

En el main nos encontramos con una estructura de carpeta a detallar

En la carpeta [tareafinal-grupo-1/.vscode/](#)

[1-arduino.json](#)

[2-settings.json](#)

[3-c_cpp_properties.json](#)

1-El archivo [arduino.json](#) contiene la configuración para el entorno de desarrollo Arduino en Visual Studio Code (VSCode). Este archivo es utilizado por el entorno de desarrollo para definir las configuraciones específicas del proyecto, que contiene:

- "port": "COM3": Indica el puerto de comunicación al que está conectada la placa ESP32. **En este caso, la placa está conectada al puerto COM3.**
- "configuration":
"JTAGAdapter=default,PSRAM=disabled,PartitionScheme=default,CPUFreq=240,FlashMode=qio,FlashFreq=80,FlashSize=4M,UploadSpeed=921600,LoopCore=1,EventsCore=1,DebugLevel=none,EraseFlash=none": Aquí se especifican varias configuraciones para la compilación y carga del programa en la placa ESP32. Algunas de las configuraciones notables incluyen la frecuencia de la CPU, el modo y frecuencia de la memoria flash, la velocidad de carga, entre otros.
- "board": "esp32:esp32:esp32": Indica la configuración de la placa que se está utilizando en el proyecto. En este caso, **se está utilizando la placa ESP32**, y los parámetros adicionales (esp32:esp32) probablemente se refieran a la variante específica del ESP32.
- "programmer": "esptool": Especifica el programador que se utilizará para cargar el programa en la placa. **En este caso, se utiliza esptool**, que es una herramienta comúnmente utilizada para la programación de ESP8266 y ESP32.
- "sketch":
"3_Software\\Codigo_SmarHome_4Relay-BT-SW-IR-Rmaker_IoT\\Codigo_SmarHome_4Relay-BT-SW-IR-Rmaker_IoT.ino": Indica la ubicación del archivo de código fuente (sketch) del proyecto. **En este caso, el archivo se encuentra en la ruta especificada y se llama Codigo_SmarHome_4Relay-BT-SW-IR-Rmaker_IoT.ino.**

En resumen, **este archivo arduino.json se utiliza para configurar el entorno de desarrollo para compilar y cargar un programa específico en una placa ESP32 utilizando Visual Studio Code y el complemento de Arduino.**

2-El archivo `settings.json` en la carpeta `.vscode` del proyecto "tareafinal-grupo-1, " `settings.json` esta relacionado con la configuración del entorno de desarrollo para MicroPython en Visual Studio Code. contiene=

- **"MicroPython.executeButton"**: Esta configuración define el botón de ejecución (Run) en el entorno de desarrollo de MicroPython. Aquí están los detalles específicos del botón:

"text": "▶": El texto que se mostrará en el botón, en este caso, un símbolo de reproducción.

"tooltip": "Run": El texto que se mostrará como información sobre herramientas (tooltip) al pasar el cursor sobre el botón, indicando que este botón se utiliza para ejecutar el código.

"alignment": "left": Alineación del botón a la izquierda.

"command": "extension.executeFile": El comando que se ejecutará cuando se presione el botón. En este caso, se ejecutará el archivo actual.

"priority": 3.5: La prioridad del botón.

- **"MicroPython.syncButton"**: Esta configuración define el botón de sincronización (sync) en el entorno de desarrollo de MicroPython. Aquí están los detalles específicos del botón:

"text": "\$ (sync)": El texto que se mostrará en el botón, utilizando un ícono de sincronización.

"tooltip": "sync": El texto que se mostrará como información sobre herramientas al pasar el cursor sobre el botón, indicando que este botón se utiliza para sincronizar.

"alignment": "left": Alineación del botón a la izquierda.

"command": "extension.execute": El comando que se ejecutará cuando se presione el botón. En este caso, se ejecutará la extensión para ejecutar el código.

"priority": 4: La prioridad del botón.

En resumen, estas configuraciones personalizadas estan diseñadas para facilitar la ejecución y sincronización de código MicroPython en Visual Studio Code, proporcionando botones visuales con funcionalidades específicas.

3-El archivo, `c_cpp_properties.json`, es un archivo de configuración utilizado por el entorno de desarrollo Visual Studio Code (VSCode) para configurar la inteligencia de código (IntelliSense) y otras configuraciones relacionadas con el desarrollo en C++.

Este archivo se utiliza para proporcionar información al entorno de desarrollo sobre cómo compilar y analizar el código en el proyecto. La configuración está diseñada específicamente para el desarrollo de proyectos Arduino que utilizan el compilador para ESP32, proporciona la configuración de rutas de inclusión (`includePath`) para el proyecto. Estas rutas son importantes para el compilador y el sistema de autocompletado (IntelliSense) para ubicar los archivos de encabezado y bibliotecas necesarios durante el

desarrollo. Aquí hay un análisis de las rutas de inclusión proporcionadas: algunas de las claves y valores en este archivo:

- version: Indica la versión del archivo de configuración. En este caso, es la versión 4.
- configurations: Contiene una lista de configuraciones. En este caso, hay una configuración llamada "Arduino".
- name: Es el nombre de la configuración. En este caso, "Arduino".
- compilerPath: Es la ruta al compilador que se utilizará. En este caso, es un compilador específico para ESP32.
- compilerArgs: Son los argumentos del compilador que se utilizarán. Aquí se especifican diversas opciones, como activar características específicas, desactivar advertencias y configurar estándares de código.
- intelliSenseMode: Especifica el modo de IntelliSense que se utilizará. En este caso, es "gcc-x64", indicando que se utilizará el modo IntelliSense para el compilador GCC en arquitectura x64.
- Ruta de inclusión: El parámetro "forcedInclude" especifica un archivo de encabezado que se incluirá automáticamente en el proceso de compilación. En este caso, incluye el archivo Arduino.h del paquete ESP32.
- Estándares de C y C++: Los parámetros "cStandard" y "cppStandard" especifican los estándares de C y C++ que se utilizarán durante la compilación. En este caso, se establecen en C11 y C++11, respectivamente.
- Definiciones del preprocesador: El parámetro "defines" incluye varias definiciones del preprocesador. Estas definiciones proporcionan información sobre el entorno, la plataforma y configuraciones específicas para el proceso de compilación.
- Definiciones específicas de ESP32: Hay varias definiciones relacionadas con el desarrollo para ESP32, la versión de Arduino, la arquitectura y la variante.
- Información del compilador y la plataforma: Se incluye información como la versión de IDF (ESP-IDF), detalles de la plataforma y otras definiciones específicas del compilador.
- Otras opciones de configuración: Se establecen varias otras opciones específicas del compilador y de la plataforma, incluidas algunas relacionadas con la configuración USB ("USBCON").