

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN KỸ THUẬT ĐIỆN TỬ



BÁO CÁO MÔN HỌC

Giáo viên hướng dẫn :

Phạm Trung Hiếu CA180207

Hà Nội, 9 – 2019

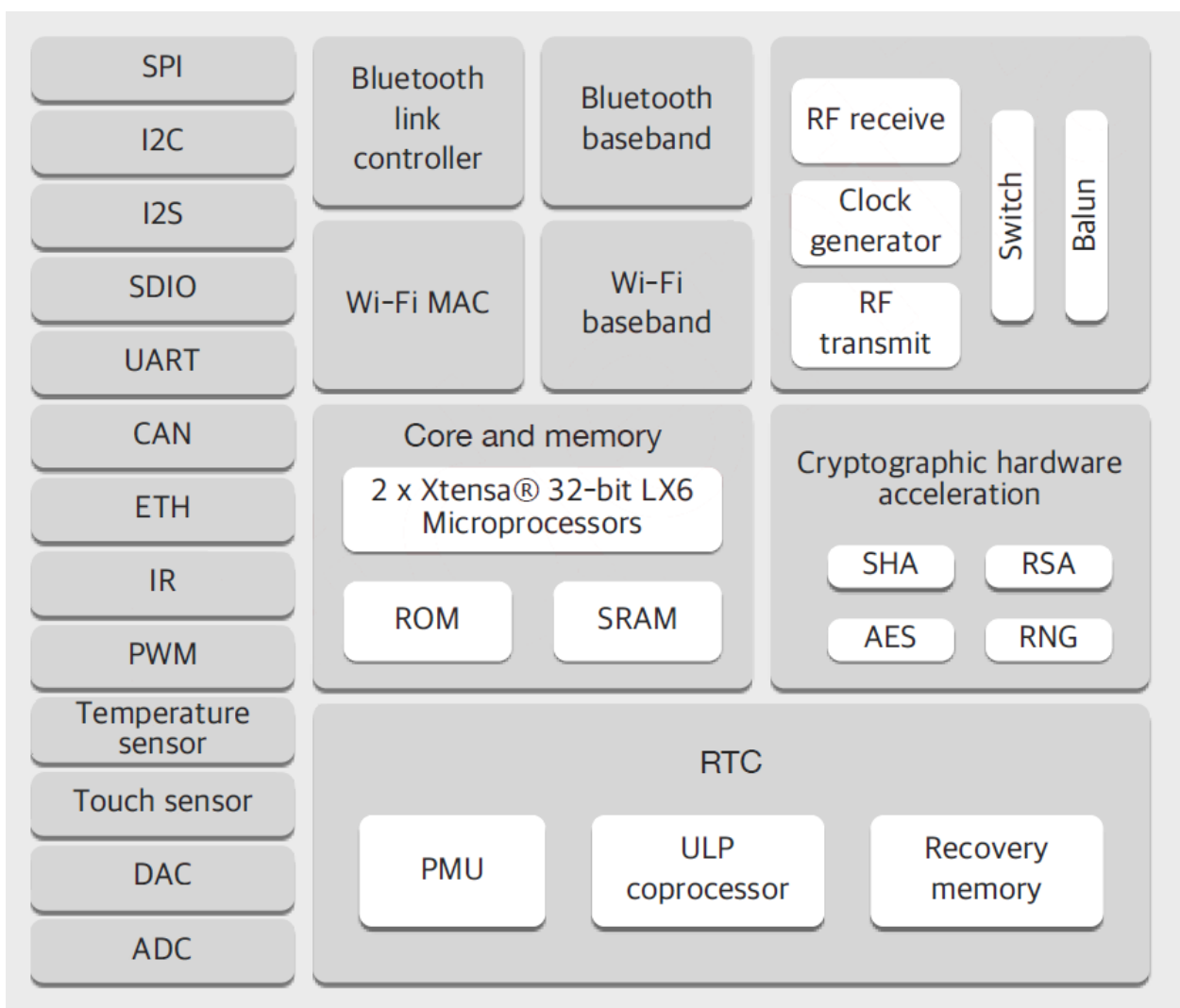
Hướng dẫn :

Đây là tài liệu cho phiên bản ổn định v3.3 của ESP-IDF.

Trong bài hướng dẫn mình sử dụng phiên bản v3.3

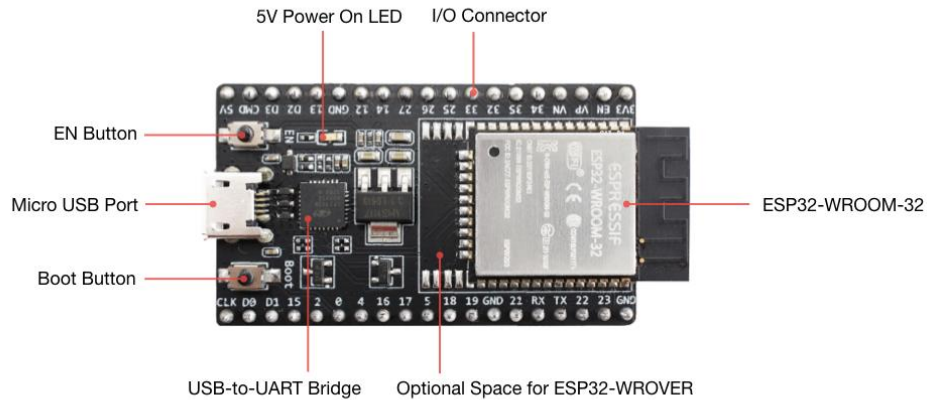
Đã có phiên bản 4.0-beta 1 thêm nhiều tính năng đc thử nghiệm .

Cài đặt môi trường phát triển ESP32 - RTOS trên Windows 10 bằng hình ảnh . Chương trình ESP32 sử dụng bộ thư viện SDK ESP-IDF (Espressif IoT Development Framework) của hãng Espressif đc phát triển dựa trên FreeRTOS có trình sửa sao cho phù hợp với ESP32 Dual-core (Dual Core tức là 1 vi xử lý có 2 nhân nhưng trong đó có 1 nhân thực và 1 nhân ảo được tạo ra nhằm hỗ trợ xử lý cho nhân thực.)



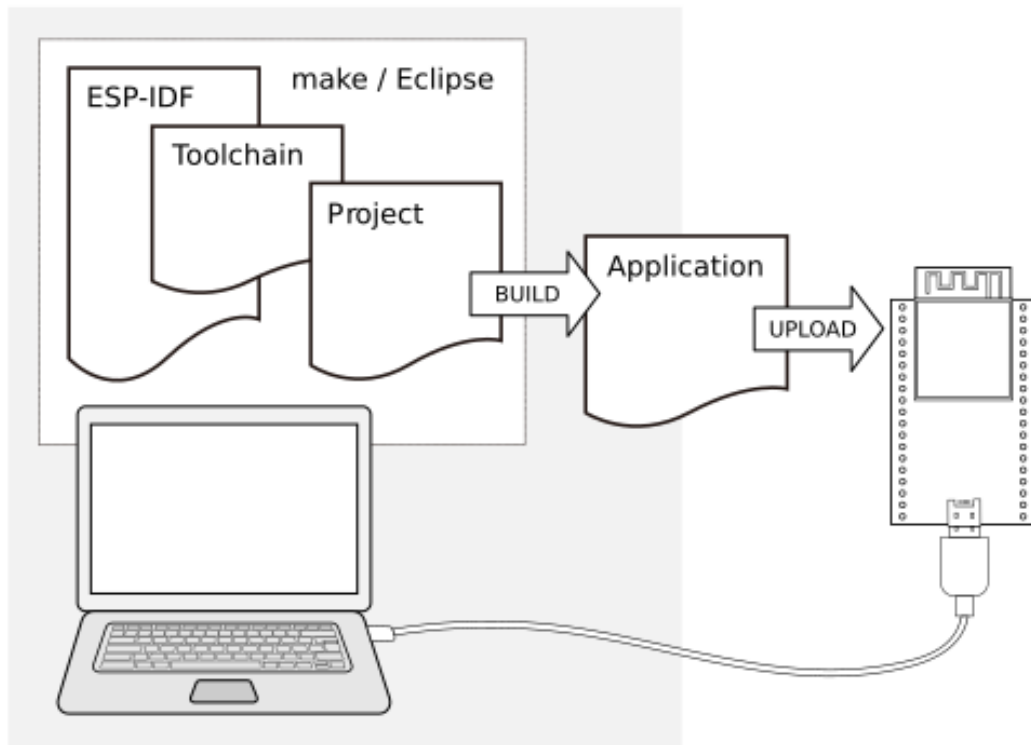
Yêu cầu phần cứng :

- **ESP32** board (mình sử dụng kit ESP32 - NodeMCU)



- **USB cable** - USB A / micro USB B
- **Computer** running Windows

Yêu cầu phần mềm :



Mô tả cấu trúc các phần mềm :

Hệ điều hành : Windows 10 – 64 bit

Môi trường MSYS2 Unix : Cung cấp cấu trúc cây thư mục , bash shell , quản lý các gói phần mềm tương tự Linux , GNU Autotools

Toolchain :

Trình biên dịch chéo

Thư viện ESP - IDF

Tool make file đc viết bằng python :

```
# make menuconfig  
# make -j4 flash monitor
```

Source code ESP32

Trình soạn thảo : visual studio code

ESP32 NodeMCU

Chú ý : trên web chính thức của hãng <https://docs.espressif.com> có hướng dẫn cài đặt môi trường phát triển đầy đủ cho Windows , Linux , Mac OS ..

Trên hệ điều hành Windows 64 bit đã có hướng dẫn cài đặt môi trường trực tiếp không sử dụng môi trường chung gian MSYS2 .

Ở những phiên bản mới ESP-IDF có thể chạy tool make trực tiếp từ Command Prompt (cmd) dòng lệnh trên win và yêu cầu một số công cụ tiên quyết được cài đặt để có thể xây dựng App cho ESP32. Các công cụ bao gồm Python , Git , trình biên dịch chéo , công cụ menuconfig , công cụ xây dựng CMake và Ninja.

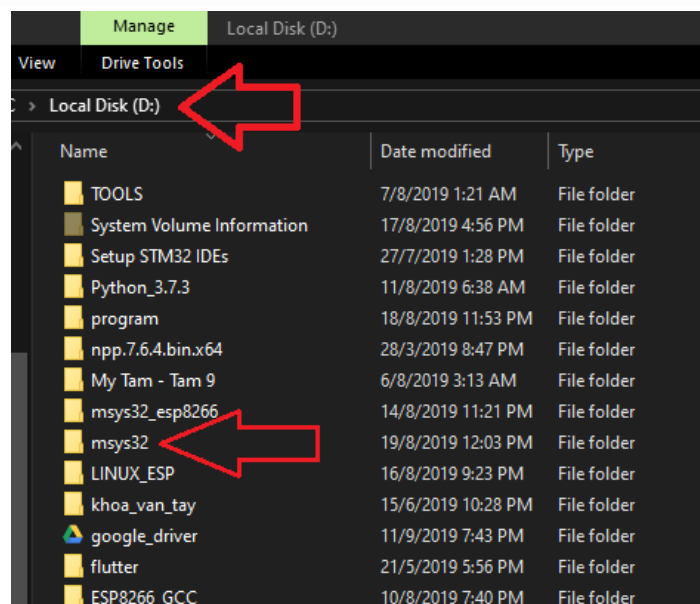
Trong bài viết này mình sẽ hướng dẫn setup môi trường sử dụng MSYS2-Unix . Môi trường này chạy độc lập trên win và có thể copy toàn bộ thư mục môi trường cũng thư viện đi bất kỳ máy tính cá nhân nào có sử dụng Win mà không cần cài đặt lại môi trường phát triển , dễ quản lý và backup nếu gặp sự cố .

Thiết lập môi trường :

Tải xuống tệp zip toolchain & MSYS2 tất cả trong một của Windows từ [dl.espressif.com](https://dl.espressif.com/dl/esp32_win32_msys2_environment_and_toolchain-20181001.zip):

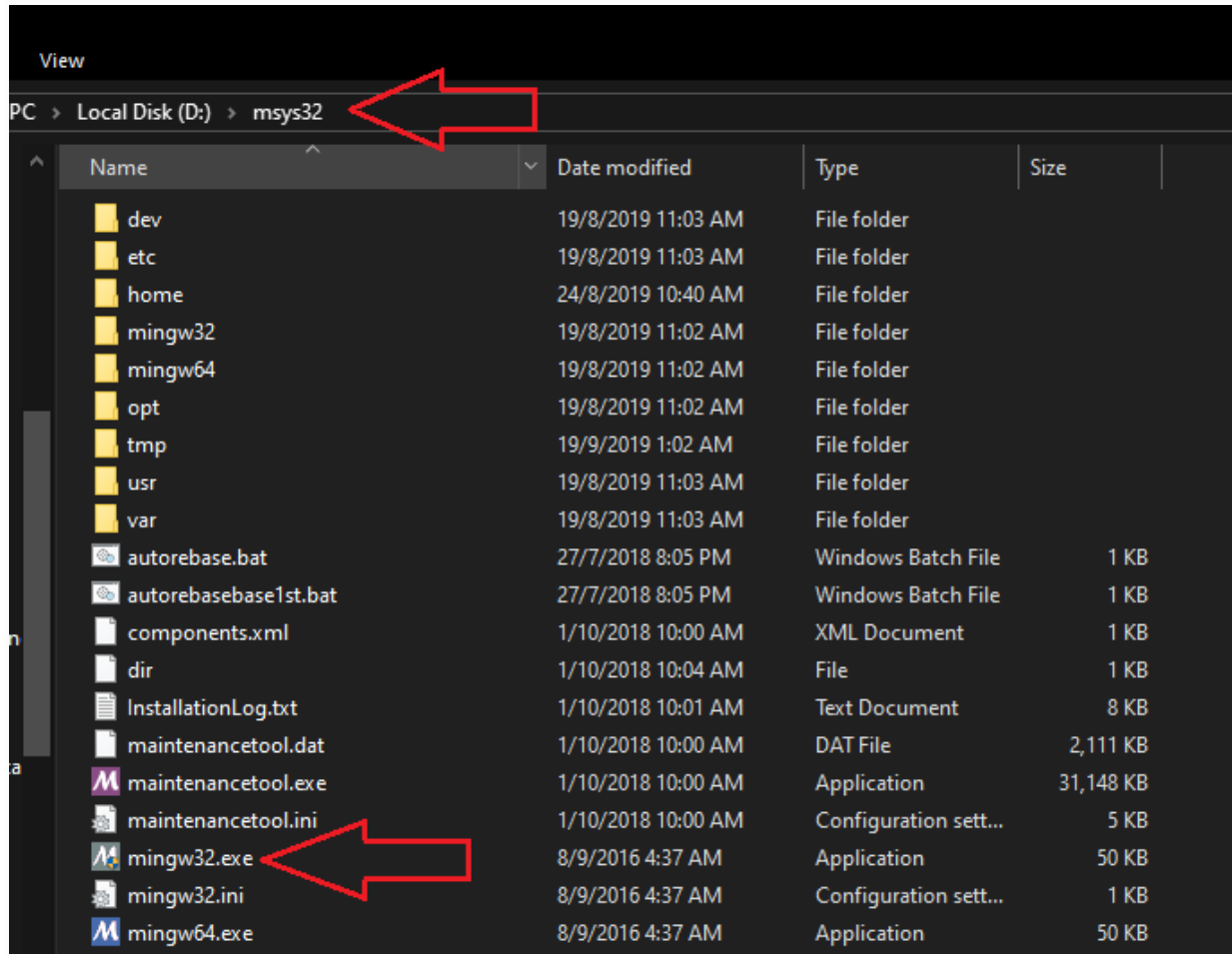
https://dl.espressif.com/dl/esp32_win32_msys2_environment_and_toolchain-20181001.zip

Giải nén tệp zip đến D:\(hoặc một số vị trí khác, nhưng hướng dẫn này giả định D:\) và nó sẽ tạo một msys32 thư mục với môi trường được chuẩn bị trước.



Kiểm tra môi trường vừa giải nén :

Mở cửa sổ MSYS2 MINGW32 vd : `D:\msys32\mingw32.exe`



Sau khi cửa sổ đã mở nhập các lệnh sau :

```
phamh@DESKTOP-95FIFUD MINGW32 ~  
# pwd  
/home/phamh  
  
phamh@DESKTOP-95FIFUD MINGW32 ~  
# mkdir -p ~/esp  
  
phamh@DESKTOP-95FIFUD MINGW32 ~  
# cd esp/  
  
phamh@DESKTOP-95FIFUD MINGW32 ~/esp  
# ls -l  
total 150  
-rwxr-xr-x 1 phamh phamh 140056 Sep 14 12:46 a.exe  
drwxr-xr-x 1 phamh phamh 0 Sep 13 18:07 code_esp32  
drwxr-xr-x 1 phamh phamh 0 Aug 24 10:40 esp-idf  
-rw-r--r-- 1 phamh phamh 1923 Aug 19 10:49 msys2.reg  
-rw-r--r-- 1 phamh phamh 66 Sep 14 12:46 text.c  
-rw-r--r-- 1 phamh phamh 64 Sep 14 12:45 text.c~  
drwxr-xr-x 1 phamh phamh 0 Aug 24 10:41 xtensa-esp32-elf  
  
phamh@DESKTOP-95FIFUD MINGW32 ~/esp  
# |
```

Giải thích lệnh shell :

```
# pwd // hiển thị đường dẫn thư mục  
# mkdir -p ~/esp // tạo 1 thư mục rỗng có tên là esp  
# cd esp/ // di chuyển vào thư mục vừa tạo  
// chú ý : nếu trong 1 thư mục có nhiều thư mục con và tên rất dài hoặc file có tên khó nhớ ta có thể dùng phím [ Tab ] để tự động lấy tên thư mục chỉ cần viết 1 chữ đầu , hay Tab 2 lần để hiển thị toàn bộ .  
  
# ls -l // hiển thị list danh sách và quyền sử dụng
```

Download thư viện ESP-IDF :

Đây là tài liệu cho phiên bản ổn định v3.3 của ESP-IDF.

Trong bài hướng dẫn mình sử dụng phiên bản v3.3

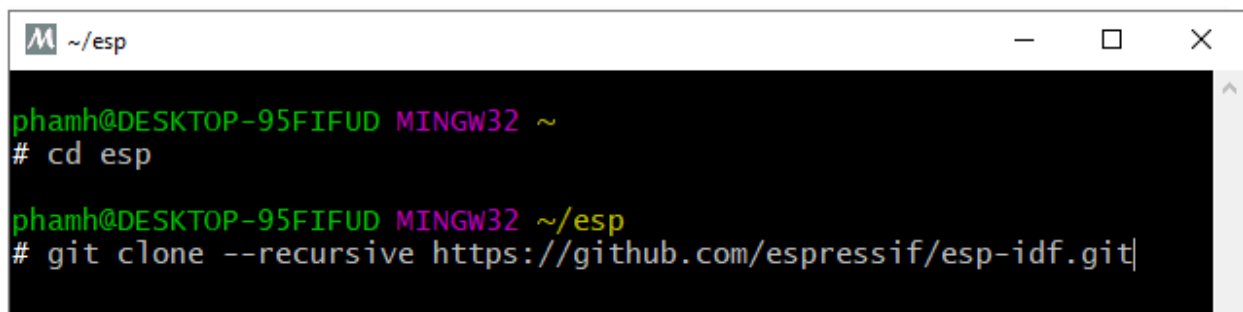
Bên cạnh toolchain , cũng cần API dành riêng cho ESP32 (thư viện phần mềm và mã nguồn). Chúng được Espressif cung cấp trong

<https://github.com/espressif/esp-idf>

Để có một bản sao cục bộ của ESP-IDF, hãy điều hướng đến thư mục cài đặt và sao chép kho lưu trữ với [**git clone**]

Mở Terminal và chạy các lệnh sau:

[git clone -b v3.3 --recursive https://github.com/espressif/esp-idf.git](#)

A screenshot of a terminal window with a black background and white text. The window title bar shows a Mac icon, the path ~/esp, and standard window controls. The terminal content shows a user named 'phamh' at a 'DESKTOP-95FIFUD' machine using 'MINGW32'. The user enters the command 'cd esp' and then 'git clone --recursive https://github.com/espressif/esp-idf.git'.

```
phamh@DESKTOP-95FIFUD MINGW32 ~  
# cd esp  
  
phamh@DESKTOP-95FIFUD MINGW32 ~/esp  
# git clone --recursive https://github.com/espressif/esp-idf.git
```


ESP-IDF sẽ được tải xuống : `~/esp/esp-idf`.

Bên trong MSYS2 đã có cài sẵn Git , chúng ta chỉ cần thực hiện lệnh .

Chú ý : chúng ta có thể down trực tiếp từ git rồi giải nén vào bên trong thư mục của MSYS2 cũng đc , không nhất thiết phải clone nếu cài offline .

*Nên nhất thiết phải để thư viện [**esp-idf**] bên trong cấu trúc thư mục của msys2 .*

Sau quá trình clone về máy thành công :

Thực hiện các lệnh như sau để kiểm tra cấu trúc thư mục :

```
Resolving deltas: 100% (162/162), done.
Submodule path 'components/nghttp/nghttp2/third-party/mruby': checked out 'd
Submodule path 'components/nghttp/nghttp2/third-party/neverbleed': checked out 'd
Submodule path 'components/protobuf-c/protobuf-c': checked out 'd
Submodule path 'components/spiffs/spiffs': checked out 'f5e26c4e9
Submodule path 'components/unity/unity': checked out '7d2bf62b7e6
Submodule path 'examples/build_system/cmake/import_lib/main/lib/t

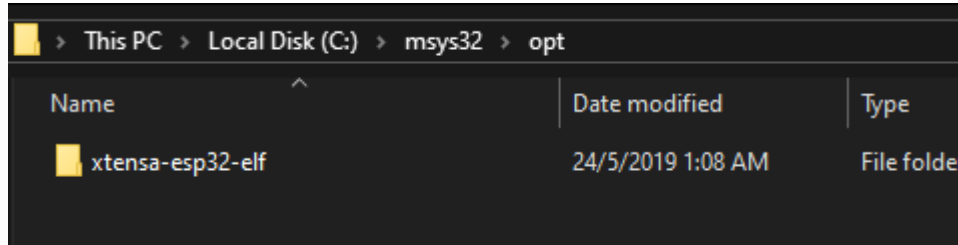
phamh@DESKTOP-95FIFUD MINGW32 ~/esp
# cd esp-idf/

phamh@DESKTOP-95FIFUD MINGW32 ~/esp/esp-idf
# ls -l
total 85
-rw-r--r-- 1 phamh phamh  770 Sep 19 23:06 add_path.sh
-rw-r--r-- 1 phamh phamh 5321 Sep 19 23:06 CMakeLists.txt
drwxr-xr-x 1 phamh phamh   0 Sep 19 23:06 components
-rw-r--r-- 1 phamh phamh 2406 Sep 19 23:06 CONTRIBUTING.rst
drwxr-xr-x 1 phamh phamh   0 Sep 19 23:06 docs
drwxr-xr-x 1 phamh phamh   0 Sep 19 23:06 examples
-rw-r--r-- 1 phamh phamh 8263 Sep 19 23:06 Kconfig
-rw-r--r-- 1 phamh phamh 11358 Sep 19 23:06 LICENSE
drwxr-xr-x 1 phamh phamh   0 Sep 19 23:06 make
-rw-r--r-- 1 phamh phamh 6487 Sep 19 23:06 README.md
-rw-r--r-- 1 phamh phamh  434 Sep 19 23:06 requirements.txt
drwxr-xr-x 1 phamh phamh   0 Sep 19 23:06 tools

phamh@DESKTOP-95FIFUD MINGW32 ~/esp/esp-idf
# |
```

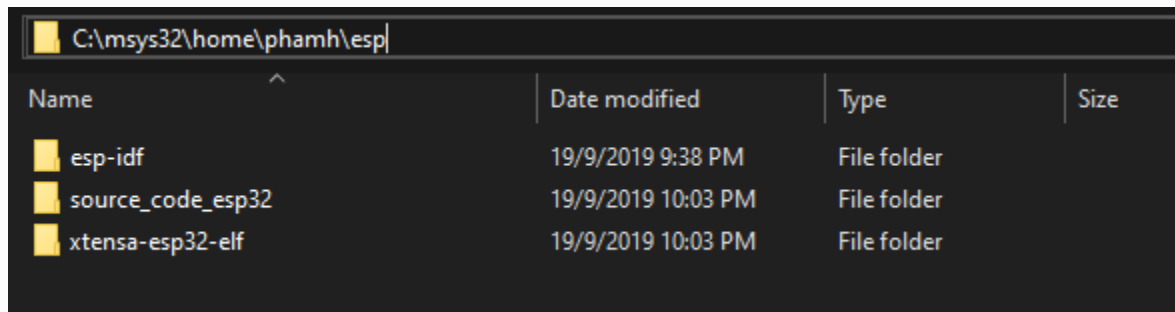
Copy trình biên dịch chéo và add biến môi trường (ADD IDF_PATH)

Trình biên dịch chéo : cho Esp32 đc nạp sẵn trong môi trường MSYS2 tải về



Ví dụ C:\msys32\opt .

chúng ta copy [xtensa-esp32-elf] để vào cùng thư mục với thư viện [esp-idf] .
tạo thêm 1 thư mục chứa toàn bộ project muốn phát triển sau này cho vào đây .



Add biến môi trường IDF_PATH :

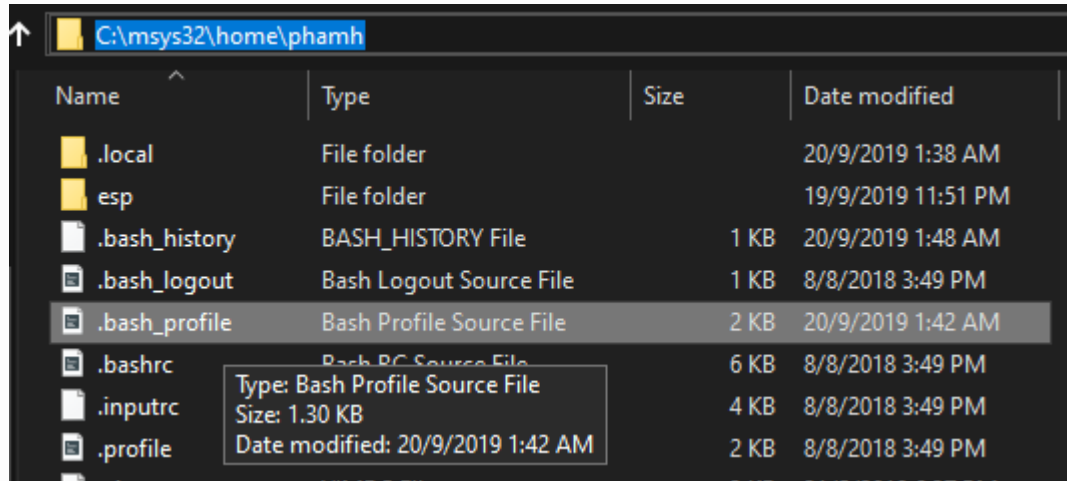
Toolchain cần thêm các biến môi trường để chạy :

IDF_PATH : cho thư viện esp-idf
PATH : cho trình biên dịch chéo

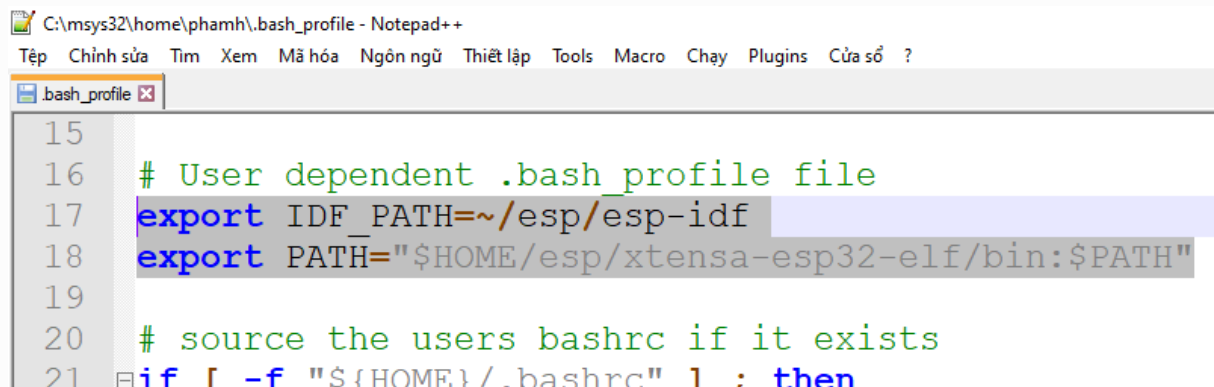
```
export IDF_PATH=~/esp/esp-idf  
export PATH="$HOME/esp/xtensa-esp32-elf/bin:$PATH"
```

Cập nhật 2 biến này vào file **.bash_profile**

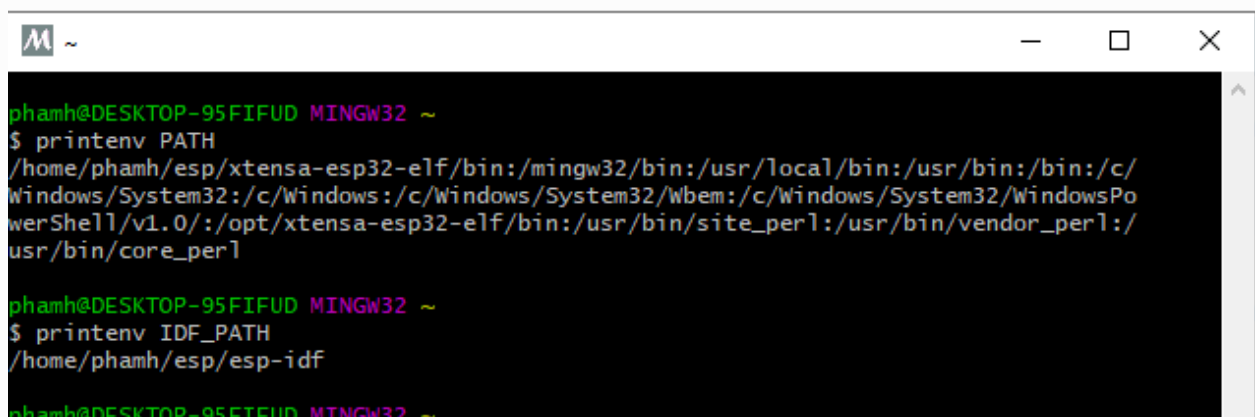
Ví dụ : C:\msys32\home\phamh



Mở file bằng trình edit bất kì :



Sau khi thêm , khởi động cửa sổ Shell chạy lệnh để kiểm tra : **printenv PATH**



Cài đặt các gói yêu cầu Python :

Các gói Python được yêu cầu bởi ESP-IDF được đặt trong

`$IDF_PATH/requirements.txt`

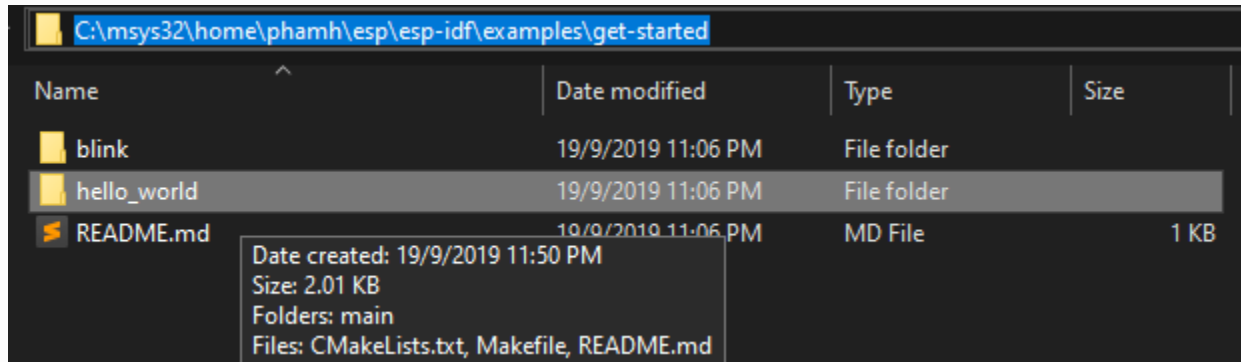
`python -m pip install --user -r $IDF_PATH/requirements.txt`

```
phamh@DESKTOP-95FIFUD MINGW32 ~  
$ python  
Python 2.7.15 (default, Sep 24 2018, 09:03:39) [GCC 7.3.0 32 bit] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
  
phamh@DESKTOP-95FIFUD MINGW32 ~  
$ python2.7 -m pip install --user -r $IDF_PATH/requirements.txt  
Requirement already satisfied: setuptools in c:/msys32/mingw32/lib/python2.7/site-packages (from -r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 4)) (40.4.3)  
Requirement already satisfied: pyserial>=3.0 in c:/msys32/mingw32/lib/python2.7/site-packages (from -r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 8)) (3.4)  
Requirement already satisfied: future>=0.15.2 in c:/msys32/mingw32/lib/python2.7/site-packages (from -r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 9)) (0.16.0)  
Requirement already satisfied: cryptography>=2.1.4 in c:/msys32/mingw32/lib/python2.7/site-packages (from -r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 10)) (2.3.1)  
Requirement already satisfied: pyparsing<2.4.0,>=2.0.3 in c:/msys32/mingw32/lib/python2.7/site-packages (from -r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 11)) (2.2.0)  
Requirement already satisfied: idna>=2.1 in c:/msys32/mingw32/lib/python2.7/site-packages (from cryptography>=2.1.4->-r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 10)) (2.7)  
Requirement already satisfied: asn1crypto>=0.21.0 in c:/msys32/mingw32/lib/python2.7/site-packages (from cryptography>=2.1.4->-r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 10)) (0.24.0)  
Requirement already satisfied: six>=1.4.1 in c:/msys32/mingw32/lib/python2.7/site-packages (from cryptography>=2.1.4->-r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 10)) (1.11.0)  
Requirement already satisfied: cffi!=1.11.3,>=1.7 in c:/msys32/mingw32/lib/python2.7/site-packages (from cryptography>=2.1.4->-r C:/msys32/home/phamh/esp/esp-idf/requirements.txt (line 10)) (1.11.0)
```

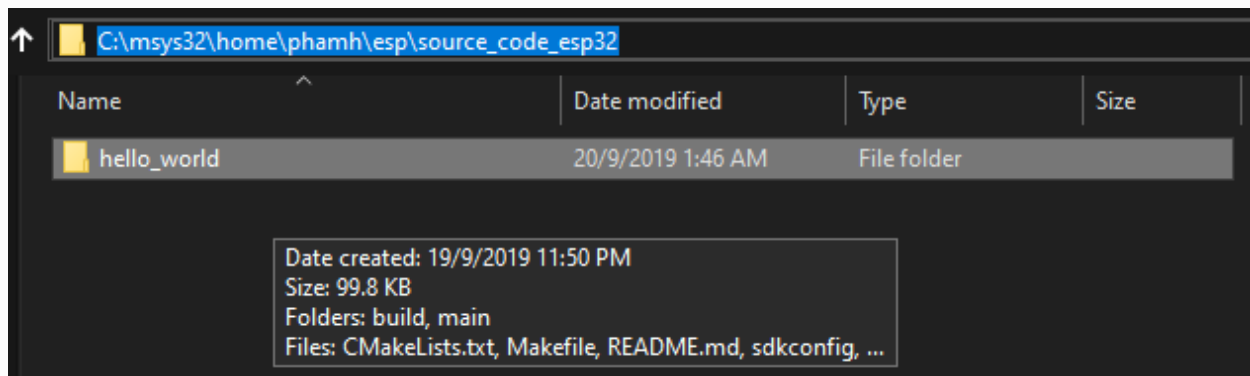
Start a Project : (đến dc phần này coi như đã đi 1/3 quãng đường rồi)

Chạy thử project bất kỳ .

Copy 1 project từ thư mục . esp\esp-idf\examples\get-started

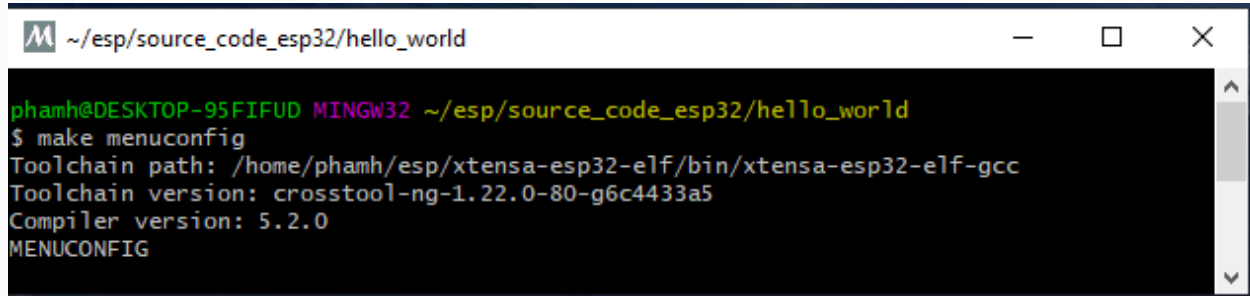


Paste vào đường dẫn : source_code_esp32 đã tạo trước đó :

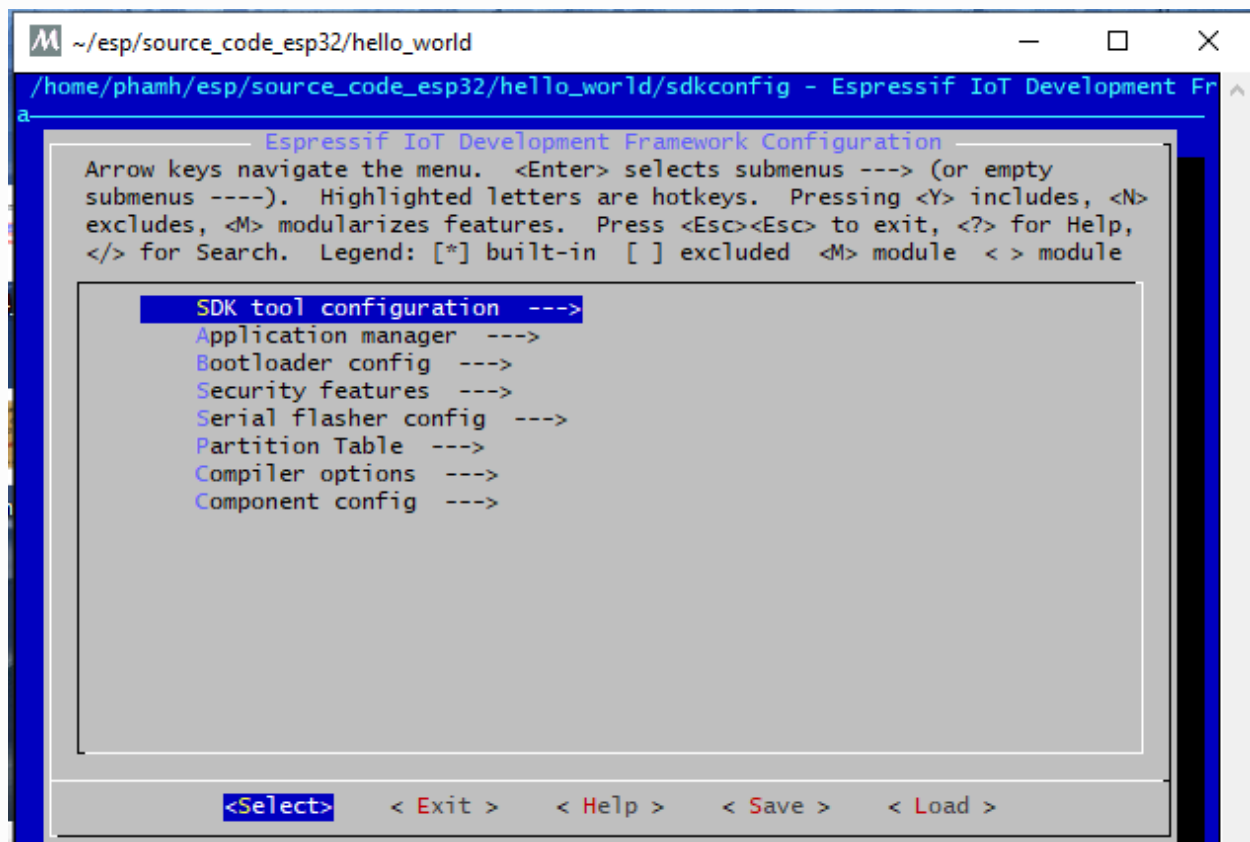


Mở Shell lên truy cập bằng dòng lệnh vào đường dẫn trên :

Nhập : **make menuconfig**



```
phamh@DESKTOP-95FIFUD MINGW32 ~/esp/source_code_esp32/hello_world
$ make menuconfig
Toolchain path: /home/phamh/esp/xtensa-esp32-elf/bin/xtensa-esp32-elf-gcc
Toolchain version: crosstool-ng-1.22.0-80-g6c4433a5
Compiler version: 5.2.0
MENUCONFIG
```



```
/home/phamh/esp/source_code_esp32/hello_world/sdkconfig - Espressif IoT Development Framework Configuration

Espressif IoT Development Framework Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ] excluded <M> module < > module

  SDK tool configuration --->
    Application manager --->
    Bootloader config --->
    Security features --->
    Serial flasher config --->
    Partition Table --->
    Compiler options --->
    Component config --->

<Select>  < Exit >  < Help >  < Save >  < Load >
```

Chú ý : cần trọng khi thao tác thay đổi trong menuconfig này , mỗi lần thay đổi config , tool sẽ buil lại source code , config ảnh hưởng trực tiếp lên RTOS Esp32 . đọc kỹ tài liệu+backup lại code trước khi thay đổi .

Phần cấu hình config sẽ để 1 phần riêng :v

Biên dịch : (yêu cầu chỉ cần biên dịch đc đã chưa cần nạp xuống kit)

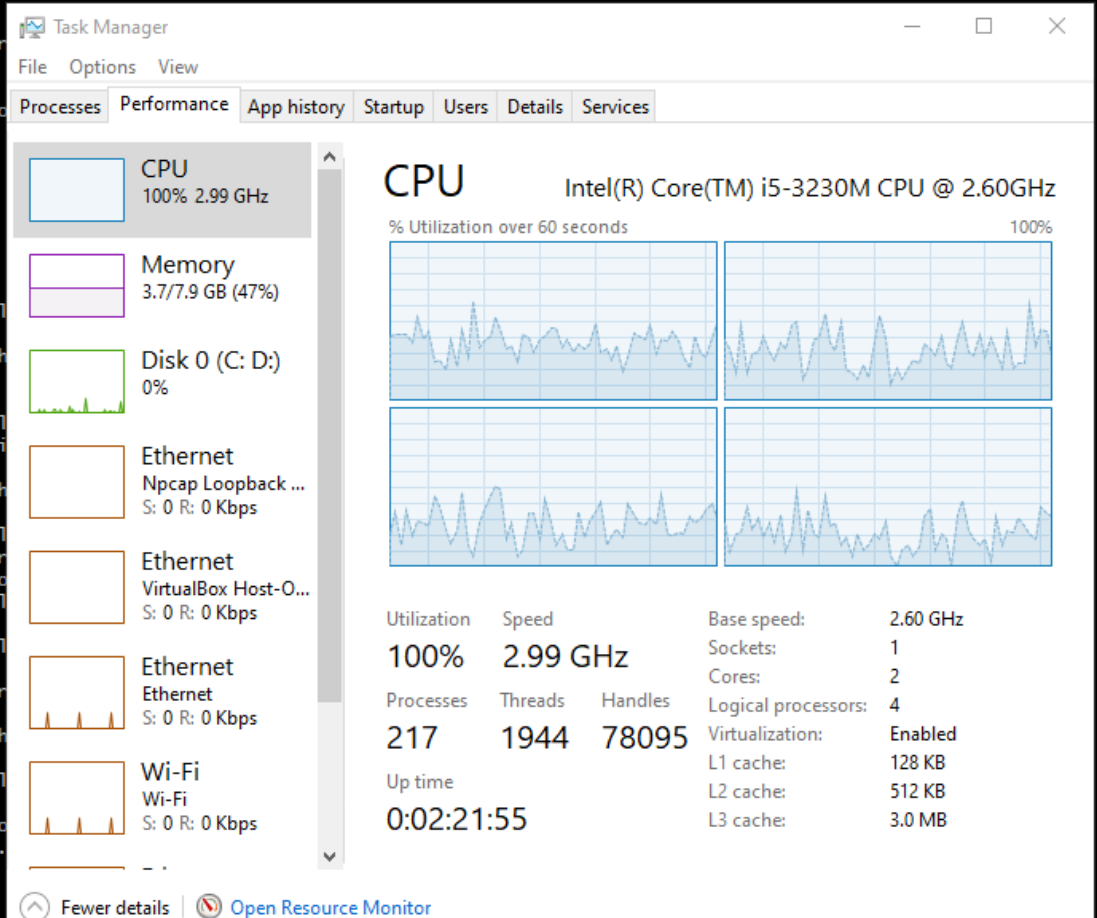
Sau khi thoát khỏi menuconfig , shell chạy lệnh :

```
# make -j4
```

Giải thích , make là 1 tool biên dịch code , tham số truyền vào [-j4] nếu máy tính của bạn có 4 luồng thì nó sẽ dùng cả 4 luồng để biên dịch code , đẩy xung nhịp lên cao , tốc độ sẽ nhanh hơn .

~/esp/source_code_esp32/hello_world

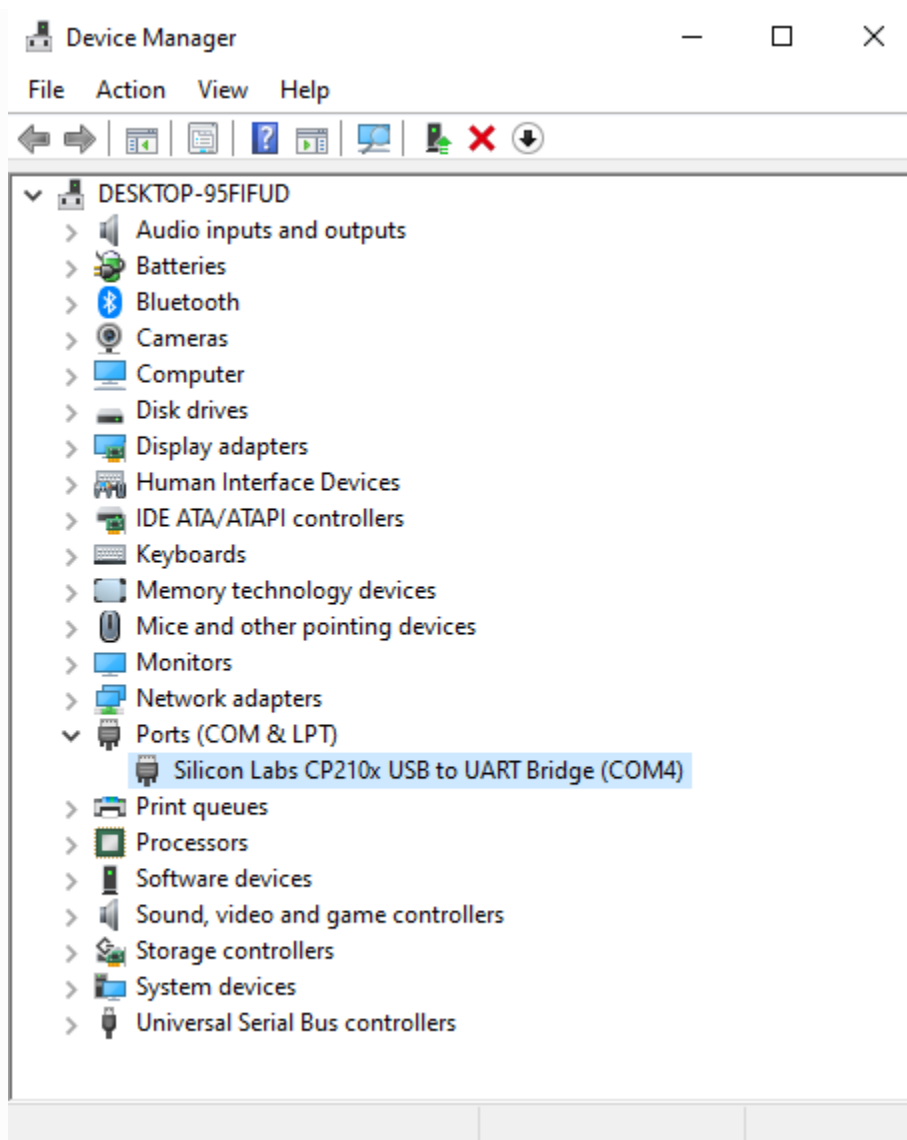
```
phamh@DESKTOP-95FIFUD MINGW32 ~/esp/source_code_esp32/hello_world
$ make -j4
Toolchain path: /home/phamh/esp/xtensa-esp32-elf/bin/xtensa-esp32-elf-gcc
Toolchain version: crosstool-ng-1.22.0-80-g6c4433a5
Compiler version: 5.2.0
CC build/app_trace/app_trace.o
Project is not inside a git repository
App "hello-world" version: 1
CC build/app_update/esp_app_desc.o
Python requirements from C:/msys32/ho
CC build/app_trace/host_file_io.o
CXX build/asio/asio/asio/src/asio.o
CC build/app_update/esp_ota_ops.o
CC build/app_trace/app_trace_util.o
CC build/app_trace/gcov/gcov_rtti.o
AR build/app_update/libapp_update.a
AR build/aws_iot/libaws_iot.a
AR build/app_trace/libapp_trace.a
CC build/bootloader_support/src/bootl
CC build/bt/bt.o
CC build/bootloader_support/src/flash
AR build/bt/libbt.a
CC build/coap/libcoap/src/address.o
CC build/bootloader_support/src/bootl
CC build/bootloader_support/src/esp_i
CC build/coap/libcoap/src/async.o
CC build/bootloader_support/src/flash
CC build/coap/libcoap/src/block.o
CC build/bootloader_support/src/bootl
CC build/bootloader_support/src/secur
CC build/coap/libcoap/src/coap_time.o
CC build/bootloader_support/src/bootl
CC build/coap/libcoap/src/debug.o
CC build/bootloader_support/src/bootl
CC build/coap/libcoap/src/encode.o
CC build/bootloader_support/src/secur
CC build/coap/libcoap/src/hashkey.o
CC build/bootloader_support/src/flash
CC build/coap/libcoap/src/mem.o
CC build/bootloader_support/src/bootl
CC build/coap/libcoap/src/net.o
AR build/bootloader_support/libbootl
CC build/console/linenoise/linenoise
AR build/asio/libasio.a
CXX build/cxx/cxx_exception_stubs.o
CC build/coap/libcoap/src/option.o
CC build/console/argtable3/argtable3
CXX build/cxx/cxx_guards.o
CC build/coap/libcoap/src/pdu.o
```



Note : done ...

Hướng dẫn cấu hình menuconfig và cách nạp xuống kit :

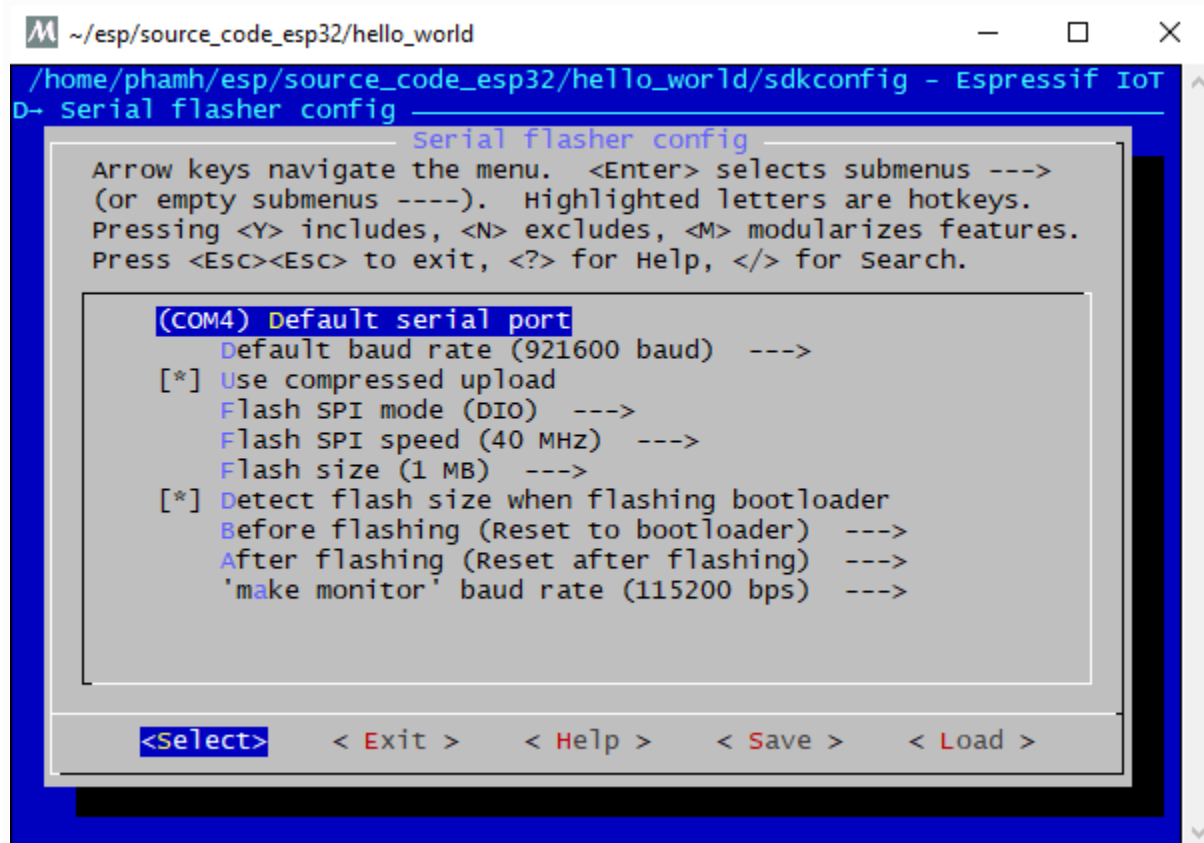
- Đầu tiên là chúng ta phải cắm kit vào laptop đã , không cần sao nạp đc :v (chú ý : check **USB cable** - USB A / micro USB B , một số dây cắm không nhận cổng COM , check **driver for cp210x usb** đã cài chưa)
- Kiểm tra kết nối và lấy tên cổng COM x



Ví dụ : cổng COM tại lap của mình là cổng [COM 4]

- Quay lại [make menuconfig]
 - Chọn [serial flasher config ---]
 - Default serial port : chuyển thành COM4
 - Baud rate : (921600)
 - Flash SPI mode : DIO
 - Flash SPI speed : 40Mhz
 - Flash SPI size :2MB

Chú ý : Các tham số còn lại để yên , make monitor baud rate 115200 tương tự như chúng ta khai báo `Serial.begin(115200);` Arduino



- Chuyển sang tab <SAVE> để lưu thông tin

- Trong [make menuconfig]
 - Chọn [compiler options]
 - Enable C++ exceptions ---

```

/home/phamh/esp/source_code_esp32/hello_world/sdkconfig - Espressif IoT
D- compiler options
Compiler options
Arrow keys navigate the menu. <Enter> selects submenus --->
(or empty submenus ----). Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features.
Press <Esc><Esc> to exit, <?> for Help, </> for Search.

Optimization Level (Debug (-Og)) --->
Assertion level (Enabled) --->
[*] Enable C++ exceptions --->
Stack smashing protection mode (None) --->
[ ] Enable -Wwrite-strings warning flag
[ ] Disable new warnings introduced in GCC 6 - 8
  
```

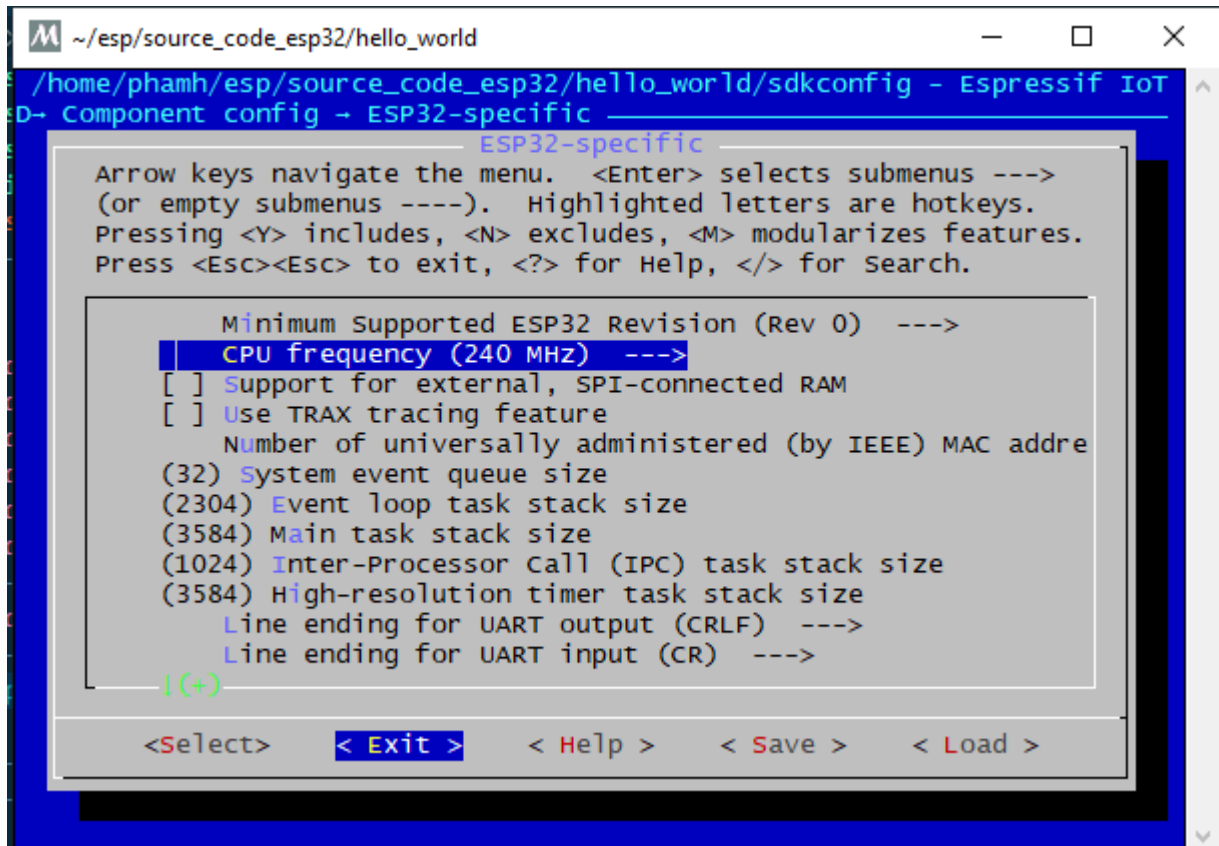
Note giải thích :

Toàn bộ thư viện ESP RTOS IDF đc viết bằng ngôn ngữ C để tối ưu hóa , do đó chúng ta không thể dùng các bộ thư viện C++ . bật tính năng này lên , sau đó trong code hay bọc hàm app_main() vào nhé .

```

main > C++ main.cpp > ...
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <iostream>
5  using namespace std;
6  /*-----*/
7  extern "C"
8  {
9      #include "freertos/FreeRTOS.h"
10     #include "freertos/task.h"
11     #include "freertos/queue.h"
12     #include "freertos/timers.h"
13     #include "driver/gpio.h"
14     #include "sdkconfig.h"
15     /*-----*/
16     #include "esp_system.h"
17     /*-----*/
18     void app_main();
19 }
20 /*-----*/
21 /*-----*/
  
```

- Trong [make menuconfig]
 - Chọn [component config---]
 - Chọn [ESP32-sepcific ----]
 - Chọn [CPU frequency ---]
 - Cấu hình tần số hoạt động của chip 240MHz



Chú ý : ngoài ra không đc thay đổi bất kỳ tham số nào trong này .

NẠP CHƯƠNG TRÌNH ĐẦU TIÊN XUỐNG KIT :

Sau khi kết thúc quá trình config cho project , <save> và <exit> tại đường dẫn thư mục của project, chạy lệnh Shell :

```
#make -j4 flash monitor  
flash : tool nạp xuống kit  
monitor : tool serial kết nối đến chip theo ( make monitor baud rate 115200)
```

```
M ~/esp/source_code_esp32/hello_world  
phamh@DESKTOP-95FIFUD MINGW32 ~/esp/source_code_esp32/hello_world  
$ make -j4 flash monitor  
Toolchain path: /home/phamh/esp/xtensa-esp32-elf/bin/xtensa-esp32-elf-gcc  
Toolchain version: crosstool-ng-1.22.0-80-g6c4433a5  
Compiler version: 5.2.0  
GENCONFIG  
Project is not inside a git repository, will not use 'git describe' to deter
```

Kết thúc quá trình biên dịch , đến quá trình nạp :

```
M ~/esp/source_code_esp32/hello_world  
AR build/wpa_suppllicant/libwpa_suppllicant.a  
Generating libwpa_suppllicant.a.sections_info  
Generating esp32.project.ld  
LD build/hello-world.elf  
esptool.py v2.8-dev  
Flashing binaries to serial port COM4 (app at offset 0x10000)...  
esptool.py v2.8-dev  
Serial port COM4  
Connecting.....  
Chip is ESP32D0WDQ6 (revision 1)  
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
```

Chú ý : tại đây khi chạy connecting..... có 1 số trường hợp không nạp đc do cổng UART vẫn đang đẩy dữ liệu , để giải quyết việc này khi tool chạy đến đây các bạn ấn giữ button boot trên kit :v

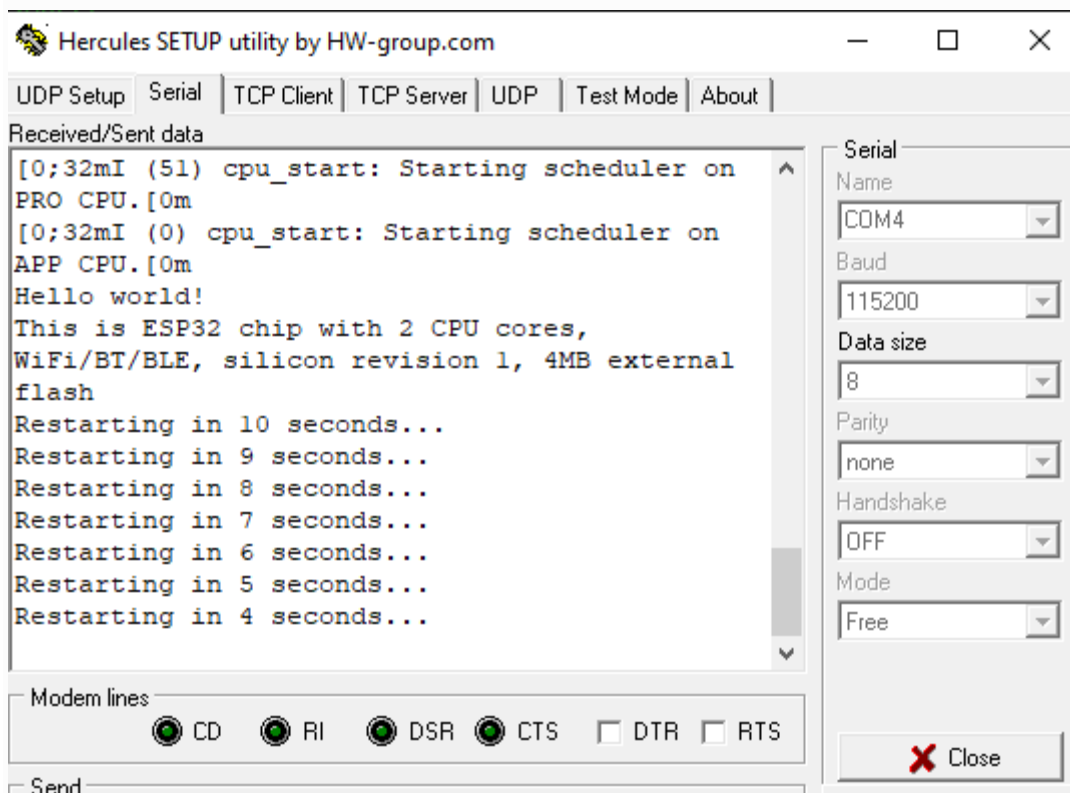


Sau khi nạp xong tool sẽ chạy luôn lệnh monitor mở serial :

```
~/esp/source_code_esp32/hello_world
I (51) cpu_start: Starting scheduler on PRO CPU.
I (0) cpu_start: Starting scheduler on APP CPU.
Hello world!
This is ESP32 chip with 2 CPU cores, WiFi/BT/BLE, silicon revision 1, 4M
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
Restarting in 7 seconds...
Restarting in 6 seconds...
Restarting in 5 seconds...
Restarting in 4 seconds...
Restarting in 3 seconds...
Restarting in 2 seconds...
Restarting in 1 seconds...
Restarting in 0 seconds...
Restarting now.
ets Jun  8 2016 00:22:57
```

Nhấn : Ctrl + C , để kết thúc , chúng ta vẫn có thể dùng tool ngoài cũng đc

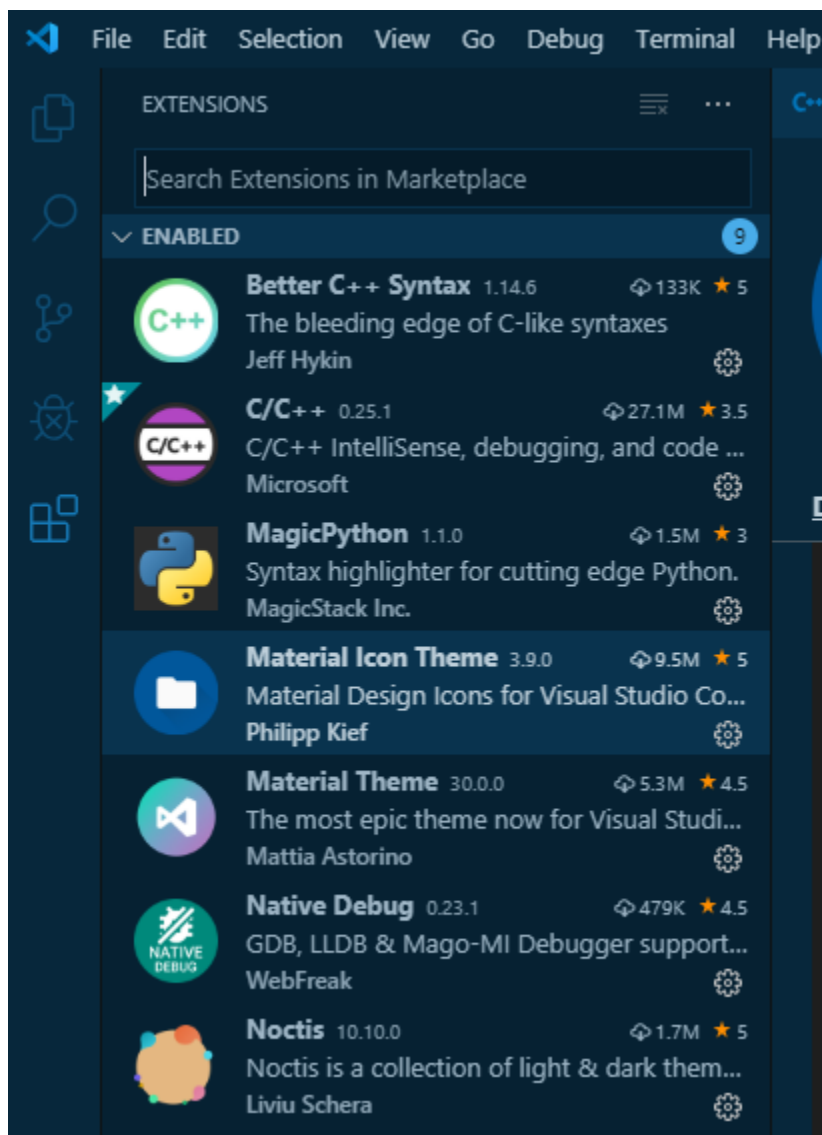
Ví dụ :



Hướng dẫn sử dụng tích hợp edit : visual studio code

Trên web <https://docs.espressif.com> có hướng dẫn cách sử dụng [Build and Flash with Eclipse IDE], trong khuôn khổ bài viết mình sẽ hướng dẫn sử dụng Vscode . Cách cài đặt Vscode lên máy ở trên mạng có rất nhiều . mọi người có thể tham khảo trước khi đọc phần này .

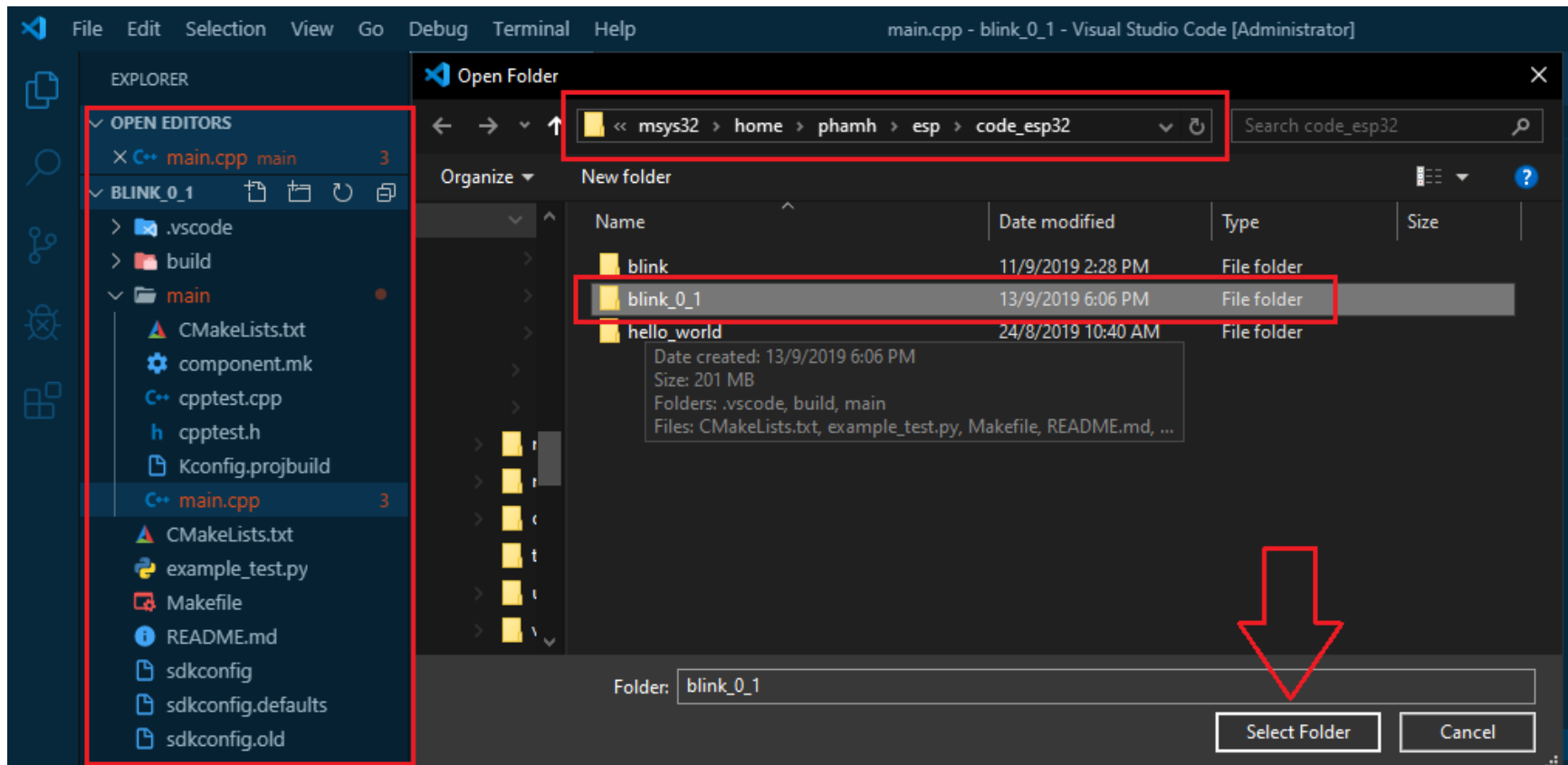
- Mở VScode
 - Chọn EXTENSIONS để cài các plugin sau :
 - C/C++
 - Native Debug
 - Theme Noctis



Khởi động lại VS code :

1. Chọn File

a. Chọn Open folder (Chọn vào folder chứa project cho esp32)

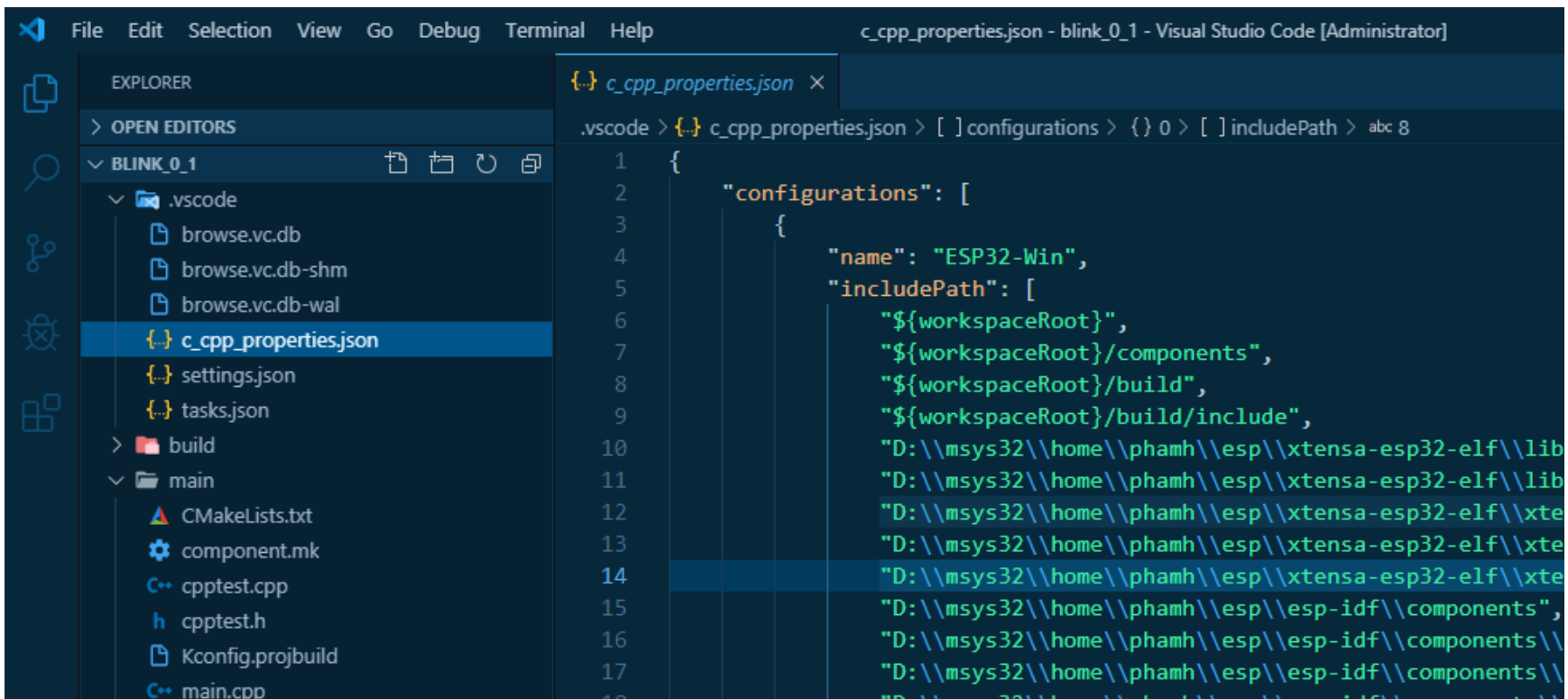


Chú ý : kiểm tra cây thư mục bên tay trái xem đã đúng chưa .

Trong cây thư mục project vừa add vào VS code :

- Create a folder name [.vscode]
 - Create a file name [c_cpp_properties.json]

chú ý : file json này mình sẽ để đính kèm trong thư mục nén rar , hãy thay đổi đường dẫn trong “includePath” để cho tương thích với môi trường phát triển của riêng bạn .



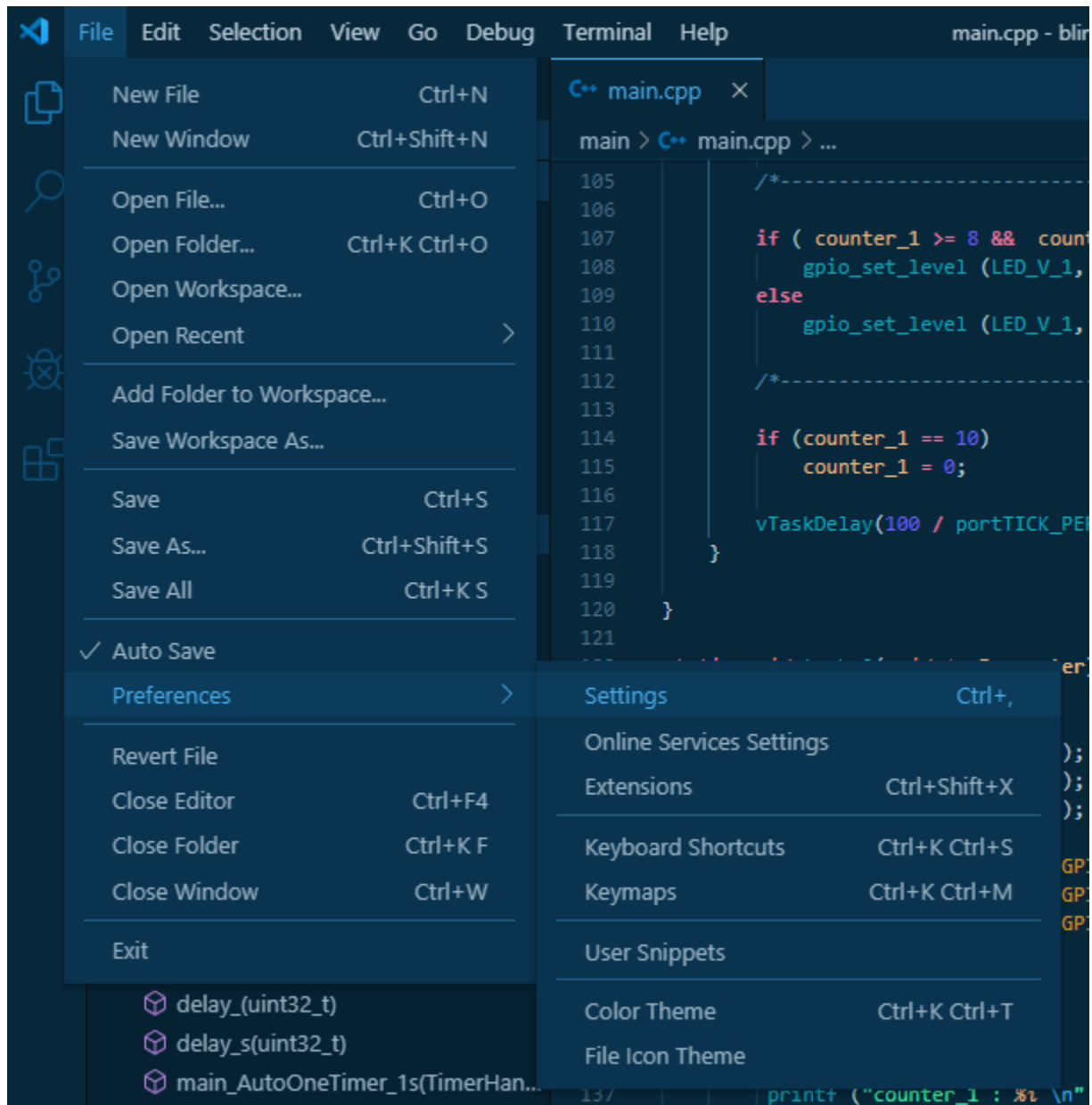
```
File Edit Selection View Go Debug Terminal Help c_cpp_properties.json - blink_0_1 - Visual Studio Code [Administrator]

EXPLORER
> OPEN EDITORS
▼ BLINK_0_1
  ▼ .vscode
    browse.vc.db
    browse.vc.db-shm
    browse.vc.db-wal
    c_cpp_properties.json
    settings.json
    tasks.json
  > build
  ▼ main
    CMakeLists.txt
    component.mk
    cpptest.cpp
    cpptest.h
    Kconfig.projbuild
    main.cpp

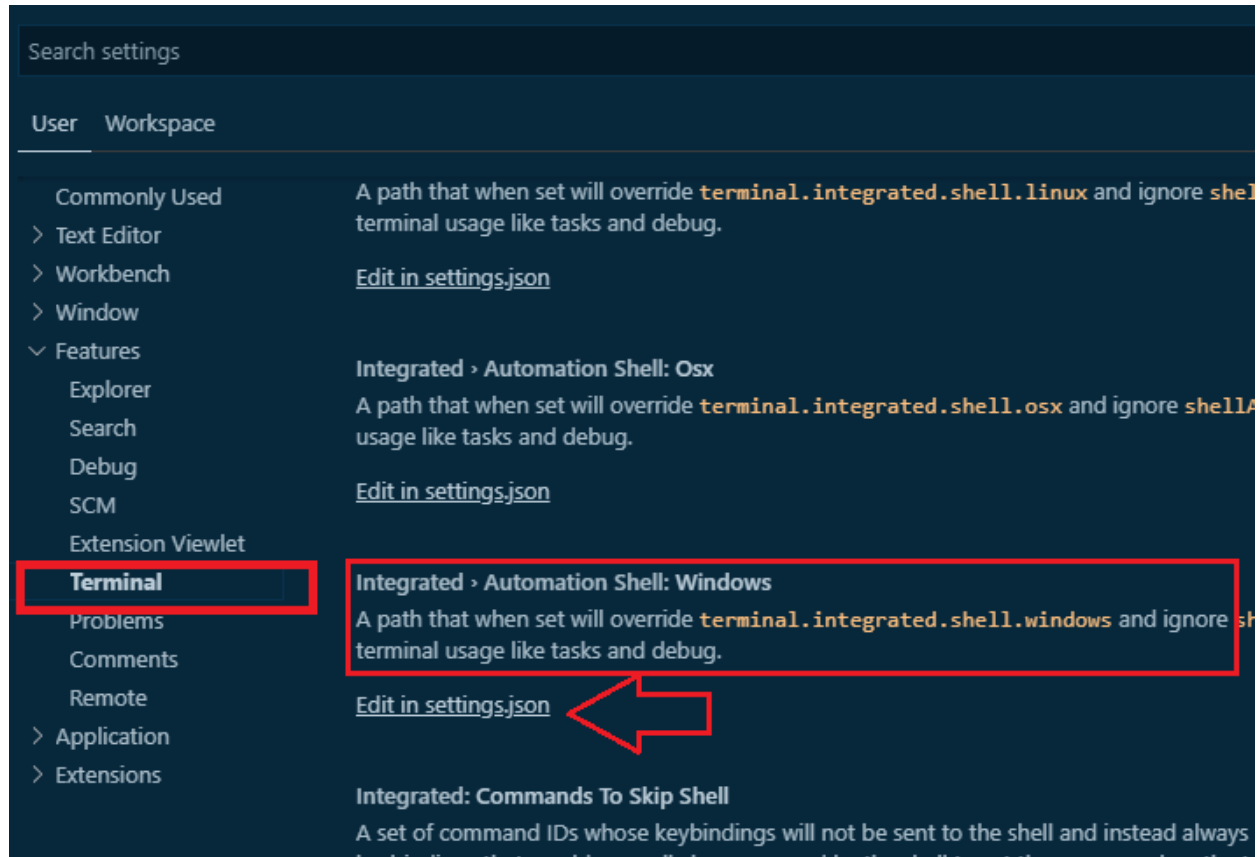
c_cpp_properties.json
1 {
2   "configurations": [
3     {
4       "name": "ESP32-Win",
5       "includePath": [
6         "${workspaceRoot}",
7         "${workspaceRoot}/components",
8         "${workspaceRoot}/build",
9         "${workspaceRoot}/build/include",
10        "D:\\msys32\\home\\phamh\\esp\\xtensa-esp32-elf\\lib",
11        "D:\\msys32\\home\\phamh\\esp\\xtensa-esp32-elf\\lib",
12        "D:\\msys32\\home\\phamh\\esp\\xtensa-esp32-elf\\lib",
13        "D:\\msys32\\home\\phamh\\esp\\xtensa-esp32-elf\\lib",
14        "D:\\msys32\\home\\phamh\\esp\\xtensa-esp32-elf\\lib",
15        "D:\\msys32\\home\\phamh\\esp\\esp-idf\\components",
16        "D:\\msys32\\home\\phamh\\esp\\esp-idf\\components",
17        "D:\\msys32\\home\\phamh\\esp\\esp-idf\\components",
18        "D:\\msys32\\home\\phamh\\esp\\esp-idf\\components"
```


Tích hợp của sổ shell bash của Msys2 vào VScode terminal :

Click [File – Preferences – Setting]



Clike [Setting – Terminal – Integrated > Automation Shell : Windows]



In file [setting.json] add :

Trong đó thay đổi đường dẫn “:/msys32/usr/bin/bash.exe” sao cho phù hợp

```
{
  "terminal.integrated.shell.windows": "C:/msys32/usr/bin/bash.exe",
  "terminal.integrated.shellArgs.windows": [
    "--login",
  ],
  "terminal.integrated.env.windows": {
    "CHERE_INVOKING": "1",
    "MSYSTEM": "MINGW32",
  },
}
```

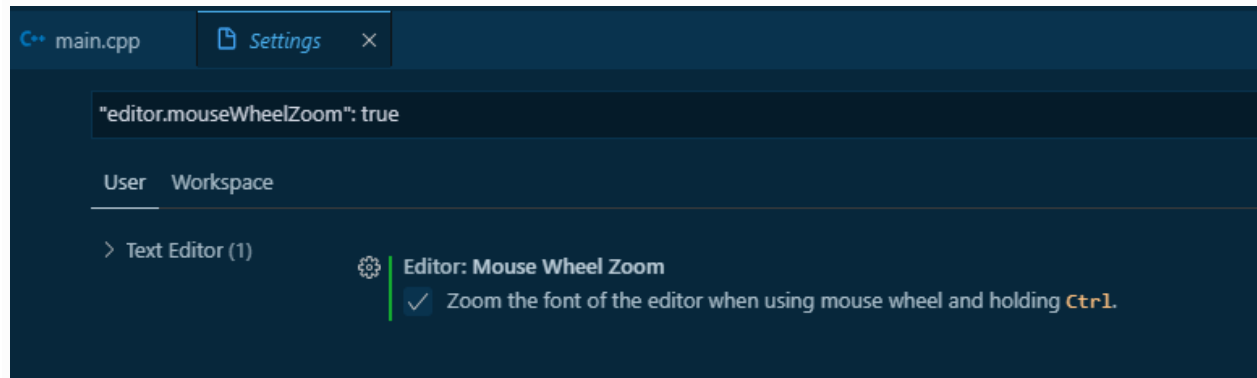
Khởi động lại VS code ! done !

link tham khảo :

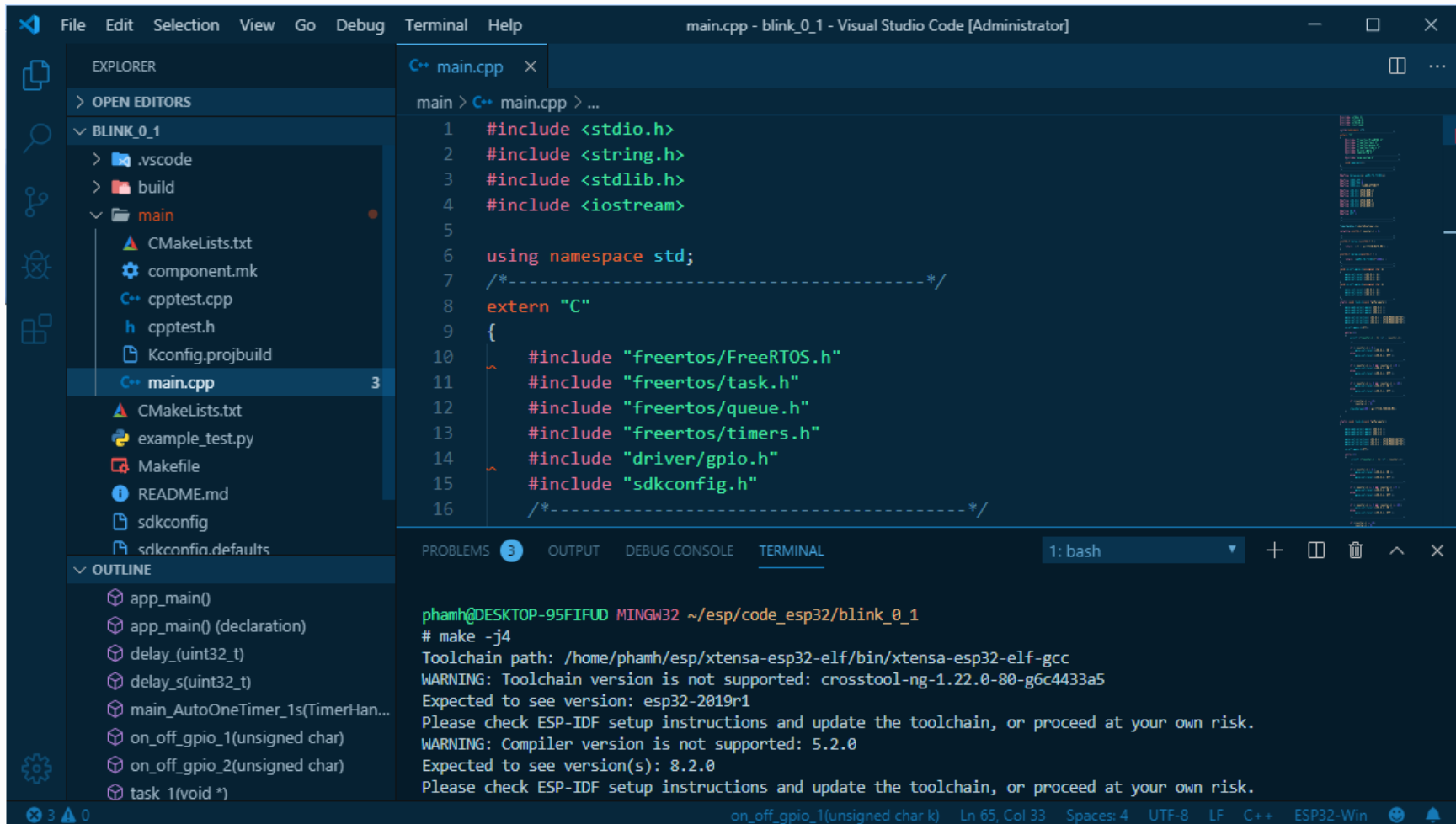
<https://github.com/VirgiliaBeatrice/esp32-devenv-vscode/blob/master/tutorial.md>

Thêm : thay đổi zoom bàn phím trên VS code bằng chuột

`"editor.mouseWheelZoom": true`



Tổng kết : Sau tất cả các bước trên , các bạn đã có một bộ tool tích hợp trong cùng 1 trình edit .
Để mở cửa sổ bash shell msys2 trong VS code [Ctrl + ~]



Note : quy trình ...

B1 : Tạo project Esp32 , đặt trong thư mục môi trường MSYS2

B2 : mở VScode và trở đến thư mục đây .

B3 : (bước này nếu ai dùng Git thì có thể đồng bộ lên server .)

B4 : mở bash shell chạy make menuconfig để cấu hình 1 lần

B5 : biên dịch thử , biên dịch lần đầu tiên sẽ tốn thời gian hơn các lần sau .

B6 : đọc tài liệu và code

Tài liệu tham khảo :

<https://docs.espressif.com/projects/esp-idf/en/latest/index.html>

<https://github.com/nkolban/esp32-snippets/tree/master/VisualStudioCode>

<https://github.com/VirgiliaBeatrice/esp32-devenv-vscode/blob/master/tutorial.md>