**1. L298N Motor Driver**

We are using DC gearbox motors, also called “TT” motors, which are often found

in two-wheel-drive robots. They are rated for 3 to 12V. We will therefore connect

an external 12V power source to the VS terminal. Because L298N has a voltage

drop of about 2V, the motors will receive 10V and spin at a slightly lower RPM.

But that’s okay.

we need to supply 5V to the logic circuitry of the L298N. We’ll use the on-board

5V regulator to draw 5V from the motor power supply, so keep the 5V-EN jumper

in place.

Now connect the L298N module’s Input and Enable pins (ENA, IN1, IN2, IN3, IN4

and ENB) to the six Arduino digital output pins (9, 8, 7, 5, 4 and 3). Note that both

Arduino output pins 9 and 3 are PWM-enabled

Finally, wire one motor to terminal A (OUT1 and OUT2) and the other to terminal

B (OUT3 and OUT4).

- Declaring the Arduino pins that are connected to the L298N’s control pins

// Motor A connections

int enA = 9;

int in1 = 8;

int in2 = 7;

// Motor B connections

int enB = 3;

int in3 = 5;

int in4 = 4;

- In the setup section

void setup() {

// Set all the motor control pins to outputs

Code

// Turn off motors - Initial state

Code

}

- In the Loop section

void loop() {

directionControl();

delay(1000);

speedControl();

delay(1000);

}

- Implement directionControl() – This function causes both motors

to spin at full speed for two seconds. It then reverses the spinning direction

of the motors and spins for two seconds. Finally, it stops the motors

- Implement speedControl() – This function uses the analogWrite()

function to generate a PWM signal that accelerates both motors from zero

to maximum speed before decelerating them back to zero. Finally, it stops

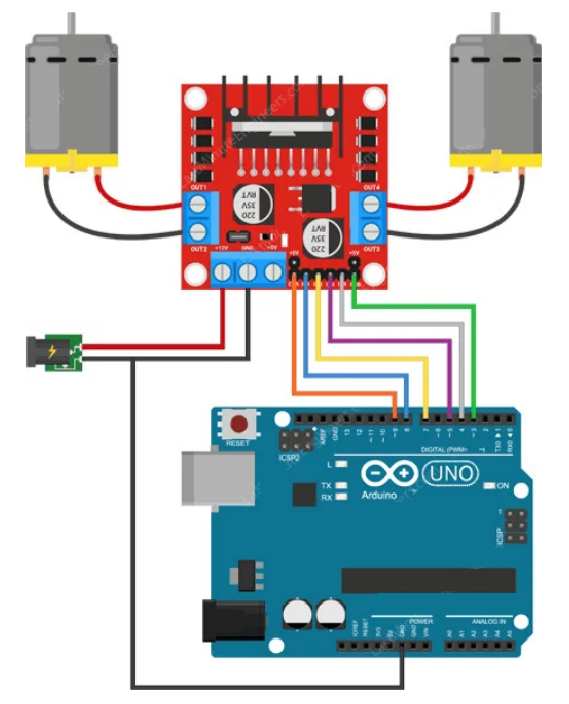
the motors

- Let's create a few methods to control the motors. These methods include:

forward(int speed), backward(int speed), left(int speed),

right(int speed), and stop()

Circuit:



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);

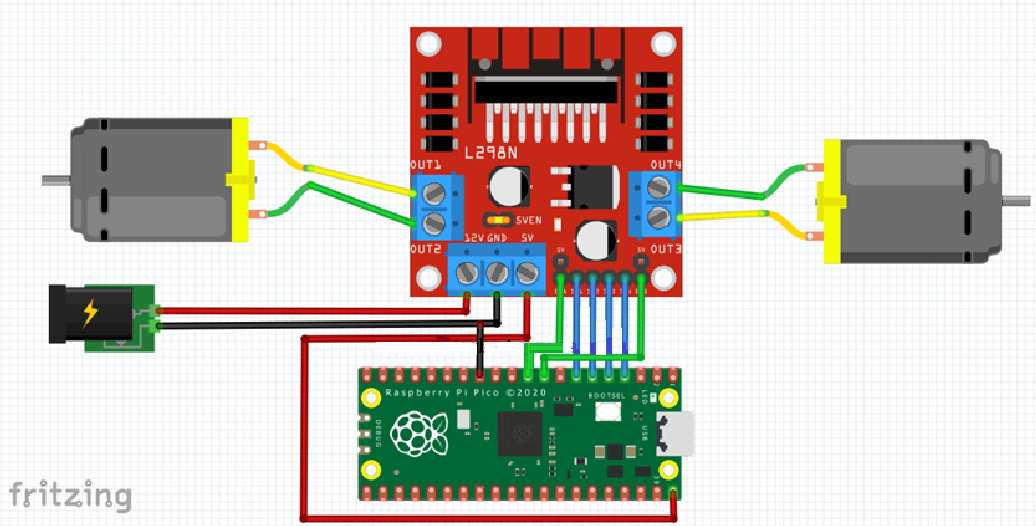
}

**2. L298N Raspberry Pi Wiring Diagram**

For this application, you’ll connect the L298N signal pins as follows:

You’ll power your Raspberry Pi using a 5V 2A battery pack

Circuit



Code:

from machine import Pin

import utime

m1 = Pin(5, Pin.OUT)

m2 = Pin(4, Pin.OUT)

m3 = Pin(3, Pin.OUT)

m4 = Pin(2, Pin.OUT)

en1 = Pin(6, Pin.OUT)

en2 = Pin(7, Pin.OUT)

en1(1) # motor 1 enable, set value 0 to disable

en2(1) # motor 2 enable, set value 0 to disable

while True:

#Both Motor in forward direction

m1(1)

m2(0)

m3(1)

m4(0)

utime.sleep(1)

#Both Motor in stop position

m1(0)

m2(0)

m3(0)

m4(0)

utime.sleep(1)

#Both Motor in Reverse direction

m1(0)

m2(1)

m3(0)

m4(1)

utime.sleep(1)

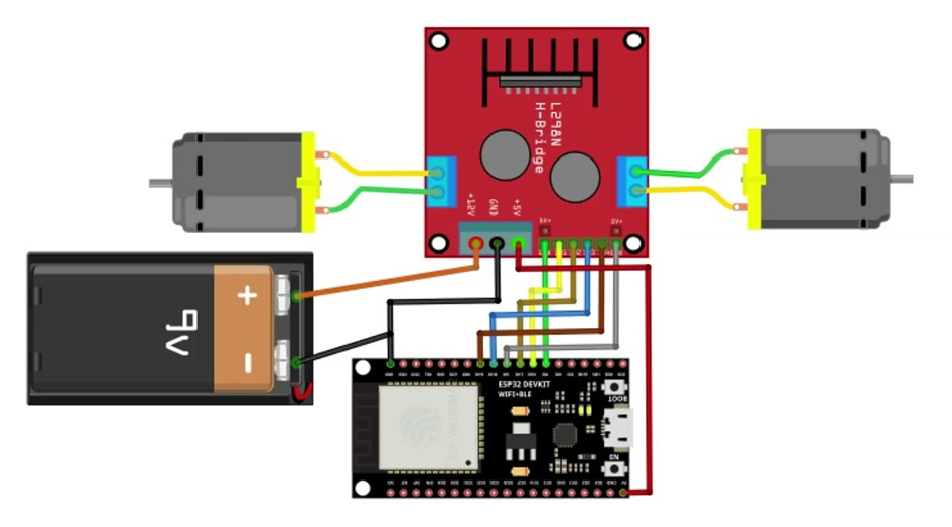
**3. NodeMCU ESP32 L298N Wiring Diagram**

The NodeMCU requires an additional Voltage Regulator (L7805 – 5V 1.5A), an

Electrolytic Capacitor (1uF/50V), and a Ceramic Capacitor (100nF – 0603). The

signal pins between the NodeMCU and L298N are given as follows:

Circuit:



Code:

// Định nghĩa chân

#define ENA 4

#define IN1 16

#define IN2 17

#define IN3 18

#define IN4 19

#define ENB 5

void setup() {

// Khởi tạo chân output

pinMode(ENA, OUTPUT);

pinMode(IN1, OUTPUT);

pinMode(IN2, OUTPUT);

pinMode(IN3, OUTPUT);

pinMode(IN4, OUTPUT);

pinMode(ENB, OUTPUT);

// Bật enable cho cả 2 motor

digitalWrite(ENA, HIGH);

digitalWrite(ENB, HIGH);

}

void loop() {

// Cả hai motor chạy tiến

digitalWrite(IN1, HIGH);

digitalWrite(IN2, LOW);

digitalWrite(IN3, HIGH);

digitalWrite(IN4, LOW);

delay(1000);

// Dừng cả hai motor

digitalWrite(IN1, LOW);

digitalWrite(IN2, LOW);

digitalWrite(IN3, LOW);

digitalWrite(IN4, LOW);

delay(1000);

// Cả hai motor chạy lùi (nếu muốn)

digitalWrite(IN1, LOW);

digitalWrite(IN2, HIGH);

digitalWrite(IN3, LOW);

digitalWrite(IN4, HIGH);

delay(1000);

}