

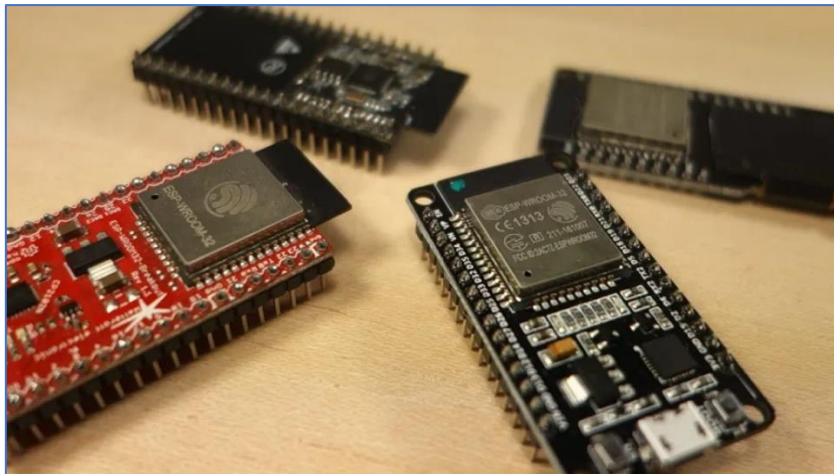
CARRERA: TECNICATURA SUPERIOR EN TELECOMUNICACIONES

MATERIA: ELECTRONICA MICROCONTROLADA

DOCENTES:

- JORGE E. MORALES, INGENIERO ELECTRICISTA ELECTRÓNICO (UNIVERSIDAD NACIONAL DE CÓRDOBA)
- C. GONZALO VERA, TECNICO SUPERIOR EN MECATRONICA (U.T.N.)

ESP32



CORDOBA, 29 DE JULIO DEL 2022.-

GRUPO N°2:

- DARIO ARRIOLA
- MARCOS JULIAN FINES
- DANIEL RODRIGUEZ
- NATALIA GALLIANI
- JEREMIAS CASTRO
- CARLA ARGENTINA WAYAR

El ESP32 es el sucesor del ESP8266, tiene más GPIO con más funcionalidades en comparación con el ESP8266.. Combina capacidades inalámbricas Wi-Fi y Bluetooth y dual-core.

Con el ESP32 puede decidir qué pines son UART, I2C o SPI; solo necesita configurar eso en el código. Esto es posible gracias a la característica de multiplexación del chip ESP32 que permite asignar múltiples funciones al mismo pin. Si no los establece en el código, los pines se utilizarán de manera predeterminada, como se muestra en la figura a continuación (la ubicación del pin puede cambiar según el fabricante).

I. ESPECIFICACIONES TECNICAS:

| | |
|--|--|
| Numero de núcleos | 2 (doble núcleo) |
| Wifi | 2,4 GHz hasta 150 Mbits/s |
| Bluetooth | BLE (Bluetooth Low Energy) y Bluetooth heredado |
| Arquitectura | 32 bits |
| Frecuencia de reloj | Hasta 240 MHz |
| RAM | 512KB |
| PINES | 30 o 36 (depende del modelo) |
| Periféricos | interfaz periférico con DMA que incluye táctil capacitivo ADC (convertidor analógico a digital) DAC (convertidor de digital a analógico) I ² C (Circuito Inter-Integrado) UART (receptor/transmisor asíncrono universal) SPI (interfaz periférica en serie) I ² S (Sonido Interchip Integrado) RMII (interfaz independiente de medios reducidos) PWM (modulación de ancho de pulso). |
| Bajo consumo de energía. - | |
| Seguridad: aceleradores de hardware para AES y SSL/TLS.- | |

II. TIPOS DE VERSIONES:

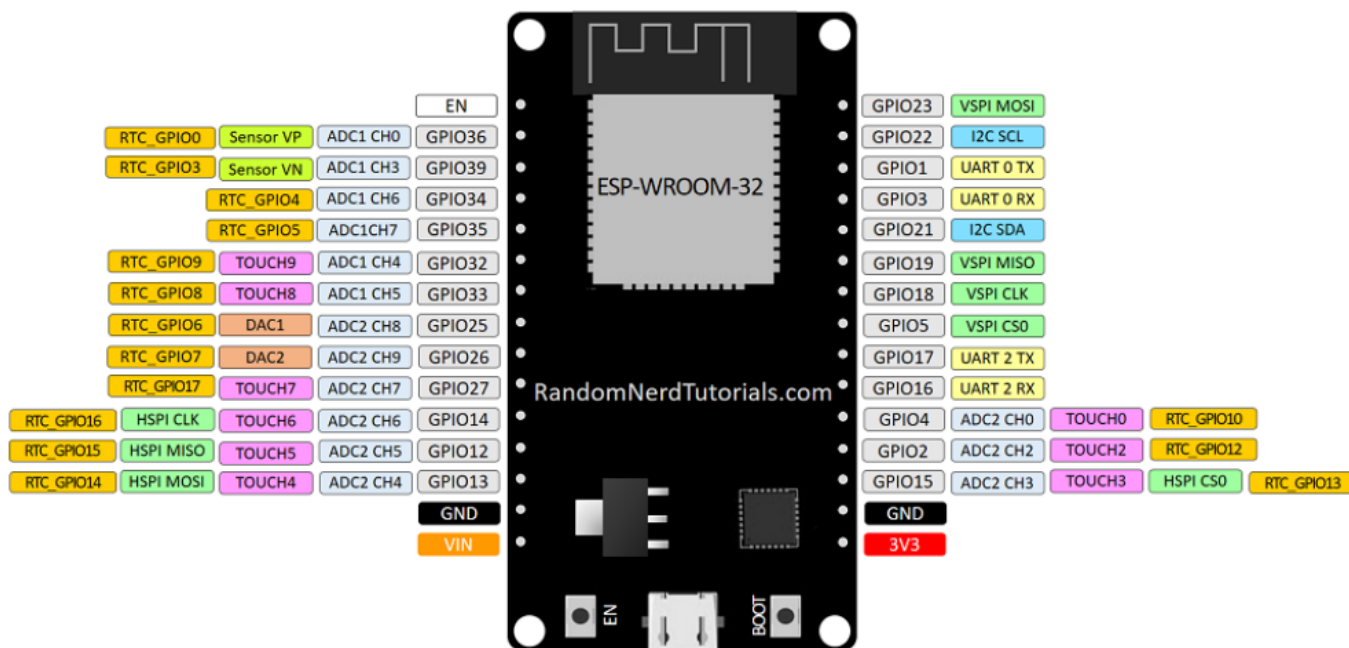


TIPOS DE PLACAS

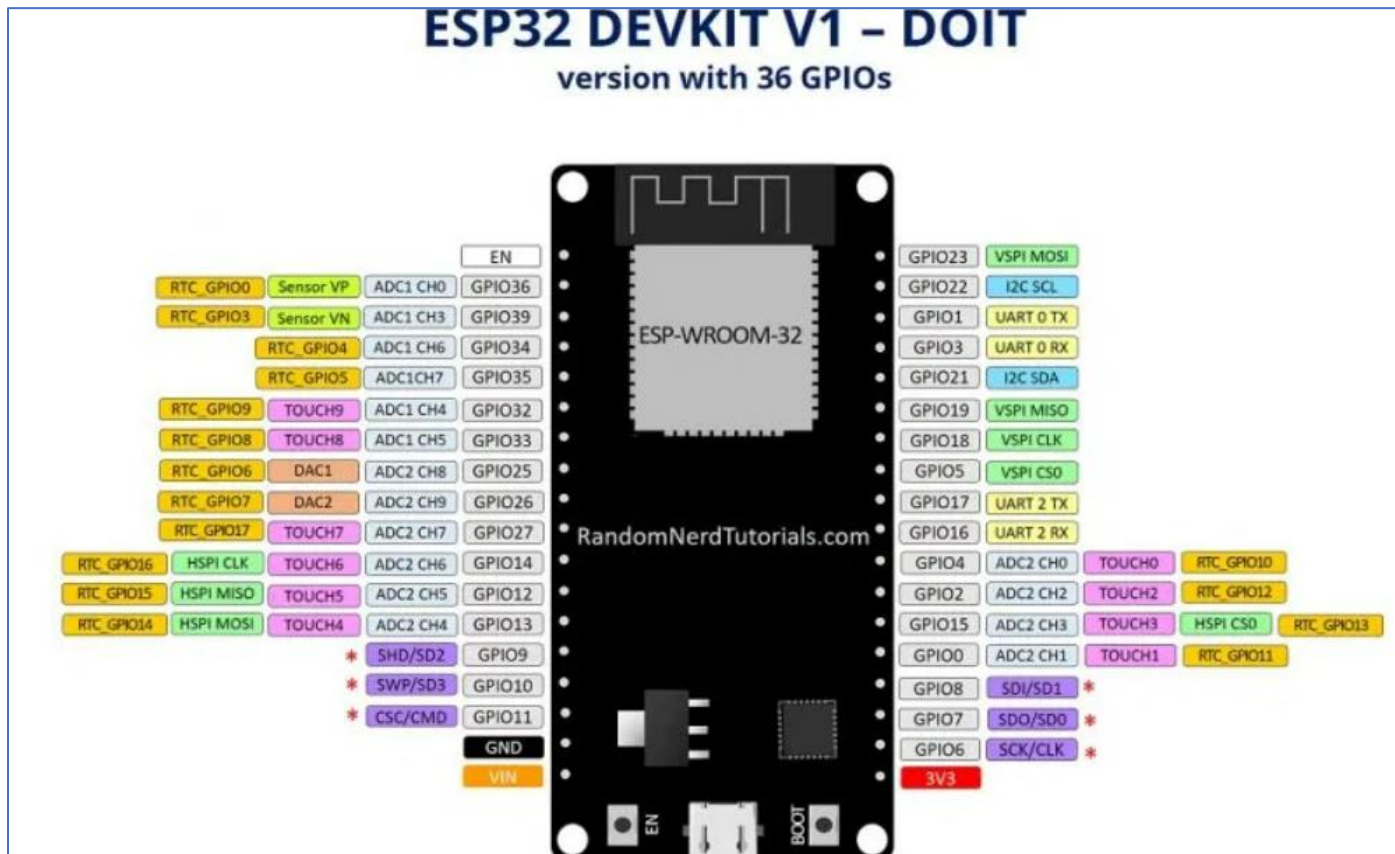
VERSION DE 30 PINES

ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



VERSION DE 36 PINES



- **CONFIGURACIÓN DE PINES Y NÚMERO DE PINES:** Para usar correctamente el ESP32 en sus proyectos, debe tener acceso al pinout de la placa (como un mapa que muestra qué pin corresponde a qué GPIO y sus características). Así que asegúrese de tener acceso al pinout de la placa que está obteniendo. De lo contrario, puede terminar usando el ESP32 incorrectamente. Recomendamos consultar nuestra Guía de referencia de ESP32 GPIO que muestra cómo usar los ESP32 GPIO correctamente.
- **INTERFAZ USB A UART Y CIRCUITO REGULADOR DE VOLTAJE:** La mayoría de las placas de desarrollo con funciones completas tienen estas dos funciones. Esto es importante para conectar fácilmente el ESP32 a su computadora para cargar el código y aplicar energía.
- **BOTONES BOOT Y RESET** para poner la placa en modo intermitente o resetear (reiniciar) la placa.
- **CONECTOR DE BATERÍA.** Si desea alimentar su ESP32 con baterías, hay placas de desarrollo que vienen con conectores para baterías LiPo; esto puede ser más útil. También puede alimentar un ESP32 "normal" con baterías a través de los pines de alimentación.
- **CARACTERÍSTICAS ADICIONALES.** Hay placas de desarrollo ESP32 con características adicionales como una pantalla OLED , un módulo LoRa , un módulo SIM800 (para GSM y GPRS), un soporte de batería o una cámara.

NOTA: Los pines resaltados en verde están bien para usar. Los resaltados en amarillo están bien para usar, pero debe prestar atención porque pueden tener un comportamiento inesperado principalmente en el arranque. No se recomienda utilizar los pines resaltados en rojo como entradas o salidas.

III. ENTORNO DE PROGRAMACION:

El ESP32 se puede programar en diferentes entornos de programación. Puedes usar:

- IDE DE ARDUINO
- ESPRESSIF IDF (MARCO DE DESARROLLO DE IOT)
- MICROPITÓN
- JAVASCRIPT
- LUA

Compatible con Arduino IDE: puede programar el ESP32 con Arduino IDE utilizando el núcleo de Arduino.(Instrucciones de instalación de Windows, Mac OS X y Linux). También puede usar otros IDE para programar el ESP32 con el núcleo de Arduino (como VS Code con la extensión PlatformIO , por ejemplo).

Compatible con MicroPython : puede programar el ESP32 con firmware MicroPython (Comience con MicroPython en ESP32).-