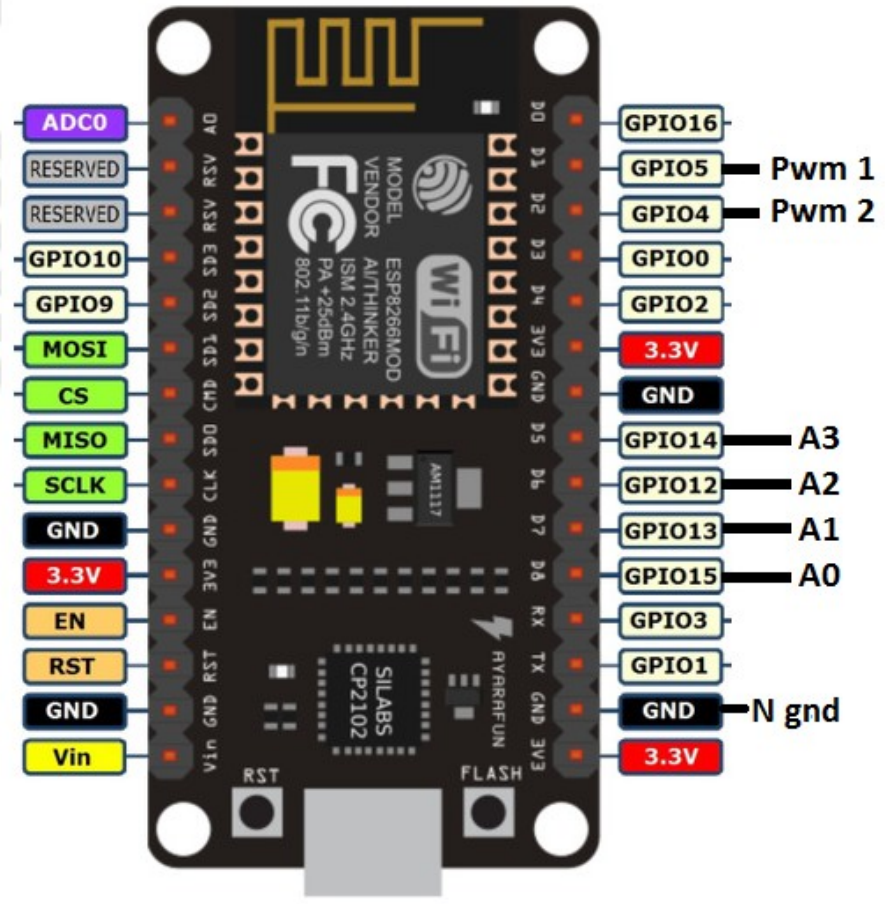
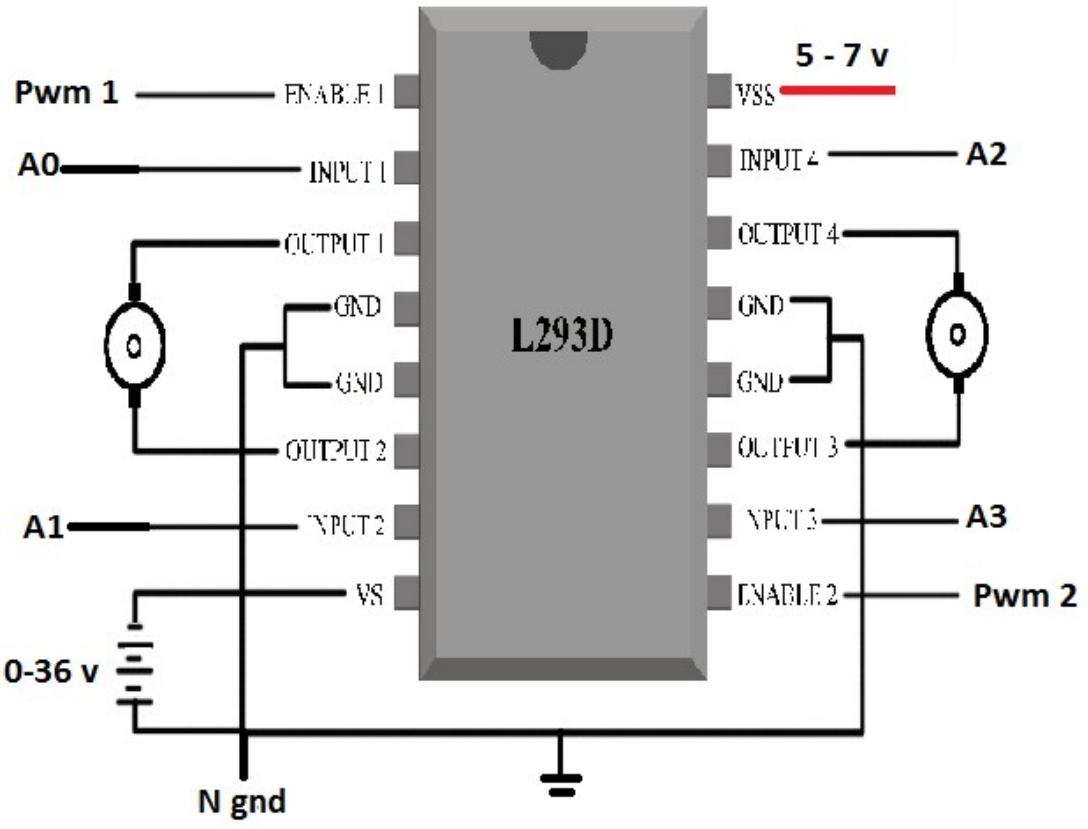
Ejercicios con ESP32



www.microcontroller-project.com





```
// 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...");
}
```

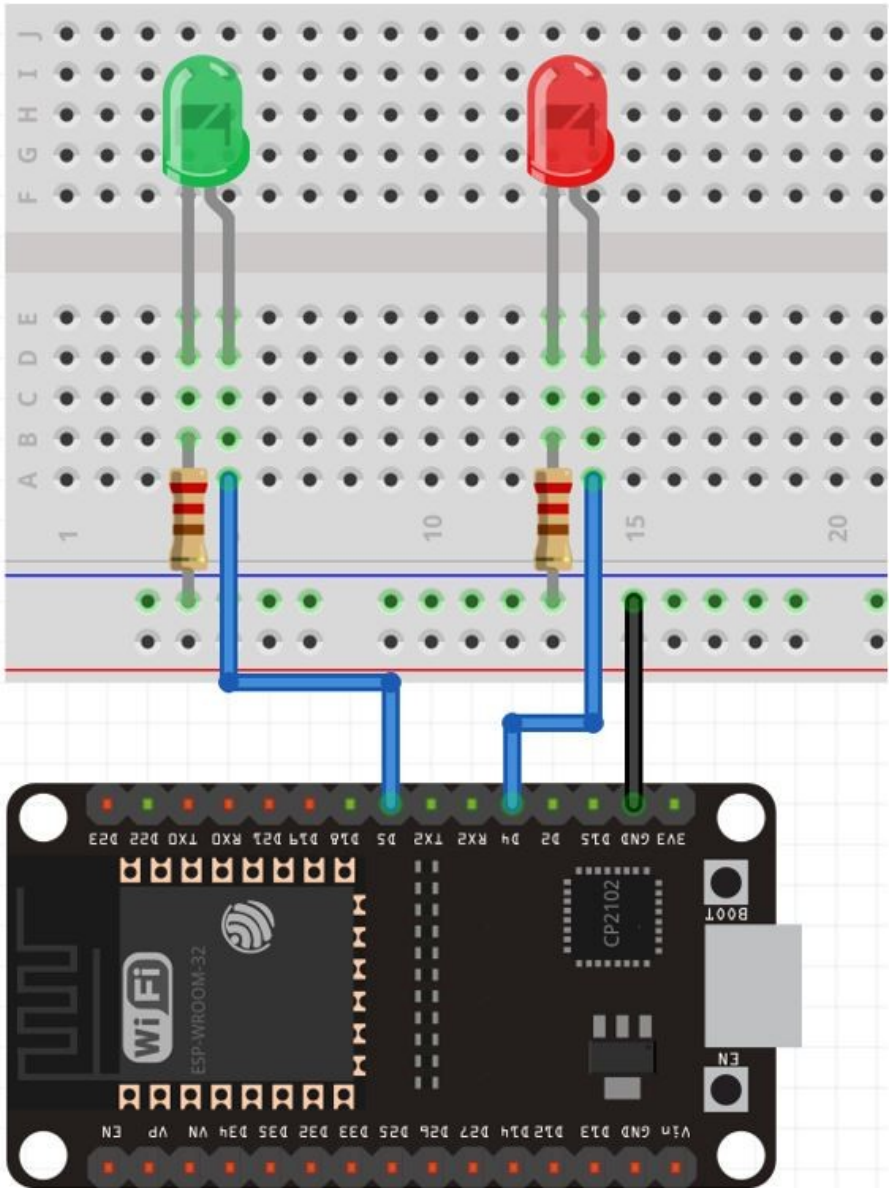
```
void loop() {
  // Move the DC motor forward at maximum speed
  Serial.println("Moving Forward");
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, HIGH);
  delay(2000);


  // Stop the DC motor
  Serial.println("Motor stopped");
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, LOW);
  delay(1000);


  // Move DC motor backwards at maximum speed
  Serial.println("Moving Backwards");
  digitalWrite(motor1Pin1, HIGH);
  digitalWrite(motor1Pin2, LOW);
  delay(2000);


  // Stop the DC motor
  Serial.println("Motor stopped");
  digitalWrite(motor1Pin1, LOW);
  digitalWrite(motor1Pin2, LOW);
  delay(1000);


  // 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;
}
```



https://microcontrollerslab.com/esp32-controller-android-mit-app-inventor/

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include <ESPmDNS.h>

MDNSResponder mdns;

const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

WebServer server(80);
String webpage = "";

int led1 = 4;
int led2 = 5;
```

```
void setup(void) {

  webpage += "<h1>ESP32 Web Server</h1><p>LED at GPIO4 <a
href=\"led1ON\"><button>ON</button></a> <a
href=\"led1OFF\"><button>OFF</button></a></p>";
  webpage += "<p>LED at GPIO5 <a href=\"led2ON\"><button>ON</button></a> <a
href=\"led2OFF\"><button>OFF</button></a></p>";

  pinMode(led1, OUTPUT);
  digitalWrite(led1, LOW);
  pinMode(led2, OUTPUT);
  digitalWrite(led2, LOW);
  delay(1000);
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.println("");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
```

```
Serial.println("");
 Serial.print("Connected to ");
 Serial.println(ssid);
 Serial.print("IP address: ");
 Serial.println(WiFi.localIP());
```

```
 if (mdns.begin("esp")) {
   Serial.println("MDNS responder started");
 }
 server.on("/", []() {
   server.send(200, "text/html", webpage);
 });

 server.on("/led1ON", []() {
   server.send(200, "text/html", webpage);
   digitalWrite(led1, HIGH);
   delay(1000);
 });

 server.on("/led1OFF", []() {
   server.send(200, "text/html", webpage);
   digitalWrite(led1, LOW);
   delay(1000);
 });

 server.on("/led2ON", []() {
   server.send(200, "text/html", webpage);
   digitalWrite(led2, HIGH);
   delay(1000);
 });


 server.on("/led2OFF", []() {
   server.send(200, "text/html", webpage);
   digitalWrite(led2, LOW);
   delay(1000);
 });
 server.begin();
 Serial.println("HTTP server started");
}
void loop(void) {
 server.handleClient();
}
```