

Contenido

- **Introducción ESP32**
- **Tarjeta ESP32 DEV KIT V1**
- **ESP32 y Arduino**
- **Simulador Wokwi**
- **GPIO, ADC y PWM**



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

ESP32



ESP32 es un **sistema en un solo chip** (SoC) de bajo costo y bajo consumo de energía con capacidades integradas de Wi-Fi y Bluetooth.

Es desarrollado por la empresa **Espressif Systems** y se basa en el procesador de doble núcleo **Xtensa LX6** con velocidades de reloj de hasta **240 MHz**.



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

ESP32



El ESP32 tiene una amplia gama de periféricos:

GPIO (Entrada/Salida de propósito general) que pueden ser utilizados para conectar pulsadores, ledes y otros dispositivos digitales.

ADC (Convertor analógico-digital) y **DAC** (Convertor digital-analógico) para la entrada y salida de señales analógicas.

PWM (Modulación de ancho de pulso) para generar señales que pueden ser usadas para controlar la velocidad de los motores o la intensidad de un led.

UART, SPI e I2C que son interfaces de comunicación para conectar dispositivos externos.

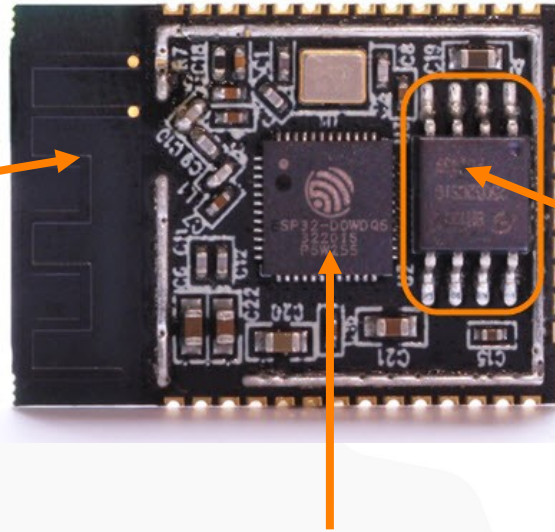
Sensores táctiles capacitivos para detectar el toque.



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

ESP32

Antena



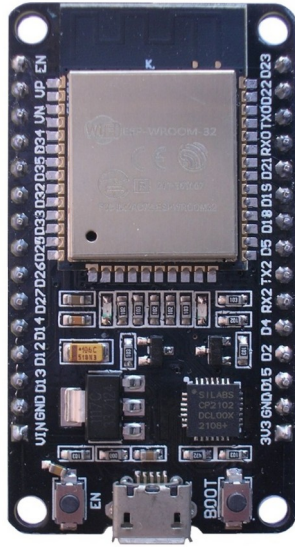
Memoria
Flash

ESP32



March 25th
day.arduino.cc
#ArduinoDay2023

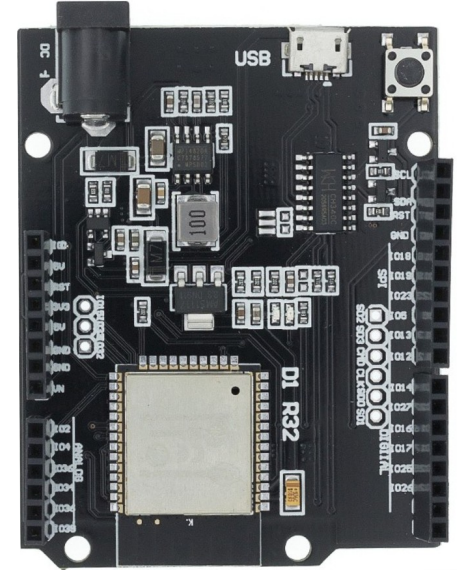
Tarjetas basadas en ESP32



ESP32 DevKit V1



ESP32-CAM

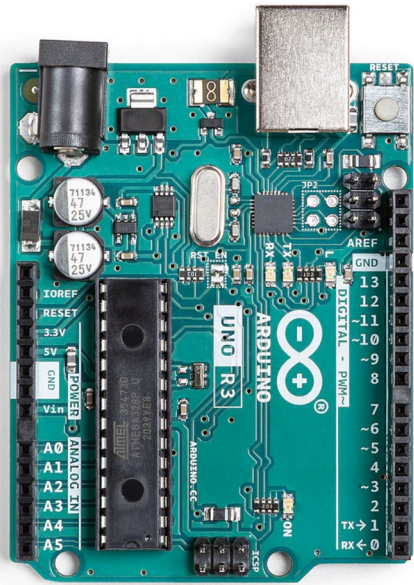


WeMos ESP32 D1



March 25th
day.arduino.cc
#ArduinoDay2023

Tarjetas Arduino vs ESP32

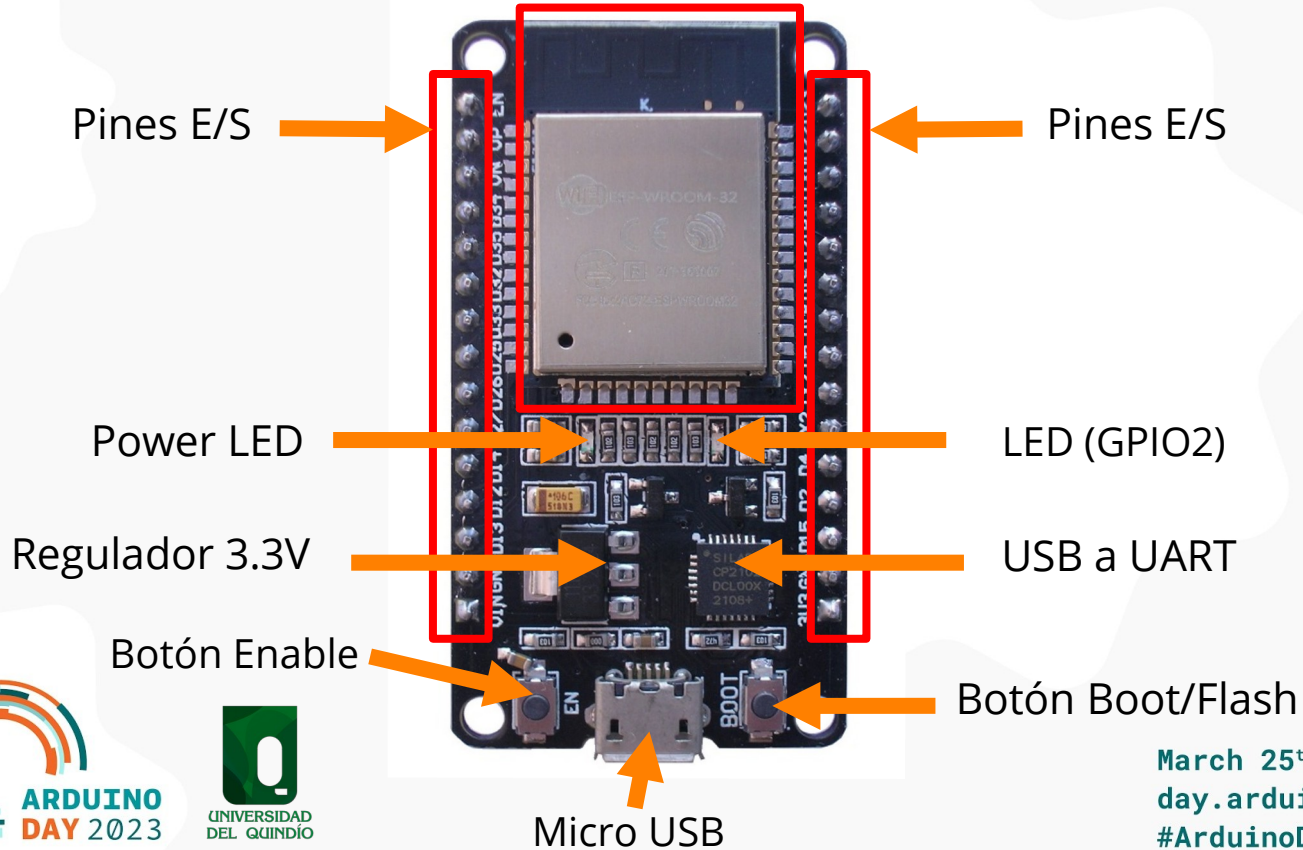


| Función | ESP32 | Arduino Uno |
|-------------------------|-------------------------|-------------------------------|
| Microcontrolador | Xtensa Dual Core 32-bit | ATmega328P 8-bit |
| Flash memory | 4MB | 32KB |
| SRAM | 520MB | 2KB |
| Clock | 240MHz | 16MHz |
| Interfaces inalámbricas | Wi-Fi, Bluetooth | Se necesitan módulos externos |



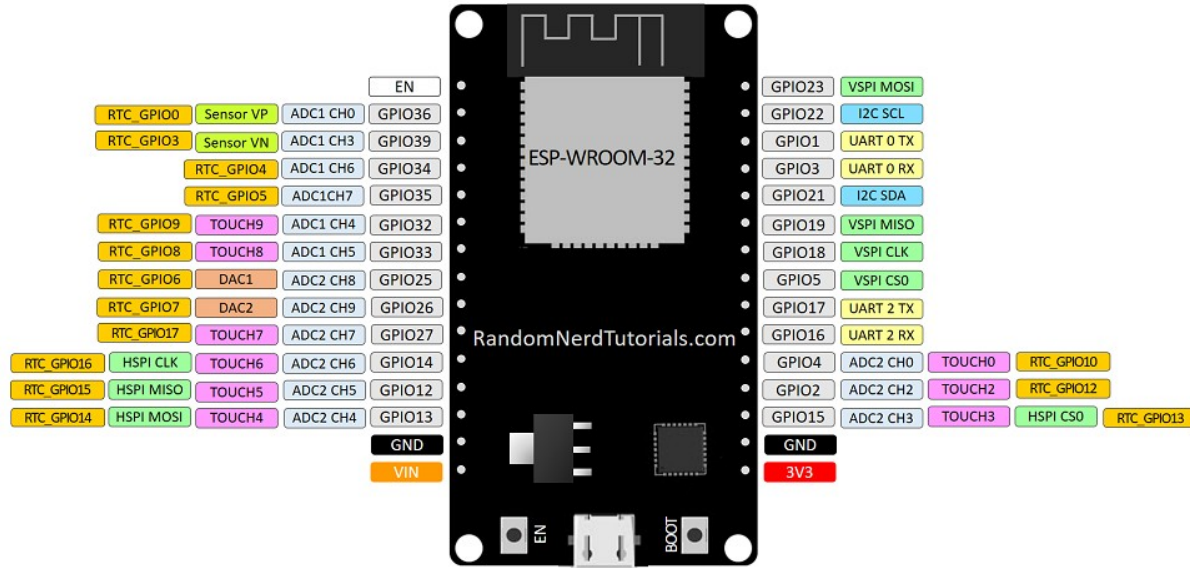
March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

ESP32 DEVKITV1



March 25th
day.arduino.cc
#ArduinoDay2023

ESP32 DEVKITV1 Pinout



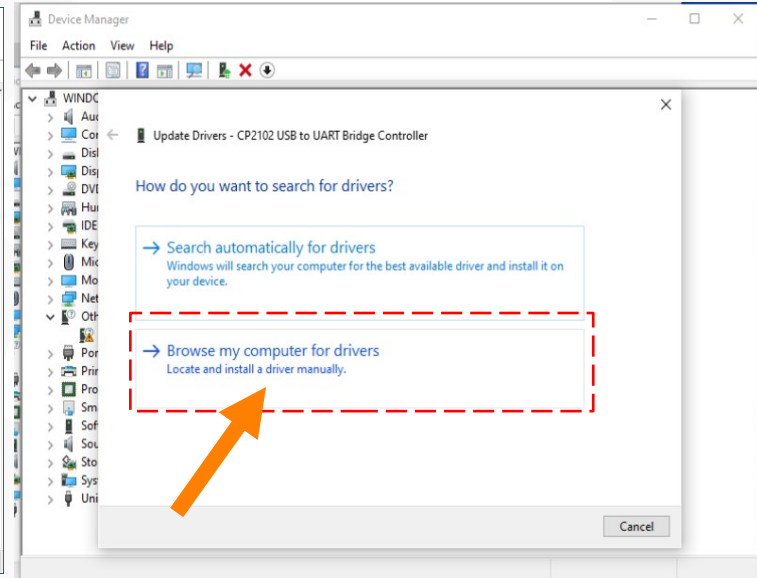
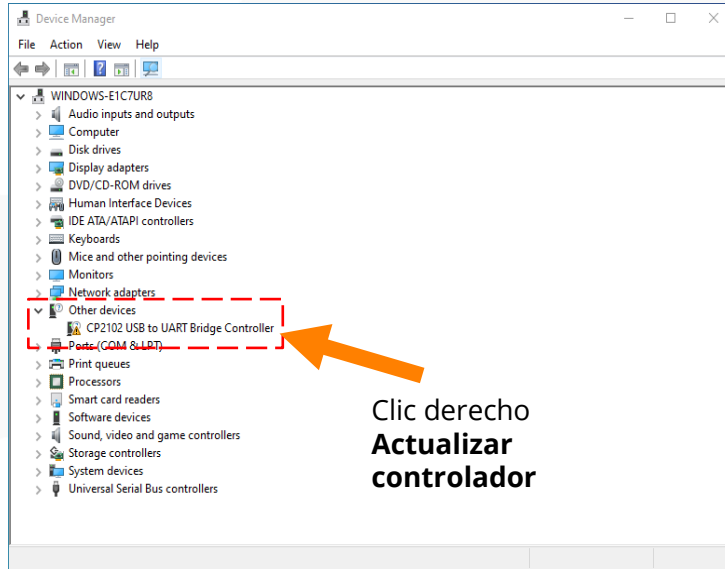
<https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>



March 25th
day.arduino.cc
#ArduinoDay2023

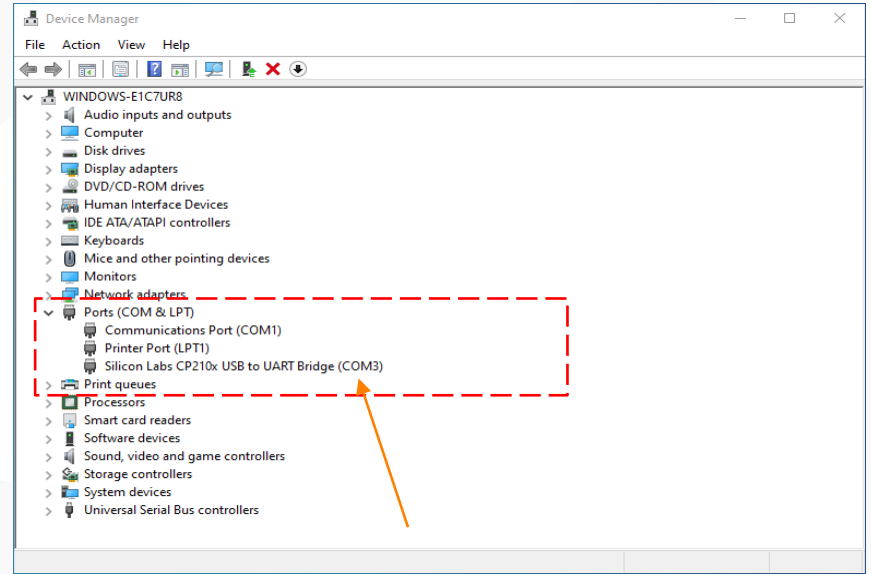
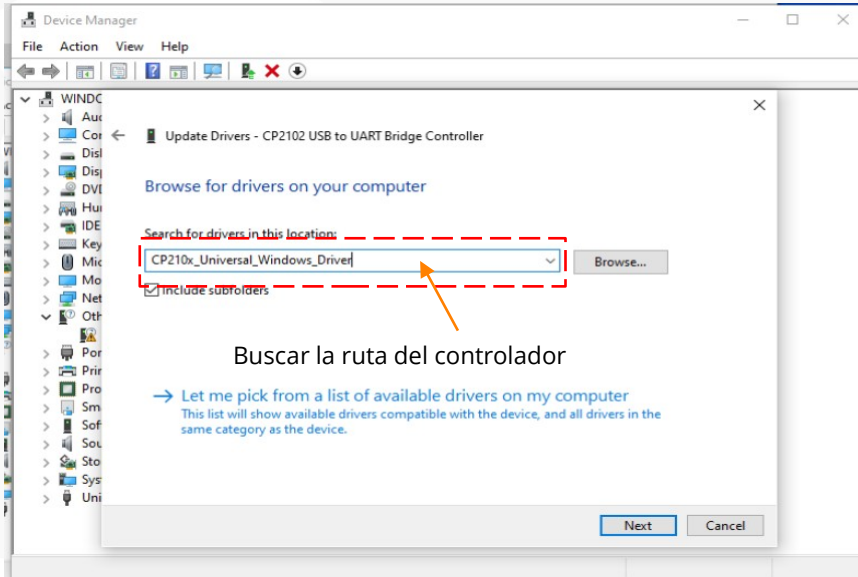
Instalación de controladores

1. Descargar: [CP210x USB to UART Bridge VCP Drivers](#)
2. Abrir el Administrador de dispositivos



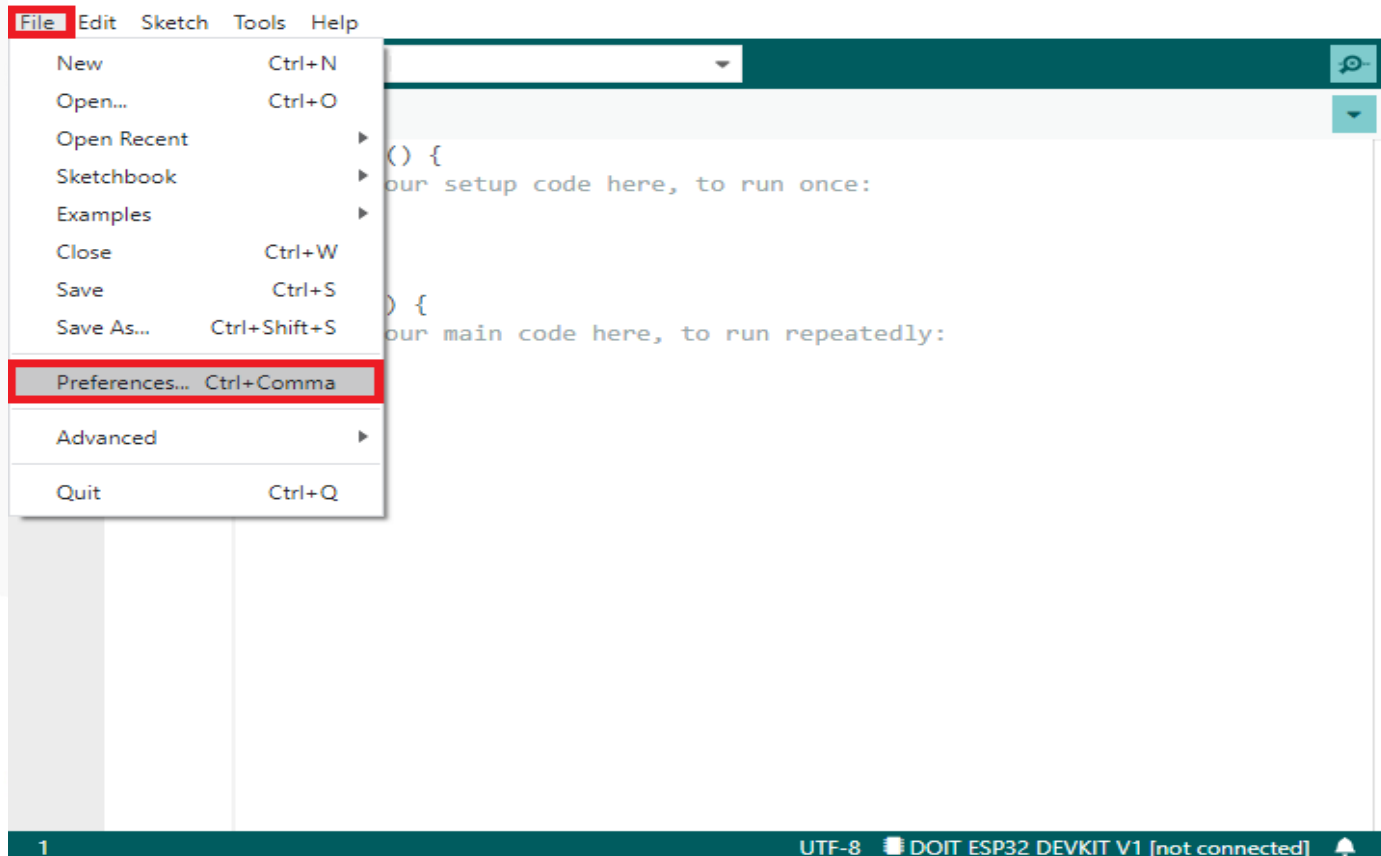
March 25th
day.arduino.cc
#ArduinoDay2023

Instalación de controladores



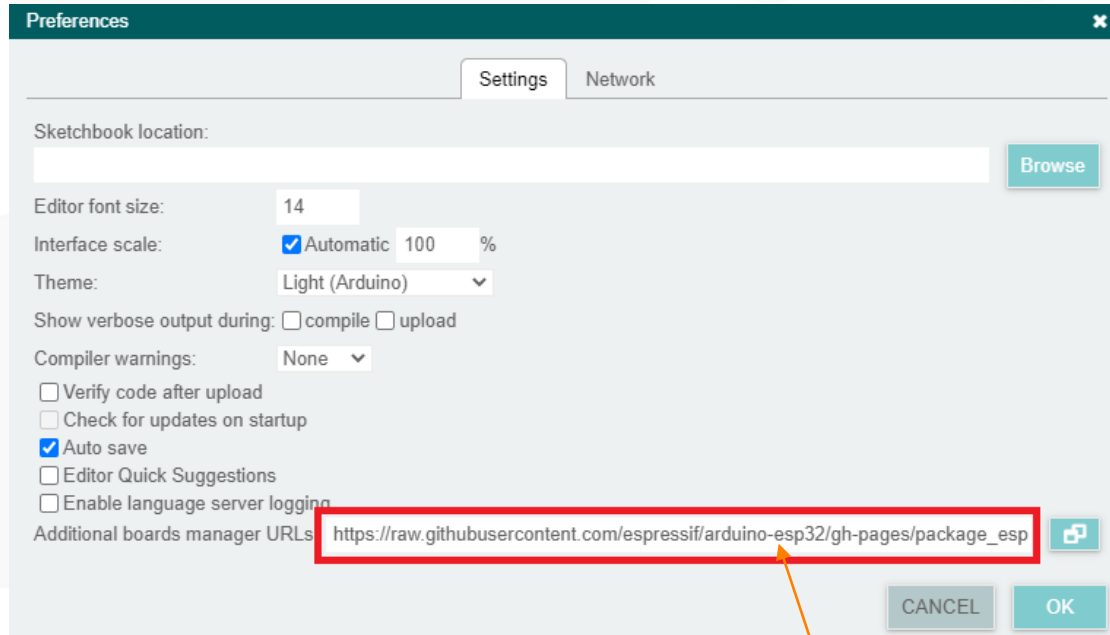
March 25th
day.arduino.cc
#ArduinoDay2023

ESP32 y Arduino IDE



March 25th
day.arduino.cc
#ArduinoDay2023

ESP32 y Arduino IDE



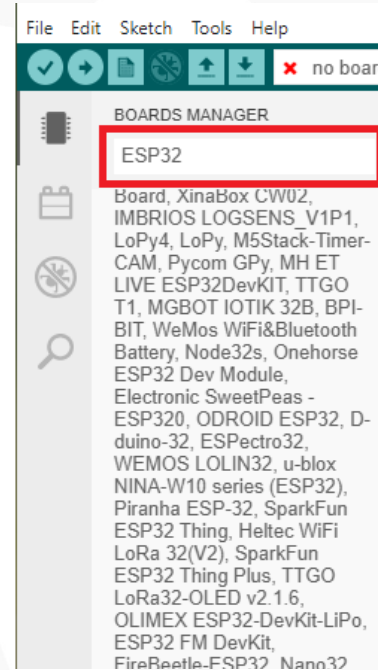
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json



March 25th
day.arduino.cc
#ArduinoDay2023

ESP32 y Arduino IDE

- 1) Seleccionar "**Herramientas**" en la barra de menú.
- 2) Seleccionar "**Placa**" y luego "**Gestor de tarjetas**".
- 3) Buscar "**ESP32**" en el cuadro de búsqueda y seleccionar la opción de "esp32" de Espressif Systems.
- 4) Clic en "**Instalar**" para instalar el soporte de ESP32.
- 5) Esperar a que se complete la instalación.
- 6) Seleccionar "**Herramientas**" en la barra de menú y luego seleccione la tarjeta ESP32 a usar.



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

Prueba básica (Blink)

// la función setup se ejecuta una sola vez cuando se presiona reset o se conecta la tarjeta

```
void setup() {  
  // inicializa el pin digital LED_BUILTIN como salida.  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

// la función loop se ejecuta una y otra vez por siempre

```
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // enciende el LED (HIGH es el nivel de voltaje 3.3V)  
  delay(1000); // espera por un segundo  
  digitalWrite(LED_BUILTIN, LOW); // apaga el LED colocando un voltaje LOW 0V  
  delay(1000); // espera por un segundo  
}
```



March 25th
day.arduino.cc
#ArduinoDay2023

Simulador Wokwi (<https://wokwi.com/>)

WOKWI
Simulate IoT Projects in Your Browser

[Discord Community](#) [Facebook Group](#)

We're adding new stuff all the time, want us to tell You?

Your email address [Stay Updated](#)

Featured Simulation Projects

- Simon Game with Score**
- Nano Pong**
- 32x32 LED Matrix Tunnel**



March 25th
day.arduino.cc
[#ArduinoDay2023](https://twitter.com/ArduinoDay2023)

ESP32 GPIO

GPIO (General Purpose Input/Output) son pines que se pueden configurar para ser **entradas o salidas digitales**. HIGH corresponde a 3.3V y LOW corresponde a 0V.

El ESP32 cuenta con 40 GPIO, pero no todos se pueden usar y tampoco están disponibles en las tarjetas de desarrollo.

Los pines **2, 4, 5 y 13 al 33** se pueden usar sin ningún problema.

Los pines **34 al 39** solo se pueden usar como entrada.

Los pines del **6 al 11** están conectados al SPI Flash.

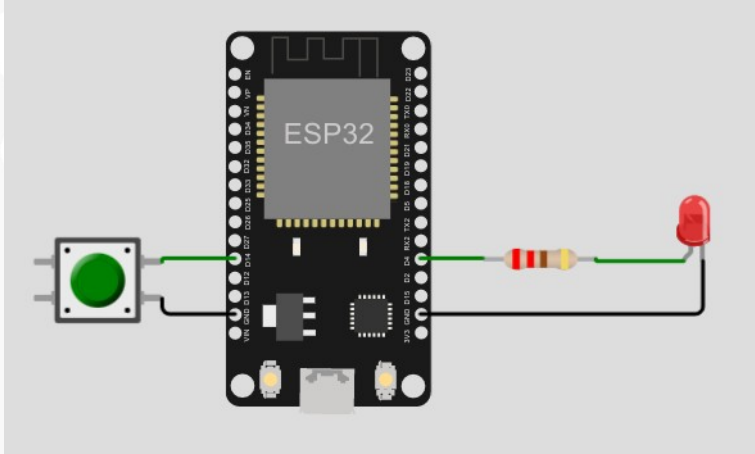
Los pines **0, 1, 3 y 12** tienen funciones asociadas al boot.

Todos los GPIOs cuentan con una o más funciones adicionales. Todos los GPIOs pueden ser configurados como interrupciones.



March 25th
day.arduino.cc
#ArduinoDay2023

ESP32 GPIO



```
const int led = 4;  
const int button = 14;
```

```
void setup() {  
  // inicializa el pin digital 4 como salida.  
  pinMode(led, OUTPUT);  
  // inicializa el pin digital 14 como entrada  
  // con resistencia de pull-up habilitada.  
  pinMode(button, INPUT_PULLUP);  
}
```

```
void loop() {  
  // lee el valor del pulsador  
  // (sin presionar HIGH, presionado LOW)  
  int value = digitalRead(button);  
  // escribe el valor leído en el led  
  digitalWrite(led, value);  
}
```

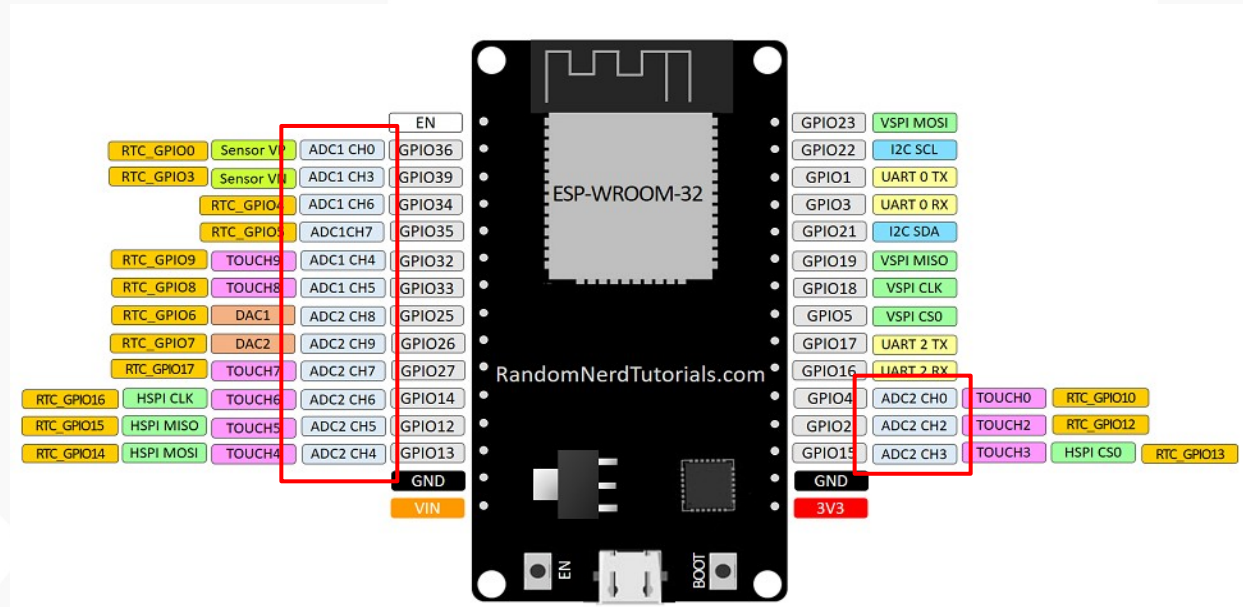


March 25th
day.arduino.cc
#ArduinoDay2023

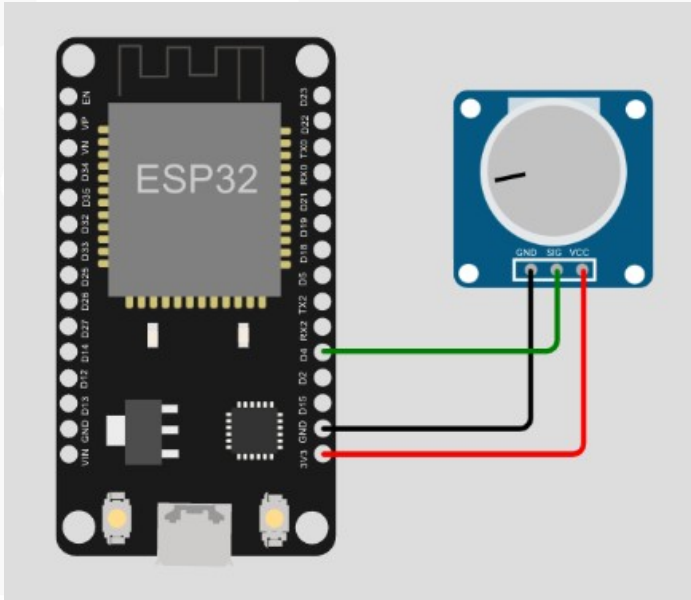
ESP32 ADC

El ESP32 tiene **dos ADC (Convertor Analógico a Digital)** integrados de **12 bits**, que se pueden utilizar para medir señales analógicas.

El rango de entrada analógica del ADC del ESP32 es de 0 a 3.3V, y el valor de la medición se convierte en un número entero de 12 bits que varía de 0 a 4095.



ESP32 ADC



```
const int pot = 4;
```

```
void setup() {  
  // configuración puerto serie  
  Serial.begin(115200);  
}
```

```
void loop() {  
  // lectura del voltaje que entrega el potenciómetro  
  int adc_value = analogRead(pot);  
  // convierte el valor leído a voltaje  
  float voltage = (adc_value * 3.3) / 4095.0;  
  
  // imprime el valor en el monitor serie  
  Serial.printf("%d, %0.2fV\n", adc_value, voltage);  
  // espera 300 ms para una nueva lectura  
  delay(300);  
}
```

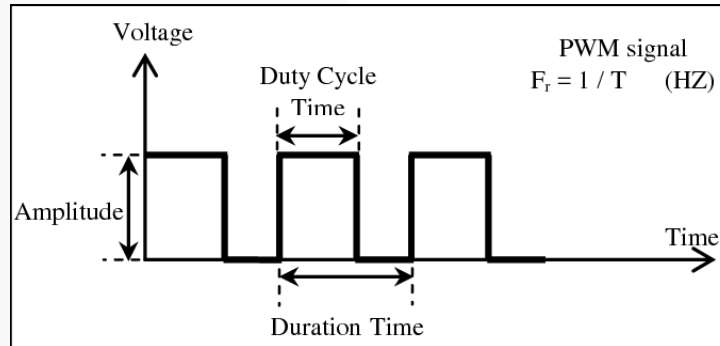


March 25th
day.arduino.cc
#ArduinoDay2023

ESP32 PWM

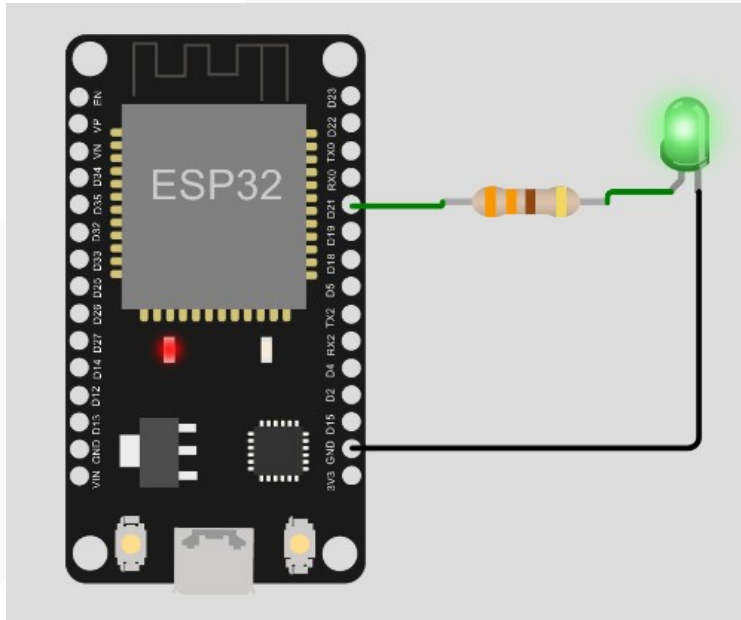
El ESP32 tiene 16 canales **PWM (Pulse Width Modulation)** que le permite generar señales PWM en cualquier pin que pueda ser configurado como salida.

Una señal PWM tiene una frecuencia de trabajo y un ciclo útil que hace referencia al tiempo que permanece en alto la señal durante un periodo.



March 25th
day.arduino.cc
#ArduinoDay2023

ESP32 PWM



```
const int led = 21;
```

```
void setup() {  
  // inicializa el pin 21 como salida  
  pinMode(led, OUTPUT);  
}
```

```
void loop() {  
  // ciclo de 0 a 255  
  for(int duty = 0; duty < 256; duty++ ) {  
    // cambio del ciclo útil de la señal PWM  
    analogWrite(led, duty);  
    // se espera 20 ms  
    delay(20);  
  }  
}
```



March 25th
day.arduino.cc
#ArduinoDay2023

