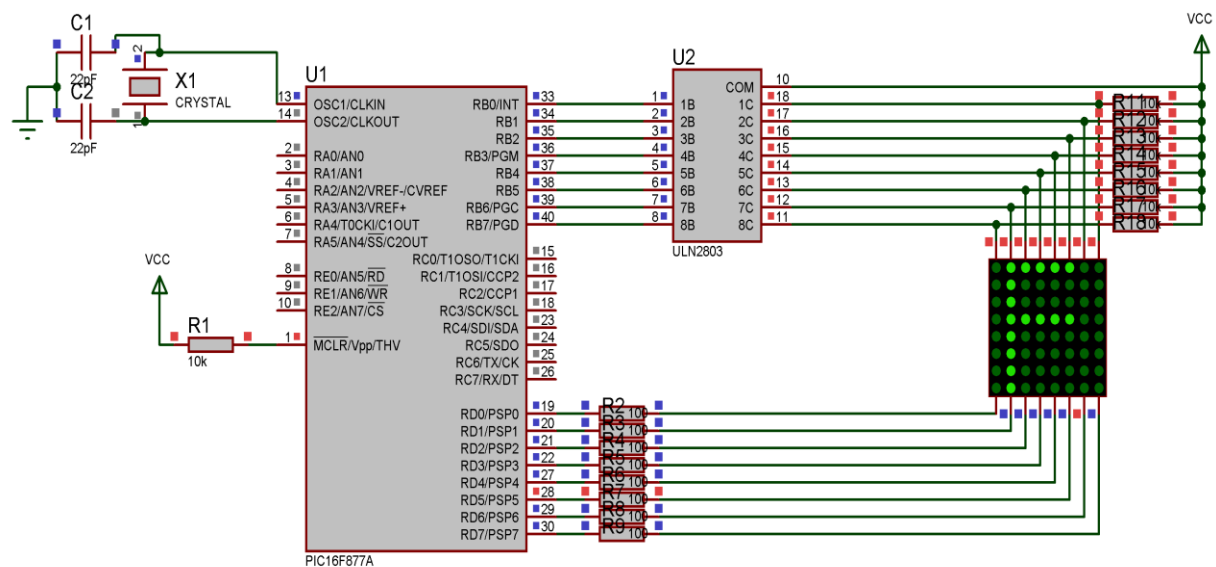


## Code:

```
void main() {  
    TRISB = 0x00;  
    TRISD = 0x00;  
  
    while(1)  
    {  
        PORTD = 1<<7;  
        PORTB = 0x00;  
        Delay_us(100);  
  
        PORTD = 1<<6;  
        PORTB = 0x00;  
        Delay_us(100);  
  
        PORTD = 1<<5;  
        PORTB = 0x90;  
        Delay_us(100);  
        PORTD = 1<<4;  
        PORTB = 0x90;  
        Delay_us(100);  
  
        PORTD = 1<<3;  
        PORTB = 0x90;  
        Delay_us(100);  
  
        PORTD = 1<<2;  
        PORTB = 0x90;  
        Delay_us(100);  
  
        PORTD = 1<<1;  
        PORTB = 0xFF;  
        Delay_us(100);  
  
        PORTD = 1<<0;  
        PORTB = 0x00;  
        Delay_us(100);  
    }  
}
```

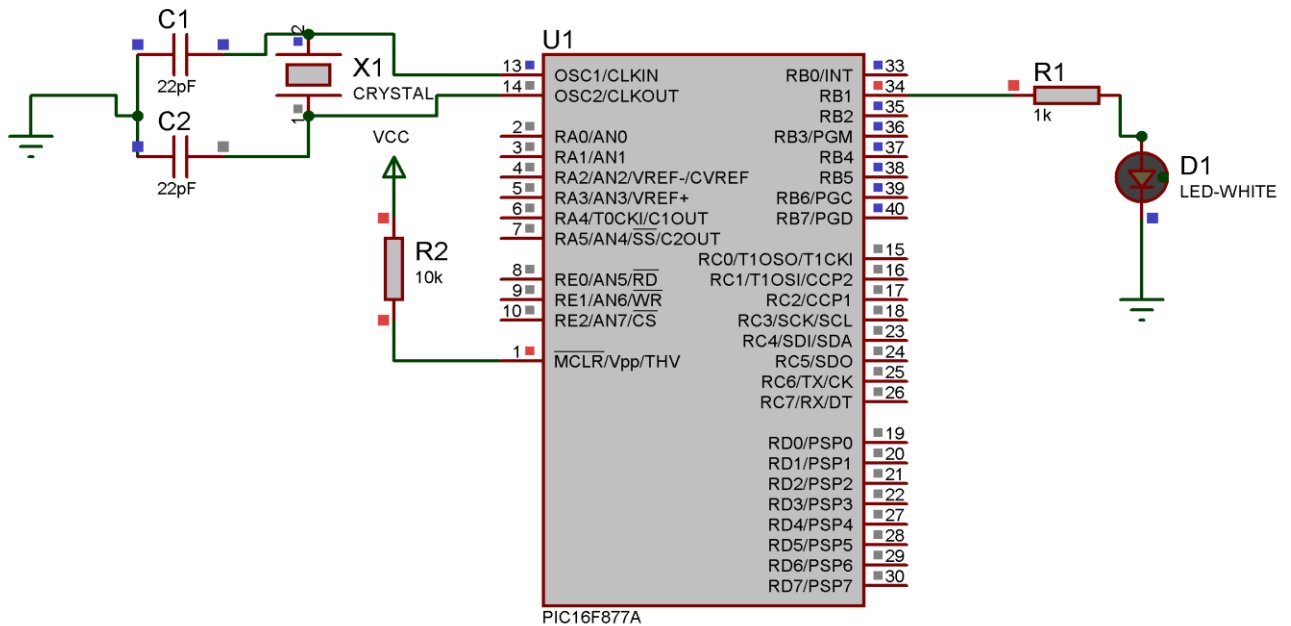
## Circuit Diagram:



### Code:

```
void main() {  
    TRISB = 0;  
    while(1)  
    {  
        PORTB = 0x02;  
        Delay_ms(1000);  
        PORTB = 0x00;  
        Delay_ms(1000);  
    }  
}
```

### Circuit Diagram:

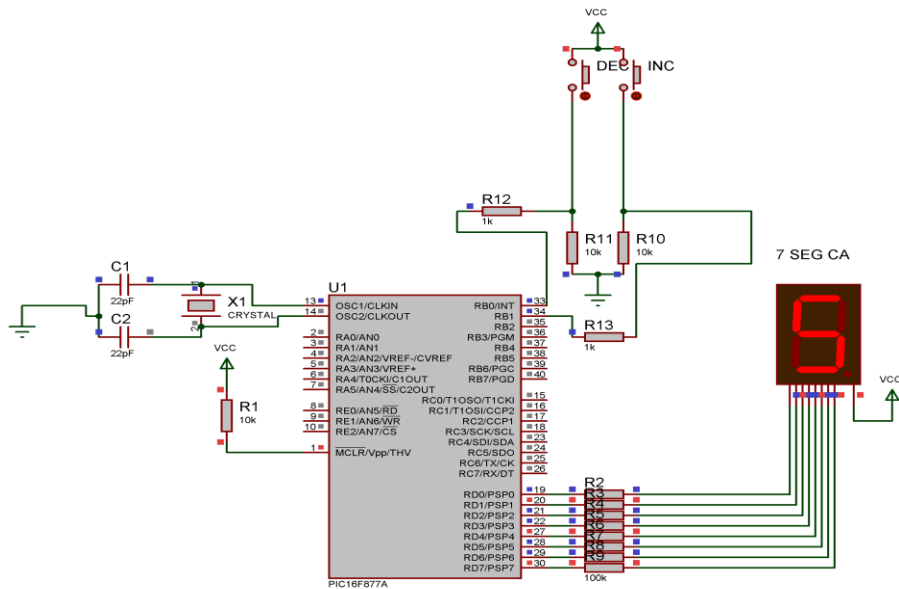


## Code:

```
char ch[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92,
0x82, 0xF8, 0x80, 0x90};
void main() {
    int i = 0;
    TRISD = 0x00;
    TRISB.F0 = 1;
    TRISB.F1 = 1;
    PORTD = 0xFF;
    while(1)
    {
        if(PORTB.F1 == 1) {
            Delay_ms(150);
            if(PORTB.F0 == 1) {
                if(i < 9) {
                    i++;
                }
                PORTD = ch[i];
            }
        }
    }
}
```

```
if(PORTB.F0 == 1)
{
    Delay_ms(150);
    if(PORTB.F0 == 1) {
        if(i > 0)
        {
            i--;
        }
        PORTD = ch[i];
    }
}
```

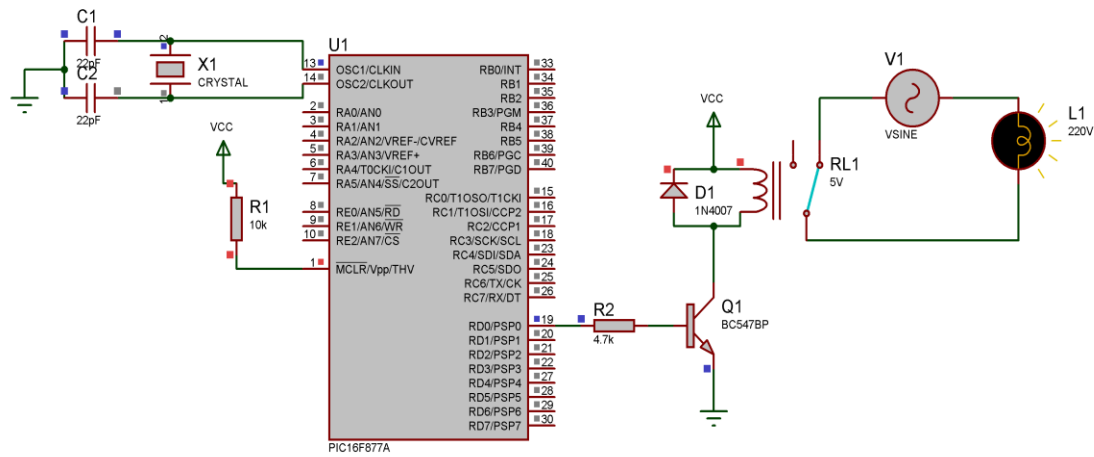
## Circuit Diagram:



## Code:

```
void main() {  
    TRISD.F0 = 0;  
    PORTD.F0 = 0;  
  
    while(1)  
    {  
        PORTD.F0 = 1;  
        Delay_ms(1000);  
        PORTD.F0 = 0;  
        Delay_ms(1000);  
    }  
}
```

## Circuit Diagram:

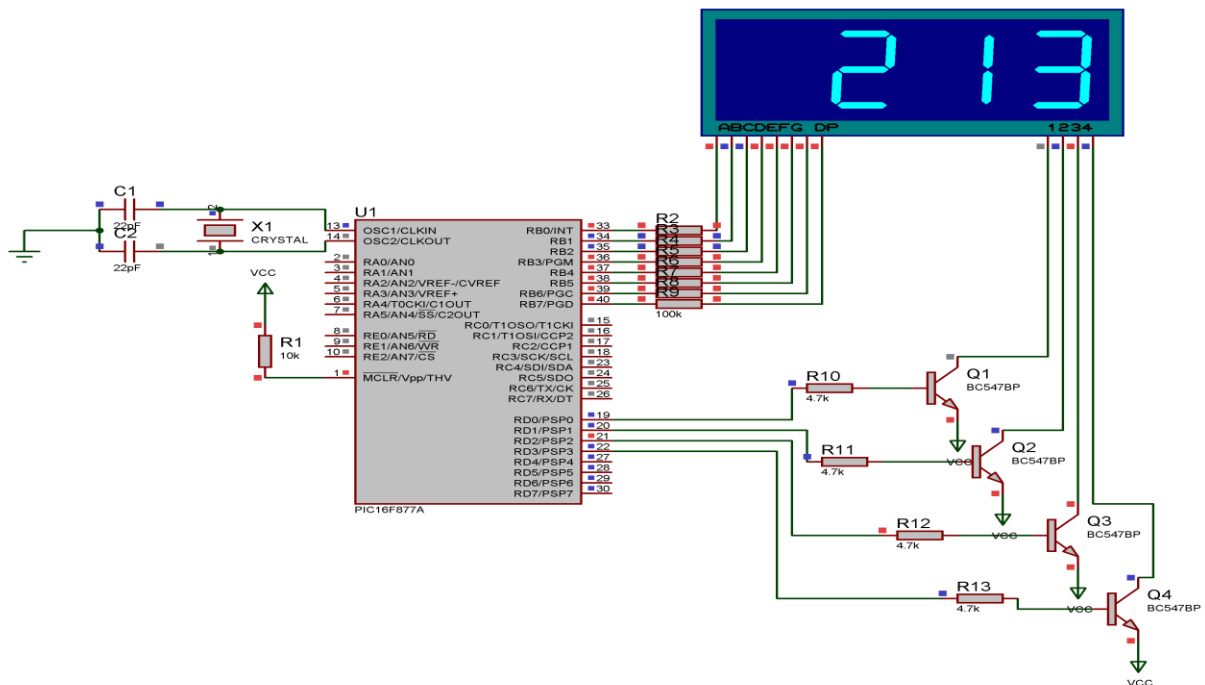


## Code:

```
char ch[] = {0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92,
0x82, 0xF8, 0x80, 0x90};
void main() {
    int i, j, firstDigit, secondDigit, thirdDigit,
fourthDigit;
    TRISB = 0x00;
    PORTB = 0x00;
    TRISD = 0x00;
    PORTD = 0x00;
    while(1){
        for(i = 0; i < 10000; i++) {
            firstDigit = i / 1000;
            secondDigit = (i / 100) % 10;
            thirdDigit = (i / 10) % 10;
            fourthDigit = i % 10;
            for(j = 0; j < 12; j++) {
                if(i >= 1000) {
                    PORTD.F0 = 1;
                    PORTB = ch[firstDigit];
                    Delay_ms(5);
                    PORTD.F0 = 0;
                }
            }
        }
    }
```

```
        if(i >= 100){
            PORTD.F1 = 1;
            PORTB = ch[secondDigit];
            Delay_ms(5);
            PORTD.F1 = 0;
        }
        if(i >= 10){
            PORTD.F2 = 1;
            PORTB = ch[thirdDigit];
            Delay_ms(5);
            PORTD.F2 = 0;
        }
        PORTD.F3 = 1;
        PORTB = ch[fourthDigit];
        Delay_ms(5);
        PORTD.F3 = 0;
    }
}
```

## Circuit Diagram:



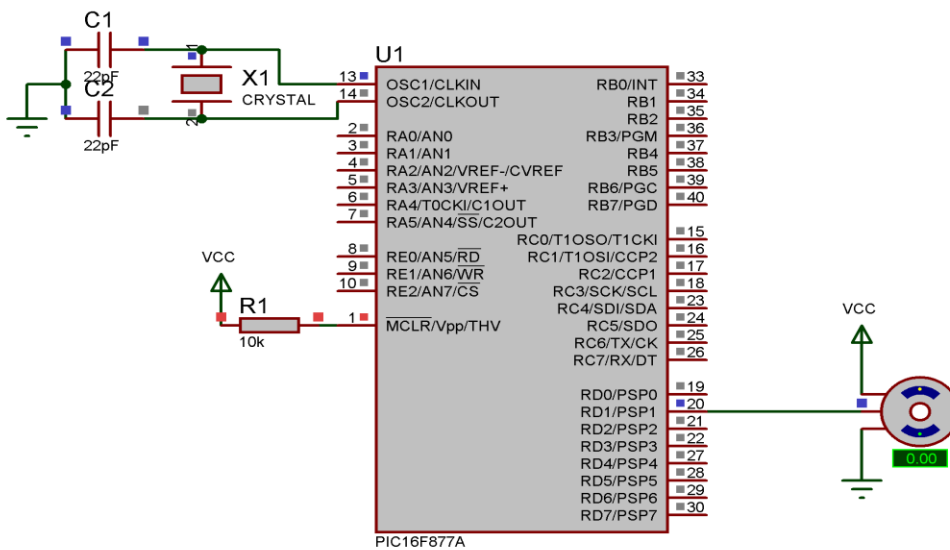
## Code:

```
void rotate0()
{
    int i = 0;
    for(i = 0; i < 50; i++)
    {
        PORTD.F1 = 1;
        Delay_us(800);
        PORTD.F1 = 0;
        Delay_us(19200);
    }
}
```

```
void rotate90()
{
    int i = 0;
    for(i = 0; i < 50; i++)
    {
        PORTD.F1 = 1;
        Delay_us(1500);
        PORTD.F1 = 0;
        Delay_us(18500);
    }
}
```

```
void rotate180()
{
    int i = 0;
    for(i = 0; i < 50; i++)
    {
        PORTD.F1 = 1;
        Delay_us(2200);
        PORTD.F1 = 0;
        Delay_us(17800);
    }
}
```

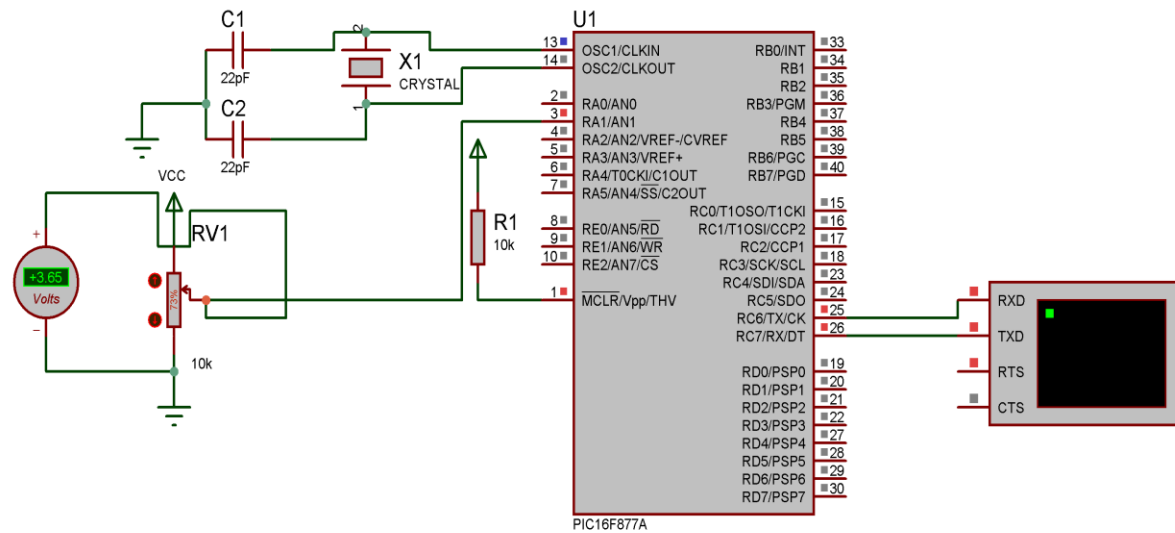
## Circuit Diagram:



## Code:

```
int ADCValue;
char ch[4];
void main() {
    UART1_Init(9600);
    ADC_Init();
    while(1)
    {
        ADCValue = ADC_Read(1);
        IntToStr(ADCValue, ch);
        UART1_Write_Text("Analog Value- ");
        UART1_Write_Text(ch);
        strcpy(ch, "");
        UART1_Write(13);
        Delay_ms(1000);
    }
}
```

## Circuit Diagram:



## LIST OF EXPERIMENTS

Serial no.	Experiments	Page
1	To design and implement an LED Blinking circuit using a PIC microcontroller	
2	To design and implement Counting from 0 to 9 on a 7-Segment Display using PIC Microcontroller.	
3	To design and implement Controlling AC Current using DC Current with a Mechanical Relay	
4	To design and implement Displaying a 4-Digit Number on a 7-Segment using Multiplexing with PIC Microcontroller	
5	To design and implement Analog signal input in the microcontroller or Display ADC value in the virtual terminal	
6	To design and implement Dot Matrix Display Interfacing With PIC16F877A Microcontroller	
7	To design and implement Interfacing Servo Motor with PIC Microcontroller	