

Circuit Diagram:

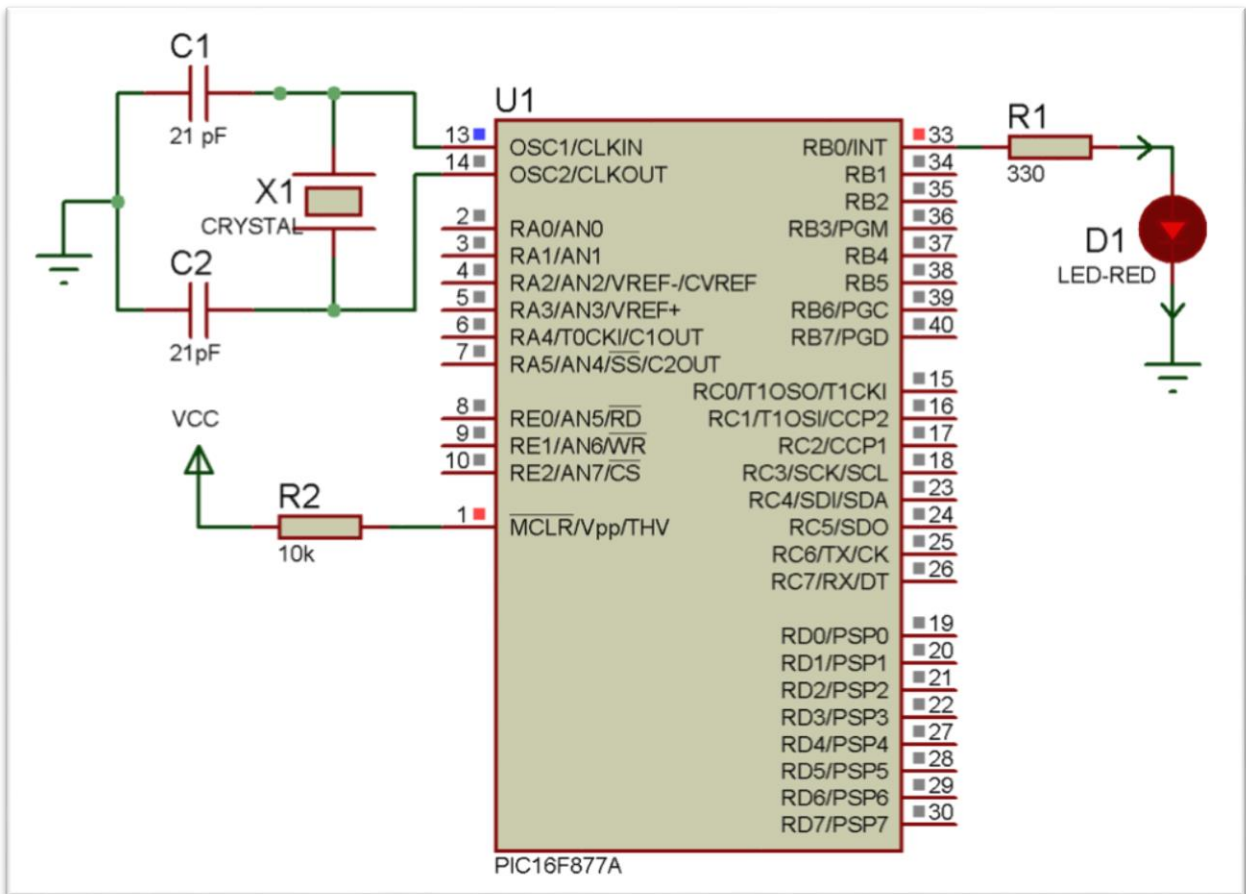


Figure: Interfacing the LED blinking using PIC16F877A

Source Code:

```
void main(){
    TRISB = 0x00;
    portb = 0x00;
    while(1){
        portb.fo = 0xff;
        delay_ms(500);
        portb.fo = 0x00;
        delay_ms(500);
    }
}
```

Circuit Diagram:

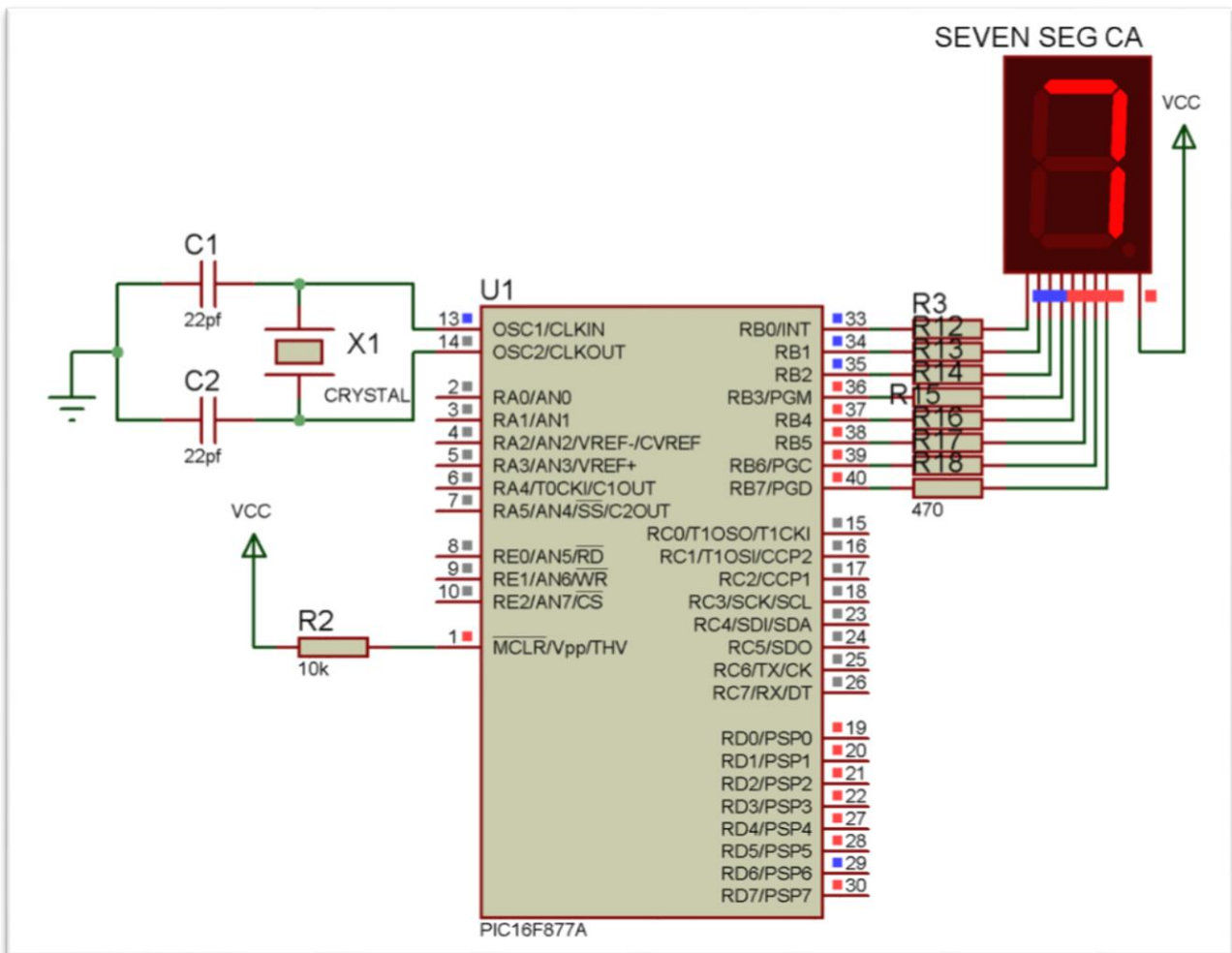


Figure: Interfacing the 7 segment display using PIC16F877A

Source Code:

```
char arraCC[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
char arraCA[] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
void main() {
    int i = 0;
    TRISD = 0x00; //set all pins of port d
    TRISB = 0x00;

    for(i = 9; i >= 0; i--){
        portb = arraCA[i];
        delay_ms(1000);
        if(i == 0) i = 10; //again restart
    }
}
```

Circuit Diagram:

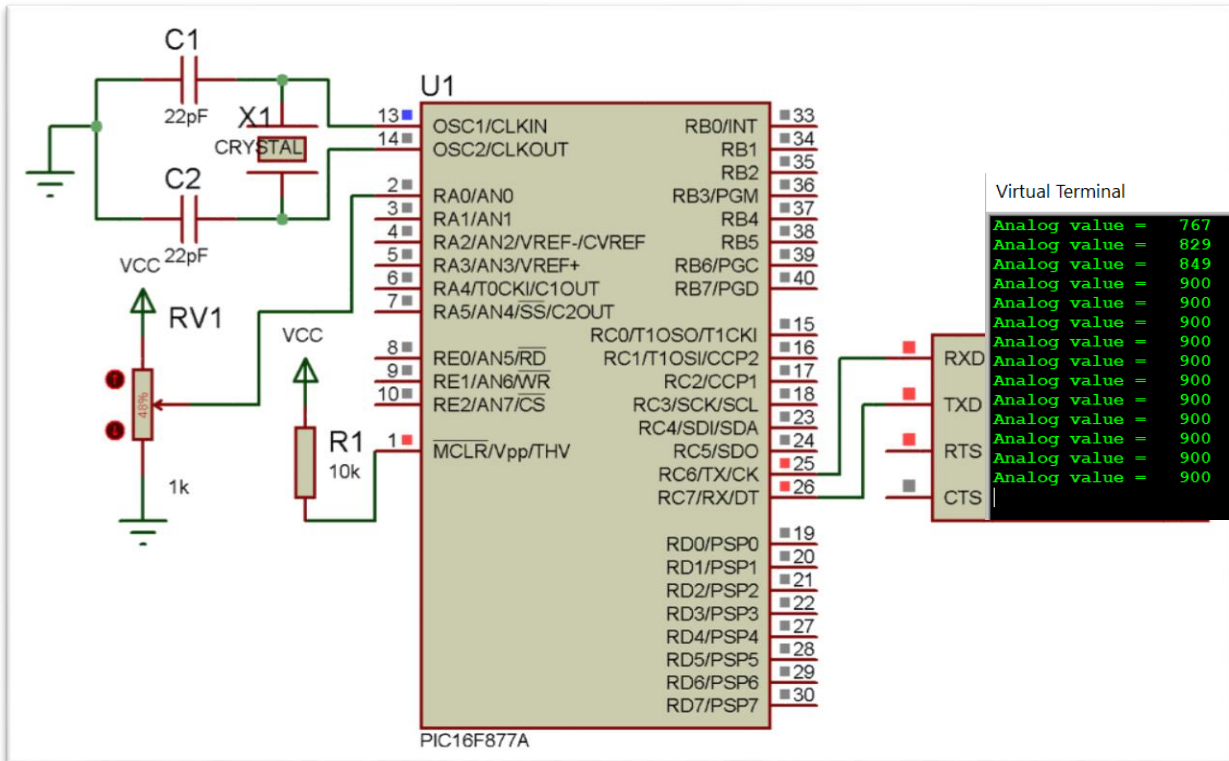


Figure: Reading the ADC value in virtual terminal using PIC16F877A

Source Code:

```
int valADC, valADC1;
char x[4];
void main() {
    UART1_Init(9600);
    ADC_Init();
    while(1) {
        valADC = ADC_Read(0);
        valADC1 = ADC_Read(1);
        IntToStr(valADC, x);
        UART1_Write_Text("Analog value =");
        UART1_Write_Text(x);
        UART1_Write_Text(" ");
        IntToStr(valADC1, x);
        UART1_Write_Text("Analog value =");
        UART1_Write_Text(x);
        UART1_Write(13);
        Delay_ms(1000);
    }
}
```

Circuit Diagram:

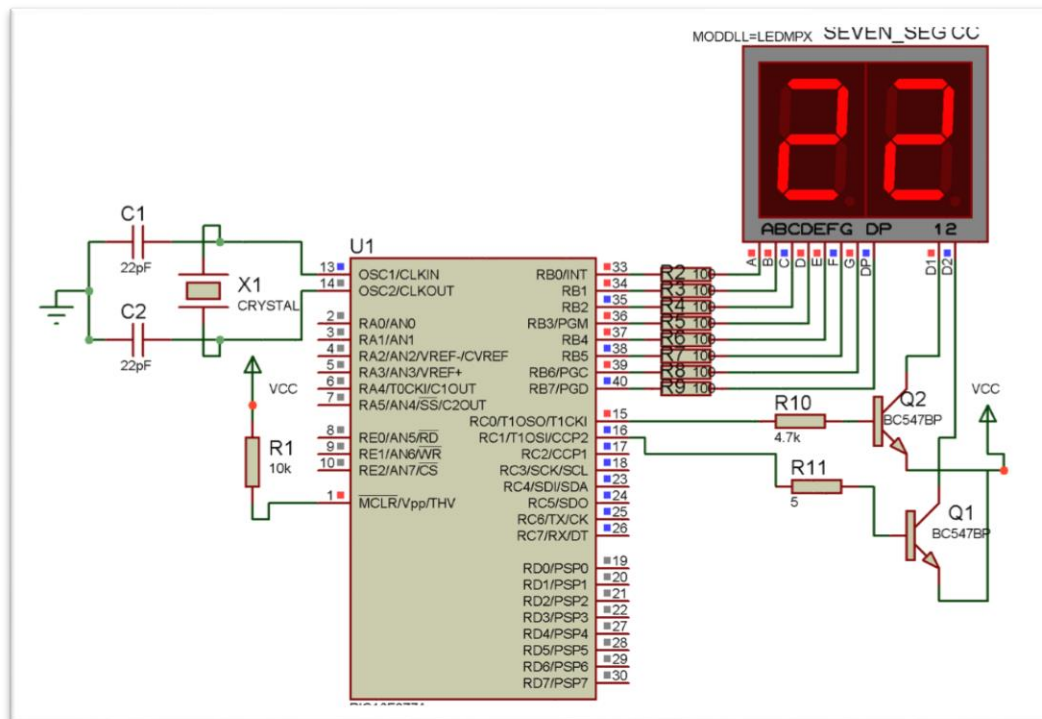


Figure: Interfacing the 2-digit 7-segment multiplexing using PIC16F877A

Source Code:

```
char arraCC[] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
void main() {
    int mod = 0, res = 0, i = 0, k = 0, state = 1;
    TRISB = 0x00;
    portb = 0x00;
    TRISC = 0x00;
    portc = 0x00;
    while(1) {
        for(i = 0; i <= 22; i++) {
            res = i/10;
            mod = i%10;
            for(k = 0; k < 50; k++) {
                portc.f0 = 0x00; // active power for digit left
                portb = arraCC[res]; // provide data for two digit
                delay_ms(10); portc.f0 = 0xFF; // Deactive power for digit right
                portc.f1 = 0x00; // active power for digit right
                portb = arraCC[mod];
                delay_ms(10);
                portc.f1 = 0xFF; // deactive power for digit left
            }
        }
    }
}
```

Circuit Diagram:

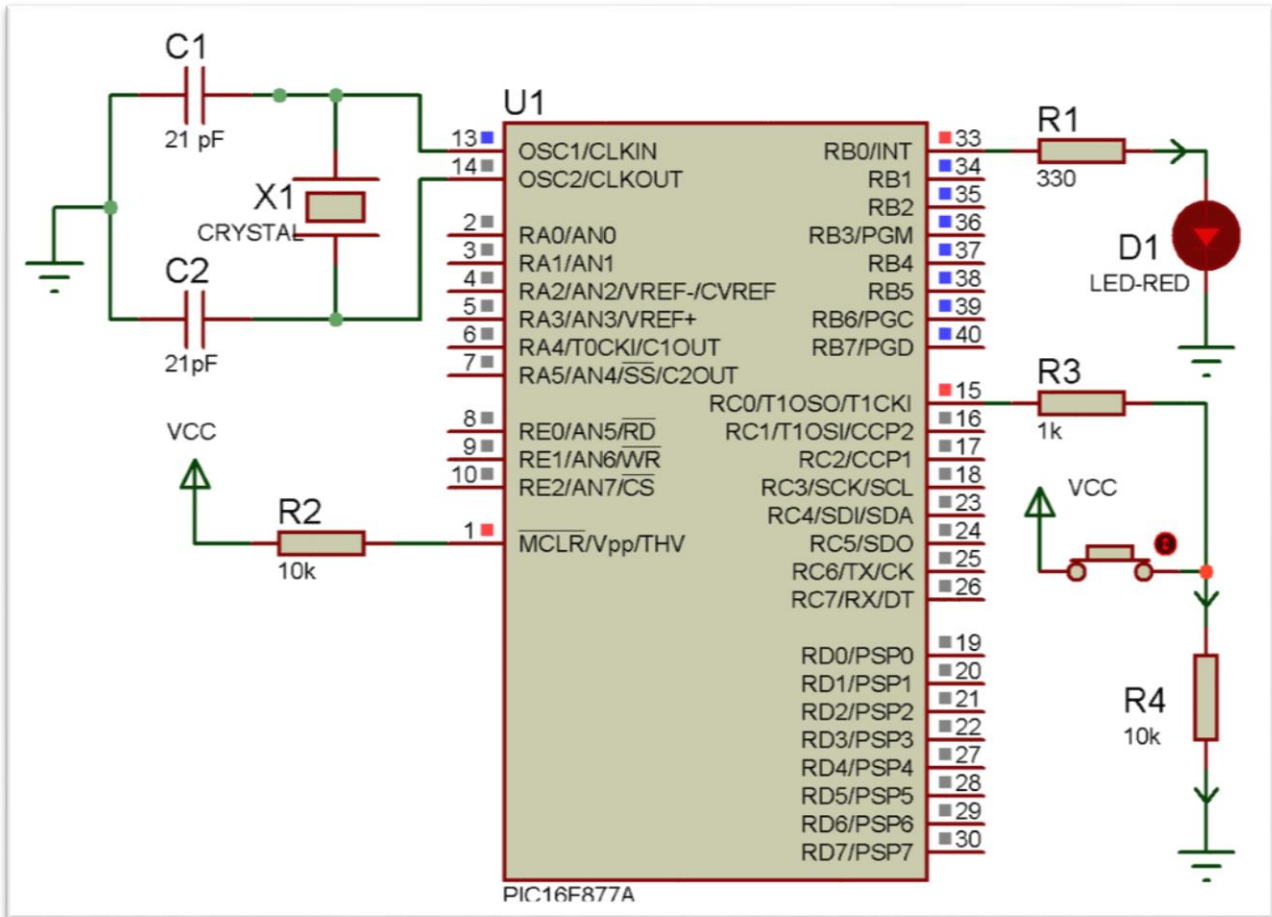


Figure: Interfacing the LED with push-button using PIC16F877A

Source Code:

```
void main(){
    TRISB = 0x00;
    TRISC = 0xFF;
    portb = 0x00;
    while(1){
        if(portc.fo == 0xFF)
            portb.fo = 0xFF;
        else
            portb.fo = 0x00;
    }
}
```

Circuit Diagram:

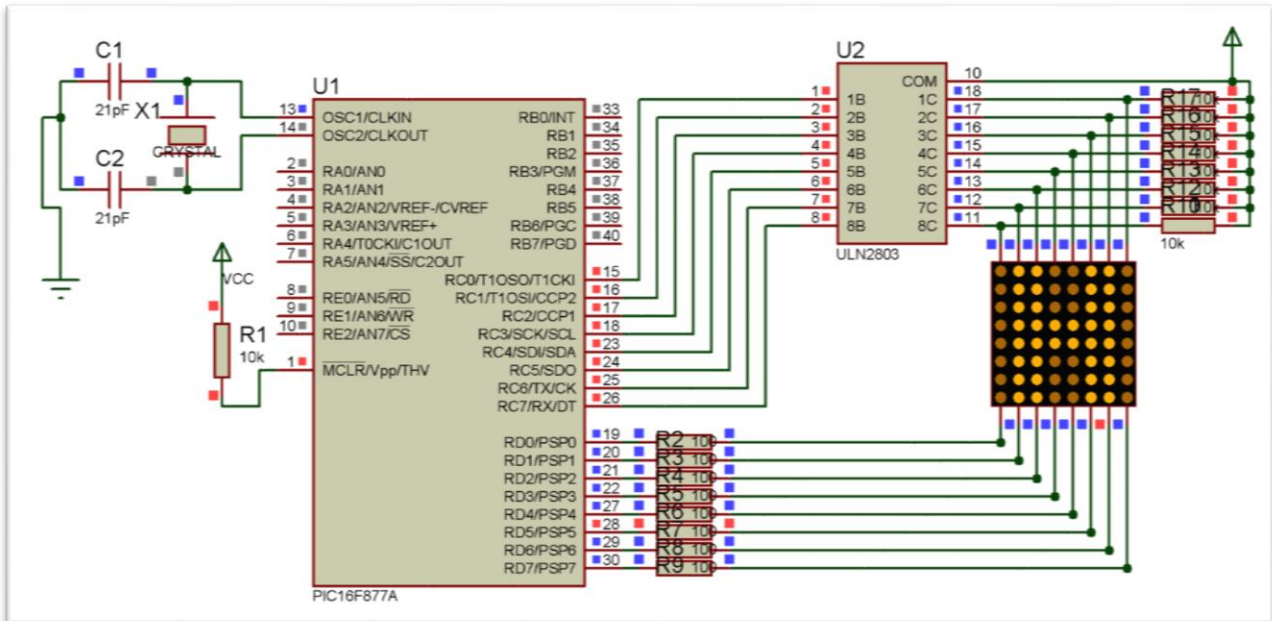


Figure: Interfacing the Dot matrix using PIC16F877A

Source Code:

```
void main() {
    TRISC = 0x00;
    TRISD = 0x00;
    while(1)
    {
        PORTD = 0x80;
        PORTC = 0x00;
        delay_ms(5);

        PORTD = 0x40;
        PORTC = 0xff;
        delay_ms(5);

        PORTD = 0x20;
        PORTC = 0xff;
        delay_ms(5);

        PORTD = 0x10;
        PORTC = 0x18;
        delay_ms(5);

        PORTD = 0x08;
        PORTC = 0x18;
        delay_ms(5);

        PORTD = 0x04;
        PORTC = 0xff;
        delay_ms(5);

        PORTD = 0x02;
        PORTC = 0xff;
        delay_ms(5);

        PORTD = 0x01;
        PORTC = 0x00;
        delay_ms(5);
    }
}
```


Circuit Diagram:

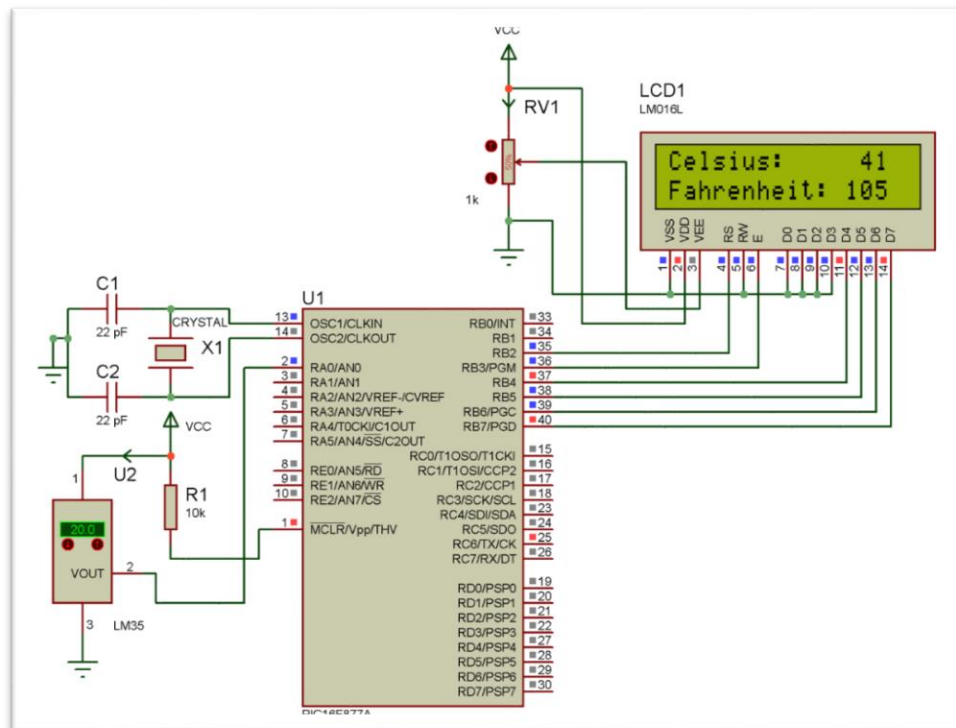


Figure: Interfacing the LM-35 temperature sensor using PIC16F877A

Source Code:

```
// LCD module connections
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;

sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;
// End LCD module connections

char cel[20], temp[40], far[20];
int i, k, valADC, farhen;
void main(){
    UART1_Init(9600); // Initialize UART
    module with baud rate 9600
    Lcd_Init(); // Initialize LCD

    Lcd_Cmd(_LCD_CLEAR); //
    Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);
    // Cursor off

    i = 0;
    while(1) {
        //Read ADC value in RA2
        valADC = ADC_Read(0);
        farhen = (valADC * 9/5) + 32;
        //Convert into string/char array
        IntToStr(valADC,cel);
        IntToStr(farhen,far);

        Lcd_Out(1,1,"Celsius:");
        Lcd_Out(1,10,cel); // Write
        text in first row
        Lcd_Out(2,1,"Fahrenheit:");
        Lcd_Out(2,10,far);
    }
}
```

Circuit Diagram:

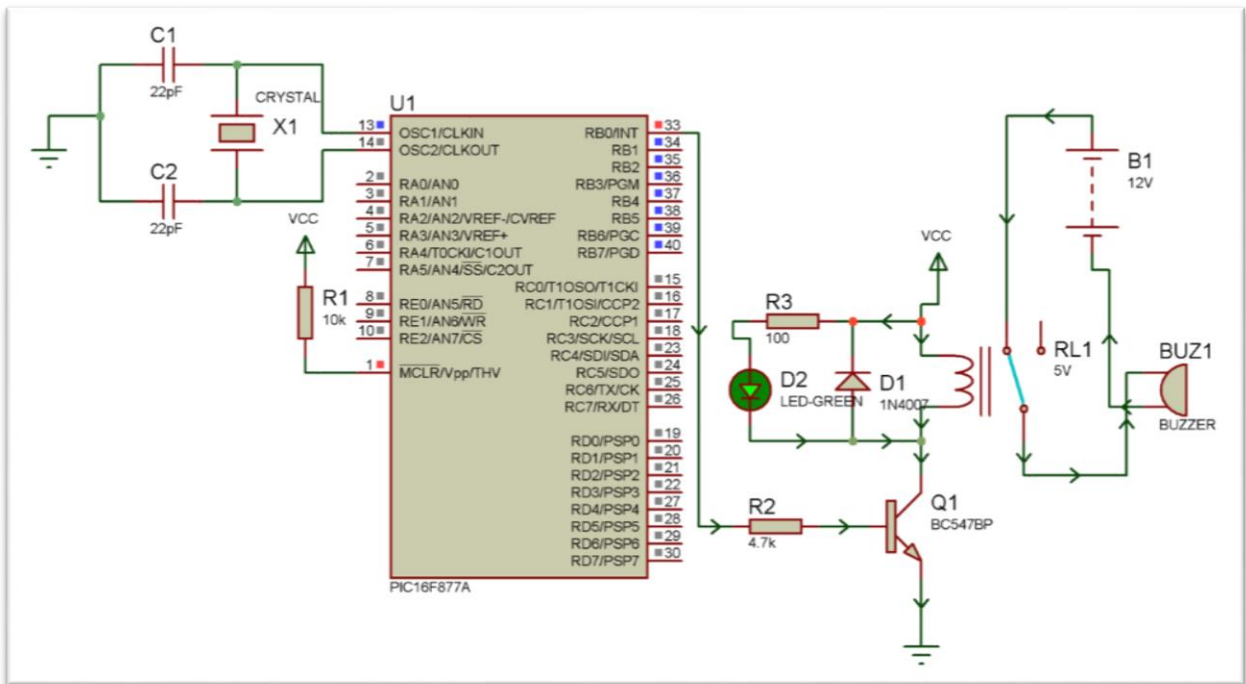


Figure: Interfacing the mechanical relay using PIC16F877A

Source Code:

```
void main() {
    TRISB = 0x00;
    portb = 0x00;
    while(1)
    {
        portb.f0 = 1;
        delay_ms(1000);
        portb.f0 = 0;
        delay_ms(1000);
    }
}
```


Circuit Diagram:

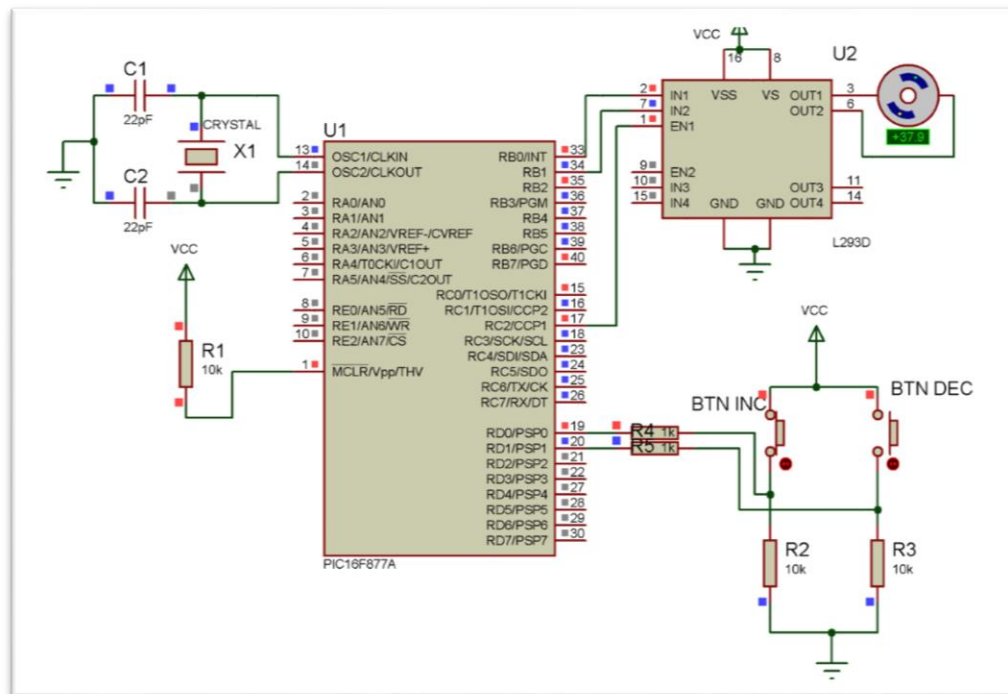


Figure: Interfacing PWM for DC motor speed controlling using PIC16F877A

Source Code:

```
void main() {
    unsigned short duty = 0;
    TRISB = 0x00;
    TRISC = 0x00;
    TRISD = 0xff;
    portb.fo = 0xff; // initialize
    portb.f1 = 0x00;
```

```
PWM1_Init(1000); // Initialize PWM
PWM1_Start(); // start PWM
PWM1_Set_Duty(duty); // set current
duty for PWM1
```

```
while(1)
{
    // Increment Button Activities
    if(portd.fo == 1)
    {
        delay_ms(200);
        if(duty <= 240)
        {
            if(portd.fo == 1)
            {
```

```

        duty = duty + 10;
        PWM1_Set_Duty(duty);
    }
}

// Decrement Button Activities
if(portd.f1 == 1)
{
    delay_ms(200);
    if(duty >= 10)
    {
        if(portd.f1 == 1)
        {
            duty = duty - 1;
            PWM1_Set_Duty(duty);
        }
    }
}
}
}

```

Circuit Diagram:

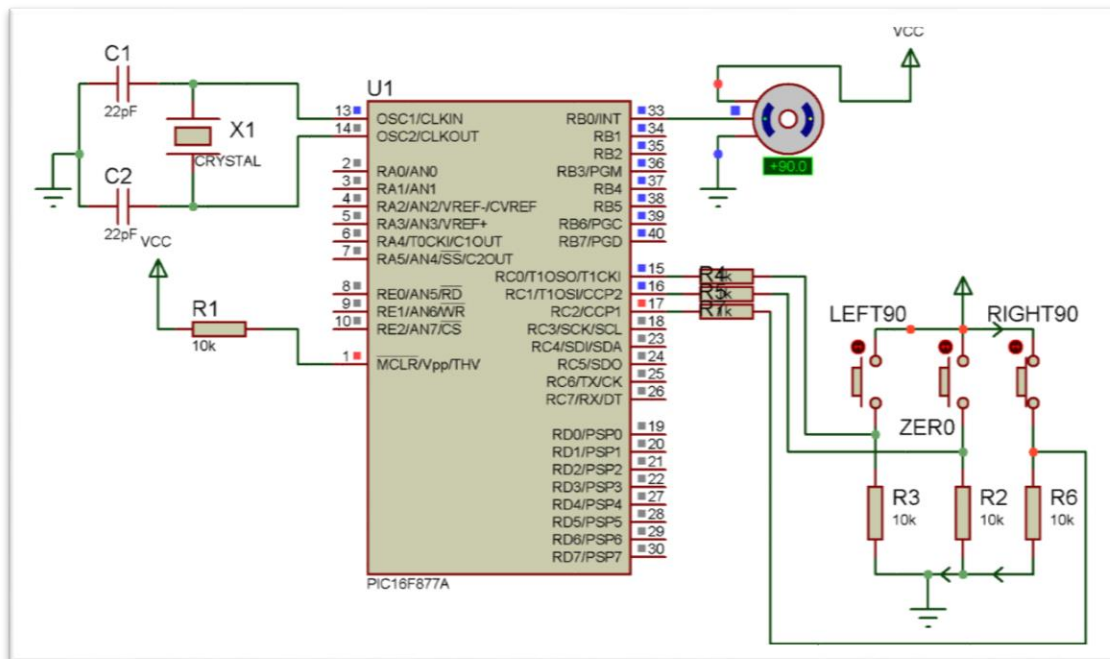


Figure: Interfacing servo using PIC16F877A

Source Code:

<pre> void rotateLeft90(); void rotateZero(); void rotateRight90(); int i; void main() { TRISB = 0X00; TRISC = 0XFF; portb = 0x00; rotateZero(); while(1) { if(portc.fo == 0xff) { rotateLeft90(); } if(portc.f1 == 0xff) { rotateZero(); } if(portc.f2 == 0xff) { rotateRight90(); } } } void rotateLeft90() { </pre>	<pre> for(i=0;i<50;i++) { portb.fo=1; delay_us(800); portb.fo=0; delay_us(19200); } } void rotateZero() { for(i=0;i<50;i++) { portb.fo=1; delay_us(1500); portb.fo=0; delay_us(18500); } } void rotateRight90() { for(i=0;i<50;i++) { portb.fo=1; delay_us(2200); portb.fo=0; delay_us(17800); } } </pre>
--	--

Circuit Diagram:

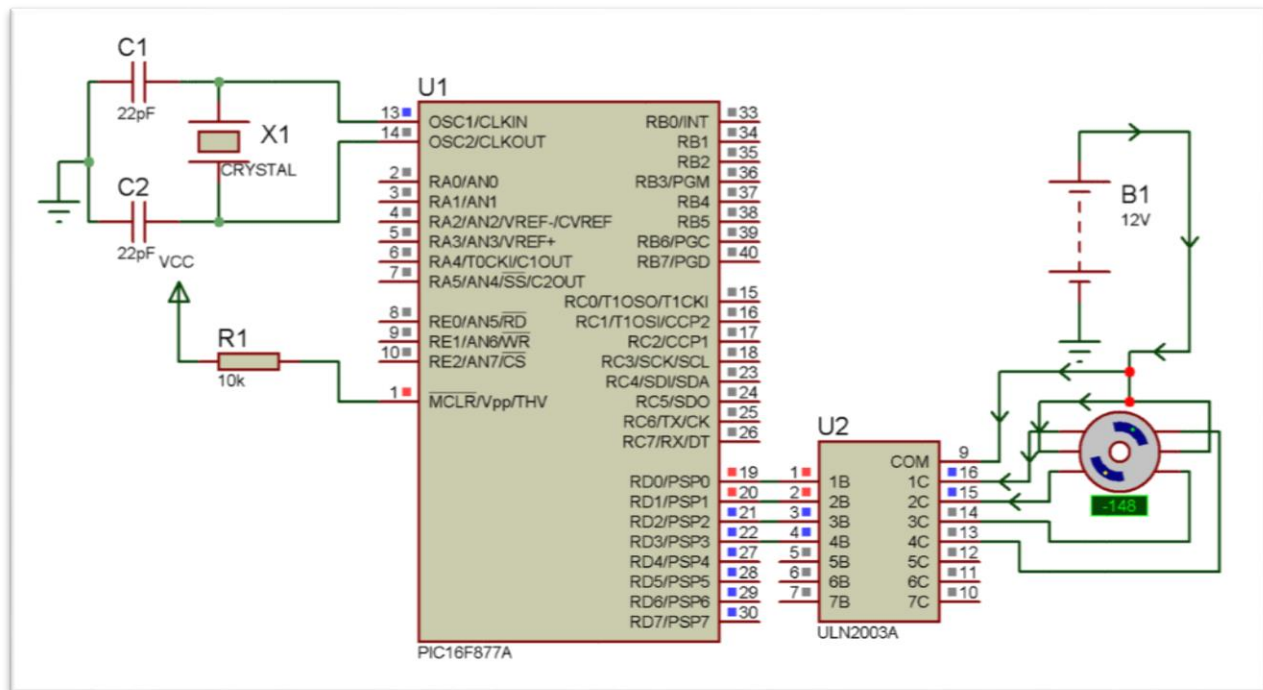


Figure: Interfacing the stepper motor using PIC16F877A

Source Code:

```
void main() {
    TRISD = 0x00;
    portd = 0x00;
    while(1)
    {
        portd = 0b00000011;
        delay_ms(50);
        portd = 0b00000110;
        delay_ms(50);
        portd = 0b00001100;
        delay_ms(50);
        portd = 0b00001001;
        delay_ms(50);
    }
}
```