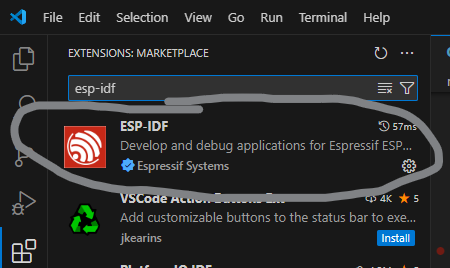
# Setting up the ESP32 for flashing

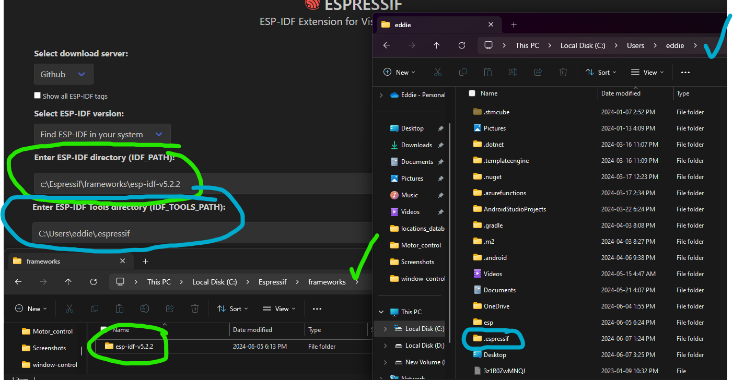


After the ESP-IDF extension is installed, on the ESP-IDF setup tab, double check your PATHs.

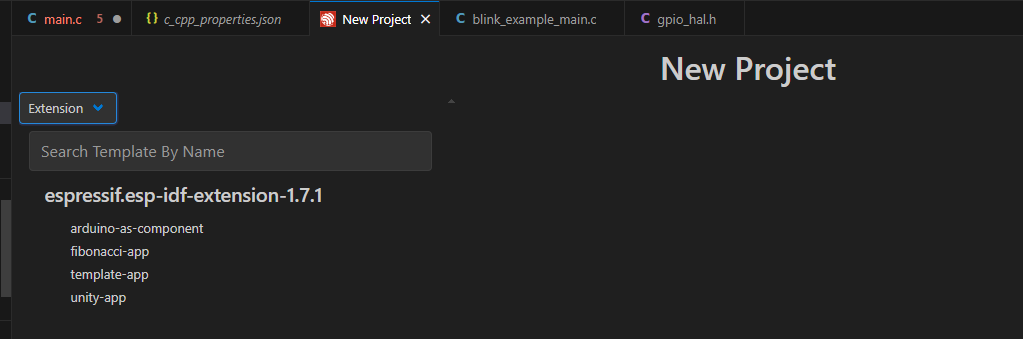


In order to get the ESP32 set up and functional, went to this website and installed ESP-IDF on vscode:

https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/windows-setup.html







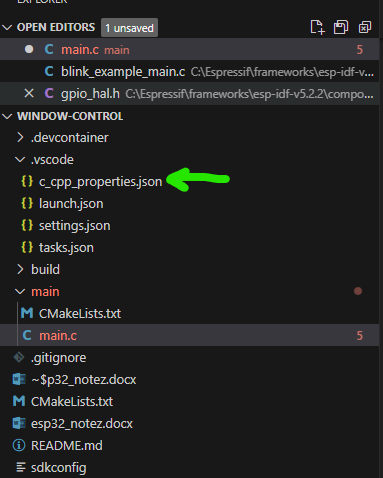
On VSCode when you press the “Install” button, it’ll ask what workspace you want to install to at the top of the screen, which in my case is the root project folder where my firmware, hardware, schematics etc. will be located.



After creating a new project and selecting the folder, I choose template-app on extension. This creates a blank project with no code other than app\_main



# Getting it to blink

I changed the code to this in c\_cpp\_properties.json

{

    "configurations": [

      {

        "name": "ESP-IDF",

        "cStandard": "c11",

        "cppStandard": "c++17",

        "includePath": [

          "${config:idf.espIdfPathWin}/components/\*\*",

          "${workspaceFolder}/\*\*"

        ],

        "browse": {

          "path": [

            "${config:idf.espIdfPathWin}/components",

            "${workspaceFolder}"

          ],

          "limitSymbolsToIncludedHeaders": false

        },

        "compilerPath": "${config:idf.toolsPathWin}\\tools\\xtensa-esp-elf\\esp-13.2.0\_20230928\\xtensa-esp-elf\\bin\\xtensa-esp32-elf-gcc.exe"

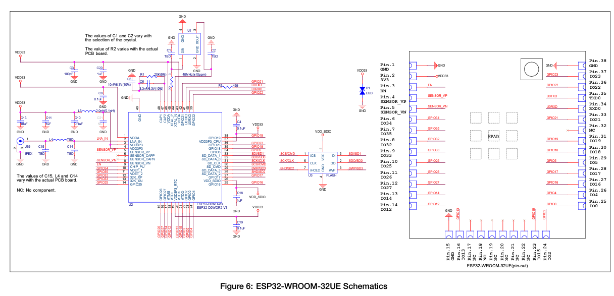
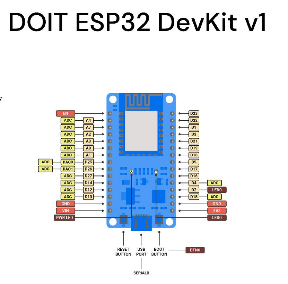
      }

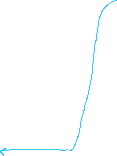
    ],

    "version": 4

  }

Getting ESP32 to work: LED0 on the board is connected to D2, or GPIO 2.

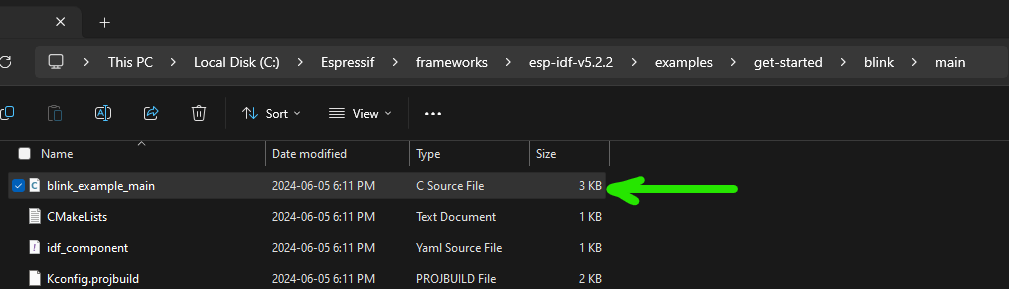




#include <stdio.h>

#include "freertos/FreeRTOS.h"

#include "freertos/task.h"

Note: this code is sort of based on their blink\_example\_main.c code. 

#include "driver/gpio.h"

#include "esp\_log.h"

#define CONFIG\_BLINK\_PERIOD 1000

//ESP32 controls window linear actuator to open and close

static const char\* TAG = "main";

static gpio\_num\_t LED\_PIN = 2;

static uint32\_t LED\_state = 0;

void configure\_led(void) {

    ESP\_LOGI(TAG, "configuring LED...");

    gpio\_reset\_pin(LED\_PIN);

    gpio\_mode\_t mode = GPIO\_MODE\_OUTPUT;

    gpio\_set\_direction(LED\_PIN, mode);

}



static void blink\_led(void) {

    gpio\_set\_level(LED\_PIN, LED\_state);



}

void app\_main(void)

{

    configure\_led();

    while(1) {

        ESP\_LOGI(TAG, "Turning the LED %s", LED\_state == true ? "ON": "OFF");

        blink\_led();

        LED\_state = !LED\_state;

        vTaskDelay(CONFIG\_BLINK\_PERIOD / portTICK\_PERIOD\_MS);

    }

}

# FreeRTOS notes

