**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN KỸ THUẬT ĐIỆN TỬ**

**\*\*\*\*\*\*\*\*\*\***

****

**BÁO CÁO MÔN HỌC**

**Giáo viên hướng dẫn :**

**Phạm Trung Hiếu CA180207**

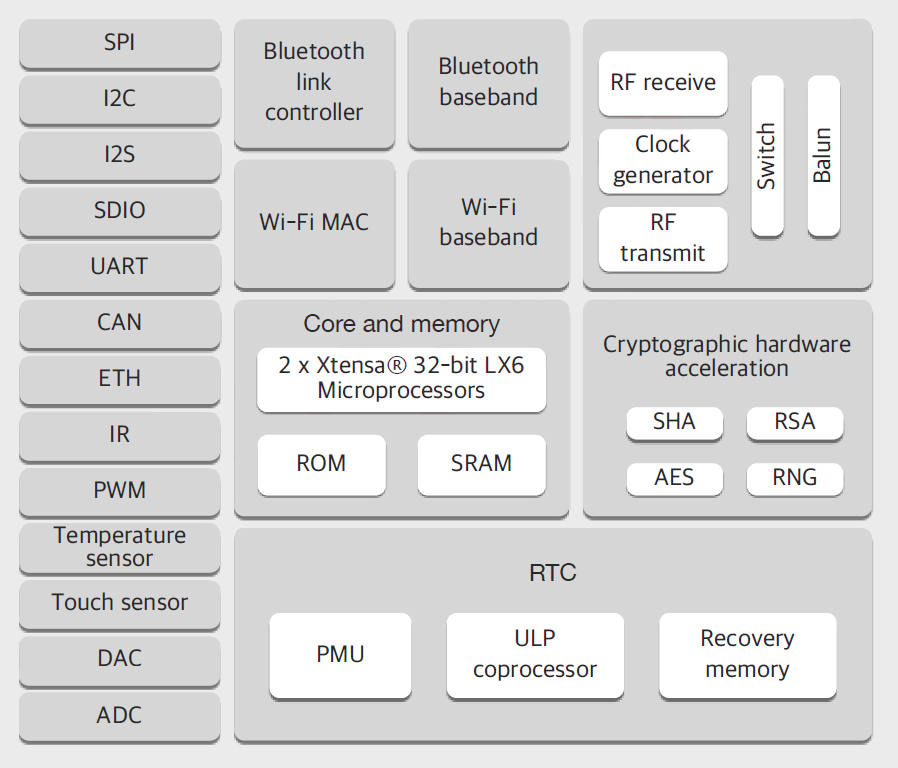
***Hà Nội, 9 – 2019***

**Hướng dẫn :**

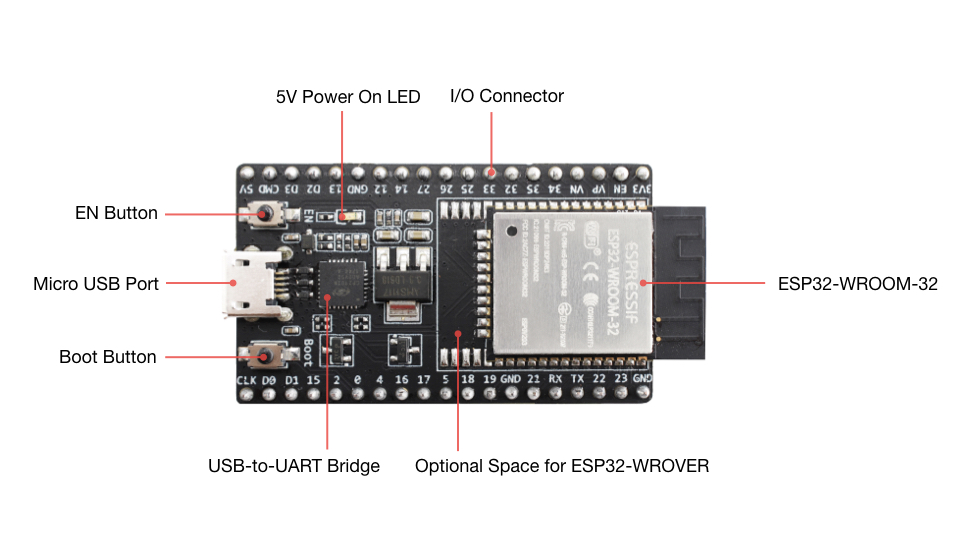
Đây là tài liệu cho phiên bản ổn định v3.3 của ESP-IDF.

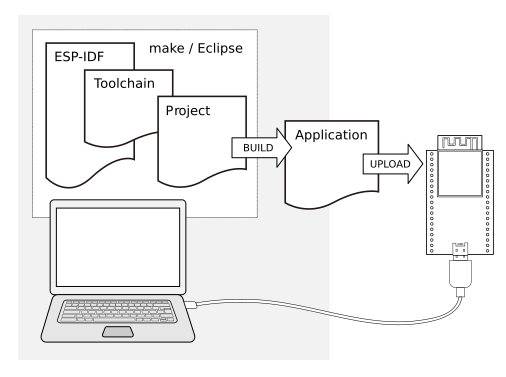
Trong bài hướng dẫn mình sử dụng phiên bản v3.3

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 bổ 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 tụ Linux , GNU Autotools ….

Toolchain :

Trình biên dịch chéo

Thư viện ESP - IDF

Trình soạn thảo : visual studio code

ESP32 NodeMCU

Source code ESP32

Tool make file đc viết bằng python :  
# make menuconfig  
# make -j4 flash monitor

Chú ý : trên web chính thức của hãng [https://docs.espressif.com](https://docs.espressif.com/) có hướng dẫn cài đặt môi trường pháp triển đầy đủ cho Windows , Linux , Mac OS ..  
trong phần này các bạn có thể hình dung ra môi trường cài đặt , các phần mềm liên quan tới nhau ….

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.

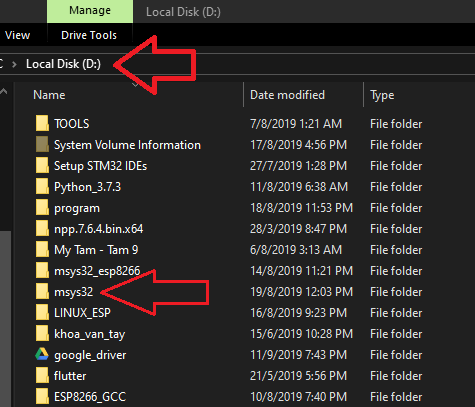
Trong bài viết này mình sẽ hướng dẫn setup môi trường sử dụng MSYS2- Unix . Môi trường này chạy độc lập trên win và có thể copy toàn bộ thư mục môi trường cũng thư viên đi bất kỳ máy tính cá nhân nào có sử dụng Win mà không cần cài đặt lại môi trường phát triển , dễ quản lý và backup nếu gặp sự cố .

**Thiết lập môi trường :**

Tải xuống tệp zip toolchain & MSYS2 tất cả trong một của Windows từ dl.espressif.com:

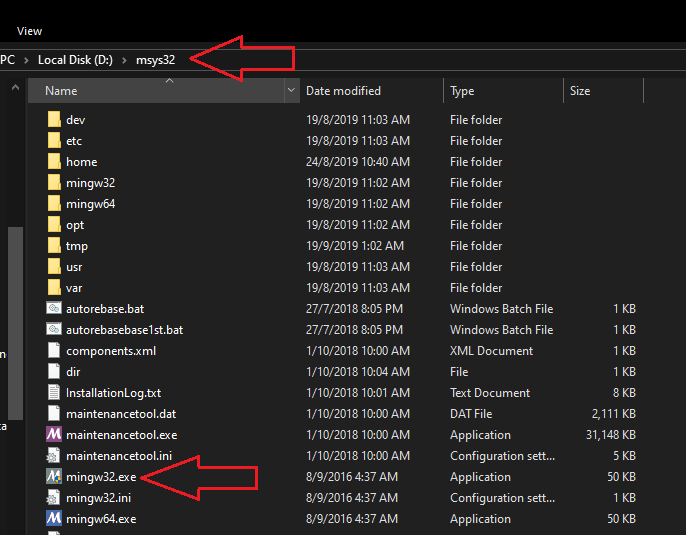
<https://dl.espressif.com/dl/esp32_win32_msys2_environment_and_toolchain-20181001.zip>

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.

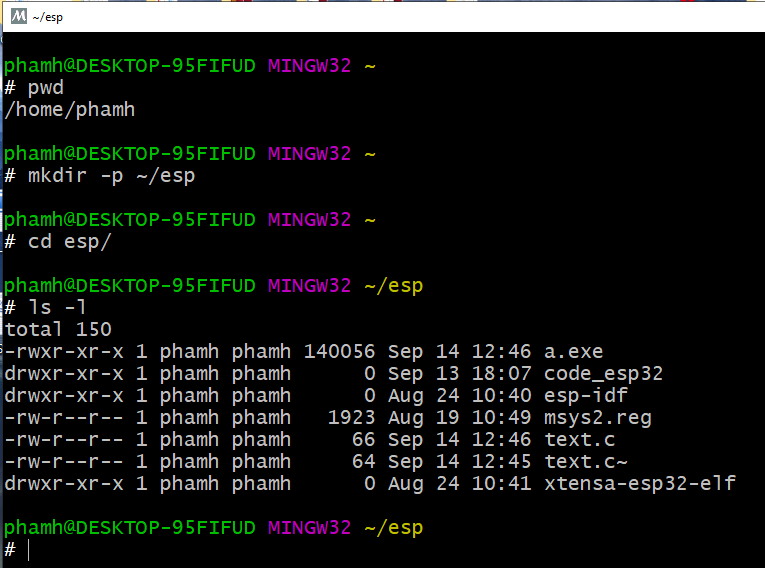


Kiểm tra môi trường vừa giải nén :

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



Sau khi cửa sổ đc mở nhập các lệnh sau :



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 hoăc 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.

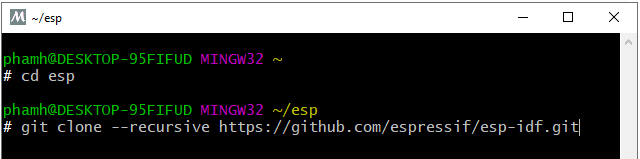
Trong bài hướng dẫn mình sử dụng phiên bản v3.3

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

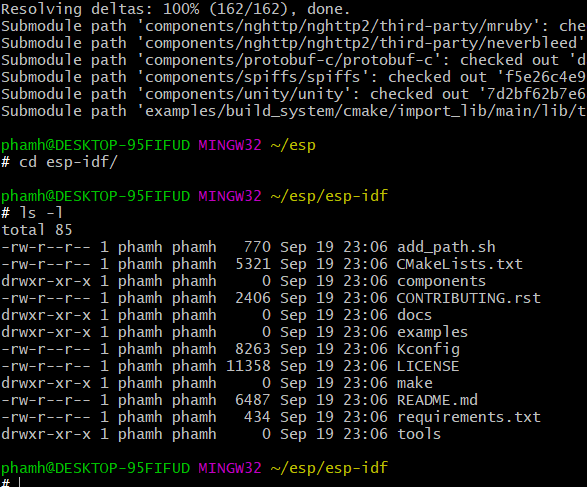


ESP-IDF sẽ được tải xuống  : ~/esp/esp-idf.   
Bên trong MSYS2 đã có cài sẵn Git , chúng ta chỉ cần thực hiện lệnh .

*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 .*

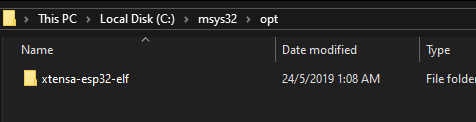
*Nên nhất thiết phải để thư viện* **[ esp-idf ]***bên trong cấu trúc thư mục của msys2 .*

Sau quá trình clone về máy thành công :

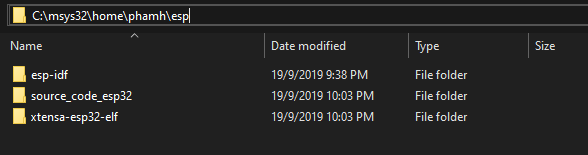
Thực hiện các lệnh như sau để kiểm tra cấu trúc thư mục :  
  


**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 chưa toàn bộ project muốn phát triển sau này cho vào đây .



**Add biến môi trường IDF\_PATH :**

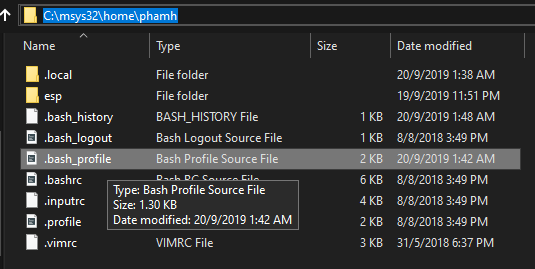
Toolchain cần thêm các biến môi trường để chạy :

IDF\_PATH : cho thư viện esp-idf   
PATH : cho trình biên dịch chéo

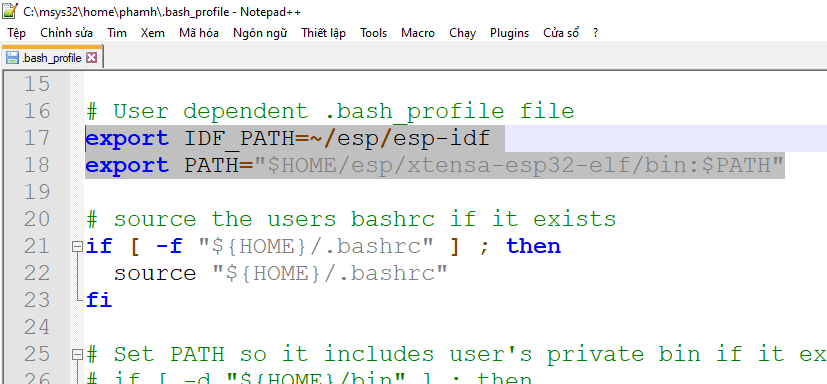
export IDF\_PATH=~/esp/esp-idf   
export PATH="$HOME/esp/xtensa-esp32-elf/bin:$PATH"

Cập nhật 2 biến này vào file .bash\_profile

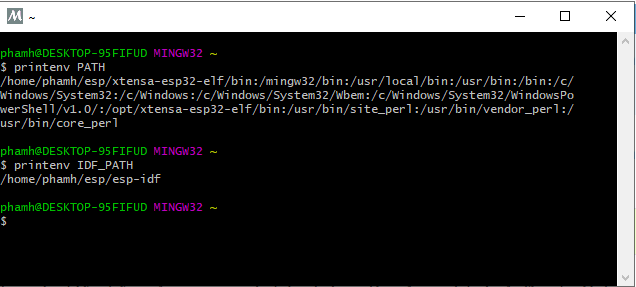
Vi dụ : C:\msys32\home\phamh



Mở file bằng trình edit bất kì :



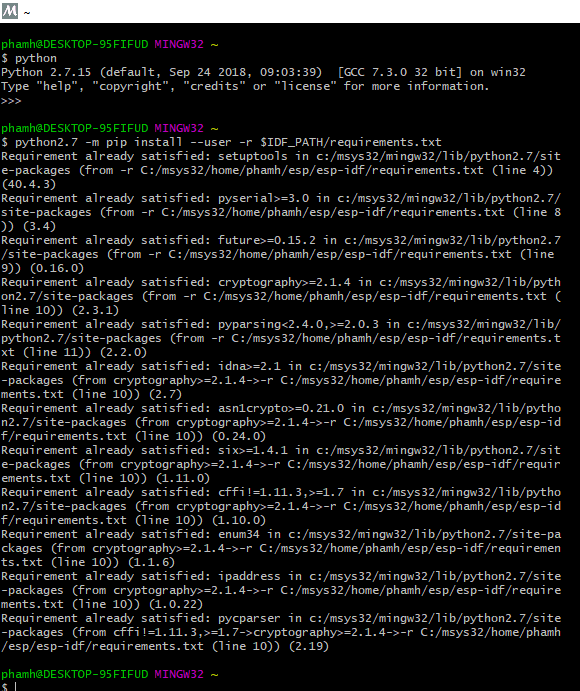
Sau khi thêm , khởi động cửa sổ Shell chạy lệnh để kiểm tra : printenv PATH



**Cài đặt các gói yêu cầu Python :**

Các gói Python được yêu cầu bởi ESP-IDF được đặt trong   
  
$IDF\_PATH/requirements.txt

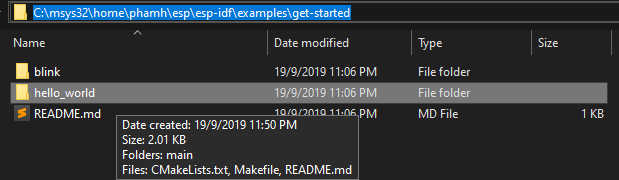
python -m pip install --user -r $IDF\_PATH/requirements.txt



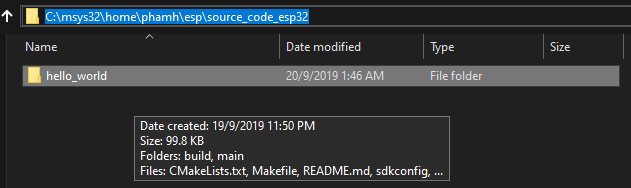
**Start a Project :** (đến đc phần này coi như đã đi 1/3 quãng đường rồi )

Chạy thử project bất kỳ .

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

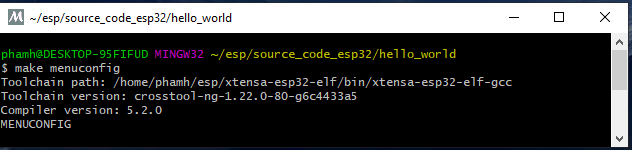


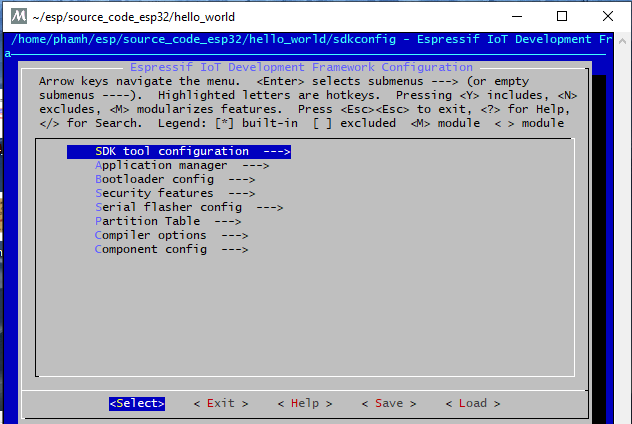
Paste vào đường dẫn : source\_code\_esp32 đã tạo trước đó :



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

Nhập : make menuconfig



**

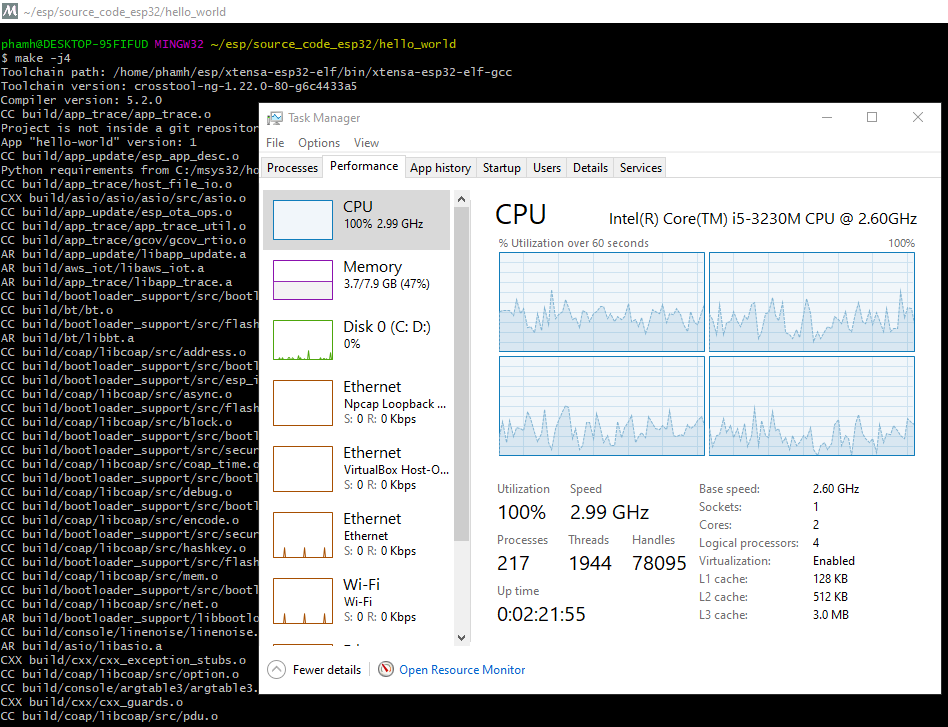
*Chú ý : cẩn trọng khi thao tác thay đổi trong menuconfig này , mỗi lần thay đổi config , tool sẽ buil lại source code , config ảnh hưởng trực tiếp lên RTOS Esp32 . đọc kỹ tài liệu+backup lại code trước khi thay đổi .*

**Phần cấu hình config sẽ để 1 phần riêng :v**

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

Sau khi thoát khỏi menuconfig , shell chạy lệnh :

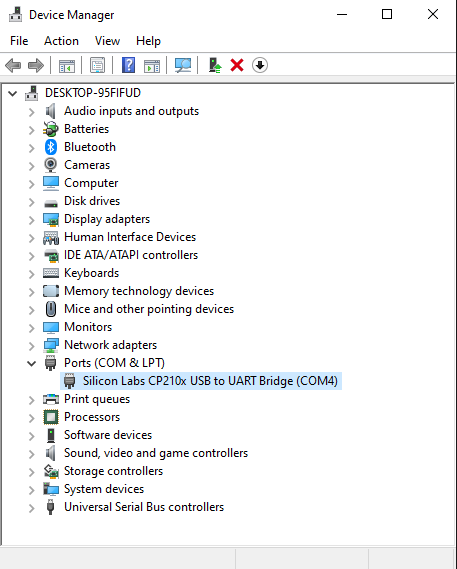
# make –j4

 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 , đấy xung nhịp lên cao , tốc độ sẽ nhanh hơn .

Note : done …

**Hướng dẫn cấu hình menuconfig và cách nạp xuống kít :**

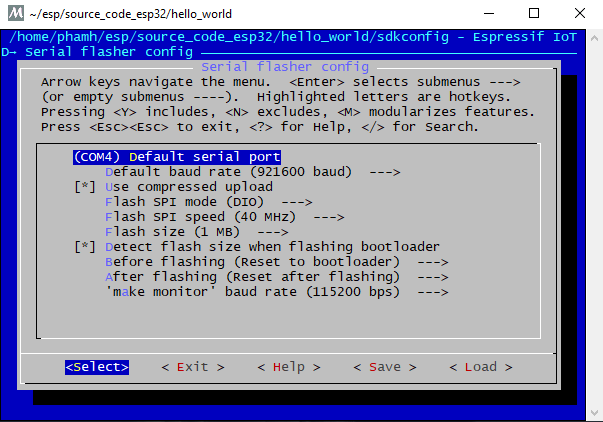
* Đầu tiên là chúng ta phải cắm kit vào laptop đã , không cắm sao nạp đc :v  
  (chú ý : check **USB cable** - USB A / micro USB B , một số dây cắm không nhận cổng COM , check driver for cp210x usb đã cài chưa )
* Kiểm tra kết nối và lấy tên cổng COM x



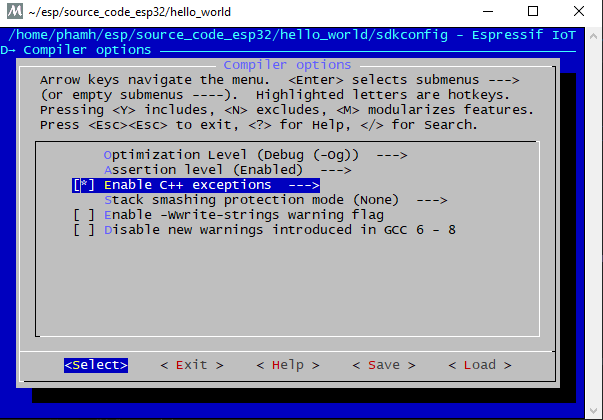
Ví dụ : cổng COM tại lap của mình là cổng [ COM 4 ]

* 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

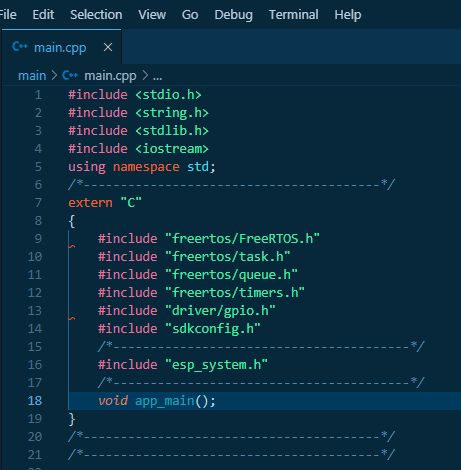
Chú ý : Các tham số còn lại để yên , make monitor baud rate 115200 tương tự như chúng ta khai báo Serial.begin(115200); Arduino



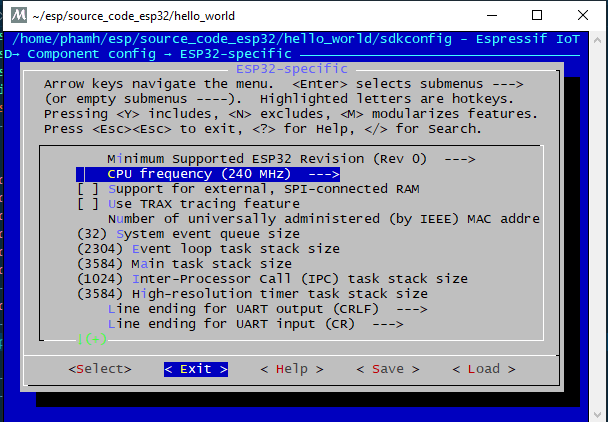
* Chuyển sang tab <SAVE> để lưu thông tin
* Trong [make menuconfig]
  + Chọn [compiler options]
    - Enable C++ exceptions ---



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 .

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

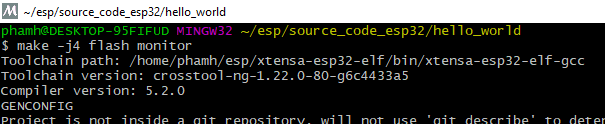


Chú ý : ngoài ra không đc thay đổi bất kỳ tham số nào trong này .

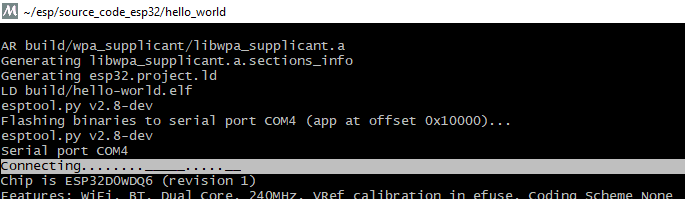
NẠP CHƯƠNG TRÌNH ĐẦU TIÊN XUỐNG KIT :

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

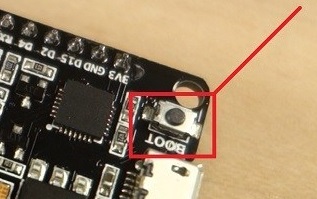
#make –j4 flash monitor  
flash : tool nạp xuống kit  
monitor : tool serial kết nối đến chip theo ( make monitor baud rate 115200)



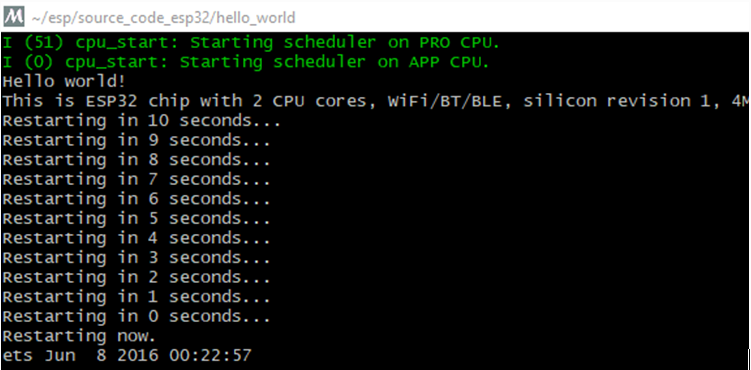
Kết thúc quá trình biên dịch , đến quá trình nạp :



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

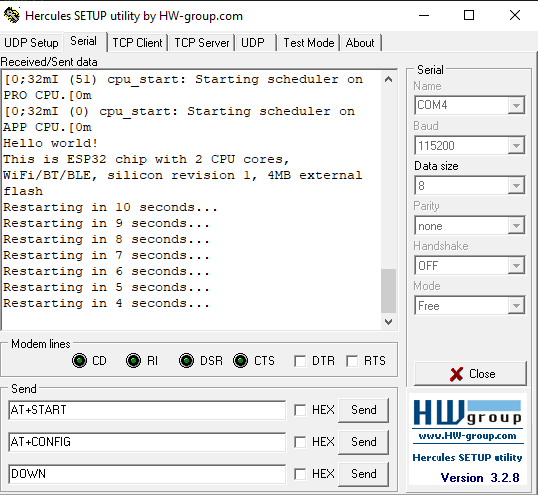


Sau khi nạp xong tool sẽ chạy luôn lệnh monitor mở serial :



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

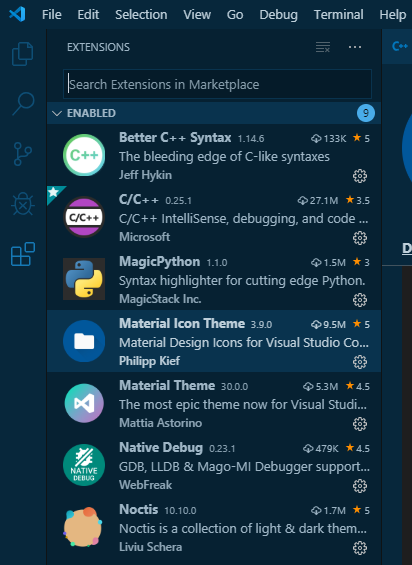
Ví dụ :



Hướng dẫn sử dụng tích hợp edit : visual studio code

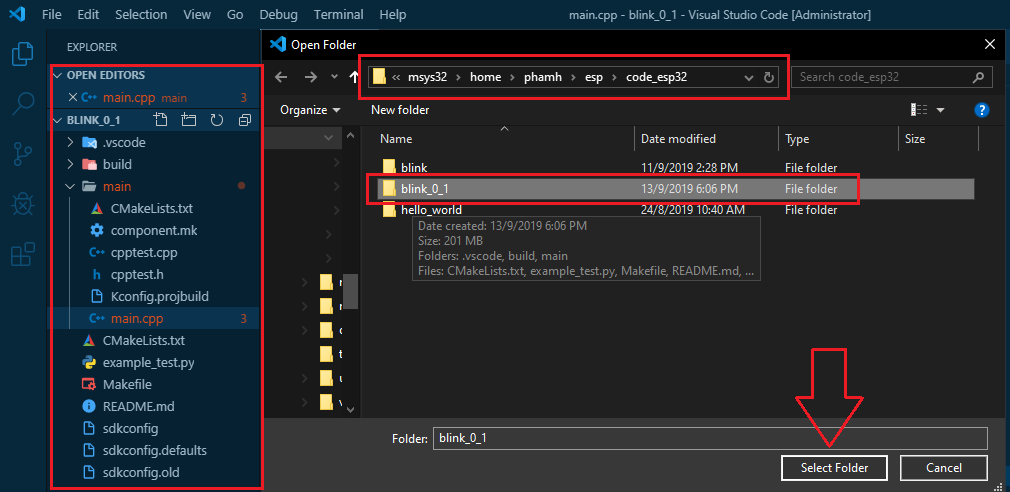
# Trên web [https://docs.espressif.com](https://docs.espressif.com/) có hướng dẫn cách sử dụng [ Build and Flash with Eclipse IDE ] , trong khuôn khổ bài viết mình sẻ hướng dẫn sử dụng Vscode . Cách cài đặt Vscode lên máy ở trên mạng có rất nhiều . mọi người có thể tham khảo trước khi đọc phần này .

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



Khởi động lại VS code :

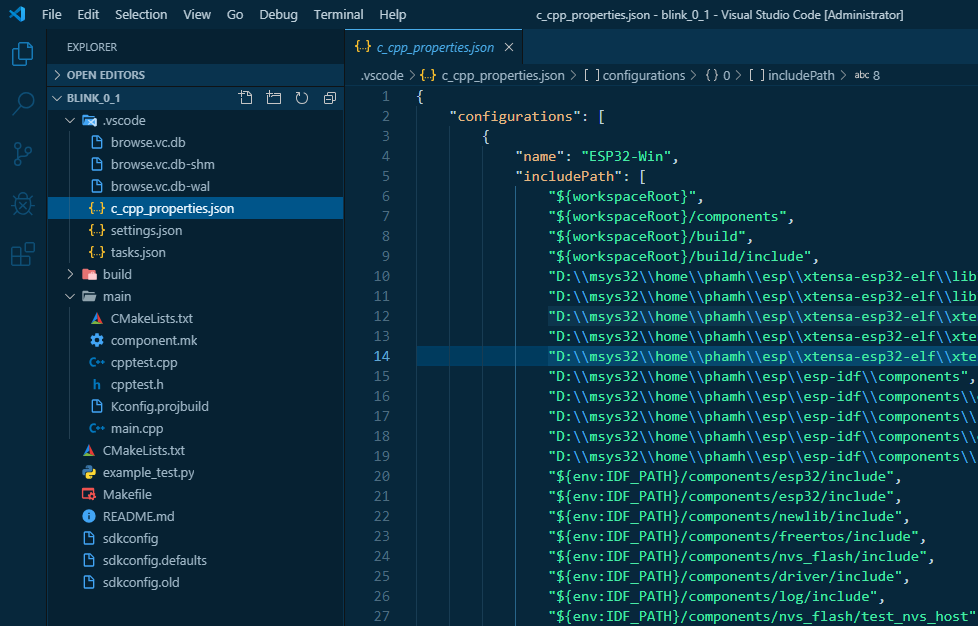
1. Chọn File
   1. Chọn Open folder (Chọn vào folder chưa project cho esp32 )



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

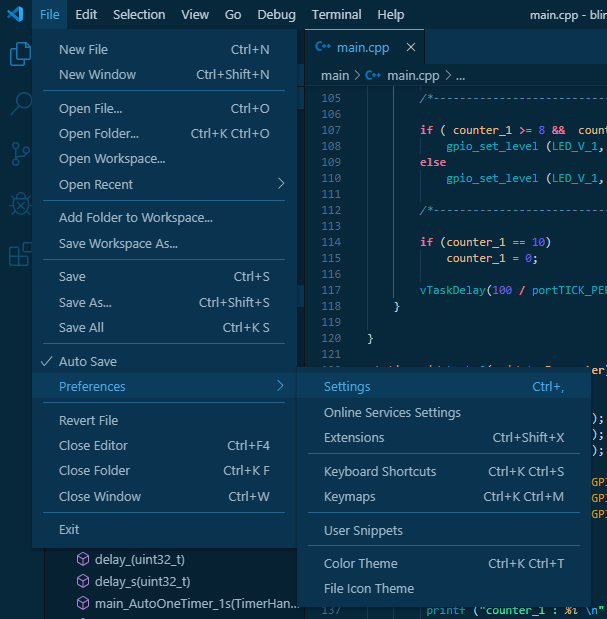
Trong cây thư mục project vừa add vào VS code :

* Create a folder name [.vscode]
  + Create a file name [c\_cpp\_properties.json]

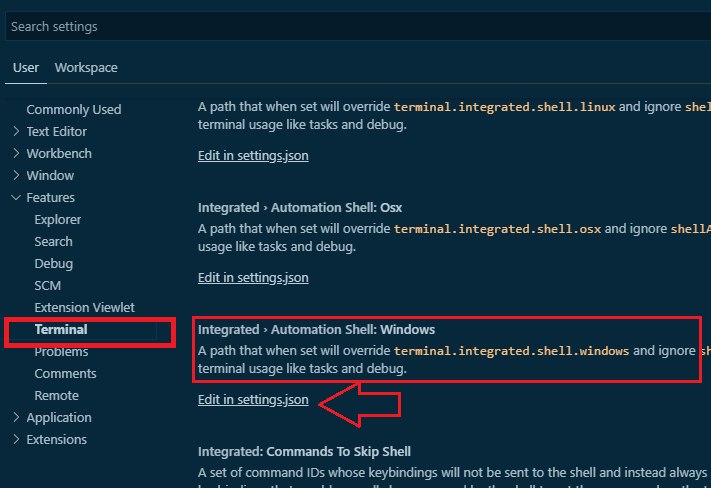
chú ý : file json này mình sẽ để đính kèm trong thư mục nén rar , hãy thay đổi đường dẫn trong “includePath” để cho tương thích với môi trường phát triển của riêng bạn .

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

Clike [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",

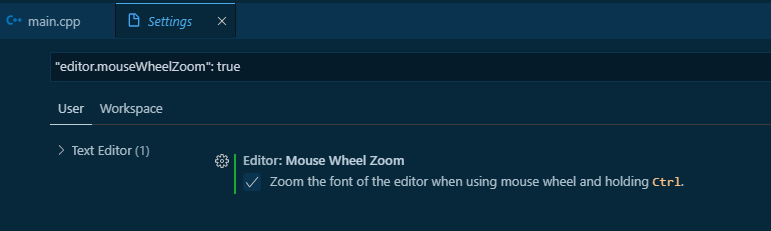
},

}

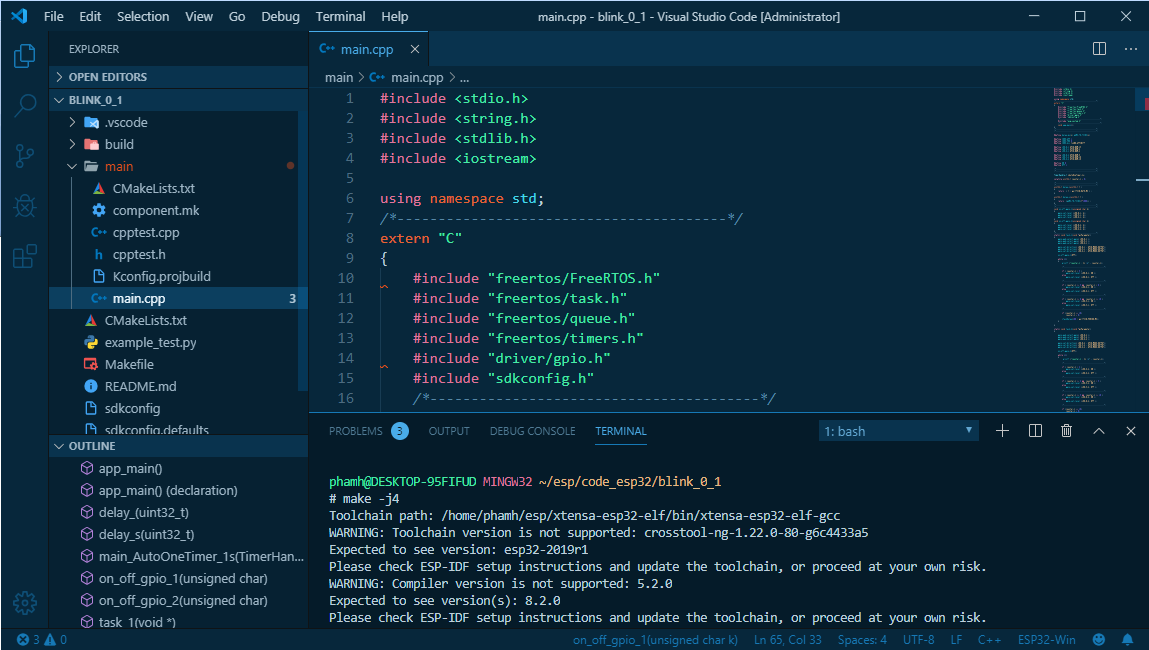
Khởi động lại VS code ! done !  
link tham khảo :   
<https://github.com/VirgiliaBeatrice/esp32-devenv-vscode/blob/master/tutorial.md>

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 số 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 trỏ đến thư mục đấy .

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 đâu 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>