

# BÀI THỰC HÀNH SỐ 01 HELLO WORLD CÀI ĐẶT CÔNG CỤ LẬP TRÌNH CHO ESP32

## 1 Yêu cầu

- Sinh viên sử dụng máy tính cá nhân để cài đặt
- Có kết nối mạng Internet
- Sinh viên có kiến thức về Hệ điều hành, đặc biệt là hiểu về biến môi trường
- Sinh viên có kiến thức về việc sử dụng command-line
- Sinh viên có kiến thức về lập trình nhúng

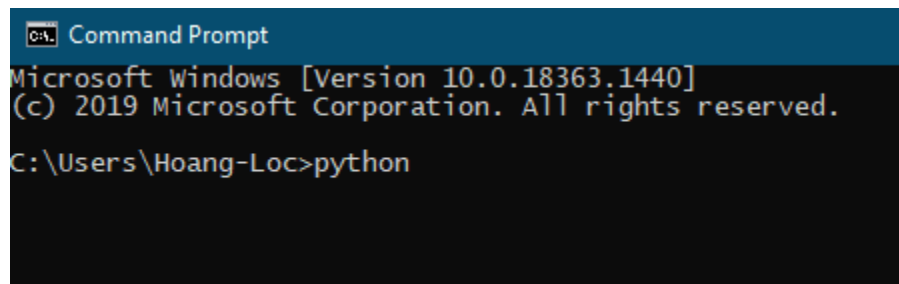
## 2 Cài đặt Công cụ lập trình

### 2.1 Cài đặt các phần mềm hỗ trợ

#### 2.1.1 Python

#### **Cách 1:**

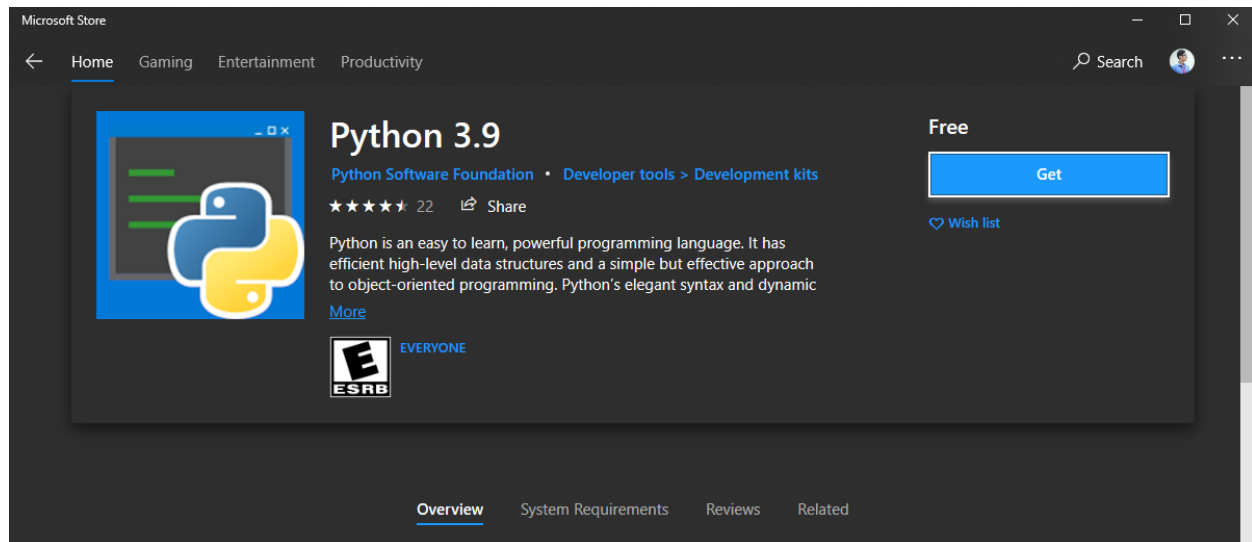
Sinh viên mở Command Prompt trên hệ điều hành Windows và gõ `python`



```
Command Prompt
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.
C:\Users\Hoang-Loc>python
```

Hệ thống sẽ tự động mở **Microsoft Store** và tìm đến `Python 3.9`

## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY



Sinh viên bấm **GET** để cài đặt

### Cách 2:

Sinh viên truy cập trang web <https://www.python.org/downloads/windows/> để tải **Python 3.9**. Lưu ý, cần phải chọn đúng file cài đặt tương ứng với hệ điều hành.

### Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		8cf053206beeca72c7ee531817dc24c7	25399571	<a href="#">SIG</a>
<a href="#">XZ compressed source tarball</a>	Source release		f0dc9000312abeb16de4eccce9a870ab	18889164	<a href="#">SIG</a>
<a href="#">macOS 64-bit Intel installer</a>	Mac OS X	for macOS 10.9 and later	a64f8b297fa43be07a3b8af9d13d554	29845662	<a href="#">SIG</a>
<a href="#">macOS 64-bit universal2 installer</a>	Mac OS X	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	fc8d028618c376d0444916950c73e263	37618901	<a href="#">SIG</a>
<a href="#">Windows embeddable package (32-bit)</a>	Windows		cde7d9bfd87b7777d7f0ba4b0cd4506d	7578904	<a href="#">SIG</a>
<a href="#">Windows embeddable package (64-bit)</a>	Windows		bd4903eb930cf1747be01e6b8dcd28a	8408823	<a href="#">SIG</a>
<a href="#">Windows help file</a>	Windows		e2308d543374e671ffe0344d3fd36062	8844275	<a href="#">SIG</a>
<a href="#">Windows installer (32-bit)</a>	Windows		81294c31bd7e2d4470658721b2887ed5	27202848	<a href="#">SIG</a>
<a href="#">Windows installer (64-bit)</a>	Windows	Recommended	efb20aa1b648a2baddd949c142d6eb06	28287512	<a href="#">SIG</a>

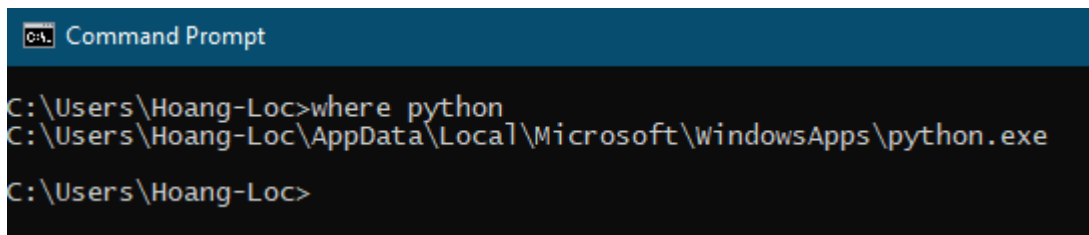
Để kiểm tra cài đặt thành công, sinh viên mở CMD và gõ **python**, nếu CMD xuất hiện dấu nhắc code như hình bên dưới có nghĩa là đã cài đặt thành công.

```
Command Prompt - python
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Hoang-Loc>python
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more
>>>
```

Để thoát khỏi chế độ code của python, gõ **exit()**

Để kiểm tra vị trí mà python được cài đặt, gõ `where python`



```
Command Prompt
C:\Users\Hoang-Loc>where python
C:\Users\Hoang-Loc\AppData\Local\Microsoft\WindowsApps\python.exe
C:\Users\Hoang-Loc>
```

### 2.1.2 Cài đặt GIT

Sinh viên truy cập <https://git-scm.com/downloads> để tải và cài đặt GIT.

Lưu ý: cần phải thêm đường dẫn đến folder `bin` của GIT vào biến môi trường (xem phần [2.2.4](#) để biết chi tiết cách thêm đường dẫn vào biến môi trường).

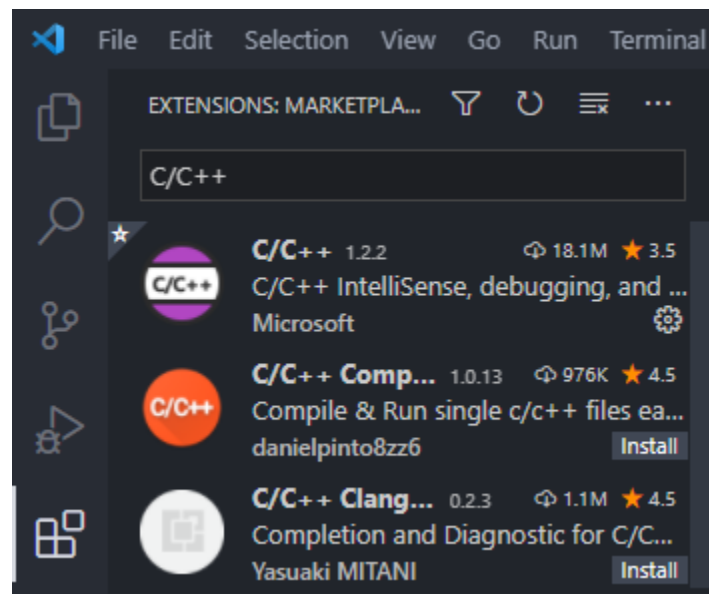
## 2.2 Lập trình C/C++ trên Visual Studio Code

### 2.2.1 Cài đặt Visual Studio Code

Sinh viên truy cập <https://code.visualstudio.com/download> để tải và cài đặt **Visual Studio Code**

### 2.2.2 Cài đặt extension C/C++ trong Visual Studio Code

Sinh viên mở phần mềm Visual Studio Code vừa cài đặt ở trên, chọn vào mục Extension và tìm `C/C++`



Sinh viên chọn extension `C/C++` và bấm **Install** để cài đặt. Sau khi cài đặt xong, sinh viên nhớ bấm nút **Enable** để kích hoạt extension.

### 2.2.3 Cài đặt Mingw-w64

**Mingw-w64** là trình biên dịch C/C++ để giúp Visual Studio Code có thể biên dịch các file sourcecode C/C++ thành các file khả thực thi (\*.exe đối với hệ điều hành Windows).

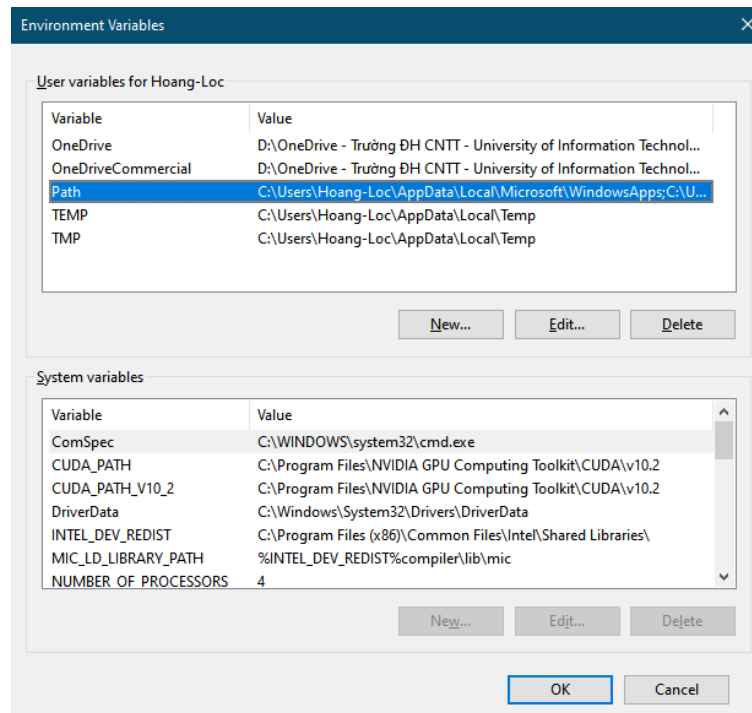
Sinh viên truy cập <https://winlibs.com/#download-release> để cài đặt.

Trong bước cài đặt, sinh viên chọn kiến trúc là **x86\_64**, để nơi cài đặt mặc định và **copy** địa chỉ nơi cài đặt. Bấm **Next** để bắt đầu cài đặt.

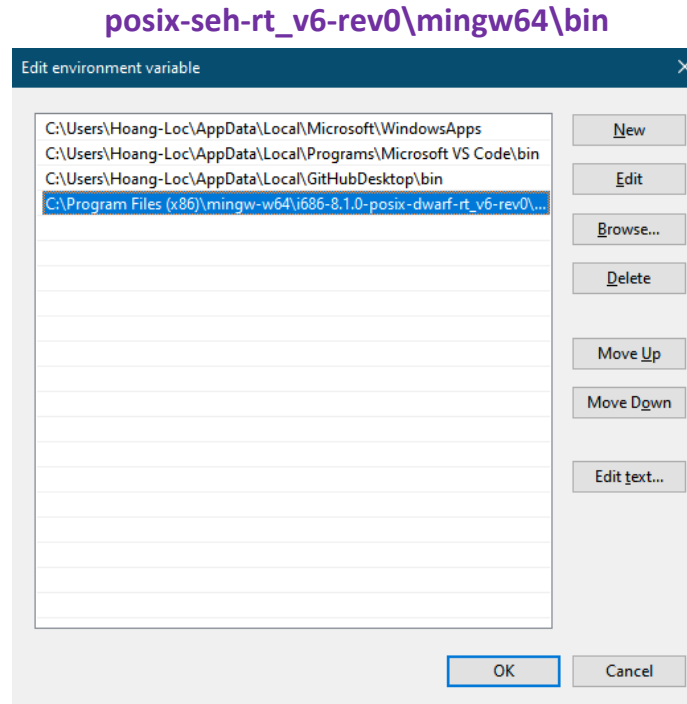
### 2.2.4 Thêm đường dẫn vào biến môi trường

Để có thể biên dịch được các file sourcecode, trình biên dịch cần được thêm vào biến môi trường (để có thể thực thi được thông qua CMD). Để thêm đường dẫn của Mingw-w64 vào biến môi trường, thực hiện như sau:

- Trong thanh tìm kiếm của hệ điều hành Windows, gõ **settings** để mở **Windows Settings**
- Tìm **Edit environment variables for your account**
  - Chọn biến môi trường **Path** và bấm **Edit** để chỉnh sửa



- Bấm **New** để thêm một đường dẫn mới vào biến môi trường và paste đường dẫn nơi cài đặt **Mingw-w64** mà sinh viên đã copy ở bước trước đó. Đường dẫn mặc định mà Mingw-w64 đề xuất là **C:\Program Files\mingw-w64\x86\_64-8.1.0-**



- Bấm **OK** để lưu lại đường dẫn vừa thêm.

Để kiểm tra xem Mingw-w64 đã được cài đặt và thêm thành công vào biến môi trường, sinh viên mở CMD và gõ `g++ --version`. Nếu CMD trả về phiên bản g++ đang sử dụng có nghĩa là trình biên dịch đã được cài đặt thành công.

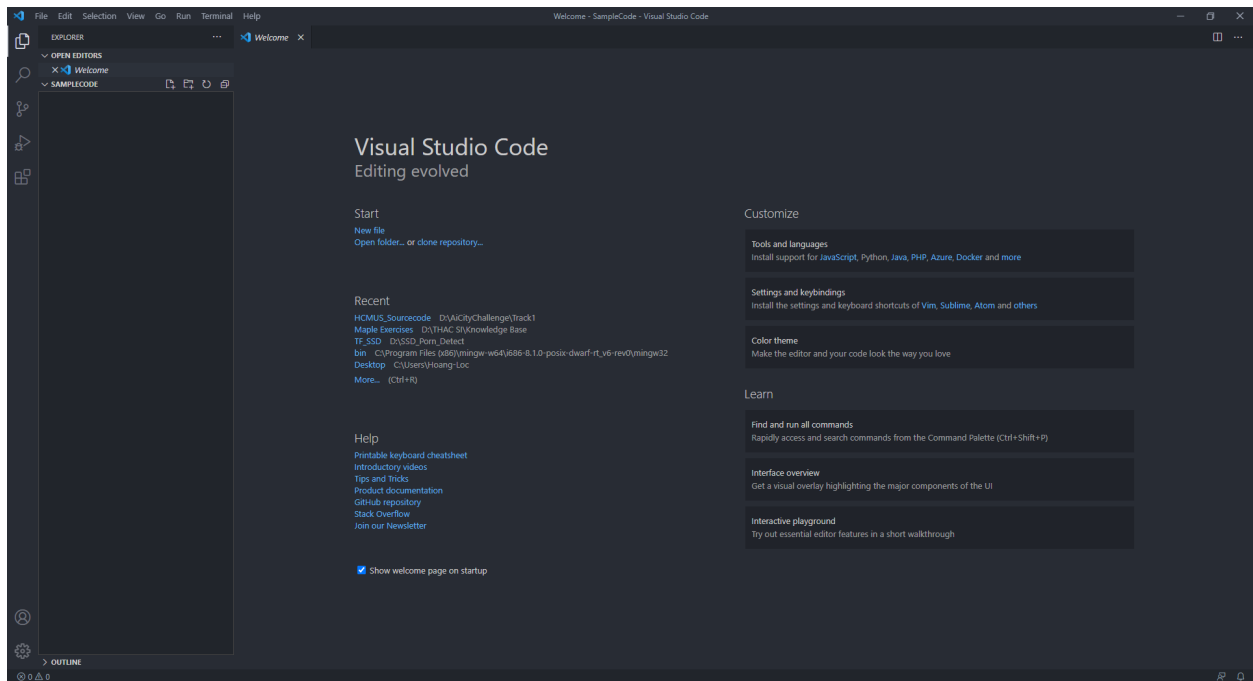
```
C:\Users\Hoang-Loc>g++ --version
g++ (i686-posix-dwarf-rev0, Built by MinGW-W64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```


Để kiểm tra nơi cài đặt của Mingw-w64, gõ `where g++` hoặc `where gcc`.

### 2.2.5 Build chương trình helloworld.c

Mở Visual Studio Code và bấm Open Folder, tạo một thư mục mới để chứa sourcecode.

# CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY



Bấm nút  để tạo một file mới, đặt tên là **helloworld.c**

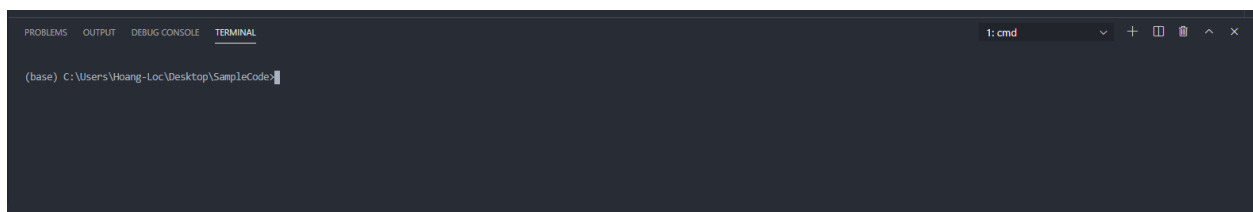
Gõ đoạn code hello world đơn giản và bấm Ctrl+S để Save. Ngoài ra, sinh viên cũng có thể chọn **File → Auto Save** để phần mềm tự động lưu tất cả những sửa đổi trong code.

```
C helloworld.c > main()
1  #include <stdio.h>
2  #include <conio.h>
3
4  void main()
5  {
6      printf("Hello world!");
7      getch();
8  }
```

Để build chương trình trên, ta có 2 cách sau:


## Cách 1 (phù hợp với những sourcecode đơn giản):

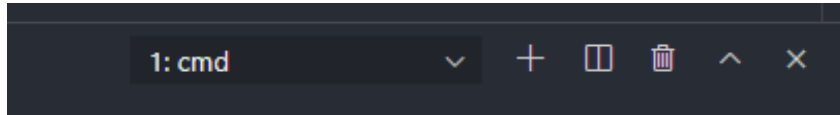
Từ menu chọn **Terminal → New Terminal**. VS Code sẽ mở một terminal như hình bên dưới



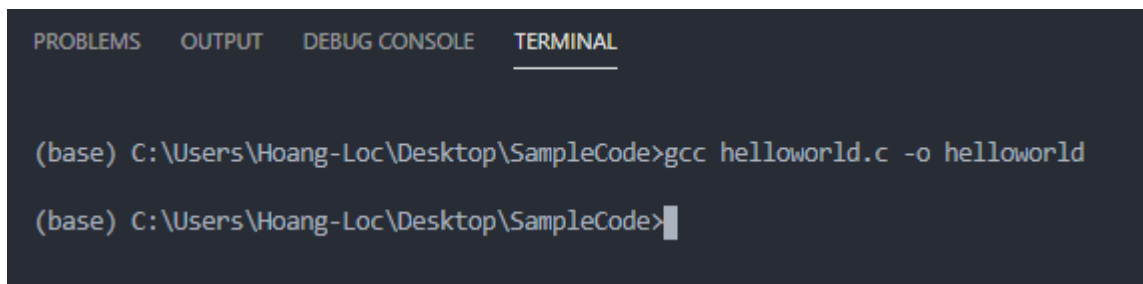
## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY

Lưu ý: mặc định, VSCode trên Windows sẽ mở terminal powershell thay vì cmd. Để chọn cmd là terminal mặc định, sinh viên thực hiện như sau:

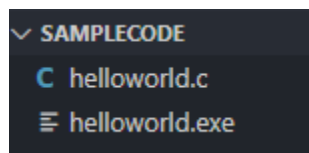
- Bấm **Ctrl + Shift + P**, gõ `shell`
- Chọn `Terminal: Select Default Shell`
- Chọn `Command Prompt`
- Tắt terminal hiện tại bằng cách bấm vào nút 
- Mở lại terminal mới, lúc này, terminal sẽ hiện là đang sử dụng cmd



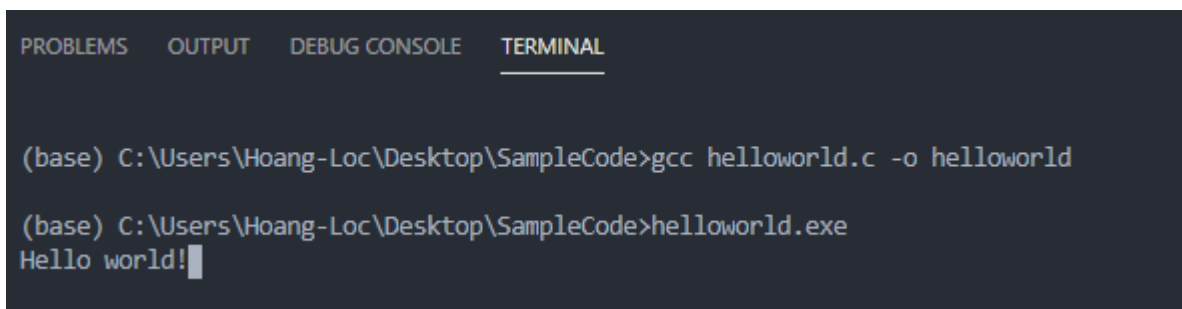
Từ terminal, gõ `gcc helloworld.c -o helloworld`



Lúc này, trình biên dịch Mingw-w64 đã cài đặt ở trên sẽ biên dịch sourcecode helloworld.c thành file khả thực thi helloworld.exe

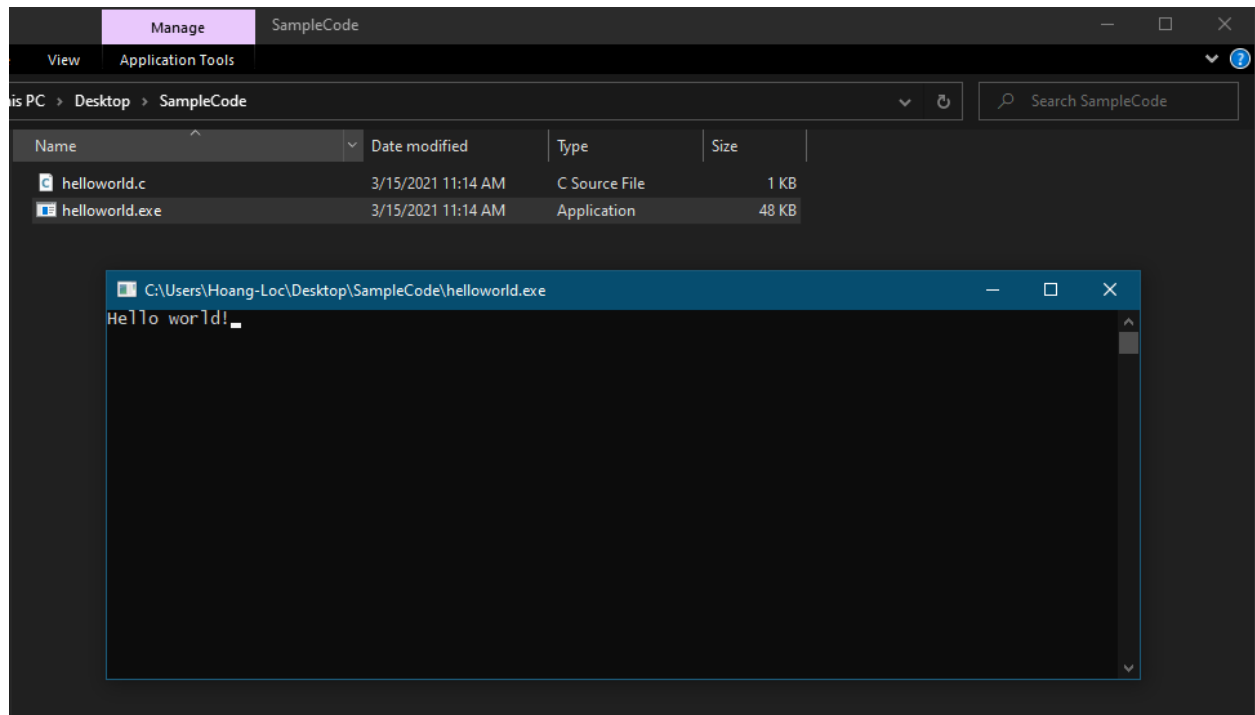


Cuối cùng, ta có thể thực thi file này bằng cách gõ `helloworld.exe` trong terminal



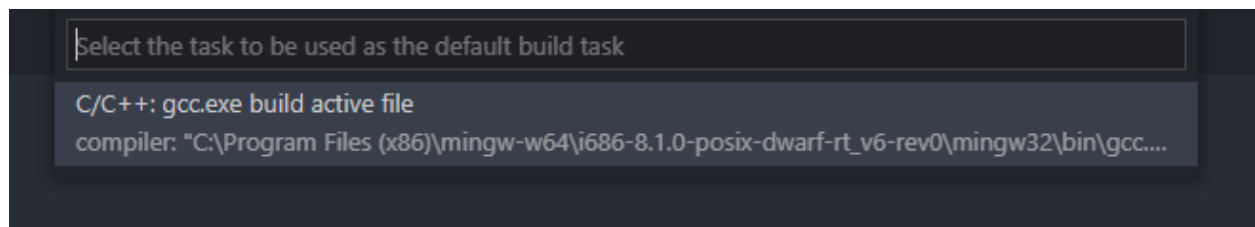
hoặc double-click vào file helloworld.exe trong Windows Explorer.

## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY



### Cách 2:

Tạo file `task.json` để định nghĩa những tác vụ cần thiết để build sourcecode bằng cách vào menu **Terminal → Configure Default Build Task → C/C++: gcc.exe build active file**



Sau khi chọn như trên, VSCode sẽ tự động tạo ra file ***tasks.json*** nằm trong thư mục ***.vscode*** với các thông tin mặc định như hình dưới





Trong đó:

- Cài đặt `"command"` định nghĩa lệnh sẽ sử dụng, cụ thể ở đây là `gcc`
- Cài đặt `"args"` định nghĩa các option, cụ thể ở đây là: `-g ${file} -o {fileDirname}\\${fileBasenameNoExtension}.exe`

Ghép 2 cài đặt trên lại, ta sẽ được một lệnh hoàn chỉnh trong CMD như sau:

```
gcc -g ${file} -o {fileDirname}\\${fileBasenameNoExtension}.exe
```

Ý nghĩa của đoạn lệnh trên là sử dụng chương trình gcc biên dịch file sourcecode là `${file}` (ở đây là file helloworld.c) thành file khả thực thi `${...}.exe` trong thư mục `${fileDirname}`. Tóm lại, về bản chất cách này và cách 1 ở trên là như nhau. Tuy nhiên, cách 2 có thể được áp dụng tốt trong các trường hợp project phức tạp, nhiều file sourcecode và phải dùng nhiều lệnh, nhiều thao tác để build. Như vậy, nếu như làm thủ công như cách 1 sẽ mất nhiều thời gian hơn.

Sau khi đã định nghĩa xong tasks.json, ta thực hiện build sourcecode bằng cách chọn menu **Terminal → Run Build Task**. Lúc này, VSCode sẽ dựa trên các tác vụ được định nghĩa trong file *tasks.json* để build. Kết quả tương tự như cách 1, ta cũng sẽ có được file helloworld.exe.

Sinh viên có thể tham khảo thêm và cách sử dụng VSCode để lập trình C/C++ tại <https://code.visualstudio.com/docs/cpp/config-mingw>

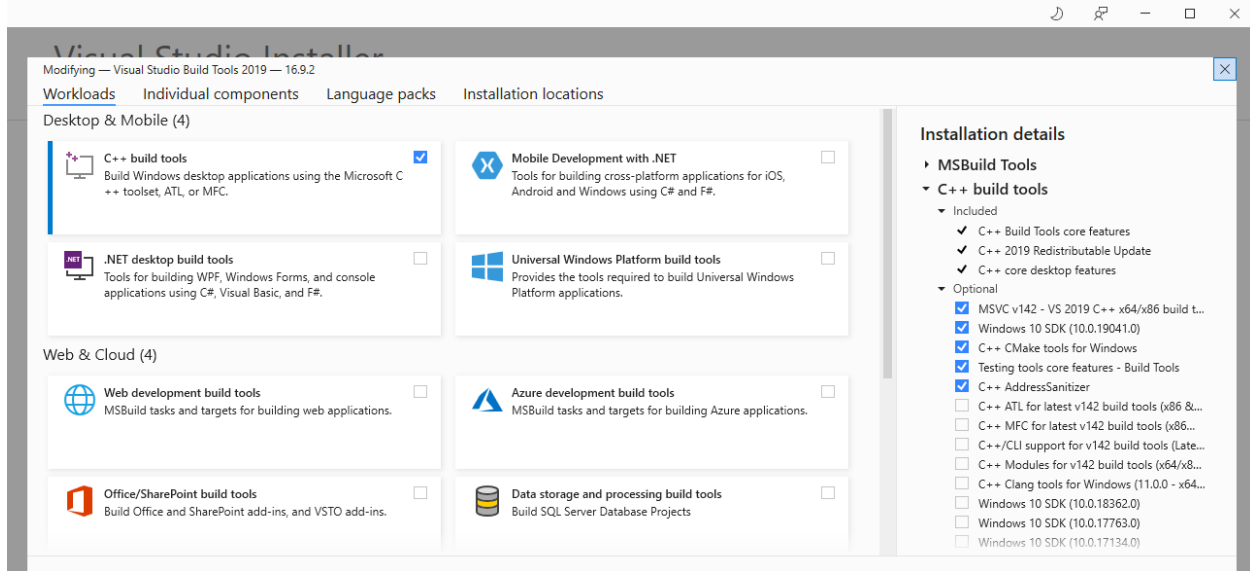
### 2.3 Cài đặt Espressif IDF

Để có thể lập trình nhúng trên ESP32, ta cần cài đặt gói thư viện **Espressif IDF**. Để cài đặt gói thư viện này, ta có thể cài đặt thủ công bằng cách tự tải thư viện về, lập trình và sử dụng thông qua CMD. Tuy nhiên, để mọi thứ trở nên đơn giản và thuận tiện hơn trong việc lập trình, debug, cũng như nạp code lên board, Espressif IDF đã hỗ trợ extension trên VSCode.

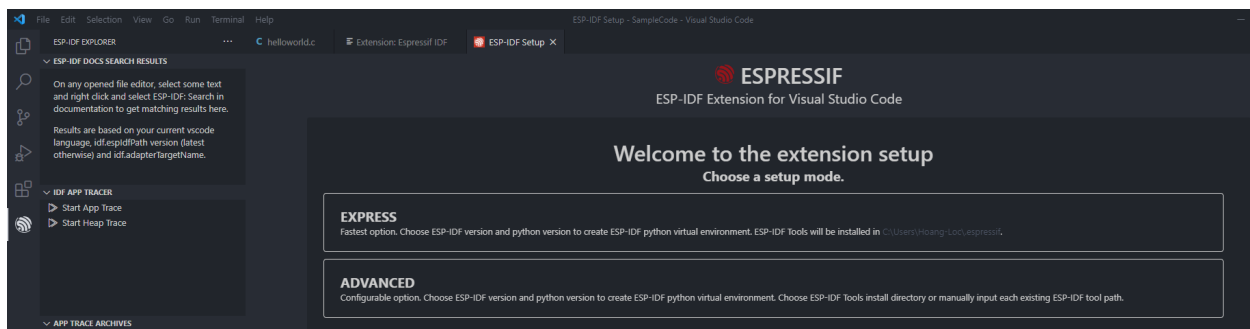
Để cài đặt extension trên, ta vào phần extension của VSCode và tìm **Espressif IDF** và bấm **Install**.

Để có thể thực hiện cấu hình được extension trên, sinh viên trước tiên cần cài đặt **Microsoft C++ Build Tools** tại <https://visualstudio.microsoft.com/visual-cpp-build-tools/>. Thực hiện lựa chọn như hình bên dưới và bấm cài đặt.

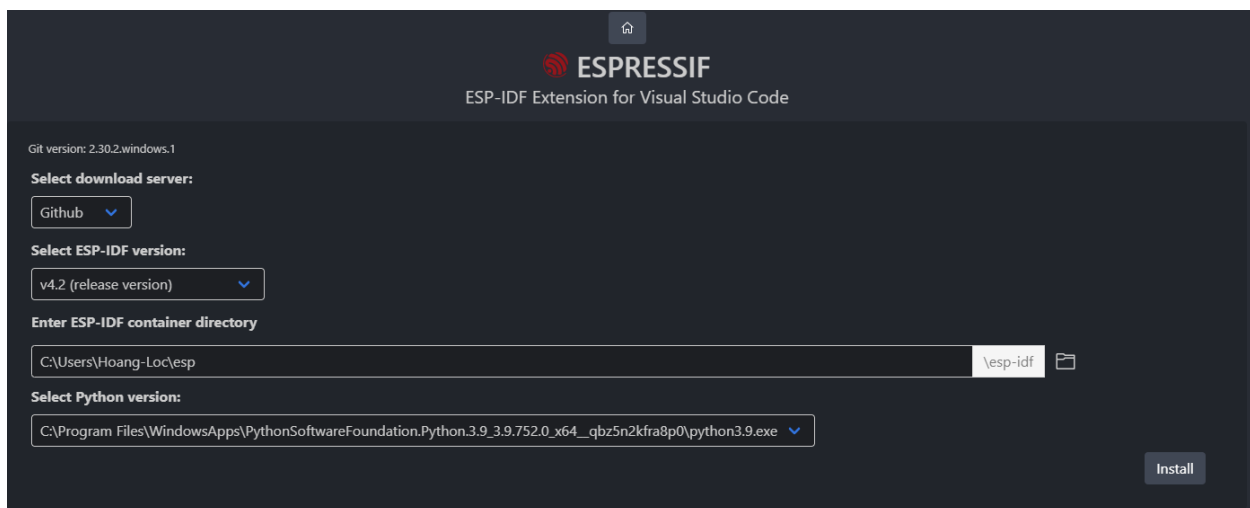
## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY



Sau khi cài đặt hoàn tất, chương trình sẽ yêu cầu khởi động lại máy tính. Thực hiện khởi động lại và mở lại VSCode. Sinh viên bấm vào biểu tượng của Espressif IDF để cấu hình cho extension Espressif IDF.

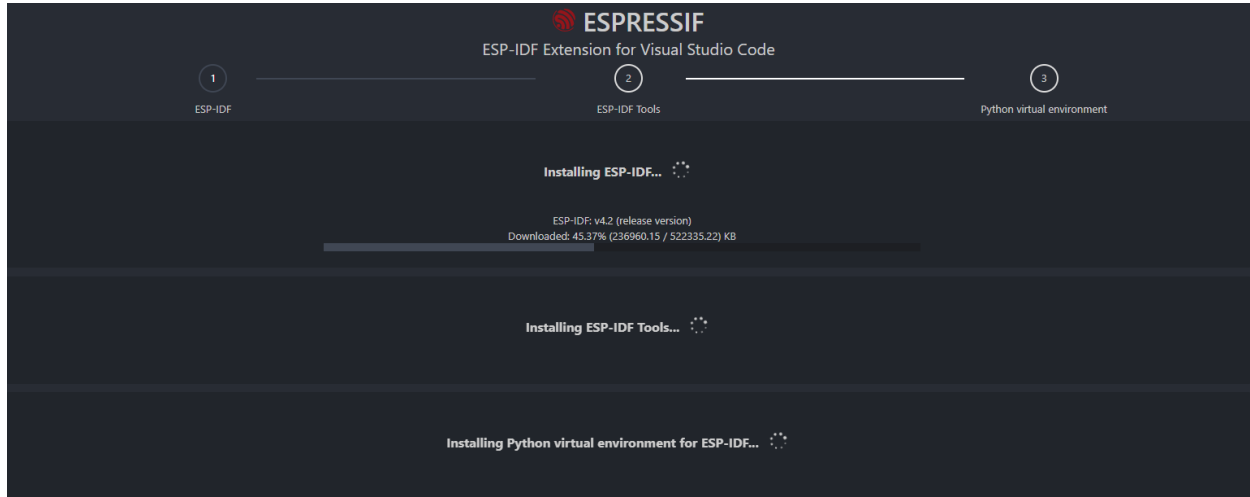


Chọn **EXPRESS** và lựa chọn các option như hình dưới. Phần mềm sẽ tự động tìm kiếm trong hệ thống Python đã được cài đặt trước đó.

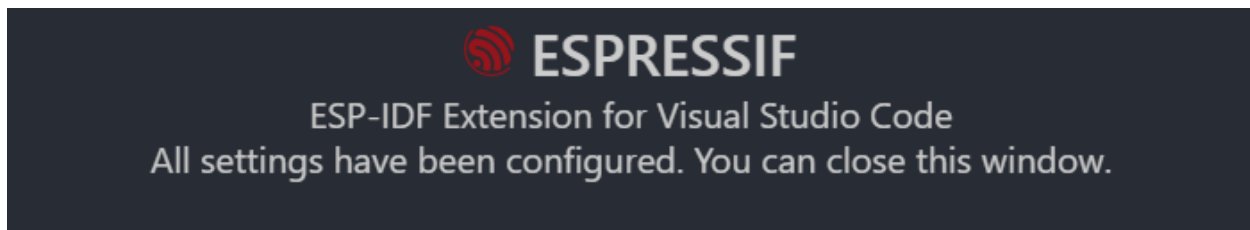


## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY

Bấm **Install** để bắt đầu cài đặt. Phần mềm sẽ tự động tải về gói thư viện Espressif IDF và các phần mềm hỗ trợ liên quan.



Sau khi cài đặt xong, extension sẽ xuất hiện màn hình như bên dưới



Nếu khi bấm **Install** mà chương trình **báo lỗi** không tải được esp-idf, sinh viên có thể thực hiện cách thay thế như sau:

- Tải gói cài đặt esp-idf dành cho Windows tại:  
<https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/windows-setup.html#get-started-windows-tools-installer>
- Chạy file cài đặt ESP-IDF Tools
- Trong quá trình cài đặt, trình cài đặt sẽ đề xuất một số cách fix một vài vấn đề mà nó phát hiện trong hệ thống. Ta có thể bấm **Apply Fixes** để áp dụng.
- Hoàn tất các bước cài đặt còn lại

Sau đó, mở VS Code, chọn extension Espressif IDF, trên giao diện cấu hình extension, chọn **ADVANCED** và cài đặt như bên dưới.

## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY



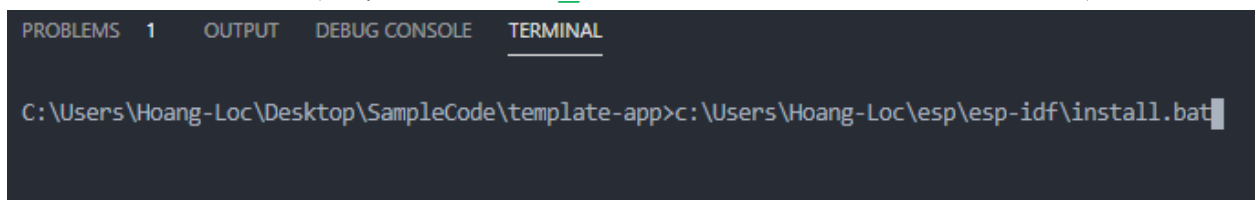
Trong đó:

- **Select ESP-IDF version:** chọn Find ESP-IDF in your system
- **ESP-IDF directory (IDF\_PATH):** trỏ tới thư mục vừa cài đặt *esp-idf*
- **Enter ESP-IDF Tools directory (IDF\_TOOLS\_PATH):** trỏ tới thư mục *.espressif* được cài đặt trong bước trước đó.

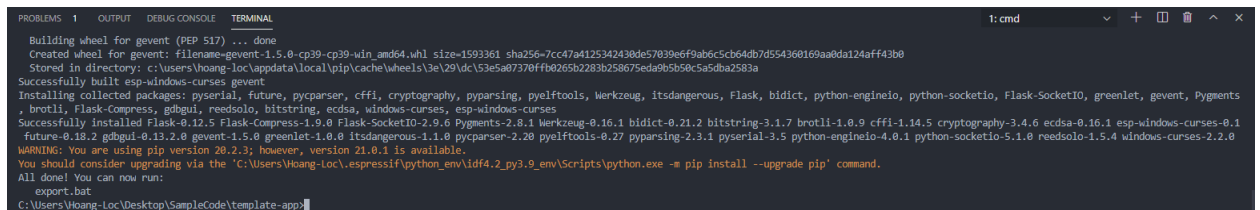
Cuối cùng, bấm **Install** và để VS Code cấu hình phần còn lại.

Tiếp theo, để có thể build được project, ta cần phải cài đặt các gói thư viện hỗ trợ bằng cách sau:

- Mở menu **Terminal** → **New Terminal**
- Thực thi file `install.bat` trong thư mục `%INSTALL_DIR%\esp\esp-idf` bằng cách nhập đường dẫn của file này vào Terminal hoặc kéo thả file này vào terminal của VSCode (thay `%INSTALL_DIR%` là thư mục chứa thư mục `esp`)



- Bấm **Enter** để thực hiện việc cài đặt. Kết quả cài đặt thành công tất cả thư viện như hình bên dưới

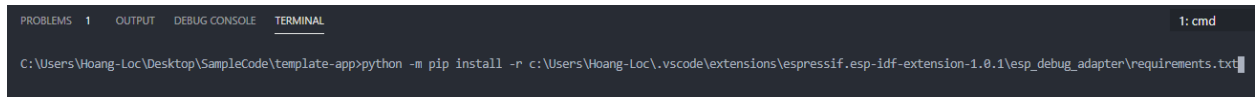


## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY

- Điều chỉnh biến môi trường bằng cách thực thi `export.bat` trong thư mục `%INSTALL_DIR%\esp\esp-idf`
- Tiếp theo, sinh viên thực hiện cài đặt các gói thư viện hỗ trợ còn lại bằng cách gõ lệnh

```
python -m pip install -r %USER_DIR%\vscode\extensions\espressif.esp-idf-extension-1.0.1\esp_debug_adapter\requirements.txt
```

(thay `%USER_DIR%` là thư mục người dùng có chứa thư mục `.vscode`)



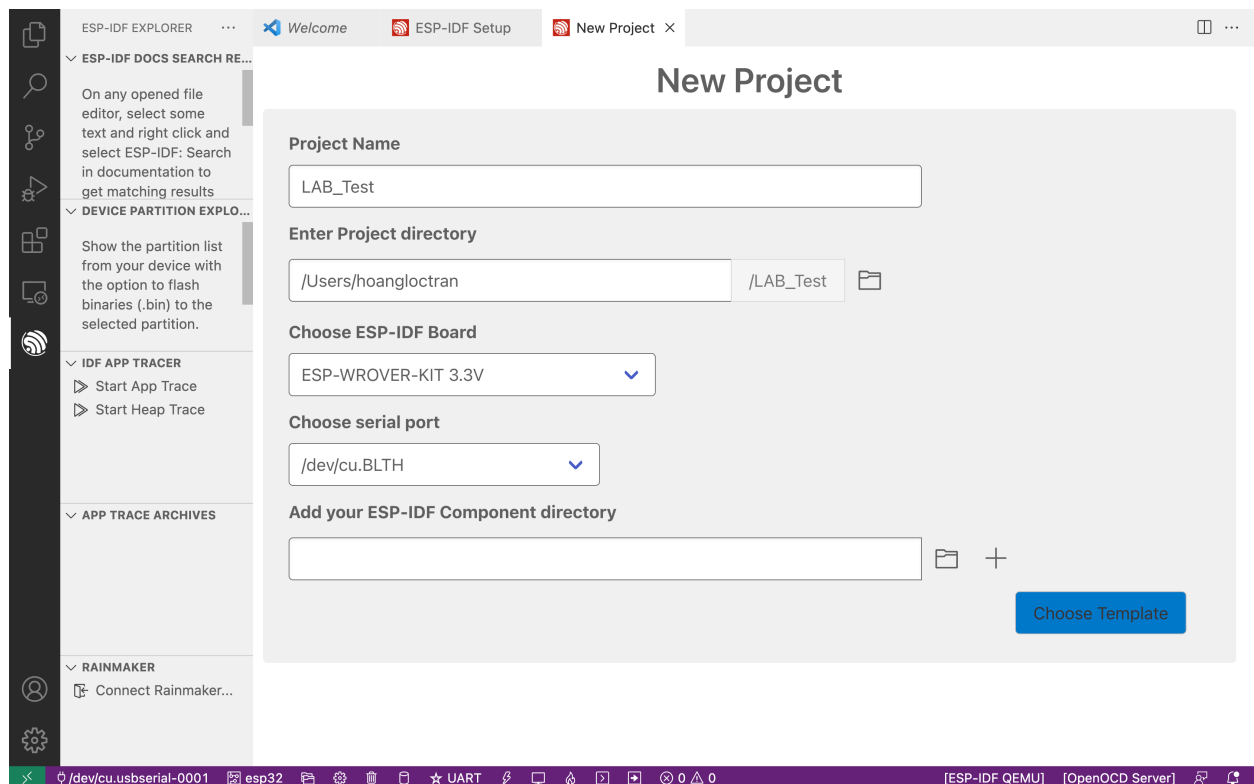
```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL 1: cmd
C:\Users\Hoang-Loc\Desktop\SampleCode\template-app>python -m pip install -r c:\Users\Hoang-Loc\.vscode\extensions\espressif.esp-idf-extension-1.0.1\esp_debug_adapter\requirements.txt
```

### 3 Sử dụng VSCode và ESP-IDF để lập trình cho ESP32

#### 3.1 Tạo project

Để tạo một project mới cho ESP32, ta thực hiện như sau:

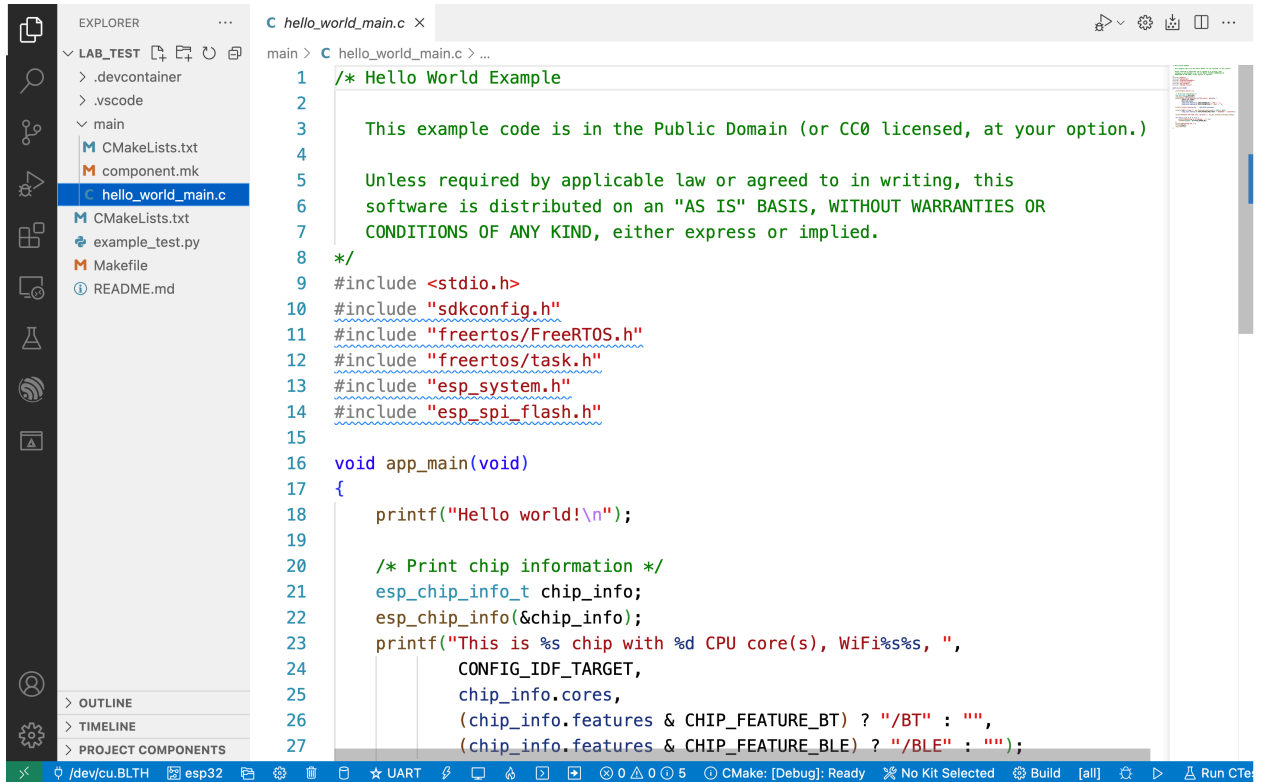
- Trong VSCode, bấm F1 và gõ `ESP-IDF: New Project`
- Điền thông tin cho project
- Bấm `Choose Template`



- Trong giao diện chọn template vừa được mở, bấm vào nút drop-down đổi `Extension` thành `ESP-IDF`.
- Trong mục `get-started`, chọn ví dụ `hello_world`.

## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY

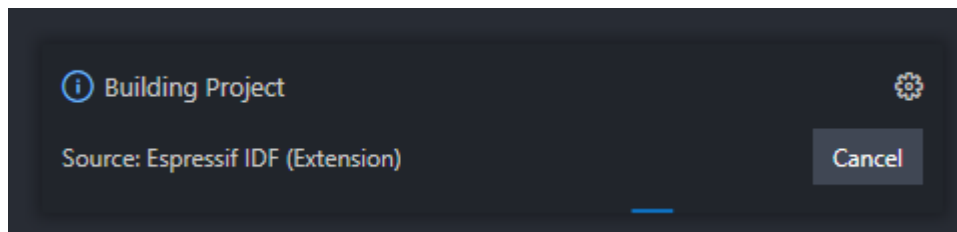
- Các thông tin về project sẽ được hiển thị trong phần bên phải của cửa sổ.
- Đọc kỹ và bấm **Create project using template hello\_world**.
- Bấm chọn mở project trong cửa sổ mới. Một project mới sẽ được tạo với chức năng được mô tả trước đó.



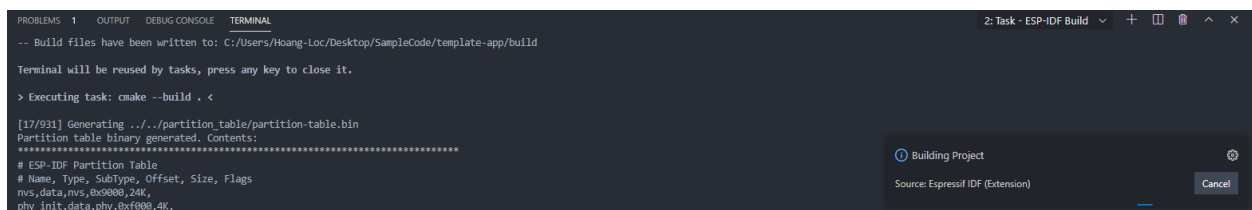
```
1 /* Hello World Example
2
3 This example code is in the Public Domain (or CC0 licensed, at your option.)
4
5 Unless required by applicable law or agreed to in writing, this
6 software is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR
7 CONDITIONS OF ANY KIND, either express or implied.
8 */
9 #include <stdio.h>
10 #include "sdkconfig.h"
11 #include "freertos/FreeRTOS.h"
12 #include "freertos/task.h"
13 #include "esp_system.h"
14 #include "esp_spi_flash.h"
15
16 void app_main(void)
17 {
18     printf("Hello world!\n");
19
20     /* Print chip information */
21     esp_chip_info_t chip_info;
22     esp_chip_info(&chip_info);
23     printf("This is %s chip with %d CPU core(s), WiFi%s%s, ",
24           CONFIG_IDF_TARGET,
25           chip_info.cores,
26           (chip_info.features & CHIP_FEATURE_BT) ? "/BT" : "",
27           (chip_info.features & CHIP_FEATURE_BLE) ? "/BLE" : "");
```

### 3.2 Build project

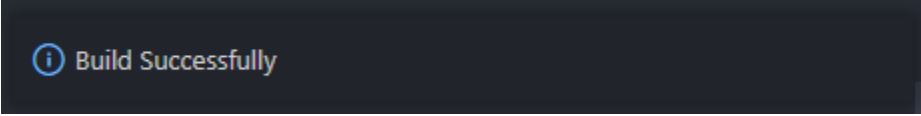
Để build project, sinh viên bấm F1 và gõ **ESP-IDF: Build Project** hoặc bấm  trên thanh công cụ ở dưới cùng của VSCode.



Quá trình lần đầu tiên build project sẽ như hình bên dưới




Kết quả build project thành công được VSCode thông báo qua popup



A dark-themed notification box with a green circular icon containing a white 'i' on the left, followed by the text 'Build Successfully' in a light green font.

### 3.3 Sử dụng KIT ESP32


#### 3.3.1 Flash project

Trước khi nhúng code vừa build lên ESP32, ta cần kết nối ESP32 với máy tính và xác định cổng COM kết nối. Cách xác định cổng kết nối như sau:

- Bấm chuột phải vào biểu tượng **Start của Windows → Device Management**
- Nếu thấy mục Other devices xuất hiện với dòng **Silicon Labs CP210x USB to UART Bridge** kèm theo dấu  thì có nghĩa là driver chưa được tải. Sinh viên tải và cài đặt driver tại <https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers>
- Sau khi cài đặt driver hoàn tất, khởi động lại máy tính và kiểm tra lại Device Management và xác định cổng COM được kết nối tới ESP32

Sau khi đã xác định được cổng COM, sinh viên thực hiện cổng COM trong VSCode bằng cách bấm nút  trên thanh công cụ và chọn cổng COM tương ứng. Một cách khác để cấu hình cổng COM là sinh viên có thể bấm vào biểu tượng  và chọn **Settings**, chọn **Extensions → ESP-IDF** và cài đặt cổng COM kết nối với ESP32 trong **Port Win** (nếu sử dụng hệ điều hành Windows)

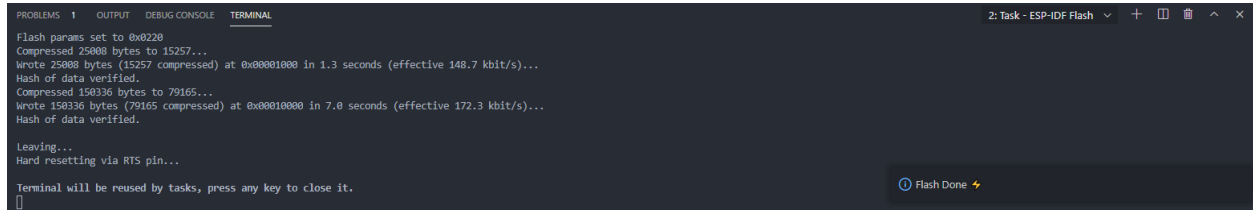


Sau khi đã xác định cổng COM kết nối giữa máy tính và ESP32, ta thực hiện nạp code bằng cách bấm F1 và gõ **ESP-IDF: Flash (UART) your project** hoặc bấm nút  trên thanh công cụ bên dưới phần mềm VSCode, sinh viên chọn phương thức kết nối là UART nếu được hỏi. Để bắt đầu quá trình nạp code, khi termina hiện

## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY

Connecting...\_\_\_\_\_ sinh viên cần bấm giữ nút **Boot** trên ESP32 đến khi code bắt đầu được nạp thì thả ra.

Kết quả khi nạp thành công được thể hiện trong hình bên dưới.




```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
Flash params set to 0x0220
Compressed 25088 bytes to 15257...
Wrote 25088 bytes (15257 compressed) at 0x00010000 in 1.3 seconds (effective 148.7 kbit/s)...
Hash of data verified.
Compressed 150336 bytes to 79165...
Wrote 150336 bytes (79165 compressed) at 0x00010000 in 7.0 seconds (effective 172.3 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

Terminal will be reused by tasks, press any key to close it.
Flash Done ⚡
```

### 3.3.2 Monitor project

Để xem kết quả chạy của ESP32, sinh viên bấm F1 và gõ **ESP-IDF: Monitor your device** hoặc bấm nút  trên thanh công cụ bên dưới phần mềm VSCode.



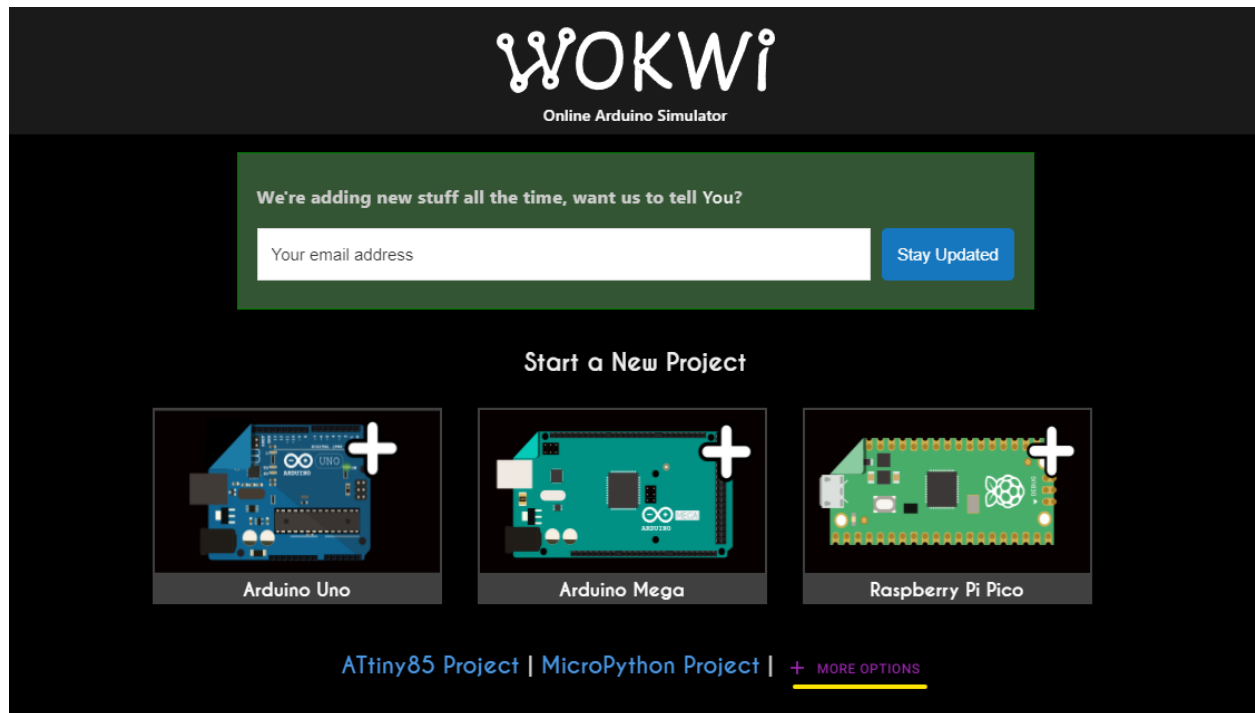
```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL
2: ESP-IDF Monitor
I (264) heap_init: At 3FFC4350 len 00010CB0 (111 KiB): D/IRAM
I (270) heap_init: At 40080EE4 len 0001011C (88 KiB): IRAM
I (277) cpu_start: Pro cpu start user code
I (295) spi_flash: detected chip: generic
I (295) spi_flash: flash io: dio
I (296) 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, 4MB external flash
Restarting in 10 seconds...
Restarting in 9 seconds...
Restarting in 8 seconds...
```

### 3.4 Sử dụng công cụ mô phỏng ESP32 trên Wokwi

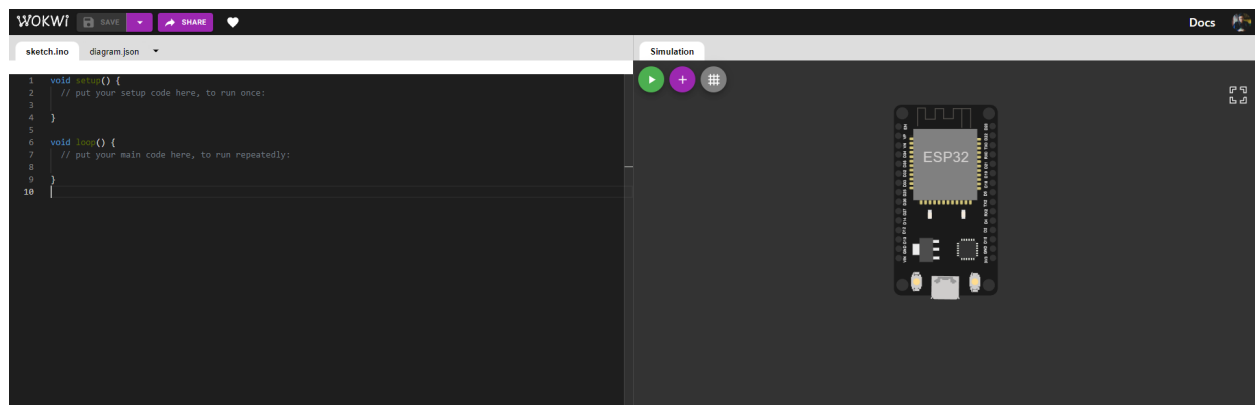
Trong trường hợp sinh viên chưa có kit ESP32, sinh viên có thể sử dụng công cụ mô phỏng ESP32 trên [Wokwi](https://wokwi.com/). Wokwi là nền tảng mô phỏng các thiết bị Arduino hoàn toàn miễn phí, trong đó có ESP32. Wokwi không chỉ hỗ trợ viết code Arduino mà còn hỗ trợ cho phép nạp file bin sau khi đã được build lên kit mô phỏng.

Trước hết, truy cập Wokwi và tạo một tài khoản: <https://wokwi.com/>





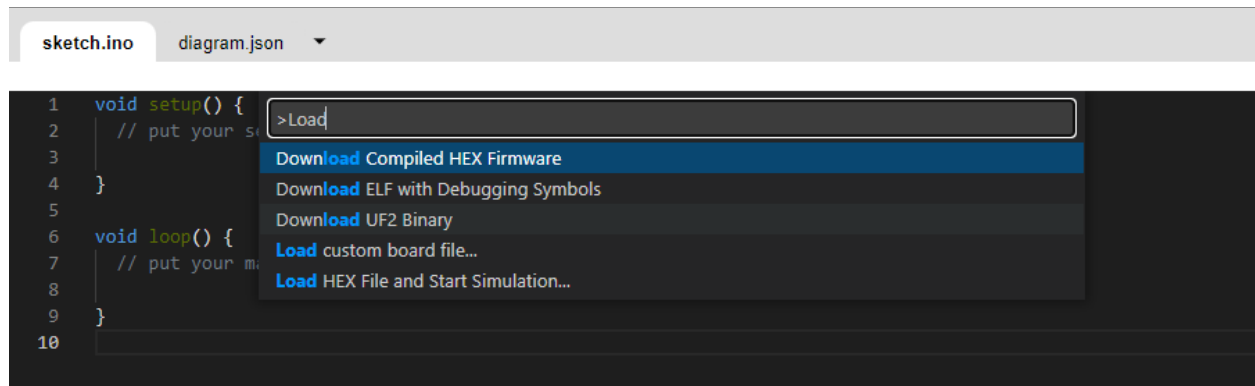
Trong phần **MORE OPTIONS**, chọn ESP32, khi đó, Wokwi sẽ chuyển sang cửa sổ code và mô phỏng.



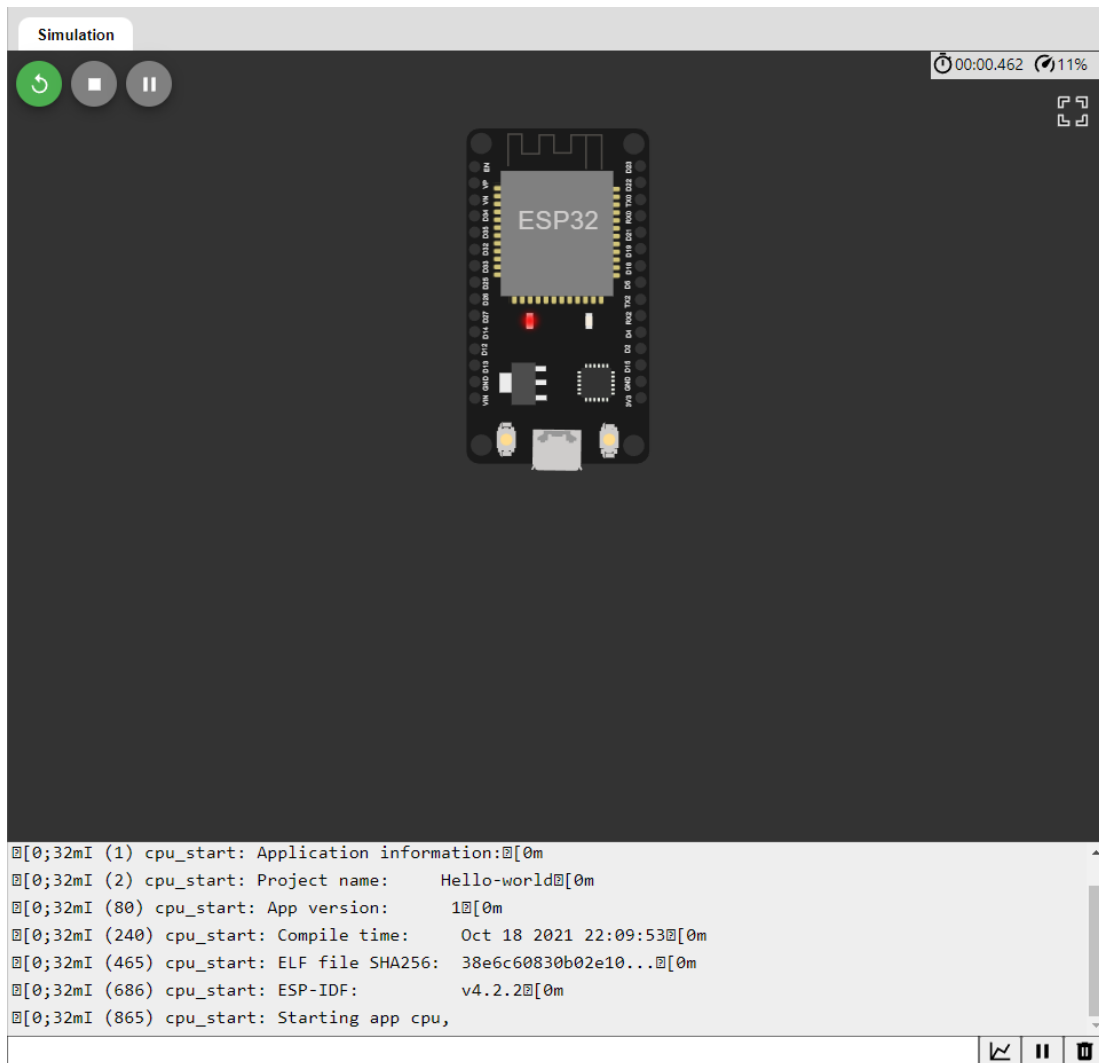
Trong cửa sổ này, chúng ta thấy bên trái là khu vực code, và bên phải là khu vực mô phỏng. Sinh viên có thể thêm các thiết bị điện tử như LED, LCD, Keypad, Servo,... để kết nối với kit.

Tiếp theo, chúng ta sẽ tải file bin đã được Build thành công trong mục 3.2 lên kit mô phỏng. Bấm phím **F1**, gõ **Load HEX file and Start Simulation**.

## CE232 - THIẾT KẾ HỆ THỐNG NHÚNG KHÔNG DÂY



Sau đó, chọn file **\*.bin** đã được build thành công ở trên và upload.



Cuối cùng, ta có kết quả mô phỏng như hình trên.

--- HẾT ---