

Funciones básicas del ESP32 en Arduino

Resumen de funciones esenciales y
específicas para el ESP32

Introducción: ¿Qué es el ESP32?

El **ESP32** es un microcontrolador de bajo costo y alto rendimiento desarrollado por **Espressif Systems**.

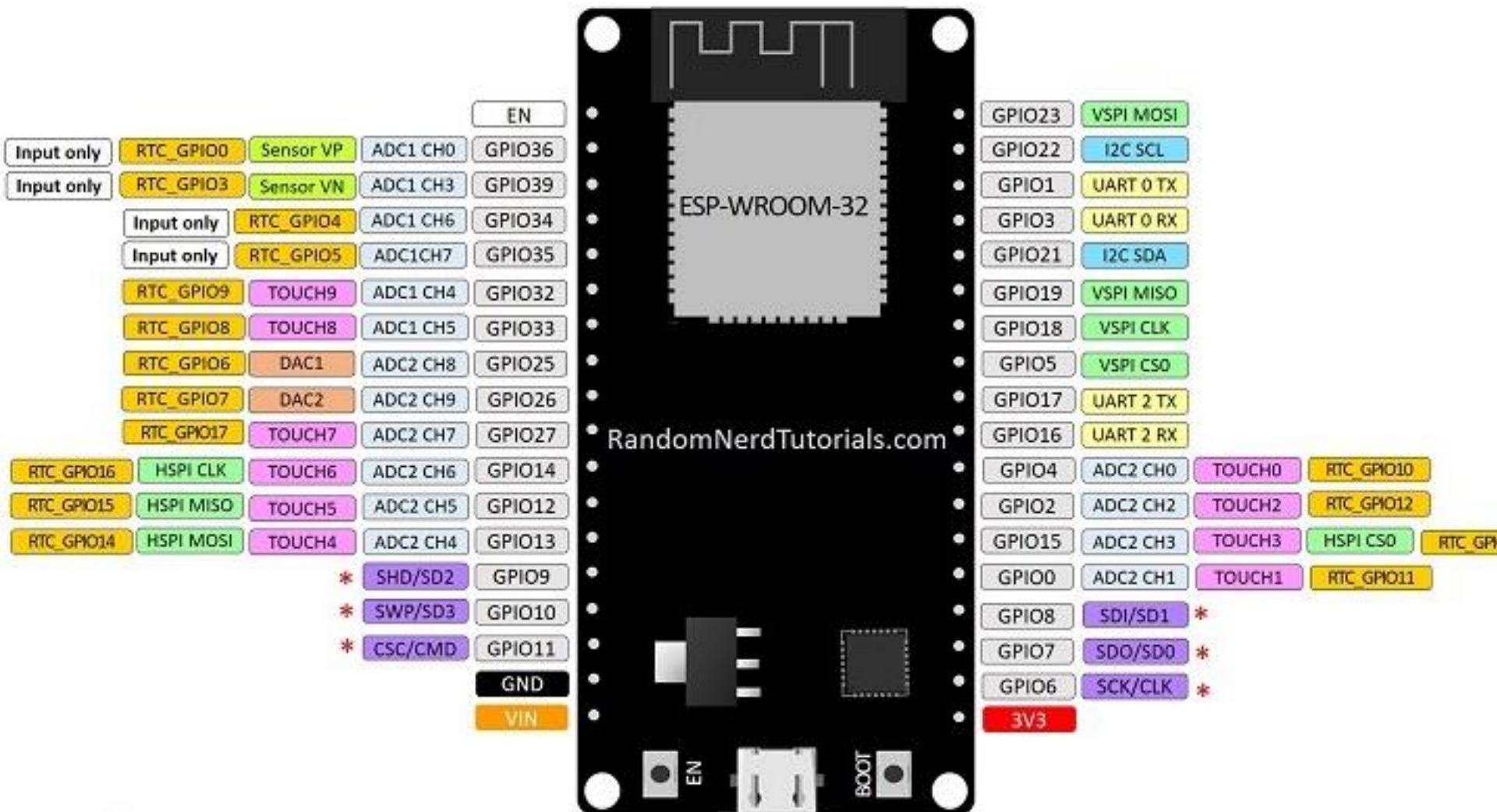
Destaca por integrar un **procesador de doble núcleo**, conectividad **Wi-Fi** y **Bluetooth** (clásico y BLE), además de múltiples periféricos como entradas y salidas digitales, conversores analógico-digital (ADC), salidas PWM, interfaces I2C, SPI y UART.

Su arquitectura lo convierte en una plataforma ideal para proyectos de **Internet de las Cosas (IoT)**, automatización, domótica y dispositivos portátiles. Gracias a su compatibilidad con el **lenguaje Arduino**, es accesible tanto para principiantes como para desarrolladores avanzados que buscan prototipar y desplegar soluciones inteligentes.



ESP32 DEVKIT V1 - DOIT

version with 36 GPIOs



* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and CSC/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.

Funciones básicas obligatorias

Función	Descripción
setup()	Se ejecuta una sola vez al iniciar el ESP32 o al reiniciarse. Sirve para configurar pines, iniciar comunicaciones, etc.
loop()	Se ejecuta en un bucle infinito después de setup(). Aquí va la lógica principal del programa.



```
1. void setup() {  
2.   pinMode(2, OUTPUT);    // Configura el pin 2 como salida  
3. }  
  
4. void loop() {  
5.   digitalWrite(2, HIGH);  // Enciende  
6.   delay(1000);           // Espera 1 segundo  
7.   digitalWrite(2, LOW);  // Apaga  
8.   delay(1000);           // Espera 1 segundo  
9. }
```

Funciones de manejo de pines



Estas funciones permiten interactuar con los pines de entrada y salida:

Función	Uso
<code>pinMode(pin, modo)</code>	Configura un pin como <code>INPUT</code> , <code>OUTPUT</code> o <code>INPUT_PULLUP</code> .
<code>digitalWrite(pin, valor)</code>	Envía un valor <code>HIGH</code> o <code>LOW</code> a un pin digital.
<code>digitalRead(pin)</code>	Lee el valor (<code>HIGH</code> o <code>LOW</code>) de un pin digital.
<code>analogWrite(pin, valor)</code>	Genera una señal PWM (0–255 por defecto, en ESP32 hasta 0–255 o 0–4095 según resolución).
<code>analogRead(pin)</code>	Lee un valor analógico de un pin ADC (0–4095 en ESP32 por defecto).

Funciones de tiempo

Función	Descripción
<code>delay(ms)</code>	Pausa en milisegundos.
<code>millis()</code>	Devuelve el tiempo en milisegundos desde que se encendió el ESP32.
<code>micros()</code>	Tiempo en microsegundos desde que se encendió.
<code>delayMicroseconds(us)</code>	Pausa en microsegundos.



Comunicación serie

Función	Uso
<code>Serial.begin(baudrate)</code>	Inicia la comunicación serie a cierta velocidad (ej. 115200).
<code>Serial.print(valor)</code>	Envía datos al monitor serie sin salto de línea.
<code>Serial.println(valor)</code>	Envía datos al monitor serie con salto de línea.
<code>Serial.available()</code>	Cantidad de datos disponibles para leer.
<code>Serial.read()</code>	Lee un byte recibido.

