

# Motors

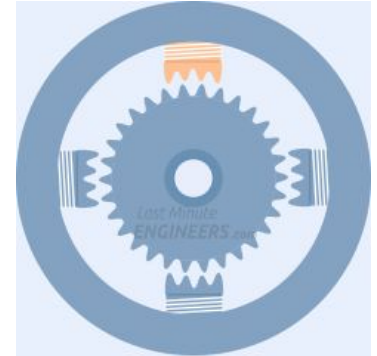
# Stepper Motor

Stepper motors are used in so many everyday items, including window blinds, 3D printers, DVD players, security cameras, and CNC machines.

Stepper motors fall somewhere between a conventional DC motor and a servo motor. They can rotate continuously like DC motors and be positioned precisely (in discrete steps) like servo motors.

Stepper motors use a cogged wheel and electromagnets to nudge the wheel round a 'step' at a time.

Each high pulse sent energizes the coil, attracting the teeth closest to the cogged wheel and rotating the motor in precise and fixed angle increments known as steps.



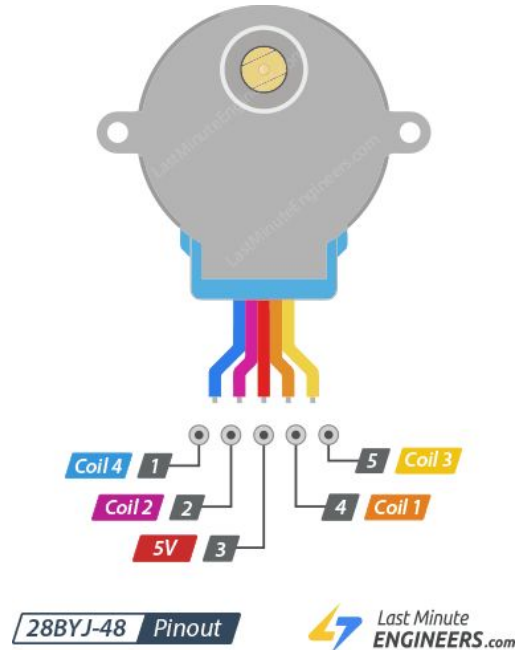
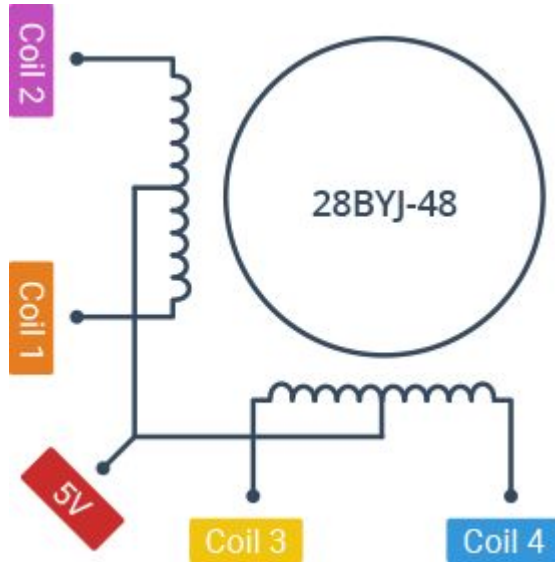
# 28BYJ-48

The 28BYJ-48 is a 5-wire unipolar stepper motor that runs on 5V. It's perfect for projects that require precise positioning, like opening and closing a vent.



# Pins

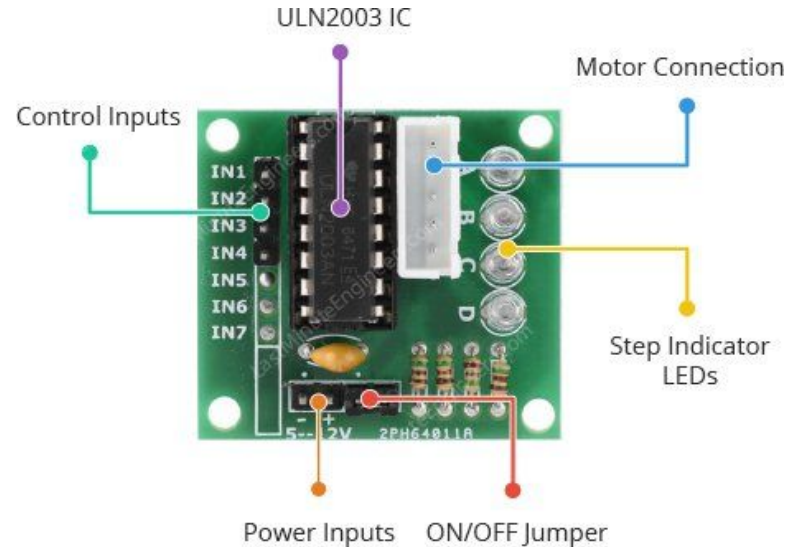
The 28BYJ-48 stepper motor has five wires. The pinout is as follows:



# ULN2003-based driver board

Because the 28BYJ-48 stepper motor consumes a significant amount of power, it cannot be controlled directly by a microcontroller such as Arduino.

To control the motor, a driver IC such as the ULN2003 is required; therefore, this motor typically comes with a ULN2003-based driver board.



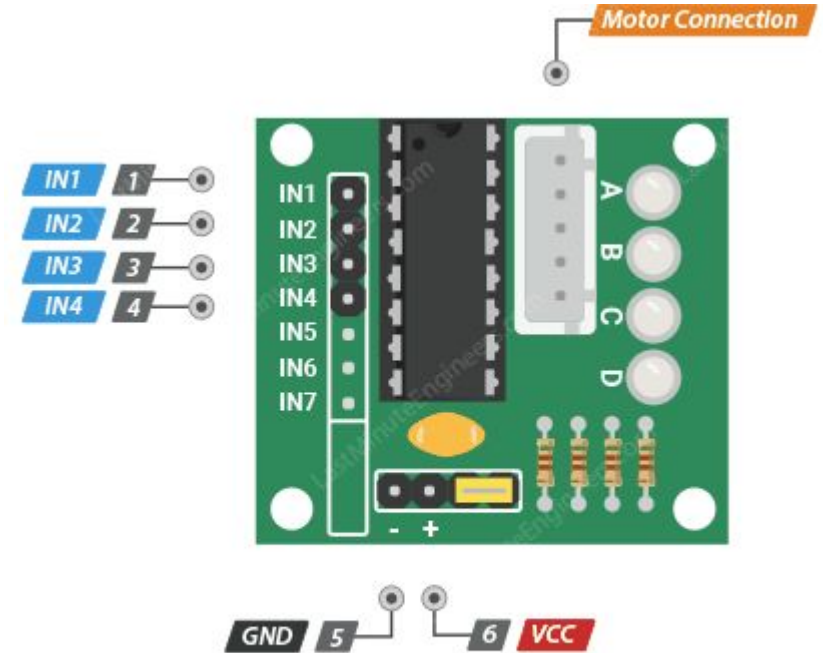
# Driver pin

**IN1 – IN4** are motor control input pins. Connect them to the Arduino's digital output pins.

**GND** is the ground pin.

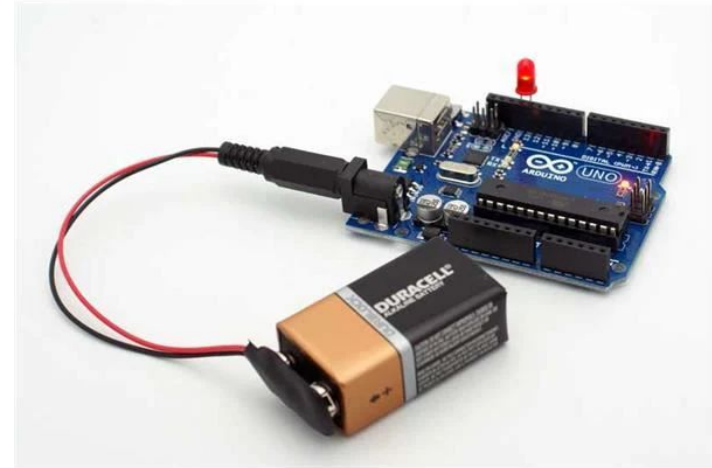
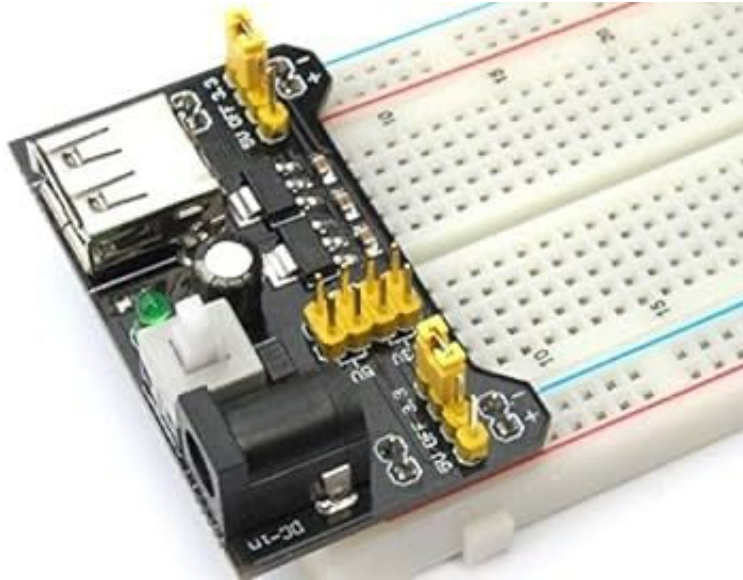
**VCC** pin powers the motor. Because the motor consumes a significant amount of power, it is preferable to use an external 5V power supply rather than from the Arduino.

**Motor Connector** This is where the motor plugs in. The connector is keyed, so it will only go in one way.

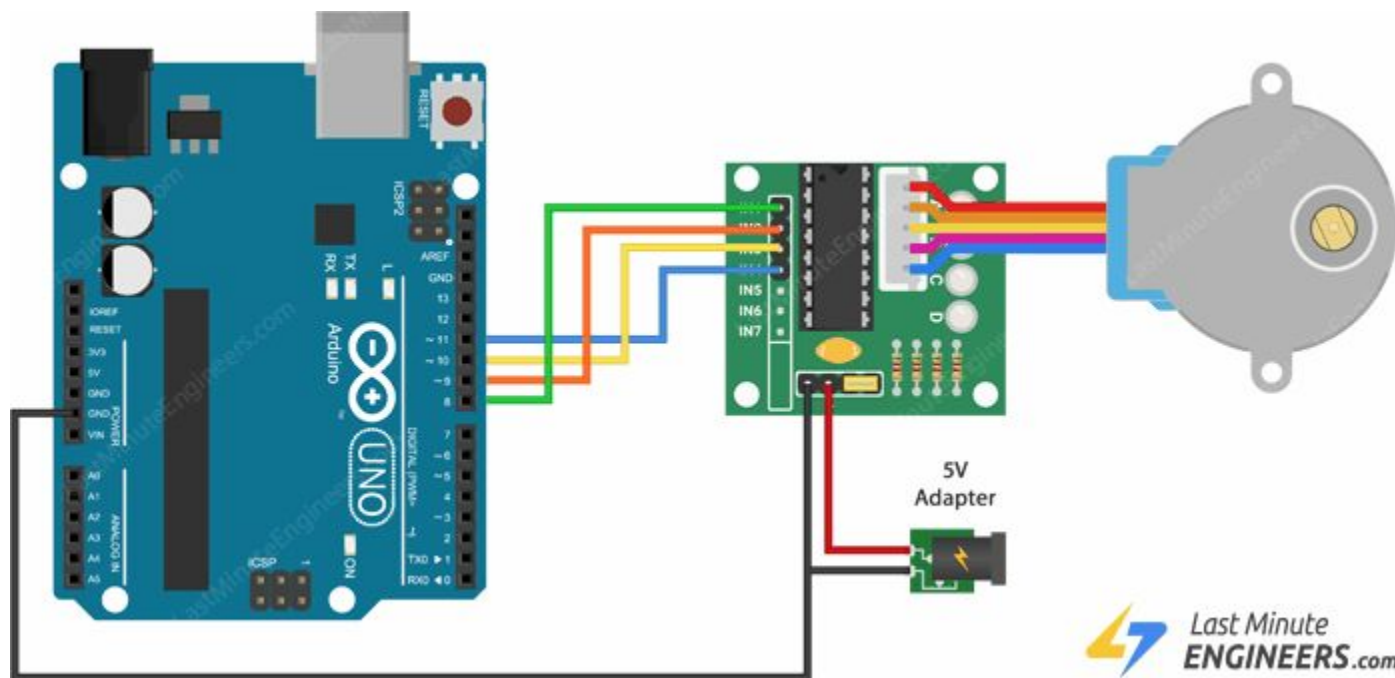


ULN2003 Driver Pinout

Use breadboard power supply module or battery







```
//Includes the Arduino Stepper Library

#include <Stepper.h>

// Defines the number of steps per rotation

const int stepsPerRevolution = 2038;

// Creates an instance of stepper class

// Pins entered in sequence IN1-IN3-IN2-IN4 for proper step sequence

Stepper myStepper = Stepper(stepsPerRevolution, 8, 10, 9, 11);

void setup() {

    // Nothing to do (Stepper Library sets pins as outputs)

}
```

```
void loop() {  
  
    // Rotate CW slowly at 5 RPM  
  
    myStepper.setSpeed(5);  
  
    myStepper.step(stepsPerRevolution);  
  
    delay(1000);  
  
  
    // Rotate CCW quickly at 10 RPM  
  
    myStepper.setSpeed(10);  
  
    myStepper.step(-stepsPerRevolution);  
  
    delay(1000);  
  
}
```

<https://www.electronicoscaldas.com/datasheet/ULN2003A-PCB.pdf>

<https://lastminuteengineers.com/28byj48-stepper-motor-arduino-tutorial/>

# DC Motor

DC motor has two wires:

Positive wire: usually red

Negative wire: usually black



# Controlling a DC Motor

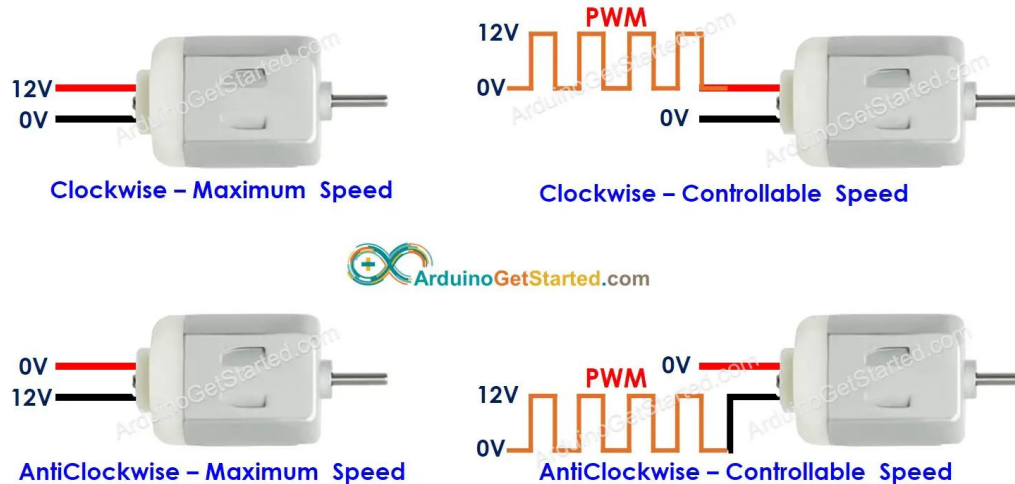
To have complete control over DC motor we have to control its speed and rotation direction. This can be achieved by combining these two techniques.

- PWM – to control speed
- H-Bridge – to control the rotation direction

# PWM – to control speed

The speed of a DC motor can be controlled by changing its input voltage. A common technique to do this is to use PWM (Pulse Width Modulation).

The more PWM the duty cycle is, the higher speed the DC motor rotates.

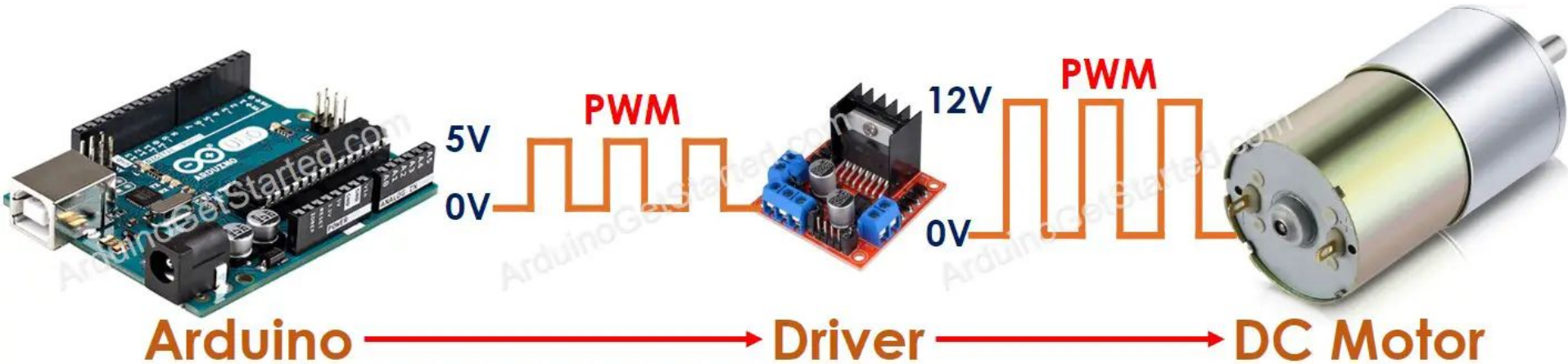


## **H-Bridge – to control the rotation direction**

The spinning direction of a DC motor can be controlled by changing the polarity of its input voltage. A common technique for doing this is to use an H-bridge.



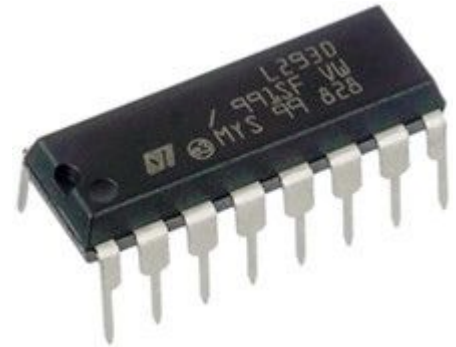
There are many kinds of chip/module (eg. L293D, L29N) can be used as DC motor drive.



# L293D Motor Driver IC

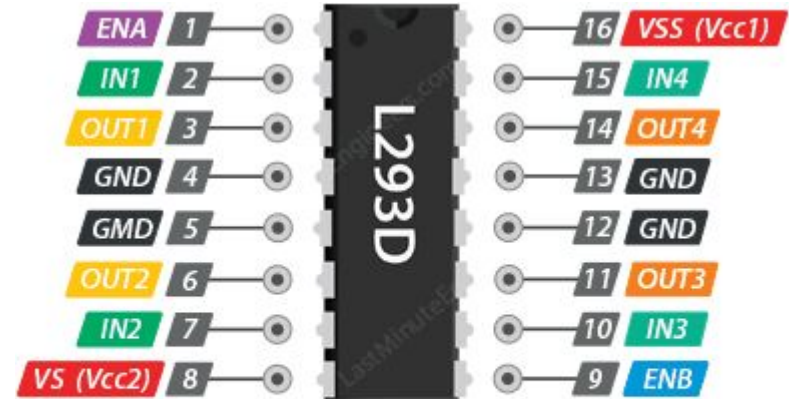
The L293D is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors or a single stepper motor. This means it can drive up to two motors individually which makes it ideal for building a two-wheeled robotic platform.

It is capable of driving four solenoids, four uni-directional DC motors, two bi-directional DC motors or one stepper motor.



# L293D Motor Driver IC Pinout

The L293D IC has a total of 16 pins that connect it to the outside world. The pinout is as follows:



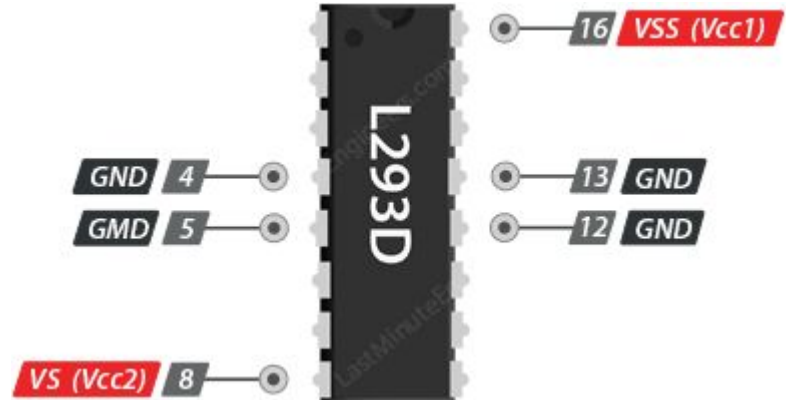
L293D Pinout

# Power Pins

**VS (Vcc2)** pin gives power to the internal H-Bridge of the IC to drive the motors. You can connect an input voltage anywhere between 4.5 to 36V to this pin.

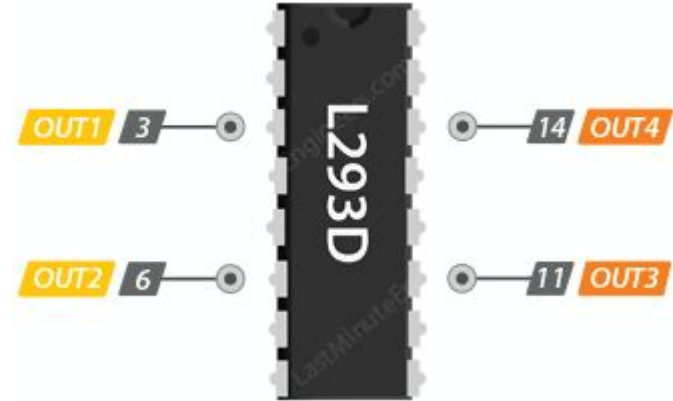
**VSS (Vcc1)** is used to drive the internal logic circuitry which should be 5V.

**GND** pins are common ground pins. All 4 GND pins are internally connected and used to dissipate the heat generated under high load conditions.



# Output channels

The L293D motor driver's output channels for the motor A and B are brought out to pins **OUT1,OUT2** and **OUT3,OUT4** respectively. You can connect two 5-36V DC motors to these pins.



# Direction Control Pins

The IC has two direction control pins for each channel. The **IN1** and **IN2** pins control the spinning direction of motor A; While **IN3** and **IN4** control the spinning direction of motor B.

The spinning direction of the motor can be controlled by applying logic HIGH (5V) or logic LOW (Ground) to these inputs. The chart below shows how this is done.

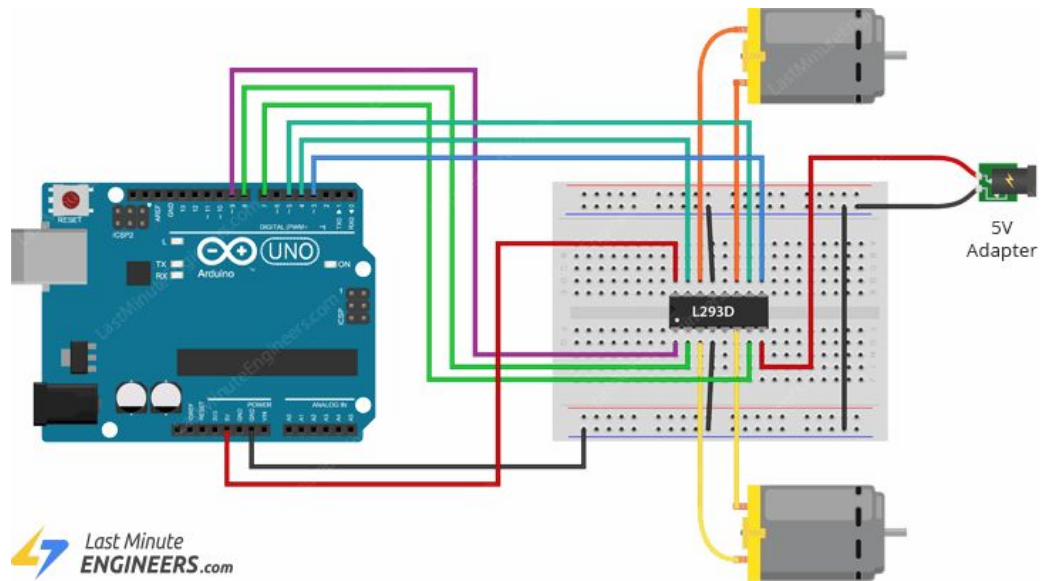
| IN1     | IN2     | Spinning Direction |
|---------|---------|--------------------|
| Low(0)  | Low(0)  | Motor OFF          |
| High(1) | Low(0)  | Forward            |
| Low(0)  | High(1) | Backward           |
| High(1) | High(1) | Motor OFF          |



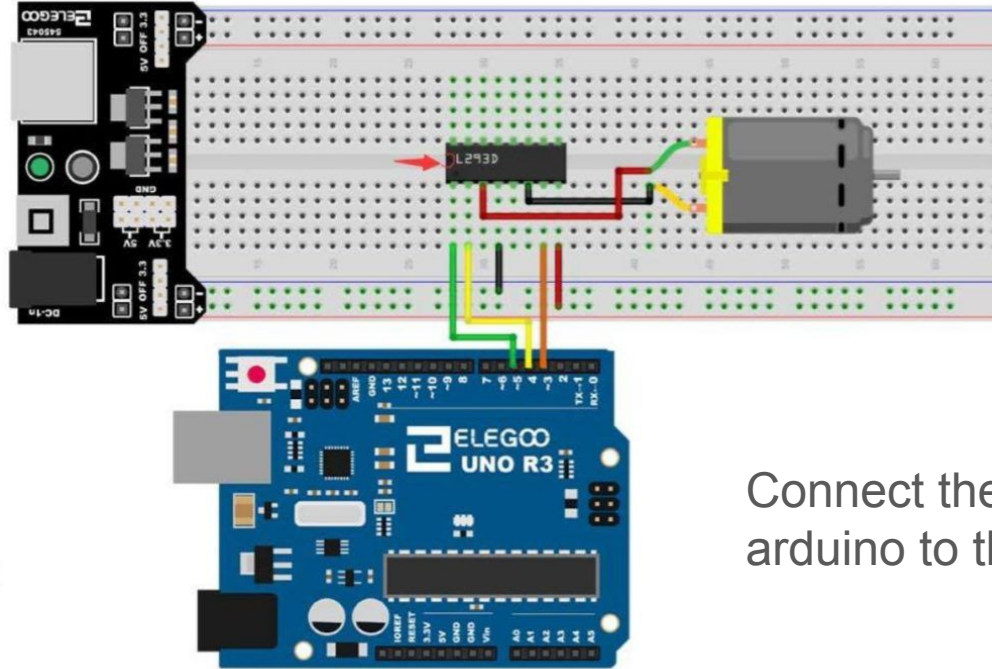
The speed control pins **ENA** and **ENB** are used to turn on/off the motors and control its speed.

Pulling these pins HIGH will cause the motors to spin, while pulling it LOW will stop them. But, with Pulse Width Modulation (PWM), you can actually control the speed of the motors.









Connect the ground from the arduino to the breadboard.

```
Arduino
1 int speedPin=5;
2 int dir1=4;
3 int dir2=3;
4 int mSpeed=90;
5
6 void setup() {
7   // put your setup code here, to run once:
8   pinMode(speedPin,OUTPUT);
9   pinMode(dir1,OUTPUT);
10  pinMode(dir2,OUTPUT);
11  Serial.begin(9600);
12
13 }
14
15 void loop() {
16   // put your main code here, to run repeatedly:
17   digitalWrite(dir1,LOW);
18   digitalWrite(dir2,HIGH);
19   analogWrite(speedPin,255);
20   delay(25);
21   analogWrite(speedPin,mSpeed);
22   delay(5000);
23 }
```

[https://www.tinkercad.com/things/33mWP95Ea5p-copy-of-arduino-power-supply/editel?sharecode=dHvFHKHplWPdrajIN1h\\_ewIaPZqVlPxj6XRoESbtHC4](https://www.tinkercad.com/things/33mWP95Ea5p-copy-of-arduino-power-supply/editel?sharecode=dHvFHKHplWPdrajIN1h_ewIaPZqVlPxj6XRoESbtHC4)

<https://arduinogetstarted.com/tutorials/arduino-dc-motor>

<https://lastminuteengineers.com/l293d-dc-motor-arduino-tutorial/>

[https://www.youtube.com/watch?v=fPLEncYrl4Q&ab\\_channel=PaulMcWhorter](https://www.youtube.com/watch?v=fPLEncYrl4Q&ab_channel=PaulMcWhorter)

<https://toptechboy.com/arduino-tutorial-37-understanding-how-to-control-dc-motors-in-projects/>

**Additionally, check DS3231 Arduino Real Time Clock module. You can use the library function for DS3231 real time clock module**