***ESP32 with DC Motor and L298N Motor Driver – Control Speed and Direction***

***Note: there are many ways to control a DC motor. We’ll be using the L298N motor driver. This tutorial is also compatible with similar motor driver modules***.

**ESP32 with Arduino IDE**

We’ll program the ESP32 using Arduino IDE. So, make sure you have the ESP32 add-on installed. Follow the next tutorial if you haven’t already:

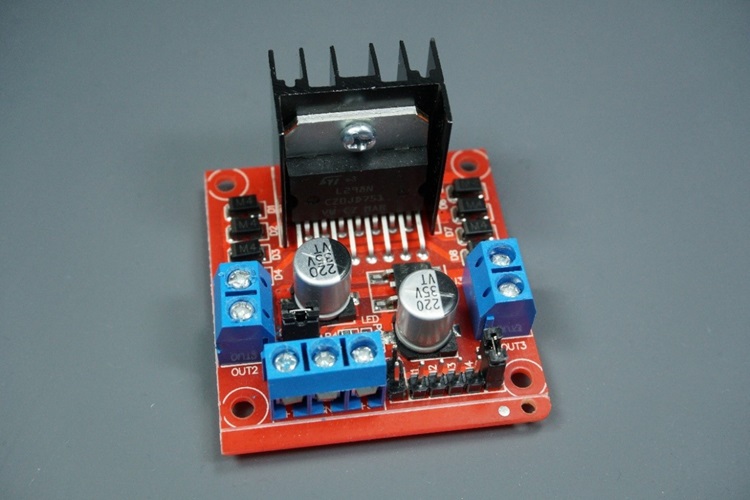
* [Installing ESP32 Board in Arduino IDE 2 (Windows, Mac OS X, Linux)](https://randomnerdtutorials.com/installing-esp32-arduino-ide-2-0/)

**Introducing the L298N Motor Driver**

There are several ways to control a DC motor. The method we’ll use here is suitable for most hobbyist motors, that require 6V or 12V to operate.

We’re going to use the L298N motor driver that can handle up to 3A at 35V. Additionally, it allows us to drive two DC motors simultaneously, which is perfect for building a robot.

The L298N motor driver is shown in the following figure:



**In summary:**

* **+12V**: The +12V terminal is where you should connect the motor’s power supply
* **GND**: power supply GND
* **+5V**: provide 5V if jumper is removed. Acts as a 5V output if jumper is in place
* **Jumper**: jumper in place – uses the motor power supply to power up the chip. Jumper removed: you need to provide 5V to the +5V terminal. If you supply more than 12V, you should remove the jumper

At the bottom right you have four input pins and two enable terminals. The input pins are used to control the direction of your DC motors, and the enable pins are used to control the speed of each motor.

* **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

There are jumper caps on the enable pins by default. You need to remove those jumper caps to control the speed of your motors. Otherwise, they will either be stopped or spinning at the maximum speed.

**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.

**Declaring motor pins**

* First, you define the GPIOs the motor pins are connected to. In this case, Input 1 for motor A is connected to GPIO 27, the Input 2 to GPIO 26, and the Enable pin to GPIO 14.
* int motor1Pin1 = 27;
* int motor1Pin2 = 26;
* int enable1Pin = 14;

In the setup(), you start by setting the motor pins as outputs.

pinMode(motor1Pin1, OUTPUT);

pinMode(motor1Pin2, OUTPUT);

pinMode(enable1Pin, OUTPUT);

**Setting the PWM  properties to control the speed**

As we’ve seen previously, you can control the DC motor speed by applying a PWM signal to the enable pin of the L298N motor driver. The speed will be proportional to the duty cycle. To use PWM with the ESP32, you need to set the PWM signal properties first.

const int freq = 30000;

const int pwmChannel = 0;

const int resolution = 8;

int dutyCycle = 200;