# SNS COLLEGE OF TECHNOLOGY

**Coimbatore-35**
**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC
with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna
University, Chennai

## DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

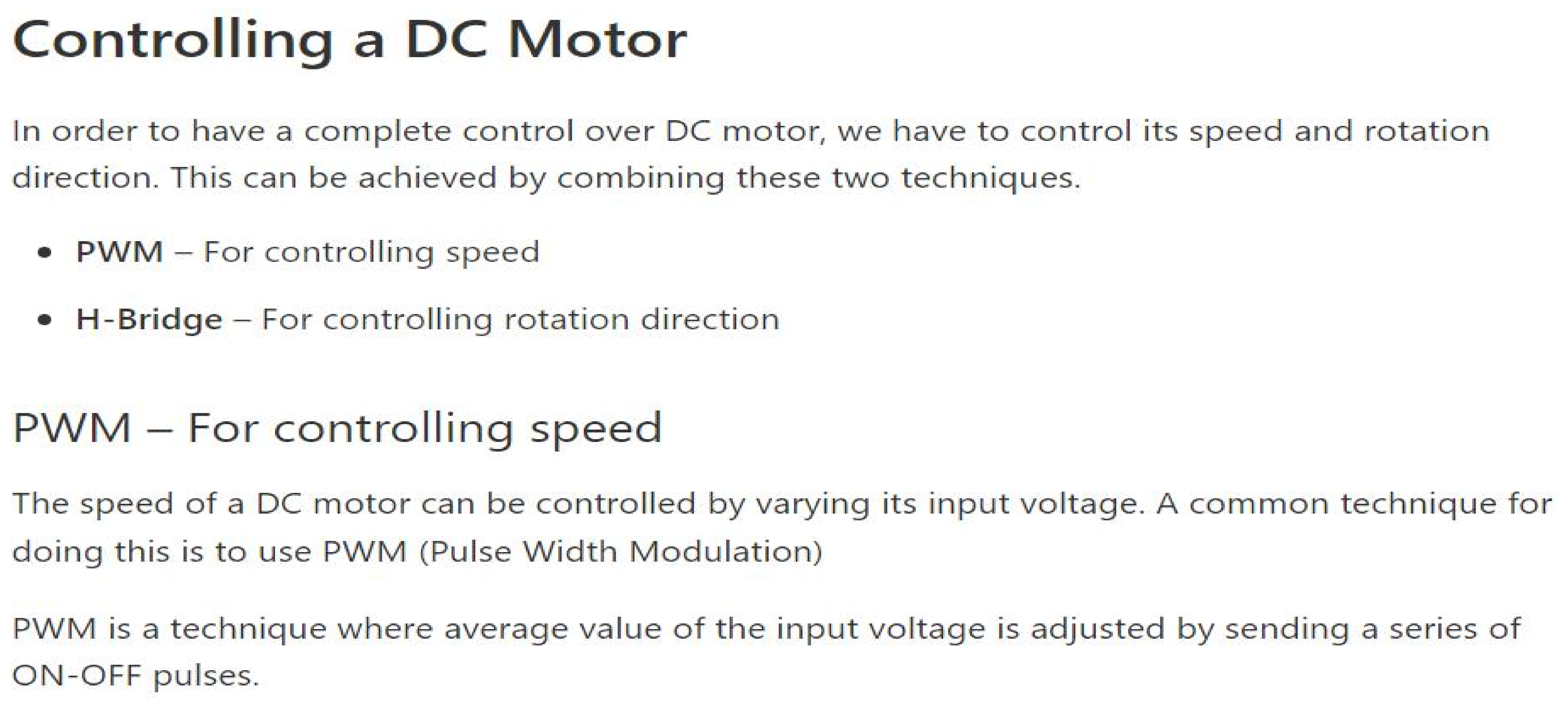
## 19ECT213-IoT SYSTEM ARCHITECTURE

II YEAR/ IV SEMESTER

UNIT 3 – MICROCONTROLLER AND INTERFACING TECHNIQUES FOR IoT DEVICES

**Topic 3: DC Motor with L298N Motor Driver Controller-Servos**

# Interface L298N DC Motor Driver Module with Arduino



## Controlling a DC Motor

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

- **PWM** – For controlling speed
- **H-Bridge** – For controlling rotation direction

### PWM – For controlling speed

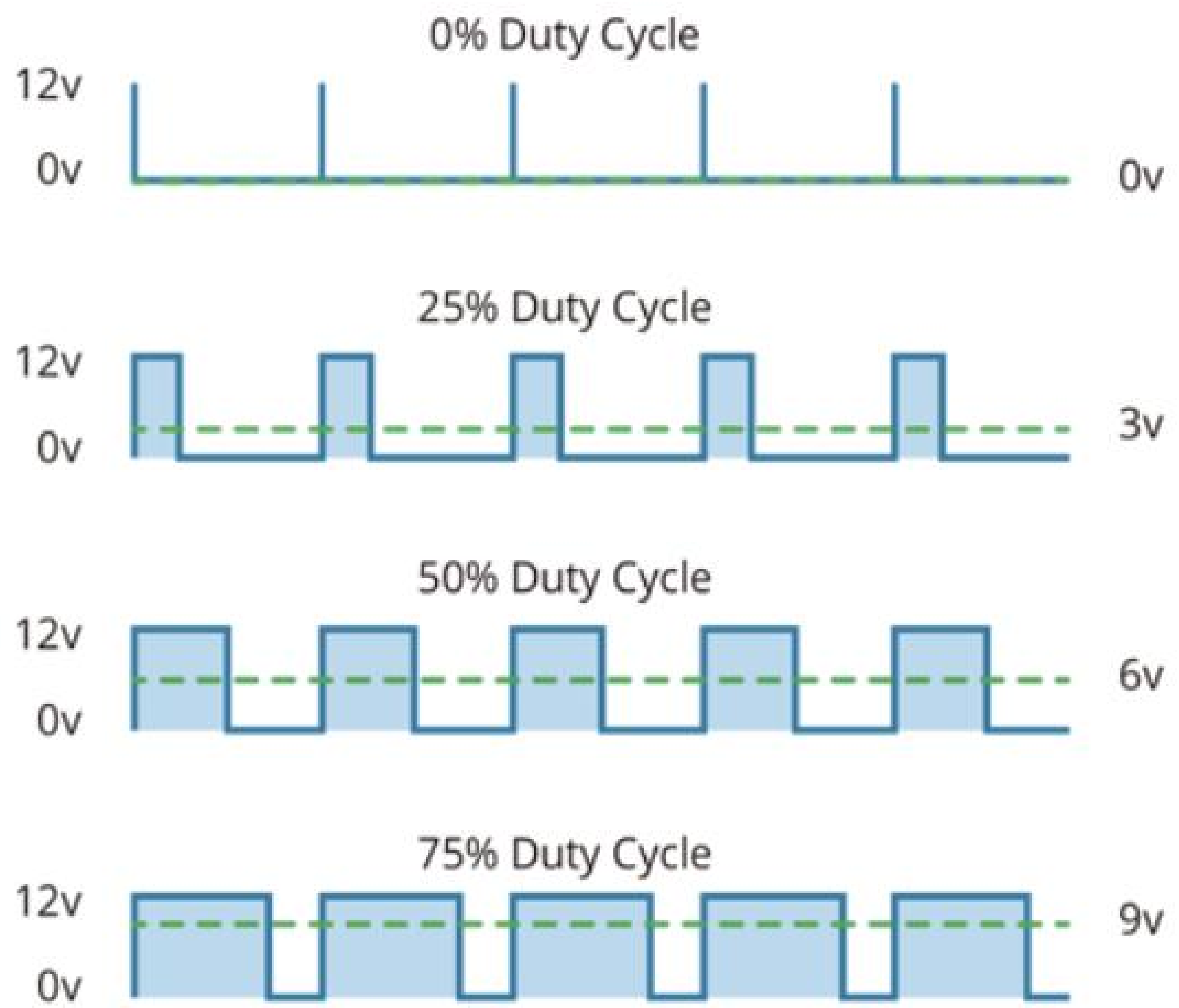
The speed of a DC motor can be controlled by varying its input voltage. A common technique for doing this is to use PWM (Pulse Width Modulation)

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

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

The higher the duty cycle, the greater the average voltage being applied to the dc motor(High Speed) and the lower the duty cycle, the less the average voltage being applied to the dc motor(Low Speed).

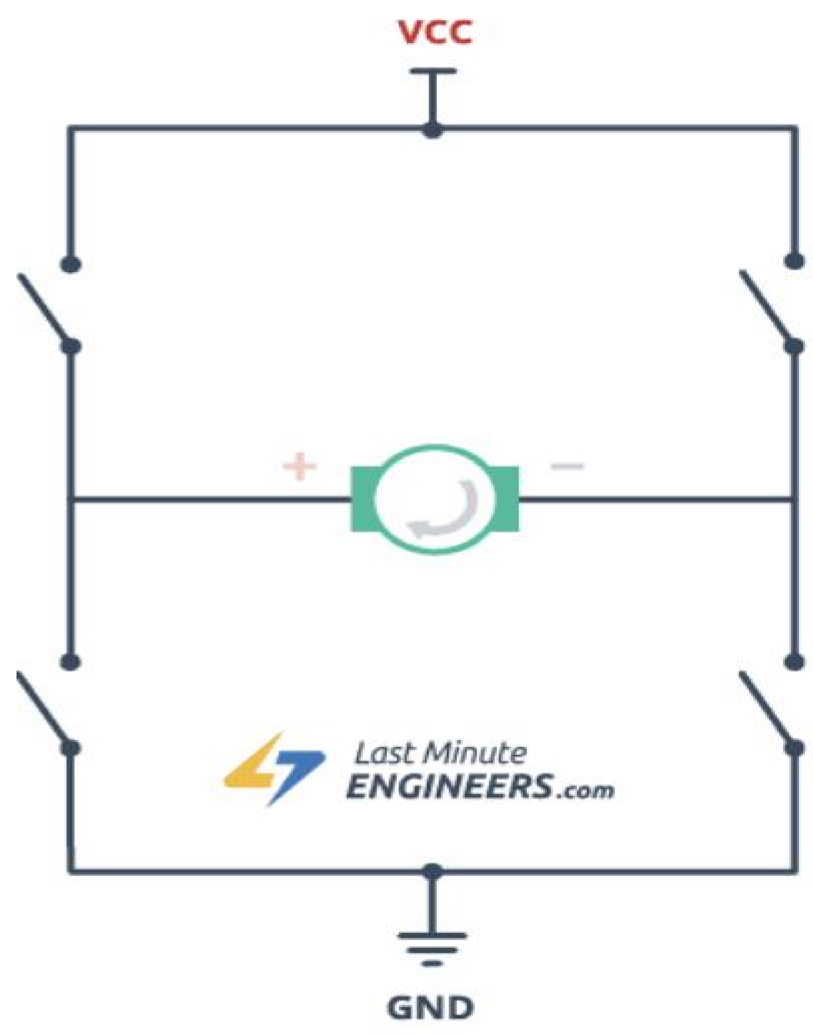
Below image illustrates PWM technique with various duty cycles andaverage voltages.



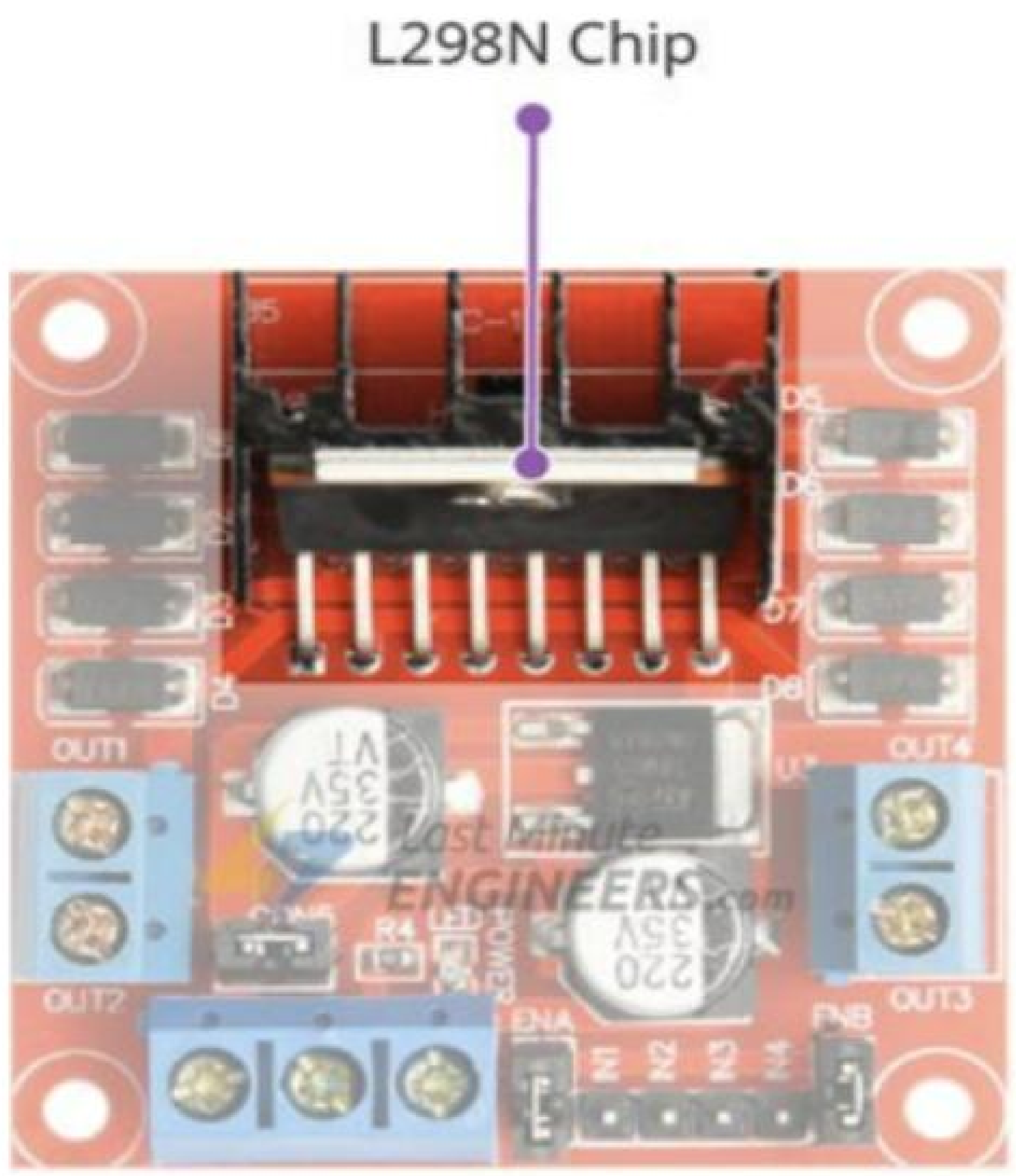
## H-Bridge – For controlling rotation direction

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.

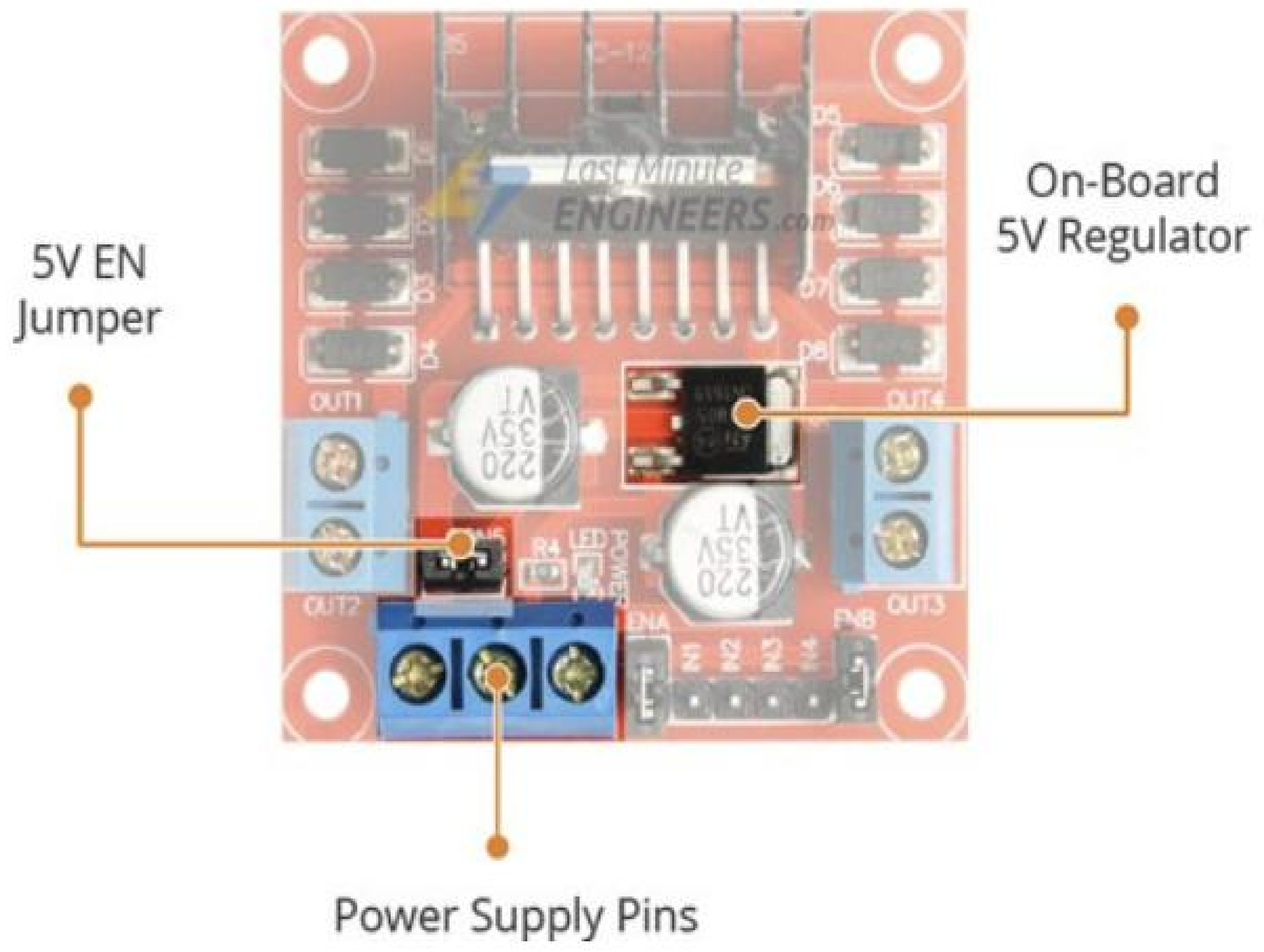
An H-Bridge circuit contains four switches with the motor at the center forming an H-like arrangement.



## L298N Motor Driver IC

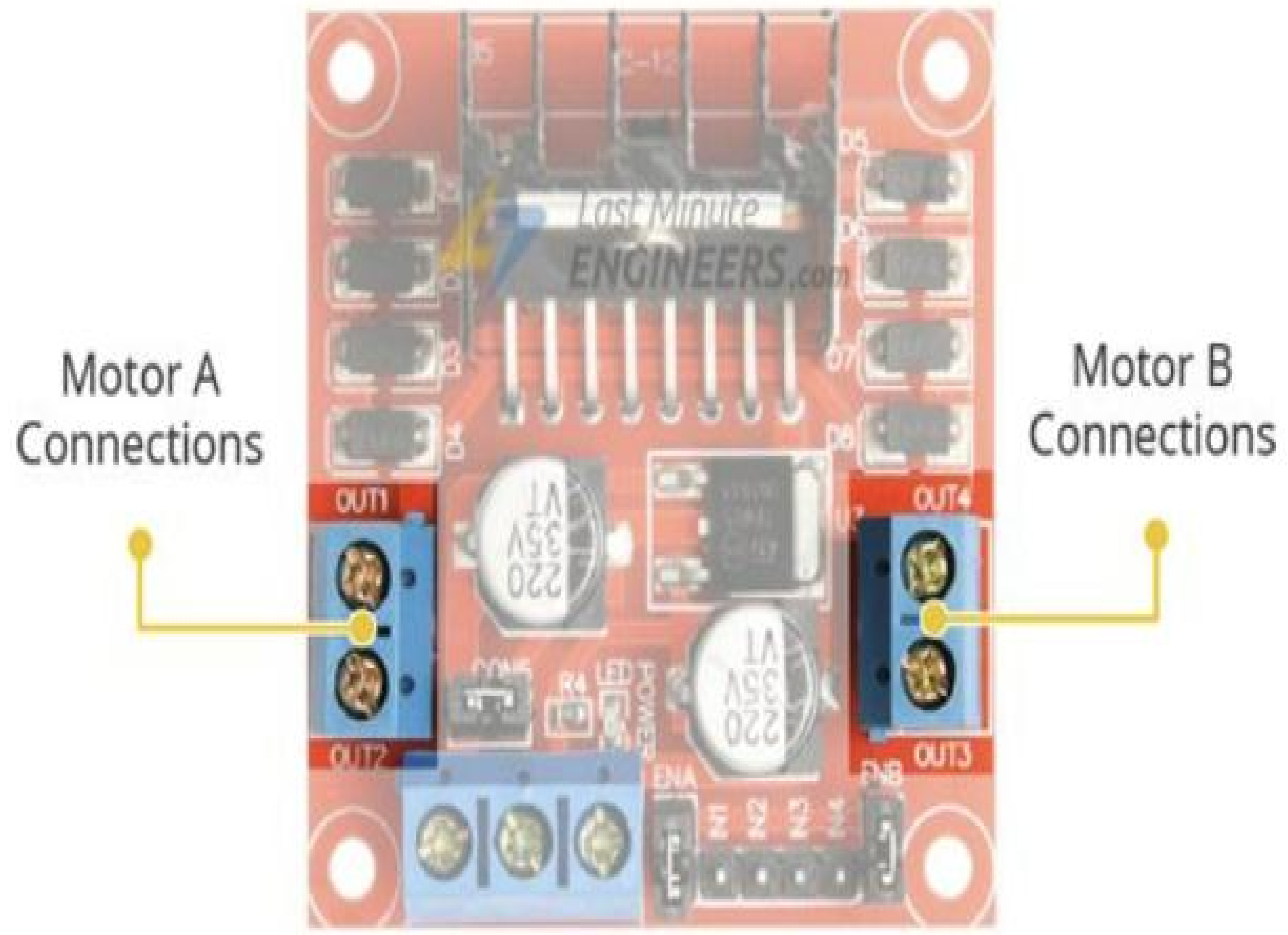


## Power Supply



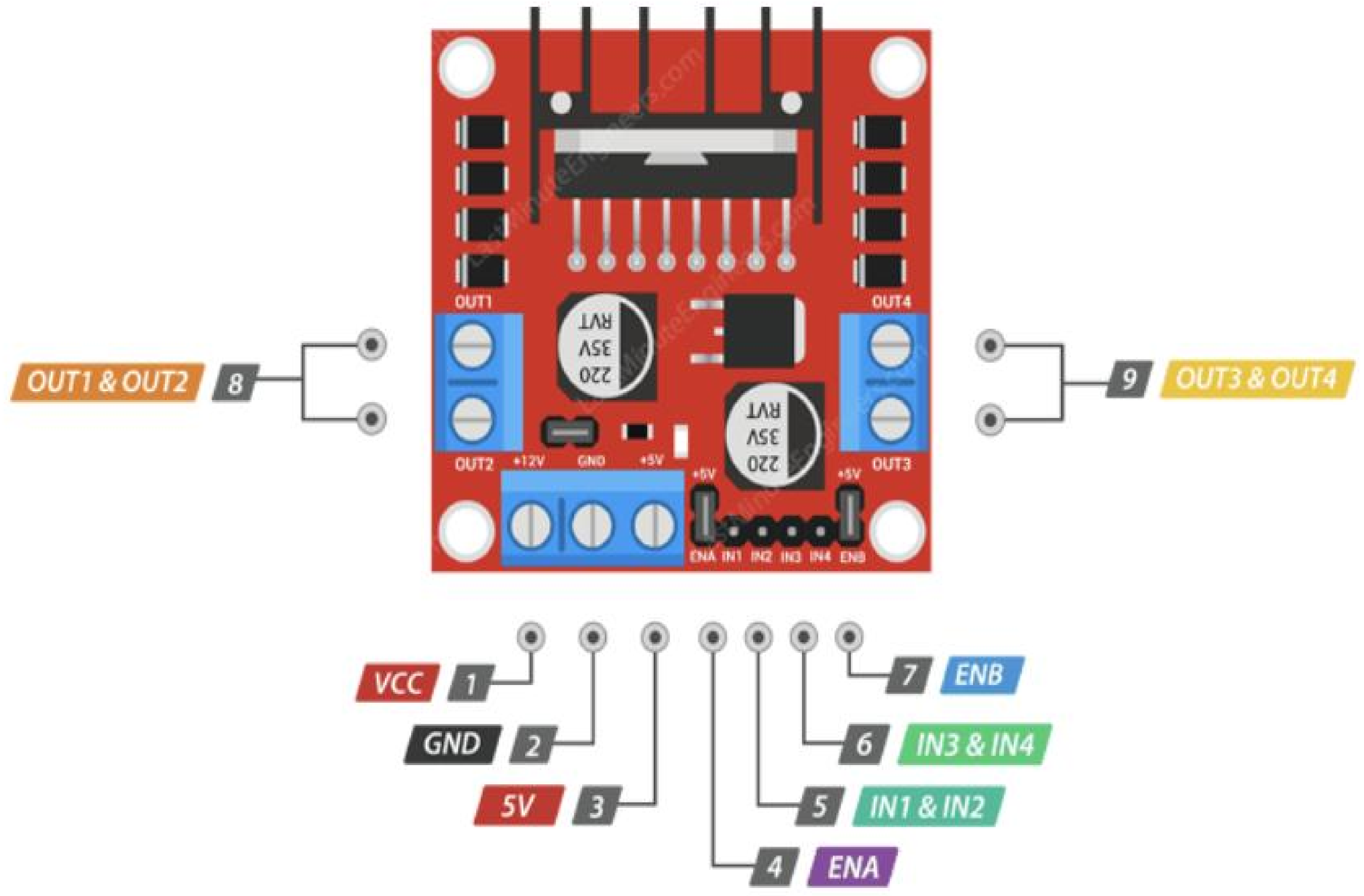
The L298N motor driver module is powered through 3-pin 3.5mm-pitch screw terminals. It consists of pins for motor power supply(Vs), ground and 5V logic power supply(Vss).

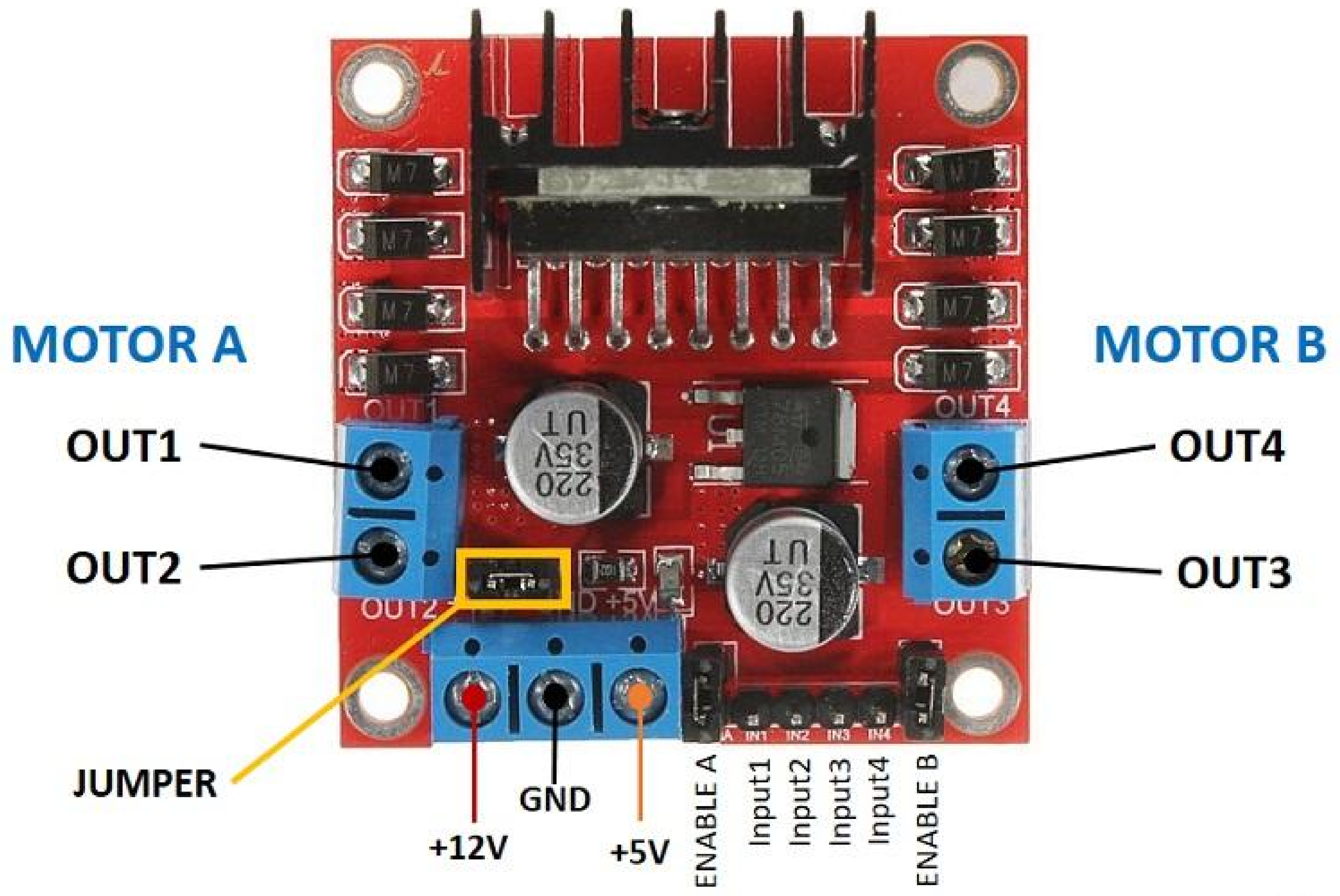
## Output Pins


Motor A
Connections
Motor B
Connections
OUT1
OUT2
OUT3
OUT4
ENA
ENB
Last Minute ENGINEERS.com

# L298N Motor Driver Module Pinout

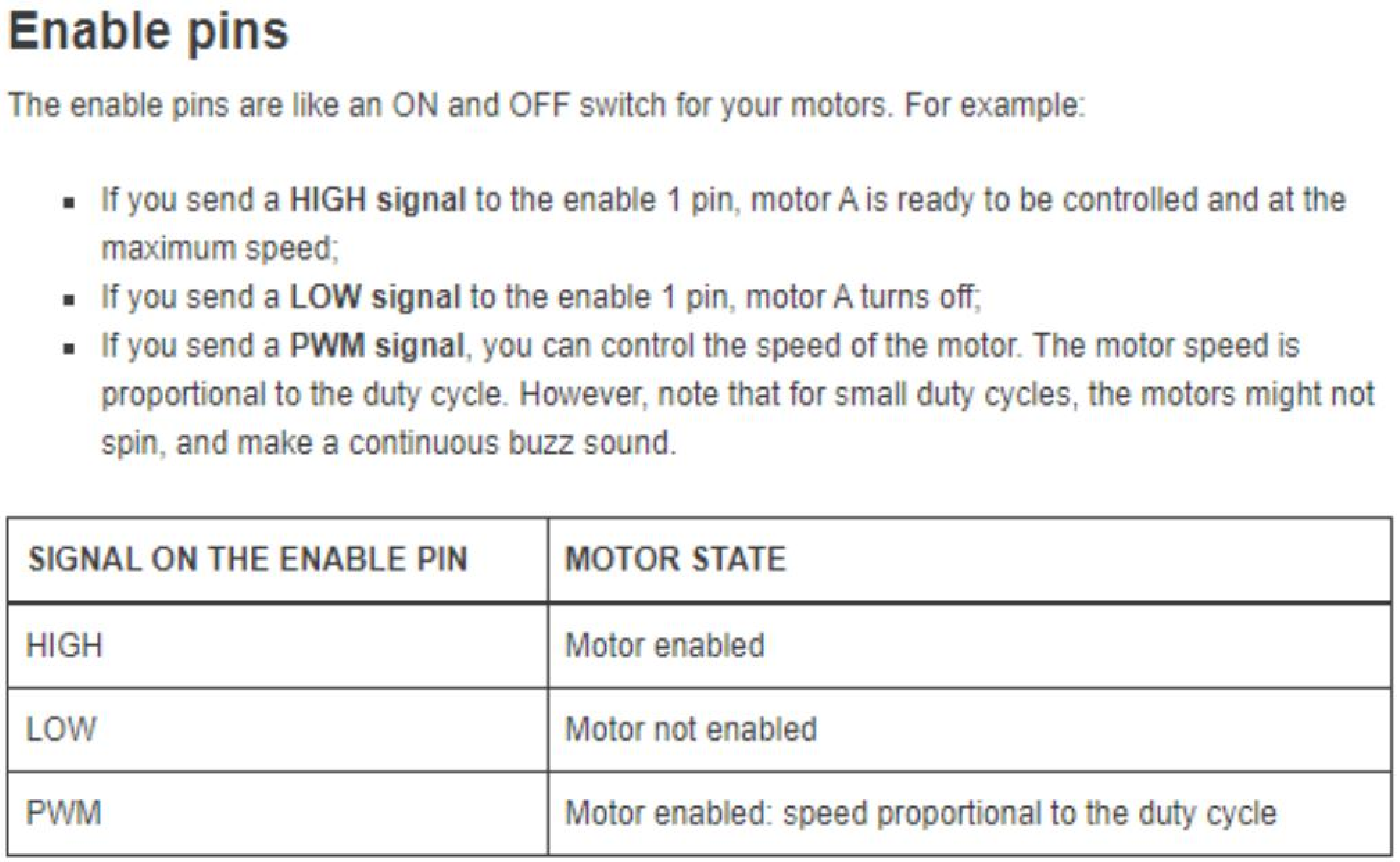
Before diving into hookup and example code, let's first take a look at its Pinout.





- **IN1:** Input 1 for Motor A
- **IN2**: Input 2 for Motor A
- **IN3**: Input 1 for Motor B
- **IN4**: Input 2 for Motor B
- **EN1**: Enable pin for Motor A
- **EN2**: Enable pin for Motor B

- **OUT1**: DC motor A + terminal
- **OUT2**: DC motor A – terminal
- **OUT3**: DC motor B + terminal
- **OUT4**: DC motor B – terminal

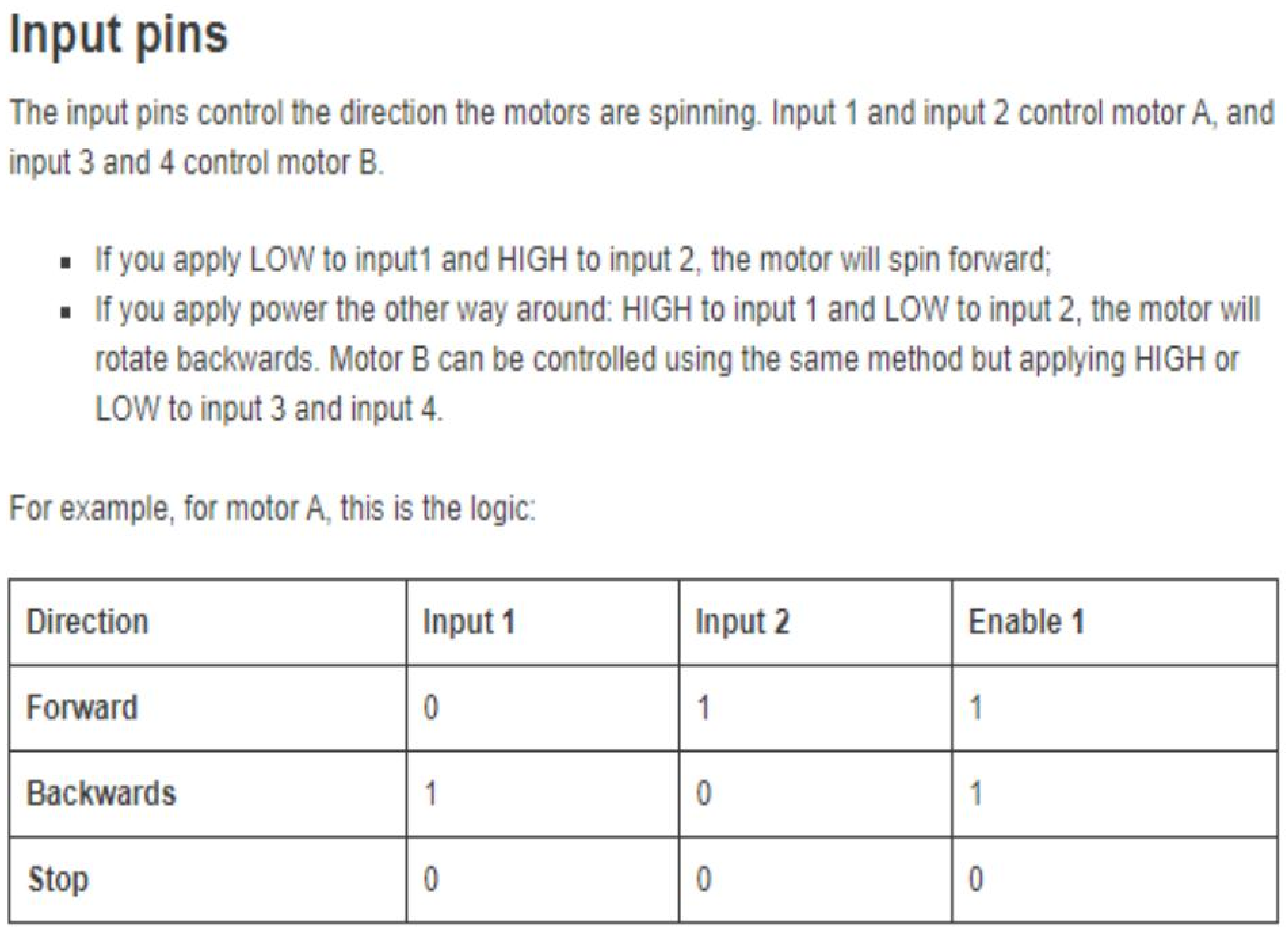


## Enable pins

The enable pins are like an ON and OFF switch for your motors. For example:

- If you send a **HIGH signal** to the enable 1 pin, motor A is ready to be controlled and at the maximum speed;
- If you send a **LOW signal** to the enable 1 pin, motor A turns off;
- If you send a **PWM signal**, you can control the speed of the motor. The motor speed is proportional to the duty cycle. However, note that for small duty cycles, the motors might not spin, and make a continuous buzz sound.

| SIGNAL ON THE ENABLE PIN | MOTOR STATE |
|---|---|
| HIGH | Motor enabled |
| LOW | Motor not enabled |
| PWM | Motor enabled: speed proportional to the duty cycle |



## Input pins

The input pins control the direction the motors are spinning. Input 1 and input 2 control motor A, and input 3 and 4 control motor B.

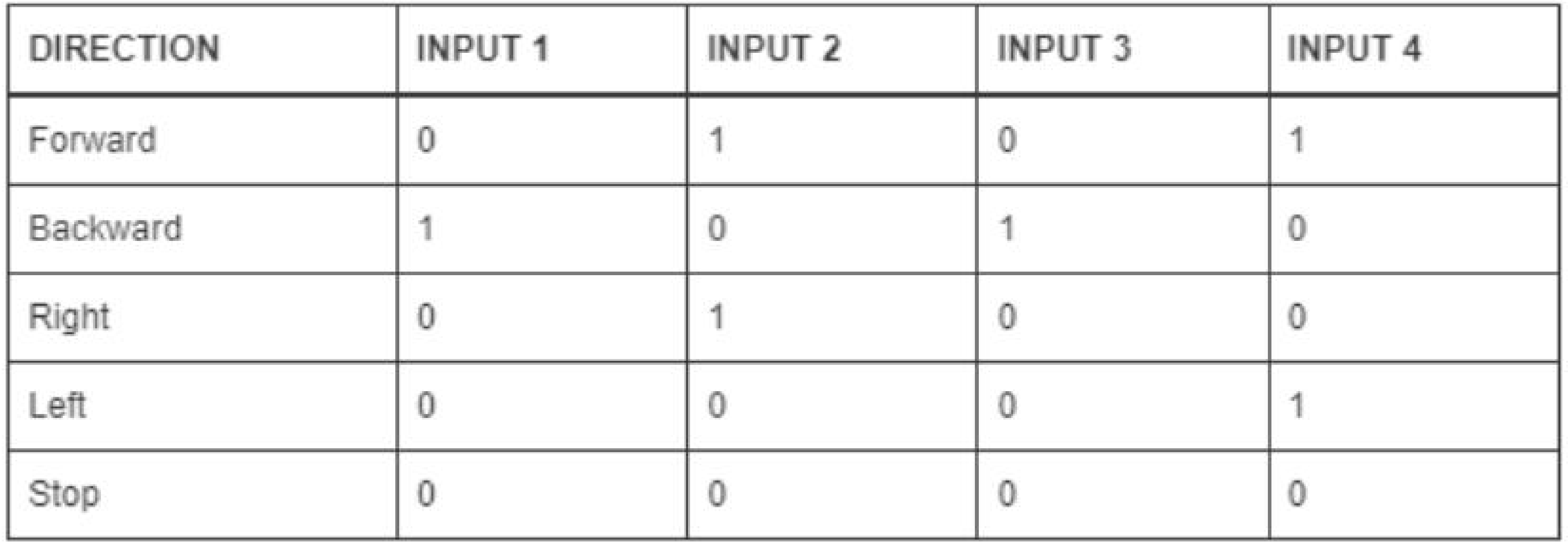
- If you apply LOW to input1 and HIGH to input 2, the motor will spin forward;
- If you apply power the other way around: HIGH to input 1 and LOW to input 2, the motor will rotate backwards. Motor B can be controlled using the same method but applying HIGH or LOW to input 3 and input 4.

For example, for motor A, this is the logic:

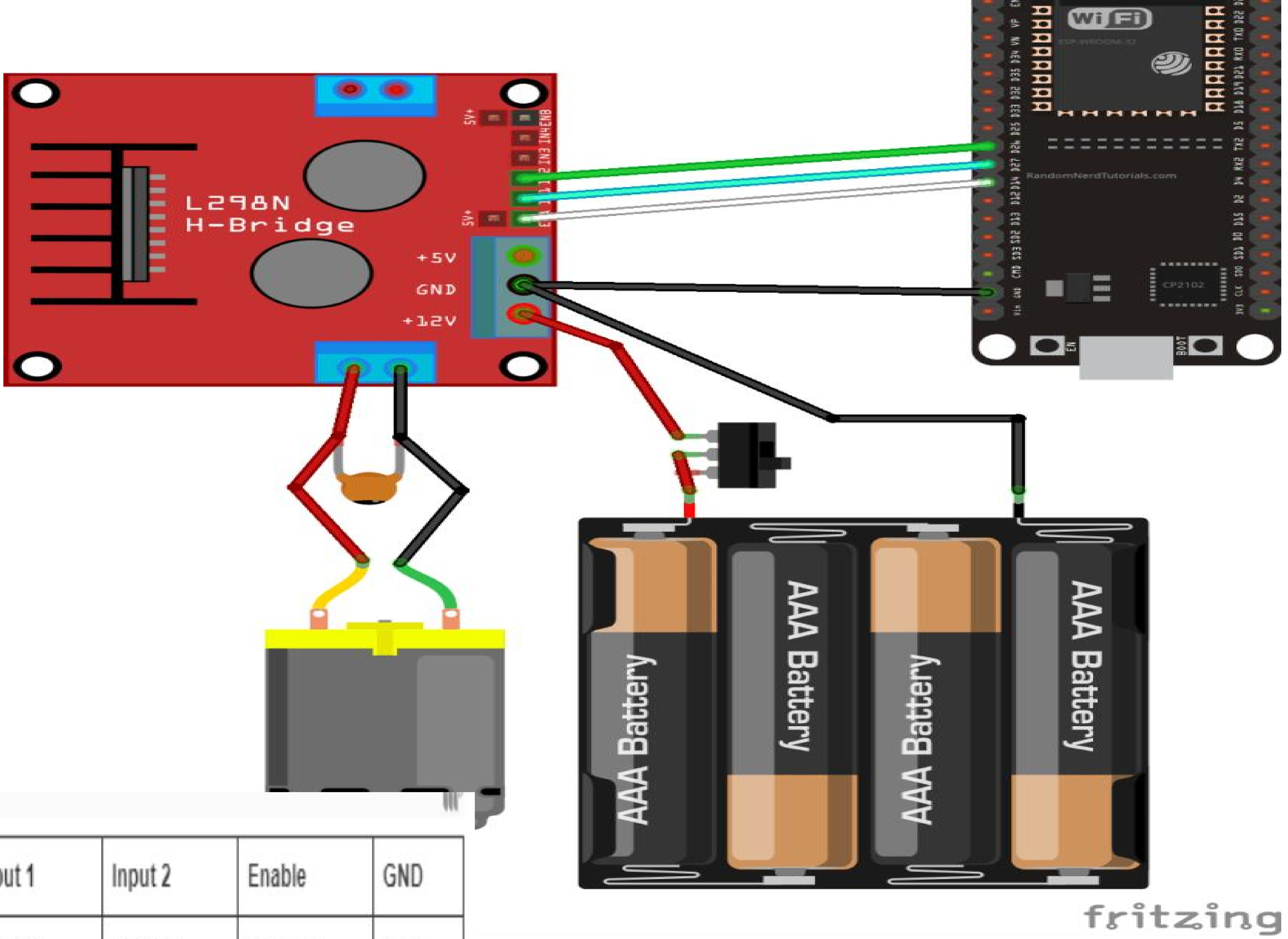
| Direction | Input 1 | Input 2 | Enable 1 |
|---|---|---|---|
| Forward | 0 | 1 | 1 |
| Backwards | 1 | 0 | 1 |
| Stop | 0 | 0 | 0 |

## Controlling 2 DC Motors – ideal to build a robot

- Eg: if you want your robot to move forward, both motors should be rotating forward. To make it go backward, both should be rotating backward.
- To turn the robot in one direction, you need to spin the opposite motor faster. For example, to make the robot turn right, enable the motor at the left, and disable the motor at the right. The following table shows the input pins' state combinations for the robot directions.



| DIRECTION | INPUT 1 | INPUT 2 | INPUT 3 | INPUT 4 |
|---|---|---|---|---|
| Forward | 0 | 1 | 0 | 1 |
| Backward | 1 | 0 | 1 | 0 |
| Right | 0 | 1 | 0 | 0 |
| Left | 0 | 0 | 0 | 1 |
| Stop | 0 | 0 | 0 | 0 |



| LN298N Motor Driver | Input 1 | Input 2 | Enable | GND |
|---|---|---|---|---|
| ESP32 | GPIO 27 | GPIO 26 | GPIO 14 | GND |

# Code: ESP32 with a DC Motor – Control Speed and Direction

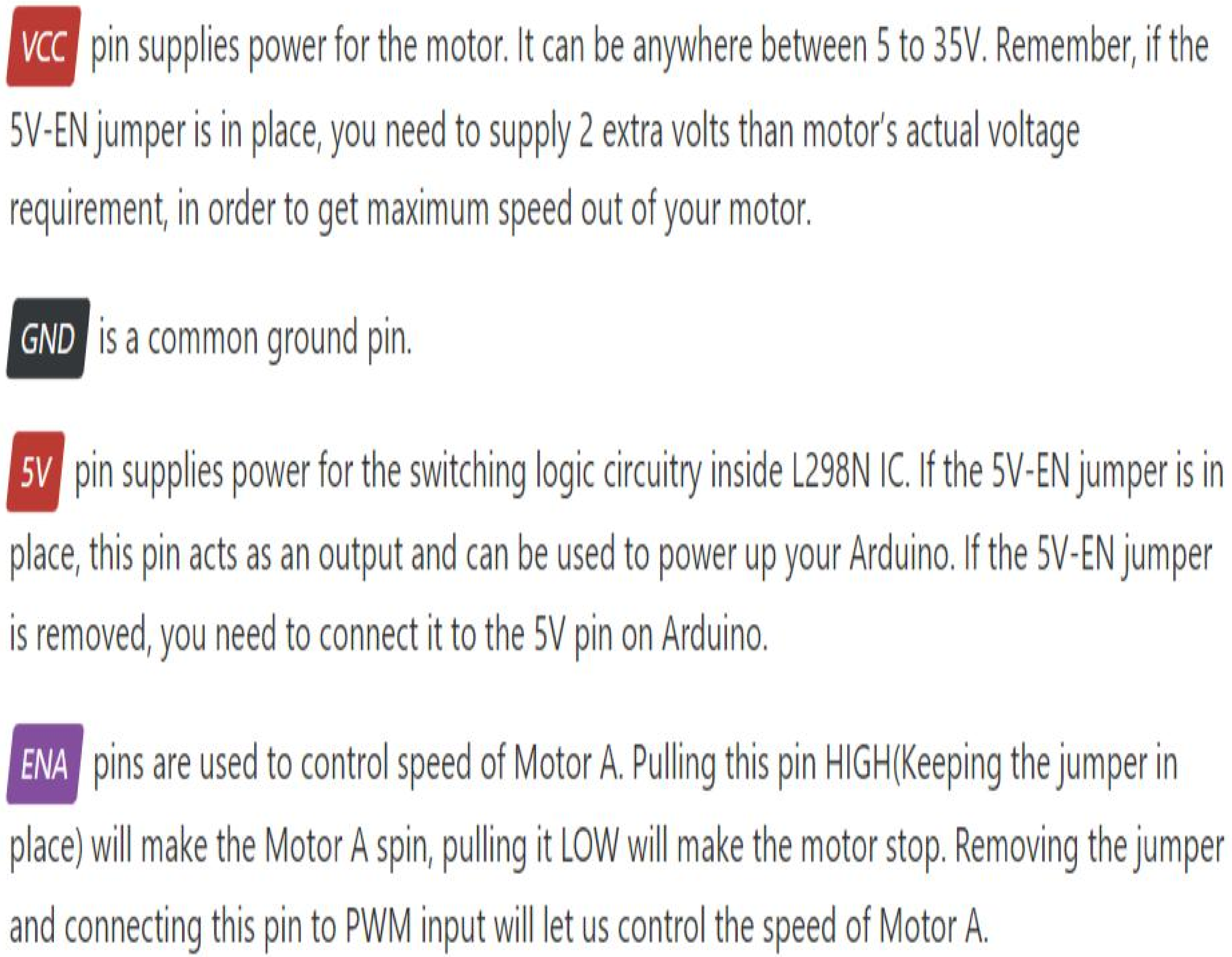
```
// Motor A
int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 14;
// Setting PWM properties
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;
void setup() {
  // sets the pins as outputs:
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(enable1Pin, OUTPUT);

```

```
// configure LED PWM functionalitites
  ledcSetup(pwmChannel, freq, resolution);
   // attach the channel to the GPIO to be controlled
  ledcAttachPin(enable1Pin, pwmChannel);
  Serial.begin(115200);
  // testing
  Serial.print("Testing DC Motor...");
}
void loop() {
  // Move the DC motor forward at maximum speed
  Serial.println("Moving Forward");
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  delay(2000);
```

**// Move DC motor forward with increasing speed**

```
digitalWrite(motor1Pin1, HIGH);
 digitalWrite(motor1Pin2, LOW);
 while (dutyCycle <= 255){
   ledcWrite(pwmChannel, dutyCycle);
   Serial.print("Forward with duty cycle: ");
   Serial.println(dutyCycle);
   dutyCycle = dutyCycle + 5;
   delay(500);
 }
 dutyCycle = 200;
}
```

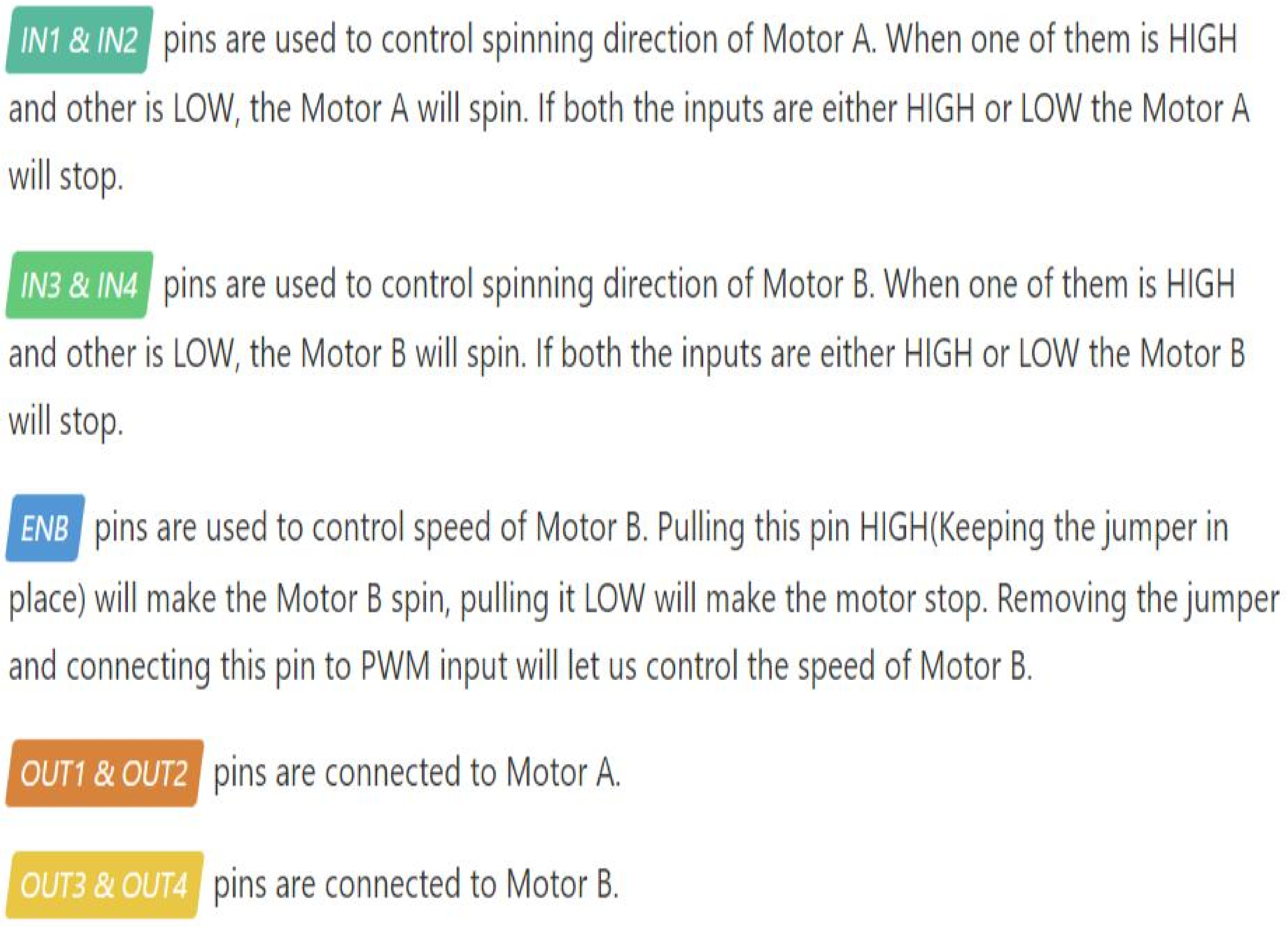


VCC pin supplies power for the motor. It can be anywhere between 5 to 35V. Remember, if the 5V-EN jumper is in place, you need to supply 2 extra volts than motor's actual voltage requirement, in order to get maximum speed out of your motor.

GND is a common ground pin.

5V pin supplies power for the switching logic circuitry inside L298N IC. If the 5V-EN jumper is in place, this pin acts as an output and can be used to power up your Arduino. If the 5V-EN jumper is removed, you need to connect it to the 5V pin on Arduino.

ENA pins are used to control speed of Motor A. Pulling this pin HIGH(Keeping the jumper in place) will make the Motor A spin, pulling it LOW will make the motor stop. Removing the jumper and connecting this pin to PWM input will let us control the speed of Motor A.



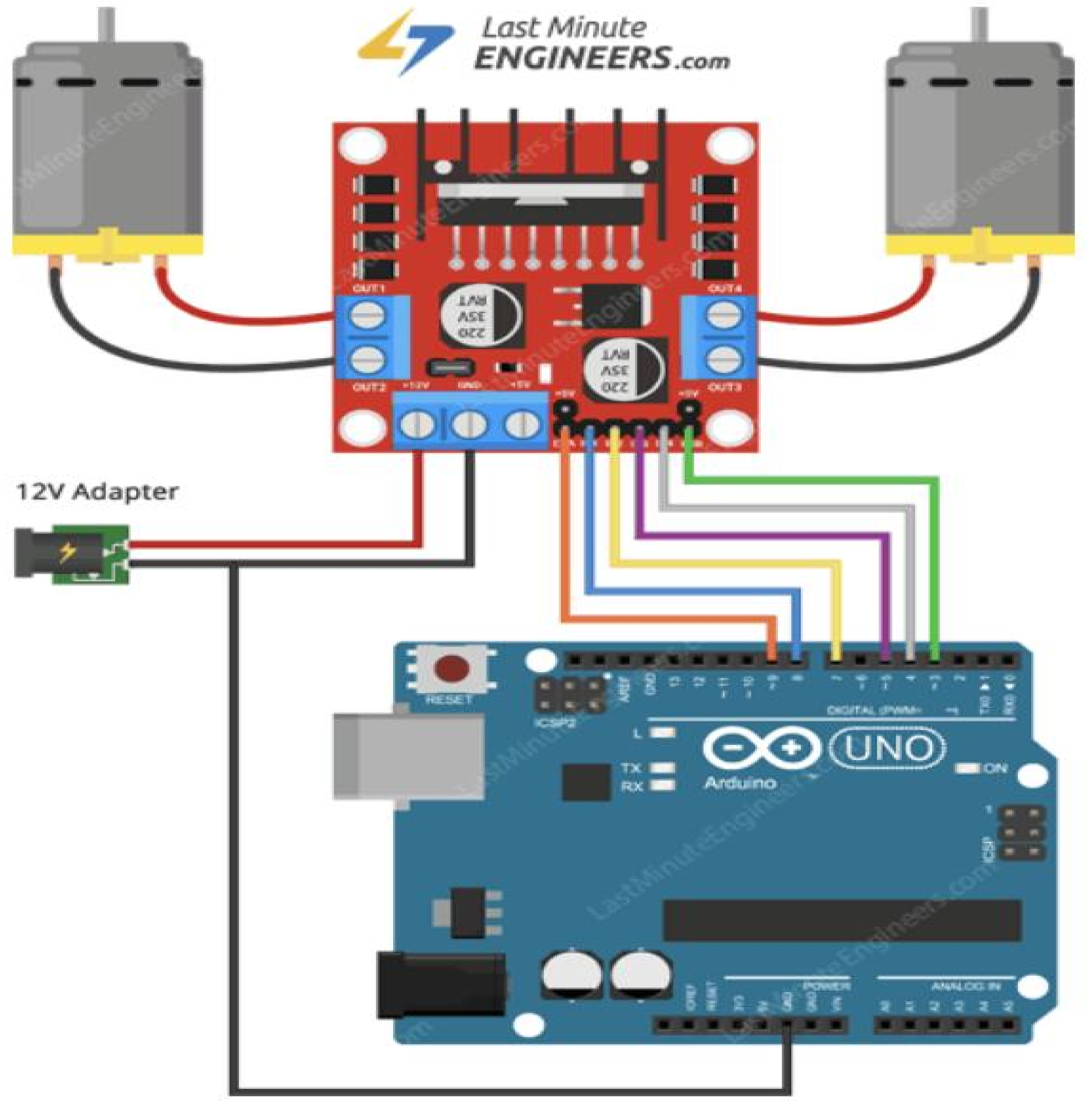
IN1 & IN2 pins are used to control spinning direction of Motor A. When one of them is HIGH and other is LOW, the Motor A will spin. If both the inputs are either HIGH or LOW the Motor A will stop.

IN3 & IN4 pins are used to control spinning direction of Motor B. When one of them is HIGH and other is LOW, the Motor B will spin. If both the inputs are either HIGH or LOW the Motor B will stop.

ENB pins are used to control speed of Motor B. Pulling this pin HIGH(Keeping the jumper in place) will make the Motor B spin, pulling it LOW will make the motor stop. Removing the jumper and connecting this pin to PWM input will let us control the speed of Motor B.

OUT1 & OUT2 pins are connected to Motor A.

OUT3 & OUT4 pins are connected to Motor B.


Last Minute ENGINEERS.com
OUT1
OUT2
OUT3
OUT4
12V Adapter
RESET
UNO
Arduino
DIGITAL (PWM~)
POWER
ANALOG IN