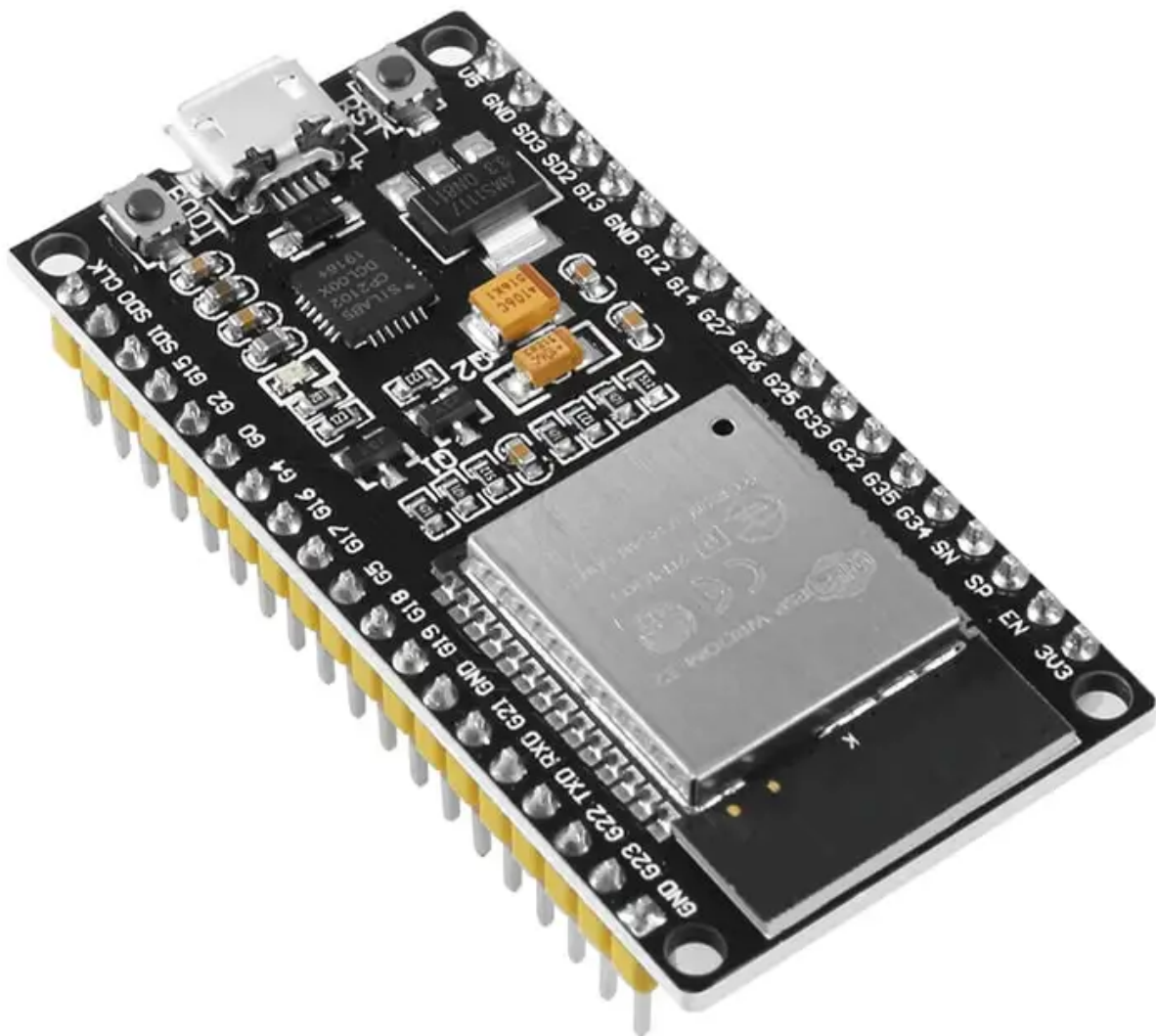


# PRACTICA 1 : BLINK

---

El objetivo de la practica es producir el parpadeo periodico de un led. Se utilizara la salida serie para depurar el programa

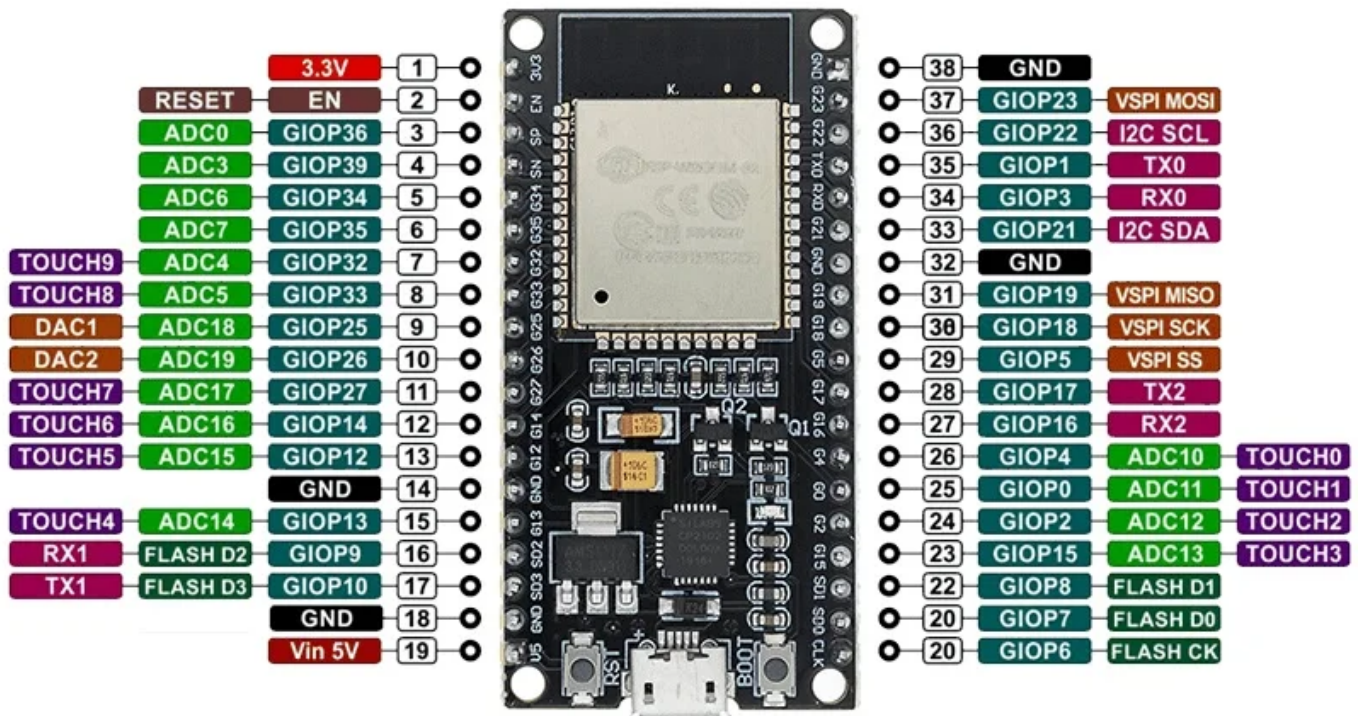
El microcontrolador que utilizaremos es el ya comentado en la introducción **ESP32**



La distribucion de pines de este procesador es el siguiente

# PINOUT

## ESP32 38 PINES ESP WROOM 32



## FUNCIONALIDAD DE LA PRACTICA

1. Iniciar pin de led como salida
2. bucle infinito
  - encender led
  - espera de 500 milisegundos
  - apagar led
  - espera de 500 milisegundos

## Código básico

```
#define LED_BUILTIN 2
#define DELAY 500
```

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(DELAY);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(DELAY);  
}
```

## Referencia

- <https://github.com/schacon/blink>

## TRABAJOS Y PREGUNTAS

1. Generar el programa y subir el código al github de cada uno
2. Modificar el programa para que incluya el envío de datos (ON y OFF) al puerto serie. Añadir la inicialización del puerto serie y el envío cada vez que cambia el estado del led
  - Iniciar pin de led como salida
  - Iniciar el terminal serie
  - bucle infinito
    - encender led
    - sacar por puerto serie mensaje *ON*
    - espera de 1000 milisegundos
    - apagar led
    - sacar por puerto serie mensaje *OFF*
    - espera de 1000 milisegundos

Sugerencias. Utilizar las funciones:

```
> Serial.begin(115200);  
  
> Serial.println("ON");  
  
> Serial.println("OFF");
```

<https://electropeak.com/learn/getting-started-with-platformio-ide-to-program-esp32/>

3. Modificar el programa para que actúe directamente sobre los registros de los puertos de entrada y salida

Sugerencias. Utilizar líneas del tipo:

```
> uint32_t *gpio_out = (uint32_t *)GPIO_OUT_REG;    // Establecer un puntero al registro de I/O

> *gpio_out |= (1 << 2); // Activar el bit correspondiente al pin 2

> *gpio_out ^= (1 << 2); // Alternar el estado del bit correspondiente al pin 2
```

4. Eliminar los delay modificar el pin de salida a uno cualquiera de los que estan disponibles i medir con el osciloscopio cual es la màxima frecuencia de apagado encendido que permite el microcontrolador. Medir la frecuencia en estos cuatro casos:
  - Con el envio por el puerto série del mensaje i utilizando las funciones de Arduino
  - Con el envio por el puerto série y accedirendo directamente a los registros
  - Sin el envio por el puerto série del mensaje i utilizando las funciones de Arduino
  - Sin el envio por el puerto série y accedirendo directamente a los registros
5. Generar un informe fichero informe.MD ( markdown ) donde se muestre el codigo, un diagrama de flujo y un diagrama de tiempos
6. Responder a la siguiente pregunta en el programa que se ha realizado cual es el tiempo libre que tiene el procesador ?

## EJERCICIOS voluntarios DE MEJORA DE NOTA

---

Elejir entre cualquiera de los siguientes:

- leer el valor de un convertidor A/D de entrada ; sacarlo por el puerto serie y sacar el mismo valor por otro pin D/A
  - <https://randomnerdtutorials.com/esp32-adc-analog-read-arduino-ide/>
- Leer el valor del sensor de temperatura interno y sacarlo por el puerto serie
  - <https://gist.github.com/xxlukas42/7e7e18604f61529b8398f7fcc5785251>

El resultado se ha de subir al github de cada uno y realizar un informe .MD