

# Connection guide for Raspberry Pi and Arduino

Trieu An Do

November 25, 2025

BETTER FINISH IT BEFORE CHRISTMAS

## Contents

<b>1</b>	<b>Motor</b>	<b>2</b>
1.1	Material . . . . .	2
1.2	Connection with Raspberry Pi . . . . .	2
1.2.1	Connection scheme . . . . .	2
1.2.2	Real connection image . . . . .	2
1.2.3	Code example . . . . .	3
1.2.4	Remark for connection with RBP . . . . .	3
1.3	Connection with Arduino . . . . .	3
1.3.1	Connection scheme . . . . .	3
1.3.2	Real connection image . . . . .	4
1.3.3	Code example . . . . .	4
1.3.4	Remark for conection with Arduino . . . . .	4
<b>2</b>	<b>NFC Reader(PN532 V3)</b>	<b>4</b>
2.1	Material . . . . .	4
2.2	Connection with RBP . . . . .	4
2.2.1	Multi-reader connection - I2C mode . . . . .	4
2.3	Connection with Arduino . . . . .	5
<b>3</b>	<b>Sensor</b>	<b>5</b>
<b>4</b>	<b>Compressor</b>	<b>5</b>

# 1 Motor

## 1.1 Material

Motor: 9V, 24V, speed changing by PWM

Raspberry Pi: Pi 4 Model B, Pi 5/Arduino Uno Rev 3

GPIO cables

Motor driver L298N

L298N Pinout:

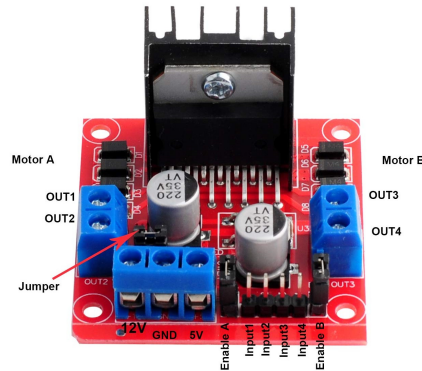


Figure 1: L298N Pin Layout

1 L298N can controll maximum 2 independent motors.

Input1 controls OUT1 and so on.

Control by H-bridge:

Low	Low	Stop
High	Low	Clockwise/Anti-Clockwise
High	Low	Clockwise/Anti-Clockwise
High	High	Brake

## 1.2 Connection with Raspberry Pi

### 1.2.1 Connection scheme

The connection scheme to connect motor to rbp using l298n

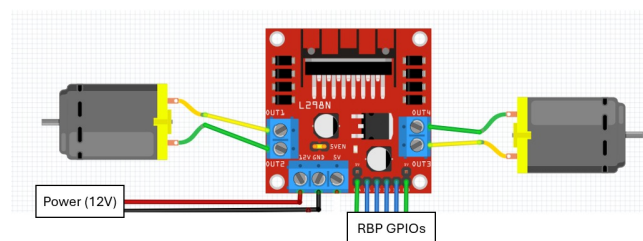


Figure 2: Connection scheme

### 1.2.2 Real connection image

The image of real connection

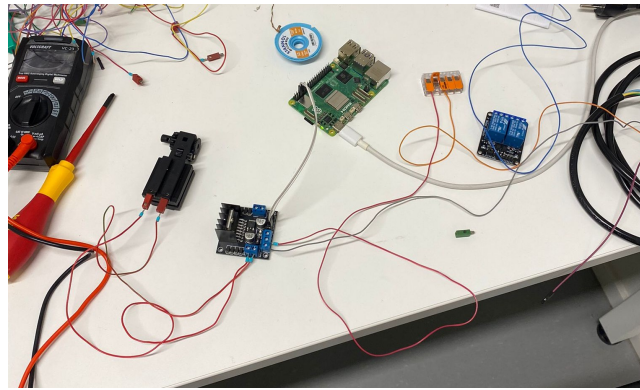


Figure 3: Real connection image

### 1.2.3 Code example

```
import RPi.GPIO as GPIO
import time

# Pin setup
IN1 = 17
IN2 = 27
ENA = 22 # PWM pin

GPIO.setmode(GPIO.BCM)
GPIO.setup(IN1, GPIO.OUT)
GPIO.setup(IN2, GPIO.OUT)
GPIO.setup(ENA, GPIO.OUT)

# Setup PWM at 1000 Hz
pwm = GPIO.PWM(ENA, 1000)
pwm.start(0) # start with speed 0

def motor_forward(speed):
    GPIO.output(IN1, GPIO.HIGH)
    GPIO.output(IN2, GPIO.LOW)
    pwm.ChangeDutyCycle(speed)

def motor_backward(speed):
    GPIO.output(IN1, GPIO.LOW)
    GPIO.output(IN2, GPIO.HIGH)
    pwm.ChangeDutyCycle(speed)

def motor_stop():
    GPIO.output(IN1, GPIO.LOW)
    GPIO.output(IN2, GPIO.LOW)
    pwm.ChangeDutyCycle(0)

try:
    print("Motor forward")
    motor_forward(60) # 60% speed
    time.sleep(2)

    print("Motor backward")
    motor_backward(60) # 60% speed
    time.sleep(2)

    print("Stop")
    motor_stop()
    time.sleep(1)

except KeyboardInterrupt:
    pass

finally:
    pwm.stop()
    GPIO.cleanup()
```

### 1.2.4 Remark for connection with RBP

- 1 L298N occupies at least 4 GPIOs on RBP (for 2 motors).
- L298N can be powered by RBP or external power source.

## 1.3 Connection with Arduino

### 1.3.1 Connection scheme

The connection scheme to connect motors to arduino uno rev 3 using l298n is the same as connection with RBP

### 1.3.2 Real connection image

The image of real connection

Figure 4: Real connection image

### 1.3.3 Code example

```
// Motor control pins
int IN1 = 7;
int IN2 = 8;
int ENA = 9;    // must be a PWM pin (~)

void setup() {
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);

  // Start with motor stopped
  analogWrite(ENA, 0);
}

void motorForward(int speed) {
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, speed);    // 0-255
}

void motorBackward(int speed) {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);

  analogWrite(ENA, speed);    // 0-255
}

void motorStop() {
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, LOW);
  analogWrite(ENA, 0);
}

void loop() {
  // Forward for 2 seconds
  motorForward(150);    // about 60% speed
  delay(2000);

  // Backward for 2 seconds
  motorBackward(150);
  delay(2000);

  // Stop for 1 second
  motorStop();
  delay(1000);
}
```

### 1.3.4 Remark for conection with Arduino

- 1 L298N occupies at least 4 GPIOs on Arduino (for 2 motors).
- Recommend connection: Motor → L298N → Arduino → RBP

## 2 NFC Reader(PN532 V3)

### 2.1 Material

### 2.2 Connection with RBP

#### 2.2.1 Multi-reader connection - I2C mode

If the readers have different I2C address, then we only need one bus, but if they have same address, and that address can't be changed, then here come some potential solutions:

**I2C Multiplexer ( max 8 readers):** [https://www.adafruit.com/product/2717?srsId=AfmB0orhzn\\_5ry0sL1W0VA1TeGnU4hh1WHF7dkA4JKQbefiaHu6vNRcs](https://www.adafruit.com/product/2717?srsId=AfmB0orhzn_5ry0sL1W0VA1TeGnU4hh1WHF7dkA4JKQbefiaHu6vNRcs)

<https://learn.adafruit.com/adafruit-tca9548a-1-to-8-i2c-multiplexer-breakout>

**Create new buses:** <https://www.instructables.com/Raspberry-PI-Multiple-I2c-Devices/>

**1 I2C, 1 SPI:** ChatGPT, not sure, need testing.

### **2.3 Connection with Arduino**

## **3 Sensor**

## **4 Compressor**