Experiment Name: DC Motor Speed Control Using PWM and Microcontroller.

PWM: PWM stands for Pulse Width Modulation. A modulation technique that generates variable-width pulses to represent the amplitude of an analog input signal.

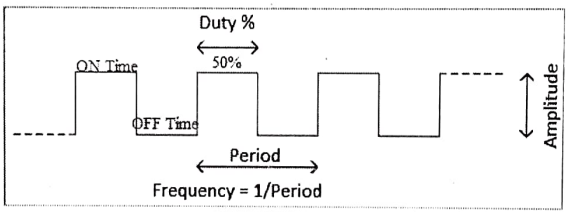
To understand PWM as a type of signal which can be produced from a digital IC such as microcontroller or 555 timer. The signal thus produced will have a train of pulses and these pulses will be in form of a square wave. That is, at any given instance of time the wave will either be high or will be low. For the ease of understanding let us consider a 5V PWM signal, in this case the PWM signal will either be 5V (high) or at ground level 0V (low). The duration at which the signals stays high is called the "*on time*" and the duration at which the signal stays low is called as the "*off time*".

Duty cycle of the PWM:

As told earlier, a PWM signal stays on for a particular time and then stays off for the rest of the period. The percentage of time in which the PWM signal remains HIGH (on time) is called as duty cycle. If the signal is always ON it is in 100% duty cycle and if it is always off it is 0% duty cycle. The formulae to calculate the duty cycle is shown below.

Duty Cycle =Turn ON time/ (Turn ON time + Turn OFF time)

The following image represents a PWM signal with 50% duty cycle. As you can see, considering an entire **time period** (on time + off time) the PWM signal stays on only for 50% of the time period.



By controlling the Duty cycle from 0% to 100% we can control the "*on time*" of PWM signal and thus the width of signal. Since we can modulate the width of the pulse, it got its iconic name "*Pulse width Modulation*"

Frequency of a PWM:

The frequency of a PWM signal determines how fast a PWM completes one period. One Period is the complete ON and OFF time of a PWM signal as shown in the above figure. The formulae to calculate the Frequency is given below

Frequency = 1/Time Period
Time Period = On time + Off time

Normally the PWM signals generated by microcontroller will be around 500 Hz, such high frequencies will be used in high speed switching devices like inverters or converters. But not all applications require high frequency. For example to control a servo motor we need to produce PWM signals with 50Hz frequency, so the frequency of a PWM signal is also controllable by program for all microcontrollers.

Features of L293D:

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Output Current 1 A Per Channel (600 mA for L293D)

- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

The Pin diagram of the L293D is given below:

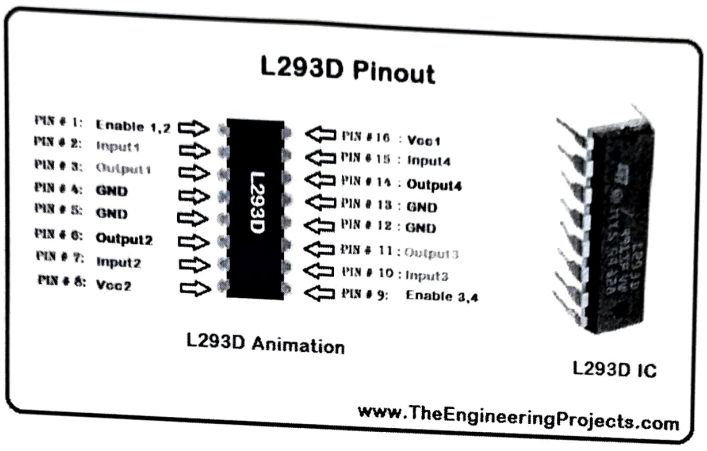


Figure: Pin diagram of L293D

Circuit Diagram:

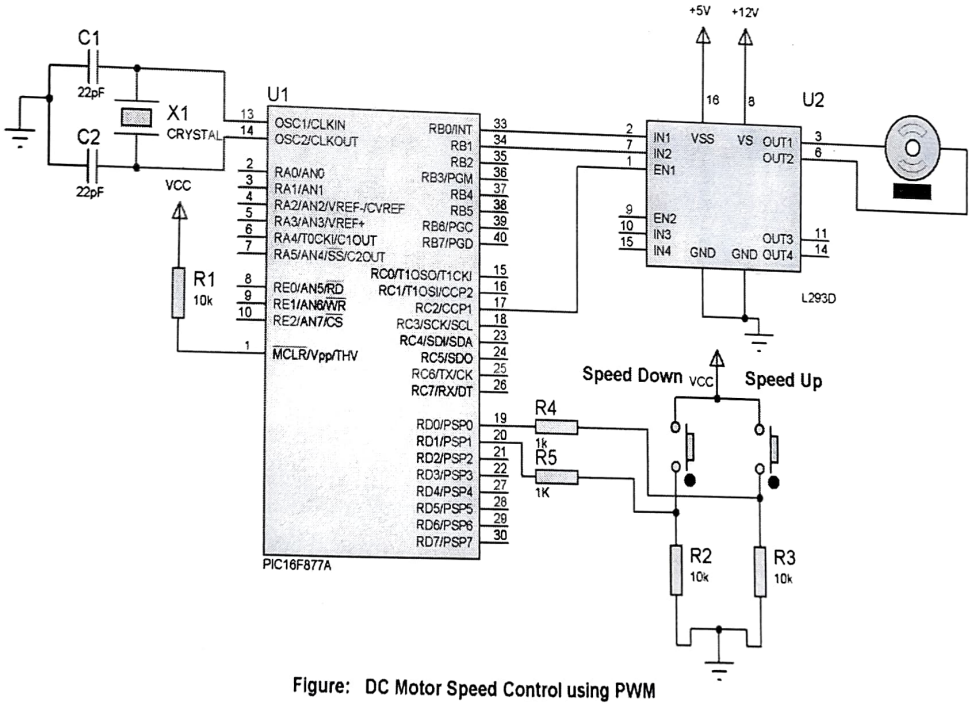


Figure: DC Motor Speed Control using PWM

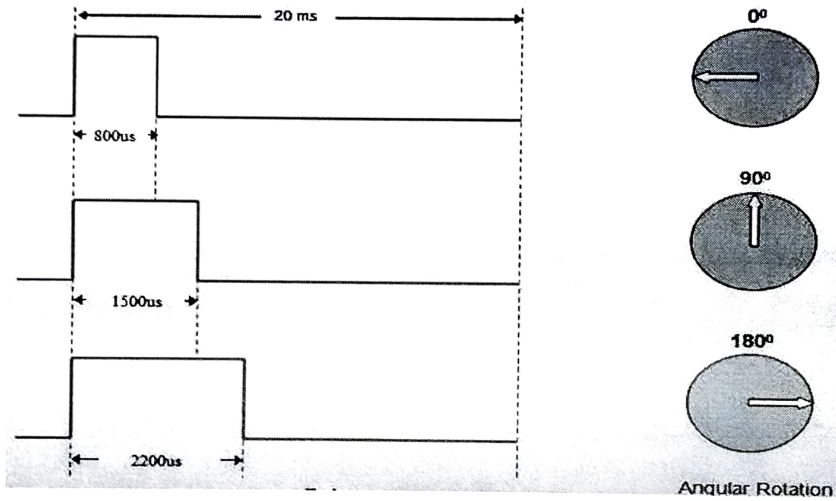
MikroC Code:

```
void main()
{
 short duty  = 0; //initial value for duty

 TRISD = 0xFF; //PORTD as input
 TRISB = 0x00; //PORTB as output
 //Run motor in anticlock wise
 PORTB.F0=0xff;
 PORTB.F1=0x00;

 PWM1_Init(1000);  //Initialize PWM1
 PWM1_Start();  //start PWM1
 PWM1_Set_Duty(duty); //Set current duty for PWM1

 while (1)       // endless loop
 {
   if (RD0_bit && duty<250) //if button on RD0 pressed
   {
     Delay_ms(100);
     if (RD0_bit && duty<250)
     {
        duty = duty + 10;  //increment current_duty
        PWM1_Set_Duty(duty);  //Change the duty cycle
     }
   }
   if (RD1_bit && duty >0) //button on RD1 pressed
   {
    Delay_ms(100);
    if (RD1_bit && duty >0)
    {
       duty = duty - 10;  //decrement duty
       PWM1_Set_Duty(duty);
    }
   }
  Delay_ms(10);      // slow down change pace a little
 }
}
```

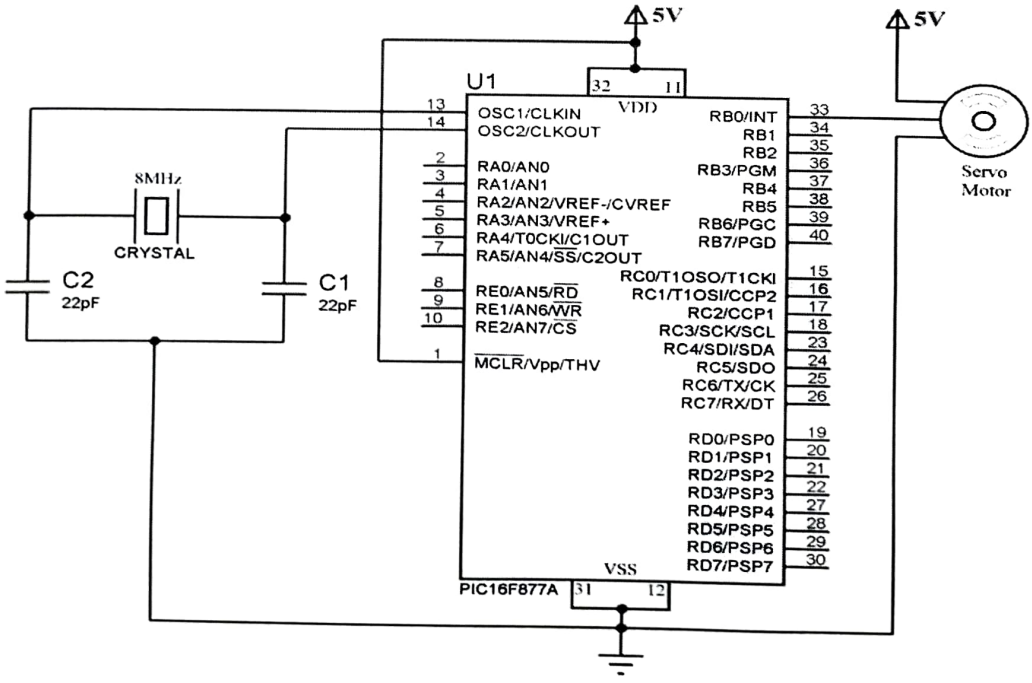
Experiment Name: Interfacing Servo Motor with PIC Microcontroller

Theory:

Servo Motor uses error sensing negative feedback to control the precise angular position. Servos are used for precise positioning in robotic arms, legs, RC Aeroplanes, Helicopters etc. Please read the article Servo Motor for more information about its working and construction. Hobby Servo Motors have three wires, two of them (RED and BLACK) are used to given power and the third one is used to give control signals. Servo can be easily be controlled using microcontrollers using Pulse Width Modulated (PWM) signals on the control wire. Here we are using a servo whose angular rotation is limited to 0 – 180°. We can control the exact angular position by using a pulse, whose width varying from 1 millisecond to 2 millisecond on the control wire. The actual behavior of a particular motor depends upon its manufacture, please refer the datasheet of the particular motor for that.



Circuit Diagram:



Code in Mikro C:

```
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);
}
```