Experiment no: 11

Name of the experiment: Interfacing stepper motor with PIC microcontroller.

objective(s):

1. To know ~~the~~ about the stepper motor

2. To learn about the microcontroller.

Theory: A stepper motor is a brushless, synchronous DC electric motor, which divides the full rotation into a number of equal steps. It finds great application in field of microcontrollers such as robotics. Please refer the article stepper motor or step motor for detailed information about working of stepper motor, types and modes of operation. Unipolar motor is the most popular stepper motor among electronics hobbyist because of its ease of operation and availability. Here I explaining the working of Unipolar and Bipolar Stepper Motor with PIC 16F877A Microcontroller.

Stepper motor can be easily interfaced with PIC microcontroller by using readymade ICs such as, L293D or ULN2003.

Wave Drive: In this mode only one stator electromagnet is energised at a time. It has the same number of steps as the full step drive but the torque is significantly less. It is rearely used. It can be used where power consumption is more important than torque

Wave Drive stepping sequence				
Step	A	B	C	D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Full Drive: In this model two stator electromagnets are energised at a time. It is the usual method used for driving and the motor will run at its full torque in this mode of driving.

Full Drive stepping sequence				
Step	A	B	C	D
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

Half Drive: In this stepping mode, alternatively one and two phases are energised. This mode is commonly used to increase the angular resolution of the motor but the torque is less approximately 70% of its half step position. We can see that the

11-3

angular resolution doubles in Half Drive mode.

Half Drive stepping sequence				
step	A	B	C	D
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

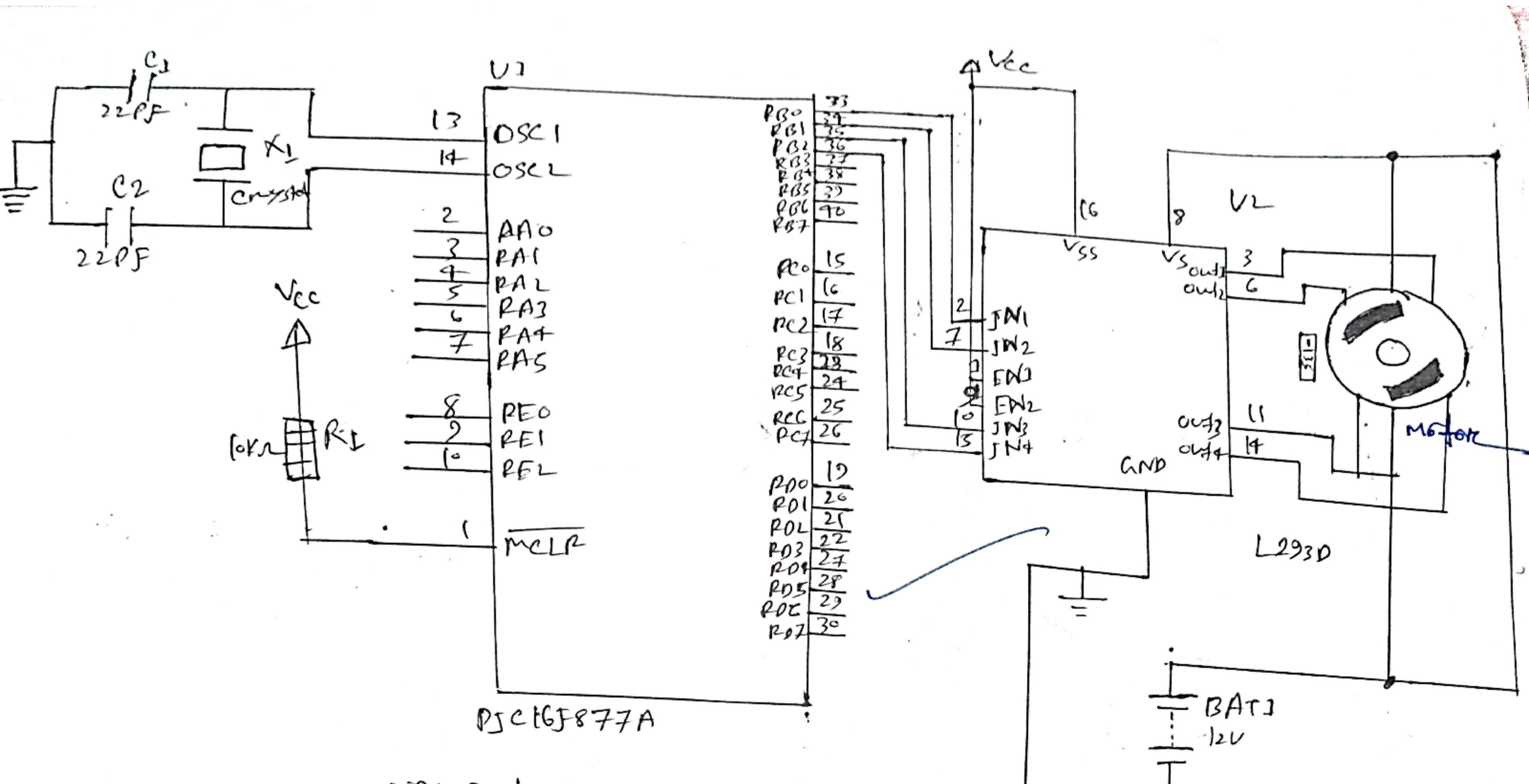

fig1 Interfacing stepper motor with PIC microcontroller

Source code:

Code for wave Drive:

```
Void main()
 {CMCON =0x07;
  ADCON1=0x06;
  TRISB = 0;
   PORTB= 0X0F;
 do
 {PORTB = 0b00000001;
   Delay_ms (500);
   PORTB =0b00000010;
   Delay_ms (500);
   PORTB= 0b00000100;
   Delay_ms(500);
   PORTB =0b00001000;
   Delay_ms (500);
 }while(1);
}
```

Code for full drive:

```
 Void main()
 {cmcON= 0x07;
  AD CON1 =0x06;
 TRISB=0;
  PORTB= 0X0F;

 do
 {
 PORTB =0b00000011;
 Delay_ms(500);
 PORTB= 0b00000110;
 Delay_ms(500);
 PORTB= 0b00001100;
 Delay_ms (500);
 PORTB =0b00001001;
 Delay_ms(500);
```

```
while (1);
}
```

Code for Half Drive!

```
void main()
{CMCON = 0x07;
 ADCON1 = 0x06;
 TRISB = 0;
 PORTB = 0x0F;
do
 {PORTB = 0b00000001;
  Delay_ms(500);
  PORTB = 0b00000011;
  Delay_ms(500);
  PORTB = 0b00000010;
  Delay_ms(500);
  PORTB = 0b00000110;
  Delay_ms(500);
  PORTB = 0b00000100;
  Delay_ms(500);
  PORTB = 0b00001100;
  Delay_ms(500);
  PORTB = 0b00001000;
  Delay_ms(500);
  PORTB = 0b00001001;
  Delay_ms(500);
 } while(1);
}
```