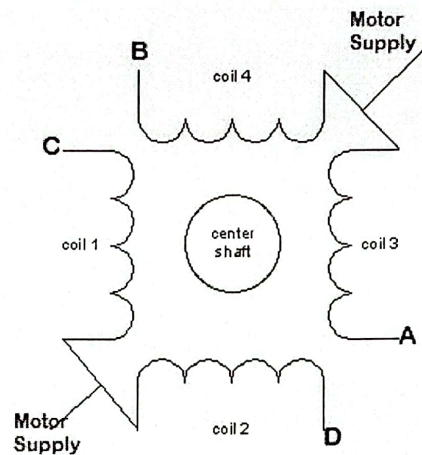


## Problem name: Interfacing Stepper Motor with PIC Microcontroller

**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. As I said in the article Stepper Motor or Step Motor, we have three different types of stepping modes for unipolar stepper motor.

**Note:** 1 – Represents Supply Voltage and 0 – Represents Ground



*Unipolar Stepper Motor Windings*

### 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 rarely 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 mode 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.



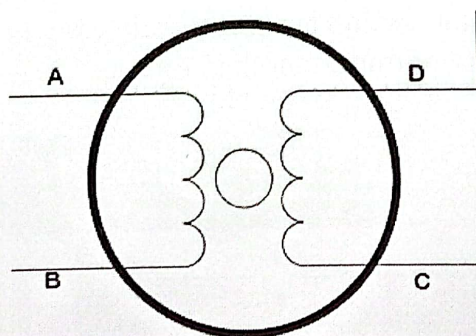
| Full Drive Stepping Sequence |   |   |   |   |
|------------------------------|---|---|---|---|
| Step                         | A | B | C | D |
| 1                            | 1 | 1 | 0 | 0 |
| 2                            | 0 | 1 | 1 | 0 |
| 3                            | 0 | 0 | 1 | 1 |
| 4                            | 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% at its half step position (when only a single phase is on). We can see that the 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 |

### Driving Bipolar Motor



*Bipolar Stepper Motor Windings*

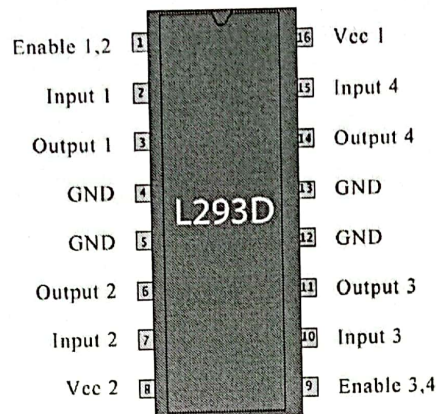
Bipolar motors are simpler in construction as it contains two coil and no centre tap. Being simple, driving is little complex compared to unipolar motors. To reverse the magnetic polarity of stator windings, current through it must be reversed. For this we should use H-Bridge. Here I using L293d, H-Bridge Motor Driver for that. We can distinguish bipolar motors from unipolar motors by measuring the coil resistance. In bipolar motors we can find two wires with equal resistance.

| Bipolar Motor Stepping Sequence |   |   |   |   |
|---------------------------------|---|---|---|---|
| Step                            | A | B | C | D |



|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 |
| 3 | 0 | 1 | 0 | 0 |
| 4 | 0 | 0 | 0 | 1 |

### Interfacing Unipolar Stepper Motor with PIC Microcontroller:



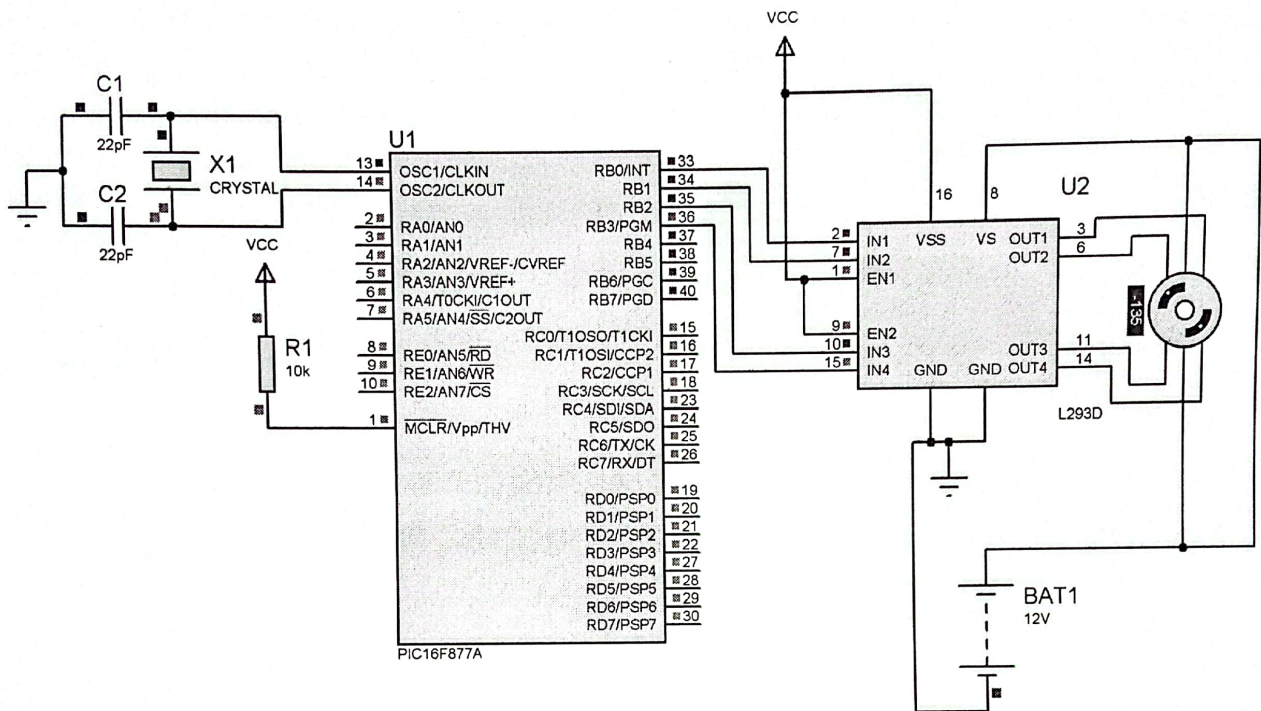
[www.electroSome.com](http://www.electroSome.com)

PIN Diagram of L293D

L293D and L293 are dual H-bridge motor drivers. The L293D can provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V while L293 can provide up to 1A at same voltages. Both ICs are designed to drive inductive loads such as dc motors, bipolar stepping motors, relays and solenoids as well as other high-current or high-voltage loads in positive-supply applications. All inputs of these ICs are TTL compatible and output clamp diodes for inductive transient suppression are also provided internally. These diodes protect our circuit from the Back EMF of DC Motor. In both ICs drivers are enabled in pairs, with drivers 1 and 2 are enabled by a high input to 1,2EN and drivers 3 and 4 are enabled by a high input to 3,4EN. When drivers are enabled, their outputs will be active and in phase with their inputs. When drivers are disabled, their outputs will be off and will be in the high-impedance state.

ULN2003 is a monolithic high current and high voltage Darlington transistor arrays. ULN2003 consists of seven NPN Darlington Transistor pairs that have high-voltage outputs with common-cathode clamp diode for switching inductive loads. The collector-current rating of a single darlington pair can be up to 500mA. We can connect Darlington Transistor pairs in parallel if higher current capability is needed.

## Circuit Diagram:



## Code in Mikro C:

### MikroC Code for Wave Drive

```
void main()
{
    CMCON = 0x07; // To turn off comparators
    ADCON1 = 0x06; // To turn off analog to digital converters
    TRISB = 0; // PORT B as output port
    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);
}
```

### MikroC Code for Full Drive

```
void main()
{
    CMCON = 0x07; // To turn off comparators
    ADCON1 = 0x06; // To turn off analog to digital converters
```



```

TRISB = 0; // PORT B as output port
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);
}

```

## **MikroC Code for Half Drive**

```

void main()
{
    CMCON = 0x07; // To turn off comparators
    ADCON1 = 0x06; // To turn off analog to digital converters
    TRISB = 0; // PORT B as output port
    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);
}

```