thingsRoam
academy

ARDUINO COURSE GUIDE

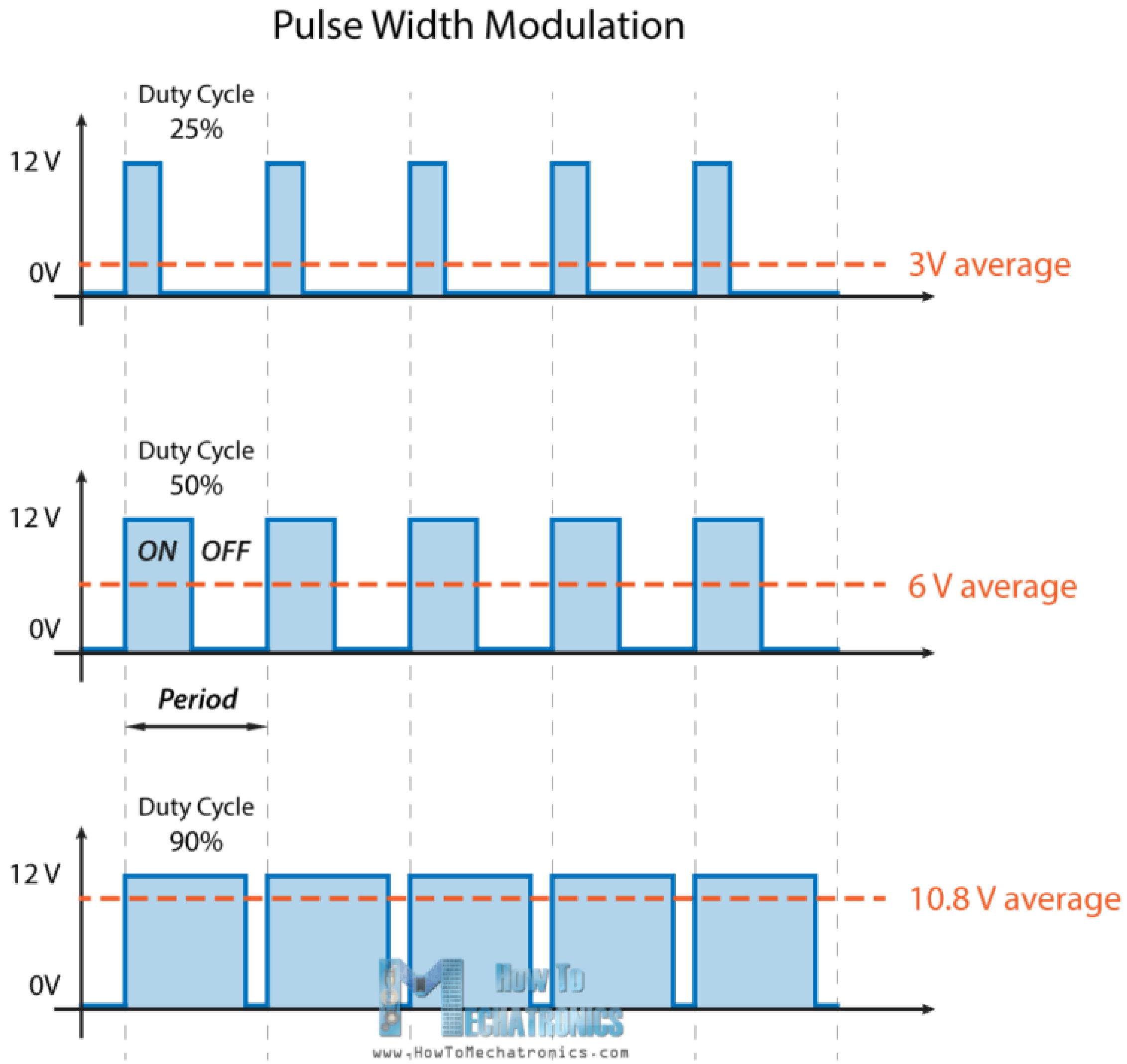
ARDUINO
OPEN-SOURCE
COMMUNITY

What are Motors?

An electric motor is an electrical machine that converts electrical energy into mechanical energy. Electric motors can be powered by direct current (DC) sources, such as from batteries, motor vehicles or rectifiers, or by alternating current (AC) sources, such as a power grid, inverters or electrical generators.

DC Motor Speed Control with PWM:

PWM, or pulse width modulation is a technique which allows us to adjust the average value of the voltage that's going to the electronic device by turning on and off the power at a fast rate. The average voltage depends on the duty cycle, or the amount of time the signal is ON versus the amount of time the signal is OFF in a single period of time.



1. DC MOTORS:

A **DC motor** is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy. The most common types rely on the forces produced by magnetic fields.

DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate. If you switch the leads, the motor will rotate in the opposite direction.

1. Interfacing DC Motor and controlling its speed with Arduino

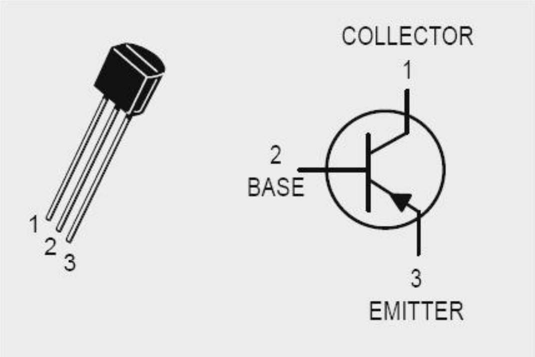
Components:

- Arduino Uno
- USB cable
- Breadboard
- DC Motor
- PN2222 Transistor
- 1N4001 diode
- 270 Ω Resistor (Color Code: Red, Violet, Brown Gold)
- Jumper wires

  Operating Voltage: 5 volts

Transistor:

The small DC motor, is likely to use more power than an Arduino digital output can handle directly. If we tried to connect the motor straight to an Arduino pin, there is a good chance that it could damage the Arduino.
A small transistor like the PN2222 can be used as a switch that uses just a little current from the Arduino digital output to control the much bigger current of the motor.



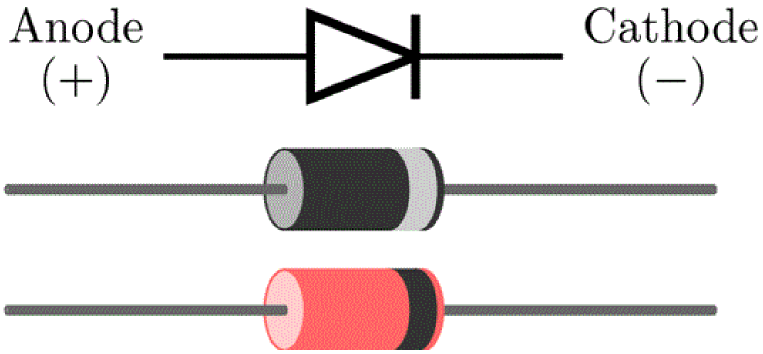
PIN CONFIGRATION:

1-**Collector** → Diode Positive Pin and Motor Wire
2-**Base** → One leg of Resistor (2nd Leg of Resistor connected to pin 3)
3-**Emitter** → Connected to GND

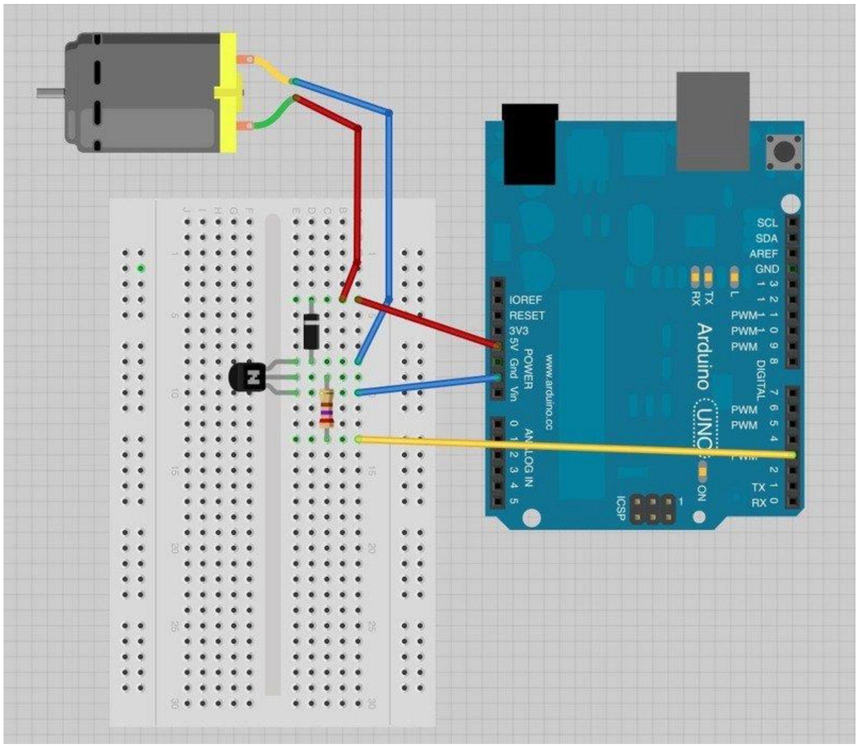
Diode:

Diode is a semiconductor device with two terminals, typically allowing the flow of current in one direction only.

There is a diode connected across the connections of the motor. Diodes only allow electricity to flow in one direction (the direction of their arrow).



Circuit Diagram:



Code:

```
int motorPin = 3;

void setup()
{
  pinMode(motorPin, OUTPUT);
  Serial.begin(9600);
  while (! Serial);
  Serial.println("Speed 0 to 255");
}


void loop()
{
  if (Serial.available())
  {
    int speed = Serial.parseInt();
    if (speed >= 0 && speed <= 255)
    {
      analogWrite(motorPin, speed);
    }
  }
}
```

parseInt():

Looks for the next valid integer in the incoming serial stream. parseInt() inherits from the Stream utility class.

Syntax: *Serial.parseInt()*

In the 'loop' function, the command 'Serial.parseInt' is used to read the number entered as text in the Serial Monitor and convert it into an 'int'.

You could type any number here, so the 'if' statement on the next line only does an analog write with this number if the number is between 0 and 255.

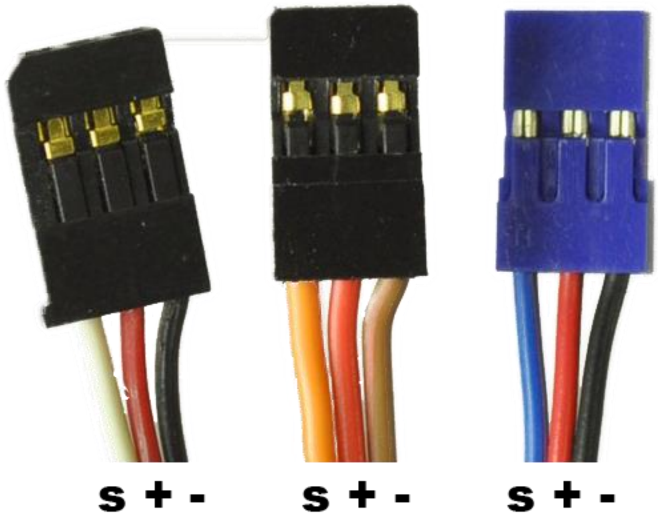
2. SERVO MOTORS:

A **servo motor** is an electrical device which can push or rotate an object with great precision. If you want to rotate and object at some specific angles or distance, then you use servo motor. It is just made up of simple motor which run through **servo mechanism**. If motor is used is DC powered then it is called DC servo motor, and if it is AC powered motor then it is called AC servo motor. We can get a very high torque servo motor in a small and light weight packages. Doe to these features they are being used in many applications like toy car, RC helicopters and planes, Robotics, Machine etc.

Servos have integrated gears and a shaft that can be precisely controlled. Standard servos allow the shaft to be positioned at various angles, usually between 0 and 180 degrees. You can also buy 'continuous' servos that can rotate through the full 360 degrees.

Servo Wiring:

The servo motor has three leads. The color of the leads varies between servo motors, but the **red lead is always 5V** and **GND will either be black or brown**. The other lead is the **control lead** and this is usually **orange or yellow**.



Servo Connections:
Red = Arduino 5V
Black/Brown = Arduino GND
Yellow/Orange = Arduino (pin9)

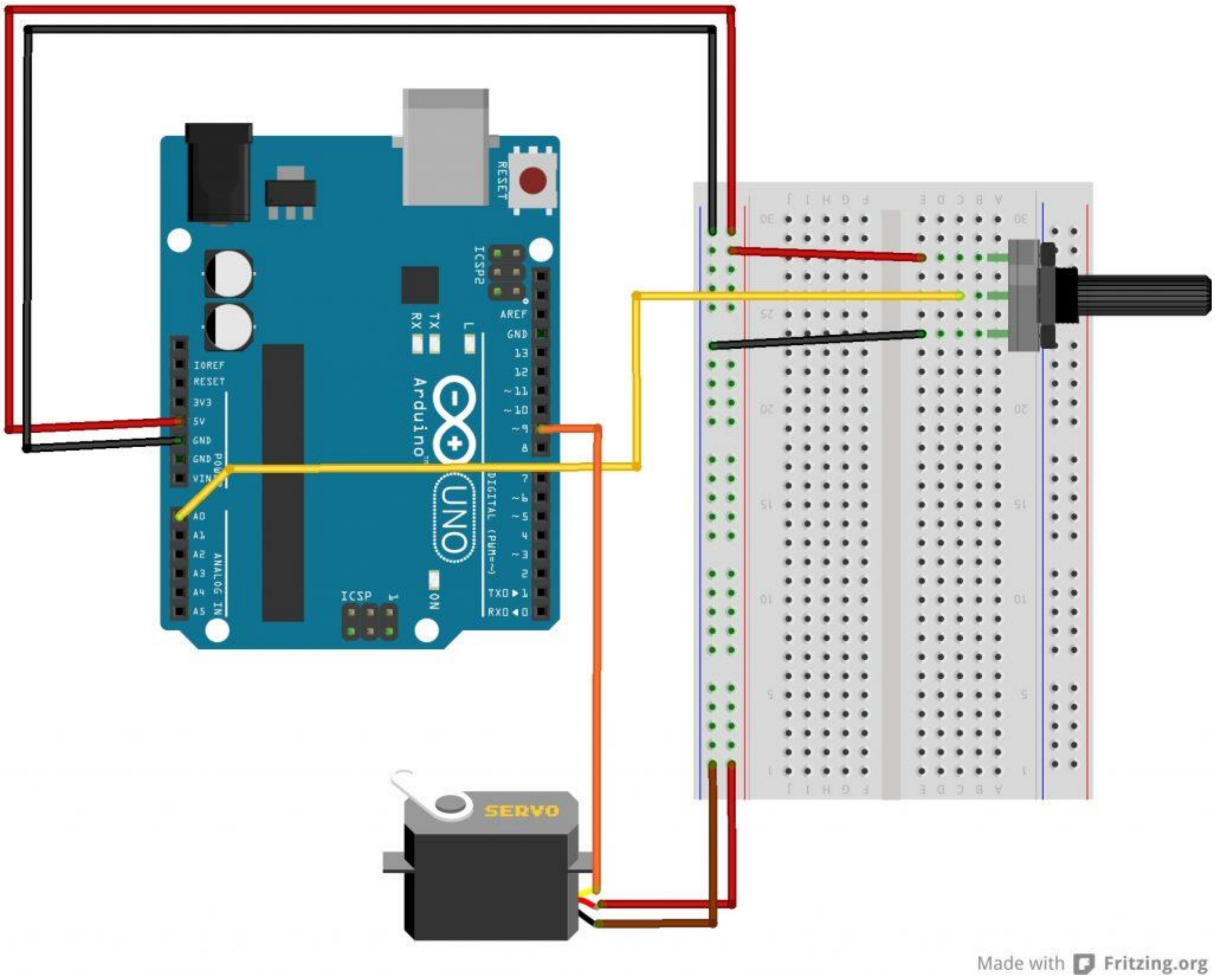
1. Controlling Servo Motor with POT and Arduino (KNOB)

Components:

- Arduino Uno
- USB cable
- Breadboard
- Servo Motor
- Jumper wires

Operating Voltage: 5 volts

Circuit Diagram:



Code:

```
#include <Servo.h>

unsigned int potPin = 0;
unsigned int servoPin = 9;
Servo servo;

void setup()
{
  servo.attach(servoPin);  // Attach the Servo variable to a pin.
}

void loop()
{
  int reading = analogRead(potPin);     // 0 to 1023
  int angle = reading / 6;
  servo.write(angle);  // Writes a value to the servo, controlling the shaft accordingly.(in degrees)
}
```

To set the position of the servo, we take an analog reading from A0. This gives us a value of between 0 and 1023. Since the servo can only rotate through 180 degrees, we need to scale this down. Dividing it by six will give us an angle between 0 and 170, which will do just fine.

Running the code from Arduino Knob Example:

map():

Re-maps a number from one range to another. That is, a value of **fromLow** would get mapped to **toLow**, a value of **fromHigh** to **toHigh**, values in-between to values in-between, etc.

Syntax: `map(value, fromLow, fromHigh, toLow, toHigh)`

Parameters

value: the number to map

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

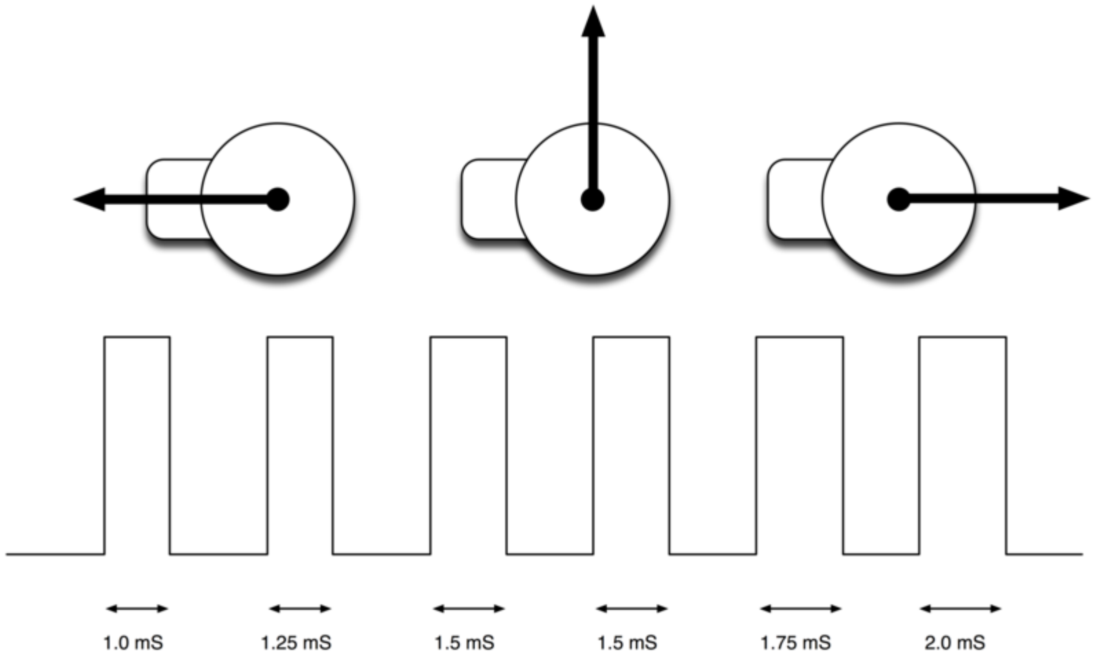
fromHigh: the upper bound of the value’s current range

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

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

HOW SERVO WORKS WITH ARDUINO:

The position of the servo motor is set by the length of a pulse. The servo expects to receive a pulse roughly every 20 milliseconds. If that pulse is high for 1 millisecond, then the servo angle will be zero, if it is 1.5 milliseconds, then it will be at its centre position and if it is 2 milliseconds it will be at 180 degrees.

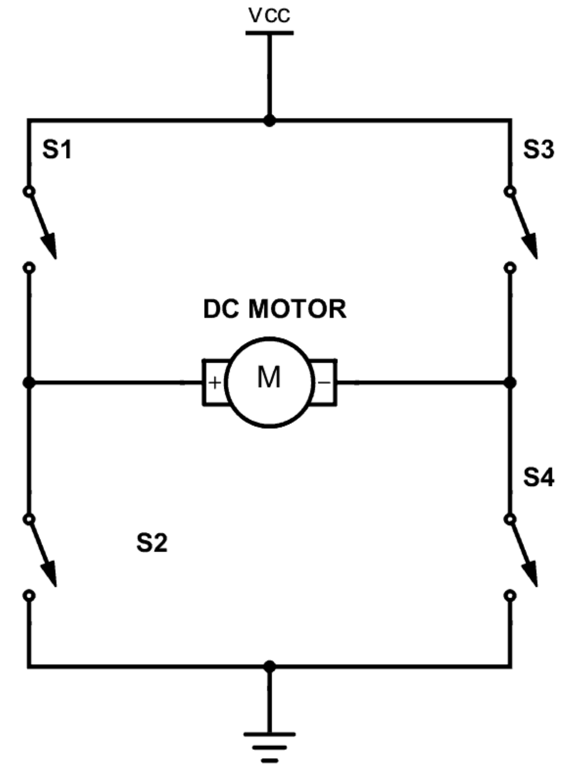


```
lcd.noBlink(); // Turns off the blinking LCD cursor
lcd.cursor(); // Displays an underscore (line) at the position to which the next character will be written
delay(4000);
lcd.noCursor(); // Hides the LCD cursor
lcd.clear(); // Clears the LCD screen
}
```

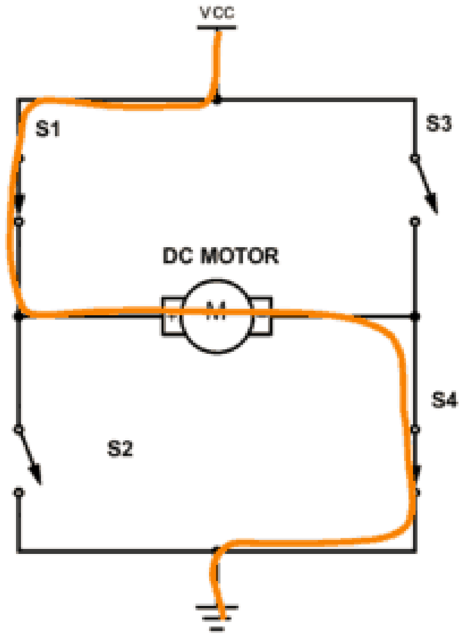
H-BRIDGE (MOTOR DRIVER):

H-Bridge:

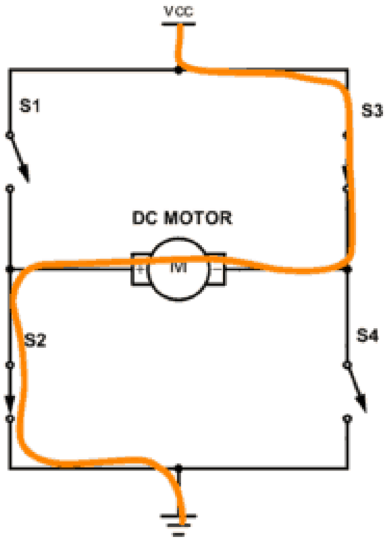
An H-Bridge circuit contains four switching elements, transistors or MOSFETs, with the motor at the center forming an H-like configuration. An H-bridge is a simple circuit that lets you control a DC motor to go backward or forward. By activating two particular switches at the same time we can change the direction of the current flow, thus change the rotation direction of the motor.



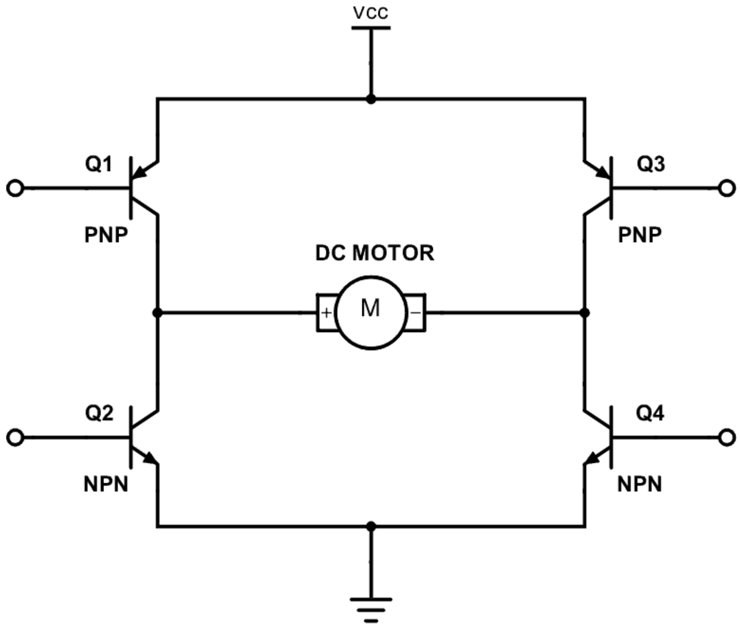
A DC motor spins either backward or forward, depending on how you connect the plus and the minus.
If you close switch 1 and 4, you have plus connected to the left side of the motor and minus to the other side. And the motor will start spinning in one direction.



If you instead close switch 2 and 3, you have plus connected to the right side and minus to the left side. And the motor spins in the opposite direction.



You can build an H-bridge with four transistors. Since the transistor can be a switch, you’ll be able to make the motor spin in either direction by turning on and off the four transistors in the circuit above.Usually, you control the transistors from a microcontroller, such as Arduino.



1. L298N Motor Driver:

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.

HOW IT WORKS:

The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.

Power:

This depends on the voltage used at the motors VCC. The module has an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly.

We can note here that this IC makes a voltage drop of about 2V. So for example, if we use a 12V power supply, the voltage at motors terminals will be about 10V, which means that we won't be able to get the maximum speed out of our 12V DC motor.

Logic Control Inputs:

Next are the logic control inputs. The Enable A and Enable B pins are used for enabling and **controlling the speed** of the motor. If a jumper is present on this pin, the motor will be enabled and work at **maximum speed**, and if we remove the jumper we can connect a **PWM input** to this pin and in that way control the speed of the motor. If we connect this pin to a **Ground** the motor will be **disabled**.

Input Pin Configuration:

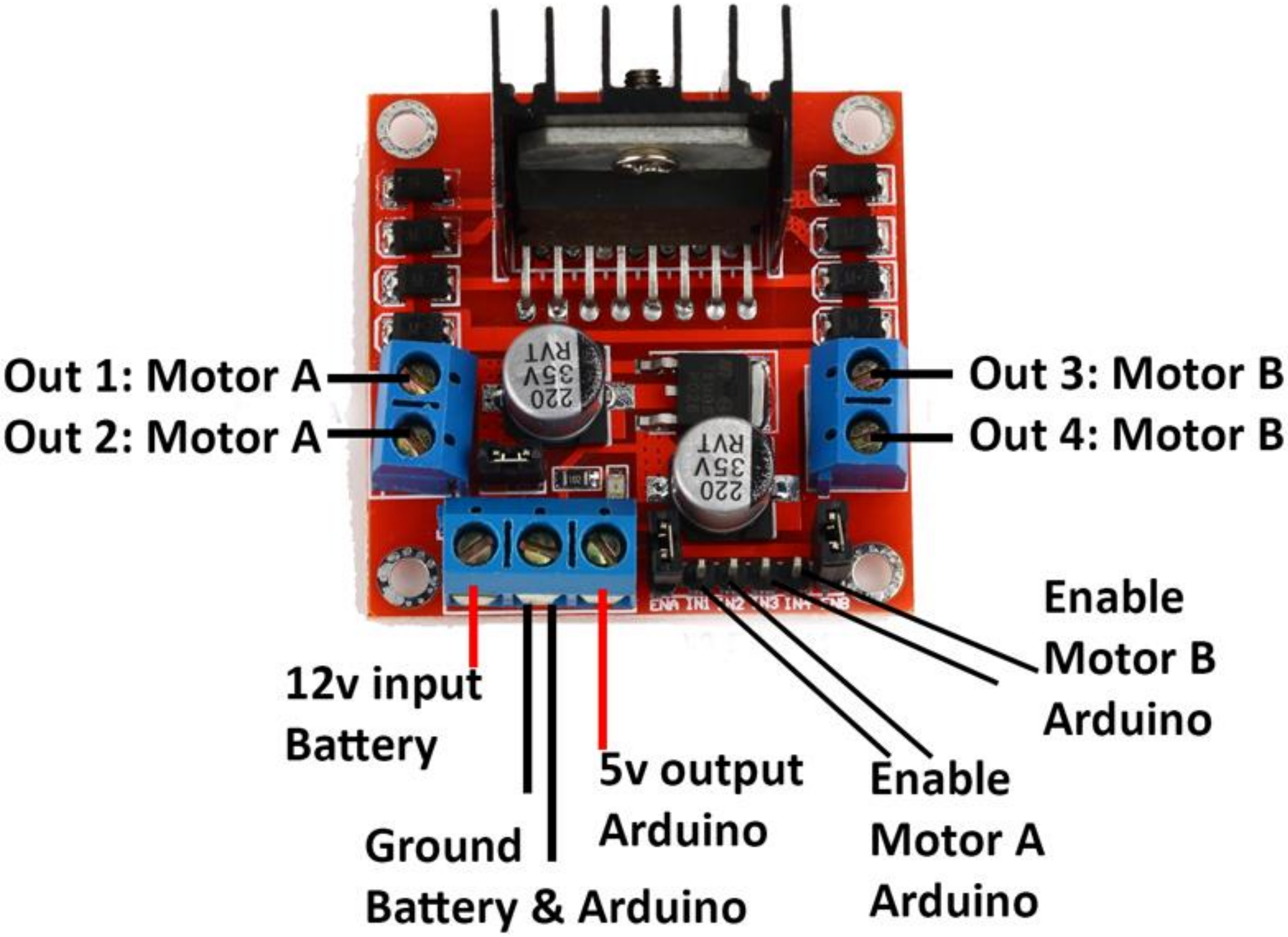
Next, the Input **1** and Input **2** pins are used for controlling the rotation direction of the **motor A**, and the inputs **3** and **4** for the motor **B**. Using these pins, we actually control the switches of the H-Bridge inside the L298N IC.

Input 1 = **LOW**, Input 2 **= HIGH**, Motor moves **Forward**
Input 1 **= HIGH**, Input 2 **= LOW**, Motor moves **Backward**
Input 1 **= HIGH/LOW**, Input 2 **= HIGH/LOW**, Motor **Stops**

In case both inputs are same, either LOW or HIGH the motor will stop. The same applies for the inputs 3 and 4 and the motor B.

L298N Dual H-Bridge Motor Drive

Setup for Arduino and 12v Battery



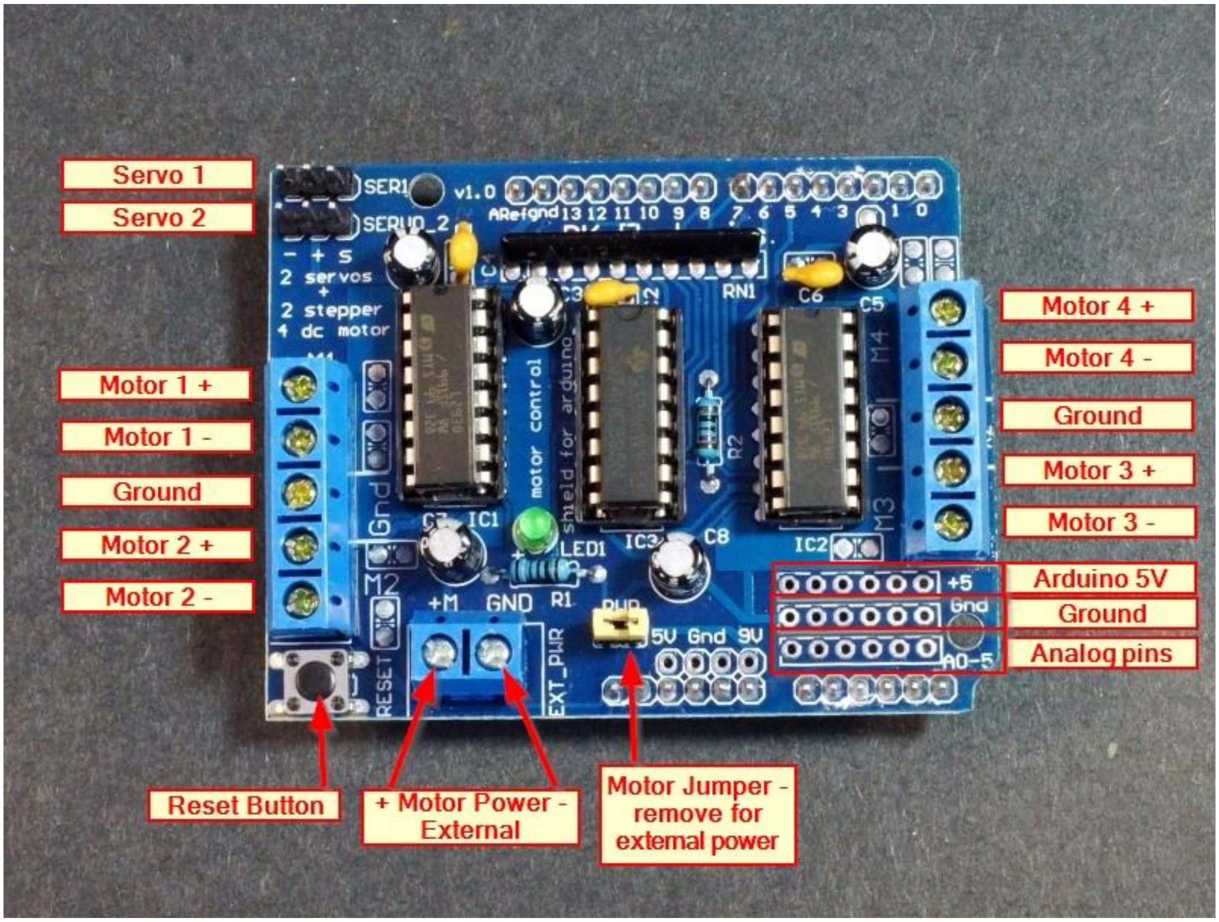
2. L293D Motor Driver Shield:

This motor driver shield is based on L293D motor driver chip which is designed to provide bidirectional drive currents of up to 1.2A each bridge with thermal shutdown protection at voltages from 4.5 V to 36V.

This motor driver expansion board is based on the L293D chip which is designed to drive up to 4 bidirectional DC motors with individual 8-bit speed selection. It can also drive 2 stepper motors (unipolar or bipolar), single coil or double coil, interleaved or micro-stepping. It contains 4 H-bridges which provide up to 0.6 A per bridge (1.2A peak) at voltages from 4.5 V to 36 V.

This motor driver shield is capable of driving:

- Four DC motors and two servos
- Two DC motors, stepper motor, and two-way servo
- Two stepper motors and servos

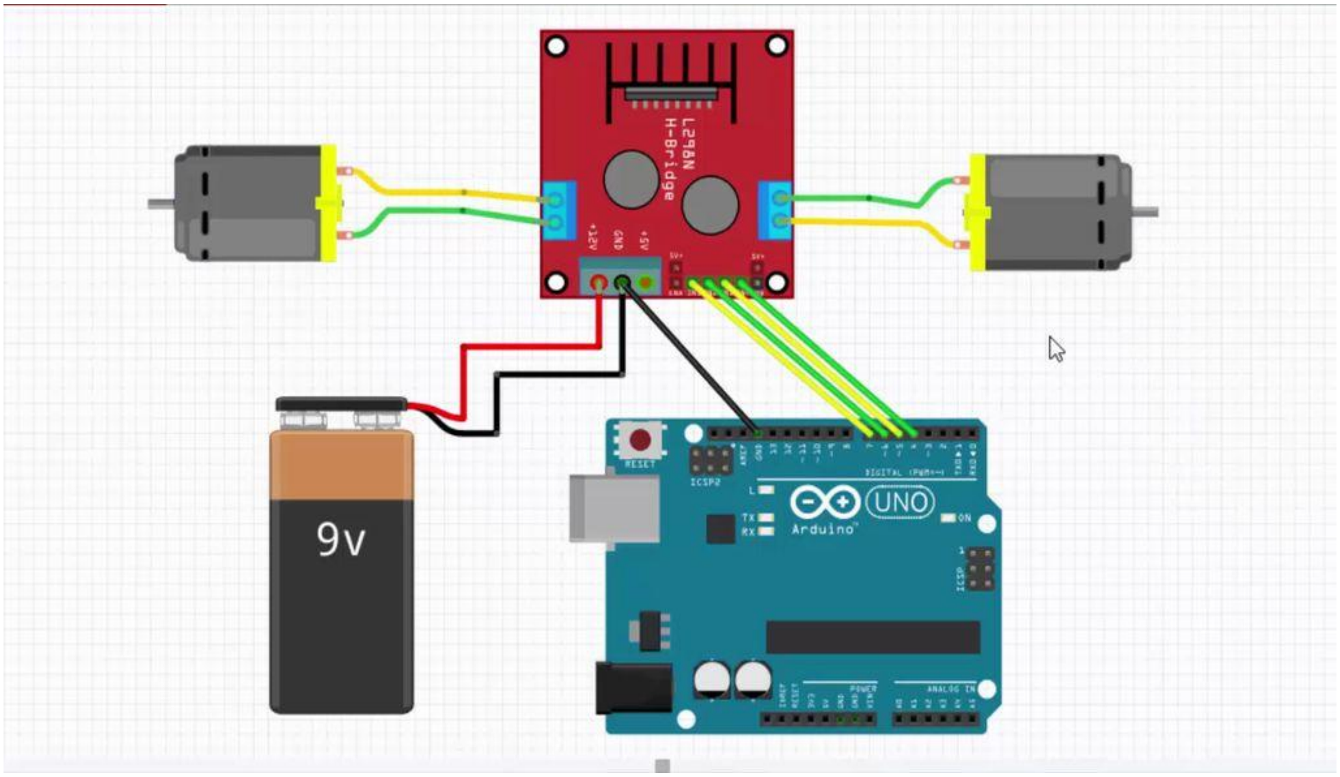


4. DC Motor Control with Motor Driver L298N

Components:

- Arduino Uno
- USB cable
- Breadboard
- L298N Driver Module
- DC Motor
- External Battery 9V/12V
- Jumper wires
  Operating Voltage: 5/9/12 volts

Circuit Diagram:



Coding:

```
int motorpin1= 7 ;
int motorpin2= 8;
int enable1= 6 ;

void fwd();
void bckwd();
void stop();

void setup() {
pinMode(motorpin1, OUTPUT);
pinMode(motorpin2, OUTPUT);
pinMode(enable1, OUTPUT);
}
void loop() {
fwd();
delay(3000);
Serial.print("Motor is running forward");

bckwd();
delay(3000);
Serial.print("Motor is running backward");

stop();
delay(3000);
Serial.print("Motor is stopped");
}

void fwd() {
digitalWrite(motorpin1, LOW);
digitalWrite(motorpin2, HIGH);
analogWrite(enable1, 255);
}
void bckwd()
{
digitalWrite(motorpin1, LOW);
digitalWrite(motorpin2, HIGH);
analogWrite(enable1, 255);
}
void stop()
{
digitalWrite(motorpin1, LOW);
digitalWrite(motorpin1, LOW);
}
```