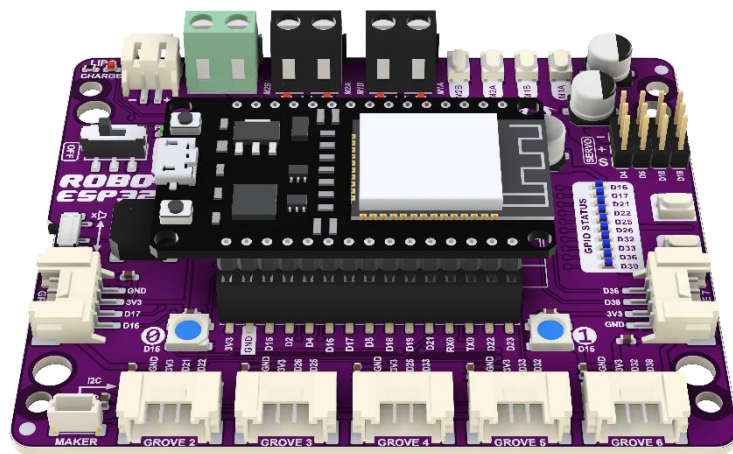


Programación de Robo ESP32 con IDE Arduino



Profesor: José Manuel Ruiz Gutiérrez
Junio 2025

Contenido

Primeros pasos con Robo ESP32 y Arduino IDE	3
Configuración del Robo ESP32	3
Configuración de la instalación del IDE de Arduino	3
Instalación de la placa ESP32 en Arduino IDE	5
Configuración de la instalación del controlador CP210x para la placa ESP32	9
Verifique la instalación y seleccione la placa y el puerto COM correctos.....	11
Conexión de Robo ESP32	12
Método # 1 : Cable USB micro B (5V).....	12
Método # 2 : Batería de iones de litio / Li-Po de celda única (3.7V).....	13
Método # 3 : Vin (3.6-6V)	13
Ejemplos de programación de Robo ESP32 con IDE Arduino	15
Blink.....	15
Estructura de nuestro programa.....	16
Procedimiento de actuación	16
Control de LED RGB.....	20
Usando entradas Digitales	23
Sonido.....	25
Control de un Motor de CC	28
Esquema de montaje.....	30
Control de un servomotor.....	31
Leer el valor analógico	34
Leer el valor digital	37
Manejo de SSD1315 módulo OLED a través de I2C	40
Puerto de Grove	40
Puerto Maker	42

Nota: El presente texto se ha elaborado usando como texto base el [manual que Cytron](#) ofrece entre los materiales de aprendizaje sobre Robo ESP32.

Primeros pasos con Robo ESP32 y Arduino IDE

Este capítulo te ayudará a poner en marcha tus proyectos de Robo ESP32 con algunas instrucciones sencillas. Hablaremos de:

1. Configuración del Robo ESP32
2. Instalación de la placa ESP32 en Arduino IDE
3. Configuración de la instalación del controlador CP210x para la placa ESP32
4. Enciende el Robo ESP32
5. Ejemplos de código fundamental
 1. Parpadear
 2. Iluminando el Robo ESP32
 3. Presione los botones
 4. Suelta el ritmo
 5. Motor de CC
 6. Micro servomotor
 7. Leer el valor del sensor analógico
 8. Leer el valor del sensor digital
 9. Visualización de texto en SSD1315 módulo OLED a través de I2C

Configuración del Robo ESP32

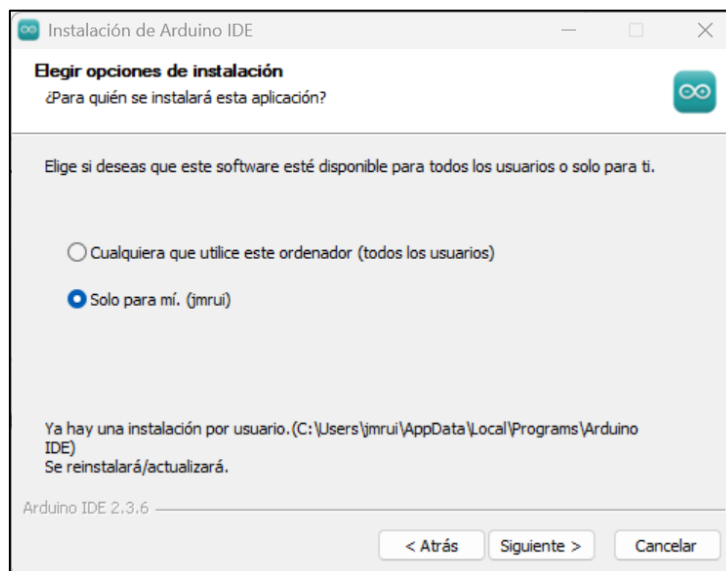
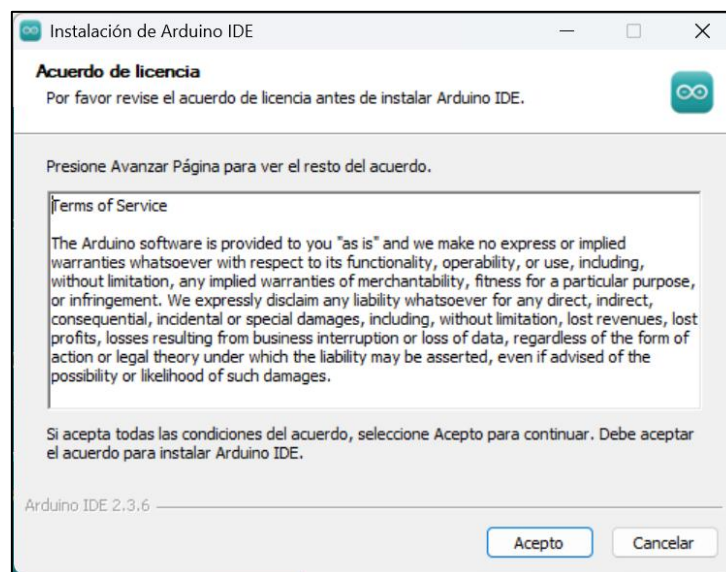
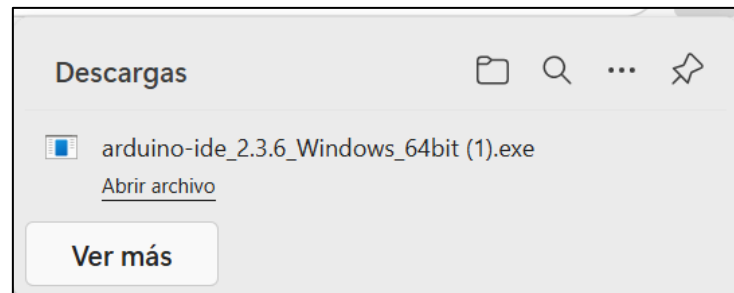
Veamos cómo hemos de configurar la tarjeta Robo ESP32 y repasaremos algunos consejos simples para solucionar problemas si no puede conectar su ESP32 a su computadora portátil.

Configuración de la instalación del IDE de Arduino

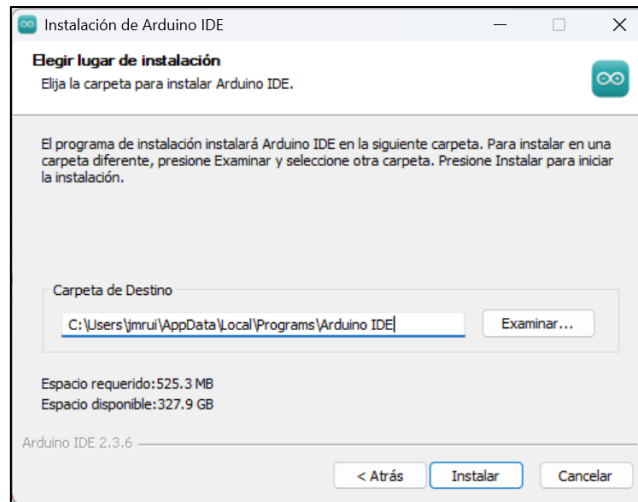
1. Abra el navegador y busque el IDE de Arduino o simplemente haga clic en este enlace [IDE de Arduino](#) para descargar el IDE de Arduino. Cuando abra el enlace, el navegador será el mismo que se muestra en la figura a continuación. Si está utilizando Windows, puede hacer clic en Windows (Win 10 y versiones posteriores). De lo contrario, si usa macOS o Linux, puede descargar según sus preferencias. [Software | Arduino](#)



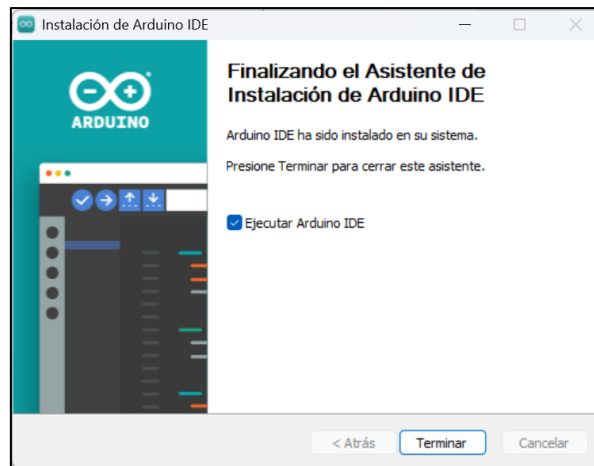
2. Después de descargar la configuración, abra la carpeta y haga clic en arduino-ide para instalar. Haga clic en "Ejecutar", "Acepto" y "Siguiente"
- 3.



4. Haga clic en "Instalar" y espere hasta que finalice la instalación.

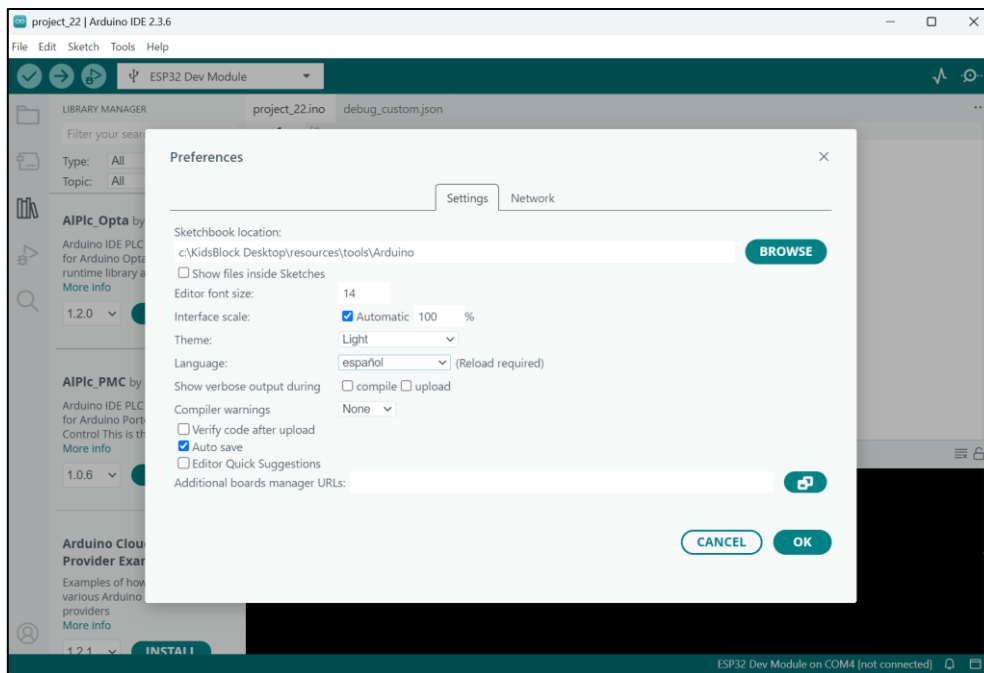


5. Ya está completando la configuración de instalación del IDE de Arduino. Haz clic en "Finalizar" para abrir el IDE de Arduino.

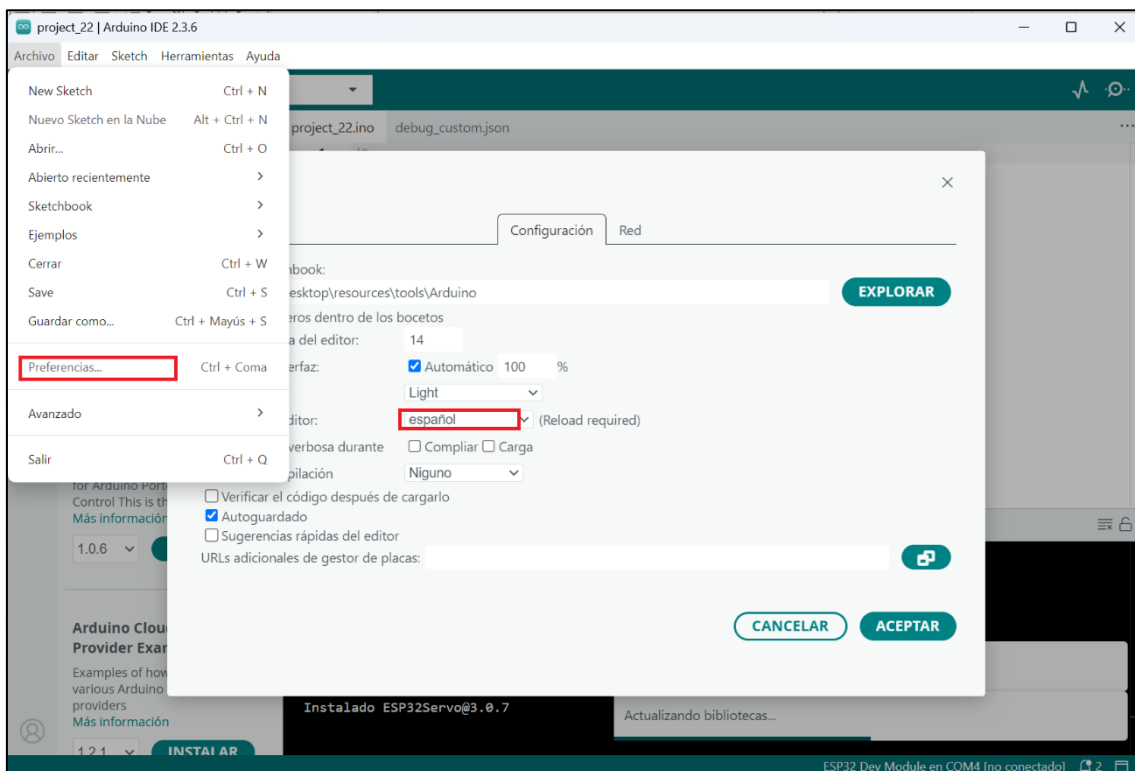


Instalación de la placa ESP32 en Arduino IDE

Para instalar la placa ESP32 en su IDE de Arduino, siga las siguientes instrucciones:



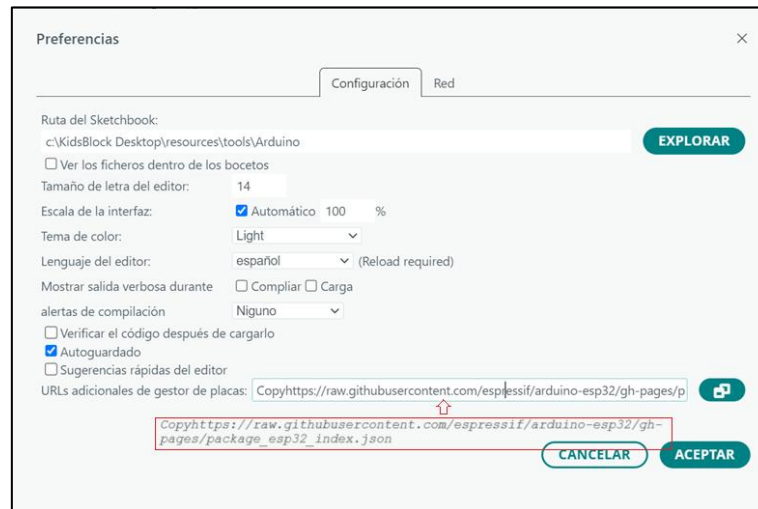
1. En tu IDE de Arduino, ve a **Archivo > Preferencias**



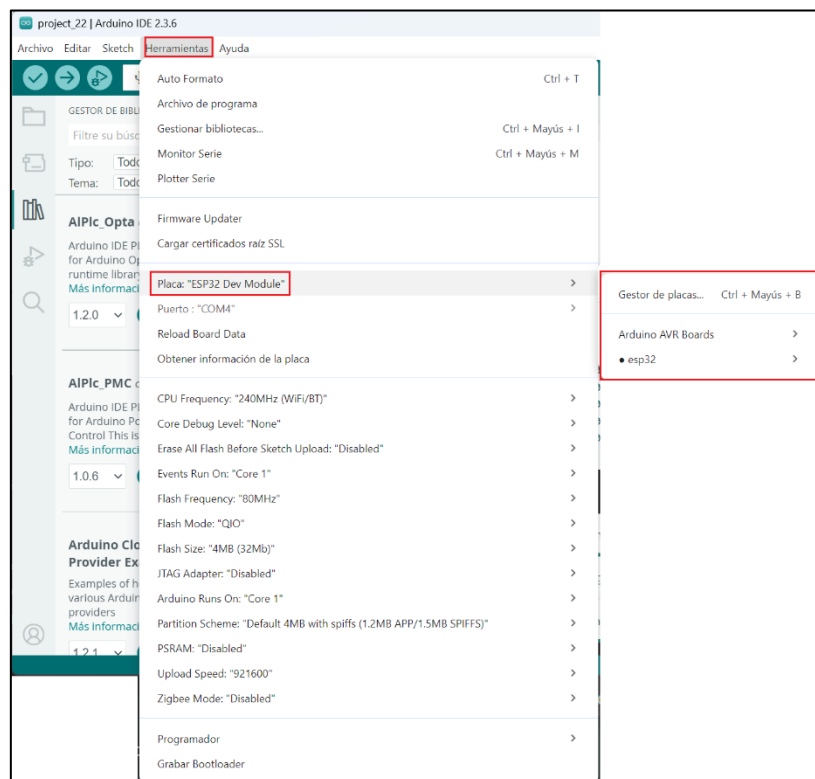
2. Ingresa lo siguiente en el campo "URL adicionales de Board Manager":

Copyhttps://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

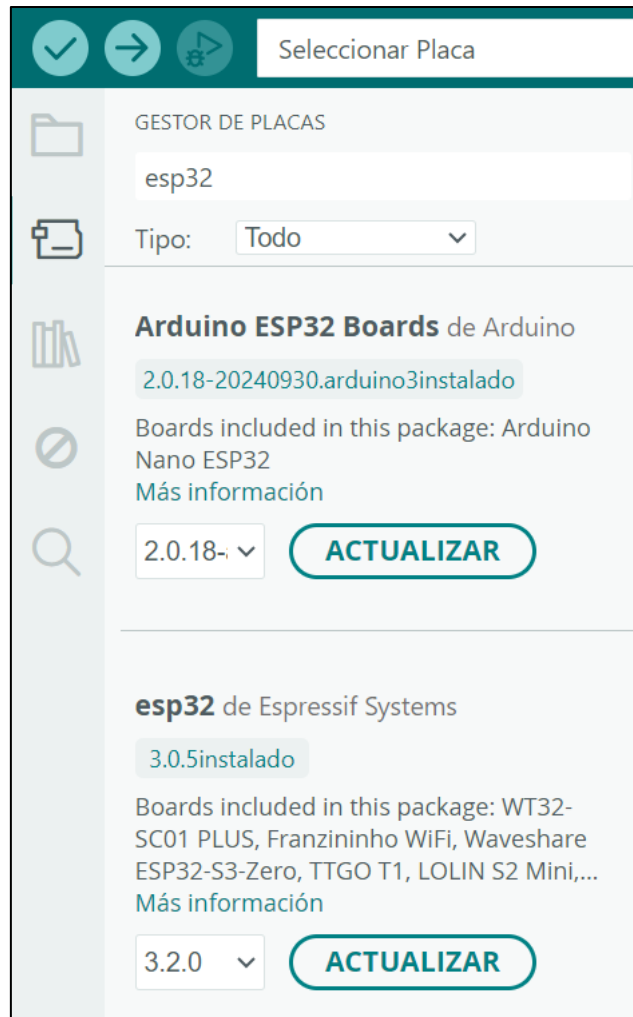
3. A continuación, haga clic en el botón "Aceptar"



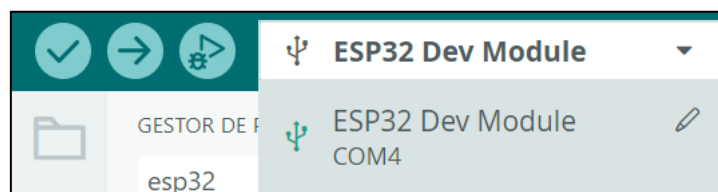
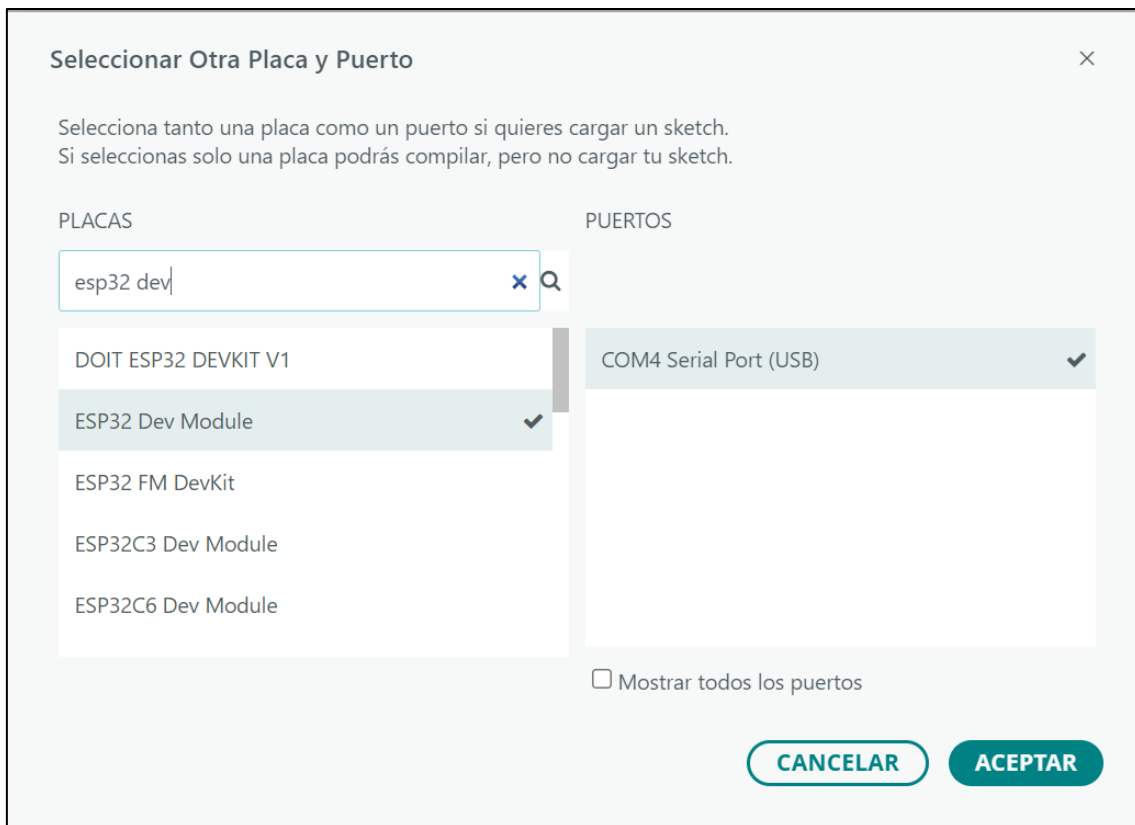
4. Tablero abierto Arduino Board Manager en **Tool> Board> Board Manager**. El menú desplegable debería gustar esto.



5. Escriba "ESP32" en la barra de búsqueda y presione enter.
6. Haga clic en instalar la placa **esp32** y reinicie el IDE de Arduino



7. Vuelva a abrir Arduino IDE y navegue hasta **Tool> Board> ESP32 Arduino> ESP Dev Module**



Configuración de la instalación del controlador CP210x para la placa ESP32

En esta sección, lo guiaremos a través de la instalación del controlador CP210x, que es esencial si está utilizando una placa compatible con CP210x por primera vez, como NodeMCU ESP32. Este controlador garantiza la comunicación adecuada entre su computadora y la placa, lo que permite la funcionalidad de USB a serie para cargar código e interactuar con su dispositivo.

1. Descargue CP210x [aquí](#) si está utilizando la placa compatible con ESP32 por primera vez.

The screenshot shows the Silicon Labs website navigation bar with links for Products, Applications, Ecosystems, Resources, and Company. The breadcrumb trail indicates the path: Developer Tools > USB to UART Bridge VCP Drivers. The 'DOWNLOADS' tab is active, displaying a list of software drivers. The 'CP210x Windows Drivers' link is highlighted with a red box. Other links include 'CP210x Universal Windows Driver', 'CP210x VCP Mac OSX Driver', 'CP210x VCP Windows', and 'CP210x Windows Drivers with Serial Enumerator'. A 'Show 6 more Software' link is also present.

2. Extraiga los archivos descargados a la carpeta deseada.

3. Instale el controlador para su sistema operativo, para nuestro caso, elegimos x64.

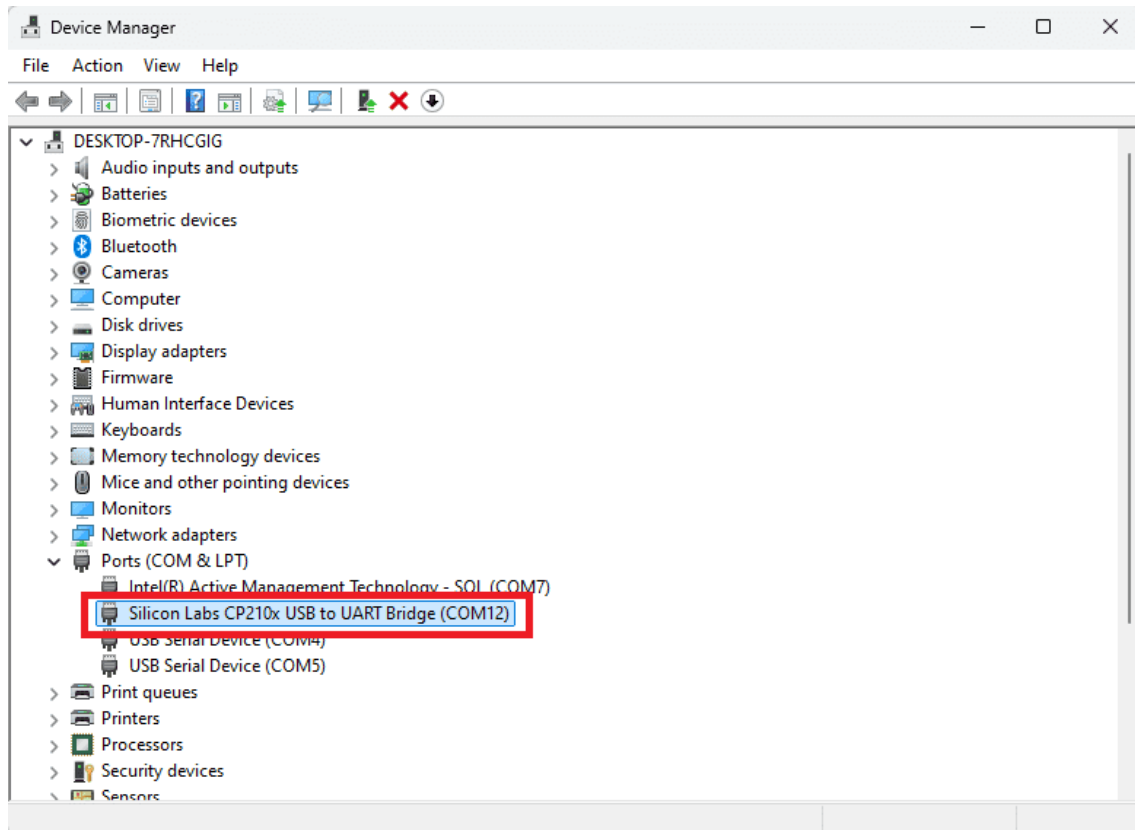
The screenshot shows a Windows File Explorer window titled 'CP210x_Windows_Drivers'. It displays a list of files and folders. The 'CP210xVCPInstaller_x64' file is highlighted with a red box and a red arrow. Other files include 'x64', 'x86', 'CP210xVCPInstaller_x86', 'dpinst', 'SLAB_License_Agreement_VCP_Windo...', 'slabvcp', and 'v6-7-6-driver-release-notes'.

The screenshot shows the 'CP210x USB to UART Bridge Driver Installer' window. The title bar reads 'CP210x USB to UART Bridge Driver Installer'. The main window displays the text 'Completing the Installation of the CP210x USB to UART Bridge Driver'. Below this, it states 'The drivers were successfully installed on this computer. You can now connect your device to this computer. If your device came with instructions, please read them first.' A table shows the driver name 'Silicon Laboratories Inc. ...' and its status 'Ready to use'. At the bottom, there are buttons for '< Back', 'Finish', and 'Cancel'. The 'CP210xVCPInstaller_x64' file is highlighted with a red box and a red arrow in the background File Explorer window.

4. Reinicia tu computadora.

Verifique la instalación y seleccione la placa y el puerto COM correctos

1. Puede verificar el número de puerto de su ESP32 en el "**Administrador de dispositivos**" de su computadora



2. En el menú "**Seleccionar otra placa y puerto**", elija el modelo exacto de placa y el mismo número de puerto que corresponde a la placa ESP32 conectada a su computadora.

Conexión de Robo ESP32

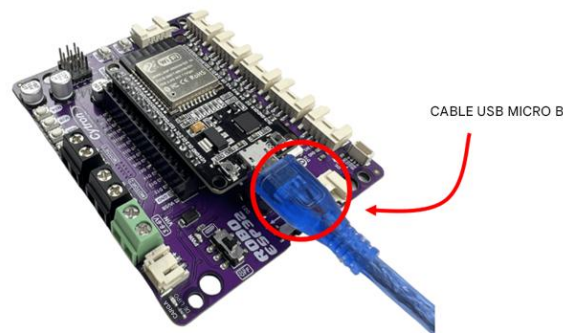
Esta guía lo guiará a través de tres opciones convenientes para encender su Robo ESP32, lo que permite flexibilidad en cualquier configuración:

1. Micro USB (puerto VUSB-microUSB)
2. Batería de una sola celda (puerto LiPo VBAT-JST PH2.0)
3. Alimentación externa (bloque de terminales VIN):

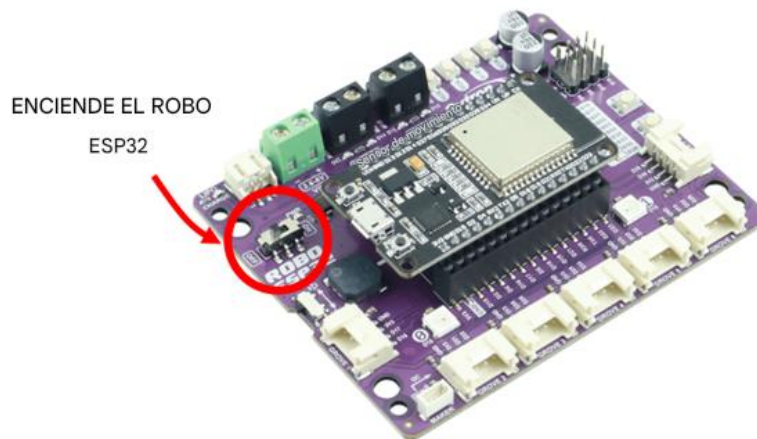
¡Elige la opción que mejor se adapte a las necesidades de tu proyecto y pon en marcha tu Robo ESP32!

Método # 1 : Cable USB micro B (5V)

1. Utilice un cable USB micro B para conectar su NodeMCU ESP32 a la computadora.



2. Encienda el interruptor de encendido en Robo ESP32.

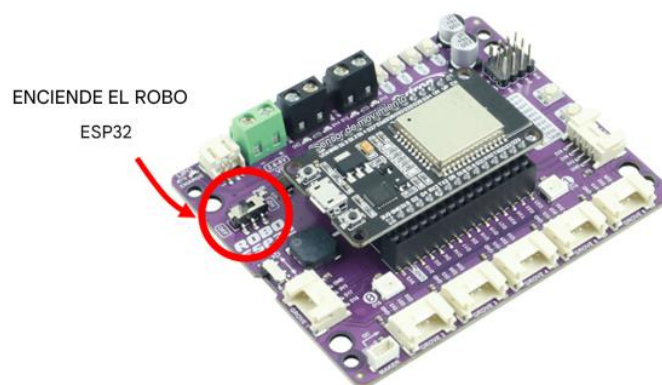


Método # 2 : Batería de iones de litio / Li-Po de celda única (3.7V)

1. Conecte una batería Li-Po / Li-Ion de una sola celda al conector de la batería Li-Po



2. Encienda el interruptor de encendido en Robo ESP32

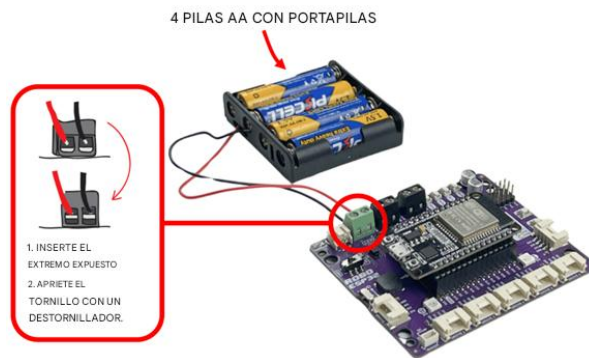


Nota: La batería Li-Po / Li-Ion de una sola celda se puede recargar mediante el cable USB micro B para conectar NodeMCU ESP32

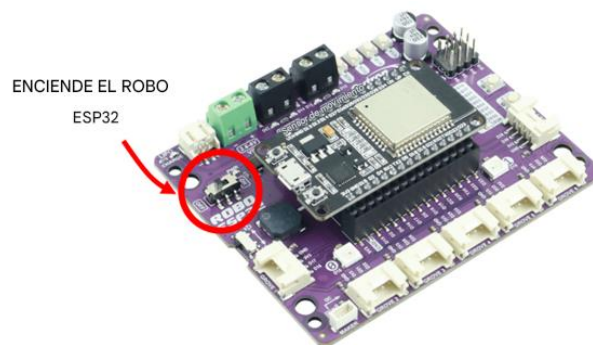
Nota: Encienda siempre el Robo ESP32 mientras flashea el código o ejecuta el código cargado.

Método # 3 : Vin (3.6-6V)

1. Inserte las pilas AA en un soporte de pilas con los cables de los extremos expuestos.
2. Inserte y apriete el extremo expuesto en el bloque de terminales Vin (verde claro).



3. Encienda el interruptor de encendido en Robo ESP32.



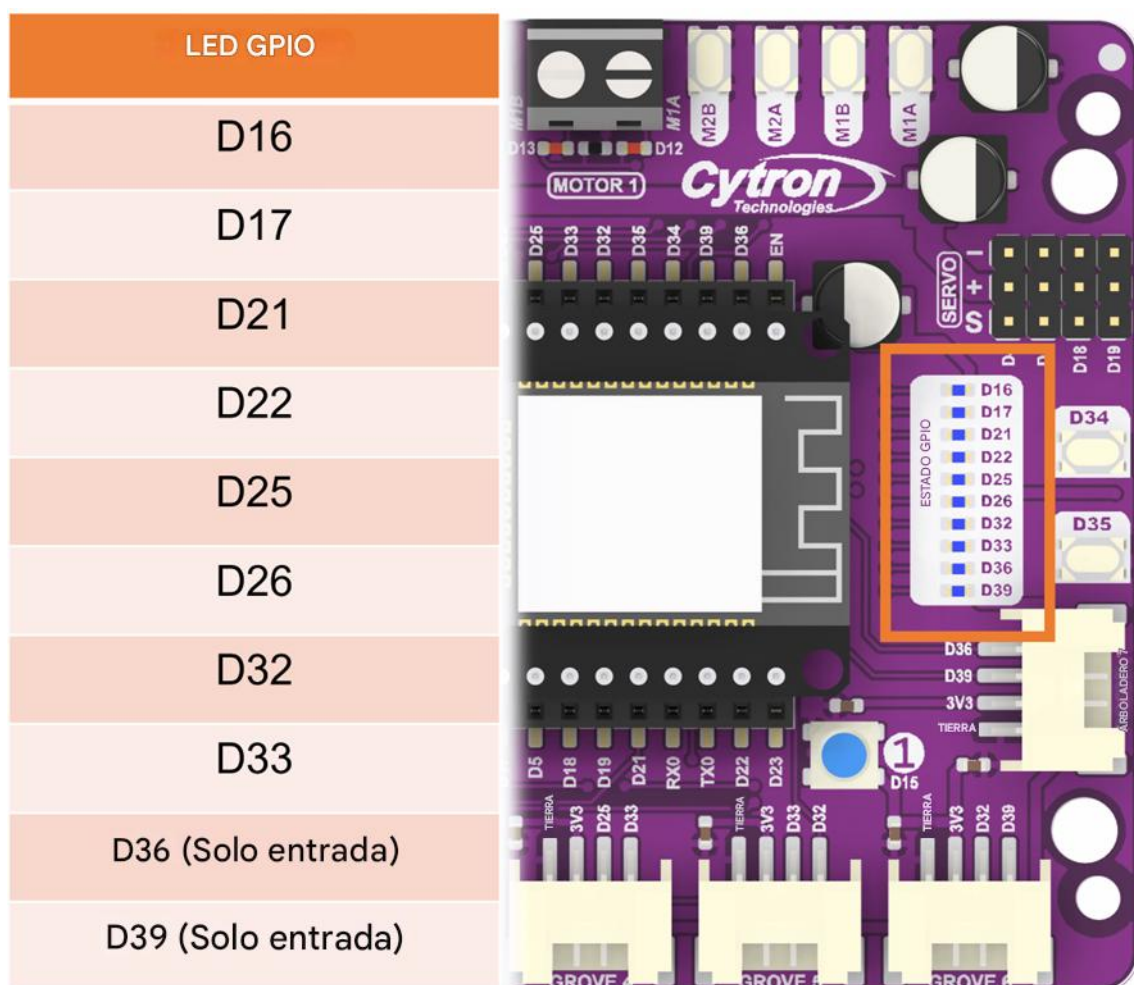
Nota: Encienda siempre el Robo ESP32 mientras flashea el código o ejecuta el código cargado.

Ejemplos de programación de Robo ESP32 con IDE Arduino

Blink

Esta guía le mostrará cómo encender los LED de su Robo ESP32. Cada LED representa un pin designado, que se puede utilizar para solucionar problemas de entrada/salida y para crear efectos visuales.

La siguiente figura ilustra la configuración GPIO para cada uno de los LED de E/S integrados.



Programación:

Con esta primera practica vamos a activar los leds de salida que se encuentran en la propia placa Robo ESP32. Las salida que vamos a activar son las que se muestran en la figura anterior (LED GPIO).

Nuestro programa contara con la estructura clásica de un programa en el IDE Arduino:

```

/* Comentarios

*/

// Definición de variables y parámetros

void setup() {
  // ponga su código de configuración aquí, para ejecutar una vez:
}

void loop() {
  // poner tu código principal aquí, para ejecutar repetidamente:
}

```

Estructura de nuestro programa

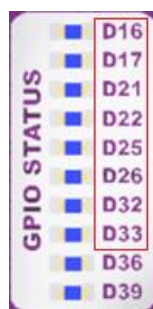
El programa deberá realizar las siguientes operaciones:

El bucle de ejecución realizará las siguientes acciones

1. Apagar todos los LEDs
2. Esperar 1500 ms
3. Activar i desactivar con un tiempo intermedio de 200 ms cada una de las salidas.
4. Esperar 1000 ms
5. Iniciar
6. Establecer un parpadeo de todos los LEDs pasando de Alto a Bajo los leds con un intervalo de 0,5 seg.

Procedimiento de actuación

1. Conecte su ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
2. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
3. Copia y pega el siguiente código en tu editor:



8 Salidas Digitales
implicadas


```

/*
DESCRIPCIÓN:
Este código de ejemplo utiliza el Robo ESP32 para probar la funcionalidad de
los ledes IO.
Inicialmente, todos los ledes estarán apagados. Los ledes parpadearán
alternativamente cada 0,5 segundos y finalmente todos los ledes permanecerán
encendidos.
Esta prueba asegura que los ledes en el Robo ESP32 están funcionando
correctamente.

AUTHOR   : Cytron Technologies Sdn Bhd
WEBSITE  : www.cytron.io
EMAIL    : support@cytron.io
*/

const int led1 = 16; //led conectado al pin D16
const int led2 = 17; //led Conectado al pin D17
const int led3 = 21; //led Conectado al pin D21
const int led4 = 22; //led Conectado al pin D22
const int led5 = 25; //led Conectado al pin D25
const int led6 = 26; //led Conectado al pin D26
const int led7 = 32; //led Conectado al pin D32
const int led8 = 33; //led Conectado al pin D33

void setup() {

    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);
    pinMode(led4, OUTPUT);
    pinMode(led5, OUTPUT);
    pinMode(led6, OUTPUT);
    pinMode(led7, OUTPUT);
    pinMode(led8, OUTPUT);
}

void loop() {
    //Apague los ledes
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
    digitalWrite(led6, LOW);
    digitalWrite(led7, LOW);
    digitalWrite(led8, LOW);
}

```

```

delay(1500); //Esperar 1.5 segundos

// Patrón intermitente alternante para todos los ledes
digitalWrite(led1, HIGH);
delay(200);
digitalWrite(led1, LOW);
delay(200);
digitalWrite(led2, HIGH);
delay(200);
digitalWrite(led2, LOW);
delay(200);
digitalWrite(led3, HIGH);
delay(200);
digitalWrite(led3, LOW);
delay(200);
digitalWrite(led4, HIGH);
delay(200);
digitalWrite(led4, LOW);
delay(200);
digitalWrite(led5, HIGH);
delay(200);
digitalWrite(led5, LOW);
delay(200);
digitalWrite(led6, HIGH);
delay(200);
digitalWrite(led6, LOW);
delay(200);
digitalWrite(led7, HIGH);
delay(200);
digitalWrite(led7, LOW);
delay(200);
digitalWrite(led8, HIGH);
delay(200);
digitalWrite(led8, LOW);

delay(1000); // Pausa durante 1 segundos antes de la siguiente secuencia

// Establecer un nuevo parpadeo
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
digitalWrite(led5, HIGH);
digitalWrite(led6, HIGH);
digitalWrite(led7, HIGH);
digitalWrite(led8, HIGH);

delay(500); //Retraso de 0,5 segundos

```

```
digitalWrite(led1, LOW);
digitalWrite(led2, LOW);
digitalWrite(led3, LOW);
digitalWrite(led4, LOW);
digitalWrite(led5, LOW);
digitalWrite(led6, LOW);
digitalWrite(led7, LOW);
digitalWrite(led8, LOW);

delay(500); //Retraso de 0,5 segundos

// Repetir el patrón de nuevo
digitalWrite(led1, HIGH);
digitalWrite(led2, HIGH);
digitalWrite(led3, HIGH);
digitalWrite(led4, HIGH);
digitalWrite(led5, HIGH);
digitalWrite(led6, HIGH);
digitalWrite(led7, HIGH);
digitalWrite(led8, HIGH);

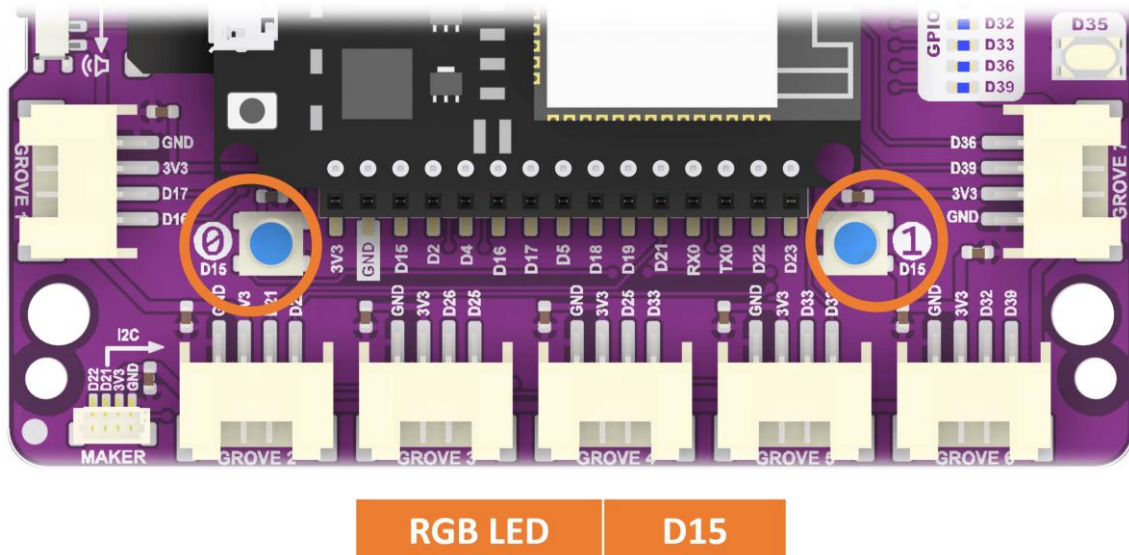
delay(500); //Retraso de 0,5 segundos
```

[Ver Guía de inicio / Arduino / Blink / Blink.ino](#) entregado con  [por emgithub](#)

4. Verifique y cargue el script en su ESP32.
5. Encienda el interruptor de encendido de Robo ESP32. La salida esperada es que los LED parpadearán alternativamente cada 0,5 segundos en orden de D16 a D33 y, finalmente, todos los LED permanecerán encendidos.

Control de LED RGB

Esta guía le mostrará cómo encender el LED WS2812 RGB en su placa Robo ESP32. Hay dos LED RGB integrados, ambos controlados por el mismo pin, D15.



Nota: Deberá instalar la biblioteca [Adafruit NeoPixel](https://github.com/adafruit/Adafruit_NeoPixel/tree/master) para usar este código.

1. Conecte su ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
2. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
3. Copia y pega el siguiente código en tu editor:

```
/*
DESCRIPCIÓN:
Este código de ejemplo utilizará Robo ESP32 para iluminar los ledes RGB a
bordo.
Los ledes RGB cambiarán sus colores secuencialmente individualmente.

AUTHOR   : Cytron Technologies Sdn Bhd
WEBSITE  : www.cytron.io
EMAIL    : support@cytron.io
TRADUCCIÓN: José Manuel Ruiz

REFERENCIA:
Adafruit_NeoPixel library link:
https://github.com/adafruit/Adafruit_NeoPixel/tree/master
```

```

*/

#include <Adafruit_NeoPixel.h>

// Declare the pin number of the Neopixel LED and the number of the leds
const int RGBPin = 15;    // RGB pin for Robo ESP32
const int numPixels = 2;  //Number of RGB LEDs

// Initialize the NeoPixel RGB LEDs on pin D15
Adafruit_NeoPixel pixels(numPixels, RGBPin, NEO_GRB + NEO_KHZ800);

void setup() {

    pixels.begin(); // Inicializar la biblioteca NeoPixel
    pixels.clear();  // Establecer todos los colores de píxel a 'off'
    pixels.show();   // Envía colores de píxeles actualizados al hardware.
}

void loop() {

    /*

    pixels.Color() takes the RGB values, from 0.0.0 to 255.255.255
    NeoPixels are numbered from 0 to (number of pixels - 1)

    */

    pixels.setPixelColor(0, pixels.Color(200, 0, 0)); // Rojo
    pixels.setPixelColor(1, pixels.Color(200, 0, 200)); // Magenta
    pixels.show(); // Enviar los
colores de píxel actualizados al hardware.
    delay(1000);

    pixels.setPixelColor(0, pixels.Color(0, 200, 0)); // Verde
    pixels.setPixelColor(1, pixels.Color(200, 200, 0)); // Amarillo
    pixels.show(); // Enviar los
colores de píxel actualizados al hardware.
    delay(1000);

    pixels.setPixelColor(0, pixels.Color(0, 0, 200)); // Azul
    pixels.setPixelColor(1, pixels.Color(0, 200, 200)); // Cyan
    pixels.show(); // Envía los
colores de píxel actualizados al hardware.
    delay(1000);

    pixels.setPixelColor(0, pixels.Color(0, 0, 0)); // RGB apagado
    pixels.setPixelColor(1, pixels.Color(0, 0, 0)); // RGB desactivado

```

```
pixels.show(); // Enviar los colores de píxel actualizados al hardware.  
delay(1000);  
}
```

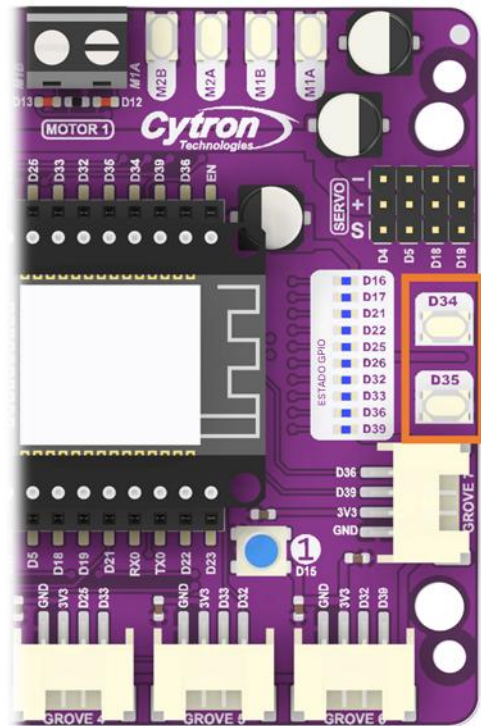
Ver en bruto[Guía de inicio / Arduino / RGB / RGB.ino](#) entregado con  [emgithub](#)

4. Verifique y cargue el boceto en su ESP32.
5. Apile y encienda el interruptor de encendido de Robo ESP32. **La salida esperada** es que los LED RGB se iluminen individualmente en tres colores diferentes secuencialmente hasta que se apaguen y luego se repitan.

Usando entradas Digitales

Esta guía muestra cómo utilizar los dos botones integrados en el Robo ESP32 con Arduino IDE. Los botones están conectados a los pines D34 y D35. Puede programar estos botones para realizar diferentes tareas, como encender y apagar los LED, reproducir sonidos o cambiar entre modos cuando se presionan.

Número de botones de usuario	GPIO
2	D34
	D35



1. Conecte su ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
2. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
3. Copia y pega el siguiente código en tu editor:

```
/*  
DESCRIPCIÓN:  
Este código de ejemplo mostrará cómo usar el botón User en el Robo ESP32  
como una entrada.  
En este código, el botón se utilizará para controlar un led a bordo.  
Si se pulsa el botón, el led se encenderá durante 0,5 segundos y luego se  
apagará  
  
AUTHOR   : Cytron Technologies Sdn Bhd  
WEBSITE  : www.cytron.io  
EMAIL    : support@cytron.io
```

TRADUCCIÓN: José Manuel Ruiz

*/

```
const int led1 = 21;
const int led2 = 22;
const int button1 = 34;
const int button2 = 35;
```

```
void setup() {
```

```
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(button1, INPUT_PULLUP);
    pinMode(button2, INPUT_PULLUP);
```

```
}
```

```
void loop() {
```

```
    // Se pulsa el botón de verificación 1 (D34)
```

```
    if (!digitalRead(button1)) {
        // LED D21 is light up for 0.5s then turned off.
        digitalWrite(led1, HIGH);
        delay(500);
        digitalWrite(led1, LOW);
    }
```

```
    // Se pulsa el botón de verificación 2 (D35)
```

```
    if (!digitalRead(button2)) {
        // El led D22 se enciende durante 0,5 segundos y luego se apaga.
        digitalWrite(led2, HIGH);
        delay(500);
        digitalWrite(led2, LOW);
    }
```

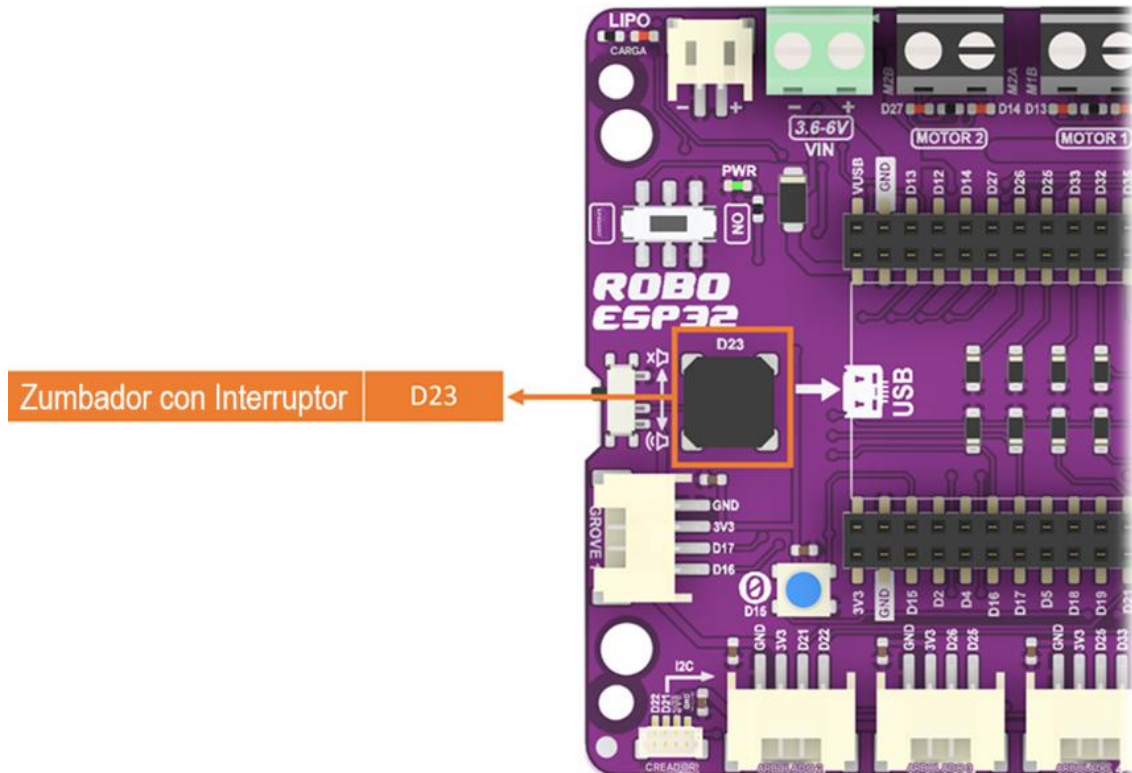
```
}
```

Ver Guía de inicio/Arduino/UserButton/UserButton.ino entregado con  **emgithub**

1. Verifique y cargue el boceto en su ESP32.
2. Apile y encienda el interruptor de encendido de Robo ESP32. La salida esperada es que el botón de usuario se convierta en una entrada para la placa. Después de presionar cada botón, hará que el LED se encienda durante 0,5 segundos y luego se apague.

Sonido

El Robo ESP32 viene con un zumbador incorporado en el pin D23, perfecto para crear efectos de sonido y melodías. En esta sección se proporciona código de ejemplo para ayudarle a programar el zumbador rápidamente. Además, un interruptor de silencio le permite apagar el zumbador cuando no está en uso, liberando el pin D23 para otros fines.



Conecte su ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.

Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.

Copie y pegue el siguiente código en su editor:

```
/*  
DESCRIPCIÓN:  
Este código de ejemplo utilizará el buzzer del Robo ESP32 para  
reproducir los tonos.  
El botón de usuario también se utiliza en este código. Al inicio, una  
breve melodía se reproducirá  
y luego el código esperará a que se pulse el botón para reproducir otro  
conjunto de tonos cortos.  
  
AUTHOR : Cytron Technologies Sdn Bhd  
WEBSITE : www.cytron.io
```

```

EMAIL      : support@cytron.io
TRADUCCIÓN: José Manuel Ruiz
*/

const int BuzzerPin = 23;    // Zumbador Robo ESP32 conectado al pin D23
const int UserButton1 = 34;  // Botón de usuario 1 conectado al pin D34
const int UserButton2 = 35;  // Botón de usuario 2 conectado al pin D35

// Notas para la canción "Happy Birthday"
int melody_note[] = { 392, 392, 440, 392, 523, 494, 392, 392, 440, 392,
587, 523 };

// Duraciones cortas de las notas melódicas para tempo rápido
(milisegundos)
int melody_duration[] = { 100, 100, 200, 200, 200, 270, 100, 100, 200,
200, 200, 270 };

void setup() {
    // Establecer el zumbador como un dispositivo de salida
    pinMode(BuzzerPin, OUTPUT);

    // Establecer el botón como un dispositivo de entrada con pull-up
    interno
    pinMode(UserButton1, INPUT_PULLUP);
    pinMode(UserButton2, INPUT_PULLUP);

    // Reproduce la melodía "Feliz cumpleaños" durante la configuración a
    un tempo rápido
    play_melody(BuzzerPin);
}

void loop() {
    // Comprobar si el botón está presionado (LOW significa presionado
    debido a INPUT_PULLUP)
    if (!digitalRead(UserButton1)) {
        // Reproducir una melodía corta cuando se pulsa el botón
        tone(BuzzerPin, 262, 100); // Toca Nota C
        delay(100);
        tone(BuzzerPin, 659, 100); // Toca Nota E
        delay(100);
        tone(BuzzerPin, 784, 100); // Toca Nota G
        delay(100);
        noTone(BuzzerPin); // Apague el zumbador
    }
    if (!digitalRead(UserButton2)) {
        // Reproducir una melodía corta cuando se pulsa el botón
        tone(BuzzerPin, 784, 100); // Toca Nota C
        delay(100);
    }
}

```

```
tone(BuzzerPin, 659, 100); // Toca Nota E
delay(100);
tone(BuzzerPin, 262, 100); // Toca Nota G
delay(100);
noTone(BuzzerPin); // Apague el zumbador
}
}
```

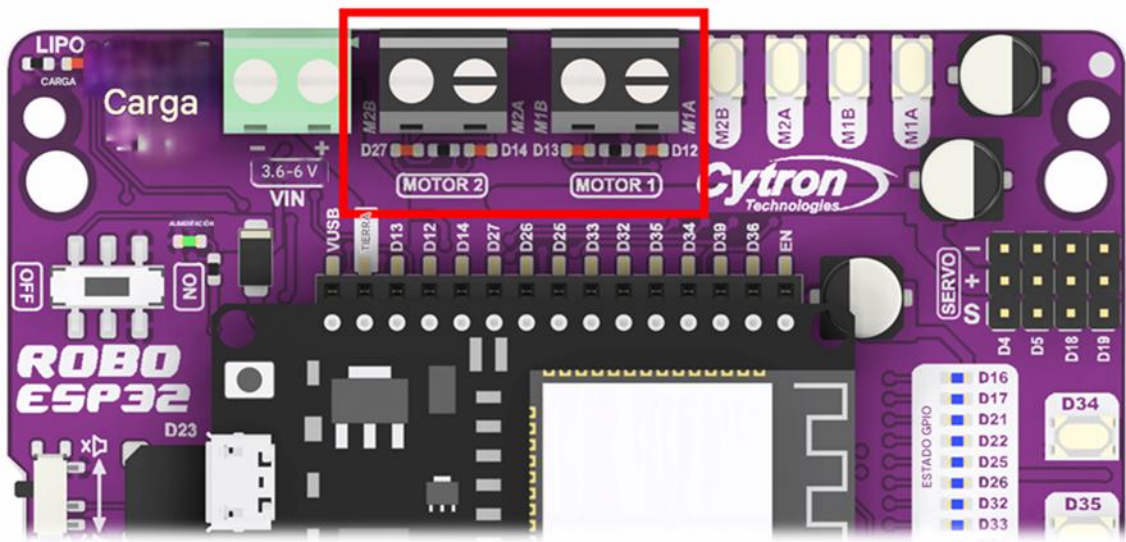
Ver Guía de inicio/Arduino/Buzzer/Buzzer.ino entregado con ❤ por emgithub

1. Verifique y cargue el boceto en su ESP32.
2. Apila y enciende el interruptor de encendido de Robo ESP32. El resultado esperado es que el timbre reproduzca la melodía de una canción de feliz cumpleaños y luego responda a las pulsaciones de botones para reproducir un tono corto.

Control de un Motor de CC

Esta guía le mostrará cómo encender un motor de CC a través de su Robo ESP32.

La siguiente figura muestra el GPIO para cada uno de los terminales del motor en Robo ESP32.



Terminales del Motor	Terminal	GPIO	Terminal	GPIO
Motor 1	M1A	D12	M1B	D13
Motor 2	M2A	D14	M2B	D27

1. Conecte su ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
2. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
3. Copia y pega el siguiente código en tu editor:

/*

DESCRIPCIÓN:
Este código de ejemplo utilizará: Robo ESP32 y NodeMCU ESP32 para controlar el motor de CC en los terminales del motor de CC Robo ESP32.

AUTHOR : Cytron Technologies Sdn Bhd
WEBSITE : www.cytron.io
EMAIL : support@cytron.io
TRADUCCIÓN: José Manuel Ruiz

```

*/

#include "CytronMotorDriver.h"

// Configurar el controlador del motor.
CytronMD motor1(PWM_PWM, 12, 13); // PWM 1A = Pin 12, PWM 1B = Pin 13.
CytronMD motor2(PWM_PWM, 14, 27); // PWM 2A = Pin 14, PWM 2B = Pin 27.

// La rutina de configuración se ejecuta una vez cuando presiona reset.
void setup() {
}

// La rutina del bucle se repite una y otra vez.
void loop() {
    motor1.setSpeed(255); // El motor 1 avanza a una velocidad del 50 %.
    motor2.setSpeed(255); // Motor 2 se desplaza hacia adelante a una
    velocidad del 50%.
    delay(1000);

    motor1.setSpeed(-255); // Motor 1 marcha hacia atrás a toda velocidad.
    motor2.setSpeed(-255); // Motor 2 se ejecuta hacia atrás a toda
    velocidad.
    delay(1000);

    motor1.setSpeed(0); // El motor 1 se detiene.
    motor2.setSpeed(0); // Motor 2 paradas.
    delay(1000);

    motor1.setSpeed(128); // El motor 1 avanza a una velocidad del 50%.
    motor2.setSpeed(128); // Motor 2 se dirige hacia adelante a una
    velocidad del 50%.
    delay(1000);

    motor1.setSpeed(-128); // Motor 1 marcha hacia atrás a toda velocidad.
    motor2.setSpeed(-128); // Motor 2 se ejecuta hacia atrás a toda
    velocidad.
    delay(1000);

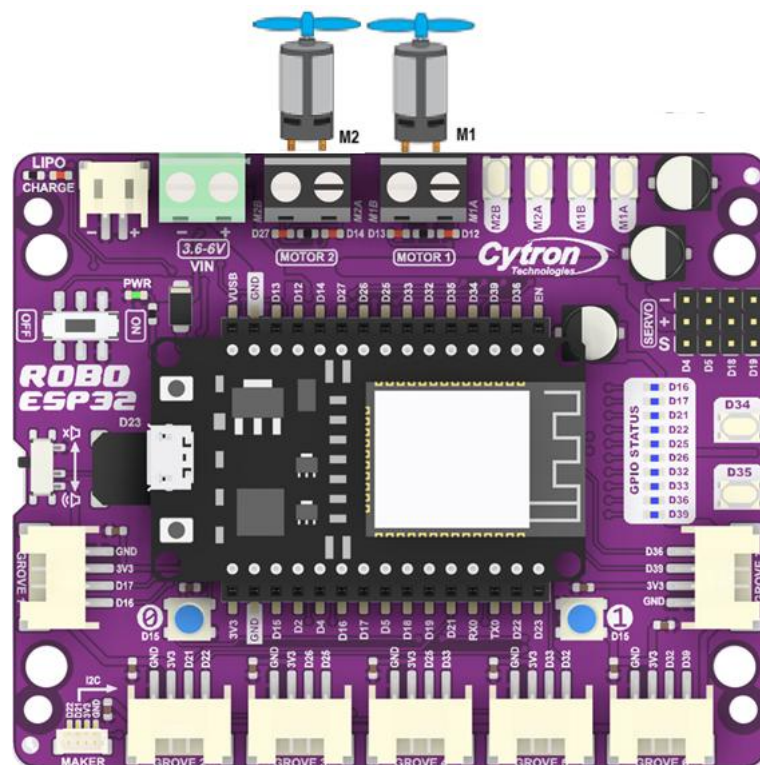
    motor1.setSpeed(0); // El motor 1 se detiene.
    motor2.setSpeed(0); // Motor 2 paradas.
    delay(1000);
}

```

Ver [Guía de inicio/Arduino/DCMotor/DCMotor.ino](#) entregado con  [emgithub](#)

4. Verifique y cargue el boceto en su ESP32.
 5. Conecte 2 motores de CC a ambos terminales del motor de CC (color negro).
 6. Conecte una batería Li-Po / Li-Ion de una sola celda a la lipo o conecte la batería AA con un soporte al (color verde claro).
 7. Asegúrese de que toda la conexión esté completa y luego encienda el interruptor de encendido. El resultado esperado de este código es que dos motores alternarán entre avanzar, retroceder y detenerse a diferentes velocidades, cada uno durante intervalos de un segundo.
- Nota:
 - Siempre puede probar las conexiones de los cables del motor de CC para ver si los cables están conectados correctamente al bloque de terminales presionando los botones de prueba del motor etiquetados como M1A, M1B, M2A o M2B mientras su Robo ESP32 está encendido.
 - Para permitir que el motor funcione hacia adelante, presione solo el botón "MXA".
 - Si desea cambiar la dirección de rotación de la rueda, puede intercambiar los cables conectados a los terminales.

Esquema de montaje



Control de un servomotor

Esta sección proporciona instrucciones para controlar micro servos en el Robo ESP32 usando Arduino IDE, conectado a los pines D4, D5, D18, D19.

Los puertos de servo incorporados del blindaje simplifican la configuración al eliminar el cableado adicional, lo que facilita la conexión de varios servos sin una fuente de alimentación externa.

Puertos servo	GPIO	Puerto		
S1	D4	S	+	-
S2	D5			
S3	D18			
S4	D19			



Nota: Asegúrese de haber instalado la biblioteca ESP32Servo.h en su IDE de Arduino

1. Conecte su NodeMCU ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
2. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
3. Copia y pega el siguiente código en tu editor:

```
/*  
DESCRIPCIÓN:  
Este código de ejemplo utilizará Robo ESP32 para controlar cuatro servomotores conectados a  
los puertos de servo integrados.  
El servomotor barrerá de 0° a 180° con un incremento de 1° cada 10 milisegundos.  
Luego, los servos retroceden de 180 grados a 0 grados con una disminución de 1 grado cada 10  
milisegundos.  
AUTOR : Cytron Technologies Sdn Bhd  
SITIO WEB : www.cytron.io  
CORREO ELECTRÓNICO : support@cytron.io  
*/
```



```

#include <ESP32Servo.h>

Crear objetos servo para cada servo
Servo servo1;
Servo servo2;
Servo servo3;
Servo servo4;

Configuración de void() {
Fije cada objeto servo a sus pines correspondientes
servo1.attach(4);
servo2.attach(5);
servo3.attach(18);
servo4.attach(19);
}

Bucle de vacío() {
Mueva los servos de 0 grados a 180 grados con un incremento de 1 grado por paso
for (int pos = 0; pos <= 180; pos += 1) {
servo1.write(pos); Establezca la posición del servo 1 en grados 'pos'
servo2.write(pos); Establezca la posición del Servo 2 en grados 'pos'
servo3.write(pos); Ajuste la posición del Servo 3 a grados 'pos'
servo4.write(pos); Establezca la posición del Servo 4 en grados 'pos'
retraso(10); Haga una pausa de 15 milisegundos para controlar la velocidad del movimiento
del servo
}

Mueva los servos de 180 grados a 0 grados con una disminución de 1 grado por paso
for (int pos = 180; pos >= 0; pos -= 1) {
servo1.write(pos); Establezca la posición del servo 1 en grados 'pos'
servo2.write(pos); Establezca la posición del Servo 2 en grados 'pos'
servo3.write(pos); Ajuste la posición del Servo 3 a grados 'pos'
servo4.write(pos); Establezca la posición del Servo 4 en grados 'pos'
retraso(10); Haga una pausa de 15 milisegundos para controlar la velocidad del movimiento
del servo
}
}
}

/*
NOTA:
Este código está escrito para servomotores estándar. Si está utilizando un servo de rotación
continua de 360 grados,
La función servo.write (PoS) se comporta de manera diferente a la servo estándar. Controla
la velocidad en lugar de la posición.

```

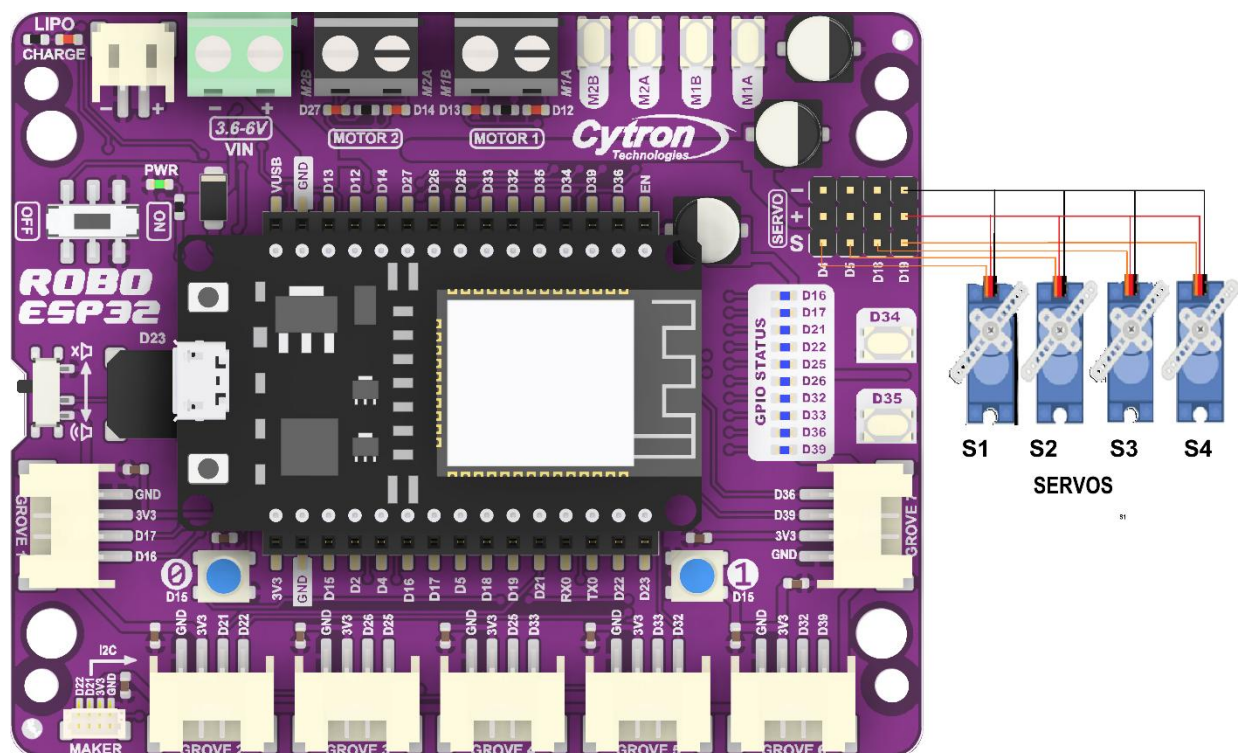

Un valor cercano a 90 significa que no hay movimiento, 0 es la velocidad máxima en una dirección y 180 es la velocidad completa en la otra dirección

*/

[Ver Guía de inicio / Arduino / Servo / Servo.ino](#) entregado con ❤ por [emgithub](#)

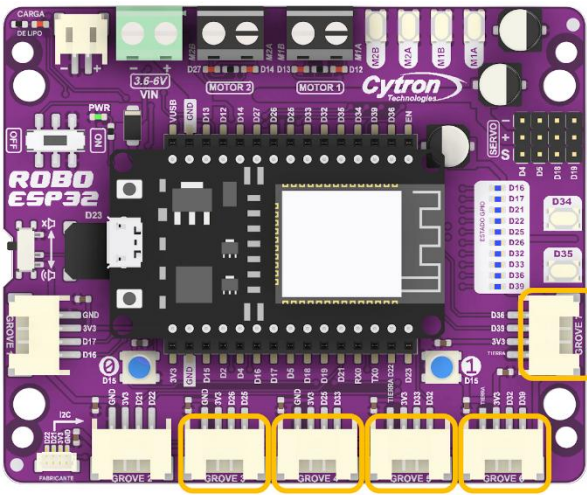
1. Verifique y cargue el boceto en su NodeMCU ESP32.
2. Conecte un micro servomotor a uno de los puertos del servo. Por ejemplo, puede conectar un micro servo SG90 al puerto D4 en Robo ESP32.
3. Conecte una batería Li-Po / Li-Ion de una sola celda al lipo o conecte la batería AA con un soporte al bloque de terminales VIN (color verde claro).
4. Asegúrese de que toda la conexión esté completa y luego encienda el interruptor de encendido. El resultado esperado es que el servomotor barre suavemente de 0° a 180°, aumentando en 1° cada 10 milisegundos. Después de alcanzar 180°, invertirá la dirección y volverá a 0°, disminuyendo en 1° cada 10 milisegundos.

Montaje:



Leer el valor analógico

Esta guía explica cómo obtener datos de sensores analógicos cuando están conectados al Robo ESP32. Equipado con siete puertos versátiles, el Robo ESP32 ofrece conexiones simples para una variedad de módulos y sensores. ¿No utilizas módulos compatibles? ¡No te preocupes! El blindaje también incluye cables adaptadores, lo que le permite aprovechar al máximo estos puertos con otros tipos de sensores.

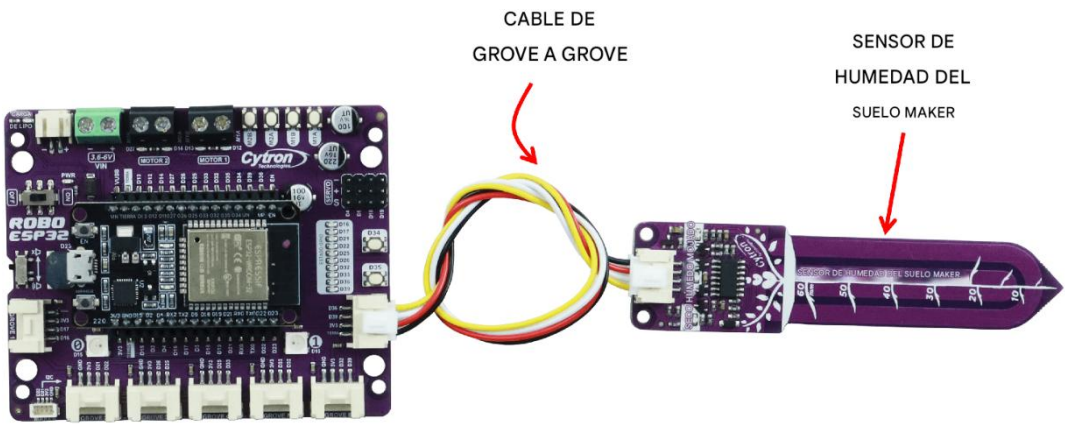


Puertos Grove	GPIO	ADC
Grove 3	D26	ADC2
	D25	ADC2
Grove 4	D21	ADC1
	D22	ADC1
Grove 5	D33	ADC1
	D32	ADC1
Grove 6	D32	ADC1
	D39	ADC1
Grove 7	D39	ADC1
	D36	ADC1

Nota: El puerto Grove 3 en el Robo ESP32 usa pines ADC2, que no se pueden usar cuando Wi-Fi está habilitado. Si su proyecto necesita Wi-Fi, es mejor usar los pines ADC1 disponibles en otros puertos Grove.

Para esta guía utilizaremos [el Maker Soil Module](#), que es un sensor analógico.

1. Conecte el módulo Maker Soil al puerto Grove 7 en el Robo ESP32, o elija cualquier puerto entre Grove 4 y Grove 7. Si su proyecto no requiere una conexión Wi-Fi, también puede utilizar el puerto 3 de Grove. La siguiente figura muestra la conexión entre el módulo y el Robo ESP32.



2. **Nota: El pin DIS es el pin de desactivación, donde una señal ALTA desactivará el sensor. Este código no utiliza este pin.**
3. Conecte su NodeMCU ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
4. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
5. Copia y pega el siguiente código en tu editor:

```
/*
DESCRIPCIÓN:
Este código de ejemplo utilizará Robo ESP32 para leer la entrada
analógica del módulo Maker Soil
y luego mostrar el resultado en el monitor serie. Este código también es
aplicable a cualquier sensor analógico.

CONEXIÓN:

-GROVE 7-      | Maker Soil
Robo ESP32     |
-----
GND            | GND
VCC            | VCC
D39            | DIS
D36            | OUT

AUTHOR   : Cytron Technologies Sdn Bhd
WEBSITE  : www.cytron.io
EMAIL    : support@cytron.io
*/

const int MakerSoil = 39;
int inputvalue = 0;
float readingvalue = 0;

void setup() {
    pinMode(MakerSoil, INPUT);
    Serial.begin(115200);
}

void loop() {
    // Leer el valor de entrada del suelo fabricante
    inputvalue = analogRead(MakerSoil);
}
```

```
//Convertir valor de entrada ADC, 3.3V veces 2 8 resolución
readingvalue = (inputvalue * 3.3 / 4096);

Serial.print("Input Value : ");
Serial.println(inputvalue);
Serial.print("Valor de lectura : ");
Serial.println(readingvalue);
Serial.println("-----");

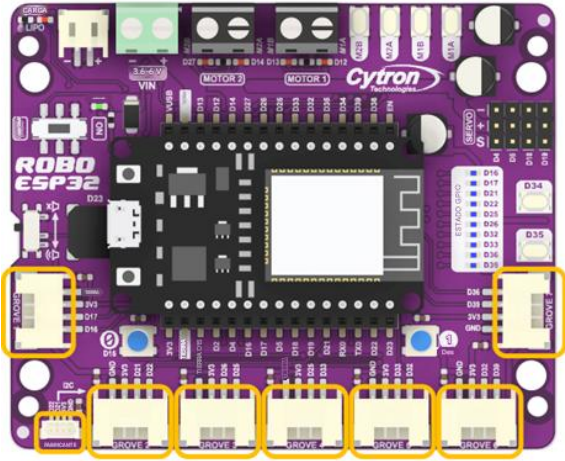
delay(1000);
}
```

[Ver Guía de inicio / Arduino / AnalogRead / AnalogRead.ino](#) entregado con ❤ [por mgithub](#)

5. Verifique y cargue el boceto en su NodeMCU ESP32.
6. Apile y encienda el interruptor de encendido de Robo ESP32. La salida esperada es la lectura del sensor que debe mostrarse en el monitor serie. Un voltaje más alto significa menos humedad.

Leer el valor digital

El Robo ESP32 cuenta con pines flexibles, etiquetados como Grove 1 a Grove 7, diseñados específicamente para leer los valores de los sensores digitales. Estos puertos permiten una integración perfecta con una variedad de sensores e interruptores, lo que le permite incorporar fácilmente entradas digitales en sus proyectos y mejorar la funcionalidad de su Robo ESP32.



Puertos Grove	GPIO	Función ADC/Pin
Grove 1	D17	TX2
	D16	RX2
Grove 2 /Maker	D21	SDA
	D22	SCL
Grove 3	D26	ADC2
	D25	ADC2
Grove 4	D21	ADC1
	D22	ADC1
Grove 5	D33	ADC1
	D32	ADC1
Grove 6	D32	ADC1
	D39	ADC1
Grove 7	D39	ADC1
	D36	ADC1

Compartido

Compartido

Para esta guía, utilizaremos un módulo genérico de [sensor ultrasónico HC-SR04](#). , que es un sensor digital.

1. Conecte el sensor ultrasónico al Grove 2 en el Robo ESP32, o seleccione otro puerto Grove según sus preferencias. La siguiente figura ilustra la conexión entre el módulo de sensor y el Robo ESP32.
2. Conecte su NodeMCU ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
3. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
4. Copia y pega el siguiente código en tu editor:

```
/*
DESCRIPCIÓN:
Este código de ejemplo utilizará Robo ESP32 para leer la entrada digital
del sensor ultrasónico
y luego mostrar el resultado en monitor serie. Este código también es
aplicable a cualquier sensor digital.
```

CONEXIONES

-GROVE 2-	Ultrasonic Sensor
Robo ESP32	

GND	GND
VCC	VCC
D26	TRIG
D25	ECHO

AUTHOR : Cytron Technologies Sdn Bhd

WEBSITE : www.cytron.io

EMAIL : support@cytron.io

TRADUCTOR: José Manuel Ruiz

*/

// Define números de pines

const int trigPin = 26; // Conectar al pin 26 desde Robo ESP32

const int echoPin = 25; // Conectar al pin 25 desde Robo ESP32

// Define variables

int distance; // Defines variables

long duration;

void setup() {

pinMode(trigPin, OUTPUT); // Establece el trigPin como una salida

pinMode(echoPin, INPUT); // Establece el echoPin como una entrada

Serial.begin(115200); // Inicia la comunicación en serie

}

void loop() {

 // Borra el trigPin

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

 // Establece el trigPin en el estado HIGH durante 10 microsegundos

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

 // Lee el echoPin, devuelve el tiempo de viaje de la onda sonora en microsegundos

 duration = **pulseIn**(echoPin, HIGH);

 // Cálculo de la distancia

 distance = duration * 0.034 / 2;

 // Imprime la distancia en el monitor serie

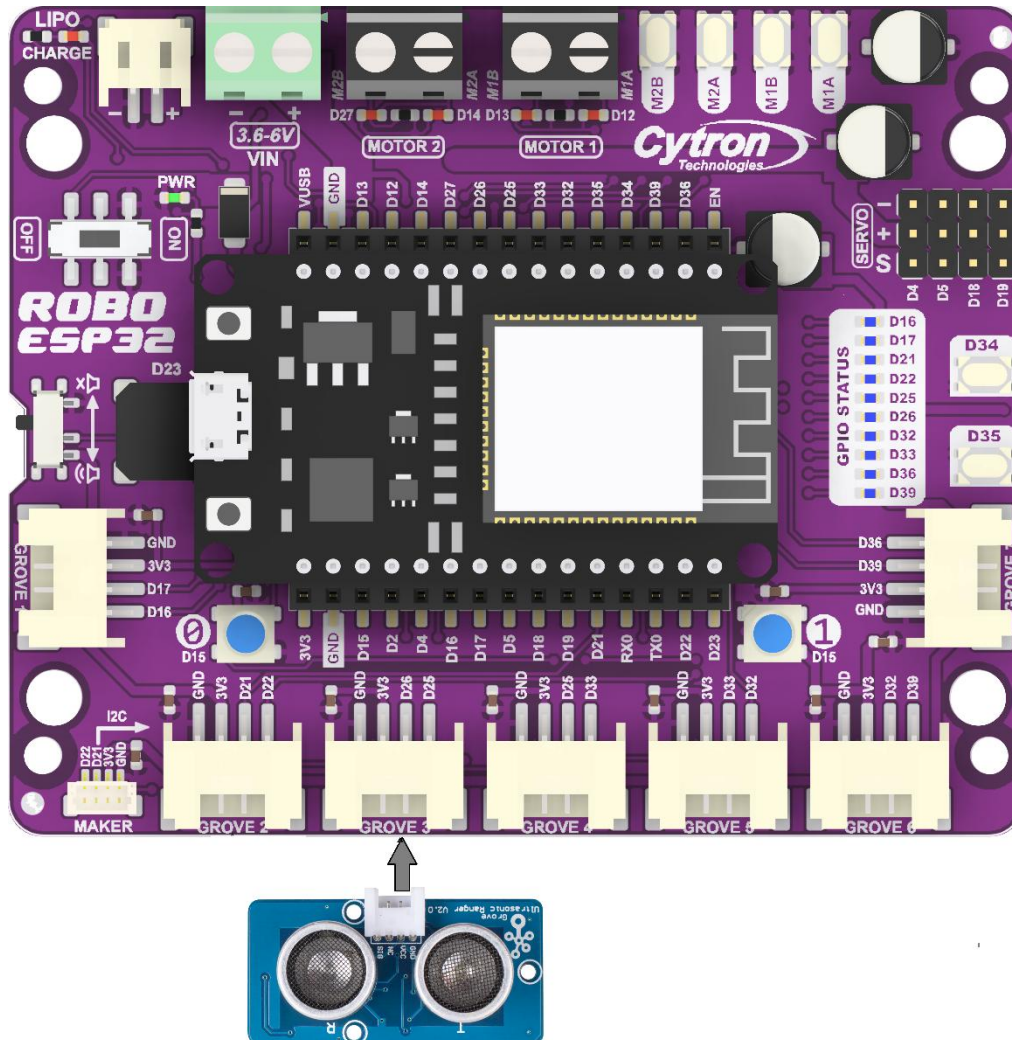
Serial.print("Distancia: ");


```

Serial.print(distance);
Serial.println(" cm");
}

```

Montaje:



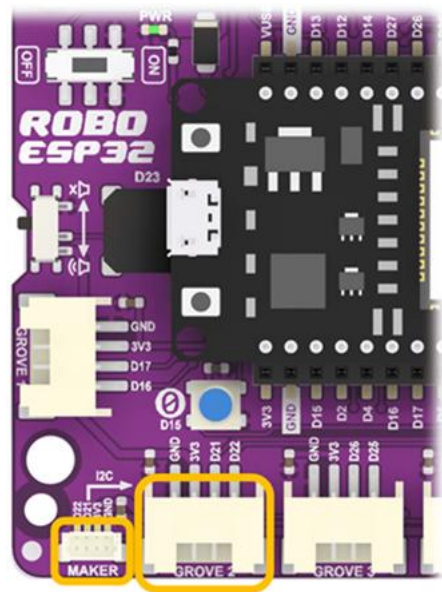
[Ver Guía de inicio/Arduino/DigitalRead/DigitalRead.ino](#) entregado con ❤️ [por emgithub](#)

5. Verifique y cargue el boceto en su NodeMCU ESP32.
6. Apile y encienda el interruptor de encendido de Robo ESP32. La lectura de salida esperada del sensor debe mostrarse en el monitor serie

Manejo de SSD1315 módulo OLED a través de I2C

El Robo ESP32, con sus versátiles puertos Grove y Maker, ofrece una plataforma conveniente para integrar varios sensores y módulos. Esta guía demostrará cómo conectar y utilizar el módulo OLED SSD1315 para mostrar texto en el Robo ESP32.

El Robo ESP32 proporciona múltiples pines I2C a través de sus puertos Grove, lo que permite opciones de conectividad flexibles. El puerto Maker, en particular, comparte los mismos pines I2C que Grove 2, lo que amplía aún más las posibilidades para dispositivos basados en I2C como el módulo OLED.



Puertos Grove	GPIO
Maker	D22
	D21
Grove 2	D21
	D22

Puerto de Grove

1. Conecte el módulo OLED al puerto Grove 2 mediante un cable Grove. La siguiente figura muestra la conexión entre el módulo y el Robo ESP32.
2. Conecte su NodeMCU ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
3. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
4. Copia y pega el siguiente código en tu editor:

Montaje:


```

#endif

U8G2_SSD1306_128X64_NONAME_F_SW_I2C u8g2(U8G2_R0, /* clock(SCL)*/ SCL,
/*data(SDA)*/ SDA, /* reset=*/U8X8_PIN_NONE);

void setup(void) {
  u8g2.begin();
}

void loop(void) {
  u8g2.clearBuffer();
  u8g2.setFont(u8g2_font_ncenB08_tr);

  // Adjusted coordinates for better positioning
  u8g2.drawStr(5, 15, "Hola mundo!");
  u8g2.drawStr(5, 30, "vale");

  u8g2.sendBuffer();
  delay(1000);
}

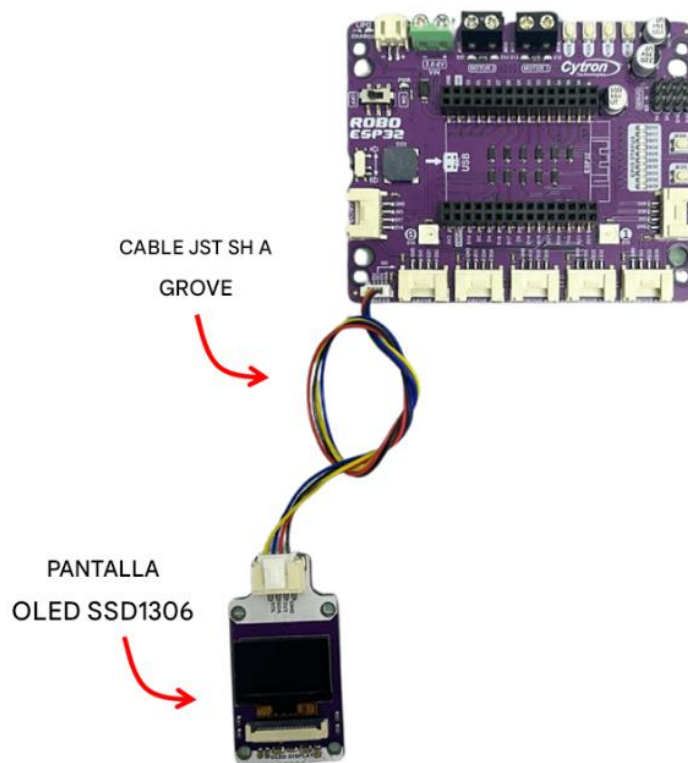
```

Ver Guía de inicio / Arduino / OLED / OLED.ino entregado con  **emgithub**

5. Verifique y cargue el boceto en su NodeMCU ESP32.
6. Apile y encienda el interruptor de encendido de Robo ESP32. El resultado esperado es que el módulo OLED muestre "Hola mundo" en la primera línea y "¡sí!" en la segunda línea.

Puerto Maker

1. Conecte el módulo OLED al puerto Maker mediante un cable JST SH a Grove o un cable JST SH a pines hembra. La siguiente figura muestra la conexión entre el módulo y el Robo ESP32.



2. Conecte su NodeMCU ESP32 a la computadora. Asegúrese de utilizar un cable de datos adecuado al programar la placa.
3. Abra el IDE de Arduino y asegúrese de que la placa y el puerto COM correctos estén seleccionados.
4. Utilice los mismos códigos que la guía anterior o puede copiar y pegar el siguiente código en su editor:
5. Verifique y cargue el boceto en su NodeMCU ESP32.
6. Apile y encienda el interruptor de encendido de Robo ESP32. El resultado esperado es que el módulo OLED muestre "Hola mundo" en la primera línea y "¡sí!" en la segunda línea.