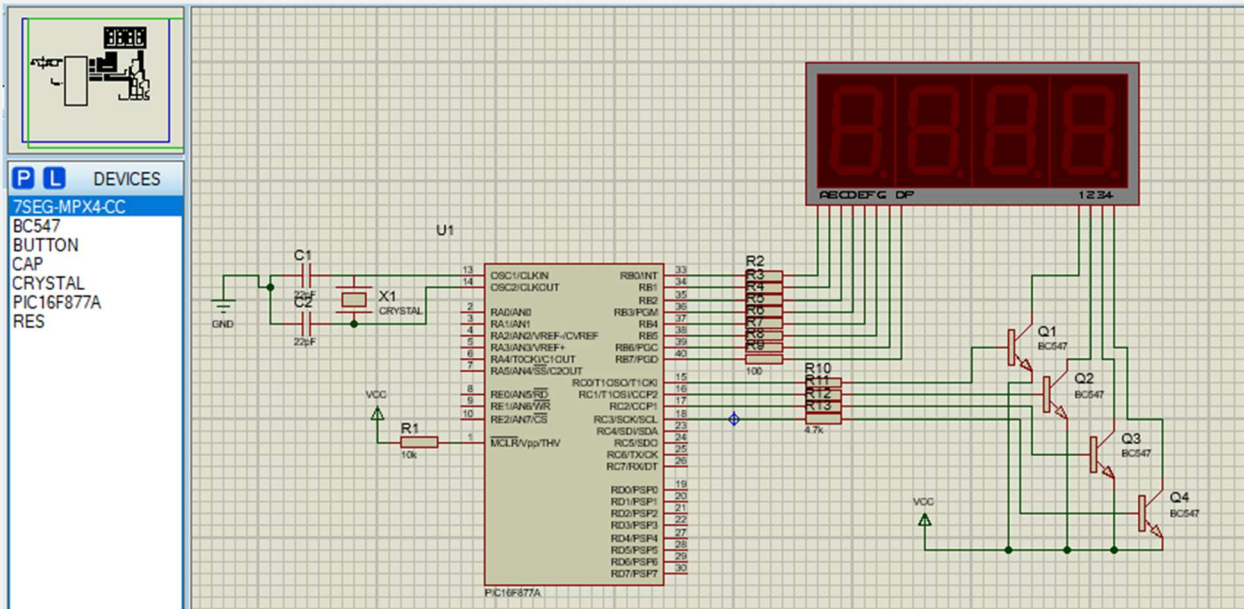


P-1: Display 7250 to 7300 sequentially after one second duration using 4 digits seven segment display.



```
char arraycc[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F}; // 0-9
```

```
int i;
```

```
void main() {
```

```
    int number = 7250; // Starting number
```

```
    int digits[4];     // Store individual digits
```

```
    TrisB = 0x00; // 7-segment data port
```

```
    TrisC = 0x00; // Digit select port
```

```
    PORTB = 0x00;
```

```
    PORTC = 0x00;
```

```
    while(1) {
```

```
        // Split number into digits
```

```
        digits[0] = number / 1000;    // Thousands
```

```
        digits[1] = (number / 100) % 10; // Hundreds
```

```
        digits[2] = (number / 10) % 10; // Tens
```

```
        digits[3] = number % 10;      // Units
```

```
        // Multiplex 4-digit 7-segment display
```

```
        for( i=0; i<100; i++) { // Repeat to maintain display (~1s)
```

```
            PORTC.F0 = 0;
```

```
            PORTB = arraycc[digits[0]];
```

```
            delay_ms(3);
```

```
            PORTC.F0 = 1;
```

```

    PORTC.F1 = 0;
    PORTB = arraycc[digits[1]];
    delay_ms(3);
    PORTC.F1 = 1;

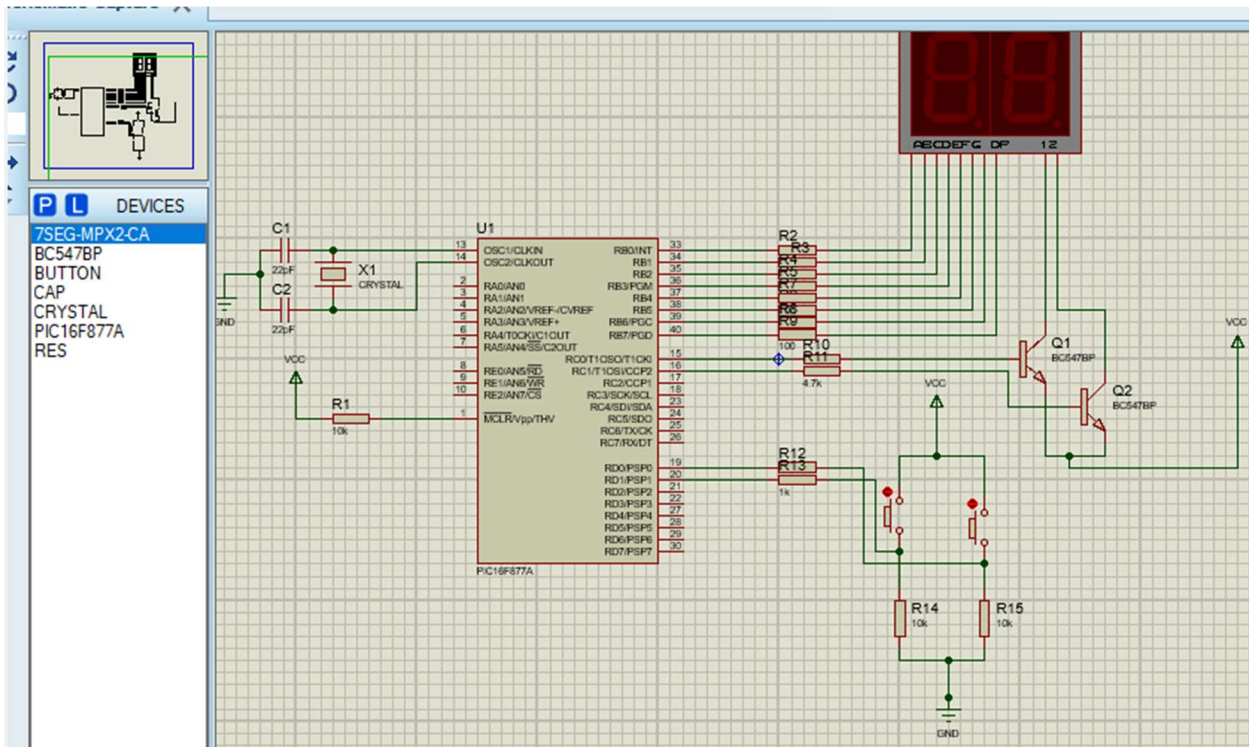
    PORTC.F2 = 0;
    PORTB = arraycc[digits[2]];
    delay_ms(3);
    PORTC.F2 = 1;

    PORTC.F3 = 0;
    PORTB = arraycc[digits[3]];
    delay_ms(3);
    PORTC.F3 = 1;
}

number++; // Increment number
if(number > 7300) {
    number = 7250; // Reset to start
}
}
}

```

P-2: Write a program for PIC microcontroller to increase and decrease a number using push button and 2 digits seven segment display.



```
char arrayca[] = { 0x40, 0x79, 0x24, 0x30, 0x19, 0x12, 0x02, 0x78, 0x00, 0x10 }; // 0-9 segment codes
```

```
int tens,ones;
```

```
void main() {
```

```
    int number = 0; // 0-99 number
```

```
    // Set port directions
```

```
    TrisB = 0x00; // PortB as output (data for 7-segment)
```

```
    TrisC = 0x00; // PortC as output (digit select)
```

```
    TrisD = 0x03; // PortD D0,D1 as input (buttons)
```

```
    // Initialize ports
```

```
    PortB = 0x00;
```

```
    PortC = 0x00;
```

```
    while (1) {
```

```
        // ---- Button 1 pressed ? increment number ----
```

```
        if (PortD.F0 == 1) {
```

```
            Delay_ms(150); // Debounce
```

```
            if (PortD.F0 == 1) {
```

```
                number++;
```

```
                if (number > 99) number = 0; // Wrap 0-99
```

```
            }
```

```
        }
```

```

// ---- Button 2 pressed ? decrement number ----
if (PortD.F1 == 1) {
    Delay_ms(150); // Debounce
    if (PortD.F1 == 1) {
        number--;
        if (number < 0) number = 99; // Wrap 0-99
    }
}

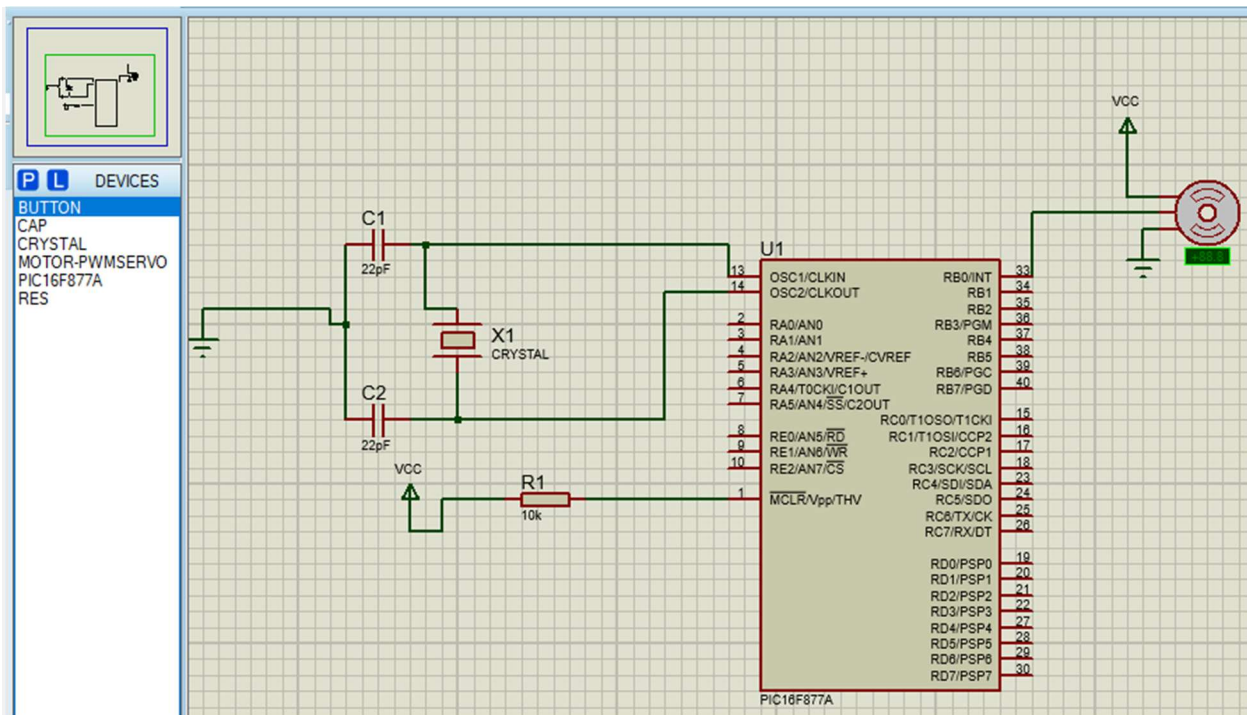
// ---- Separate digits ----
tens = number / 10;
ones = number % 10;

// ---- Display tens digit ----
PortC.F0 = 1;    // Enable first 7-segment
PortB = arrayca[tens];
Delay_ms(5);
PortC.F0 = 0;    // Disable first 7-segment

// ---- Display ones digit ----
PortC.F1 = 1;    // Enable second 7-segment
PortB = arrayca[ones];
Delay_ms(5);
PortC.F1 = 0;    // Disable second 7-segment
}
}

```

P-3: Write a program for controlling a Servo Motor between 0°–180°.



```

void servoRotate0() //0 Degree
{
    unsigned int i;
    for(i=0;i<50;i++)
    {
        PORTB.F0 = 1;
        Delay_us(800);
        PORTB.F0 = 0;
        Delay_us(19200);
    }
}

void servoRotate90() //90 Degree
{
    unsigned int i;
    for(i=0;i<50;i++)
    {
        PORTB.F0 = 1;
        Delay_us(1500);
        PORTB.F0 = 0;
        Delay_us(18500);
    }
}

void servoRotate180() //180 Degree
{
    unsigned int i;

```

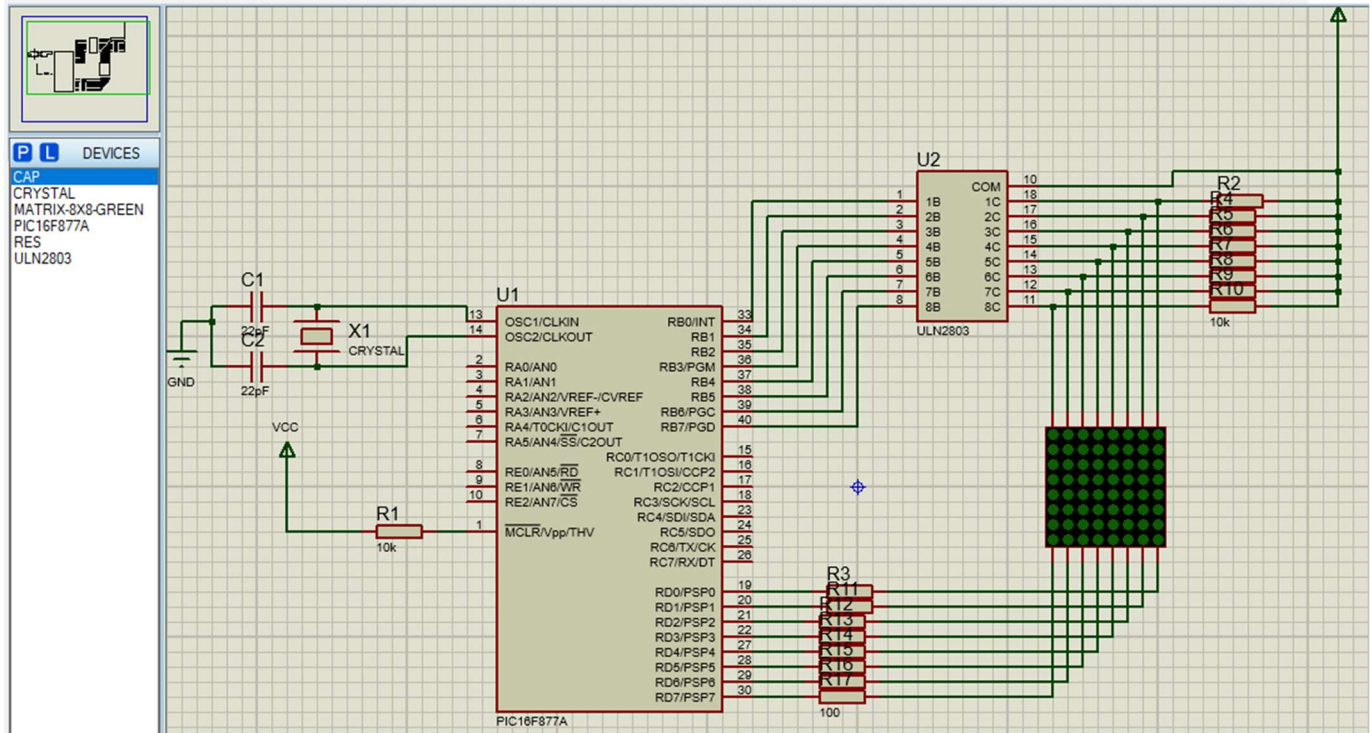
```

for(i=0;i<50;i++)
{
    PORTB.F0 = 1;
    Delay_us(2200);
    PORTB.F0 = 0;
    Delay_us(17800);
}
}

void main()
{
    TRISB = 0; // PORTB as Ouput Port
do
{
    servoRotate0(); //0 Degree
    Delay_ms(2000);
    servoRotate90(); //90 Degree
    Delay_ms(2000);
    servoRotate180(); //180 Degree
}
while(1);
}

```

P-4: Write a program to display letter "R" in dot matrix display.



```

void MSDelay(unsigned char Time)
{

```



```

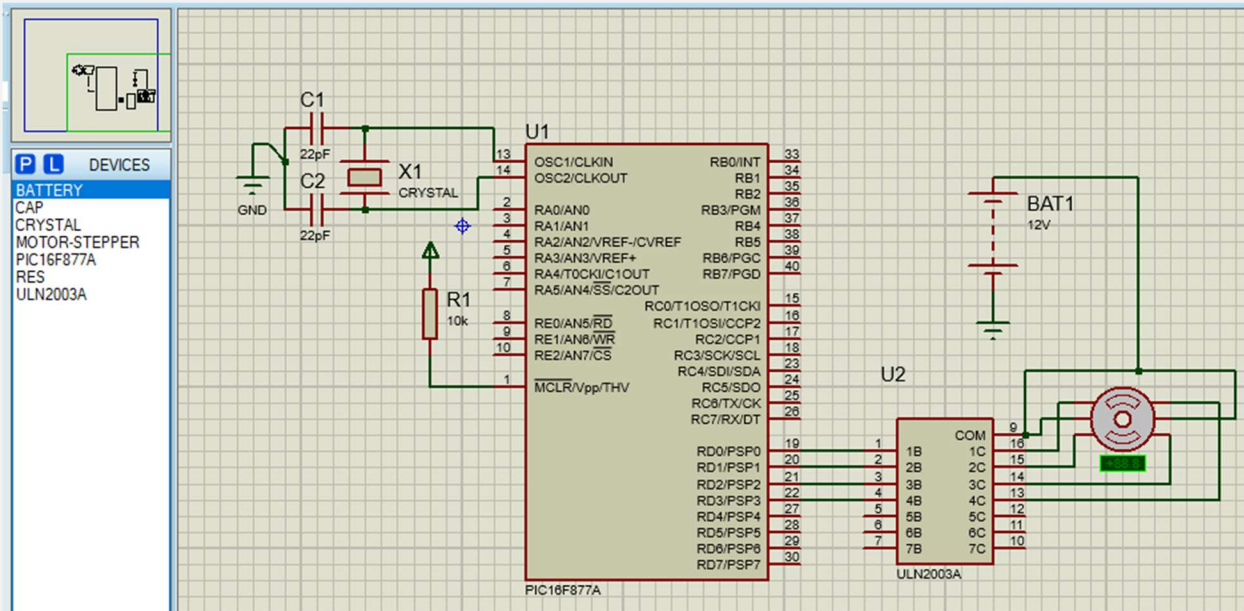
unsigned char y,z;
for(y=0;y<Time;y++)
    for(z=0;z<20;z++);
}

void main()
{
    TRISB = 0b00000000; // All PORTC pins as output
    TRISD = 0b00000000; // All PORTD pins as output

    while(1)
    {
        //portd=row ,portb=column
        PORTD = 0b10000000; PORTB = 0b00000000; MSDelay(10);
        PORTD = 0b01000000; PORTB = 0b11111111; MSDelay(10);
        PORTD = 0b00100000; PORTB = 0b11111111; MSDelay(10);
        PORTD = 0b00010000; PORTB = 0b11011000; MSDelay(10);
        PORTD = 0b00001000; PORTB = 0b11011000; MSDelay(10);
        PORTD = 0b00000100; PORTB = 0b11111111; MSDelay(10);
        PORTD = 0b00000010; PORTB = 0b11111111; MSDelay(10);
        PORTD = 0b00000001; PORTB = 0b00000000; MSDelay(10);
    }
}

```

P-5: Develop a program to rotate a Stepper Motor at different angles.



```

void main()
{
    TRISD = 0b00000000; // PORT D as output port
}

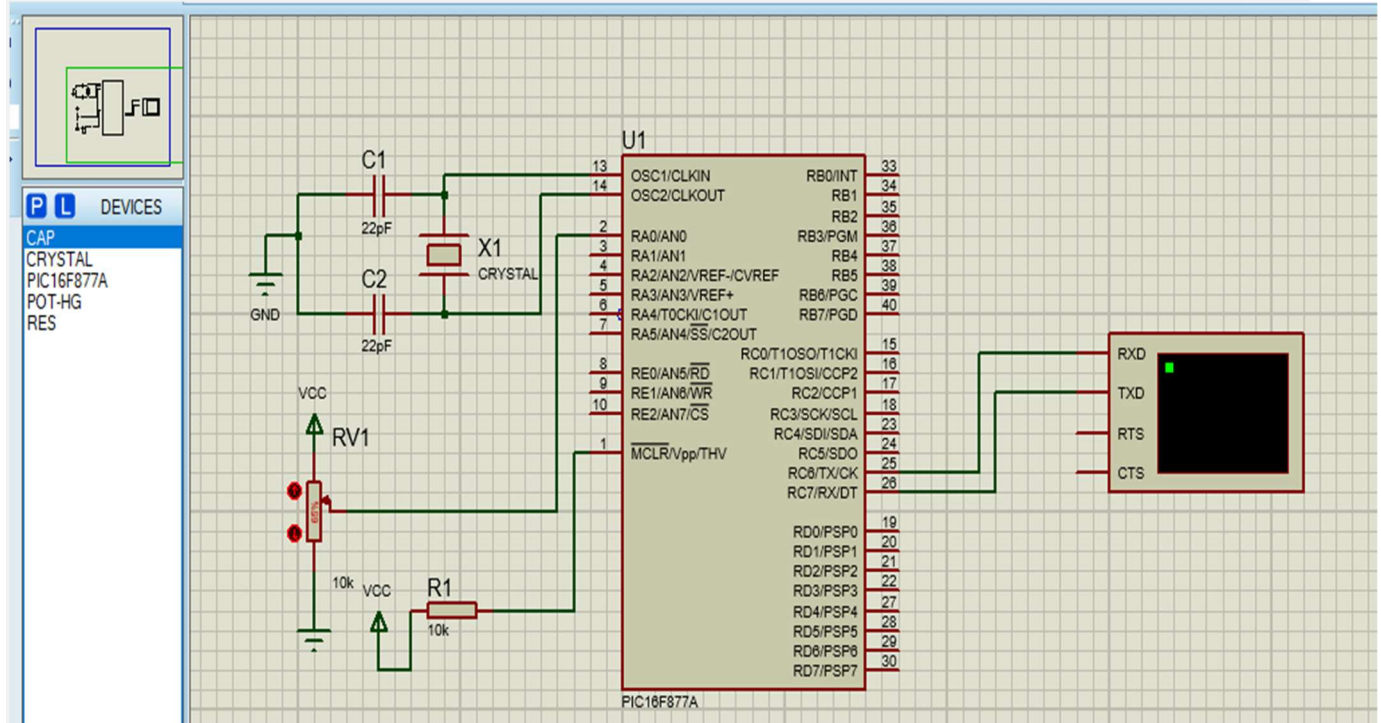
```

```

PORTD = 0b1111111;
do
{
PORTD = 0b00000011;
Delay_ms(500); // delay of 0.5s
PORTD = 0b00000110;
Delay_ms(500);
PORTD = 0b00001100;
Delay_ms(500);
PORTD = 0b00001001;
Delay_ms(500);
// energizing two phases at a time
}while(1); // loop executed infinite times
}

```

P-6: Write a program to simulate Analog to Digital Conversion using virtual terminal.



```

//create a variable that will hold the ADC value
int valADC;
//create a char array
char x[4];
void main()
{

```

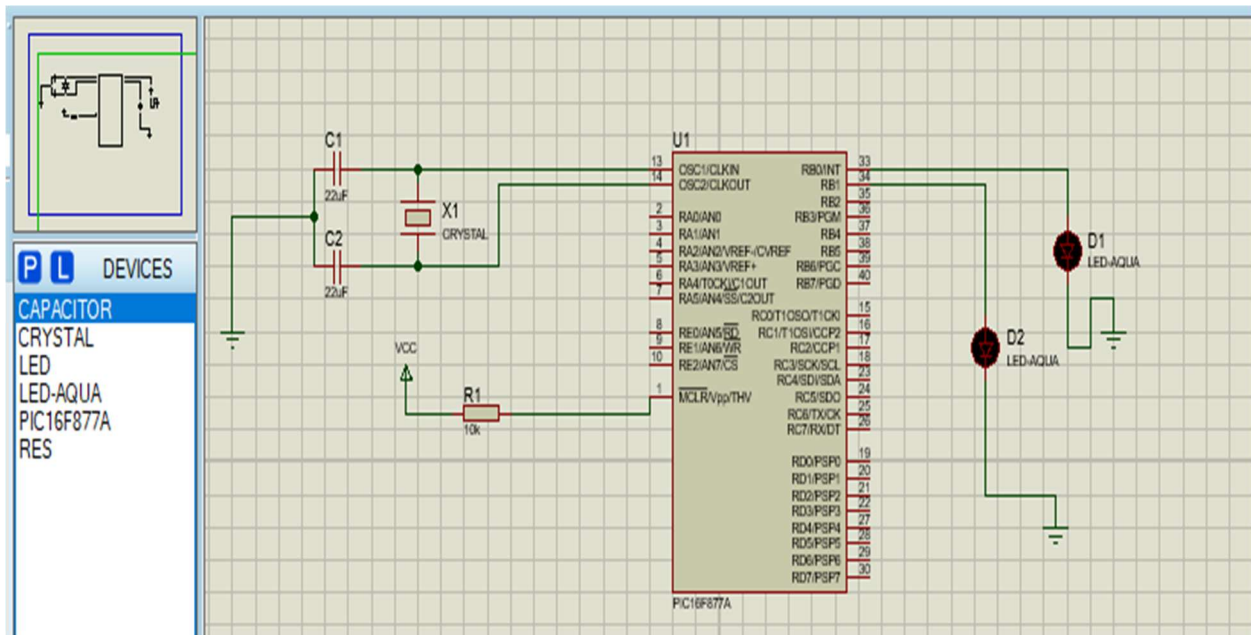


```

//initialize UART
UART1_Init(9600);
//initialize ADC
ADC_Init();
//create a loop
while(1)
{
//Read ADC value in RA0
valADC = ADC_Read(0);
//convert into string/char array
IntToStr(valADC,x);
//Print
UART1_Write_Text("Analog value = ");
UART1_Write_Text(x);
//clear char array
strcpy(x,"");
UART1_Write(13);
Delay_ms(1000);
}
}

```

P-7: Simulate a program to glow an LED for 1 to 5 seconds using loop. Also, increase the LED ON Time and decrease the LED OFF Time in every cycle with a total constant Cycle Time of 6 seconds.



```

// Custom delay function (variable ms)
void myDelay_ms(unsigned int time_ms) {
    unsigned int i;
    for(i = 0; i < time_ms; i++) {
        Delay_ms(1); // Built-in mikroC constant delay
    }
}

void main() {
    unsigned int on_time, off_time;

    TRISB = 0x00; // PORTB as output
    PORTB = 0x00; // All pins OFF initially

    while(1) {
        // Cycle: ON time increases 1 ? 5 sec, OFF time decreases 5 ? 1 sec
        for(on_time = 1; on_time <= 5; on_time++) {
            off_time = 6 - on_time; // Keep total cycle time = 6 sec

            // LED ON
            PORTB.F0 = 1; // Turn ON LED at RB0
            myDelay_ms(on_time * 1000); // Delay ON_time seconds

            // LED OFF
            PORTB.F0 = 0; // Turn OFF LED
            myDelay_ms(off_time * 1000); // Delay OFF_time seconds
        }
    }
}

```

P-10: Develop a program to control the projector screen controller. The controller has one motor, three buttons for forward motor direction, reverse motor direction and stop function.


```

    while(portD.f1==1);
}

// Button 3 (Stop) ? Reset state
if(portD.f2==1) {
    delay_ms(50);
    if(portD.f2==1) {
        state=0; // stop
    }
    while(portD.f2==1);
}

// Motor control
if(state==1) {
    portB.f0=1;
    portB.f1=0;
    portB.f2=1;
}
else if(state==2) {
    portB.f0=0;
    portB.f1=1;
    portB.f2=1;
}
else { // state == 0 (Stop)
    portB.f0=0;
    portB.f1=0;
    portB.f2=0;
}
}
}

```

P-11: Write a program to control the speed of a DC motor based on your room temperature and PWM Technique.

```
char display[16] = "";  
void main() {  
    unsigned int result;  
    unsigned short duty;  
    float volt, temp;  
    trisb = 0;
```

```

trisa = 1;
trisc.f2 = 0;
lcd_init();
lcd_cmd(_lcd_clear);
lcd_cmd(_lcd_cursor_off);

//run the motor
duty = 0;
portb.f6 = 1;
portb.f7 = 0;

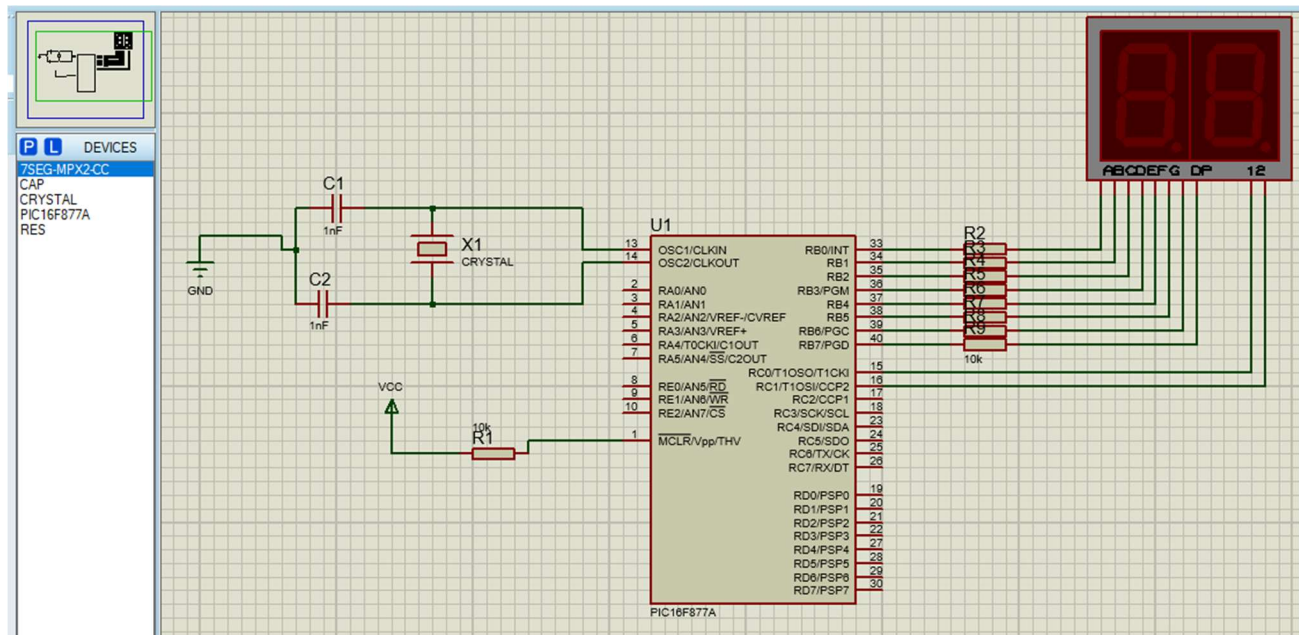
PWM1_Init(1000);
PWM1_Start();
PWM1_Set_Duty(duty);
while(1)
{
    result = adc_read(0);
    volt = result*4.88;
    temp = volt/10;

    lcd_out(1,1,"Temp = ");
    floattostr(temp,display);
    lcd_out_cp(display);
    lcd_chr(1,16,223); //223 = degree sign
    lcd_out_cp(" C");

    if(temp > 20 && duty <=240){
        Delay_ms(3000);
        if(temp > 20 && duty <=240){
            duty += 10;
            PWM1_Set_Duty(duty);
        }
    }
    if(temp < 20 && duty >= 10){
        Delay_ms(3000);
        if(temp < 20 && duty >= 10){
            duty -= 10;
            PWM1_Set_Duty(duty);
        }
    }
}
}

```


P-8: Show some serial numbers in a LCD or 7 Segment display after a specific time interval, as the numbers start with one user-assigned number and terminate with another.



```
char arr[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F}; // 0-9 patterns
```

```
void main() {
    int i, start = 25, end = 70; // user assigned range
    int right, left, j;

    TRISB = 0x00; // PORTB as output (7-seg data lines)
    TRISC = 0x00; // PORTC as output (digit select)

    PORTB = 0x00;
    PORTC = 0x00;

    for(i = start; i <= end; i++) // loop from start to end
    {
        left = i / 10; // tens digit
        right = i % 10; // units digit

        // show current number for some time
        for(j = 0; j < 20; j++) // multiplexing loop
        {
            // show left digit
            PORTB = arr[left];
            PORTC.F0 = 0;
```

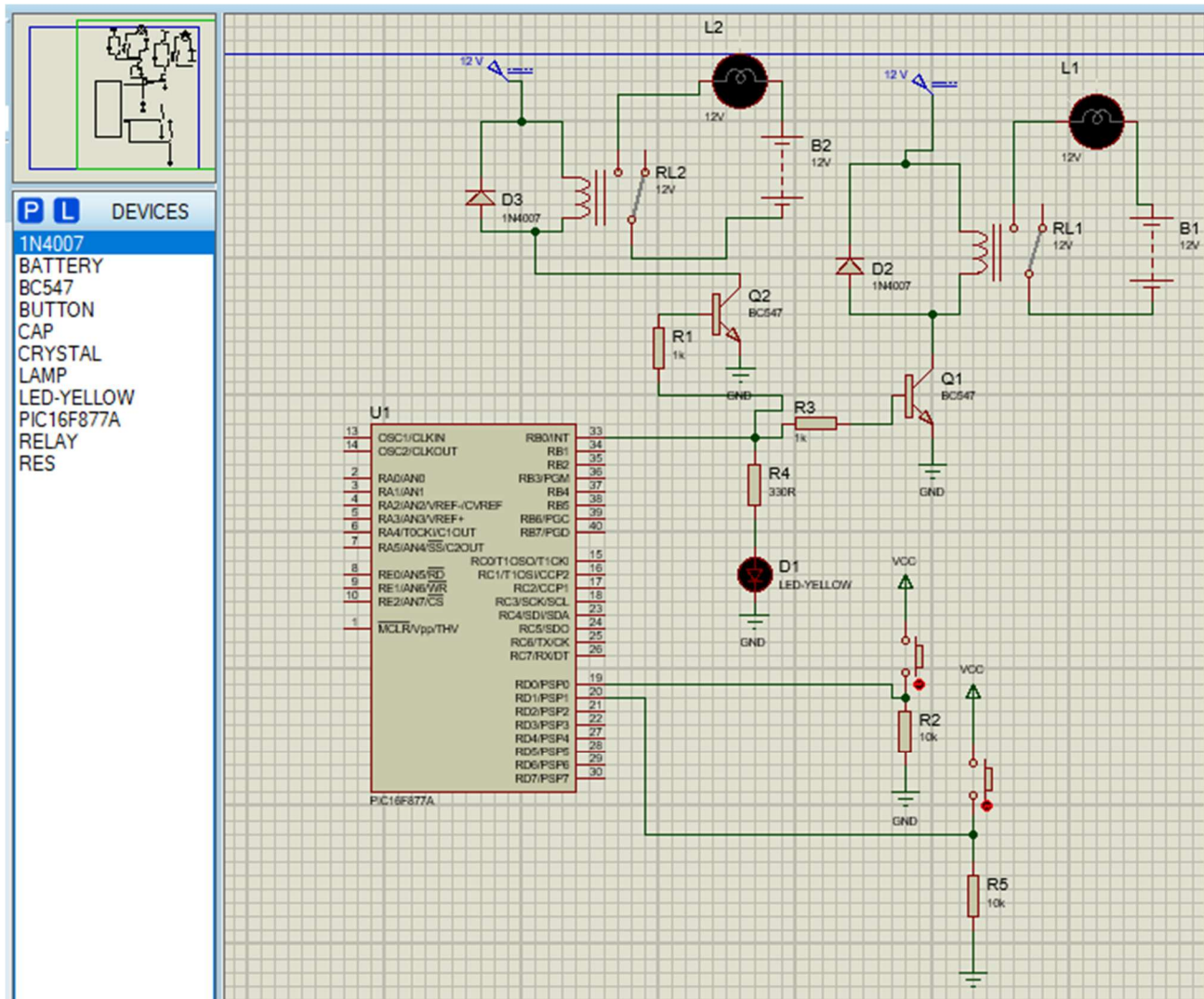
```
    Delay_ms(10);
    PORTC.F0 = 1;

    // show right digit
    PORTB = arr[right];
    PORTC.F1 = 0;
    Delay_ms(10);
    PORTC.F1 = 1;
}

    Delay_ms(500); // <-- time interval between each number (0.5s)
}

while(1); // stop after reaching 'end'
}
```

P-9: Write a program to control any high voltage loads through relay for a user specific time. The time should be inserted by the system user.



```

void main() {
    ADCON1 = 0x06;    // Make all pins digital
    TRISD = 0xFF;     // PORTD all inputs (switches)
    TRISB = 0x00;     // PORTB all outputs (relay/LED)

    PORTD = 0x00;     // Clear PORTD
    PORTB = 0x00;     // Clear PORTB

    while(1) {
        // Button on RD0 = ON
        if(PORTD.F0 == 1) {
            Delay_ms(20);    // debounce
            if(PORTD.F0 == 1) // confirm pressed
                PORTB.F0 = 1; // Relay ON
        }
    }
}

```

```
}

// Button on RD1 = OFF
if(PORTD.F1 == 1) {
    Delay_ms(20);    // debounce
    if(PORTD.F1 == 1) // confirm pressed
        PORTB.F0 = 0; // Relay OFF
}
}
}
```