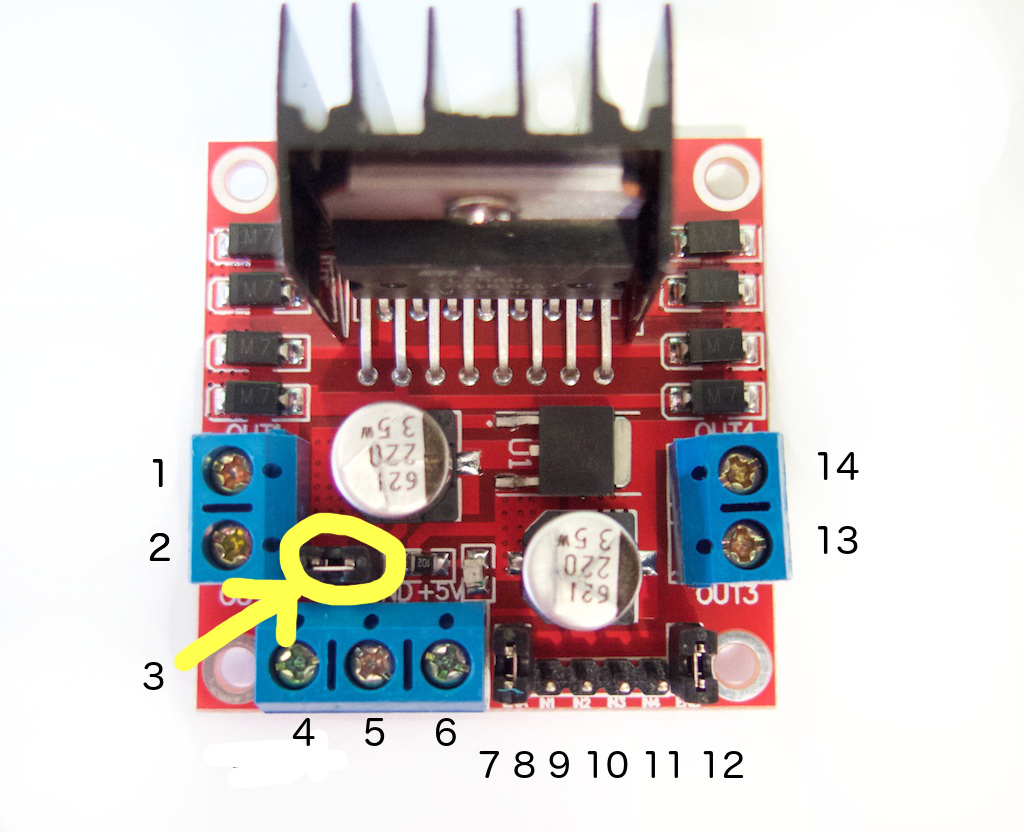
**Technocrats Electrical Vacation Task 2**

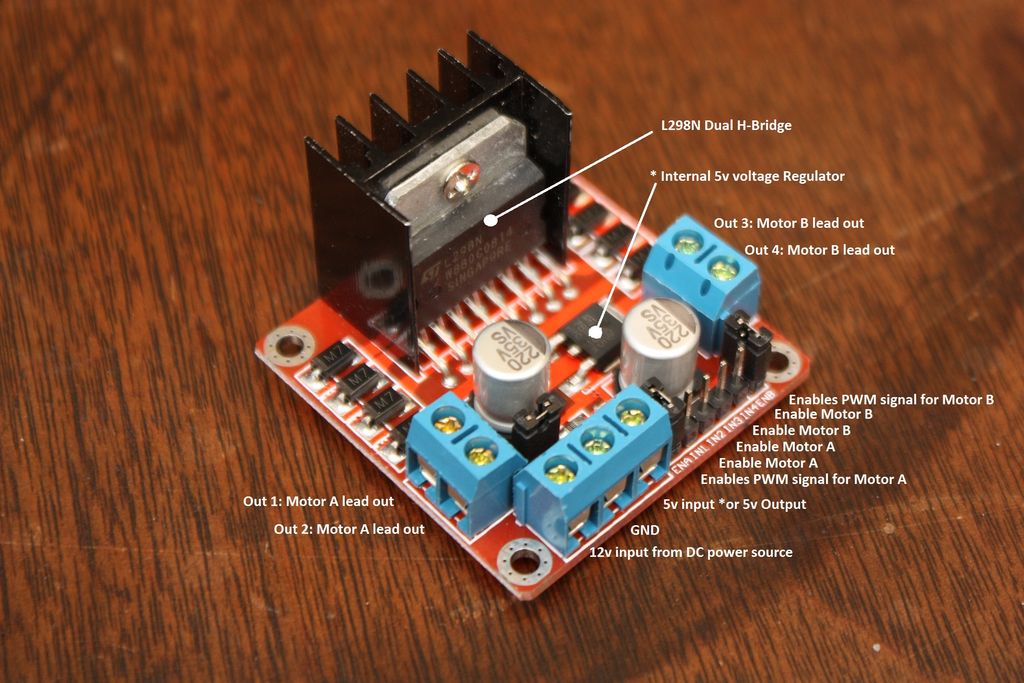
**Motor Driver Operation**

**Module pinouts**

Consider the following image - match the numbers against the list below the image:

[](https://tronixlabs.com.au/robotics/motor-controllers/l298n-dual-motor-controller-module-2a-australia/)

1. DC motor 1 "+" or stepper motor A+
2. DC motor 1 "-" or stepper motor A-
3. 12V jumper - remove this if using a supply voltage greater than 12V DC. This enables power to the onboard 5V regulator
4. Connect your motor supply voltage here, maximum of 35V DC. Remove 12V jumper if >12V DC
5. GND
6. 5V output if 12V jumper in place, ideal for powering your Arduino (etc)
7. DC motor 1 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
8. IN1
9. IN2
10. IN3
11. IN4
12. DC motor 2 enable jumper. Leave this in place when using a stepper motor. Connect to PWM output for DC motor speed control.
13. DC motor 2 "+" or stepper motor B+
14. DC motor 2 "-" or stepper motor B-

[](https://cdn.instructables.com/FRJ/7W76/HZDYE1YD/FRJ7W76HZDYE1YD.LARGE.jpg)

To control one or two DC motors is quite easy with the [L298N H-bridge module](http://tronixlabs.com/robotics/motor-controllers/l298n-dual-motor-controller-module-2a/). First connect each motor to the A and B connections on the L298N module.

If you’re using two motors for a robot (etc) ensure that the polarity of the motors is the same on both inputs. Otherwise you may need to swap them over when you set both motors to forward and one goes backwards!

Next, connect your power supply – the positive to pin 4 on the module and negative/GND to pin 5. If you supply is up to 12V you can leave in the 12V jumper (point 3 in the image above) and 5V will be available from pin 6 on the module.

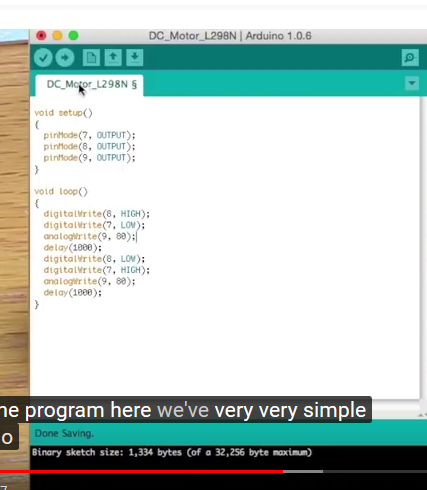
This can be fed to your Arduino’s 5V pin to power it from the motors’ power supply. Don’t forget to connect Arduino GND to pin 5 on the module as well to complete the circuit. Now you will need six digital output pins on your Arduino, two of which need to be PWM (pulse-width modulation) pins.

PWM pins are denoted by the tilde (“~”) next to the pin number, for example in the image of the Arduino Uno's digital pins.

Finally, connect the Arduino digital output pins to the driver module. In our example we have two DC motors, so digital pins D9, D8, D7 and D6 will be connected to pins IN1, IN2, IN3 and IN4 respectively. Then connect D10 to module pin 7 (remove the jumper first) and D5 to module pin 12 (again, remove the jumper).

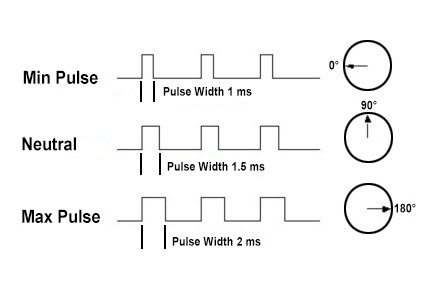
The motor direction is controlled by sending a HIGH or LOW signal to the drive for each motor (or channel). For example for motor one, a HIGH to IN1 and a LOW to IN2 will cause it to turn in one direction, and a LOW and HIGH will cause it to turn in the other direction.

However the motors will not turn until a HIGH is set to the enable pin (7 for motor one, 12 for motor two). And they can be turned off with a LOW to the same pin(s). However if you need to control the speed of the motors, the PWM signal from the digital pin connected to the enable pin can take care of it.



SERVO MOTOR

Servos are controlled by sending an electrical pulse of variable width, or pulse width modulation (PWM), through the control wire. There is a minimum pulse, a maximum pulse, and a repetition rate. A servo motor can usually only turn 90° in either direction for a total of 180° movement. The motor's neutral position is defined as the position where the servo has the same amount of potential rotation in the both the clockwise or counter-clockwise direction. The PWM sent to the [motor](https://www.jameco.com/webapp/wcs/stores/servlet/JamecoSearch?langId=-1&storeId=10001&catalogId=10001&freeText=motor&search_type=jamecoall) determines position of the shaft, and based on the duration of the pulse sent via the control wire; the [rotor](https://www.jameco.com/webapp/wcs/stores/servlet/JamecoSearch?langId=-1&storeId=10001&catalogId=10001&categoryName=cat_3540&subCategoryName=Electromechanical%20%2F%20Switches%20%2F%20Rotary&category=354055&refine=1&position=1&history=kv7hqebe%7CfreeText~rotor%5Esearch_type~jamecoall%5EprodPage~50%5Epage~SEARCH%252BNAV%405hha4bcd%7Ccategory~35%5EcategoryName~category_root%5Eposition~1%5Erefine~1%5EsubCategoryName~Electromechanical%5EprodPage~50%5Epage~SEARCH%252BNAV) will turn to the desired position. The servo motor expects to see a pulse every 20 milliseconds (ms) and the length of the pulse will determine how far the motor turns. For example, a 1.5ms pulse will make the motor turn to the 90° position. Shorter than 1.5ms moves it in the counter clockwise direction toward the 0° position, and any longer than 1.5ms will turn the servo in a clockwise direction toward the 180° position.

*Variable Pulse width control servo position*

When these servos are commanded to move, they will move to the position and hold that position. If an external force pushes against the servo while the servo is holding a position, the servo will resist from moving out of that position. The maximum amount of force the servo can exert is called the torque rating of the servo. Servos will not hold their position forever though; the position pulse must be repeated to instruct the servo to stay in position.

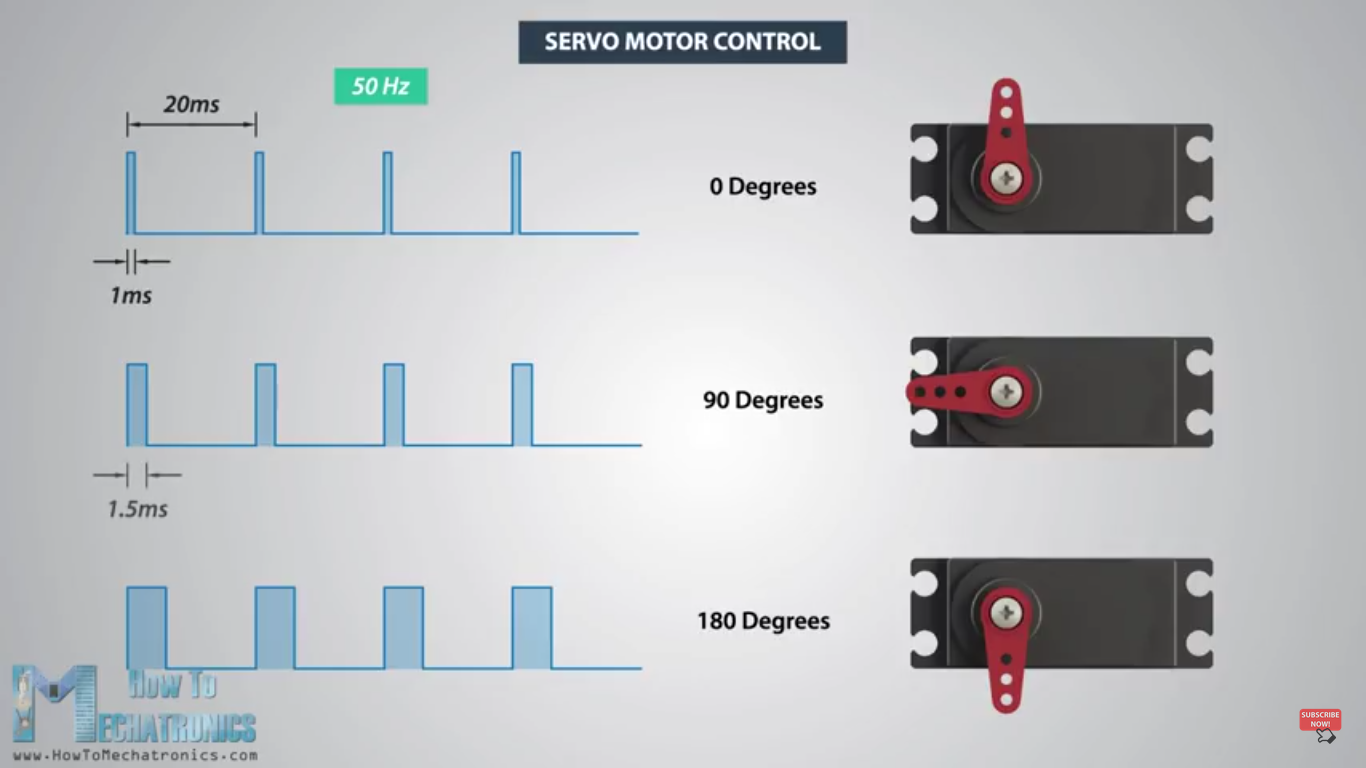
Servo motor works on **PWM (Pulse width modulation)** principle, means its angle of rotation is controlled by the duration of applied pulse to its Control PIN. Basically servo motor is made up of **DC motor which is controlled by a variable resistor (potentiometer) and some gears**. High speed force of DC motor is converted into torque by Gears. We know that WORK= FORCE X DISTANCE, in DC motor Force is less and distance (speed) is high and in Servo, force is High and distance is less. Potentiometer is connected to the output shaft of the Servo, to calculate the angle and stop the DC motor on required angle.

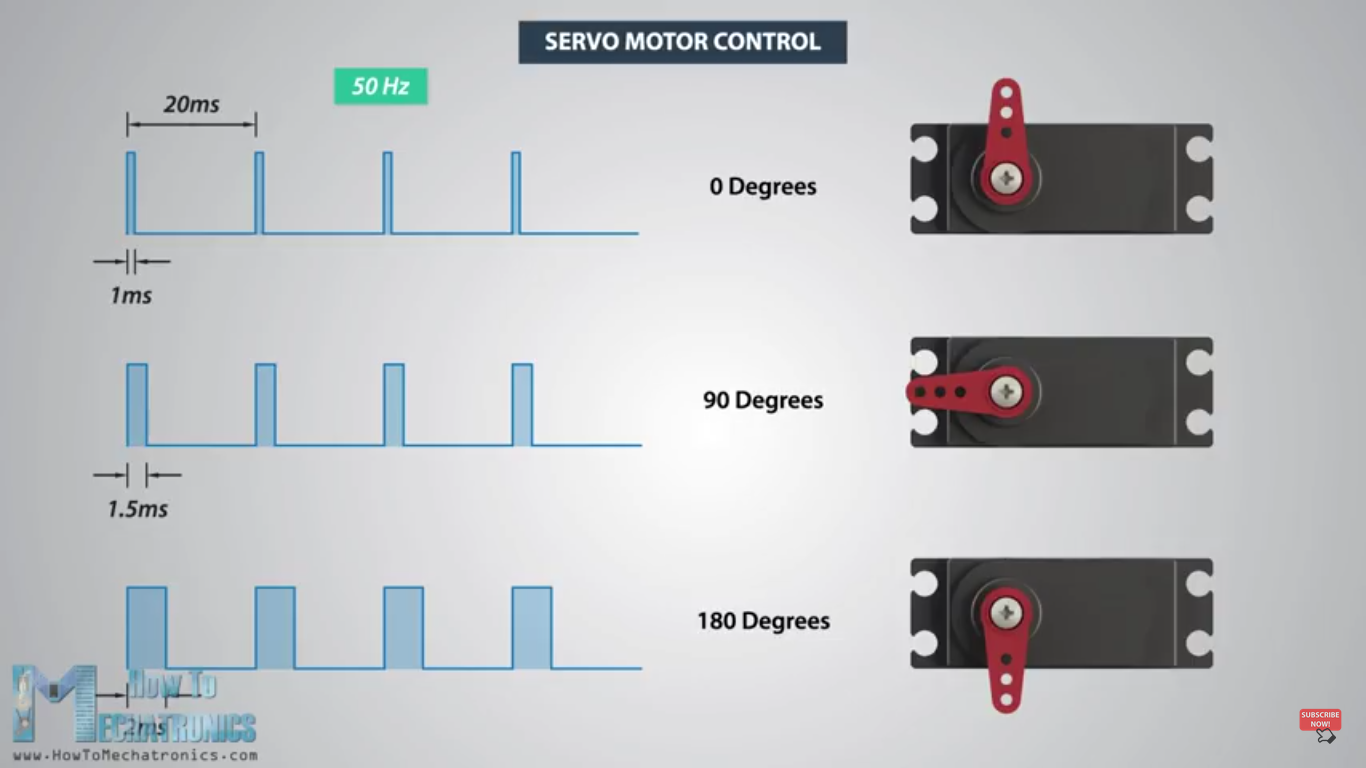
There are two types of servo motors - AC and DC. AC servo can handle higher current surges and tend to be used in industrial machinery. [DC servos](https://www.jameco.com/webapp/wcs/stores/servlet/JamecoSearch?langId=-1&storeId=10001&catalogId=10001&categoryName=cat_75&subCategoryName=Robotics%20%2F%20Motors&category=7545&refine=1&position=1&history=ko9l7dni%7CfreeText~servo%2Bmotor%5Esearch_type~jamecoall%5EprodPage~50%5Epage~SEARCH%252BNAV%40gu9oa7zv%7Ccategory~75%5EcategoryName~category_root%5Eposition~1%5Erefine~1%5EsubCategoryName~Robotics%5EprodPage~50%5Epage~SEARCH%252BNAV) are not designed for high current surges and are usually

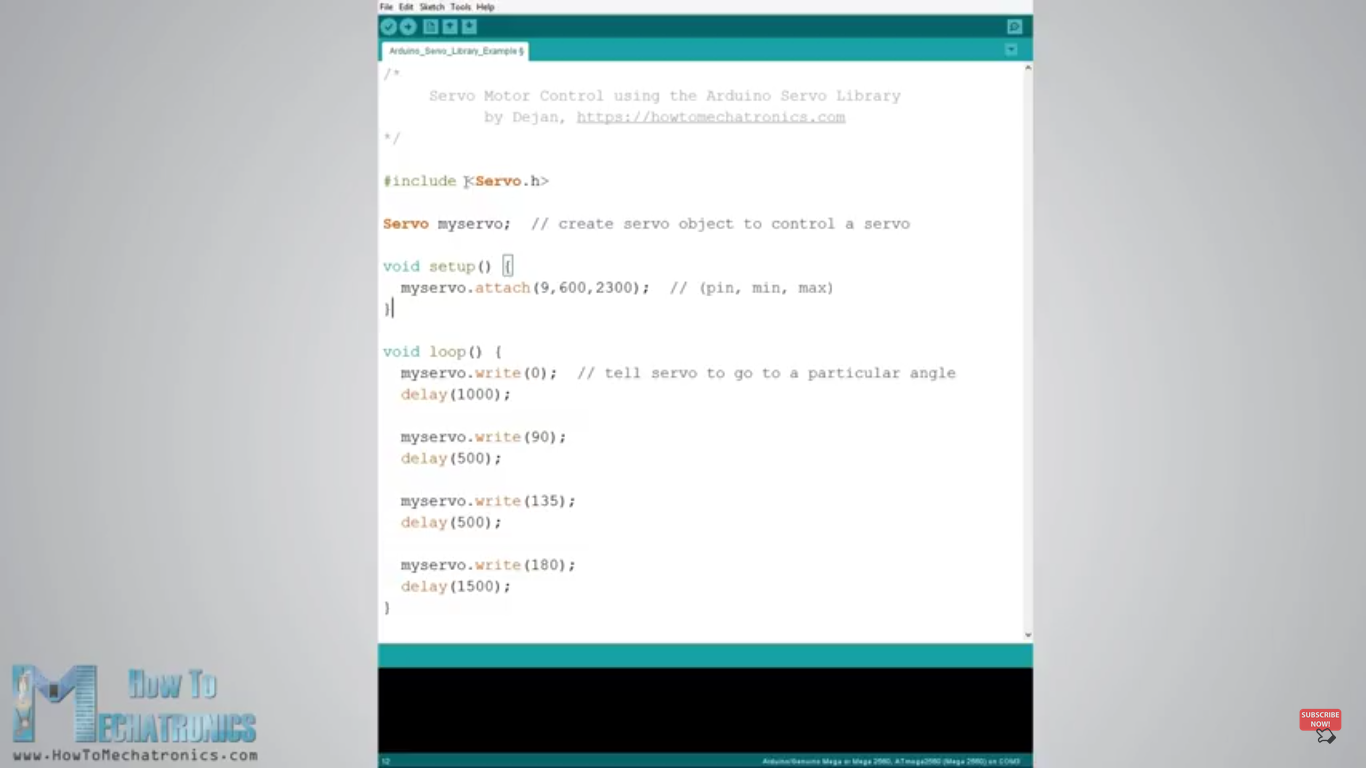
better suited for smaller applications. Generally speaking, DC motors are less expensive than their AC counterparts. These are also servo motors that have been built specifically for continuous rotation, making it an easy way to get your robot moving. They feature two ball bearings on the output shaft for reduced friction and easy access to the rest-point adjustment [potentiometer](https://www.jameco.com/webapp/wcs/stores/servlet/JamecoSearch?langId=-1&storeId=10001&catalogId=10001&categoryName=category_root&subCategoryName=Passive%20Components&category=20&refine=1&position=1&history=d0ww5ora%7CfreeText~potentiometer%5Esearch_type~jamecoall%5EprodPage~50%5Epage~SEARCH%252BNAV).

Most servo motors have the following three connections:

* Black/Brown ground wire.
* Red power wire (around 5V).
* Yellow or White PWM wire.







Note:-

Syntax

map(value, fromLow, fromHigh, toLow, toHigh)

Parameters

value: the number to map

fromLow: the lower bound of the value’s current range

fromHigh: the upper bound of the value’s current range

toLow: the lower bound of the value’s target range

toHigh: the upper bound of the value’s target range

STEPPER MOTOR

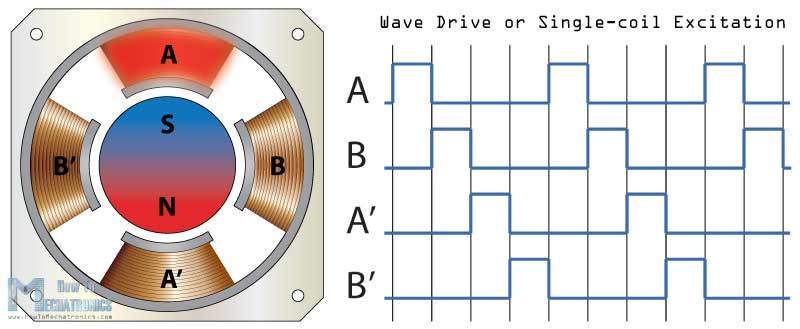
## Working Principle

Stepper motor is a brushless DC motor that rotates in steps. This is very useful because it can be precisely positioned without any feedback sensor, which represents an open-loop controller. The stepper motor consists of a rotor that is generally a permanent magnet and it is surrounded by the windings of the stator. As we activate the windings step by step in a particular order and let a current flow through them they will magnetize the stator and make electromagnetic poles respectively that will cause propulsion to the motor. So that’ the basic working principle of the stepper motors.

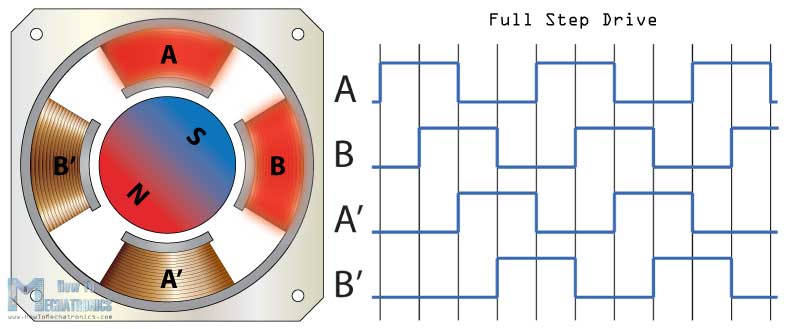
### Working-Principle of Stepper Motor

## Driving Modes

There are several different ways of driving the stepper motor. The first one is the Wave Drive or Single-Coil Excitation. In this mode we active just one coil at a time which means that for this example of motor with 4 coils, the rotor will make full cycle in 4 steps.

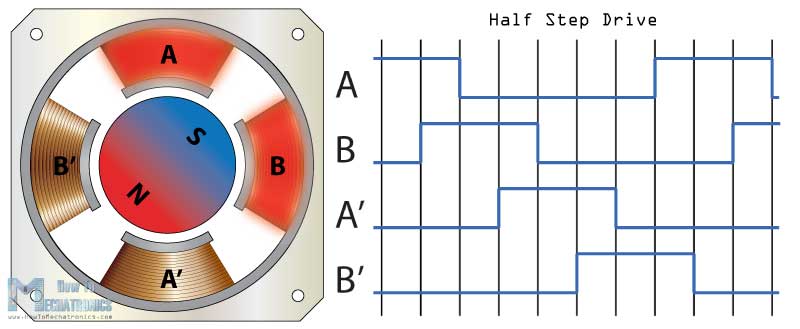


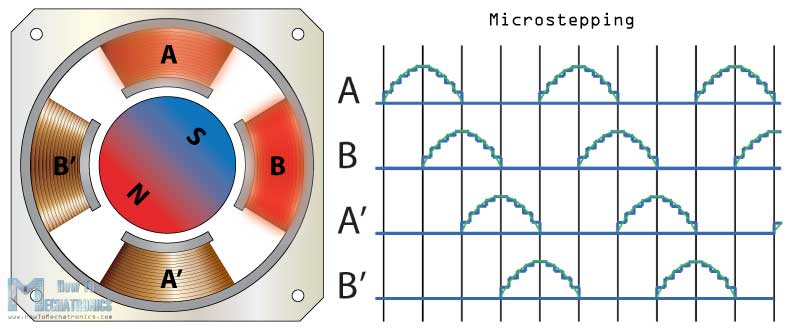
Next is the Full step drive mode which provides much higher torque output because we always have 2 active coils at a given time. However this doesn’t improve the resolution of the stepper and again the rotor will make a full cycle in 4 steps.



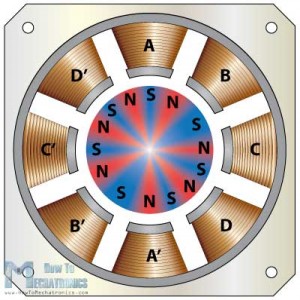
For increasing the resolution of the stepper we use the Half Step Drive mode. This mode is actually a combination of the previous two modes.

Here we have one active coil followed by 2 active coils and then again one active coil followed by 2 active coils and so on. So with this mode we get double the resolution with the same construction. Now the rotor will make full cycle in 8 steps.

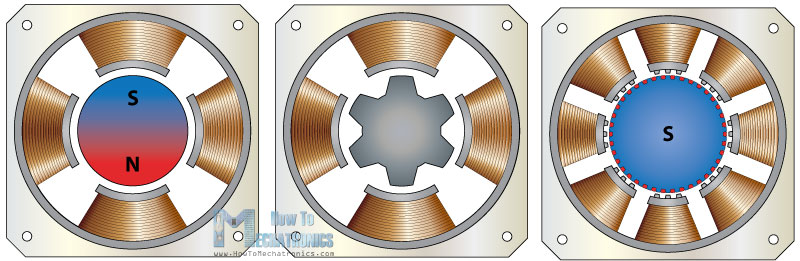
However the most common method of [ontrolling stepper motors](https://howtomechatronics.com/tutorials/arduino/how-to-control-stepper-motor-with-a4988-driver-and-arduino/) nowadays is the Microstepping. In this mode we provide variable controlled current to the coils in form of sin wave. This will provide smooth motion of the rotor, decrease the stress of the parts and increase the accuracy of the stepper motor.



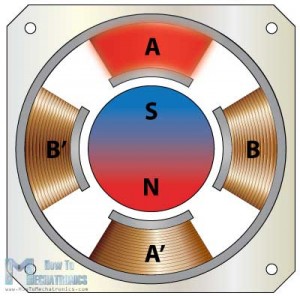
Another way of increasing the resolution of the stepper motor is by increasing the numbers of the poles of the rotor and the numbers of the pole of the stator.



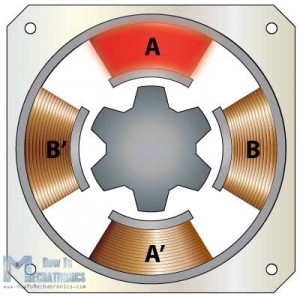
## Stepper Motor Types by Construction

By construction there are 3 different types of stepper motors: permanent magnet stepper, variable reluctance stepper and hybrid synchronous stepper motor.  


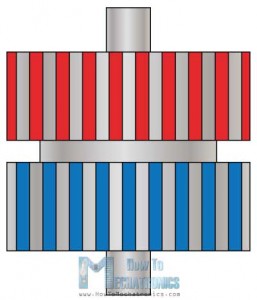
The **Permanent Magnet** stepper has a permanent magnet rotor which is driven by the stators windings. They create opposite polarity poles compared to the poles of the rotor which propels the rotor.



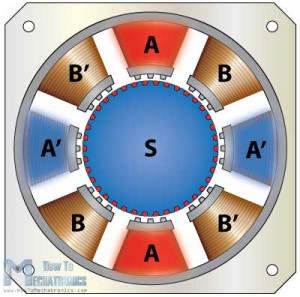
The next type, the **Variable Reluctant** stepper motor uses a non-magnetizes soft iron rotor. The rotor has teeth that are offset from the stator and as we active the windings in a particular order the rotor moves respectively so that it has minimum gab between the stator and the teeth of the rotor



The **Hybrid Synchronous** motor is combinations of the previous two steppers. It has permanent magnet toothed rotor and also a toothed stator. The rotor has two sections, which are opposite in polarity and their teeth are offset as shown here.



This is a front view of a commonly used hybrid stepper motor which has 8 poles on the stator that are activated by 2 windings, A and B. So if we activate the winding A, we will magnetize 4 poles of which two of them will have South polarity and two of them North polarity.



We can see that in such a way the rotors teeth are aligned with the teeth of poles A and unaligned with the teeth of the poles B. That means that in the next step when we turn off the A poles and activate the B poles, the rotor will move counter clock wise and its teeth will align with the teeth of the B poles.

If we keep activating the poles in a particular order the rotor will move continuously. Here we can also use different driving modes like the wave drive, full step drive, half step drive and microstepping for even further increasing the resolution of the stepper motor.

# Types of Steppers

There are a wide variety of stepper types, some of which require very specialized drivers. For our purposes, we will focus on stepper motors that can be driven with commonly available drivers. These are: Permanent Magnet or Hybrid steppers, either 2-phase bipolar, or 4-phase unipolar.

# Motor Size

One of the first things to consider is the work that the motor has to do. As you might expect, larger motors are capable of delivering more power. Stepper motors come in sizes ranging from smaller than a peanut to big NEMA 57 monsters.   
  
Most motors have torque ratings. This is what you need to look at to decide if the motor has the strength to do what you want.  
  
NEMA 17 is a common size used in 3D printers and smaller CNC mills. Smaller motors find applications in many robotic and animatronic applications. The larger NEMA frames are common in CNC machines and industrial applications.  
  
The NEMA numbers define standard faceplate dimensions for mounting the motor. They do not define the other characteristics of a motor. Two different NEMA 17 motors may have entirely different electrical or mechanical specifications and are not necessarily interchangeable.

# Step Count

The next thing to consider is the positioning resolution you require. The number of steps per revolution ranges from 4 to 400. Commonly available step counts are 24, 48 and 200.  
  
Resolution is often expressed as degrees per step. A 1.8° motor is the same as a 200 step/revolution motor.  
  
The trade-off for high resolution is speed and torque. High step count motors top-out at lower RPMs than similar size. And the higher step-rates needed to turn these motors results in lower torque than a similar size low-step-count motor at similar speeds.

# Gearing

Another way to achieve high positioning resolution is with gearing. A 32:1 gear-train applied to the output of an 8-steps/revolution motor will result in a 512 step motor.  
  
A gear train will also increase the torque of the motor. Some tiny geared steppers are capable of impressive torque. But the tradeoff of course is speed. Geared stepper motors are generally limited to low RPM applications.

# Shaft Style

Another thing to consider is how the motor will interface with the rest of the drive system. Motors are available with a number of shaft styles:

* **Round or "D" Shaft**: These are available in a variety of standard diameters and there are many pulleys, gears and shaft couplers designed to fit. "D" shafts have one flattened side to help prevent slippage. These are desirable when high torques are involved.
* **Geared shaft**: Some shafts have gear teeth milled right into them. These are typically designed to mate with modular gear trains.
* **Lead-Screw Shaft**: Motors with lead-screw shafts are used to build linear actuators. Miniature versions of these can be found as head positioners in many disk drives.

Arduino code

/\* Stepper Motor Control \*/

#include <Stepper.h>

const int stepsPerRevolution = 90;

// 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);

void setup() {

// set the speed at 60 rpm:

myStepper.setSpeed(5);

// initialize the serial port:

Serial.begin(9600);

}

void loop() {

// step one revolution in one direction:

Serial.println("clockwise");

myStepper.step(stepsPerRevolution);

delay(500);

// step one revolution in the other direction:

Serial.println("counterclockwise");

myStepper.step(-stepsPerRevolution);

delay(500);

}

