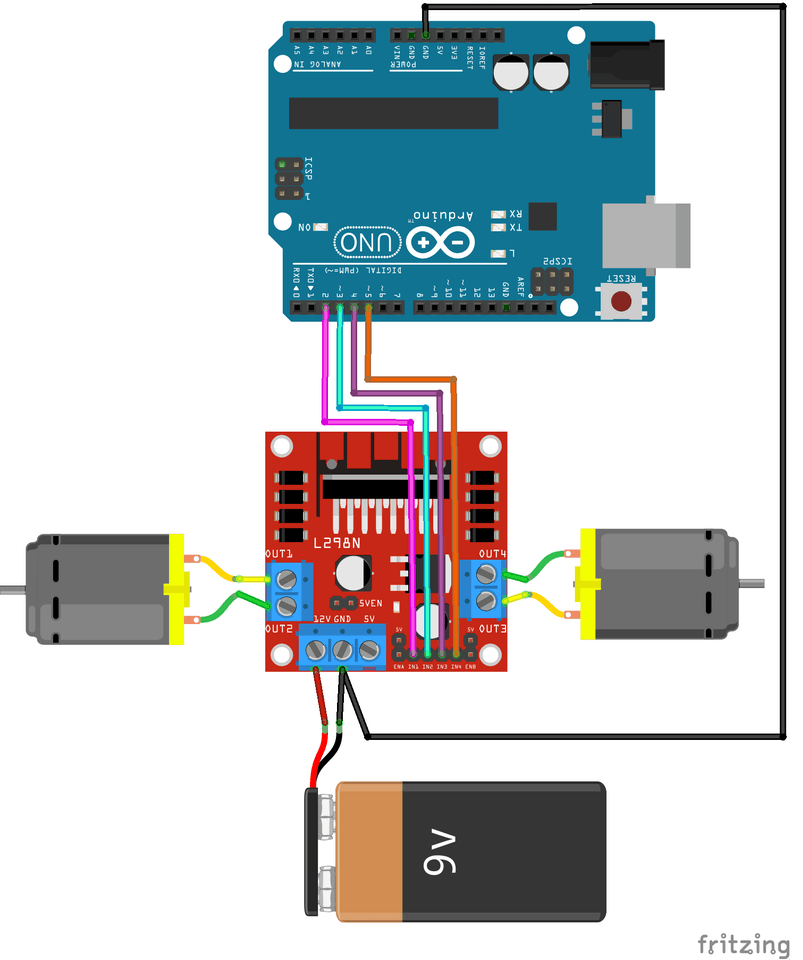
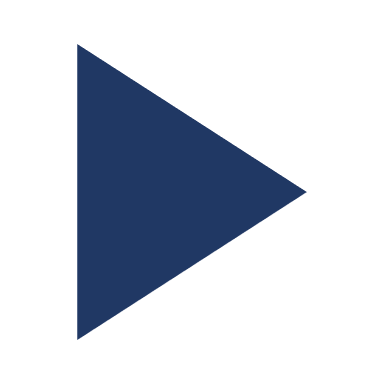
# Caret Left with solid fillMotor Driver

**Eng. Mahmoud Karam**



****

## Content

[Motor Driver 0](#_Toc79865354)

[Content 1](#_Toc79865355)

[1.1 Introduction 2](#_Toc79865356)

[1.1.1 Controlling DC Motor 2](#_Toc79865357)

[1.2 L298 Module 3](#_Toc79865358)

[1.2.1 L298N Module Pinout 4](#_Toc79865359)

[1.2.2 Power Supply 5](#_Toc79865360)

[1.2.3 Voltage Drop of L298N 6](#_Toc79865361)

[1.2.4 Output Pins 6](#_Toc79865362)

[1.2.5 Control Pins 7](#_Toc79865363)

[1.3 Wiring motor driver module with microcontroller 8](#_Toc79865364)

[1.4 References 9](#_Toc79865365)

## Introduction

No robot can move without motors. So, motors is an essential part and knowing how to control them is a must. There are many motors types, but our aim is to control DC motors.

### Controlling DC Motor

In order to have a complete control over DC motor, we have to control its speed and rotation direction which can be achieved by combining two techniques:

1. PWM – For controlling speed.
2. H-Bridge – For controlling rotation direction.

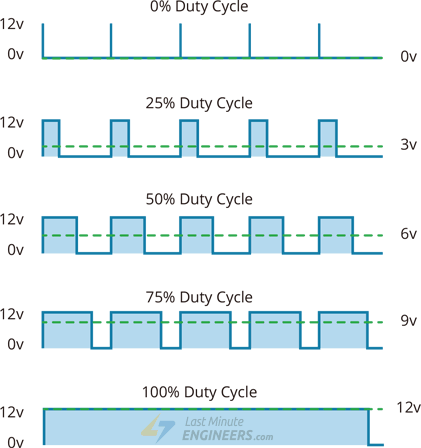
DC motors have current and voltage requirements that are beyond the capabilities of microcontrollers. It is necessary to use some external electronics to drive and control the motor, and you’ll probably need a separate power supply as well.

One of the easiest and inexpensive way to control DC motors is to interface L298N Motor Driver which is a dual H-Bridge motor driver which allows speed (PWM) and direction (H-Bridge) 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.

#### PWM – For controlling speed

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

PWM is a technique where average value of the input voltage is adjusted by sending a series of ON-OFF pulses as the following diagram.



The average voltage is proportional to the width of the pulses known as Duty Cycle.

Duty Cycle = (ON-period / Off period) \* 100

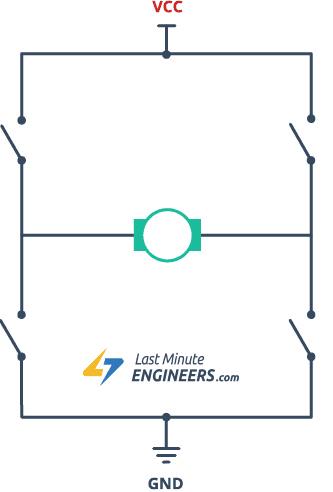
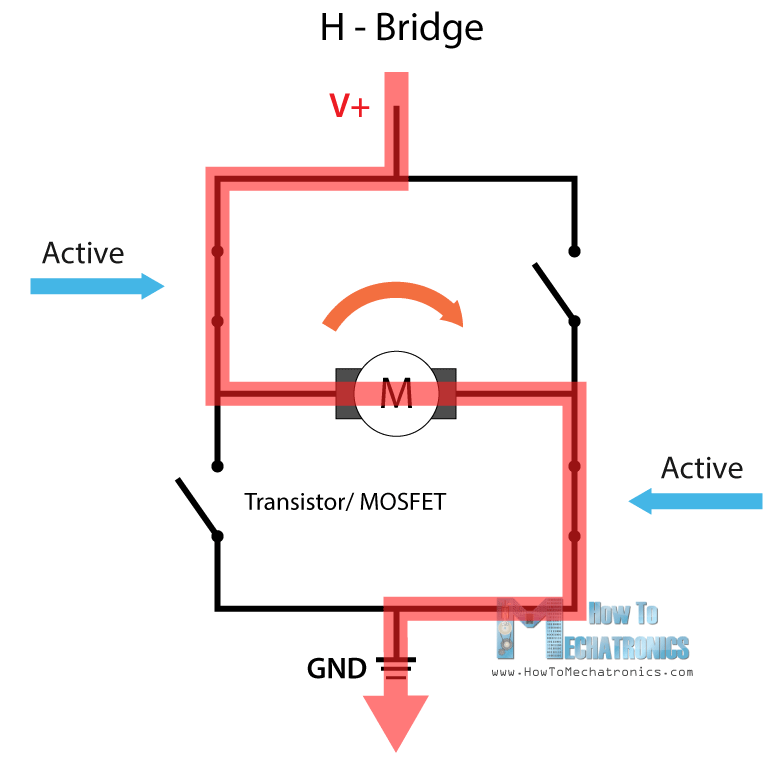
Duty cycle is proportional to the average voltage being applied to the dc motor (Speed).

Average voltage = (Duty Cycle / 100) \* Supply Voltage

#### H-Bridge – For controlling rotation direction

The DC motor’s spinning direction can be controlled by changing polarity of its input voltage. A common technique for doing this is to use an H-Bridge which contains four switches with the motor at the center forming an H-like arrangement.

Closing two particular switches at the same time reverses the polarity of the voltage applied to the motor. This causes change in spinning direction of the motor.

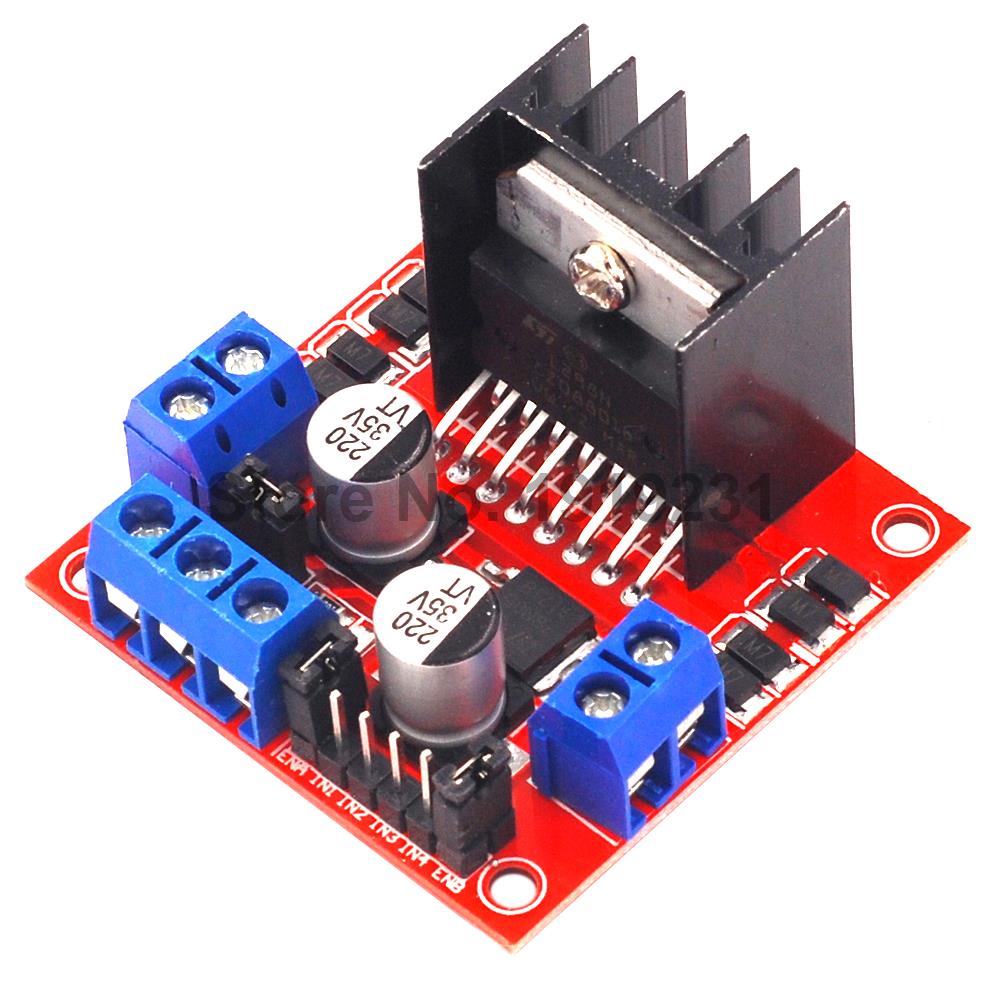
 

## L298 Module

L298N module is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.

There are two versions of the driver modules:

1. L298: Can drive two motors with 2A for each one.
2. L298N: Can drive two motors with 3A for each one.

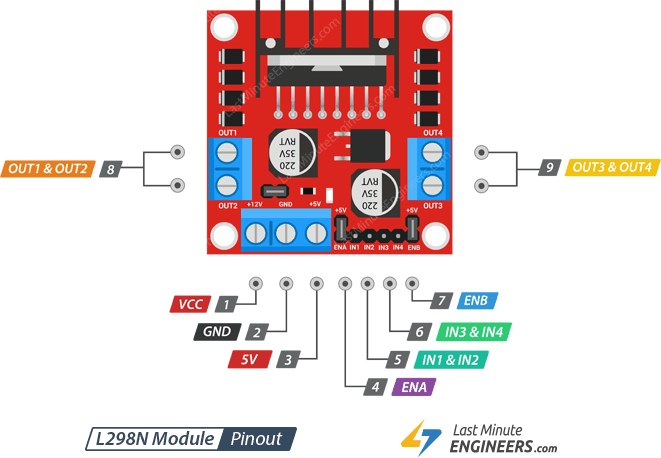


L298

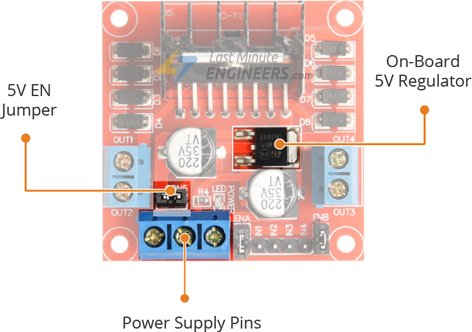
L298N

We will deal with L298 module.

### L298N Module Pinout



### Power Supply



The L298 motor driver module is powered through 3-pins:

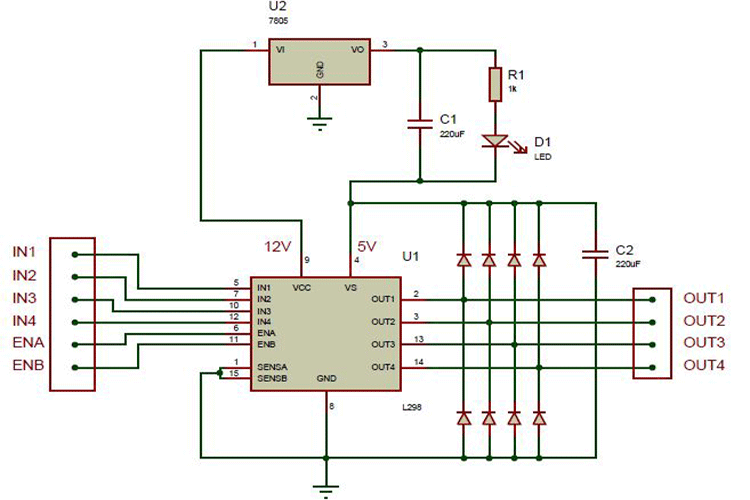
1. Motors Power supply (VCC): supplies power for the motors. from 5 to 35V. Remember, if the 5V-EN jumper is in place, supply 2 extra volts than motor’s actual voltage requirement, to get maximum speed out of motor.
2. GND: a common ground pin.
3. 5V logic power supply (Vss) (In/Out): Module has an on-board 78M05 5V regulator.

There is a jumper called 5V-EN jumper. It is used to enable/disable regulator:

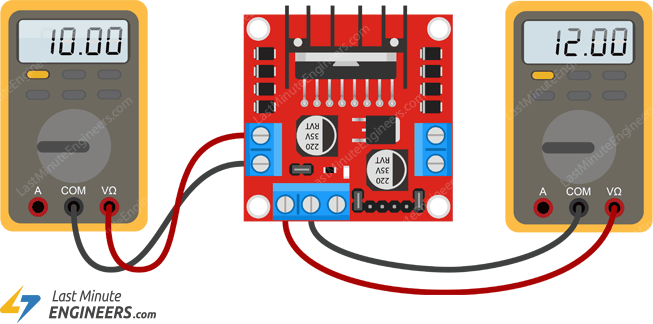
1. If the 5V-EN jumper is removed, connect 5V pin to a 5V supply to supply power for the switching logic circuitry inside L298N IC.
2. If the 5V-EN jumper is in place, this pin acts as an output and can be used to power up microcontroller, and the switching logic circuitry inside L298N IC, and other circuits with a maximum of **0.5A**.

NOTE:

1. If the motor VCC > 12V, remove the 5V-EN jumper to avoid damaging the regulator, and separate 5V should be given through 5V terminal to power the internal circuitry.



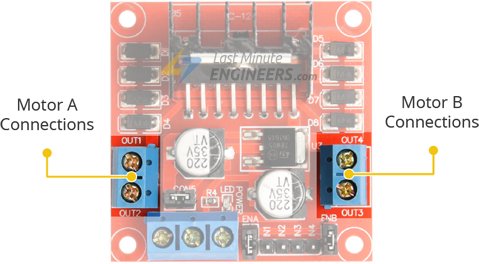
### Voltage Drop of L298N



The voltage drop of the L298 module is about 2V. This is due to the internal voltage drop in the switching transistors in the H-Bridge circuit and the voltage regulator.

So, 12V power supply terminal will be around 10V. So, to get the maximum speed of a 12V DC motor supply it with around 14V.

### Output Pins



There are two output channels for two motors having voltages between 5 to 35V:

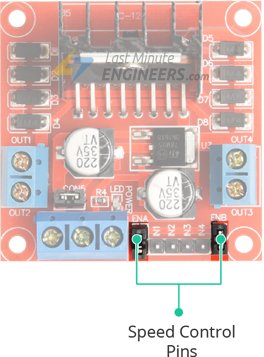
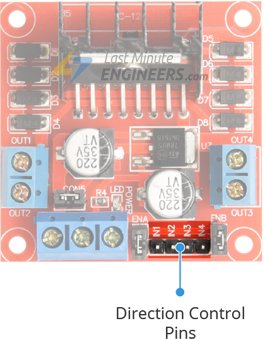
OUT1 & OUT2 for motor A.

OUT3 & OUT4 for motor B.

Each channel on the module can deliver up to 2A to the DC motor. However, the amount of current supplied to the motor depends on system’s power supply.

### Control Pins

For each channel, there are two types of control pins to control speed and spinning direction.



#### Direction Control Pins

Using the direction control pins, the motor can spin forward or backward. These pins actually control the switches of the H-Bridge circuit.

The module has two direction control pins for each channel:

IN1 & IN2 pins are used to control spinning direction of Motor A.

IN3 & IN4 pins are used to control spinning direction of Motor B.

The spinning direction of a motor is illustrated below.

Table

Description automatically generated

The next diagram shows working of case 2 in the above table for motor A.

Diagram, schematic

Description automatically generated

#### Speed Control Pins

Pins ENA and ENB are used to control speed of the motors A and B respectively:

1. Pulling these pins HIGH will make the motors spin at maximum speed. By default, there are jumpers connecting these pins to HIGH.
2. Pulling them LOW will make them stop.
3. With Pulse Width Modulation (PWM), we can control the speed of the motors by removing the jumper and connecting these pins to a PWM device.

There module come with jumpers on these pins to pull them HIGH. So, by default motors spins at the maximum speed.

So, to control the speed of motors programmatically, remove the jumpers and connect them to PWM-enabled pins on the microcontroller.

**NOTE**: Depending on the applied voltage and the motor itself, at lower speeds the motor is not able to start moving and it produces a buzzing sound. So, experiment some values and set the minimum PWM above the minimum by some offset (In 6VDC motors, use 85 minimum PWM).

## Wiring motor driver module with microcontroller

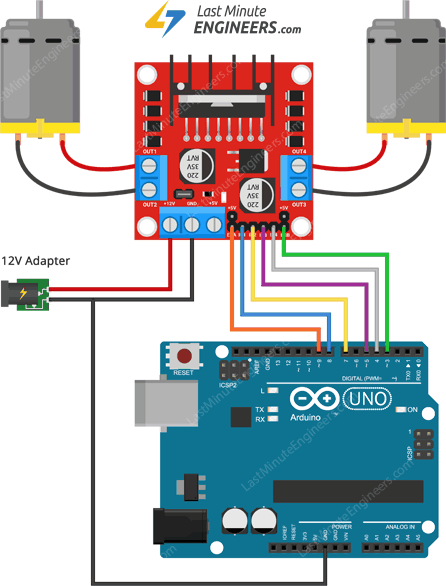
We are using DC Gearbox Motors (also known as ‘TT’ motors). They are rated for 3 to 12V.

So, to connect the motor driver to the microcontroller:

1. Connect 12V power supply to the motors at VCC and GND pins. Considering internal voltage drop of L298N IC, the motors will receive 10V and will spin at slightly lower RPM.
2. Supply 5 Volts for the L298N’s logic circuitry. On-board 5V regulator can be used and derive 5 volts from the motor power supply (VCC). so, place the 5V-EN jumper.
3. The input and enable pins(ENA, IN1, IN2, IN3, IN4 and ENB) of the L298N module are connected to six Arduino digital output pins(9, 8, 7, 5, 4 and 3).

NOTE: The Arduino output pins 9 and 3 are both PWM-enabled.

1. Connect one motor to terminal A (OUT1 & OUT2) and the other motor to terminal B (OUT3 & OUT4).



## Arduino Code

// Motor A connections

int enA = 9;

int in1 = 8;

int in2 = 7;

// Motor B connections

int enB = 3;

int in3 = 5;

int in4 = 4;

void setup() {

// Set all the motor control pins to outputs

pinMode(enA, OUTPUT);

pinMode(enB, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

// Turn off motors - Initial state

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

void loop() {

directionControl();

delay(1000);

speedControl();

delay(1000);

}

// This function lets you control spinning direction of motors

void directionControl() {

// Set motors to maximum speed

// For PWM maximum possible values are 0 to 255

analogWrite(enA, 255);

analogWrite(enB, 255);

// Turn on motor A & B

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

delay(2000);

// Now change motor directions

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(2000);

// Turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

// This function lets you control speed of the motors

void speedControl() {

// Turn on motors

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

// Accelerate from zero to maximum speed

for (int i = 0; i < 256; i++) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Decelerate from maximum speed to zero

for (int i = 255; i >= 0; --i) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Now turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

**Code Explanation**

It doesn’t require any libraries to get it working.

1. Declaring Arduino pins to which L298N’s control pins are connected.

// Motor A connections

int enA = 9;

int in1 = 8;

int in2 = 7;

// Motor B connections

int enB = 3;

int in3 = 5;

int in4 = 4;

1. motor control pins are declared as digital OUTPUT and pulled LOW.

void setup() {

// Set all the motor control pins to outputs

pinMode(enA, OUTPUT);

pinMode(enB, OUTPUT);

pinMode(in1, OUTPUT);

pinMode(in2, OUTPUT);

pinMode(in3, OUTPUT);

pinMode(in4, OUTPUT);

// Turn off motors - Initial state

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

1. In loop section, we call two user defined functions at an interval of a second.

void loop() {

directionControl();

delay(1000);

speedControl();

delay(1000);

}

* 1. directionControl(): spins both motors forward at maximum speed for two seconds, then reverses the motor’s spinning direction and spins for another two seconds, finally turns the motors off.

void directionControl() {

// Set motors to maximum speed

// For PWM maximum possible values are 0 to 255

analogWrite(enA, 255);

analogWrite(enB, 255);

// Turn on motor A & B

digitalWrite(in1, HIGH);

digitalWrite(in2, LOW);

digitalWrite(in3, HIGH);

digitalWrite(in4, LOW);

delay(2000);

// Now change motor directions

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

delay(2000);

// Turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

* 1. speedControl():accelerates both the motors from zero to maximum speed by producing PWM signals using analogWrite() function, then it decelerates them back to zero, finally it turns the motors off.

void speedControl() {

// Turn on motors

digitalWrite(in1, LOW);

digitalWrite(in2, HIGH);

digitalWrite(in3, LOW);

digitalWrite(in4, HIGH);

// Accelerate from zero to maximum speed

for (int i = 0; i < 256; i++) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Decelerate from maximum speed to zero

for (int i = 255; i >= 0; --i) {

analogWrite(enA, i);

analogWrite(enB, i);

delay(20);

}

// Now turn off motors

digitalWrite(in1, LOW);

digitalWrite(in2, LOW);

digitalWrite(in3, LOW);

digitalWrite(in4, LOW);

}

## References

|  |  |
| --- | --- |
| [1] | "Interface L298N DC Motor Driver Module with Arduino," Last Minute Engineers, [Online]. Available: https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial/. [Accessed 2021]. |
| [2] | "L298N Motor Driver – Arduino Interface, How It Works, Codes, Schematics," HowToMechatronics.com, [Online]. Available: https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/. [Accessed 2021]. |