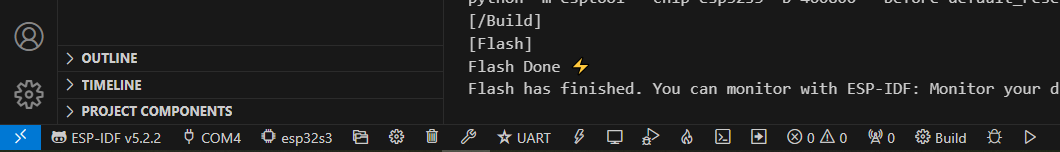


For installing and configuration go to following address:

**Installation and Use of the Visual Studio Code ESP-IDF Extension**

<https://www.youtube.com/watch?v=XDDcS7HQNlI>

After configuration select COM port and your board on bottom menu of VScode like bellow images:





UART should be selected if you are using serial communication.

For creating **new project from scratch** first select a folder then select terminal and write:



idf.py create-project –p . TEST

idf.py create-project -p . app\_main

But I don\t Know why I’m getting error

Solution:

<https://www.esp32.com/viewtopic.php?t=32794&start=10>

Hi Jorgen, I have the same problem, i.e. idf.py does not run in the PowerShell terminal which is opened by clicking Terminal and then New Terminal. However I get round this by opening an ESP-IDF terminal, this can be done by pressing ctrl E, releasing and then pressing t, or by pressing F1 to bring up the command pallet, type in ESP-IDF to filter out unwanted commands and then find Open ESD-IDF Terminal and click on it. Within the ESP-IDF terminal idf.py works perfectly. Note that I did install the environment variables by hand.

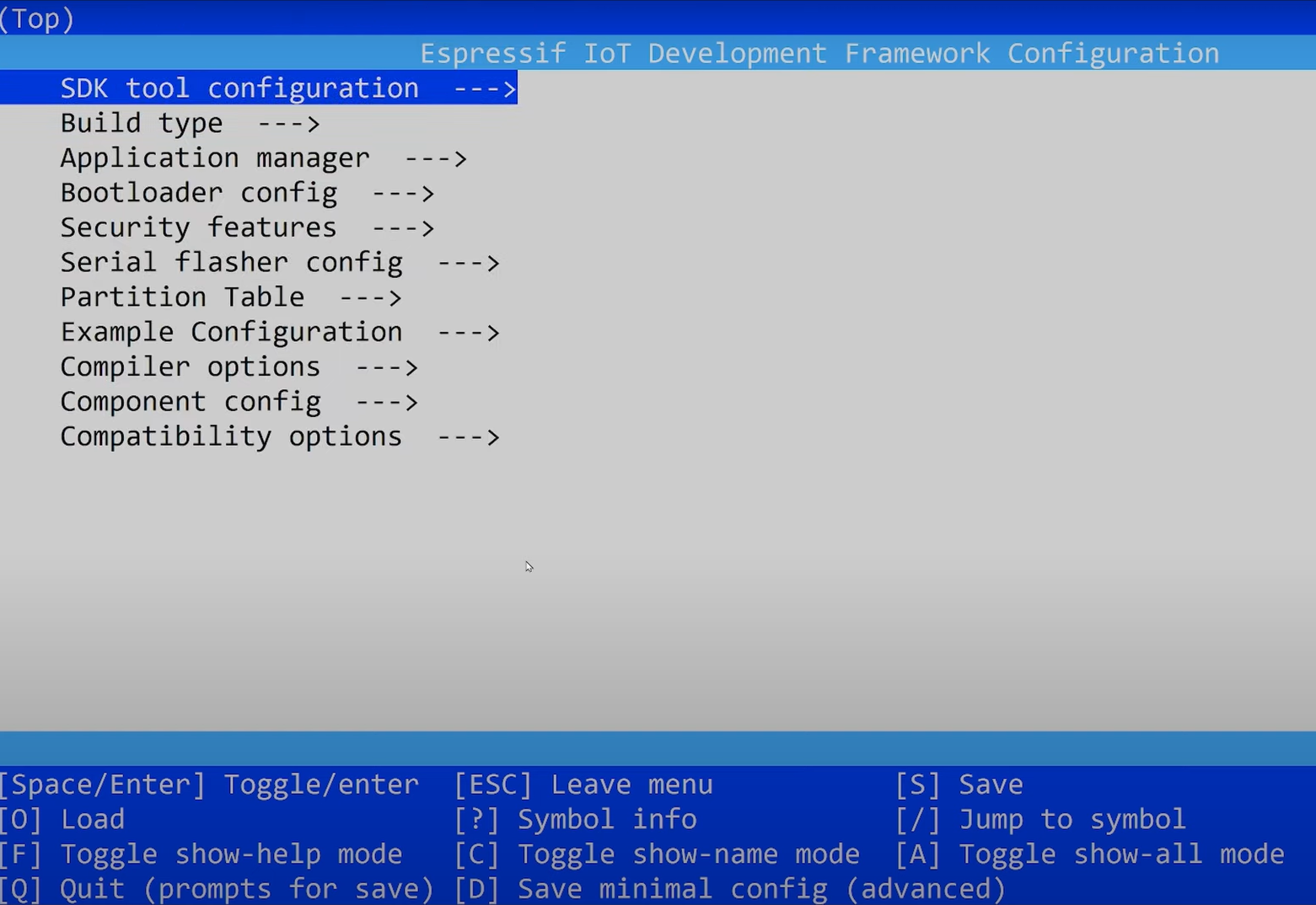
To include library watch:

**ESP32 - How to create your First ESP IDF project (From Scratch)**

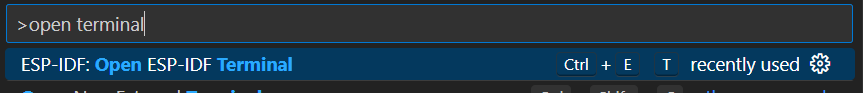
<https://www.youtube.com/watch?v=oHHOCdmLiII&t=16s>

Time: 8:14/13:10

How use configmenu:



Press F1 and type Terminal:

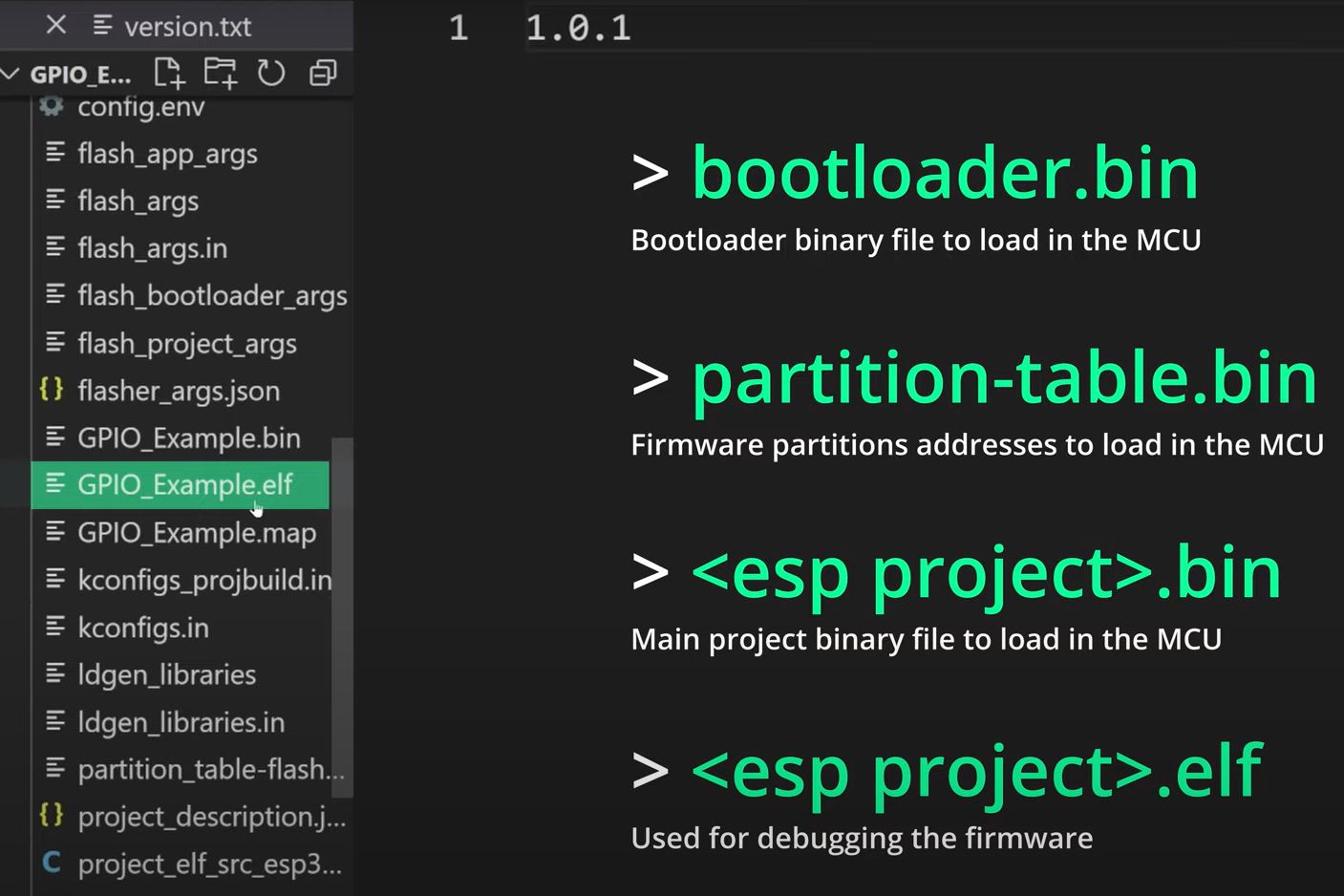


Select ESP-IDF Terminal

Now in Terminal write:

Start idf.py menuconfig

For example you can use it to change cpu frequency of ESP32



ESP-IDF Hint: esp\_mac.h header file is not included by esp\_system.h anymore. It shall then be manually included with #include "esp\_mac.h"

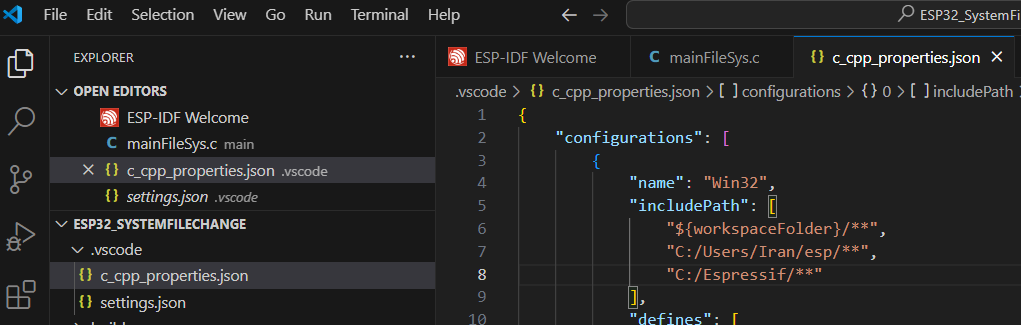
Add :

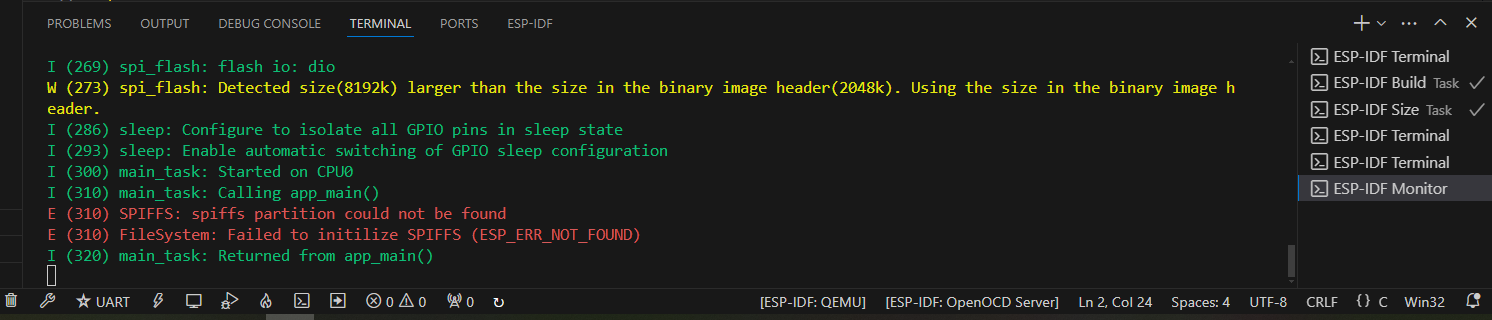
"C:/Users/Iran/esp/\*\*",

"C:/Espressif/\*\*",

"C:/Users/Iran/esp/v5.2.2/esp-idf/components/log/include"

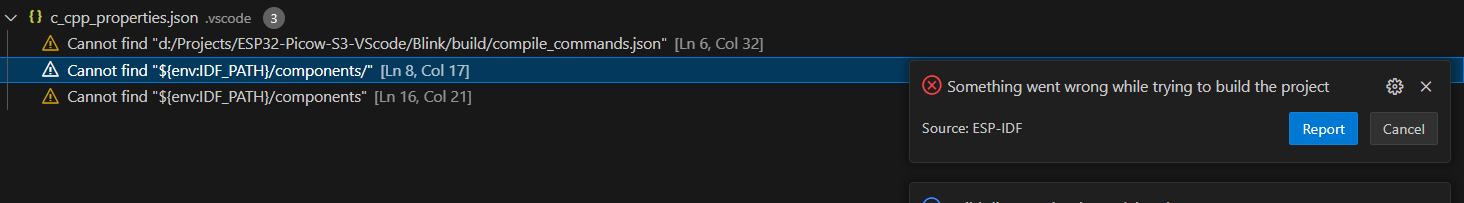
To Json file of your project

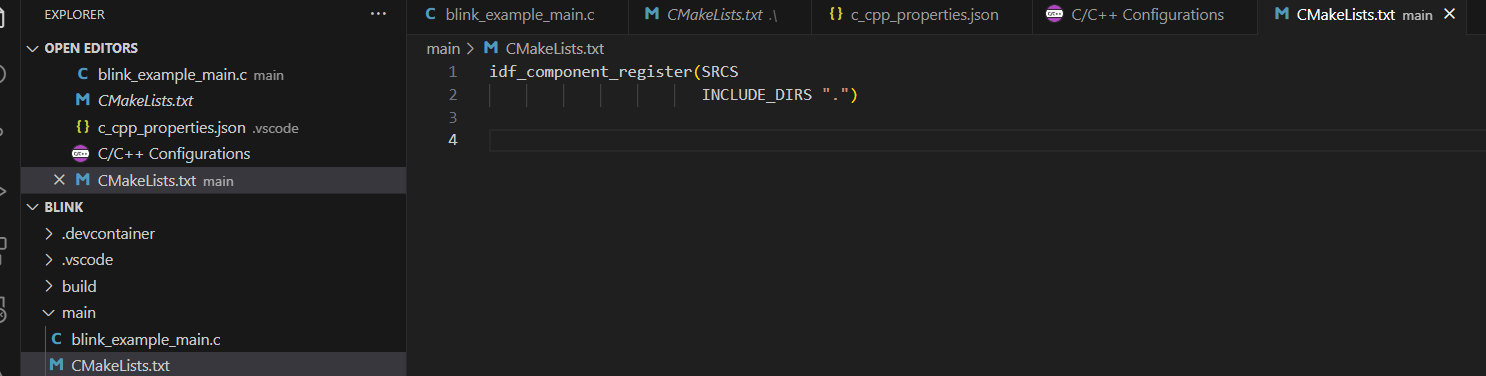




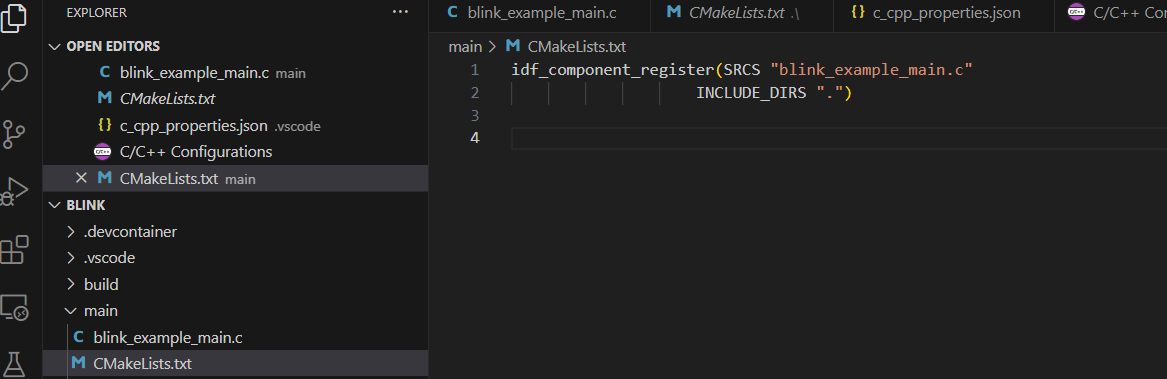
W (273) spi\_flash: Detected size(8192k) larger than the size in the binary image header(2048k). Using the size in the binary image header.

Solution:





Solution: file CMkeList.txt should contains .c file name inside like image below



Bare Metal programing of ESP32

**Bare metal GPIO driver for ESP32 | Toggling an LED**

#include <stdio.h>

//#include <cstdint>

#include "FreeRTOS.h"

#include "task.h"

#define GPIO\_OUT\_W1TS\_REG 0x60004008

#define GPIO\_OUT\_W1TC\_REG 0x6000400c

#define GPIO\_ENABLE\_REG   0x60004020

#define GPIO5             5

#define DELAY\_MS          500

void app\_main(void)

{

   volatile \_\_UINT32\_TYPE\_\_\* gpio\_out\_w1ts\_reg = (volatile \_\_UINT32\_TYPE\_\_\*) GPIO\_OUT\_W1TS\_REG;

   volatile \_\_UINT32\_TYPE\_\_\* gpio\_out\_w1tc\_reg = (volatile \_\_UINT32\_TYPE\_\_\*) GPIO\_OUT\_W1TC\_REG;

   volatile \_\_UINT32\_TYPE\_\_\* gpio\_enable\_reg = (volatile \_\_UINT32\_TYPE\_\_\*) GPIO\_ENABLE\_REG;

   \*gpio\_enable\_reg = (1 << 5);

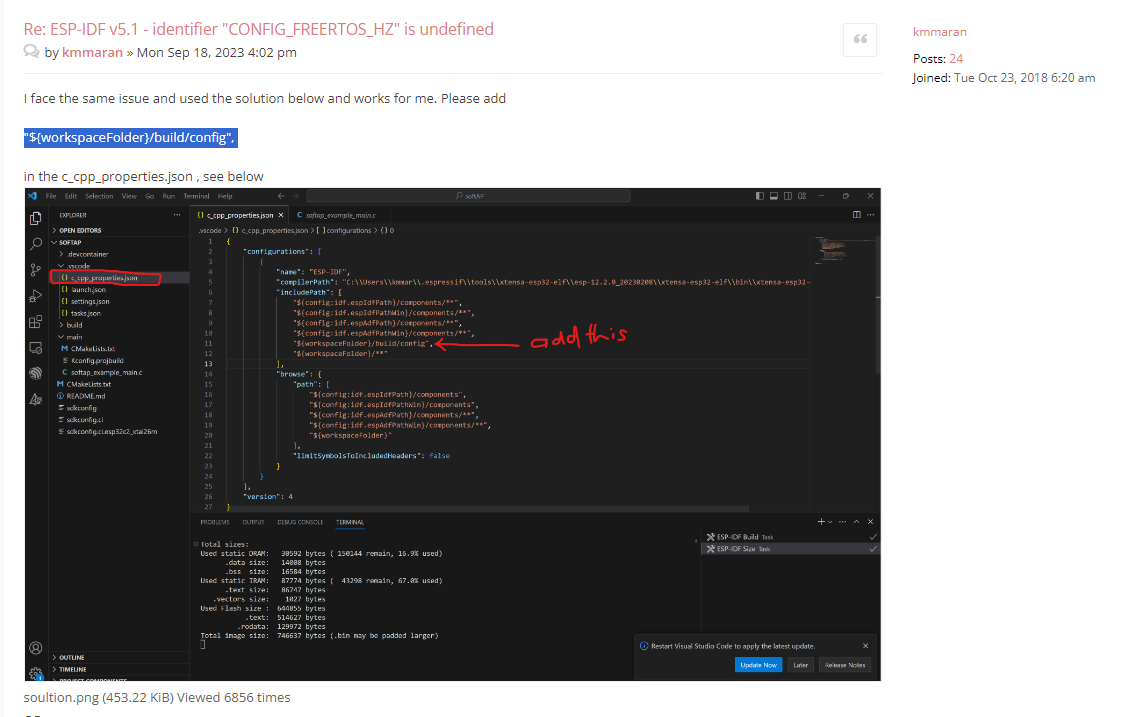
   \*gpio\_out\_w1ts\_reg = (1 << 5);

   \*gpio\_out\_w1tc\_reg = (1 << 5);

}

<https://www.youtube.com/watch?v=QrJf6CF_g_8>

identifier "CONFIG\_FREERTOS\_HZ" is undefined



Sometimes you change something in code but when it is built a deferent project although you just opened your project folder.

**Solution:**

You should full clean 

I couldn’t use GPIO43 and GPIO44 as normal input and output. It just like the freeRTOS or bootloader uses this pins as cannel for console output:

Solution:

Default is UART0 and I put that on none and add following codes to your code. And remember to use  to Full clean

    //uart\_driver\_delete(UART\_NUM\_0);

    gpio\_reset\_pin(GPIO\_NUM\_43);

    gpio\_reset\_pin(GPIO\_NUM\_44);

