***MOTOR DRIVER***

A motor driver is a little current amplifier.

The function of motor drivers is to take a low current signal and then turn it to a high current signal that can drive a motor.

There are many different kinds of motor drivers, many of the most common types categorized by maximum supply voltage, maximum output current, rated power dissipation, load voltage, packaging type and number of outputs.

The most common values for maximum supply voltage are 36V and 52V. The numbers of outputs vary from `1 to 12 but many common motor drivers have 1, 2 or 4 outputs.

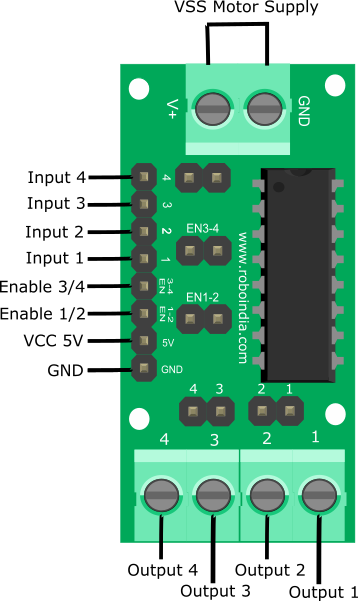
Applications for Motor Drivers:

Motor drivers can be found in a wide array of applications including:

* Relay and solenoid switching
* Stepping motor
* LED and incandescent displays
* Automotive applications
* Audio-visual equipment
* PC Peripherals
* Car audios
* Car navigation systems

The Motor Driver is a module for motors that allows you to control the working speed and direction of two motors simultaneously .This Motor Driver is designed and developed based on L293D IC.

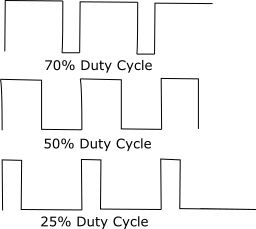
L293D is a 16 Pin Motor Driver IC. This is designed to provide bidirectional drive currents at voltages from 5 V to 36 V.



PWM Signals:

The DC motor speed in general is directly proportional to the supply voltage, so if reduce the voltage from 9 volts to 4.5 volts then our speed become half of what it originally had. But in practice, for changing the speed of a dc motor we cannot go on changing the supply voltage all the time. The speed controller PWM for a DC motor works by varying the average voltage supplied to the motor

The input signals we given to PWM controller might be an analog or digital signal according to the design of the PWM controller. The PWM controller accepts the control signal and adjusts the duty cycle of the PWM signal according to the requirements. These diagram below shows the waveforms obtained as output at different voltage requirements.



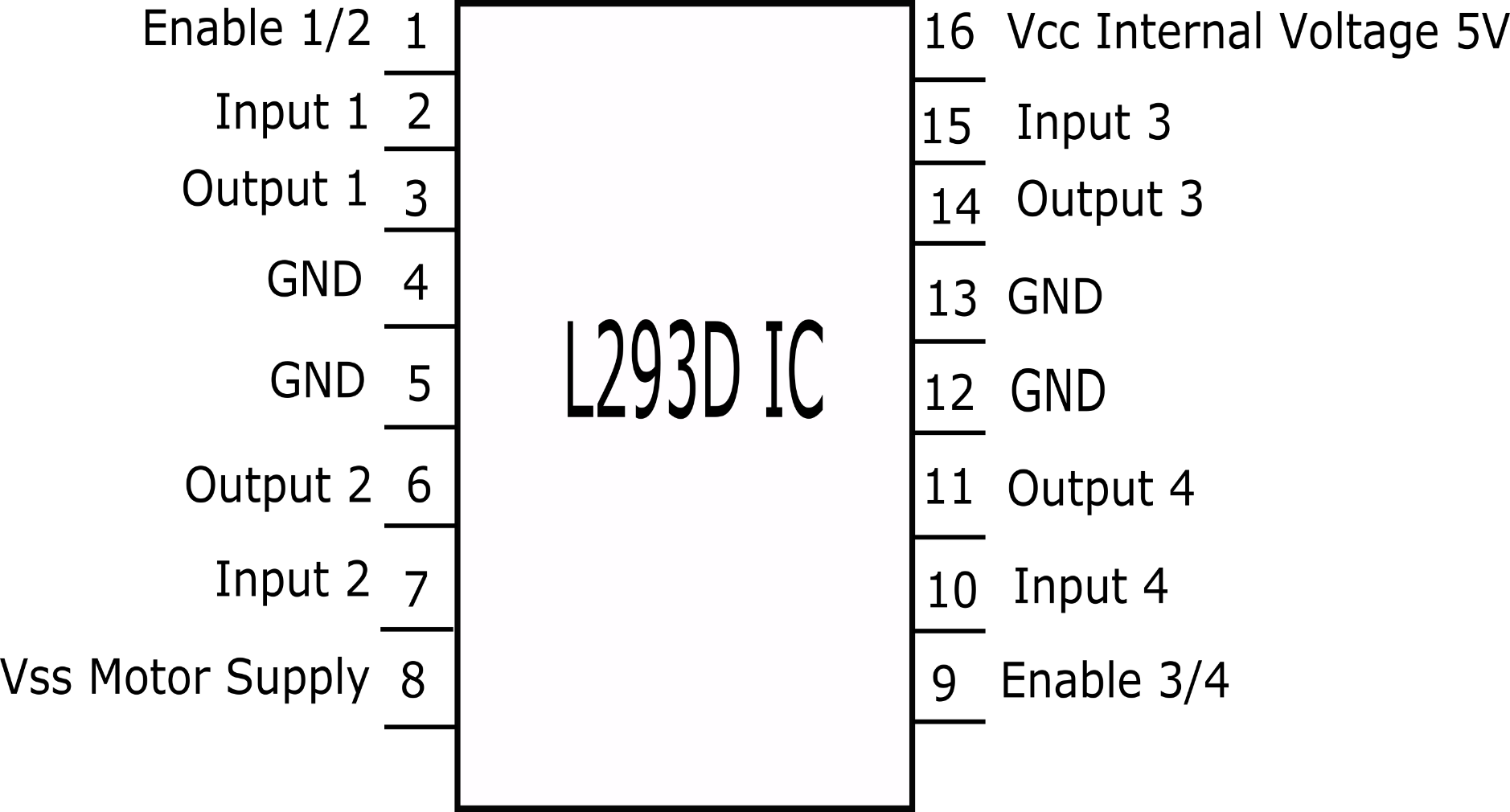
In these waves frequency is same but the ON and OFF times are different.

Hardware Required:

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Item** | **Quantity** |
| 1 | [NodeMCU](https://roboindia.com/store/nodemcu-amica-cp2102?search=nodemcu) | 1 |
| 2 | [Motor Driver](https://roboindia.com/store/Motor-Driver-DC-Geared-Stepper) | 1 |
| 3 | [DC Motor](https://roboindia.com/store/BO-Motors-90-Degrees?search=bo%20motor) | 2 |
| 4 | [Female to Male Jumper wire](https://roboindia.com/store/male-female-connectors-jumper-wire-electrical-electronics-embedded-system-prototype?product_id=253) | 10 |
| 5 | [6 x AA Battery](https://roboindia.com/store/6xaa-battery-holder-black) | 1 |

L293D IC Pin Out

The L293D is a 16 pin IC, with eight pins, on each side, to controlling of two DC motor simultaneously. There are 4 INPUT pins, 4 OUTPUT pins and 2 ENABLE pin for each motor.



Pin 1: When Enable1/2 is HIGH, Left part of IC will work, i.e motor connected with pin 3 and pin 6 will rotate.

Pin 2: Input 1, when this pin is HIGH the current will flow though output 1.

Pin 3: Output 1, this pin is connected with one terminal of motor.

Pin 4/5: GND pins

Pin 6: Output 2, this pin is connected with one terminal of motor.

Pin 7: Input 2, when this pin is HIGH the current will flow though output 2.

Pin 8: VSS, this pin is used to give power supply to connected motors from 5V to 36V maximum depends on Motor connected.

Pin 9: When Enable 3/4 is HIGH, Right part of IC will work, i.e motor connected with pin 11 and pin 14 will rotate.

Pin 10: Input 4, when this pin is HIGH the current will flow though output 4.

Pin 11: Output 4, this pin is connected with one terminal of motor.

Pin 12/13: GND pins

Pin 14: Output 3, this pin is connected with one terminal of motor.

Pin 15: Input 3, when this pin is HIGH the current will flow though output 3.

Pin 16: VCC, for supply power to IC i.e 5V.

Connections with NodeMCU

1. Module 5V (VCC) - NodeMCU Vin.

2. Module GND - NodeMCU GND.

3. Module 1 - NodeMCU D3.

4. Module 2 - NodeMCU D2.

5. Module 3 - NodeMCU D1.

6. Module 4 - NodeMCU D0.

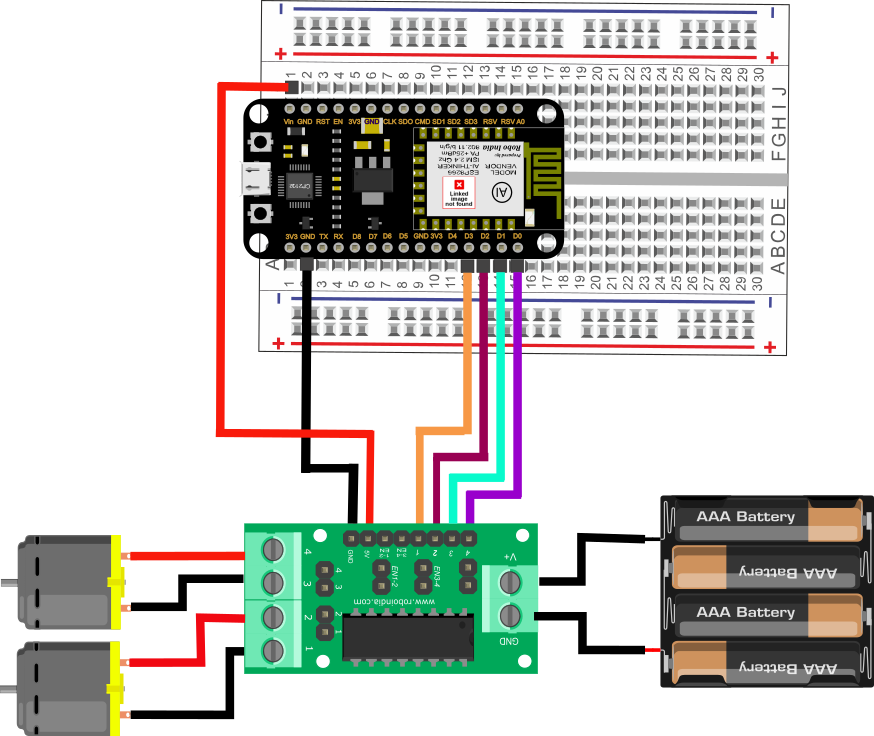
7. Module EN12 - NodeMCU D6.

8. Module EN34 - NodeMCU D5.

9. Module Motor terminals - DC motors.

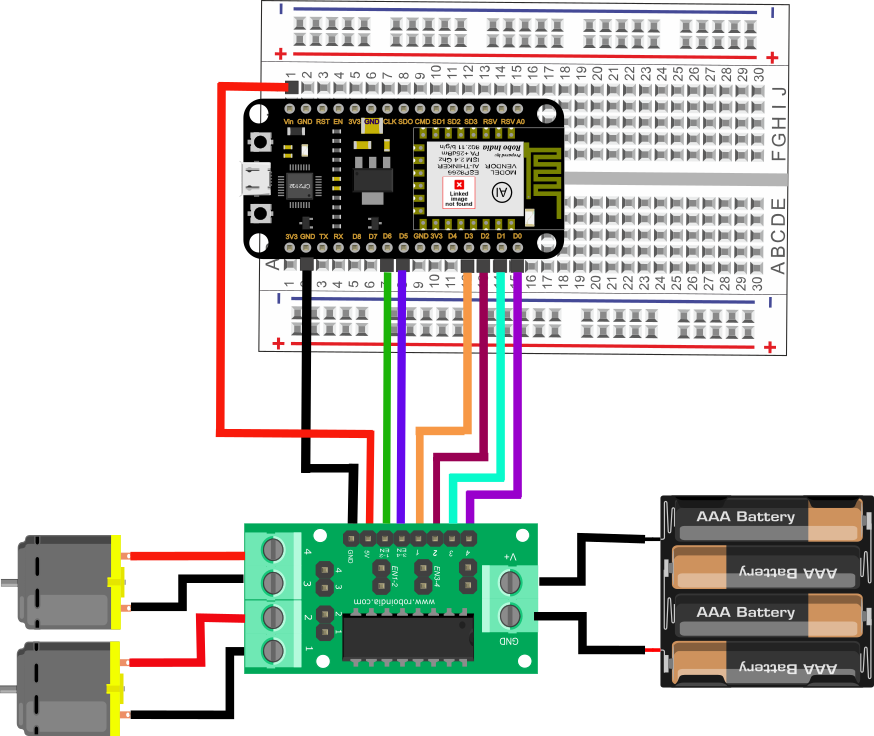
10. Module VSS power terminal- External power source of 9V.

Connection with NodeMCU



Make sure that the Jumpers are preset on the Enable 1-2 and Enable 3-4 pins of module, so that motor will be enabled and work at maximum speed.

Connection with NodeMCU using PWM Signals



Make sure to remove the Jumper preset on Enable pins of module, so that we can connect PWM input to this pin and control the speed of motors. If we connect these pins to ground , then the motor will get disabled. 

Make the connection as shown above. 

Programming1:

Here is the code to run this circuit.

[*You may download this code (Arduino Sketch) from here.*](https://roboindia.com/tutorials/admin/source32145898/shruti/Motor_Driver_arduino/MotorDriver_NodeMCU.zip)

//Tutorial on L293D IC Motor Driver with NodeMCU

// http://roboindia.com/tutorials/

//Motor A

const int inputPin1  = 5;    // Pin 15 of L293D IC

const int inputPin2  = 16;    // Pin 10 of L293D IC

//Motor B

const int inputPin3  = 4;   // Pin  7 of L293D IC

const int inputPin4  = 0;   // Pin  2 of L293D IC

void setup()

{

   pinMode(inputPin1, OUTPUT);

   pinMode(inputPin2, OUTPUT);

   pinMode(inputPin3, OUTPUT);

   pinMode(inputPin4, OUTPUT);

}

void loop()

{

   digitalWrite(inputPin1, HIGH);

   digitalWrite(inputPin2, LOW);

   digitalWrite(inputPin3, HIGH);

   digitalWrite(inputPin4, LOW);

}

Output-1

After uploading the first code you can see both motors get start rotating with maximum speed.

Programming2:

Here is the code to control the motors with PWM signals.

//Tutorial on L293D IC Motor Driver using PWM Signals

// http://roboindia.com/tutorials/

//Motor A

const int inputPin1  = 5;    // Pin 15 of L293D IC, D1 Pin of NodeMCU

const int inputPin2  = 16;    // Pin 10 of L293D IC, D0 Pin of NodeMCU

//Motor B

const int inputPin3  = 4;    // Pin  7 of L293D IC, D2 Pin of NodeMCU

const int inputPin4  = 0;    // Pin  2 of L293D IC, D3 Pin of NodeMCU

int EN1 = 12;                 // Pin 1 of L293D IC, D6 Pin of NodeMCU

int EN2 = 14;                 // Pin 9 of L293D IC, D5 Pin of NodeMCU

void setup()

{

   pinMode(EN1, OUTPUT);   // where the motor is connected to

   pinMode(EN2, OUTPUT);   // where the motor is connected to

   pinMode(inputPin1, OUTPUT);

   pinMode(inputPin2, OUTPUT);

   pinMode(inputPin3, OUTPUT);

   pinMode(inputPin4, OUTPUT);

**Serial**.begin(9600);

**Serial**.println("Enter values between 0 - 255");

}

void loop()

{

 if(**Serial**.available())

   {

     int speed = **Serial**.parseInt();

**Serial**.println(speed)

       analogWrite(EN1, speed);//sets the motors speed

       analogWrite(EN2, speed);//sets the motors speed

       digitalWrite(inputPin1, HIGH);

       digitalWrite(inputPin2, LOW);

       digitalWrite(inputPin3, HIGH);

       digitalWrite(inputPin4, LOW);

    }

}

Output-2

After uploading the program 2, open the Serial Monitor and send the input values to Arduino. You can control the speed of the DC motor by sending different values between 0 -255.

***Operation of Servo and Stepper Motors and Coding***

Servo Motor:

A servo motor is an electrical device which can push or pull or rotate an object with great precision. If you want to rotate objects at specific angles or distances, then we can use servo motors.

It is made up of simple motor which run through servo mechanism.

It also has a potentiometer and a gear kind of mechanism to control the angles at which it operates precisely.

Also we can get a very high torque servo motor in small and light weight packages.

Due to these features they are widely used in toy cars, RC gadgets and robotics!

Coding:  
#include <Servo.h>

Servo servo1;

int up = 90;

int down = 125;

int pinServo = 9;

void setup()

{

servo1.attach(9);

servo1.write(40);

pinMode(13,OUTPUT);

digitalWrite(13,HIGH);

delay(3000);

digitalWrite(13,LOW);

servo1.write(125);

delay(5000);

servo1.write(down);

delay(200);

servo1.write(up);

delay(770);

}

void loop()

{

servo1.write(90); //1

delay(200);

servo1.write(125);

delay(600);

}﻿

For coding with a servo motor we mainly include the package Servo.h and specify the working angles which in this case are 90 and 125.

We also use syntaxes such as <<.attach>> and <<.write>> to synchronize the servo motor with the arduino board.

And then we write a code to fulfil our demands.

Connections:

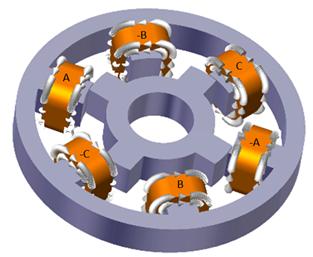
A servo motor has three pins which are ground, Vcc and a servo pin.

The Vcc goes to the 5V on the arduino UNO board and the ground pin goes to the Gnd on the board while the servo pin goes on the predefined servo pin which in this case is 9.

Stepper Motor:

A stepper motor is an electromechanical device it converts electrical power into mechanical power. Also it is a brushless, synchronous electric motor that can divide a full rotation into an expansive number of steps. The motor’s position can be controlled accurately without any feedback mechanism, as long as the motor is carefully sized to the application. Stepper motors are similar to switched reluctance motors.

The stepper motor uses the theory of operation for magnets to make the motor shaft turn a precise distance when a pulse of electricity is provided. The stator has eight poles, and the rotor has six poles. The rotor will require 24 pulses of electricity to move the 24 steps to make one complete revolution. Another way to say this is that the rotor will move precisely 15° for each pulse of electricity that the motor receives.

[](https://www.elprocus.com/wp-content/uploads/2013/05/Stepper-Motor.jpg)

### Types of Stepper Motor:

There are three main types of stepper motors, they are:

1. Permanent magnet stepper
2. Hybrid synchronous stepper
3. Variable reluctance stepper

**Permanent Magnet Stepper Motor:**Permanent magnet motors use a permanent magnet (PM) in the rotor and operate on the attraction or repulsion between the rotor PM and the stator electromagnets.

**Variable Reluctance Stepper Motor:**Variable reluctance (VR) motors have a plain iron rotor and operate based on the principle that minimum reluctance occurs with minimum gap, hence the rotor points are attracted toward the stator magnet poles.

**Hybrid Synchronous Stepper Motor:**Hybrid stepper motors are named because they use a combination of permanent magnet (PM) and variable reluctance (VR) techniques to achieve maximum power in a small package size.

### Advantages of Stepper Motor:

1. The rotation angle of the motor is proportional to the input pulse.
2. The motor has full torque at standstill.
3. Precise positioning and repeatability of movement since good stepper motors have an accuracy of 3 – 5% of a step and this error is non cumulative from one step to the next.
4. Excellent response to starting, stopping and reversing.
5. Very reliable since there are no contact brushes in the motor. Therefore the life of the motor is simply dependant on the life of the bearing.
6. The motors response to digital input pulses provides open-loop control, making the motor simpler and less costly to control.
7. It is possible to achieve very low speed synchronous rotation with a load that is directly coupled to the shaft.
8. A wide range of rotational speeds can be realized as the speed is proportional to the frequency of the input pulses.

### Applications:

1. **Industrial Machines** – Stepper motors are used in automotive gauges and machine tooling automated production equipments.
2. **Security**– new surveillance products for the security industry.
3. **Medical** – Stepper motors are used inside medical scanners, samplers, and also found inside digital dental photography, fluid pumps, respirators and blood analysis machinery.
4. **Consumer Electronics** – Stepper motors in cameras for automatic digital camera focus and zoom functions.

And also have business machines applications, computer peripherals applications.

### Operation of Stepper Motor:

Stepper motors operate differently from [DC brush motors](http://www.edgefxkits.com/four-quadrant-dc-motor-control-without-microcontroller), which rotate when voltage is applied to their terminals. Stepper motors, on the other hand, effectively have multiple toothed electromagnets arranged around a central gear-shaped piece of iron. The electromagnets are energized by an external control circuit, for example a microcontroller.

To make the motor shaft turn, first one electromagnet is given power, which makes the gear’s teeth magnetically attracted to the electromagnet’s teeth. The point when the gear’s teeth are thus aligned to the first electromagnet, they are slightly offset from the next electromagnet. So when the next electromagnet is turned ON and the first is turned OFF, the gear rotates slightly to align with the next one and from there the process is repeated. Each of those slight rotations is called a step, with an integer number of steps making a full rotation. In that way, the motor can be turned by a precise. Stepper motor doesn’t rotate continuously, they rotate in steps. There are 4 coils with 90o angle between each other fixed on the stator. The stepper motor connections are determined by the way the coils are interconnected.In stepper motor, the coils are not connected together. The motor has 90o rotation step with the coils being energized in a cyclic order, determining the shaft rotation direction. The working of this motor is shown by operating the switch. The coils are activated in series in 1 sec intervals. The shaft rotates 90o each time the next coil is activated. Its low speed torque will vary directly with current.

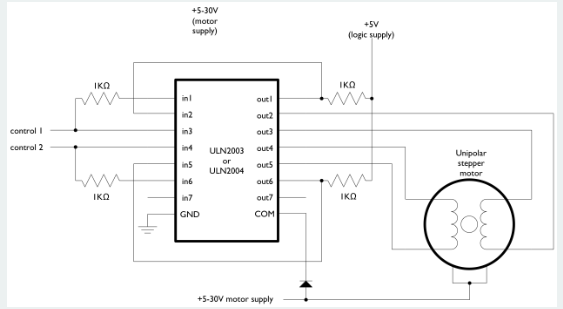
## Stepper Speed Control

Stepper motors, due to their unique design, can be controlled to a high degree of accuracy without any feedback mechanisms. The shaft of a stepper, mounted with a series of magnets, is controlled by a series of electromagnetic coils that are charged positively and negatively in a specific sequence, precisely moving it forward or backward in small "steps".

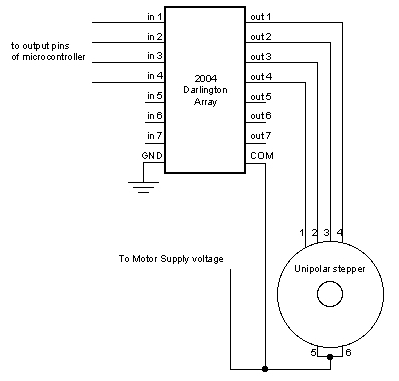
There are two types of steppers, Unipolars and Bipolars, and it is very important to know which type you are working with. For each of the motors, there is a different circuit.

UNIPOLAR::

Two pins:

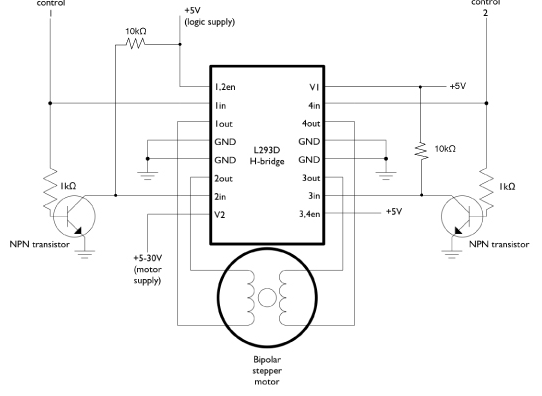


Four Pins:

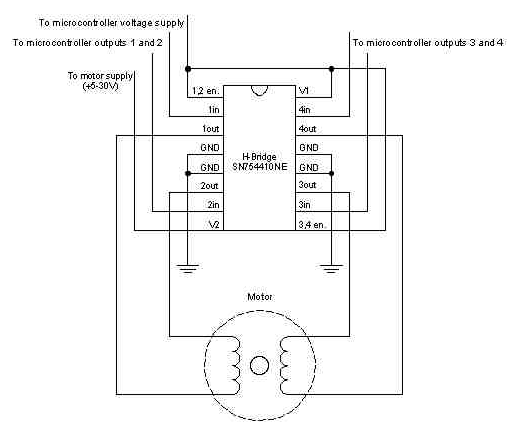


BIPOLAR::

Two pins:



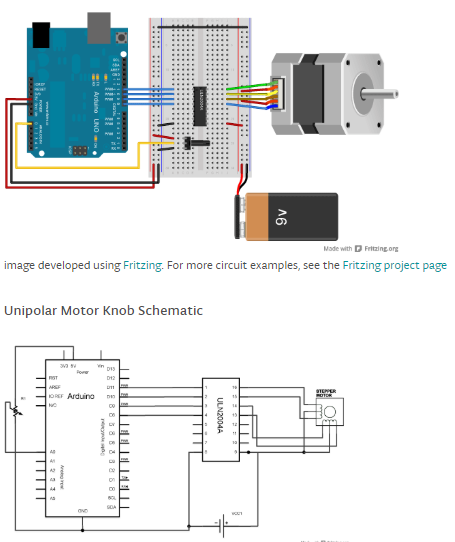
Four Pins:



### **Hardware Required**

* Arduino or Genuino Board
* 10k ohm potentiometer
* stepper motor
* U2004 Darlington Array (if using a unipolar stepper)
* SN754410ne H-Bridge (if using a bipolar stepper)
* power supply appropriate for your particular stepper
* hook-up wires
* breadboard

CIRCUIT DIAGRAM:



Code:

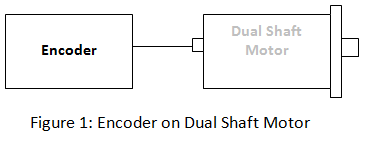
#include <Stepper.h>  
  
const int stepsPerRevolution = 200;  *// change this to fit the number of steps per revolution*  
*// for your motor*  
  
  
*// initialize the stepper library on pins 8 through 11:*  
Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);  
  
int stepCount = 0;  *// number of steps the motor has taken*  
  
void **setup**() {  
  *// nothing to do inside the setup*  
}  
  
void **loop**() {  
  *// read the sensor value:*  
  int sensorReading = analogRead(A0);  
  *// map it to a range from 0 to 100:*  
  int motorSpeed = map(sensorReading, 0, 1023, 0, 100);  
  *// set the motor speed:*  
  if (motorSpeed > 0) {  
    myStepper.setSpeed(motorSpeed);  
    *// step 1/100 of a revolution:*  
    myStepper.step(stepsPerRevolution / 100);  
  }  
}

**What is an Encoder?**

An [**encoder**](http://www.anaheimautomation.com/products/encoder/encoder-products.php) is a sensor of mechanical motion that generates digital signals in response to motion. As an electro-mechanical device, an encoder is able to provide motion control system users with information concerning position, velocity and direction. There are two different types of encoders: linear and [**rotary**](http://www.anaheimautomation.com/products/encoder/rotary-encoders-list.php?cID=422). A linear encoder responds to motion along a path, while a rotary encoder responds to rotational motion. An encoder is generally categorized by the means of its output. An incremental encoder generates a train of pulses which can be used to determine position and speed. An absolute encoder generates unique bit configurations to track positions directly.

**Block Diagram for Encoders**

**[http://www.anaheimautomation.com/manuals/forms/images/Encoder%20on%20Dual%20Shaft%20Motor%20-%20Encoder.png](http://www.anaheimautomation.com/products/encoder/encoder-products.php)**



**Basic Types of Encoders**

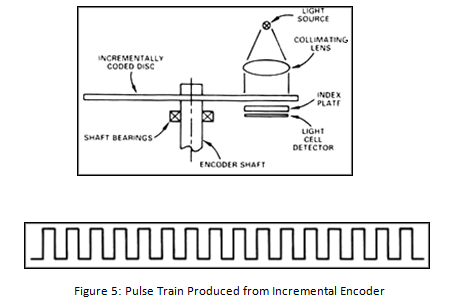
Linear and rotary encoders are broken down into two main types: the absolute encoder and the incremental encoder. The construction of these two types of encoders is quite similar; however they differ in physical properties and the interpretation of movement.

**Incremental Encoder**

**[](http://www.anaheimautomation.com/products/encoder/incremental-encoder-list.php?cID=422&cdID=362)**

An Incremental rotary encoder is also referred to as a quadrature encoder. This type of encoder utilizes sensors that use [**optical**](http://www.anaheimautomation.com/products/encoder/optical-incremental-rotary.php?tID=1054&pt=t&cID=422), mechanical or [**magnetic**](http://www.anaheimautomation.com/products/encoder/optical-incremental-rotary.php?tID=1063&pt=t&cID=422) index counting for angular measurement.

**How do Incremental Encoders Work?**   
  
[**Incremental rotary encoders**](http://www.anaheimautomation.com/products/encoder/rotary-encoders-list.php?cID=422) utilize a transparent disk which contains opaque sections that are equally spaced to determine movement. A light emitting diode is used to pass through the glass disk and is detected by a photo detector. This causes the encoder to generate a train of equally spaced pulses as it rotates. The output of incremental rotary encoders is measured in pulses per revolution which is used to keep track of position or determine speed.

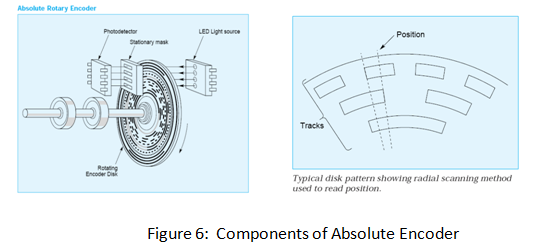


A single-channel output is commonly implemented in applications in which direction of movement is not significant. Instances in which direction sensing is important, a 2-channel, quadrature, output is used. The two channels, A and B, are commonly 90 electrical degrees out of phase and the electronic components determine the direction based off the phase relationship between the two channels. The position of an incremental encoder is done by adding up all the pulses by a counter.   
  
A setback of the incremental encoder is count loss which occurs during power loss. When restarting, the equipment must be referenced to a home position to reinitialize the counter. However, there are some incremental encoders, like those sold at Anaheim Automation, which come equipped with a third channel called the index channel. The index channel produces a single signal pulse per revolution of the encoder shaft and is often used as a reference marker. The reference marker is then denoted as a starting position which can resume counting or position tracking.   
  
**NOTE:** Incremental rotary encoders are not as accurate as absolute rotary encoders due to the possibility of interference or a misread.

**Absolute Encoder**

An absolute encoder contains components also found in incremental encoders. They implement a photodetector and LED light source but instead of a disk with evenly spaced lines on a disc, an absolute encoder uses a disk with concentric circle patterns.

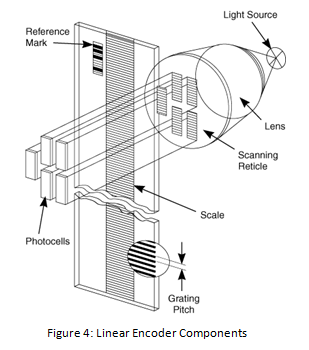
**How do Absolute Encoders Work?**   
  
Absolute encoders utilize stationary mask in between the photodetector and the encoder disk as shown below. The output signal generated from an absolute encoder is in digital bits which correspond to a unique position. The bit configuration is produced by the light which is received by the photodetector when the disk rotates. The light configuration received is translated into gray code. As a result, each position has its own unique bit configuration.



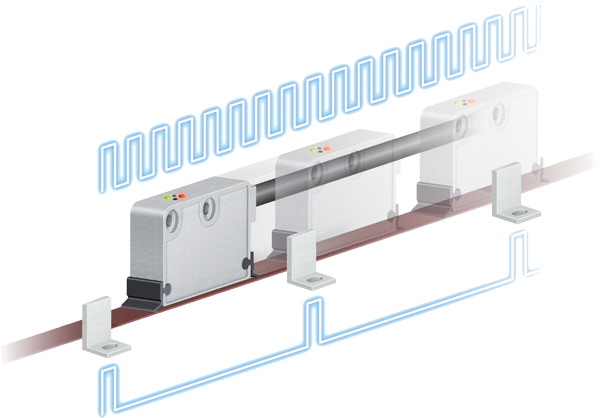
**Linear Encoder**

A linear encoder is a sensor, transducer or reading-head linked to a scale that encodes position. The sensor reads the scale and converts position into an analog or digital signal that is transformed into a digital readout. Movement is determined from changes in position with time. Both optical and magnetic linear encoder types function using this type of method. However, it is their physical properties which make them different.

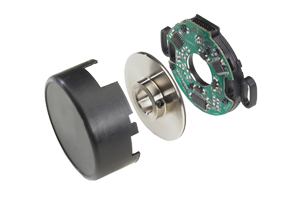
**How do Optical Linear Encoders Work?**   
  
The light source and lens produce a parallel beam of light which pass through four windows of the scanning reticle. The four scanning windows are shifted 90 degrees apart. The light then passes through the glass scale and is detected by photosensors. The scale then transforms the detected light beam when the scanning unit moves. The detection of the light by the photosensor produces sinusoidal wave outputs. The linear encoder system then combines the shifted signals to create two sinusoidal outputs which are symmetrical but 90 degrees out of phase from each other. A reference signal is created when a fifth pattern on the scanning reticle becomes aligned with an identical pattern on the scale.



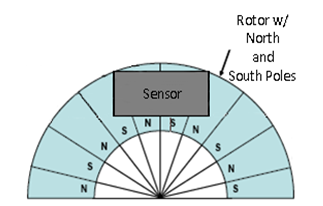
**How does a Linear Encoder Work?**   
  
A Linear Encoder system uses a magnetic sensor readhead and a magnetic scale to produce TTL or analog output for Channel A and B. As the magnetic sensor passes along the magnetic scale, the sensor detects the change in magnetic field and outputs a signal. This output signal frequency is proportional to the measuring speed and the displacement of the sensor. Since a linear encoder detects change in the magnetic field, the interference of light, oil, dust, and debris have no effect on this type of system; therefore they offer high reliability in harsh environments.



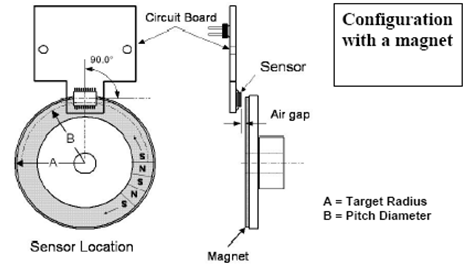
[**Magnetic Rotary Encoder**](http://www.anaheimautomation.com/products/encoder/optical-incremental-rotary.php?tID=1063&pt=t&cID=422)

**[](http://www.anaheimautomation.com/products/encoder/optical-incremental-rotary.php?tID=1063&pt=t&cID=422)**

A magnetic encoder consists of two parts: a rotor and a sensor. The rotor turns with the shaft and contains alternating evenly spaced north and south poles around its circumference. The sensor detects these small shifts in the position N>>S and S>>N. There many methods of detecting magnetic field changes, but the two primary types used in encoders are: Hall Effect and Magneto resistive. Hall Effect sensors work by detecting a change in voltage by magnetic deflection of electrons. Magneto resistive sensors detect a change in resistance caused by a magnetic field.



Hall-Effect sensing  
The Sensor produces and processes Hall-Effect signals producing a quadrature signal as is common with optical encoders. The output is generated by measuring magnetic flux distributions across the surface of the chip. The output accuracy is dependent on the radial placement of the IC with respect to the target magnet. The chip face should be parallel to the magnet so the magnet to sensor air gap is consistent across the sensor face.   
  
Magnetic encoders avoid the three vulnerabilities that optical encoders face:  
     • Seal failures which permit the entry of contaminents  
     • The optical disk may shatter during vibration or impact  
     • Bearing failures  
  
Magnetic devices designed effectively eliminate the first two failure modes and offer an opportunity to reduce bearing failures as well. Magnetic encoders do not make errors due to contamination because their sensors detect variations in magnetic fields imbedded in the rotor and oil, dirt and water do not affect these magnetic fields.



Hall-Effect sensors generally have lower cost and are less precise than magnetic resistive sensors. This means that Hall-Effect sensors, when used in an encoder produce more "jitter", or error in the signal caused by sensor variations.



**Commutation Encoders**

A commutation encoder contains the same fundamental components as incremental encoders but with the addition of commutation tracks alongside the outer edge of the disk for U/V/W output.

**How do Commutation Encoders Work?**

[**Commutation encoders**](http://www.anaheimautomation.com/manuals/forms/encoder-guide.php) utilize a transparent disk which includes opaque sections that are equally spaced to determine movement. A light emitting diode is used to pass through the glass disk and is detected by a photo detector. This causes the encoder to generate a train of equally spaced pulses as it rotates. The output of incremental rotary encoders is measured in pulses per revolution which is used to keep track of position or determine speed.   
  
The outer part of the encoder disk includes commutation tracks which provide a controller with information on the exact position of the motor poles, so that the proper controller input can be supplied to the motor. The commutation tracks of the encoder read the motor position and instruct the controller as to how to provide efficient and proper current to the motor to cause rotation. Commutation output for U/V/W can be in the form of differential output or open-collector (manufacturer dependent).

**How are Encoders Controlled?**

Encoders are controlled through the rotation the shaft it is mounted to. The shaft comes into contact with a hub which is in internal to the encoder. As the shaft rotates, it causes the disc, with both transparent and solid lines, to rotate across the circuitry of the encoder. The circuitry of the encoder contains an LED which is captured by a photoelectric diode and outputs pulses to the user. The speed at which the disc rotates will be dependent on the speed of the shaft the encoder is connected to. Anaheim Automation's [**optical**](http://www.anaheimautomation.com/products/encoder/optical-incremental-rotary.php?tID=1054&pt=t&cID=422) and [**magnetic**](http://www.anaheimautomation.com/products/encoder/optical-incremental-rotary.php?tID=1063&pt=t&cID=422) encoder lines are powered from a single +5VDC power source, and is able to sing and source 8mA each.