Circuit Diagram:

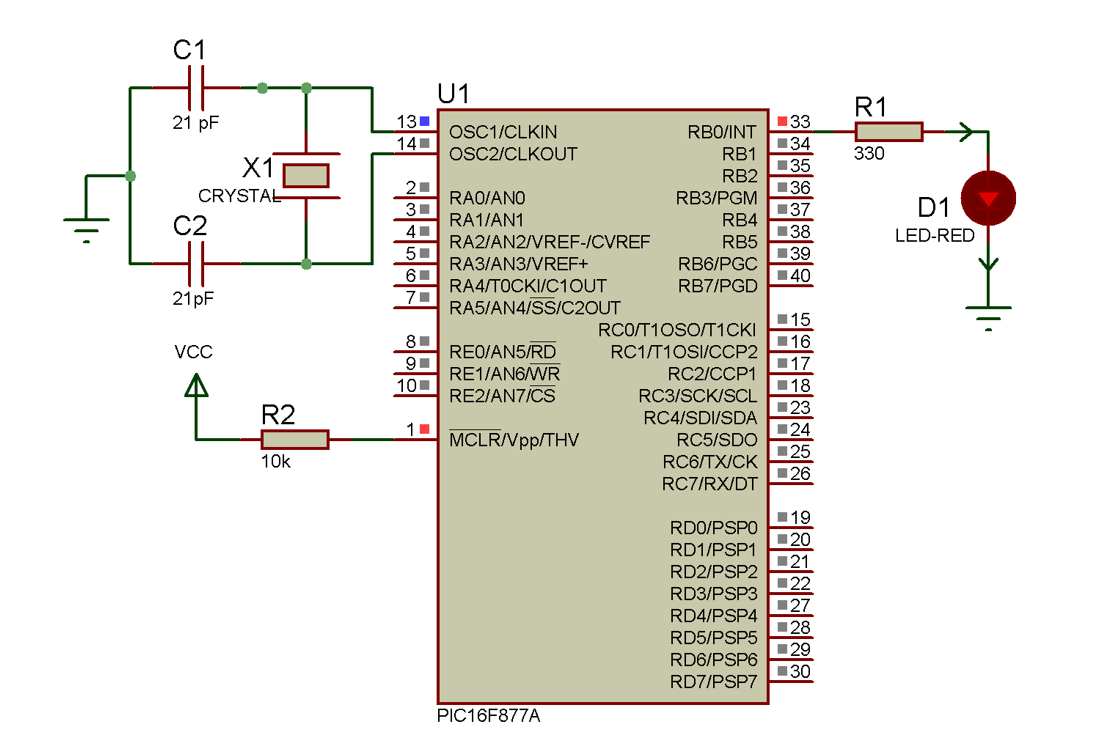


Figure: Interfacing the LED blinking using PIC16F877A

Source Code:

|  |
| --- |
| void main(){  TRISB = 0x00;  portb = 0x00;  while(1){  portb.f0 = 0xff;  delay\_ms(500);  portb.f0 = 0x00;  delay\_ms(500);  }  } |

Circuit Diagram:

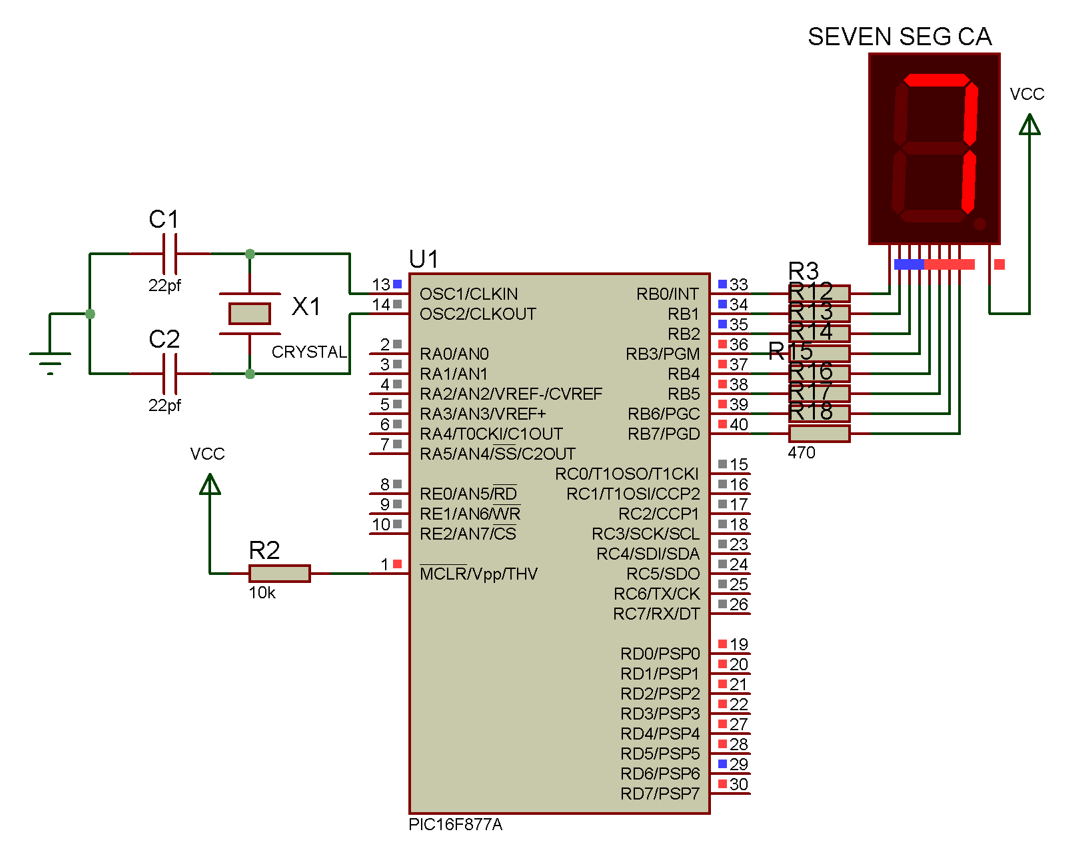


Figure: Interfacing the 7 segment display using PIC16F877A

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 pincs 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:

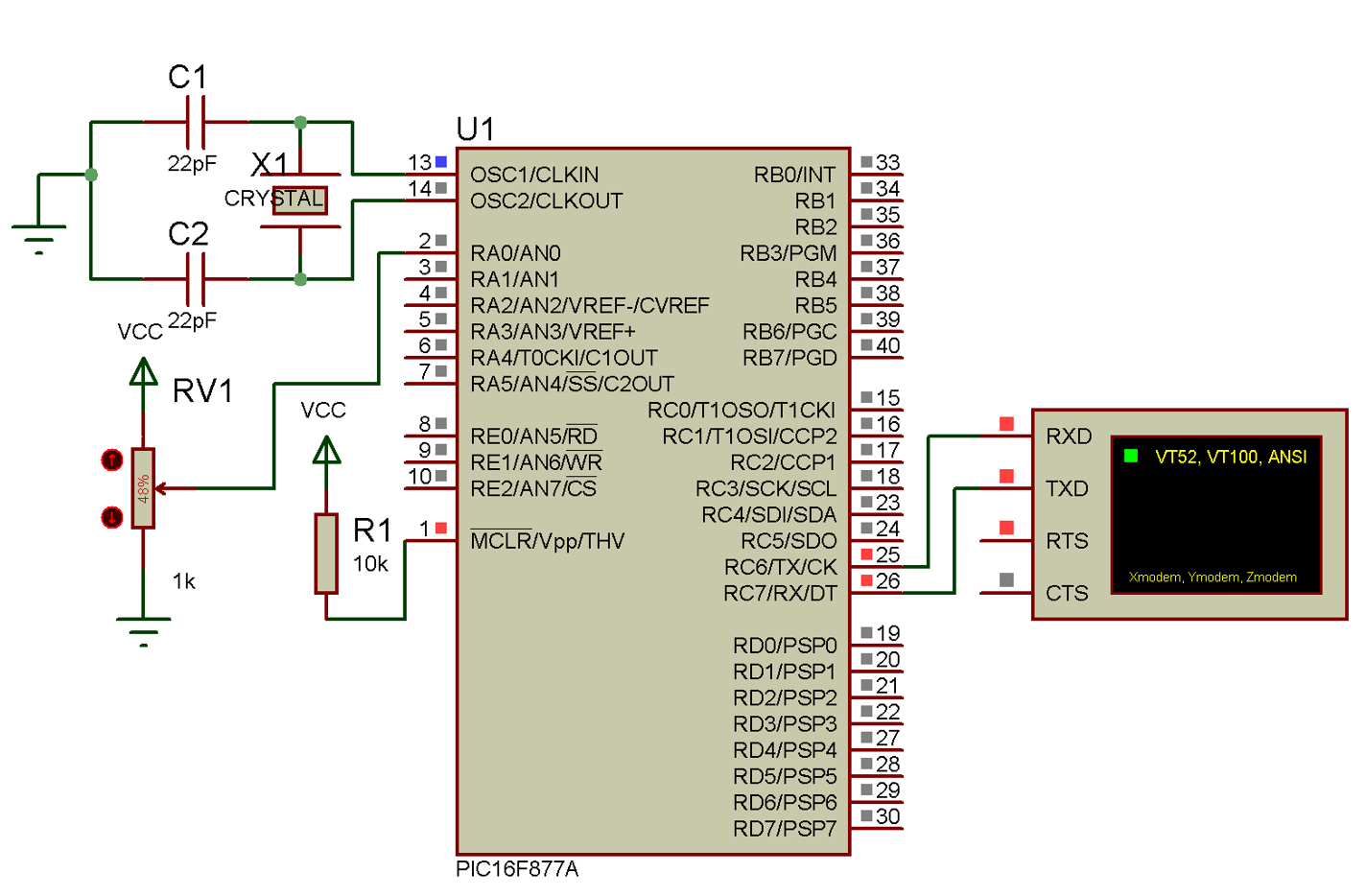
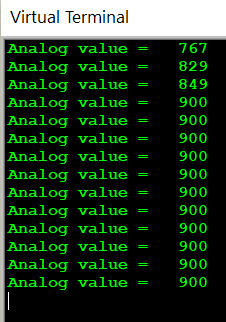


Figure: Reading the ADC value in virtual terminal using PIC16F877A

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:

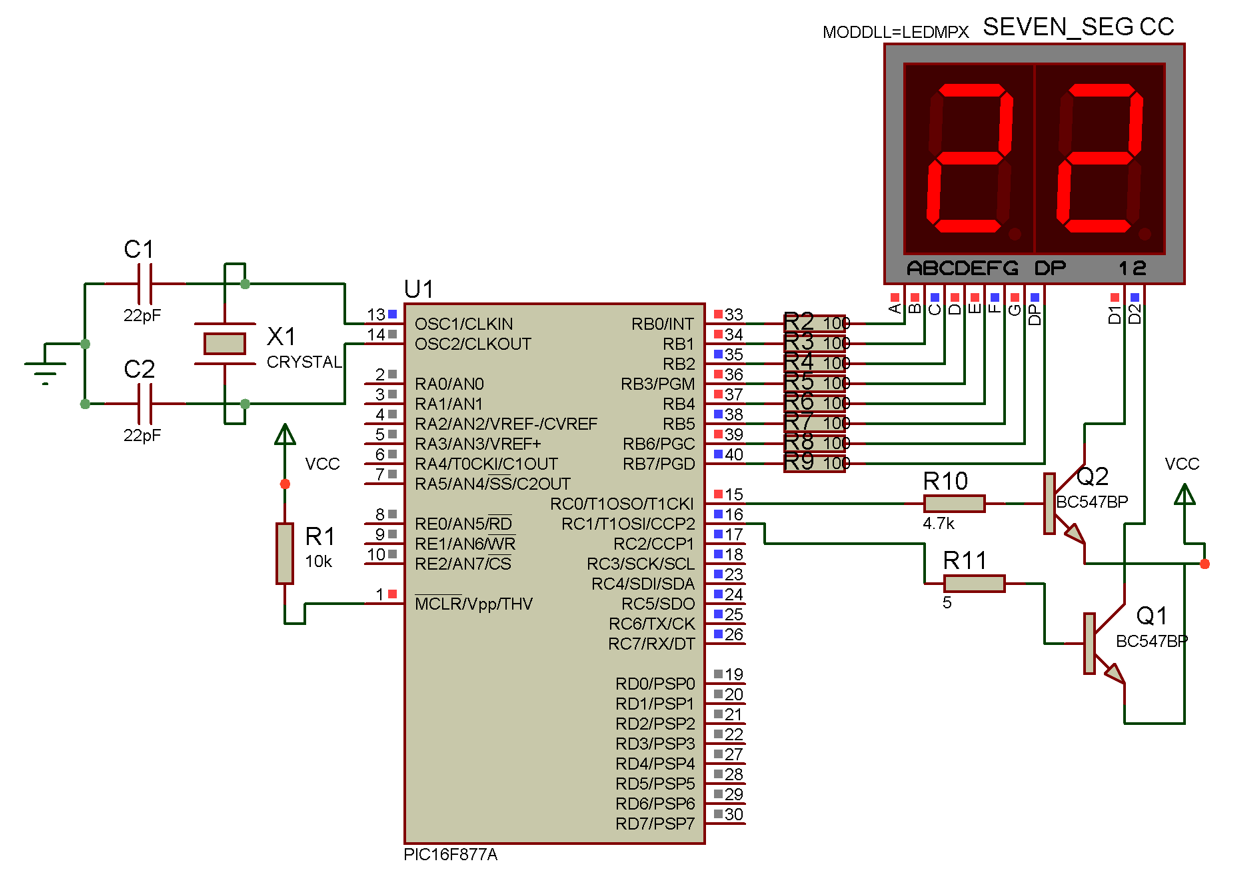


Figure: Interfacing the 2-digit 7-segment multiplexing using PIC16F877A

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:

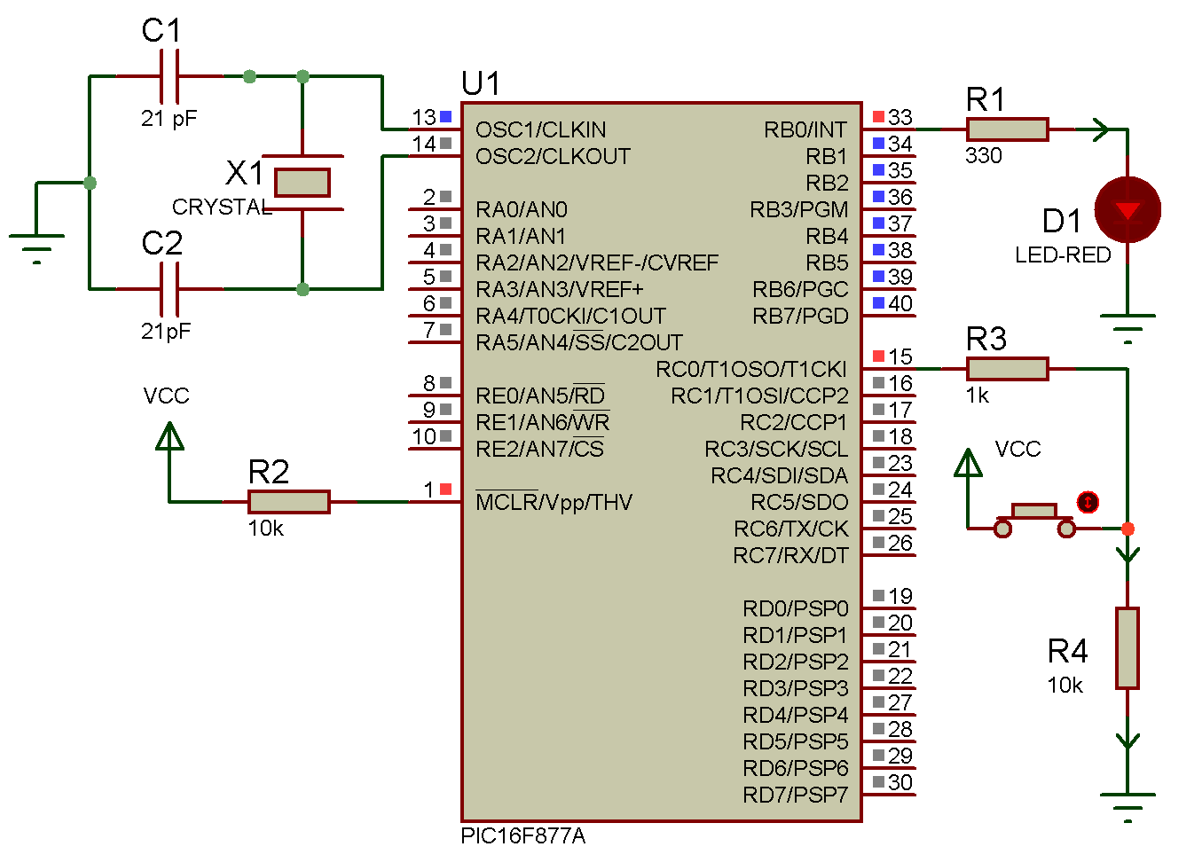


Figure: Interfacing the LED with push-button using PIC16F877A

Source Code:

|  |
| --- |
| void main(){  TRISB = 0x00;  TRISC = 0Xff;  portb = 0x00;  while(1){  if(portc.f0 == 0xff)  portb.f0 = 0xff;  else  portb.f0 = 0x00;  }  } |

Circuit Diagram:

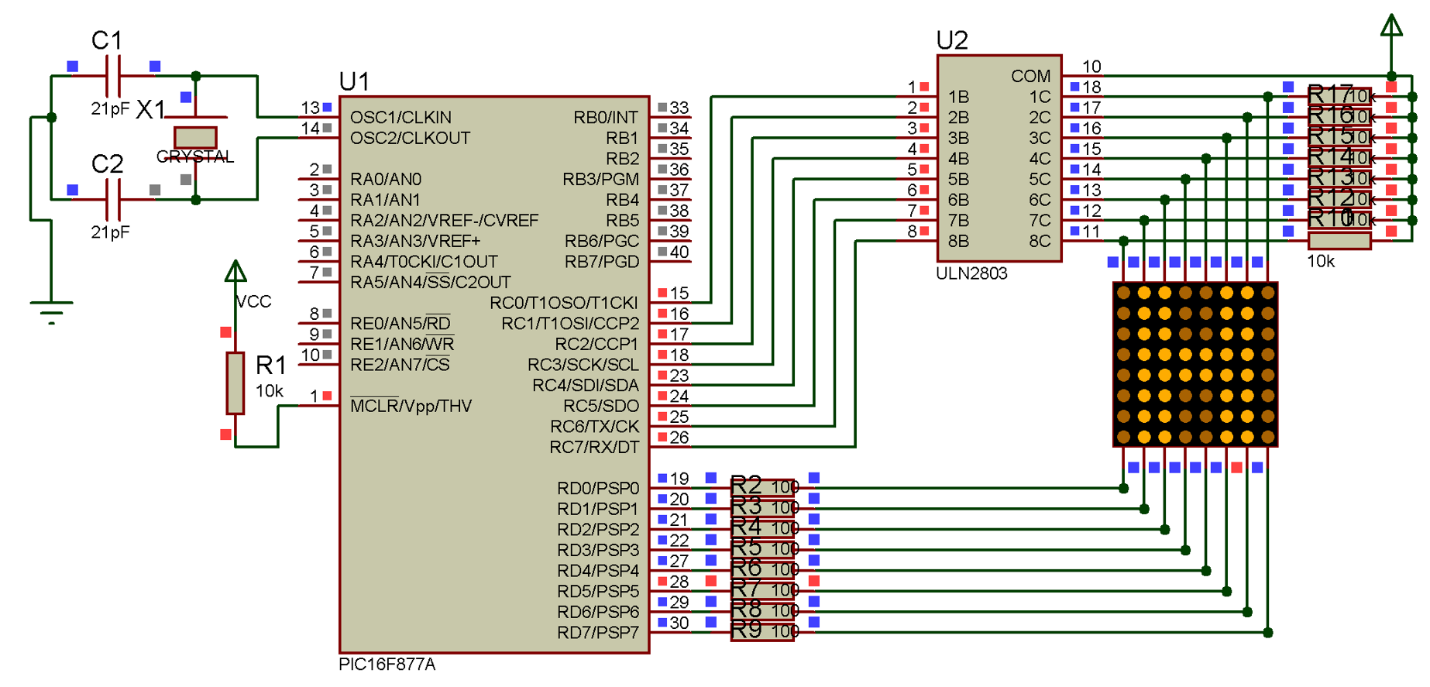


Figure: Interfacing the Dot matrix using PIC16F877A

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:

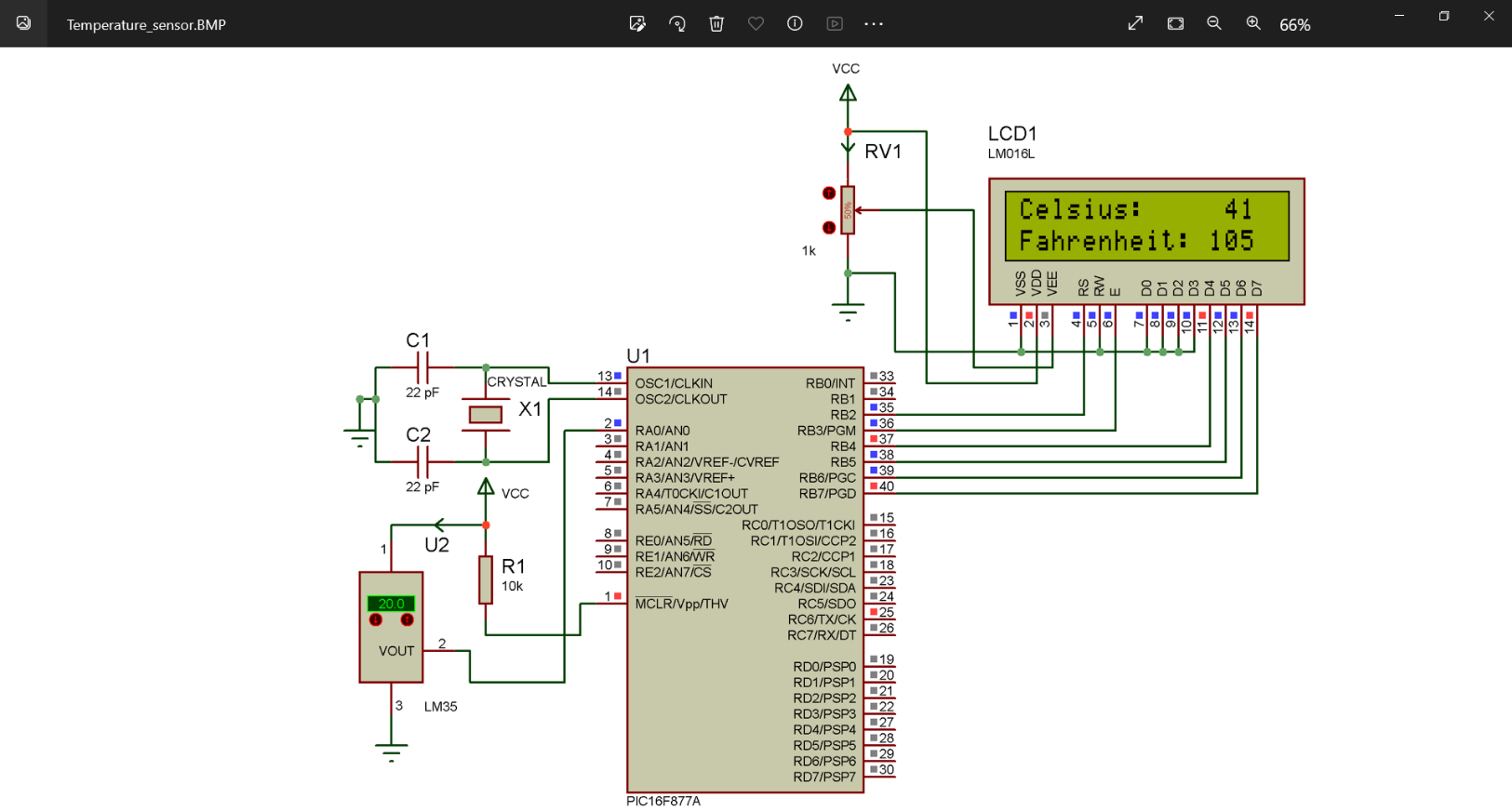


Figure: Interfacing the LM-35 temperature sensor using PIC16F877A

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:

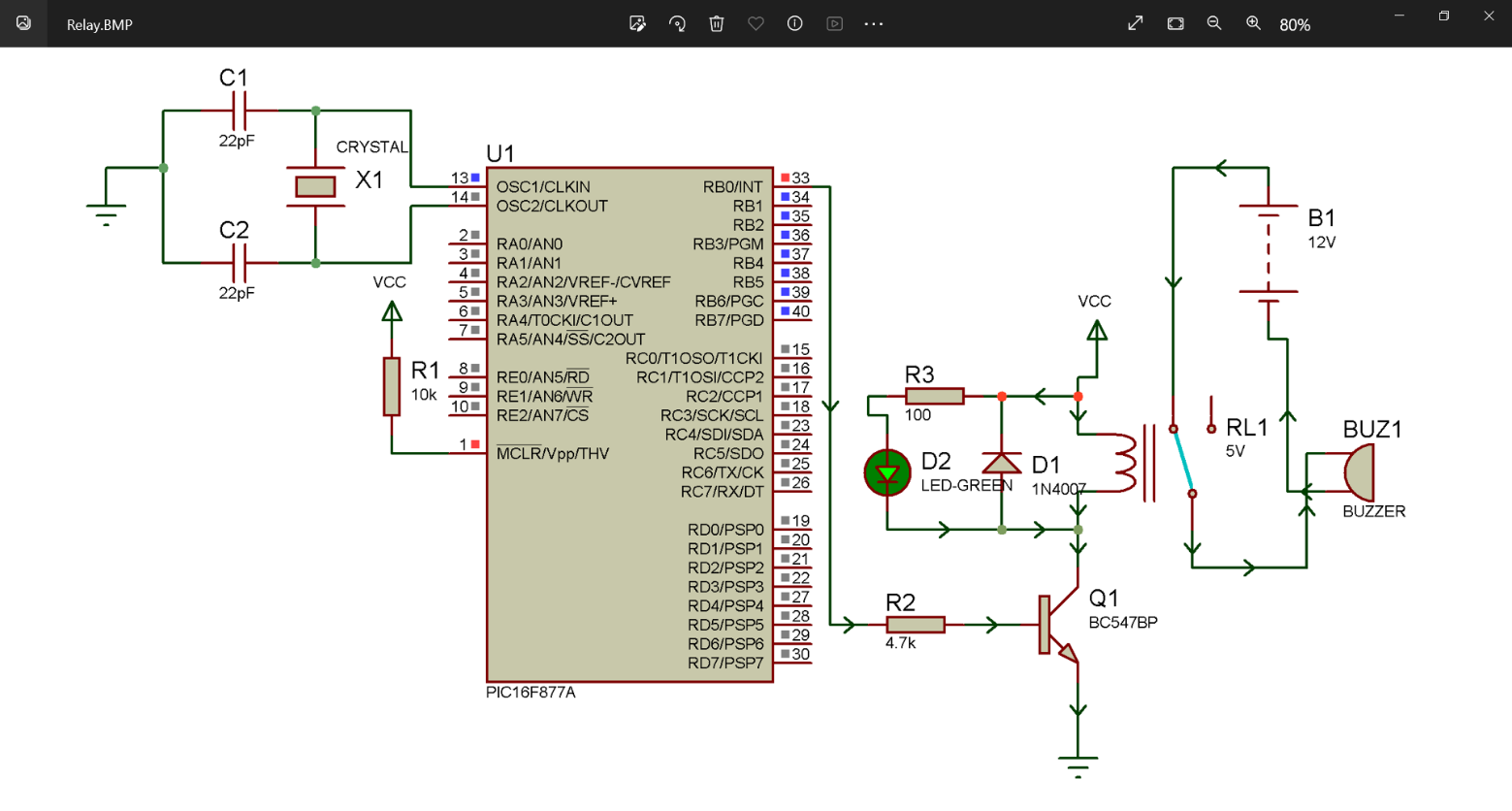


Figure: Interfacing the mechanical relay using PIC16F877A

Source Code:

|  |
| --- |
| void main() {  TRISB = 0x00;  portb = 0x00;  while(1)  {  portb.f0 = 1;  delay\_ms(1000);  portb.f0 = 0;  delay\_ms(1000);  }  } |

Circuit Diagram:

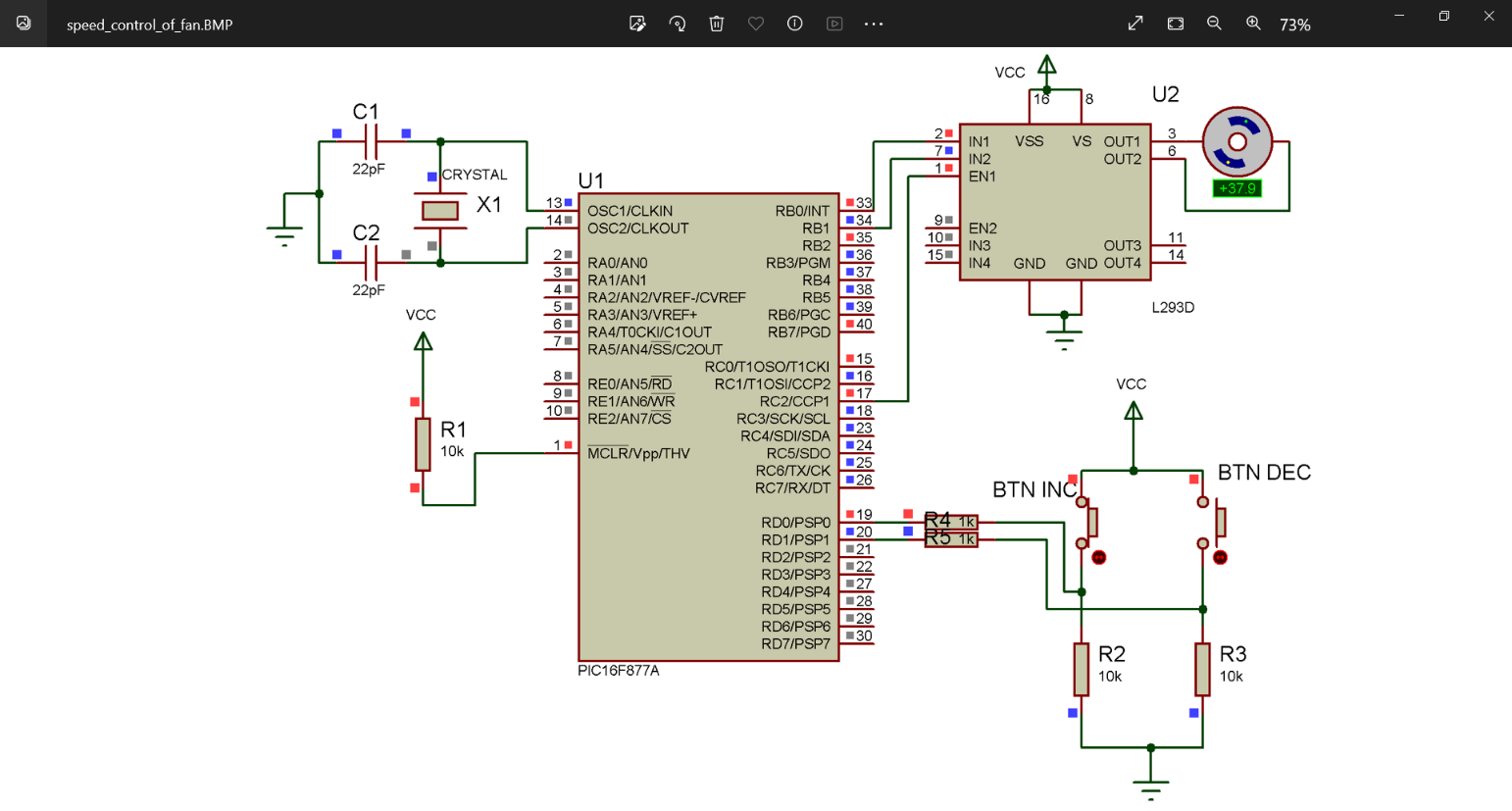


Figure: Interfacing PWM for DC motor speed controlling using PIC16F877A

Source Code:

|  |  |
| --- | --- |
| void main() {  unsigned short duty = 0;  TRISB = 0x00;  TRISC = 0x00;  TRISD = 0xff;  portb.f0 = 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.f0 ==1)  {  delay\_ms(200);  if(duty <= 240)  {  if(portd.f0 == 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:

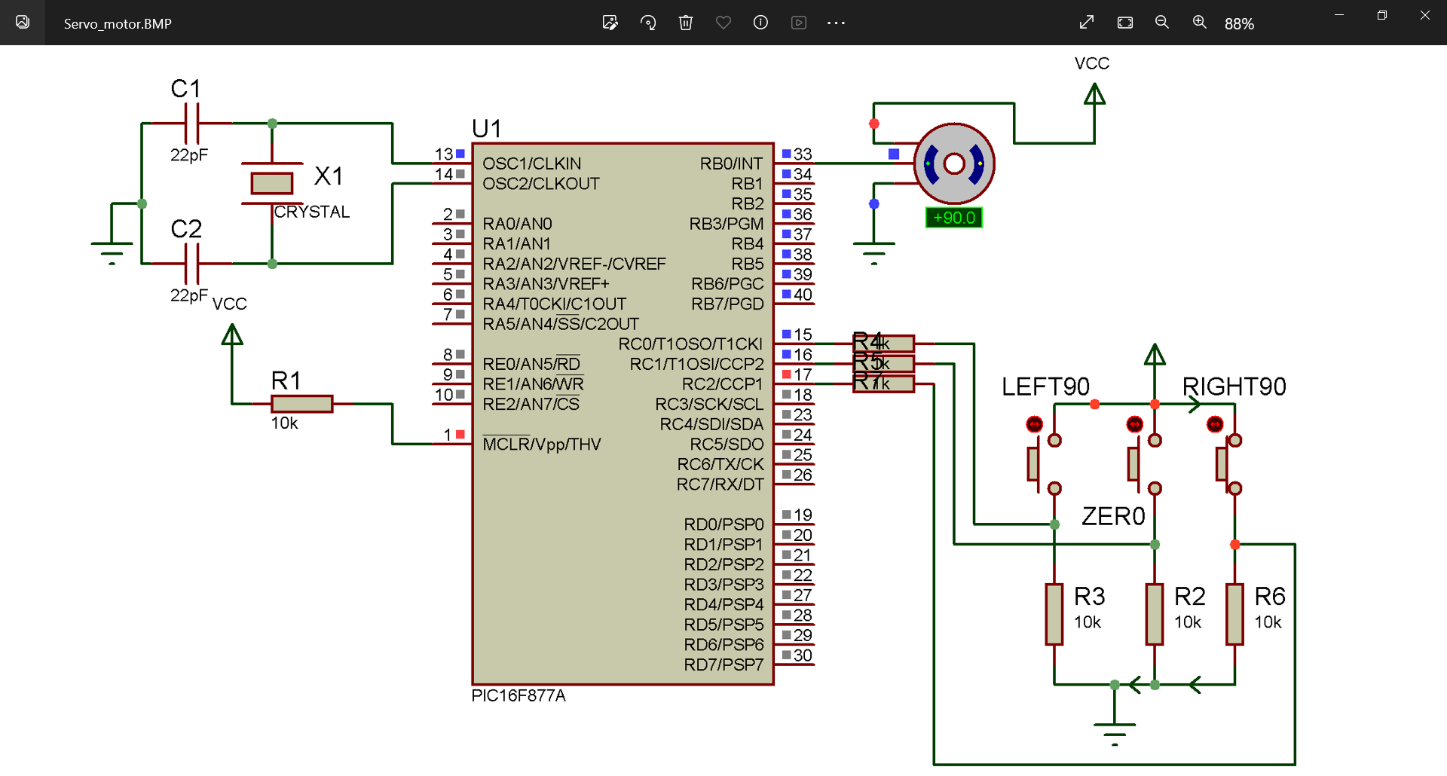


Figure: Interfacing servo using PIC16F877A

Source Code:

|  |  |
| --- | --- |
| void rotateLeft90();  void rotateZero();  void rotateRight90();  int i;  void main() {  TRISB = 0X00;  TRISC = 0XFF;  portb = 0x00;  rotateZero();  while(1) {  if(portc.f0 == 0xff) {  rotateLeft90();  }  if(portc.f1 == 0xff) {  rotateZero();  }  if(portc.f2 == 0xff) {  rotateRight90();  }  }  }  void rotateLeft90() { | for(i=0;i<50;i++) {  portb.f0=1;  delay\_us(800);  portb.f0=0;  delay\_us(19200);  }  }  void rotateZero() {  for(i=0;i<50;i++) {  portb.f0=1;  delay\_us(1500);  portb.f0=0;  delay\_us(18500);  }  }  void rotateRight90() {  for(i=0;i<50;i++) {  portb.f0=1;  delay\_us(2200);  portb.f0=0;  delay\_us(17800);  }  } |

Circuit Diagram:

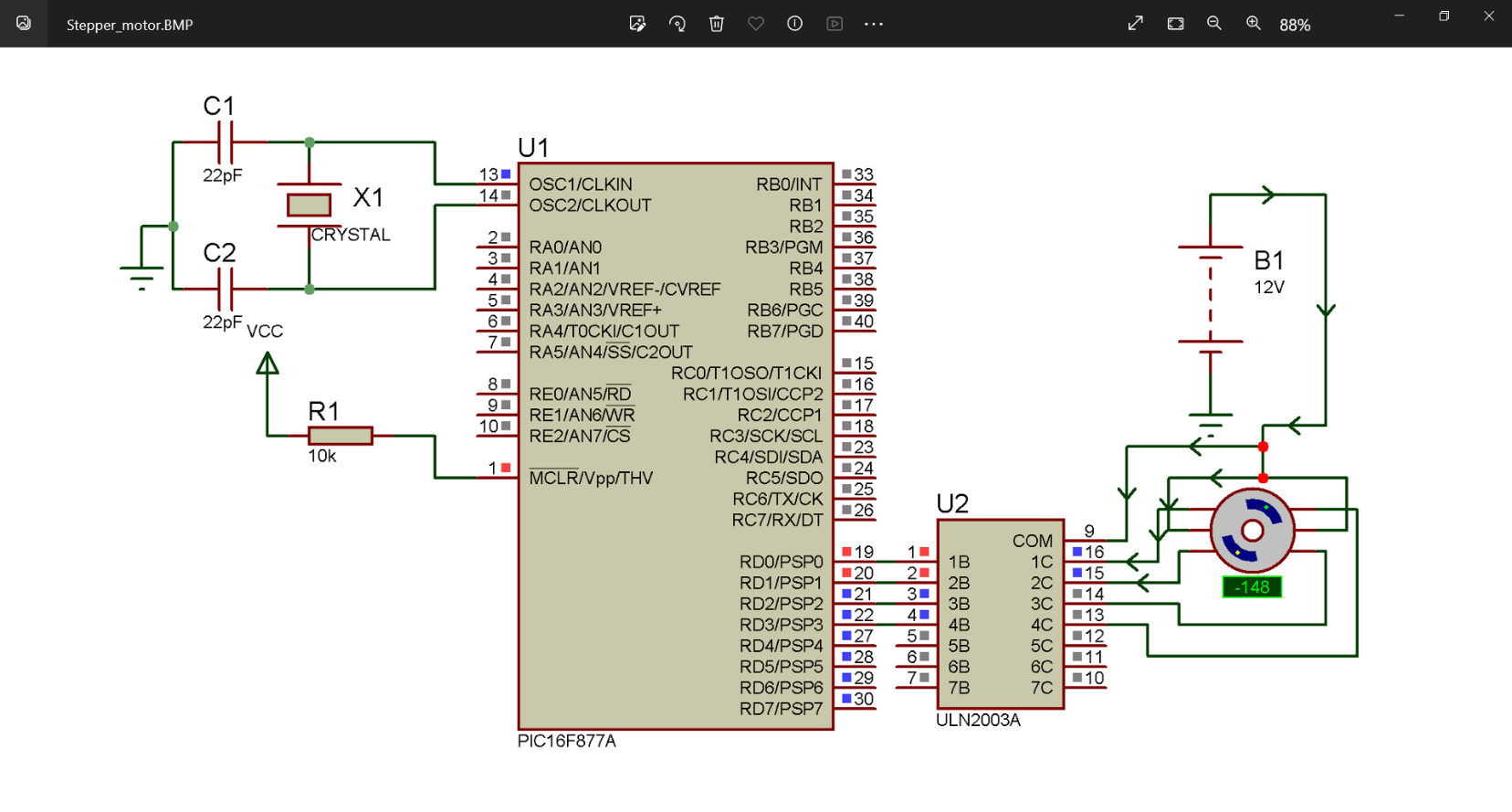


Figure: Interfacing the stepper motor using PIC16F877A

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);  }  } |