

## PROYECTO: SISTEMA DE CONTROL Y MONITOREO DISTRIBUIDO PARA LA GESTIÓN DEL AGUA

**Tarea:** uh2tk2

**Título:** Configuración del Entorno de Desarrollo

**Objetivos:**

- Instalar y configurar todas las herramientas necesarias para el desarrollo en ESP32 (IDE, bibliotecas, etc.).
  - Verificar la correcta conexión y comunicación entre el entorno de desarrollo y el hardware ESP32.
  - Documentar el proceso de configuración para futuros desarrollos.
  - Acompañar el documento con un video explicando los paso a paso.
- 

### Índice:

#### [1. Instalación del IDE y PlatformIO](#)

- [1.1 Descargar e Instalar Visual Studio Code](#)
- [1.2 Instalar PlatformIO en Visual Studio Code](#)

#### [2. Configuración de Bibliotecas Necesarias para ESP32](#)

- [2.1 Creación de un Nuevo Proyecto](#)
- [2.2 Instalación de Bibliotecas Necesarias](#)

#### [3. Conexión del ESP32 a la PC y Verificación de Comunicación](#)

- [3.1 Conexión Física](#)
- [3.2 Verificación de la Conexión](#)
- [3.3 Creación y Ejecución de un Código Simple](#)

#### [4. Circuito y conexión de ejemplo](#)

#### [5. Computacion, carga del código y ejecución](#)

- [5.1. Configuración del proyecto](#)
- [5.2. Compilación y carga \(BUILD | UPLOAD\)](#)

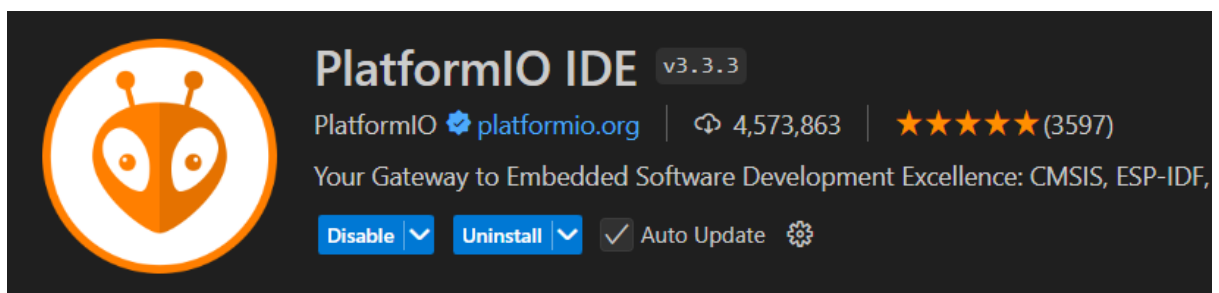
## 1. Instalación del IDE y PlatformIO

### 1.1 Descargar e Instalar Visual Studio Code

1. Visita la [página oficial de Visual Studio Code](#) y descarga la versión correspondiente a tu sistema operativo (Windows, macOS, o Linux).
2. Sigue las instrucciones de instalación para tu sistema operativo.

### 1.2 Instalar PlatformIO en Visual Studio Code

1. Abre Visual Studio Code.
2. Ve a la pestaña de **Extensiones** en el menú lateral izquierdo (icono de cuatro cuadritos).
3. En la barra de búsqueda, escribe "PlatformIO IDE".
4. Haz clic en "Install" para instalar la extensión PlatformIO IDE.

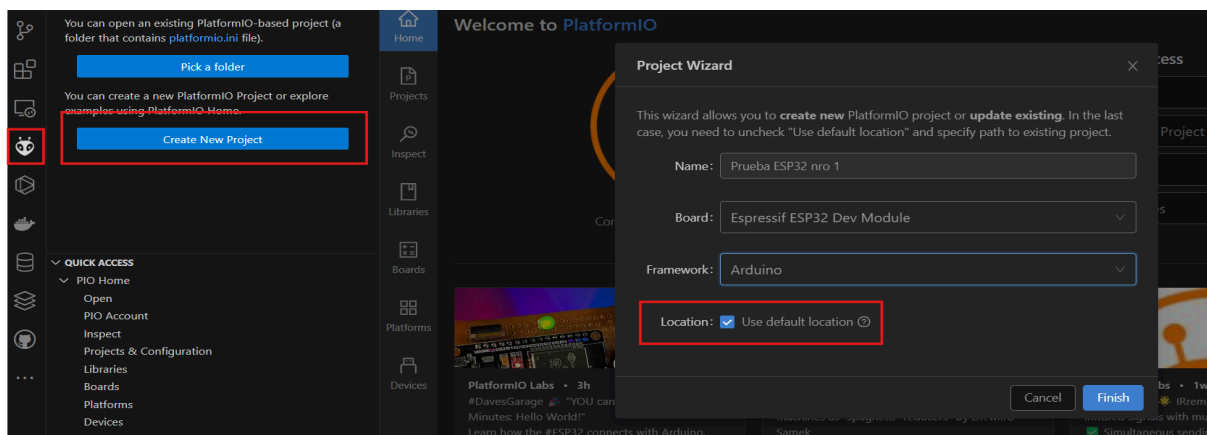


---

## 2. Configuración de Bibliotecas Necesarias para ESP32

### 2.1 Creación de un Nuevo Proyecto

1. Abre PlatformIO desde el icono en la barra lateral izquierda de Visual Studio Code.
2. Haz clic en **New Project**.
3. Completa los campos:
  - **Project Name**: Asigna un nombre a tu proyecto. En este ejemplo: "Prueba ESP32 nro 1"
  - **Board**: Selecciona "Espressif ESP32 Dev Module" (o el modelo específico de tu ESP32).
  - **Framework**: Selecciona "Arduino".
  - **Location**: Selecciona la carpeta donde quieres guardar tu proyecto.
4. Haz clic en **Finish**. PlatformIO generará el proyecto con una estructura básica.



## 2.2 Instalación de Bibliotecas Necesarias

Las bibliotecas a instalar dependen de los componentes a utilizar. Sin embargo, a continuación te indico algunas bibliotecas comunes para ESP32

En PlatformIO, ve a **PlatformIO Home > Libraries**.

### Sensores de Humedad y Temperatura

- DHT sensor library
- Adafruit BME280 Library

### Módulo WiFi

- Biblioteca WiFi (integrada en el framework de ESP32)

### Módulo LoRa

- LoRa
- MCCI LoRaWAN LMIC library

### Bluetooth Low Energy (BLE)

- ESP32 BLE Arduino

### Pantallas OLED o LCD

- Adafruit SSD1306
- Adafruit GFX Library
- LiquidCrystal\_I2C (si usas pantallas LCD con I2C)

Puedes instalar cualquier otra biblioteca necesaria utilizando la pestaña **Libraries** de PlatformIO, buscando por nombre.

## 3. Conexión del ESP32 a la PC y Verificación de Comunicación

### 3.1 Conexión Física

1. Conecta el ESP32 a tu computadora usando un cable USB.
2. Asegúrate de que el cable es de **datos y alimentación** (no solo de carga).

### 3.2 Verificación de la Conexión

1. En VS Code, abre la terminal de PlatformIO (puedes hacerlo desde la barra inferior o el menú **View > Terminal**).
2. Ejecuta el siguiente comando para ver los dispositivos conectados:

```
pio device list
```

Este comando listará los puertos serie disponibles. Debes ver tu ESP32 en la lista, usualmente identificado como **COMx** en Windows o **/dev/ttyUSBx** en Linux/macOS.

### 3.3 Creación y Ejecución de un Código Simple

El código consiste únicamente de un microcontrolador ESP32, un led y una resistencia (1K Ohm) - El mismo parpadeara en intervalos de 1 segundo.

1. En tu proyecto recién creado, abre **src/main.cpp**.
2. Escribe el siguiente código de ejemplo:
- 3.

```
#include <Arduino.h>

const byte led_gpio = 2;

void setup() {
  pinMode(led_gpio, OUTPUT);
  Serial.begin(115200); // Inicia la comunicación serial a 115200
  baudios
  Serial.println("ESP32 Iniciada"); // Mensaje inicial
}

void loop() {
  digitalWrite(led_gpio, HIGH); // turn the LED on (HIGH is the
  voltage level)
  Serial.println("LED encendido"); // Imprime el mensaje en la consola
  delay(1000); // wait for a second

  digitalWrite(led_gpio, LOW); // turn the LED off by making the
  voltage LOW
  Serial.println("LED apagado"); // Imprime el mensaje en la consola
  delay(1000); // wait for a second
}
```

Guarda el archivo y haz clic en el botón **Upload** en la barra inferior de PlatformIO (o usa el

atajo **Ctrl + Alt + U**).

Esto compilará y cargará el código en tu ESP32.

Una vez que el código se haya cargado exitosamente, abre la **Serial Monitor** en PlatformIO (icono en la barra inferior o **Ctrl + Shift + M**).

Verifica que el mensaje "**ESP32 conectado correctamente.**" aparezca en el monitor serie, seguido del mensaje "**Ejecutando el loop...**" cada segundo.

## 4. Circuito y conexión de ejemplo

Elementos del circuito

1 x Resistencia 1K Ohm

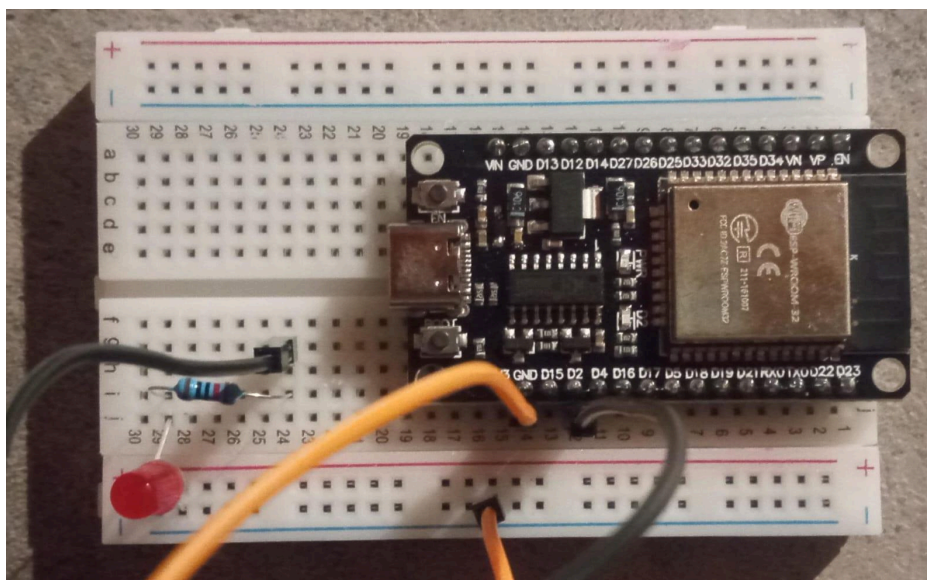
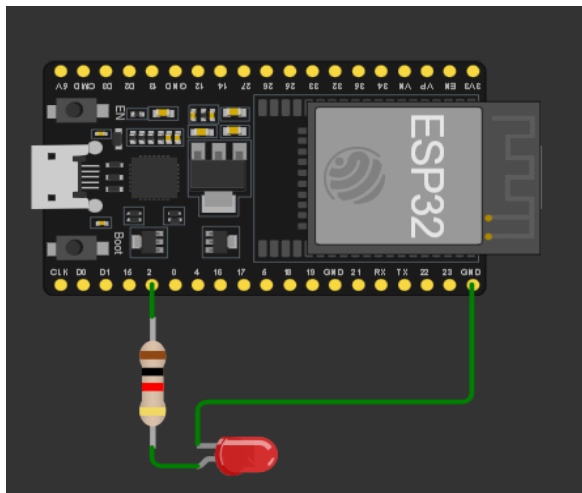
1 x Led

ESP32 serie C

Protoboard mini

2x jumpers (los cables)

\* Circuito emulado en Wokwi



\* Circuito real  
ensamblado sobre  
protoboard-mini



## 5. Computacion, carga del código y ejecución

Una vez armado el circuito, y diseñado el código, el mismo debe cargarse en el dispositivo. Como indicamos anteriormente, lo vamos a hacer mediante la extensión para el VS Code, PlatformIO. Voy a enumerar el paso a paso con la correspondiente explicación.

### 5.1. Configuración del proyecto

El proyecto compondrá principalmente de dos archivos fundamentales.

main.cpp (src > main.cpp)

platformio.ini








main.cpp en donde va el código que previamente desarrollamos y probamos en Woki (por ejemplo). Mientras en platformio.ini se encuentran la configuración de la placa, y el puerto a utilizar, como así también la velocidad del monitor (en caso de tener configurado algo a mostrar en el monitor serial).

```
[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
upload_port = COM5 ; Reemplaza COMx con el puerto correcto (en Windows)
o /dev/ttyUSBx (Linux/Mac)
monitor_speed = 115200
```

Los parámetros platform, board, framework se generan según lo seleccionado al momento de crear el proyecto. upload\_port es el puerto de salida hacia nuestro microcontrolador.

Este se puede encontrar, con el dispositivo conectado a la PC via USB en:

Administrador de dispositivos > Puertos (COM y LPT)

- >  Firmware
- >  Monitores
- >  Mouse y otros dispositivos señaladores
- >  Procesadores
- ▼  Puertos (COM y LPT)
  -  USB-SERIAL CH340 (COM5)
- >  Teclados

Monitos\_speed es el estándar de 115200 o 9600

## 5.2. Compilación y carga (BUILD | UPLOAD)

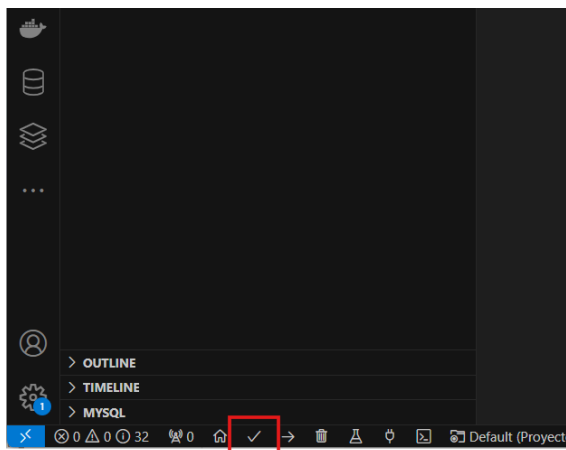
En este paso debemos tener conectado nuestro microcontrolador. Vamos a cargarle el código para que comience a ejecutar lo definido previamente. AL conectarlo debemos notar que se enciende el led de color rojo que viene incorporado en el microcontrolador.

### [PRECAUCIONES:]

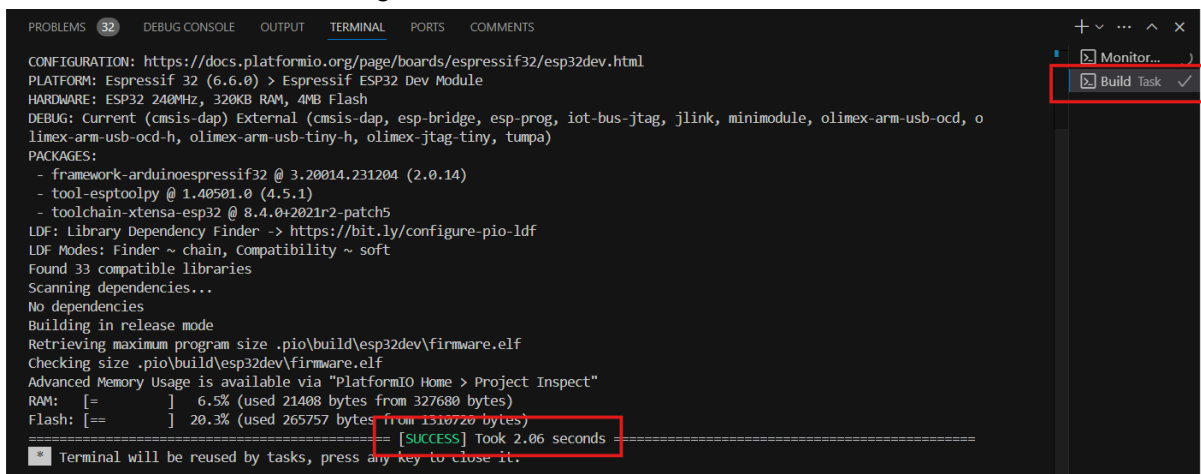
De conectarlo a la PC (o fuente de energía) ya conformando el circuito, si hay que realizar algún cambio como dar vuelta un led, quitar o poner algo, DESCONECTARLO primero. Así evitamos algún corto, y que posiblemente quememos la placa.

Ya configurado todo lo anterior, vamos a realizar el BUILD

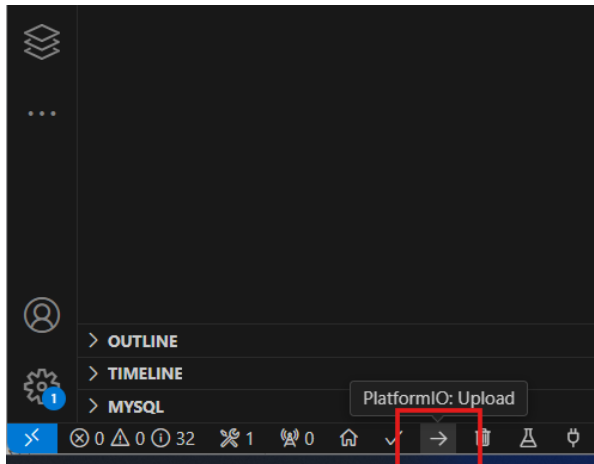
Presionamos sobre el [tilde] en la parte inferior de la pantalla, o en presionamos las teclas Crtl + Alt + B



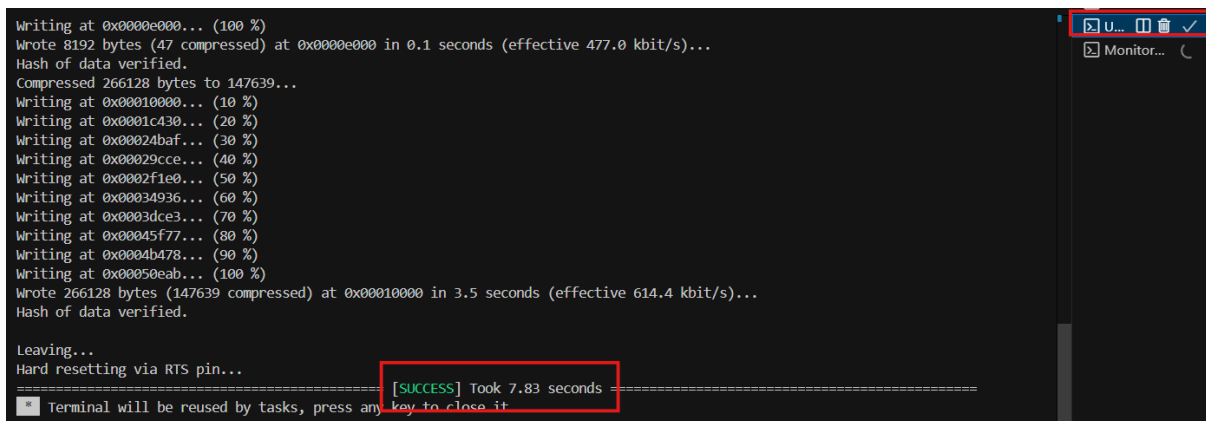
A continuación, veremos la siguiente consola:



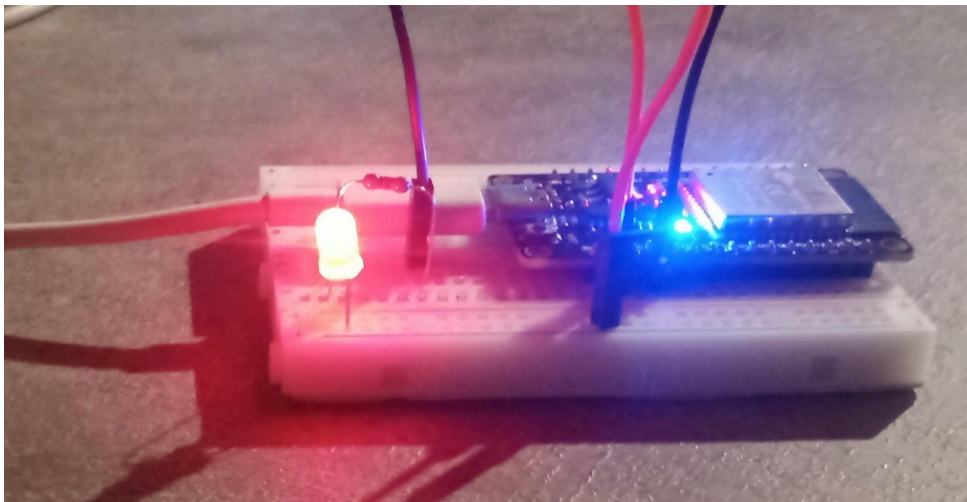
Si el resultado figura como [success] procedemos con la carga de lo realizado al dispositivo, el UPLOAD. Esto lo realizamos presionando la [flecha] o presionando el atajo Ctrl + Alt + U



Debemos ver luego de esto la siguiente pantalla. Y una vez completado exitosamente el controlador debe comenzar a ejecutar el bucle indicado en el código. Ejemplo debajo:



Notamos ya se encuentra el ciclo en ejecucion, y una luz de color azul, parte del microcontrolador, que nos indica la ejecucion correcta del proceso.



Se sumarán a la carpeta de este documento 3 videos, mostrando lo indicado aquí.