



# **Ja-Bots.com**

## **Manual de Usuario**

Tarjeta ESPBot

Versión 1.0 | Noviembre 2025

[www.ja-bots.com](http://www.ja-bots.com)

## 1. Componentes Principales de la Tarjeta ESPBot

La **tarjeta ESPBot** está diseñada como una plataforma compacta para el desarrollo de un seguidor de línea. Su arquitectura se basa en tres elementos esenciales que operan como un pequeño trío mecánico-electrónico, cada uno con un papel bien definido dentro del sistema.

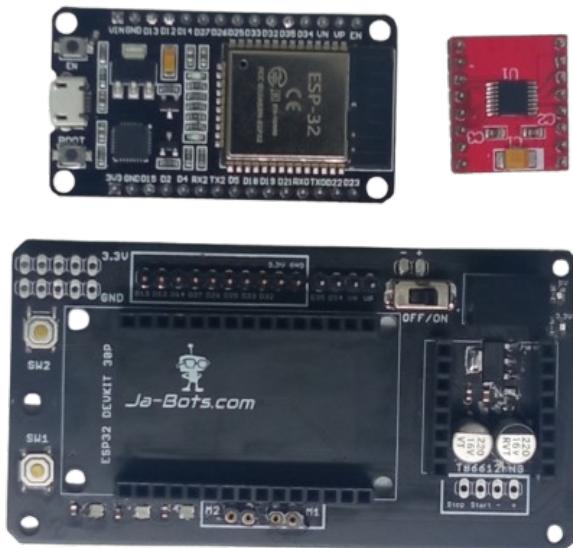


Figura 1: Componentes Principales

### 1.1. Tarjeta Base

La *tarjeta base* actúa como el soporte físico y eléctrico del módulo. En ella se distribuyen las pistas, conectores y pads que permiten montar y enlazar todos los componentes del sistema. Su diseño optimiza el ruteo para minimizar interferencias y asegurar una alimentación limpia hacia los módulos lógicos y de potencia.



Figura 2: Tarjeta Base

### 1.2. Módulo ESP32

El **ESP32** funciona como el centro de control. Desde este microcontrolador se ejecutan los algoritmos de lectura de sensores, toma de decisiones y control de motores. Su tamaño reducido y su capacidad de programación flexible lo convierten en el núcleo ideal para sistemas autónomos compactos.

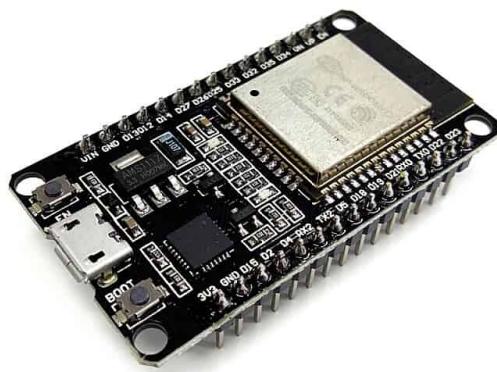


Figura 3: ESP32

### 1.3. Driver de Motores TB6612FNG

El **TB6612FNG** se encarga de traducir las órdenes del microcontrolador en señales de potencia para los motores. Este driver de doble canal ofrece una entrega de corriente estable, protección interna contra sobrecalentamiento y una respuesta rápida, adecuada para maniobras precisas en un seguidor de línea.



Figura 4: TB6612FNG

#### 1.4. Elementos Integrados en la Tarjeta ESPBot

La tarjeta ESPBot incorpora un conjunto de componentes diseñados para ofrecer robustez, flexibilidad y seguridad durante la operación del seguidor de línea. A continuación se describen los elementos más relevantes integrados en la tarjeta.

##### 1.4.1. Regulador de Tensión

La tarjeta incluye un **regulador de tensión** de alta capacidad, capaz de soportar entradas de hasta **30 V**. Este módulo asegura una alimentación estable hacia el microcontrolador, el driver de motores y los periféricos conectados, manteniendo la eficiencia y evitando sobrecargas.

##### 1.4.2. Conectores

Se incorporan **conectores tipo headers** para una conexión segura y firme de:

- La batería principal.
- Dos motores de tracción.

Estos conectores facilitan el montaje y permiten reemplazos rápidos en caso de mantenimiento.

##### 1.4.3. Elementos de Interacción

La tarjeta dispone de:

- **Dos pulsadores programables**, configurables para acciones como arranque, calibración o cambio de modo.
- **Tres indicadores LED programables**, útiles para mostrar estados del sistema o diagnósticos básicos.

#### 1.4.4. Pads de Expansión

Para ampliar las capacidades del sistema, se incluyen:

- Pads para un **módulo de inicio remoto**, permitiendo activar o desactivar el robot a distancia.
- **Dos pads de uso libre**, destinados a futuras expansiones o integraciones personalizadas.

#### 1.4.5. Control y Protección

La tarjeta integra además:

- Un **interruptor ON/OFF** de fácil acceso para controlar la alimentación general.
- **Protección contra sobrecorriente**, evitando daños en casos de consumo excesivo.
- **Protección contra inversión de polaridad**, salvaguardando el sistema ante conexiones erróneas de la batería.

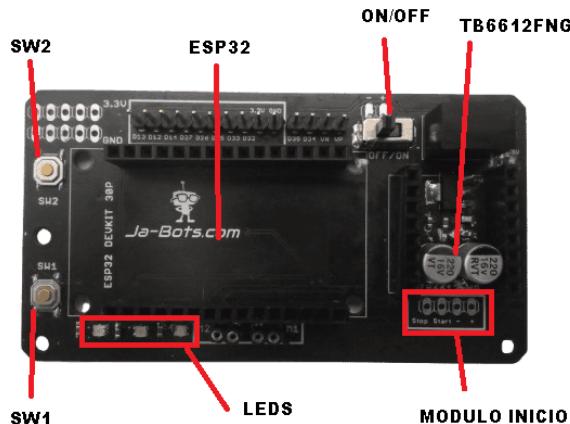


Figura 5: Partes de la Tarjeta ESPBot

## 2. Ubicación de los Módulos en la Tarjeta ESPBot

En esta sección se describen los pasos para colocar cada módulo en su posición correspondiente dentro de la tarjeta ESPBot. La correcta orientación de cada componente garantiza un funcionamiento estable y evita errores de montaje.

### 2.1. Instalación del Driver TB6612FNG

El primer elemento que debe colocarse es el **TB6612FNG**. Su posición está marcada en la tarjeta y es importante respetar la orientación del encapsulado. El pin **VM** debe alinearse exactamente con la referencia indicada en la serigrafía de la tarjeta, tal como se aprecia en la imagen del manual. Ubicar este pin correctamente asegura que la alimentación del módulo de potencia llegue a los pines adecuados y evita fallas o sobrecarga del driver.

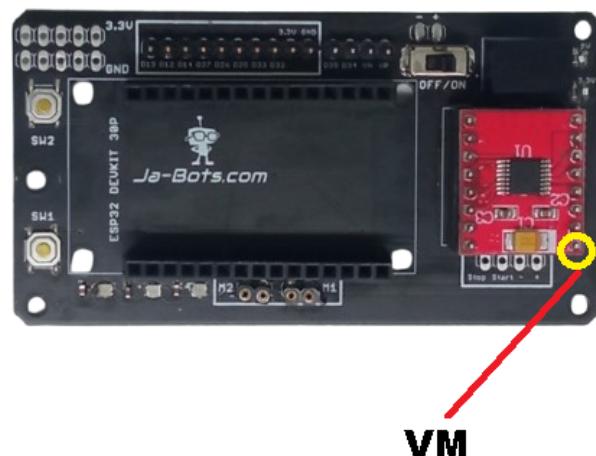


Figura 6: Ubicación TB6612FNG

## 2.2. Ubicación del ESP32

El **ESP32** se instala en su zócalo correspondiente, siguiendo la orientación definida en la tarjeta. El **puerto USB debe quedar dirigido hacia el borde externo**, lo que facilita la conexión del cable USB durante la programación y depuración. Una orientación incorrecta puede desplazar las líneas de alimentación y señal, por lo que se recomienda verificar dos veces su posición antes de soldar o insertar el módulo.

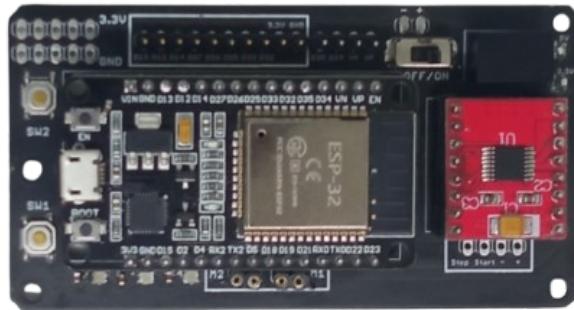


Figura 7: Ubicación ESP32

### 2.3. Verificación General

Antes de continuar con el montaje de otros componentes, es aconsejable revisar que ambos módulos:

- Estén alineados con la serigrafía.
- No presenten pines doblados.
- Coincidan con la orientación indicada en el diseño original de la tarjeta.

Esta verificación evita errores que pueden resultar difíciles de corregir una vez concluido el montaje.

## 3. Programación Inicial del ESP32

Con los módulos instalados, procedemos a cargar el primer programa de prueba en el **ESP32**. Este paso permite verificar el correcto funcionamiento de los pulsadores y los indicadores LED integrados en la tarjeta ESPBot.

### 3.1. Conexión y Preparación

Conectamos un **cable USB** entre el ESP32 y el equipo de cómputo. Para este tutorial utilizaremos el código **PruebaPulsadores\_ESPBot**, disponible en la página oficial Tarjeta Seguidor de Línea ESPBot

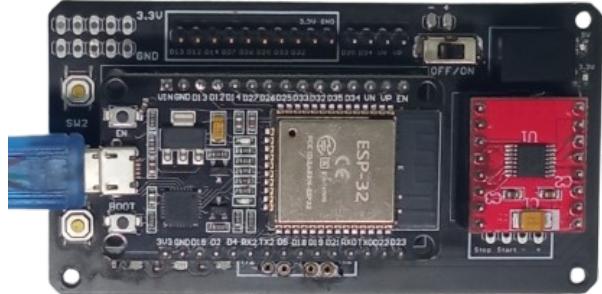


Figura 8: Conexión del USB

### 3.2. Carga del Programa

Abrimos el **Arduino IDE** y seguimos estos pasos:

1. Copiamos y pegamos el código de **PruebaPulsadores\_ESPBot** en un nuevo proyecto, este se encuentra en la pagina de la tienda .

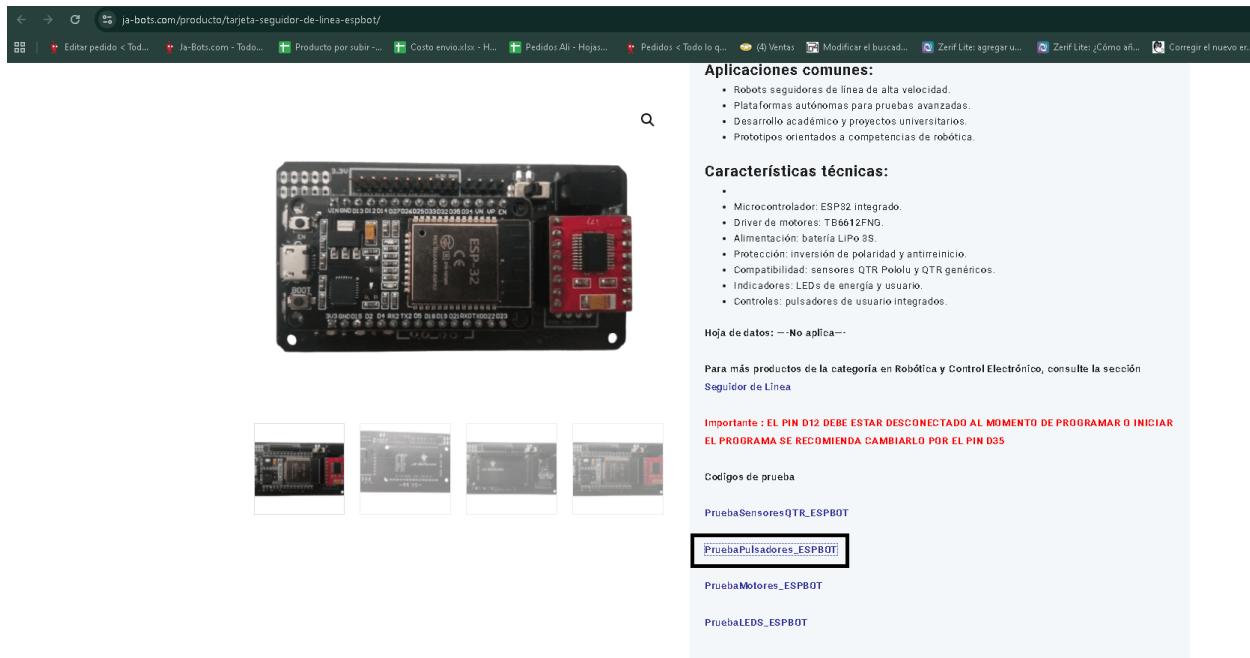


Figura 9: Códigos de Prueba en la Página de la Tienda

```

#include <Adafruit_NeoPixel.h>

#define LED_PIN 5
#define NUM_LEDS 3
#define SW1 22
#define SW2 23

Adafruit_NeoPixel leds(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);

int ledSeleccionado = 0;
int colorIndex = 0;

uint32_t colores[] = {
    leds.Color(255, 255, 0), // Amarillo
    leds.Color(0, 0, 255), // Azul
    leds.Color(255, 0, 0), // Rojo
    leds.Color(0, 255, 0), // Verde
    leds.Color(255, 255, 255) // Blanco
};
const int numColores = sizeof(colores) / sizeof(colores[0]);

bool lastSW1 = HIGH;
bool lastSW2 = HIGH;

void setup() {
    leds.begin();
    leds.show();

    pinMode(SW1, INPUT_PULLUP);
    pinMode(SW2, INPUT_PULLUP);
}

void loop() {
    bool sw1 = digitalRead(SW1);
    bool sw2 = digitalRead(SW2);

    // Cambiar color
    if (sw1 == LOW && lastSW1 == HIGH) {
        colorIndex = (colorIndex + 1) % numColores;
        leds.setPixelColor(ledSeleccionado, colores[colorIndex]);
        leds.show();
        delay(200); // Antirrebote
    }

    // Cambiar LED seleccionado
    if (sw2 == LOW && lastSW2 == HIGH) {
        ledSeleccionado = (ledSeleccionado + 1) % NUM_LEDS;
        delay(200); // Antirrebote
    }

    lastSW1 = sw1;
    lastSW2 = sw2;
}

```

Figura 10: Archivo TXT , Copiar y pegar el código

2. En el menú *Herramientas*, seleccionamos la **placa ESP32**.
3. Elegimos el **puerto serial** correspondiente al dispositivo conectado.

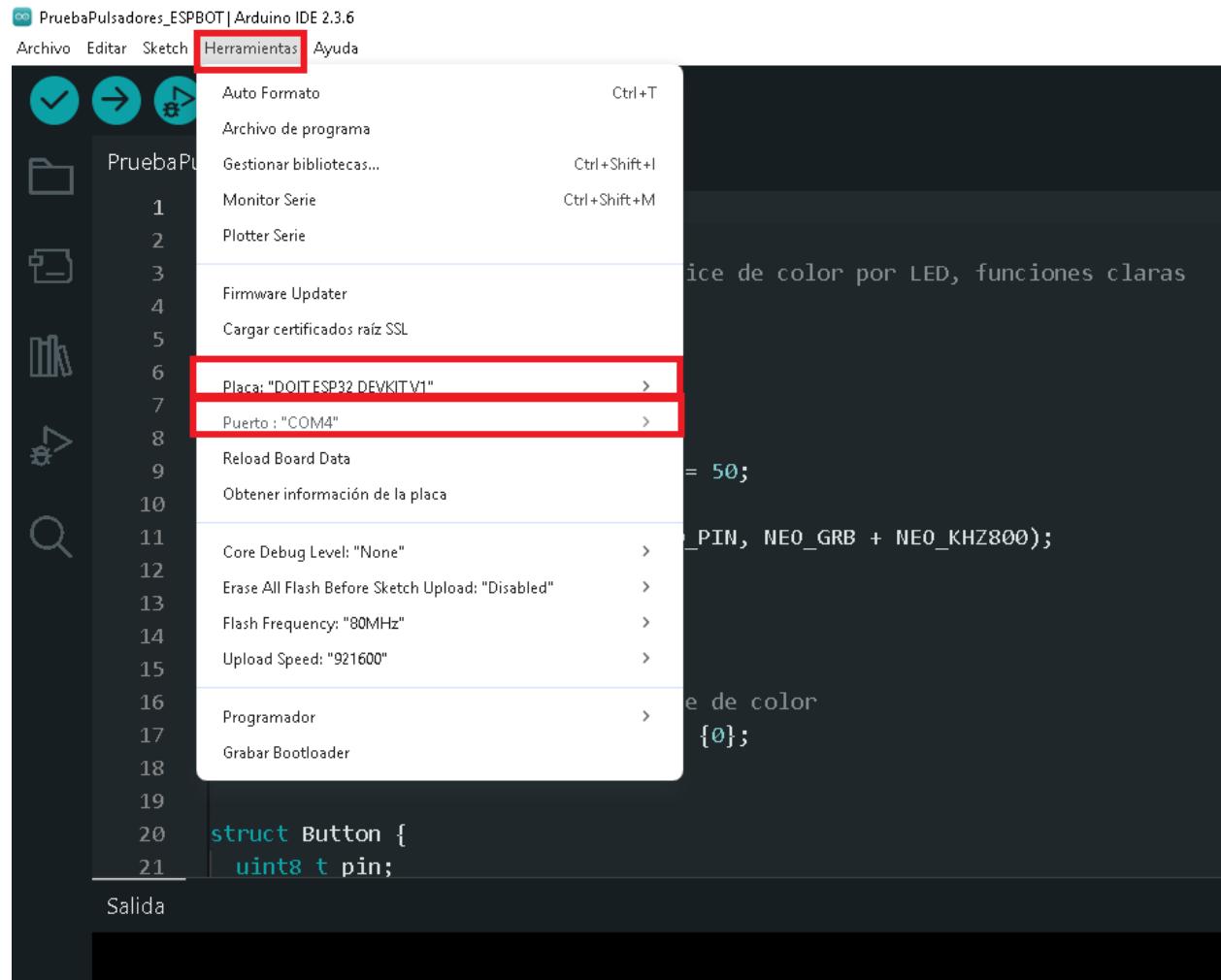


Figura 11: Selección del ESP32 y El puerto COM

4. Presionamos el botón **Subir** para compilar y transferir el programa al microcontrolador.

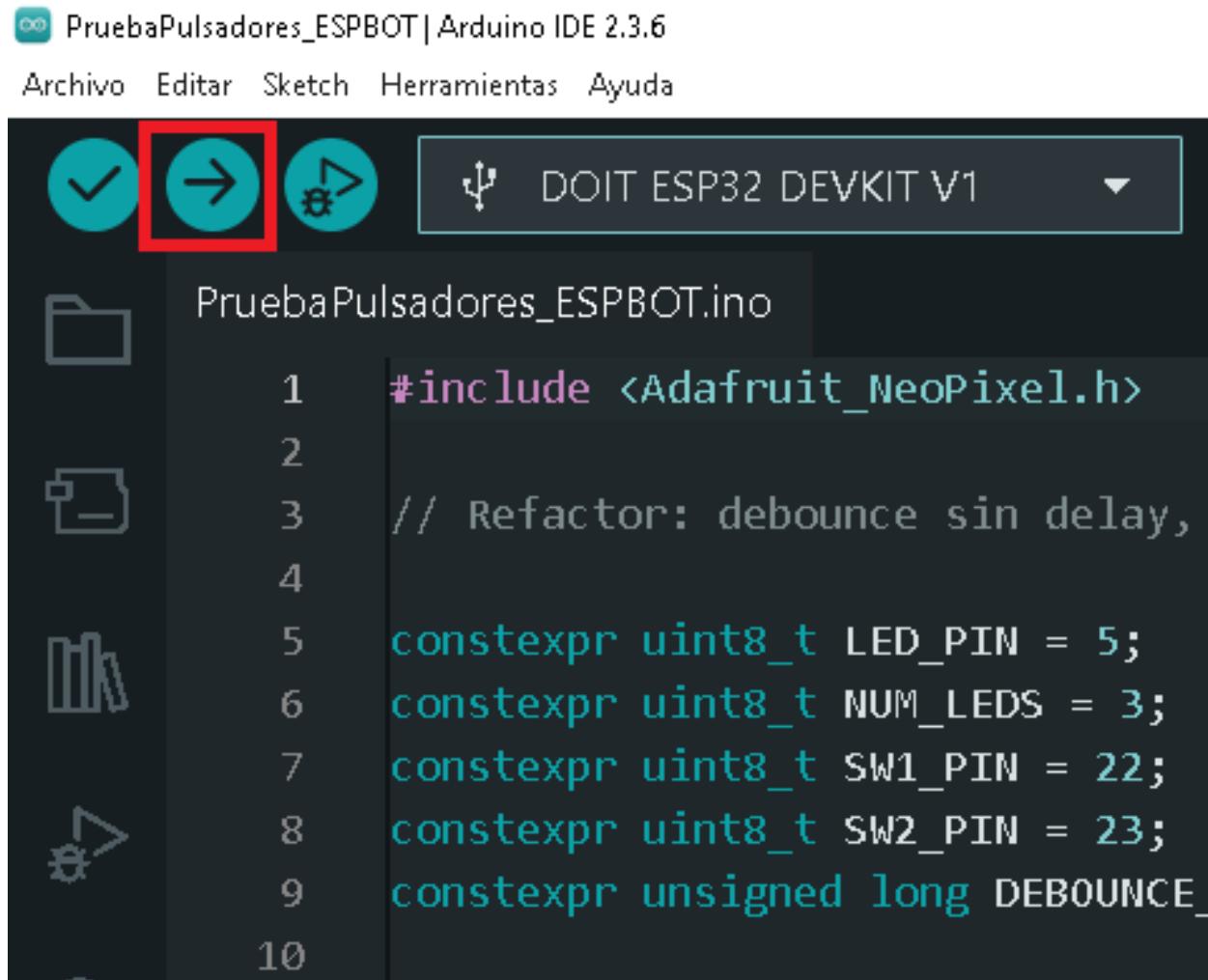


Figura 12: Subir el Programa al ESP32

Si la carga es exitosa, el ESP32 reiniciará automáticamente y comenzará a ejecutar el programa.

### 3.3. Conexión de la Batería

Conectamos la batería con la polaridad que se muestra a continuación , se recomienda una batería LiPo 2S.

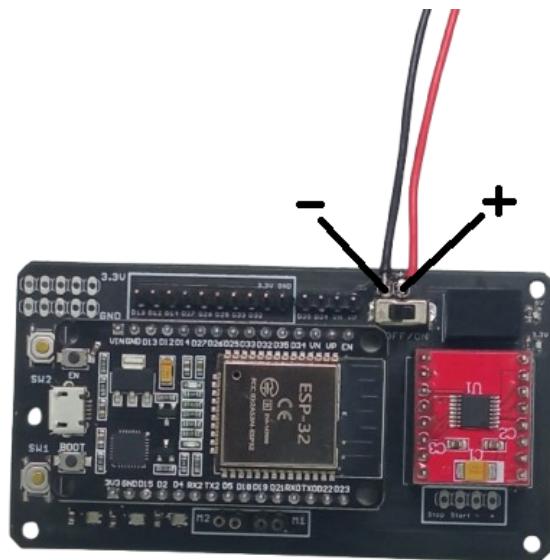


Figura 13: Conexión de la Batería

Encendemos el interruptor ON/OFF

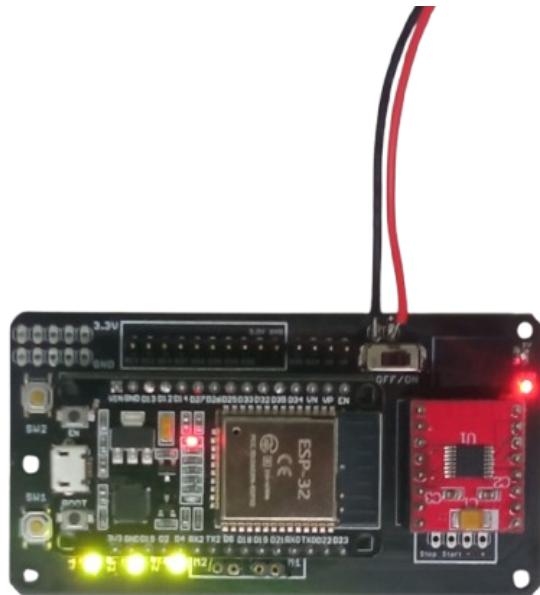


Figura 14: Encendido de la Tarjeta

### 3.4. Comportamiento del Programa de Prueba

El código de prueba permite comprobar el funcionamiento básico de los pulsadores y LEDs integrados en la tarjeta:

- Al presionar **SW2**, seleccionamos un color del **LED**.

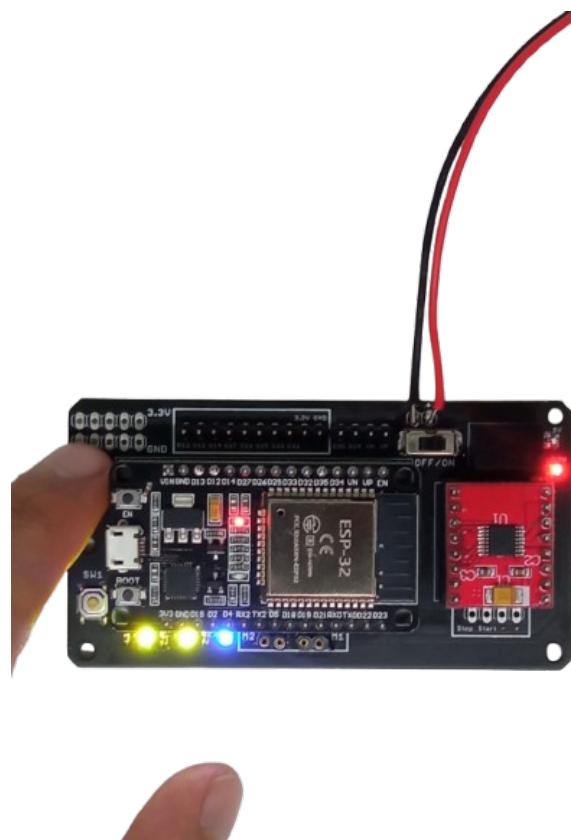


Figura 15: Prueba SW2

- Al presionar **SW1**, se escoge otro **LED**.

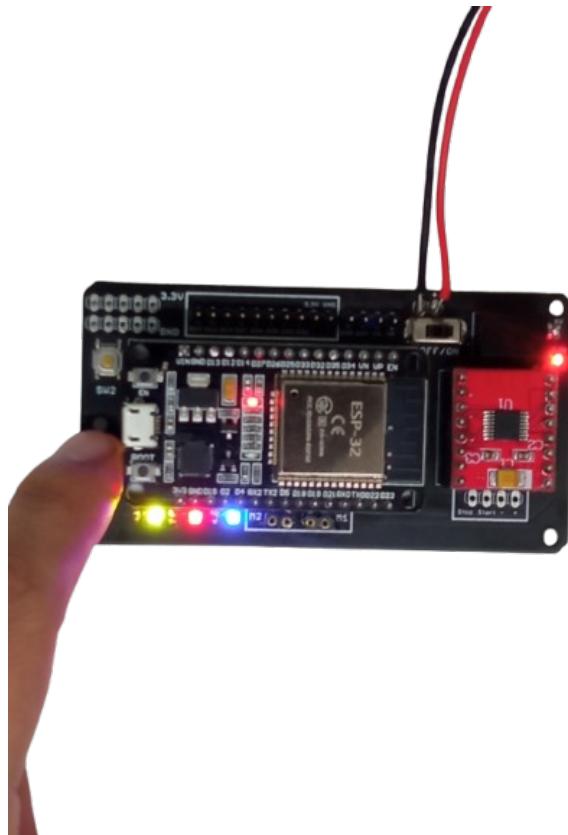


Figura 16: Prueba SW1

Este comportamiento confirma que las líneas de entrada y salida entre la tarjeta y el ESP32 están correctamente conectadas y operativas.

#### 4. Recomendaciones Finales

Para garantizar un funcionamiento confiable y prolongar la vida útil de la tarjeta ESPBot, es conveniente tener en cuenta las siguientes recomendaciones antes de iniciar cualquier prueba o montaje adicional.

- Verificar siempre la **orientación y correcta colocación** de cada módulo instalado en la tarjeta.
- Probar los demás **códigos de ejemplo** disponibles en la página oficial, los cuales permiten explorar distintas funciones de la tarjeta.
- Asegurar que la **batería esté correctamente polarizada** al conectarla en la bornera correspondiente.
- Comprobar periódicamente el **nivel de carga** de la batería para evitar apagados inesperados durante el funcionamiento.

Con estas verificaciones básicas, la tarjeta ESPBot queda lista para avanzar hacia el desarrollo completo de un **seguidor de línea**. Si deseas adquirir sensores o motores de alto rendimiento, nuestro equipo de ventas puede orientarte. Y si necesitas apoyo técnico durante tu proyecto, nuestro equipo de ingeniería está disponible para ayudarte.