

## Tìm hiểu về hệ sinh thái STM32Cube

Hệ sinh thái STM32Cube là một giải pháp phần mềm hoàn chỉnh cho các dòng vi điều khiển STM32 và các dòng vi xử lý mà nó sử dụng. Hệ sinh thái này dành cho những người dùng đang tìm kiếm một môi trường phát triển đầy đủ và miễn phí cho STM32, cũng như dành cho những người đang sử dụng một số IDE khác như KeilC hoặc IAR. Trong hệ sinh thái này có sự tích hợp của các thành phần gồm STM32CubeMX, STMCubeProgrammer và STM32CubeMonitor.

### Tóm lại, STM32Cube là gì?

STM32Cube là sự kết hợp của các công cụ phần mềm và các thư viện phần mềm nhúng:

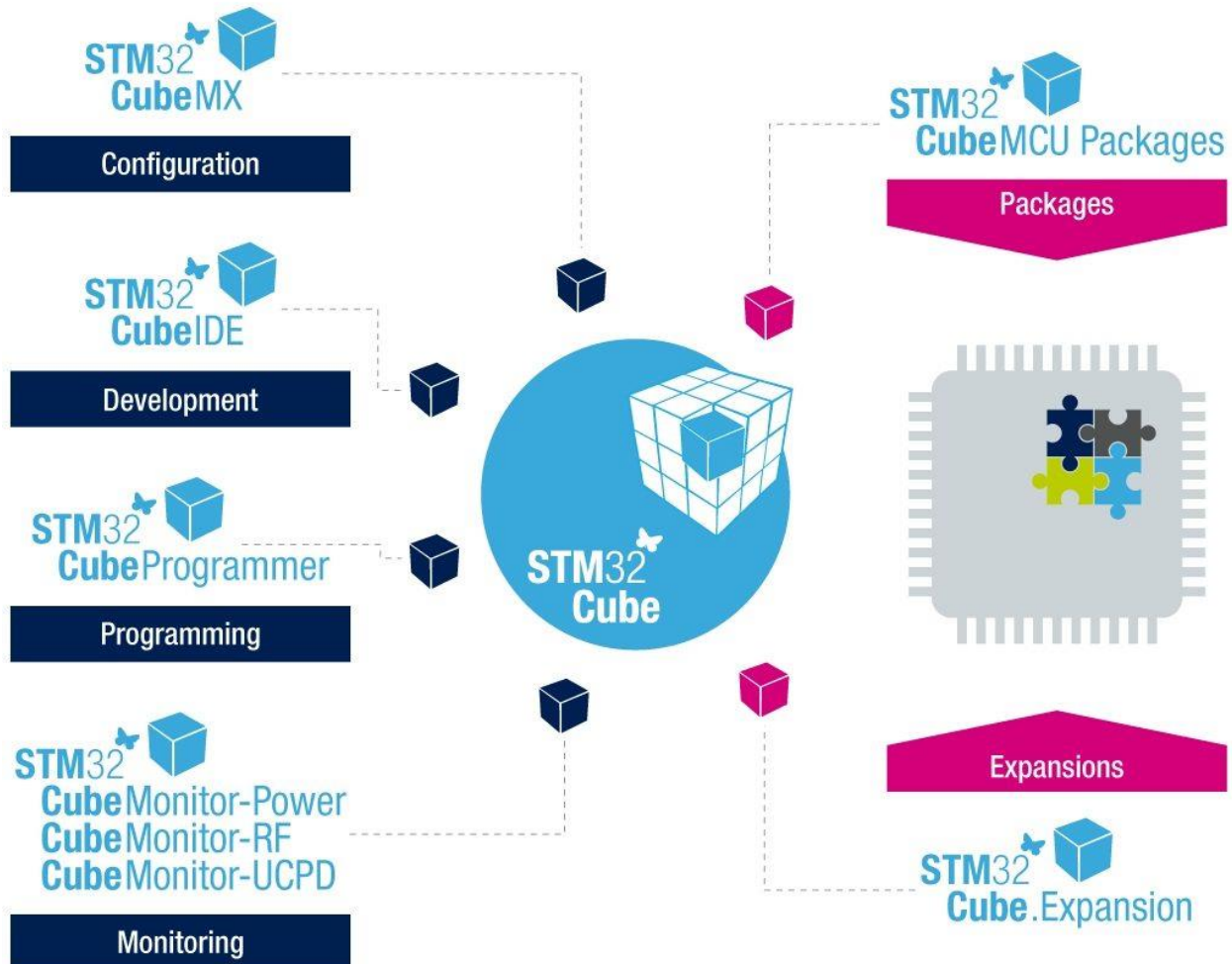
- Đầy đủ các công cụ phần mềm hỗ trợ chạy trên máy tính giúp giải quyết tất cả những nhu cầu trong một chu trình phát triển dự án hoàn chỉnh.
- Các phần mềm nhúng được thiết kế để chạy trên các dòng vi điều khiển STM32 và các vi xử lý tương ứng với nhiều chức năng khác nhau từ các driver cho từng ngoại vi của vi điều khiển đến những tính năng định hướng ứng dụng nâng cao.

### Có những gì trong hệ sinh thái STM32Cube?



Software  
Tools

Embedded  
Software



#### Bộ công cụ phần mềm STM32Cube:

- **STM32CubeMX**, một công cụ phần mềm được sử dụng để cấu hình cho bất kỳ thiết bị STM32 nào. Giao diện người dùng đồ họa dễ sử dụng và có thể sinh code C từ việc cấu hình cho các thiết bị lõi ARM Cortex-M và tạo Linux device tree source cho các thiết bị lõi ARM Cortex-A.
- **STM32CubeIDE**, một môi trường phát triển tích hợp. Dựa trên các giải pháp nguồn mở như Eclipse, GNU C/C++ toolchain. IDE này bao gồm các tính năng báo cáo biên dịch chương trình và các tính năng gỡ lỗi nâng cao. Nó cũng được tích hợp thêm công cụ STM32CubeMX bên trong để tiện cho việc cấu hình và sinh code.
- **STM32CubeProgrammer**, một công cụ cung cấp một môi trường dễ sử dụng và hiệu quả cho việc đọc, ghi và xác minh các thiết bị và bộ nhớ ngoài thông qua nhiều chuẩn giao thức truyền thông có sẵn như JTAG, SWD, UART, USB DFU, I2C, SPI, CAN, ...vv.

- **STM32CubeMonitor**, cung cấp các công cụ giám sát mạnh mẽ giúp cho đội ngũ phát triển có thể tinh chỉnh các hoạt động và hiệu suất của ứng dụng mà họ đã xây dựng trong thời gian thực.

#### Các phần mềm nhúng trong hệ sinh thái STM32Cube:

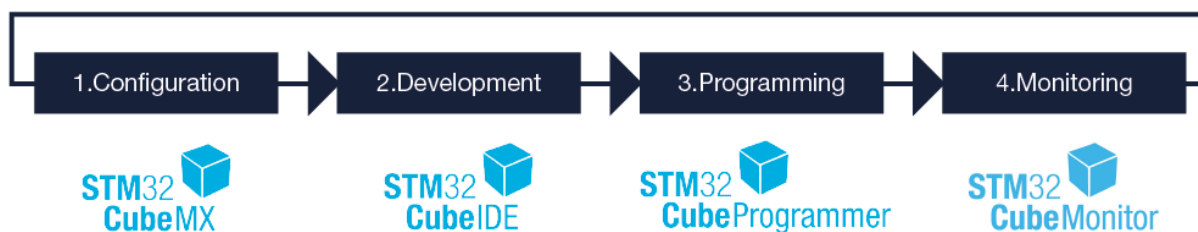
- **STM32Cube MCU và MPU packages**, dành riêng cho từng dòng vi điều khiển STM32  
Các package này cung cấp tất cả các thành phần phần mềm nhúng (thư viện) cần có để sử dụng các ngoại vi của dòng vi điều khiển STM32 tương ứng. Chúng bao gồm các driver (HAL, low-layer, v.v.), các middleware và các chương trình ví dụ mẫu được sử dụng trong nhiều trường hợp thực tế.
- **STM32Cube expansion packages**, dành cho các giải pháp định hướng ứng dụng.  
Đây là các package mở rộng của STM32Cube MCU nhằm cung cấp thêm các thành phần phần mềm nhúng, các package này có thể được thiết kế bởi hãng ST hoặc các đối tác của họ để tạo ra thêm các phần mềm nhúng cho dòng vi điều khiển STM32 để gia tăng sự tiện nghi khi lựa chọn sử dụng dòng vi điều khiển STM32.

#### STM32Cube, một giải pháp tiết kiệm thời gian toàn diện cho mọi người dùng STM32

STM32Cube mang tất cả các công cụ và các phần mềm nhúng tới cho người dùng STM32 một cách đơn giản, tích hợp. Hãng ST đưa ra một giải pháp thân thiện hơn với các doanh nghiệp: Miễn phí bản quyền, các kỹ sư phát triển được hưởng lợi từ phần mềm miễn phí và nguồn mở và họ có thể chia sẻ các mã nguồn mà họ đã phát triển trên các thiết bị STM32.

#### Các phần mềm phát triển:

Bộ công cụ phát triển phần mềm STM32Cube dựa trên quy trình phát triển lặp 4 bước:



Mỗi công cụ trong hệ sinh thái STM32Cube ở trên được ST phát hành thành các bản độc lập. Các công cụ phát triển khác dành cho STM32 từ các đối tác đủ điều kiện hoặc bên thứ 3 bao gồm IAR, EWARM hoặc Keil MDK-ARM IDE có thể nhận được các lợi ích từ việc khai thác các tính năng của các phần mềm trong hệ sinh thái STM32Cube.

Part Number	Configuration	Initialization code Generation	Code Edition	Code Building	Debugging	Binary Programming	Monitoring
STM32CubeMX	✓	✓					
STM32CubeIDE	✓ (Integrates STM32CubeMX)	✓ (Integrates STM32CubeMX)	✓	✓	✓	✓	
STM32CubeProgrammer						✓	
STM32CubeMonitor							✓

## Phần mềm nhúng

### Các STM32Cube MCU Package

Một STM32Cube MCU Package là một bộ phần mềm nhúng điều khiển các thiết bị ngoại vi của vi điều khiển hoặc vi xử lý.

Mỗi package bao gồm các driver tiêu chuẩn, trong phiên bản được tối ưu hóa là Low-Layer API và phiên bản portable HAL API. Một số middleware cũng được bổ sung dành cho các dòng vi điều khiển.

Part Number	Low-Layer API	HAL API	FreeRTOS RTOS	FatFS File System	Graphics STemWin	Graphics TouchGFX	USB Host ST Library	USB Device ST Library	LwIP TCP/IP	Touch Sense ST Library	USBPD Library	OpenAMP Library	BLE Stack	OpenThread Stack	TF-M	mbd TLS	mbd crypto
STM32CubeF0	✓	✓	✓	✓	✓			✓		✓							
STM32CubeF1	✓	✓	✓	✓	✓		✓	✓	✓							✓	
STM32CubeF2	✓	✓	✓	✓	✓		✓	✓	✓							✓	
STM32CubeF3	✓	✓	✓	✓	✓			✓		✓							
STM32CubeF4	✓	✓	✓	✓	✓	✓	✓	✓	✓							✓	
STM32CubeF7	✓	✓	✓	✓	✓	✓	✓	✓	✓							✓	
STM32CubeH7	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓				✓	
STM32CubeG0	✓	✓	✓	✓							✓						
STM32CubeG4	✓	✓	✓	✓				✓			✓						
STM32CubeL0	✓	✓	✓	✓				✓		✓							
STM32CubeL1	✓	✓	✓	✓	✓		✓	✓		✓							
STM32CubeL4	✓	✓	✓	✓	✓		✓	✓		✓							
STM32CubeL5	✓	✓	✓	✓				✓		✓	✓				✓	✓	✓
STM32CubeWB	✓	✓	✓	✓				✓		✓			✓	✓			
STM32CubeMP1	✓	✓	✓									✓					

### STM32Cube Expansion Packages

Các gói mở rộng STM32Cube chứa các thành phần phần mềm nhúng bổ sung cho các chức năng của STM32Cube MCU Packages và cho phép sử dụng các thiết bị STM32 trong các lĩnh vực như cảm biến, quản lý năng lượng, kết nối âm thanh cùng với dòng điều khiển STM32 được thiết kế riêng phù hợp với các yêu cầu ứng dụng.

Số lượng các STM32Cube Expansion Package liên tục tăng nhờ hãng ST và các đối tác.

Ngày nay, các kỹ sư phát triển có thể tìm các gói phần mềm mở rộng sử dụng cho việc kết nối đám mây (Amazon AWS, Microsoft Azure, IBM Watson,...), LoRa, kết nối mạng di động, NFC, các giao thức truyền thông công nghiệp, thư viện tiền điện tử, sensor drivers, các thuật toán điều khiển motor, các thư viện self-test, và nhiều hơn nữa

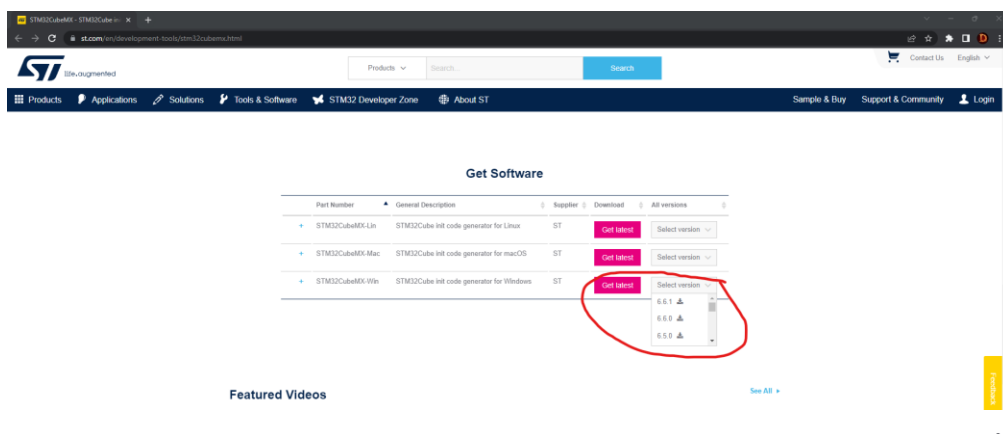
# Hướng dẫn cài đặt phần mềm STM32CubeMX

STM32CubeMX là một công cụ hỗ trợ cấu hình và sinh code cho MCU STM32. Tất cả các công việc cấu hình, nâng cấp đều được thực hiện qua giao diện đồ họa. Việc này giúp cho việc lập trình trên STM32 dễ dàng hơn, rút ngắn được thời gian nghiên cứu và phát triển. Bài viết này sẽ hướng dẫn các bạn các bước cài đặt STM32CubeMX.

## Bước 1:

Tải STM32 CubeMX [tại đây](#).

Ở mục “Get Software”. Chọn phiên bản hoặc chọn Get latest. Điền thông tin họ, tên, email. Nhấn Download rồi vào mail để lấy link.



## Get Software

If you have an account on my.st.com, login and download the software without any further validation steps.

Login/Register

If you don't want to login now, you can download the software by simply providing your name and e-mail address in the form below and validating it.

This allows us to stay in contact and inform you about updates of this software.

For subsequent downloads this step will not be required for most of our software.

First Name:

Last Name:

E-mail address:

Please review our [Privacy Statement](#) that describes how we process your profile information and how to assert your personal data protection rights

☐ Please keep me informed about future updates for this software or new software in the same category

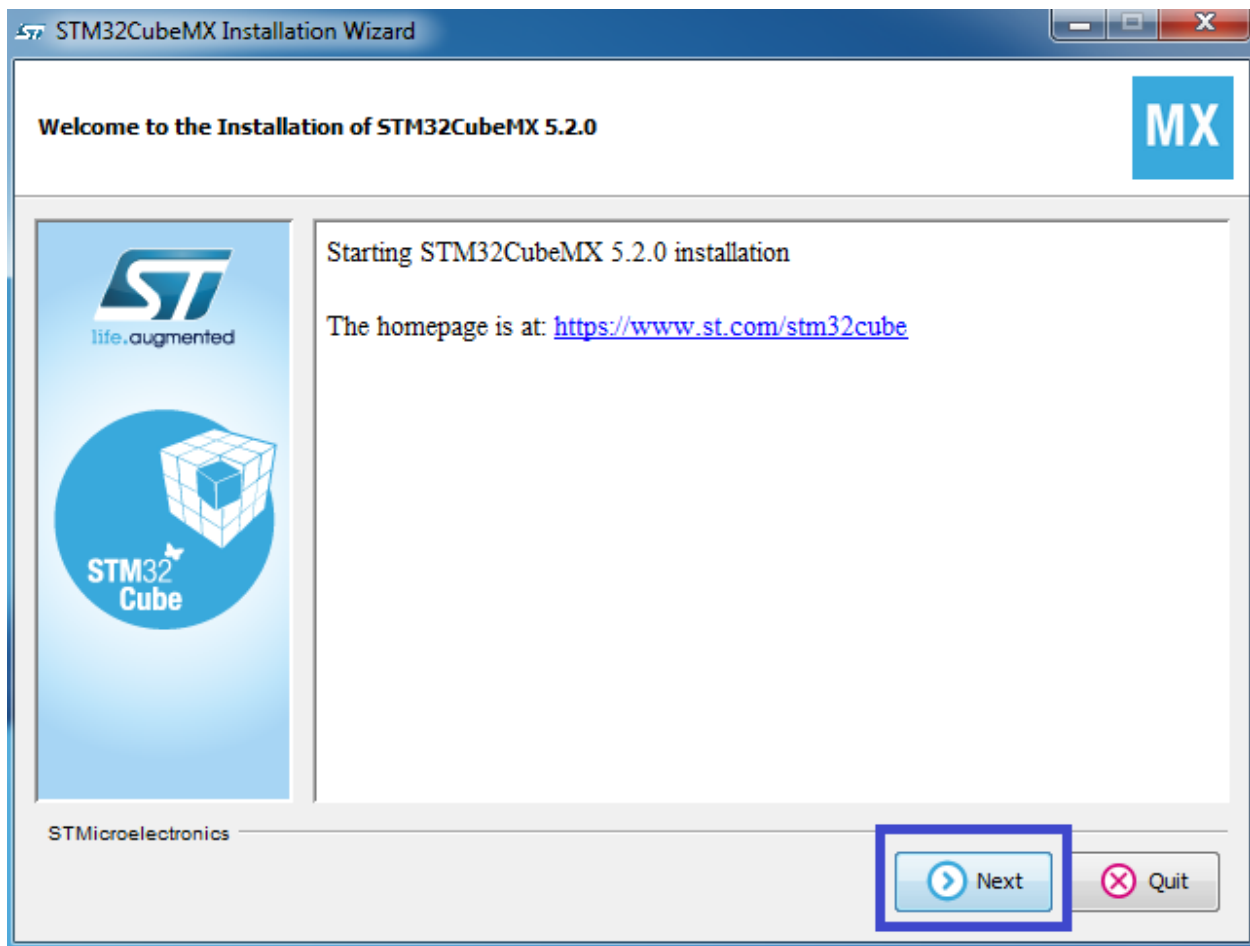
Download

Ở mục License Agreement chọn **Accept**. Sau đó chọn đường dẫn thư mục lưu file và chọn **Save**.

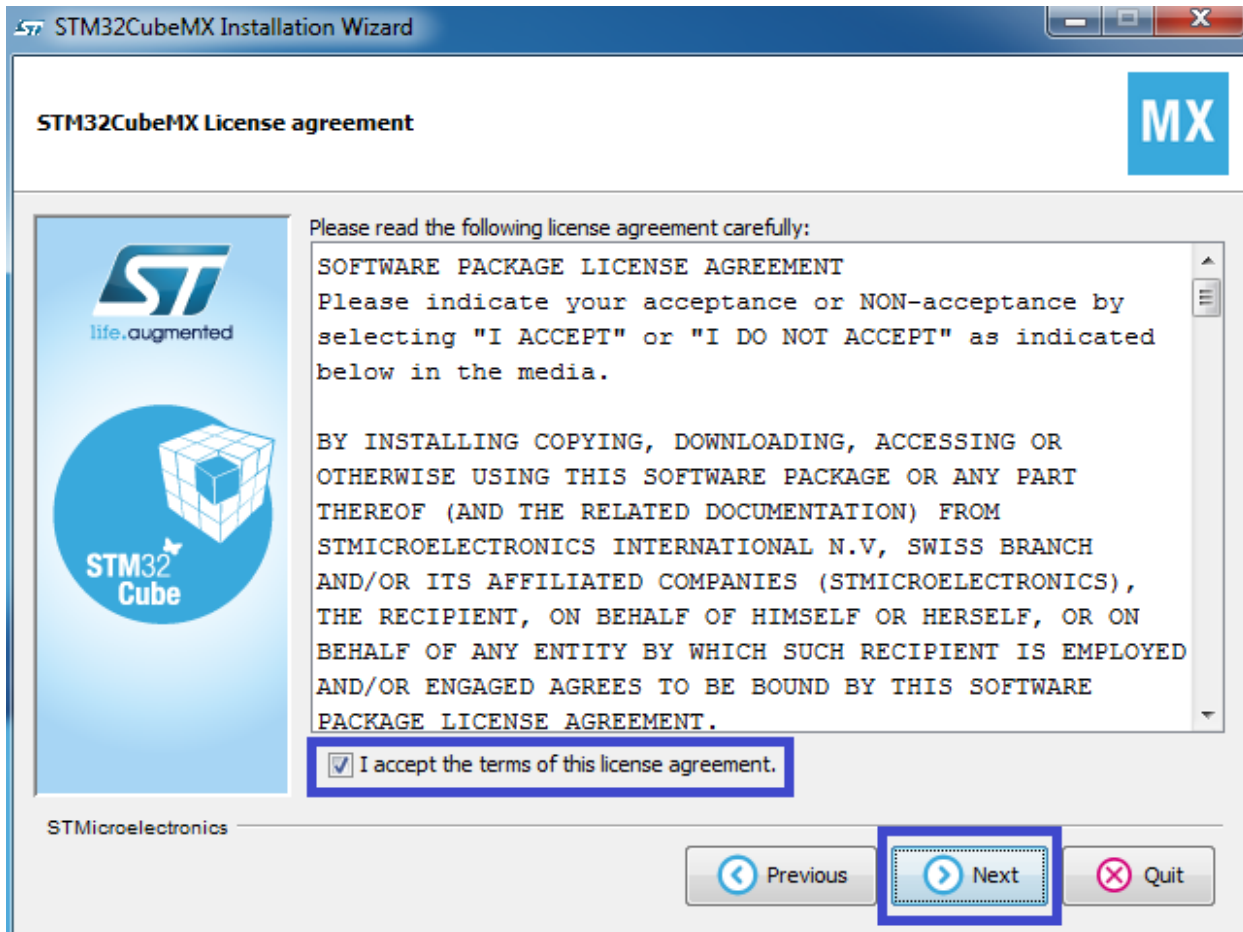
## Bước 2:

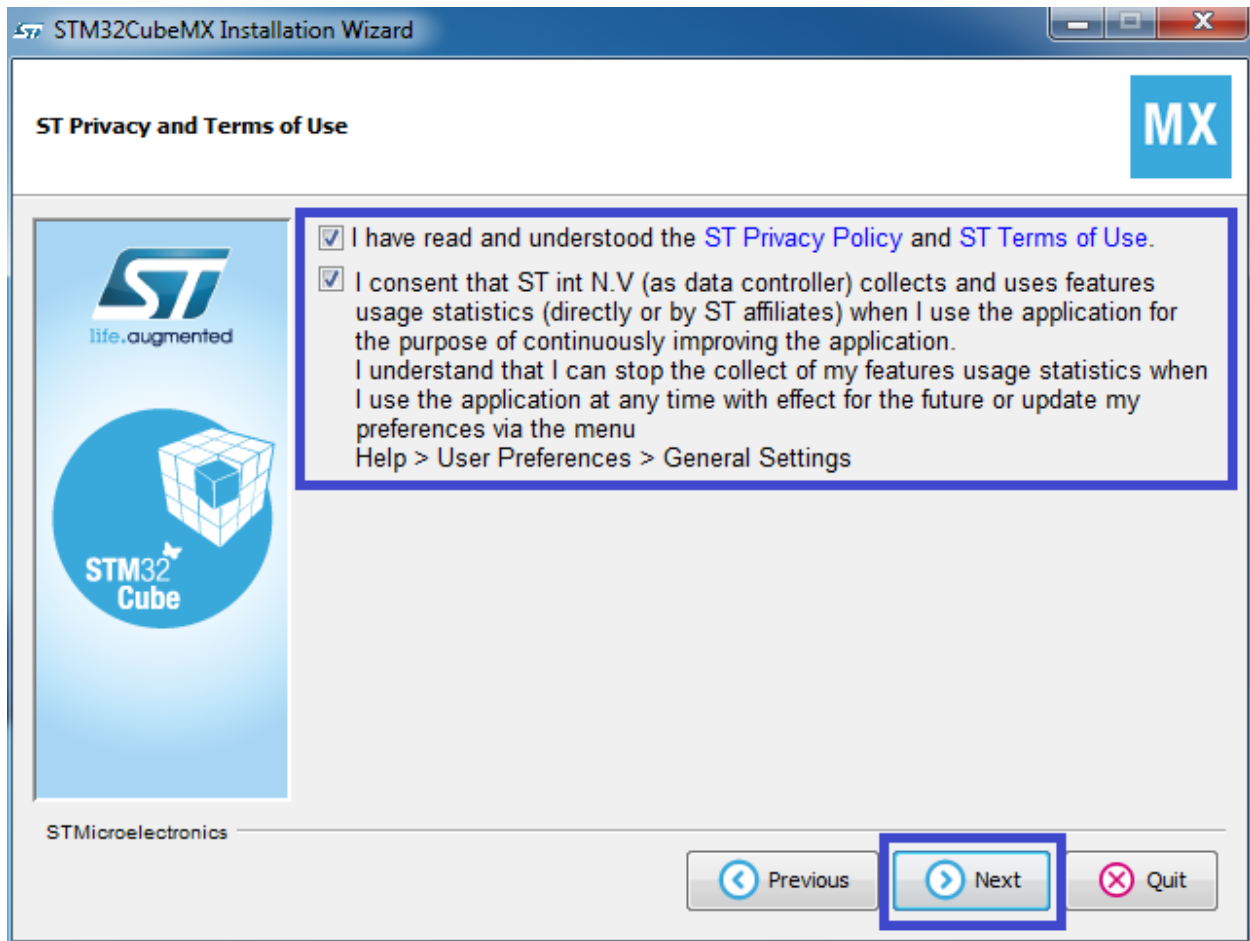
Giải nén file en.stm32cubemx-win\_vx-x-x.zip vừa tải và chạy file SetupSTM32CubeMX-x.x.x-Win.exe

Giao diện cài đặt phần mềm CubeMX:



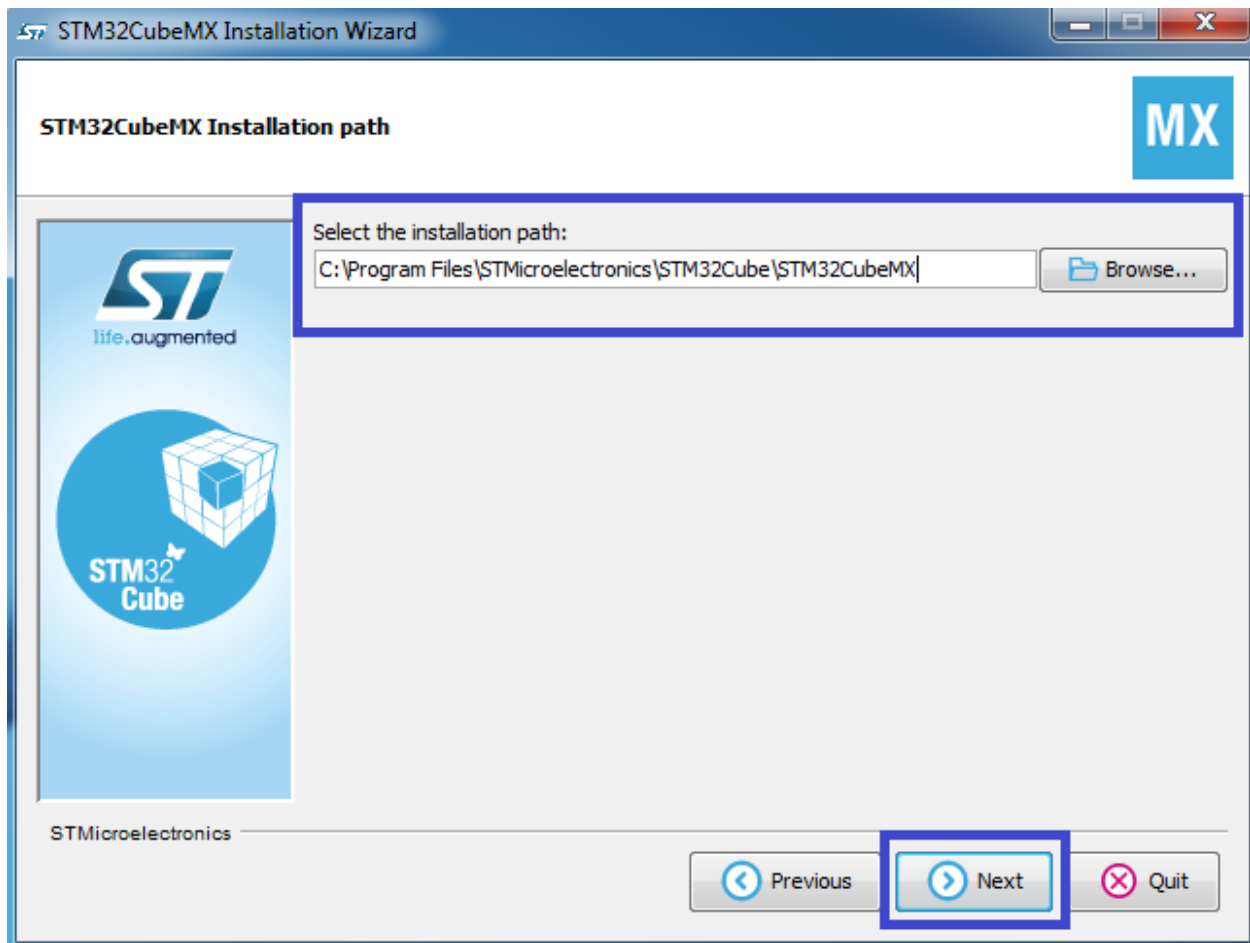
Tiếp theo, tick chọn "I accept the terms of this license agreement." và nhấn **Next**:





Chọn địa chỉ thư mục chứa file cài đặt, nên để theo thư mục gốc được định sẵn:





### Bước 3:

Nhấn **Next** cho các quá trình tiếp theo và chờ quá trình cài đặt hoàn tất.

Sau khi quá trình hoàn tất, nhấn **Done** để kết thúc việc cài đặt.

# Hướng dẫn cài đặt phần mềm lập trình Keil C $\mu$ Vision

## Bước 1:

Đầu tiên các bạn truy cập link tải phần mềm [tại đây](#).

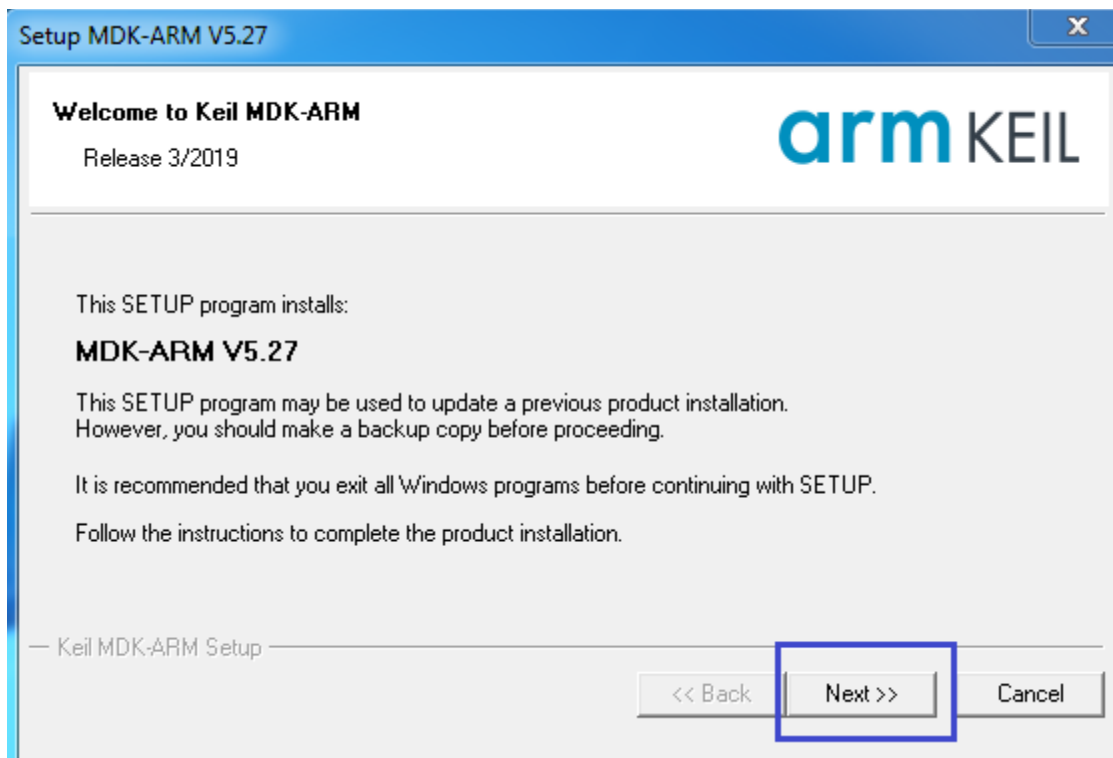
Sau đó kích chọn **MDK-Arm** để tải. (Phiên bản hiện tại là 5.37.1.0)

Tiếp theo kích chọn **MDK527.EXE**

Chọn đường dẫn để lưu file và nhấn **Save** để download.

## Bước 2:

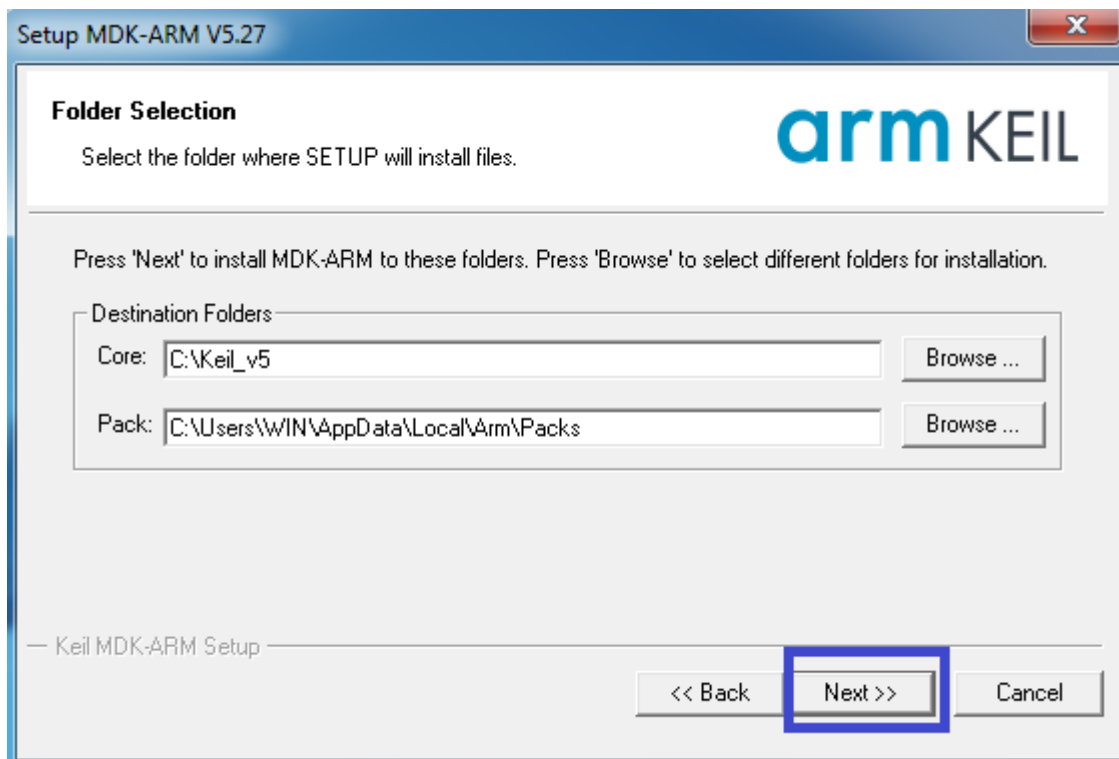
Sau khi download xong, các bạn mở file setup vừa tải lên:



Chọn *I agree to all terms of the preceding License Agreement* -> Next



Điền thông tin và chọn đường dẫn lưu phần mềm:



The screenshot shows the 'Folder Selection' window of the 'Setup MDK-ARM V5.27' installer. The window has a blue title bar and a header with the 'arm KEIL' logo. Below the header, it says 'Select the folder where SETUP will install files.' and 'Press 'Next' to install MDK-ARM to these folders. Press 'Browse' to select different folders for installation.' There are two text input fields: 'Core:' with the value 'C:\Keil\_v5' and 'Pack:' with the value 'C:\Users\WIN\AppData\Local\Arm\Packs'. Each field has a 'Browse ...' button to its right. At the bottom, there are three buttons: '<< Back', 'Next >>', and 'Cancel'. The 'Next >>' button is highlighted with a blue rectangular box.

Setup MDK-ARM V5.27

**Folder Selection**

Select the folder where SETUP will install files.

Press 'Next' to install MDK-ARM to these folders. Press 'Browse' to select different folders for installation.

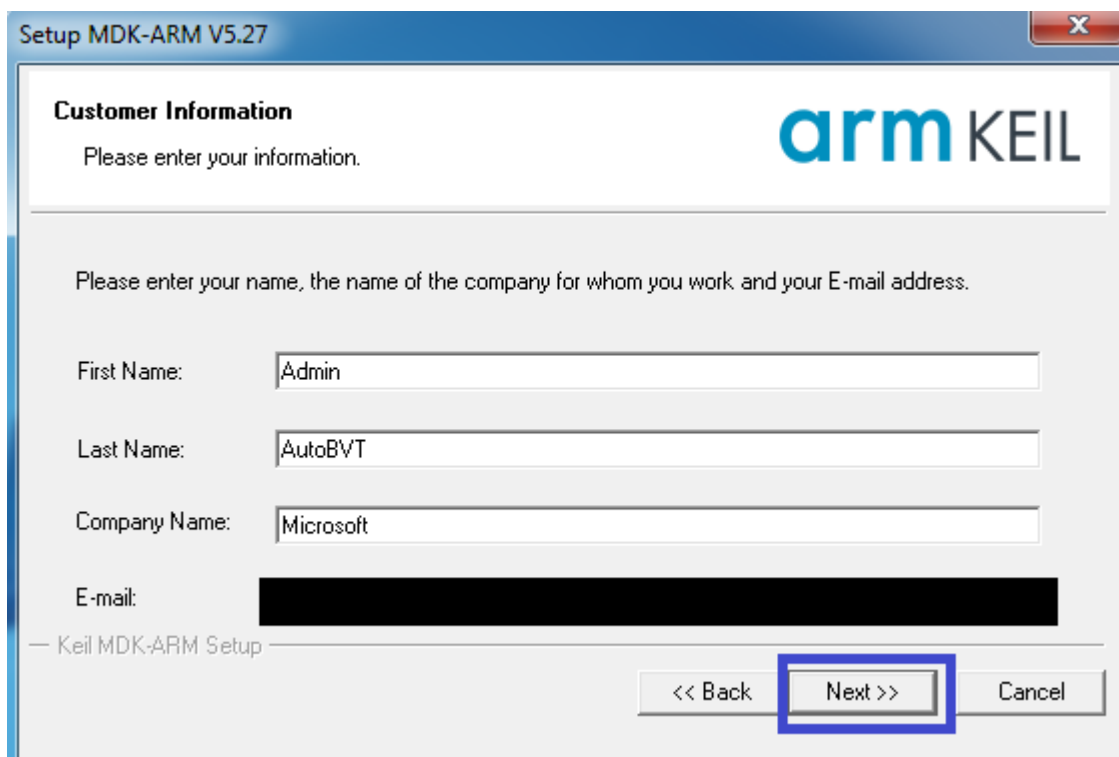
Destination Folders

Core: C:\Keil\_v5 Browse ...

Pack: C:\Users\WIN\AppData\Local\Arm\Packs Browse ...

— Keil MDK-ARM Setup —

<< Back Next >> Cancel



The screenshot shows the 'Customer Information' window of the 'Setup MDK-ARM V5.27' installer. The window has a blue title bar and a header with the 'arm KEIL' logo. Below the header, it says 'Please enter your information.' and 'Please enter your name, the name of the company for whom you work and your E-mail address.' There are four text input fields: 'First Name:' with the value 'Admin', 'Last Name:' with the value 'AutoBVT', 'Company Name:' with the value 'Microsoft', and 'E-mail:' which is currently empty. At the bottom, there are three buttons: '<< Back', 'Next >>', and 'Cancel'. The 'Next >>' button is highlighted with a blue rectangular box.

Setup MDK-ARM V5.27

**Customer Information**

Please enter your information.

Please enter your name, the name of the company for whom you work and your E-mail address.

First Name: Admin

Last Name: AutoBVT

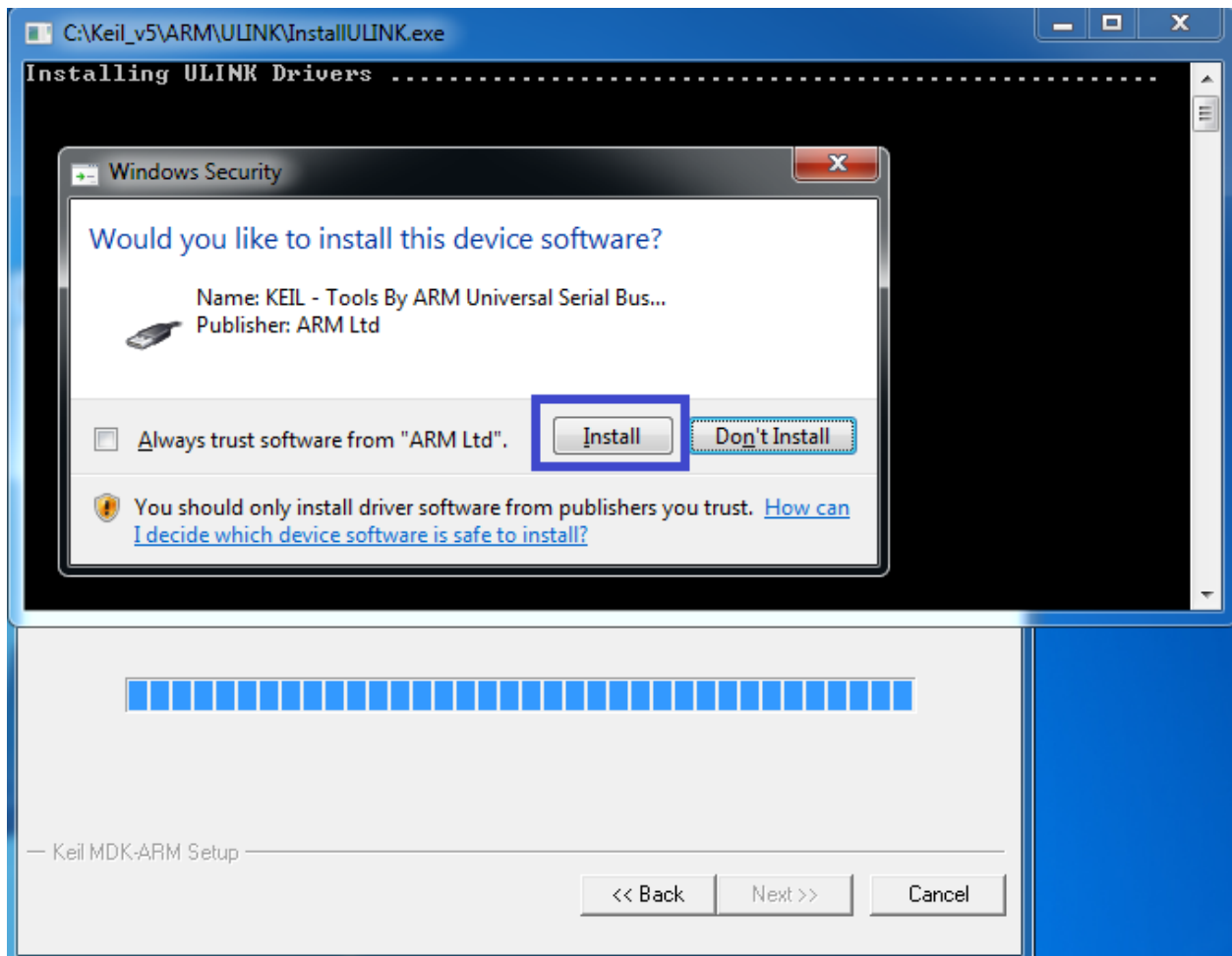
Company Name: Microsoft

E-mail:

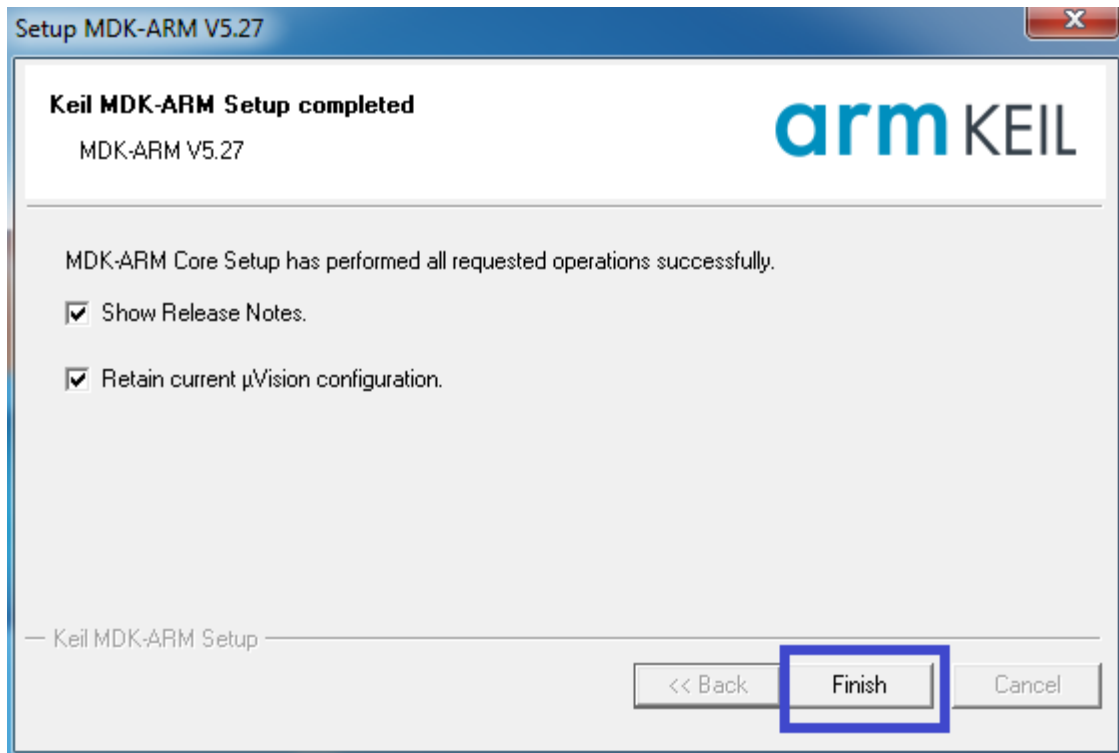
— Keil MDK-ARM Setup —

<< Back Next >> Cancel

Sau khi cài đặt hoàn tất, hộp thoại Windows Security sẽ xuất hiện. Chọn **Install** để tiếp tục:



Chọn **Finish** để kết thúc quá trình cài đặt:



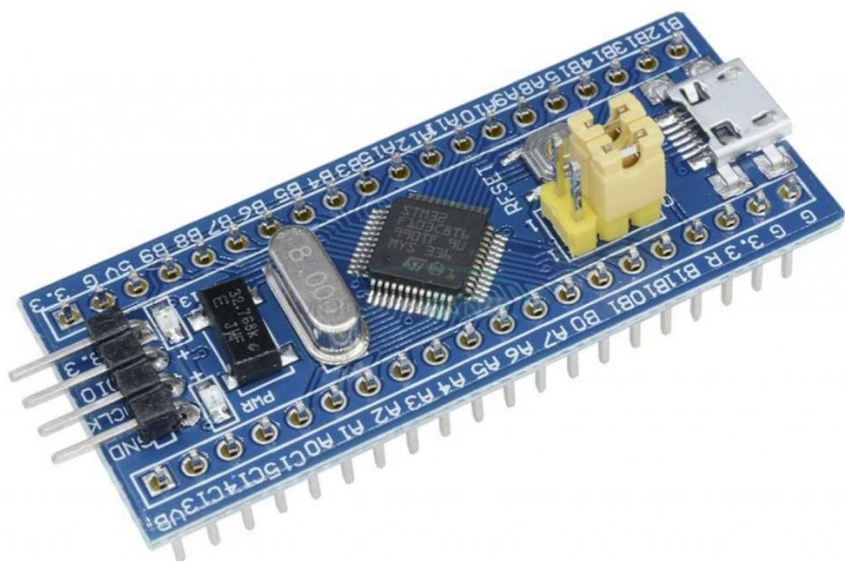
### Bước 3:

Sau khi cài đặt xong, ta tiếp tục cài đặt các gói package phù hợp với dòng chip đang sử dụng cho Keil C.

#### **\*Lưu ý:**

Đây là bản miễn phí nên dung lượng code bị giới hạn (*vẫn đủ cho những người mới tiếp cận vi điều khiển STM32 sử dụng*). Nếu như muốn sử dụng phiên bản đầy đủ thì có thể đặt mua từ trang web của nhà sản xuất.

Ngoài ra với phiên bản Keil C mới nhất, các board phát triển STM32 do bên thứ 3 phát triển thì chức năng Reset and Run không hoạt động. Vì vậy sau khi nạp code cho các board này, các bạn cần nhấn nút Reset trên board mạch thì code mới chạy.



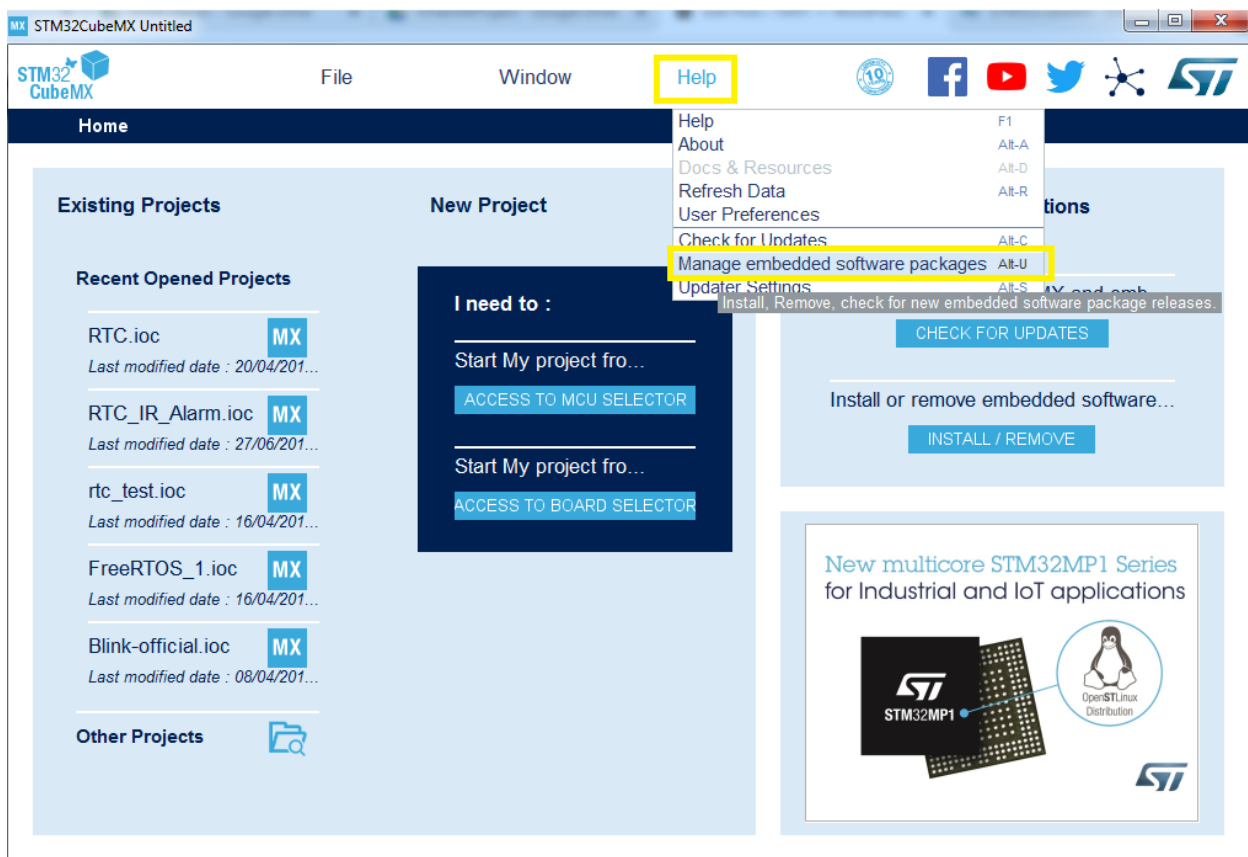
# Hướng dẫn cài đặt gói phần mềm cho các dòng STM32 trên STM32CubeMX và Keil C

Bài viết này sẽ hướng dẫn cách tải các package cần thiết cho các dòng vi điều khiển STM32 trên STM32CubeMX và Keil C sau khi đã cài đặt xong những phần mềm này. Việc cài đặt package này giúp cho STM32CubeMX có thể sinh code được sau khi cấu hình, và Keil C có thể compile và nạp chương trình được. Các bạn tải MCU Package mới nhất của dòng MCU mà các bạn cần, ví dụ ở đây mình dùng kit STM32F103C8T6 nên mình sẽ cài package dòng STM32F10x về máy:

## Bước 1:

Ở phần mềm CubeMX:

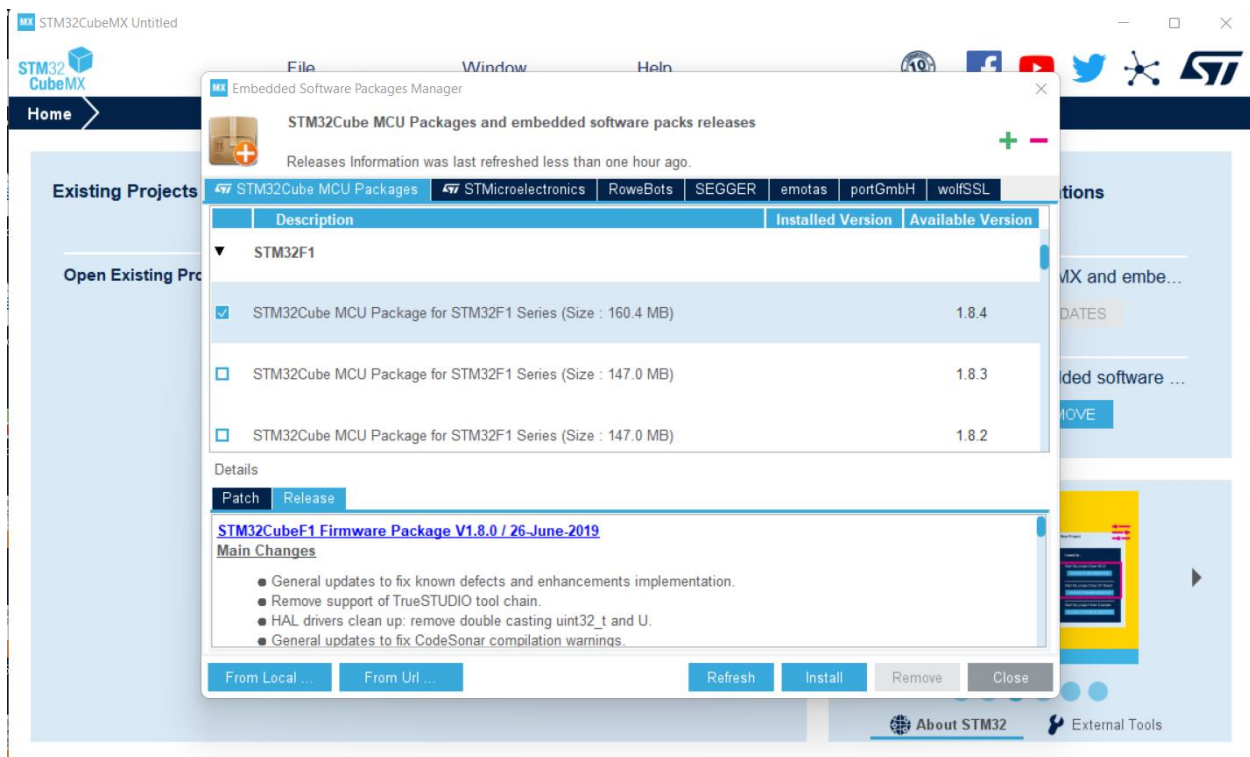
- Chọn *Help*
- Chọn *Manage embedded software packages*



## Bước 2:

- Chọn package mới nhất của dòng vi điều khiển STM32F1





– Chọn *Install Now*

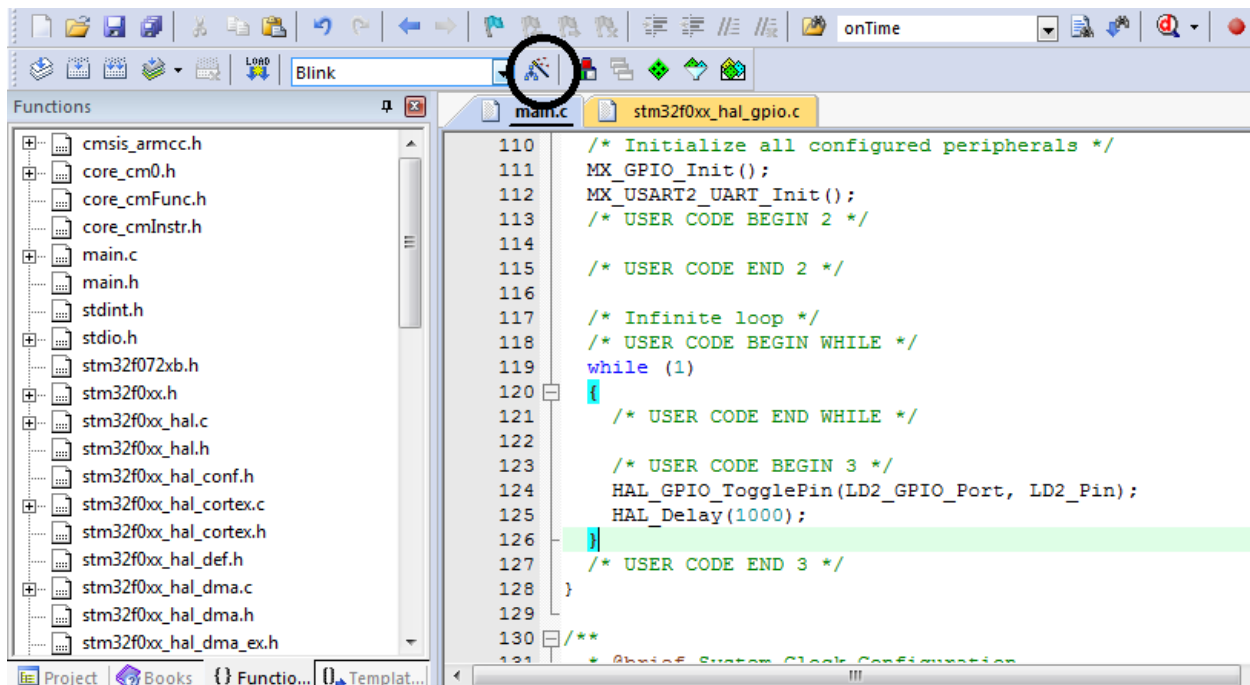
### Bước 3:

Sau khi cài đặt xong gói tin thì sẽ có dấu tích màu xanh lá cho biết đã hoàn thành việc cài đặt

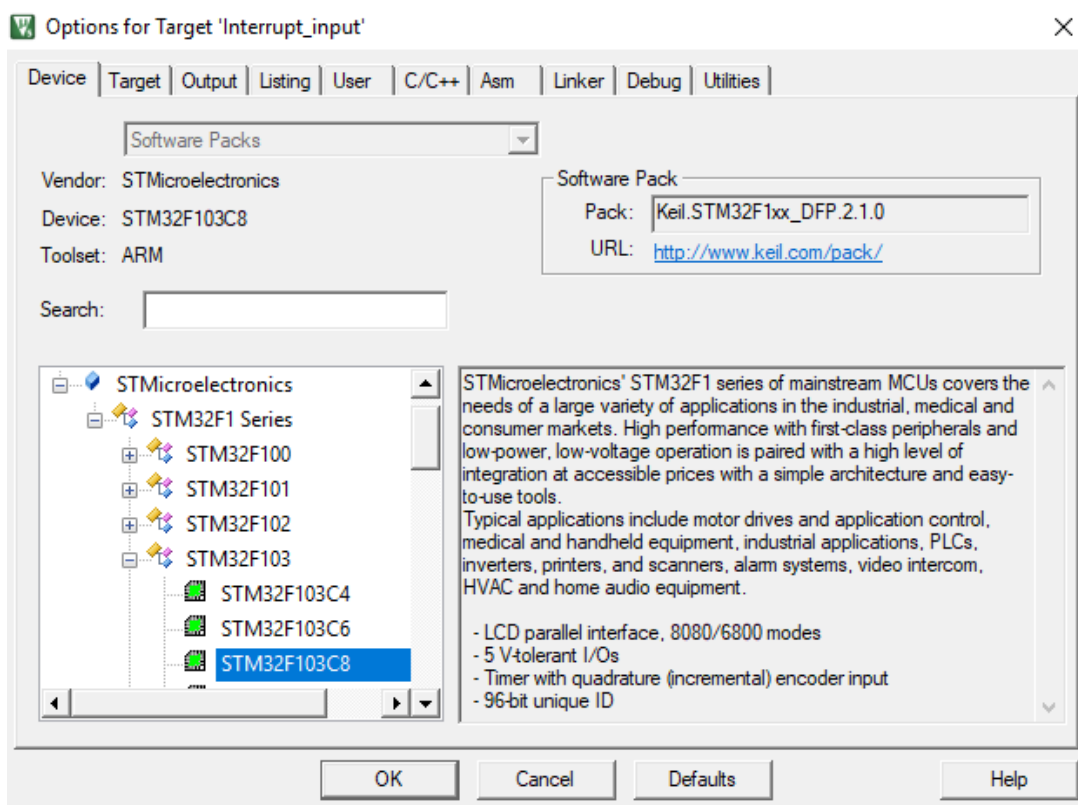
STM32Cube MCU Packages		STMicroelectronics	
	Description	Installed Version	Available Version
▶	STM32F0		
▼	STM32F1		
■	STM32Cube MCU Package for STM32F1 Series	1.7.0	1.7.0
■	STM32Cube MCU Package for STM32F1 Series	1.6.1	1.6.1
□	STM32Cube MCU Package for STM32F1 Series (Size : 97.2 MB)		1.6.0
□	STM32Cube MCU Package for STM32F1 Series (Size : 77.2 MB)		1.4.0

Khi quá trình cài đặt trên CubeMX đã hoàn thành, các bạn vào kiểm tra gói thư viện của chip STM32F103 có được cài đặt trong Keil C hay chưa bằng cách:

- Mở phần mềm Keil C uvision
- Chọn biểu tượng Options for Target trên thanh công cụ



– Nếu cài đặt đã thành công, ở ô Software Pack sẽ xuất hiện Keil.STM32F1xx\_DFP.2.1.0



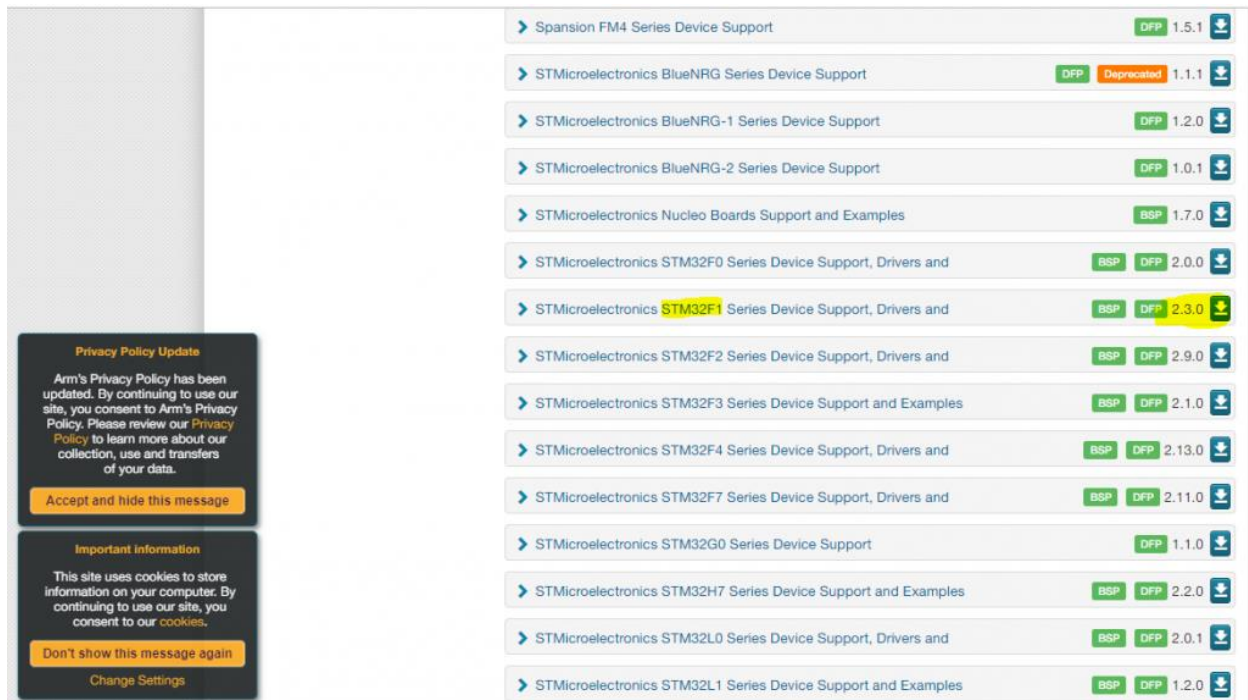
– Nếu chưa có gói package của dòng chip STM32F10X chúng ta tiếp tục cài đặt gói tin theo cách sau:

**Bước 1:**

Các bạn vào đường dẫn [tại đây](#)

### Bước 2:

Tìm kiếm và chọn link của dòng chip STM32F1, sau đó nhấn vào biểu tượng tải về:



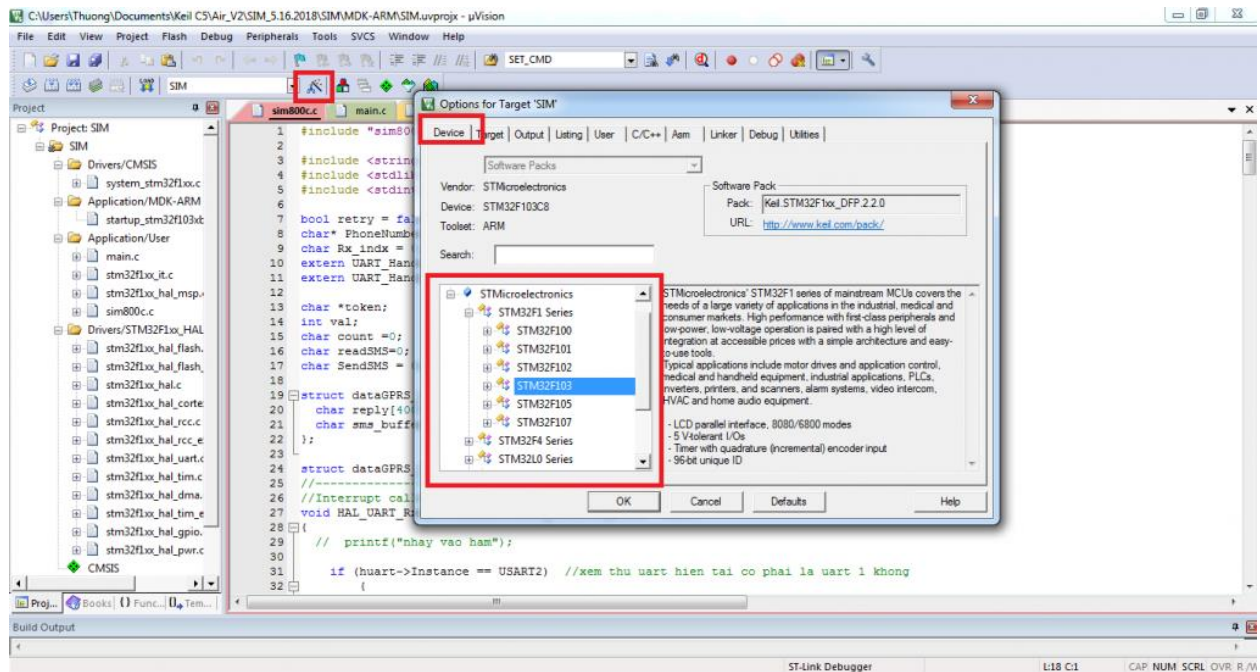
### Bước 3:

Sau khi download về các bạn mở lên và cho chạy bình thường.

### Bước 4:

Kiểm tra việc cài đặt có thành công hay không:

- Mở phần mềm Keil C
- Chọn *Device* trong *Options for target* như bước 3 ở cách đầu tiên



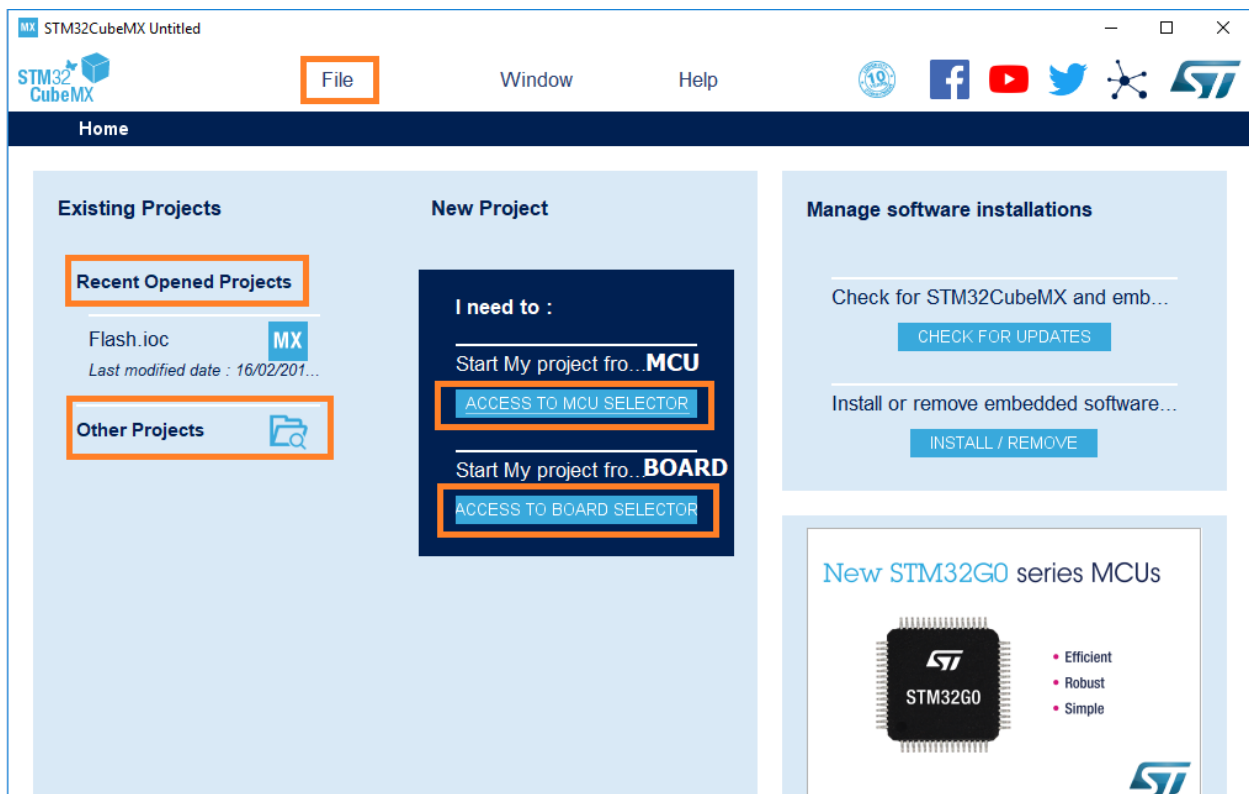
# Hướng dẫn sử dụng STM32CubeMX và Keil C để lập trình STM32

Phần mềm STM32CubeMX là một phần mềm hỗ trợ việc cấu hình các chức năng cho các dòng vi điều khiển STM32 thông qua giao diện đồ họa và tạo ra code từ các cấu hình đó. Keil C IDE là một môi trường phát triển tích hợp giúp chúng ta có thể viết code, compile, debugs, nạp chương trình xuống vi điều khiển. Bài viết này hướng dẫn cho những bạn mới bắt đầu tìm hiểu dòng vi điều khiển STM32 cách tiếp cận và sử dụng 2 phần mềm này.

## 1. Giao diện khi khởi động STM32CubeMX

Tại giao diện này các bạn có thể:

- **Tạo 1 Project mới:** File -> New Project hoặc nhấn vào ACCESS TO MCU SELECTOR nếu bạn lập trình 1 MCU STM32 bất kỳ, hoặc nhấn vào ACCESS TO BOARD SELECTOR nếu bạn lập trình trên 1 board phát triển của hãng STM32.
- **Mở project gần đây:** tại mục Recent Opened Project nhấn vào tên project bạn đã mở gần đây.
- **Mở 1 project bất kỳ:** tại mục Other Project, bấm vào biểu tượng duyệt thư mục hoặc File -> Load Project.



## 2. Tạo 1 project mới

Sau khi nhấn vào File -> New Project thì giao diện chọn vi điều khiển STM32 sẽ hiện ra:

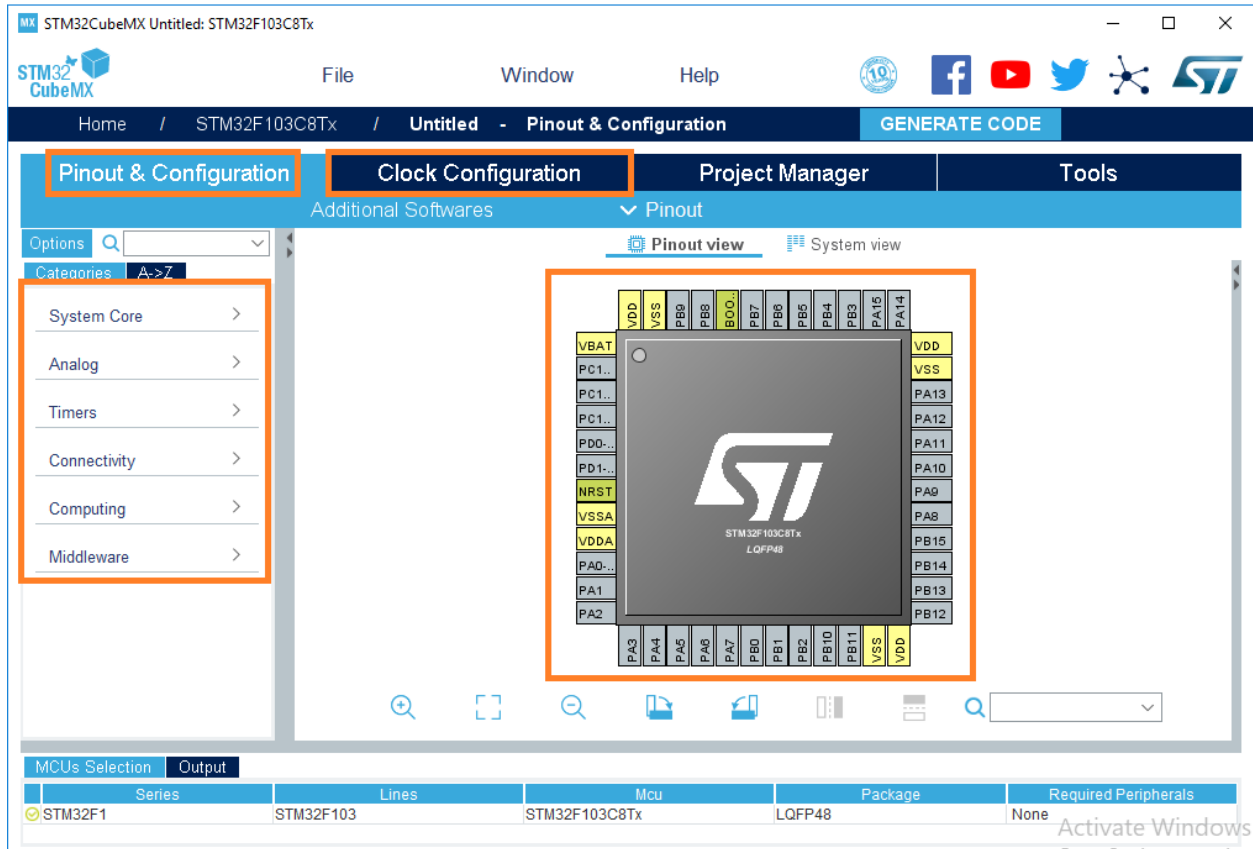
- **Chọn vi điều khiển:** tại mục Part Number Search các bạn nhập vào tên vi điều khiển mà mình muốn cấu hình (ví dụ vi điều khiển *STM32F103C8* )
- **Bắt đầu Project:** nhấn vào *Start Project*

Bên cạnh đó, các bạn cũng có thể lọc vi điều khiển theo các thông số như giá tiền (cost), số chân nhập xuất (IO), bộ nhớ chương trình (Flash, Eeprom), bộ nhớ dữ liệu (Ram), tốc độ tối đa CPU (Freq.) tại mục MCU Filters. Sau khi chọn xong vi điều khiển, các bạn có thể đọc thêm mô tả các tính năng (Features), sơ đồ khối (Block Diagram), các tài liệu của hãng (Docs & Resources) hoặc tải về Datasheet để đọc.

The screenshot shows the STM32CubeMX software interface. The 'MCU Selector' tab is selected, and 'STM32F103C8' is entered in the search field. The 'MCU Filters' sidebar on the left allows filtering by various parameters. The main panel displays the selected MCU's details, including its features and a 'Start Project' button. Below the details, a table lists the selected MCU.

Part No.	Reference	Marketing	Unit Price for	Board	Package	Flash	RAM	IO	Freq.	SFX S.
STM32F1...	STM32F10...	Active	2.056		LQFP48	64 kByt...	20 kByt...	37	72 M...	0.0

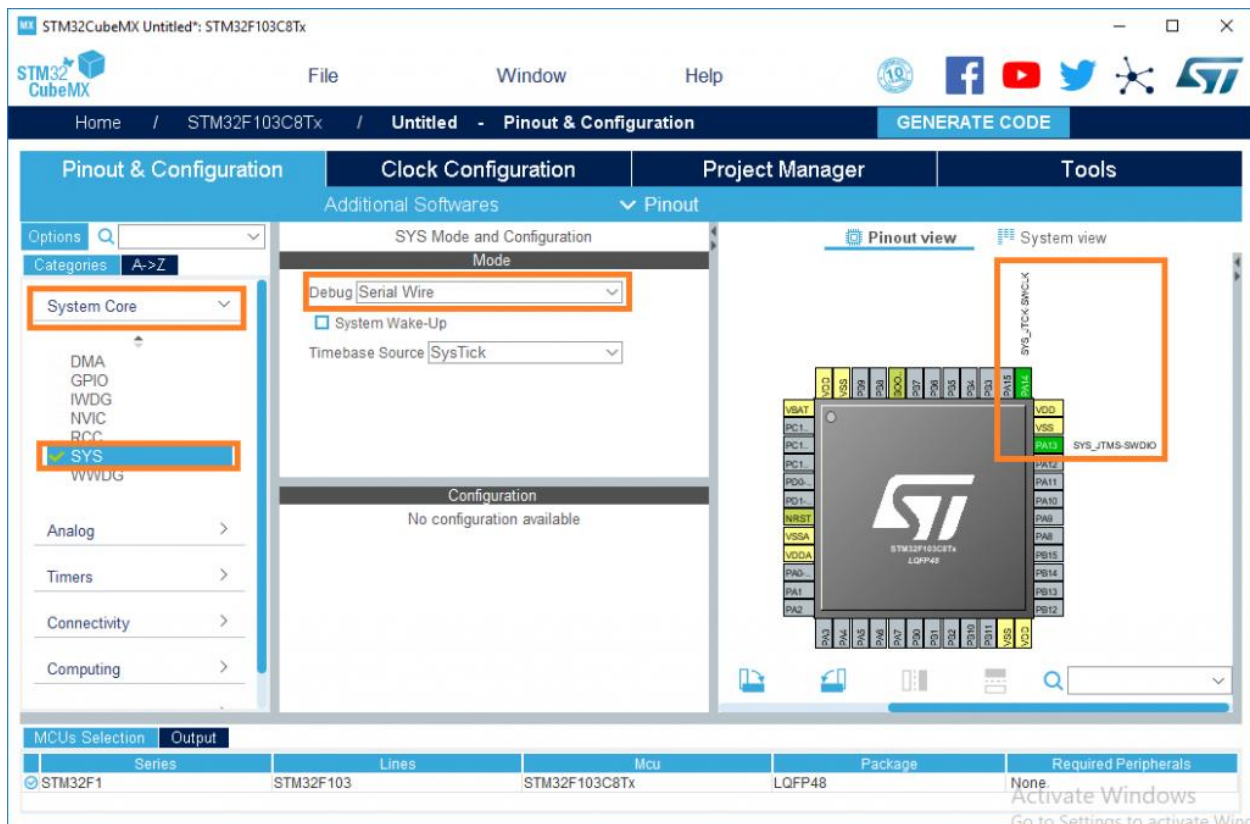
### 3. Các cấu hình



### Tại mục Pinout & Configuration:

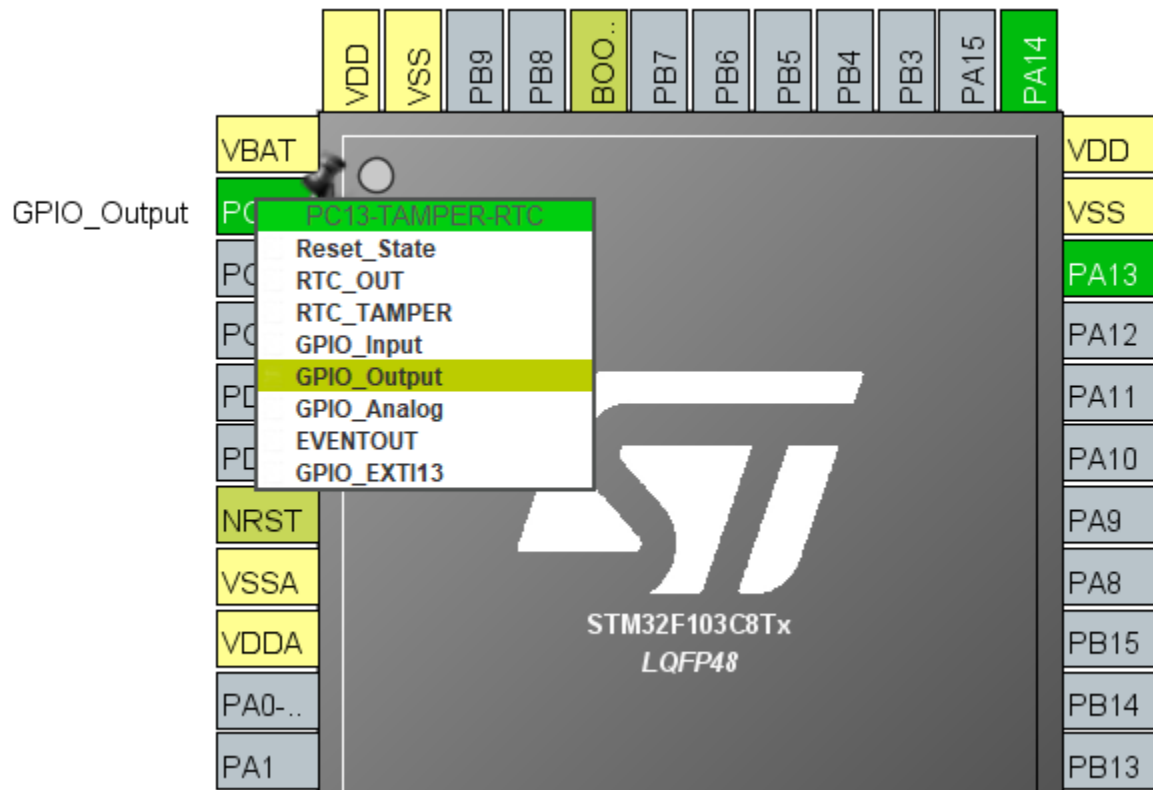
- Cấu hình nạp code:** Các bạn chọn System Core -> SYS -> Debug: Serial Wire để vi điều khiển được cấu hình nạp code thông qua chân SWDIO và SWCLK (chúng ta sử dụng mạch nạp ST- Link và kết nối với vi điều khiển STM32F103C8T6 thông qua các chân này).



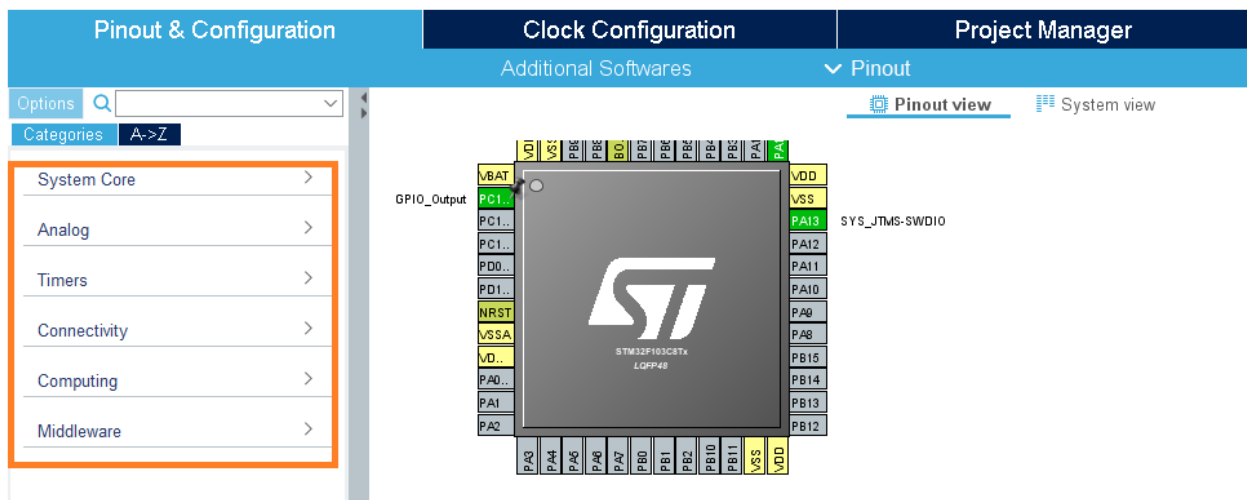


- Cấu hình các ngoại vi:** Việc cấu hình các ngoại vi như INPUT, INPUT, External Interrupt, ADC, TIMER, UART... có thể được thực hiện bằng cách chuột phải để chọn chân trực tiếp và kích chuột trái vào chân mà mình muốn cài đặt. Ví dụ : cài đặt chân PC13 hoạt động với chức năng OUTPUT (bạn có thể phóng to/thu nhỏ hình ảnh vi điều khiển bằng cách lăn chuột giữa).





Ngoài ra, các bạn cũng có thể cấu hình các ngoại vi khác tại các mục: System Core, Analog, Timers, Connectivity...

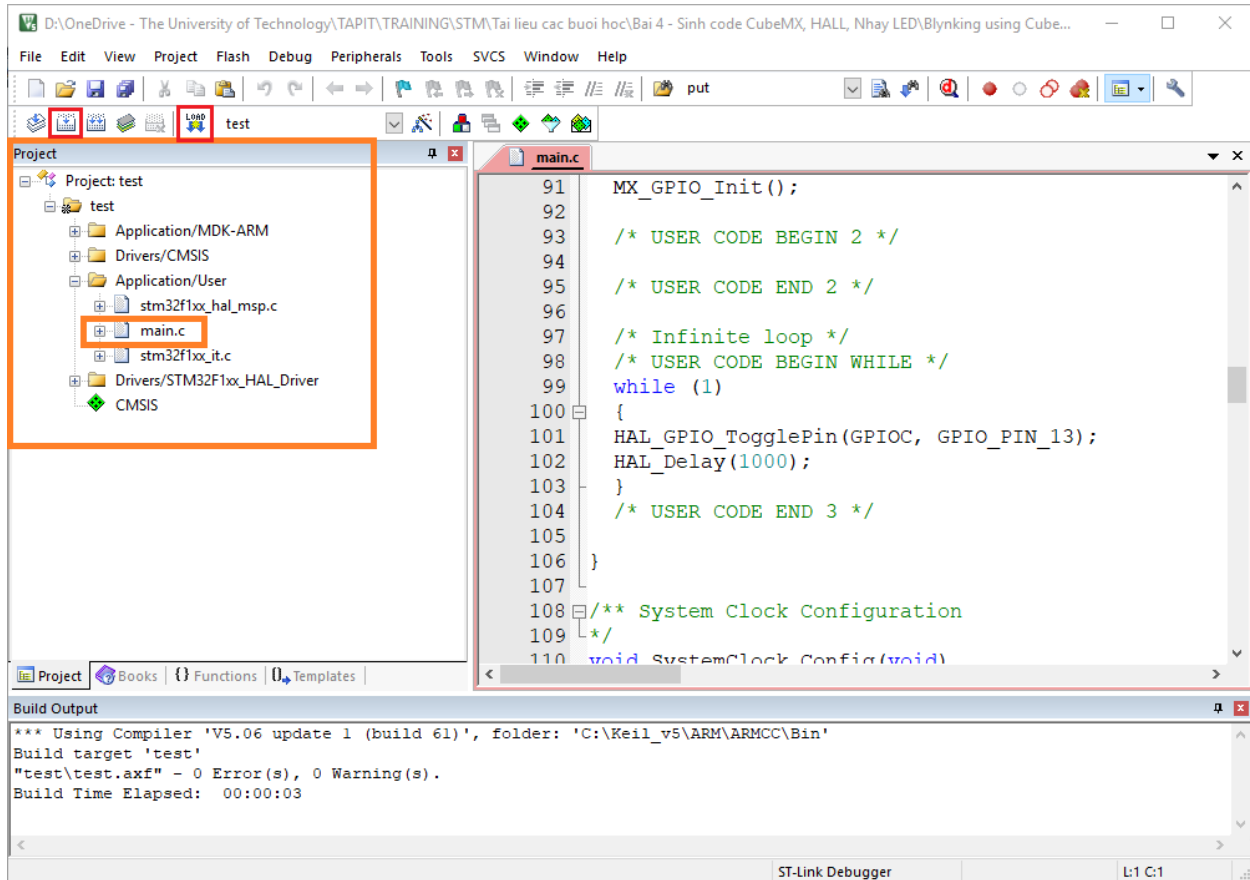


**Tại mục Clock Configuration:** Các bạn cấu hình lựa chọn nguồn tạo dao động và tần số hoạt động cho vi điều khiển (Bộ xử lý trung tâm – CPU và Peripherals – các ngoại vi) thông qua Clock tree. (Kết hợp với cấu hình RCC tại System Core)



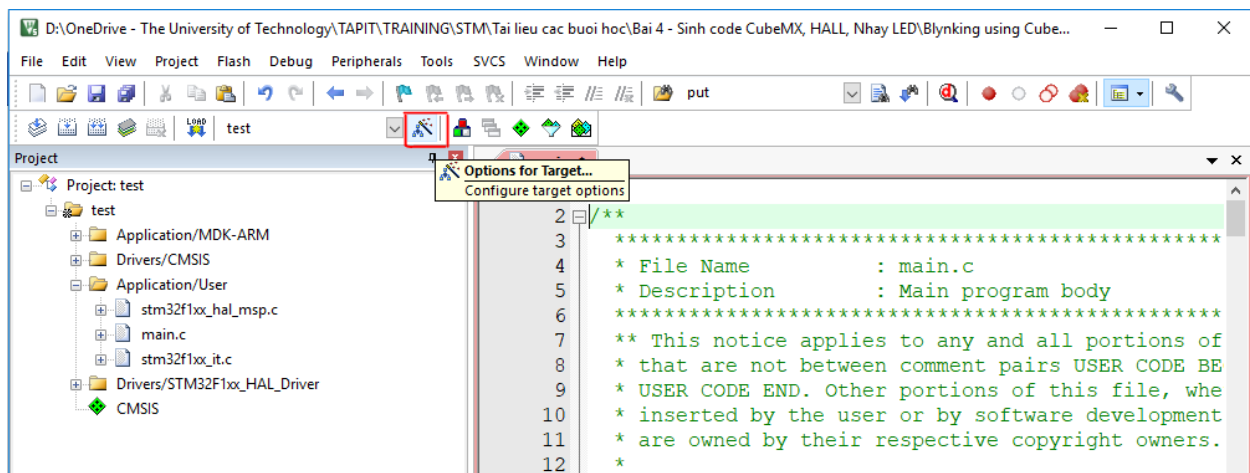
mục **Application/User**.

Có các biểu tượng **Build (F7)** để compile chương trình và **Load (F8)** để nạp chương trình.



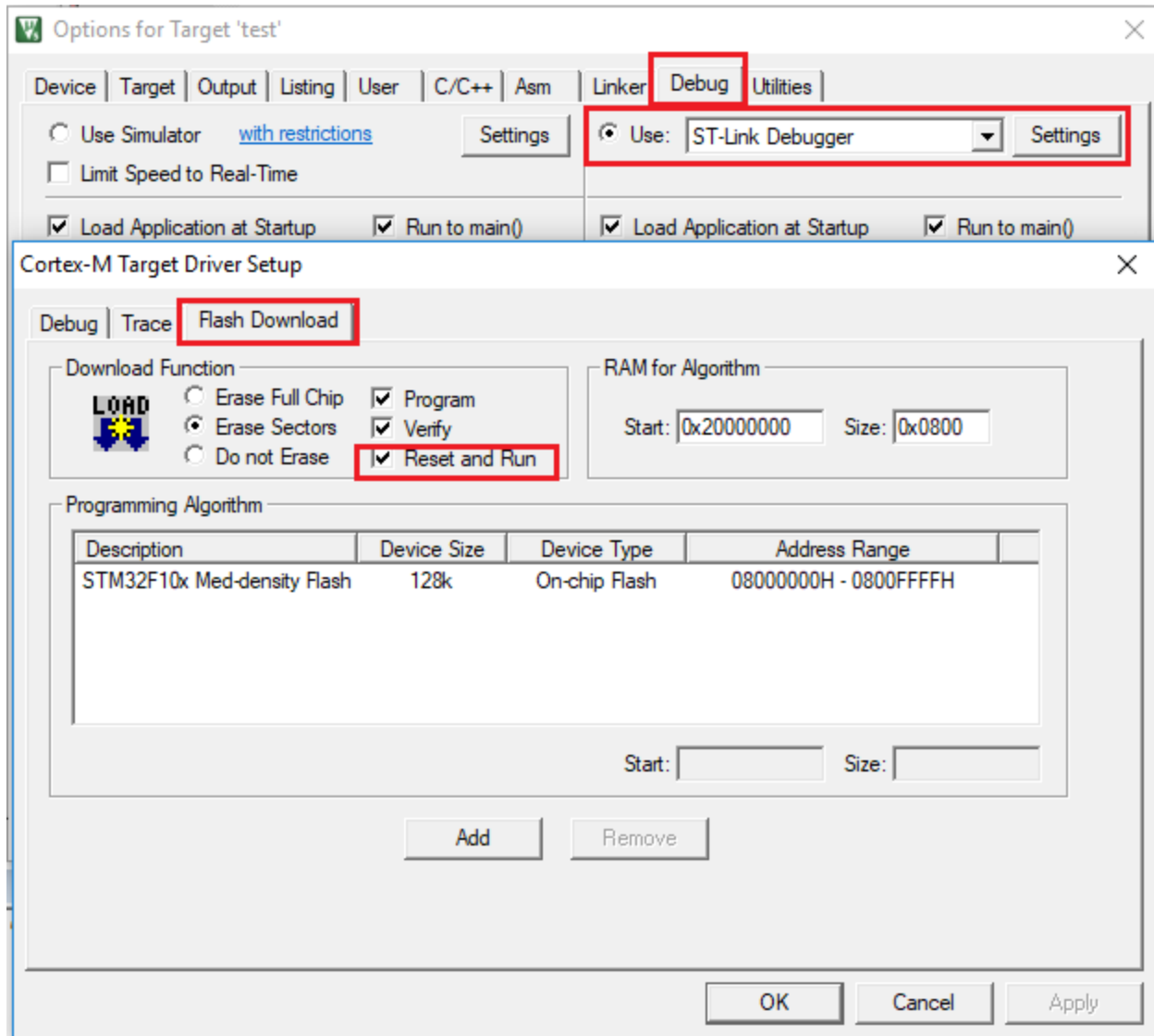
**Một số lưu ý khi nạp chương trình:**

Các bạn nhấn vào biểu tượng Option for Target hoặc **Project -> Option for Target** để thực hiện 1 số cấu hình:



Tại cửa sổ Option for Target, các bạn chọn thẻ **Debug** và tick chọn **Use ST-Link Debugger**, để nạp chương trình xuống kit (nếu chọn **Use Simulator** thì sẽ ở chế độ mô phỏng).

Nhấn vào **Settings** tại **Use: ST-Link Debugger**, cửa sổ Cortex-M Target Driver Setup, các bạn chọn thẻ **Flash Download**, tại đây nếu tick chọn **Reset and Run** thì ngay sau khi nạp, chương trình sẽ chạy ngay. Nếu không tick chọn thì các bạn nạp code xong, nhấn vào nút reset trên board mạch thì chương trình mới chạy.



# Hướng dẫn cắm dây mạch nạp ST-Link V2 sang kit STM32F103

Sơ đồ cắm dây từ mạch nạp ST-Link V2 sang kit STM32F103

3.3V – 3.3V

GND – GND

SWDIO – DIO

SWCLK – DCLK



## Hướng dẫn cài đặt Driver cho mạch nạp ST-Link V2

Mạch nạp ST-Link V2 được dùng để nạp code và thực hiện debug cho các dòng STM8 và STM32 theo chuẩn SWD. Tuy nhiên do driver STM32 ST-Link V2 không có sẵn trong Win7/8/10 bạn cần cài đặt cho nó.

### Lưu ý:

– Phần mềm cài đặt driver dưới đây chỉ sử dụng cho các phiên bản 32-bit và 64-bit của Windows® 7, 8 và 10.

– Driver cần được cài đặt trước khi kết nối ST-Link với thiết bị.

Các bước tải driver cho mạch nạp ST-Link V2:

### Bước 1:

Truy cập link tải driver [tại đây](#).

Chọn “Get Software”

Get Software		
Part Number	Supplier	Download
+ STSW-LINK009	ST	<a href="#">Get Software</a>

Tải và giải nén file *en.stsw-link009.zip*.

### Bước 2:

Trong thư mục *en.stsw-link009* sau khi đã giải nén, kích chuột phải vào file *stlink\_winusb\_install* (Windows Batch File) -> Run as administrator.

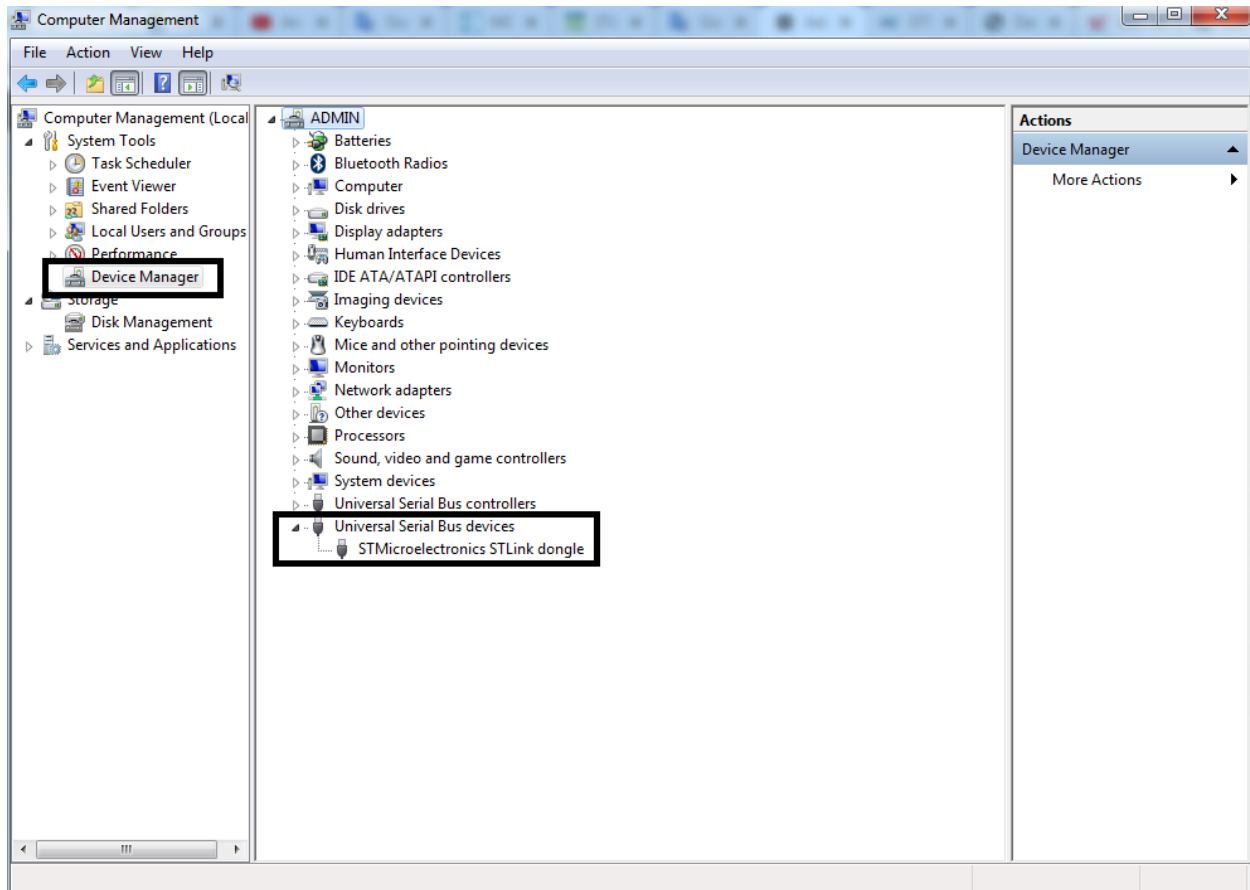
### Bước 3:

Chọn **Next** cho các quá trình cài đặt sau đó chọn **Finish**.

### Bước 4:

Kết nối ST-Link V2 với thiết bị STM32, sau đó kết nối với máy tính. Kiểm tra port để xem việc cài driver có thành công hay không. Trong bài này mình dùng kit STM32F103C8T6 nên mình sẽ kiểm tra như sau:

- Kết nối ST-Link V2 với STM32F103C8T6 như bài hướng dẫn [ở đây](#).
- Sau khi cắm vào máy tính, chọn và kích chuột phải thư mục This PC -> chọn Manage. Ở cây thư mục System Tools, kích chọn Device Manager.
- Nếu kết quả giống hình bên dưới thì việc cài đặt driver đã hoàn thiện.



# Hướng dẫn debug trên Keil C IDE – Debug chương trình vi điều khiển STM32F103

Debug (gỡ lỗi) là một kỹ năng cần có của một lập trình viên. Mục đích của Debug không chỉ dùng để loại bỏ lỗi (error) khỏi chương trình mà quan trọng debug sẽ giúp lập trình viên hiểu rõ hơn và kiểm tra được sự thực thi của chương trình. Một lập trình viên chưa trang bị cho bản thân kỹ năng debug thì sẽ rất khó kiểm soát sự thực thi chương trình của mình.

Bug (lỗi) có thể sinh ra từ chương trình bạn viết ra hoặc từ chương trình bạn kế thừa. Khi một chương trình được chạy nhưng kết quả không như bạn đang mong đợi, đó chính là lỗi (không phải lỗi cú pháp). Đối với những chương trình nhỏ, những đoạn code ngắn, bạn có thể kiểm tra các khối/dòng lệnh để tìm lỗi, nhưng cách đó sẽ không hiệu quả đối với 1 chương trình lớn hơn, với nhiều luồng xử lý, lúc này bạn nên sử dụng công cụ debug. Không chỉ vậy, để kiểm tra được nhiều nhất có thể các trường hợp mà chương trình có thể xảy ra lúc thực thi, công cụ debug sẽ giúp bạn can thiệp vào chương trình để chương trình chạy theo hướng mình muốn kiểm tra.




Trong hệ thống nhúng, có 2 cách debug chương trình của bạn:

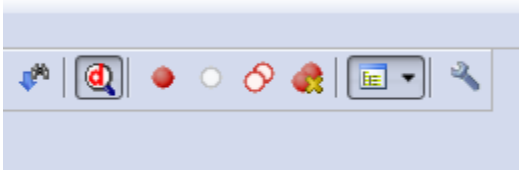
- Debug phần mềm: sử dụng debug phần mềm, không cần sử dụng board phần cứng.
- Debug phần cứng: yêu cầu phải được kết nối từ board đến máy tính.

Ở trong phần này, bài viết sẽ hướng dẫn các bạn debug phần cứng, gồm các nội dung sau:

- Quan sát sự thay đổi của thanh ghi thông qua việc nhấp nháy led
- Quan sát sự thay đổi của biến ở một kiểu dữ liệu bất kì
- Quan sát sự thay đổi của một mảng ở dữ liệu bất kì

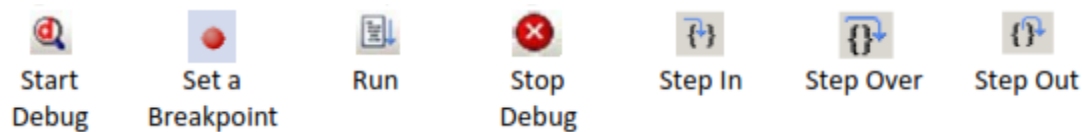
Một số công cụ trên IDE mà các bạn cần biết trước để thuận tiện trong quá trình debug:

- Nhấn **LOAD**  để nạp code vào bộ nhớ flash.
- Nhấn **debug**  để bắt đầu quá trình và bấm lại thêm 1 lần nữa để thoát chế độ debug. Bạn có thể sử dụng nút **breakpoint**  để cài đặt một điểm *break point* trong window disassembly hoặc window source.
- STM32 cho phép đến 6 break point trong quá trình debug phần cứng. Khi chương trình dừng lại ở tại 1 breakpoint, thì câu lệnh tương ứng chưa được thực thi.
- Bạn có thể quan sát câu lệnh nào đang được thực thi trên cửa sổ disassembly và cửa sổ source.



Các nút cho bộ debug





**Run:** thực hiện từ vị trí hiện tại (*hoặc vị trí bắt đầu*) cho đến khi click vào stop hoặc chương trình sẽ tạm dừng tại vị trí điểm *breakpoint*.

**Step In:** Thực thi một câu lệnh và vào một hàm nếu dòng hiện tại gọi đến hàm đó.

**Step Over:** Thực thi một câu lệnh và chạy tất các lệnh ở trong hàm nếu bước hiện tại gọi đến hàm đó.

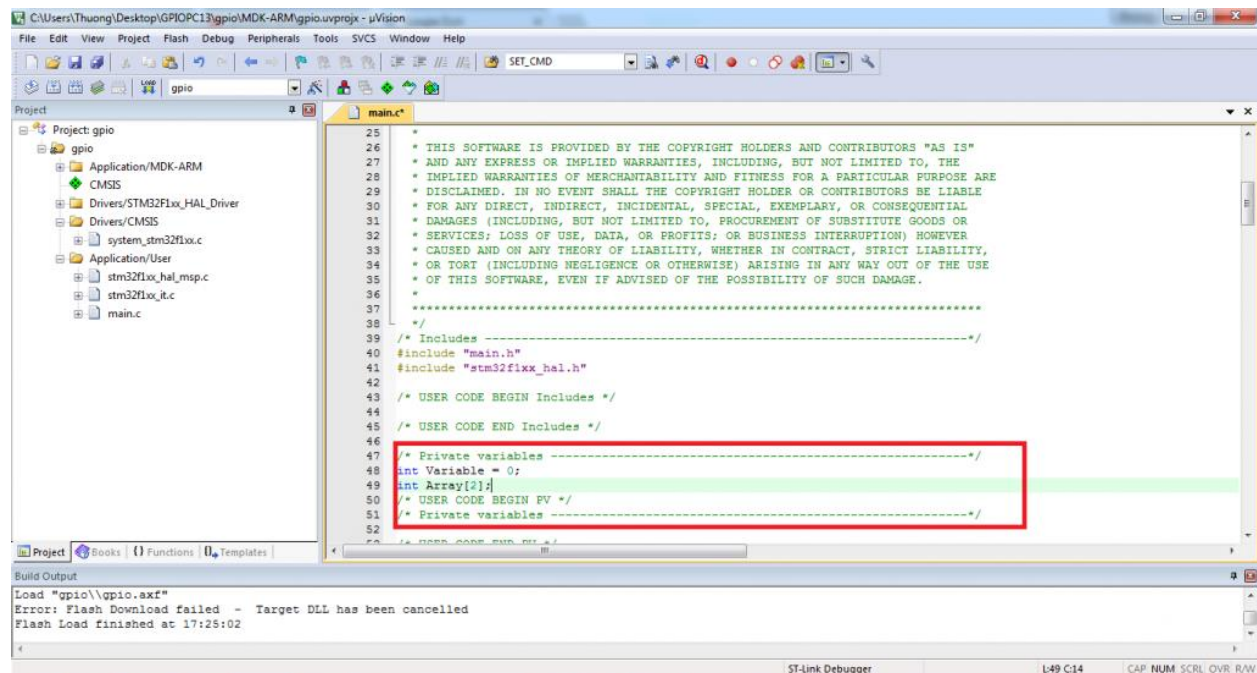
**Step Out:** Thực thi cho đến trả về hàm hiện tại

## 1/ Thực hành debug nhấp nháy led ở chân PC13 và quan sát sự thay đổi của thanh ghi, các biến và mảng ở kiểu dữ liệu bất kì:

Sau khi sinh code ở CubeMX và mở project trên KeilC

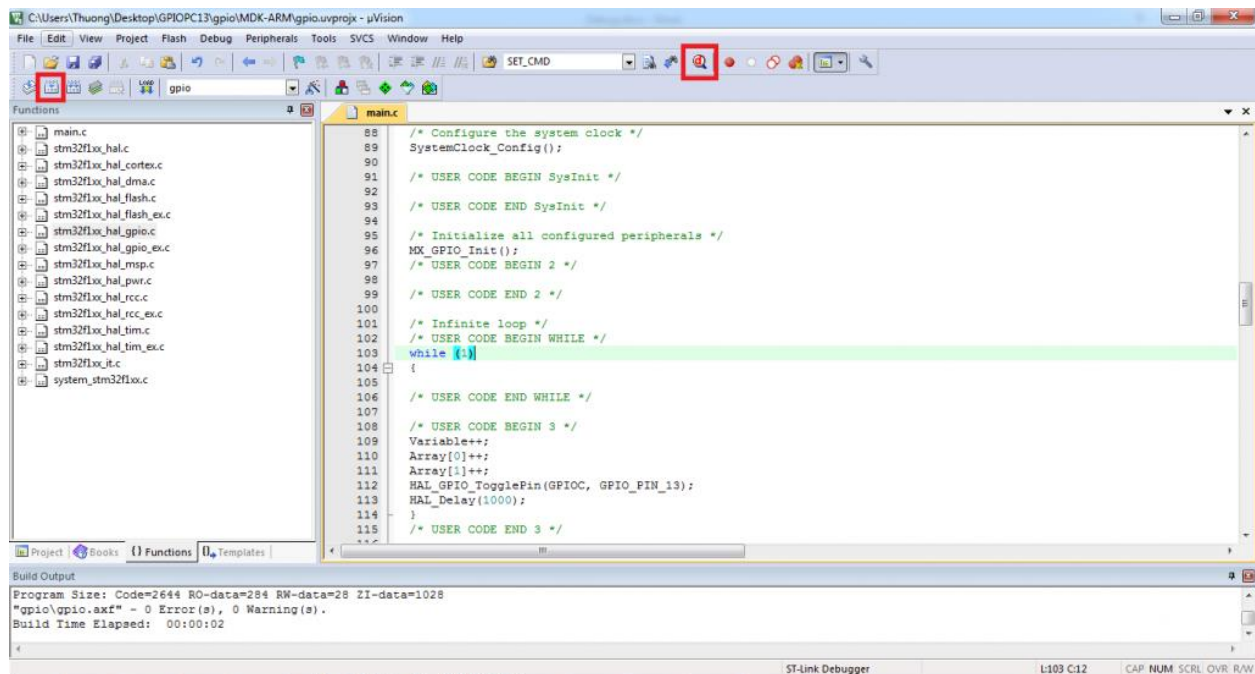
### Bước 1:

Khởi tạo một biến *Variable* kiểu và một mảng Array gồm 2 phần tử kiểu dữ liệu *integer* để quan sát sự thay đổi trong hàm main.



### Bước 2:

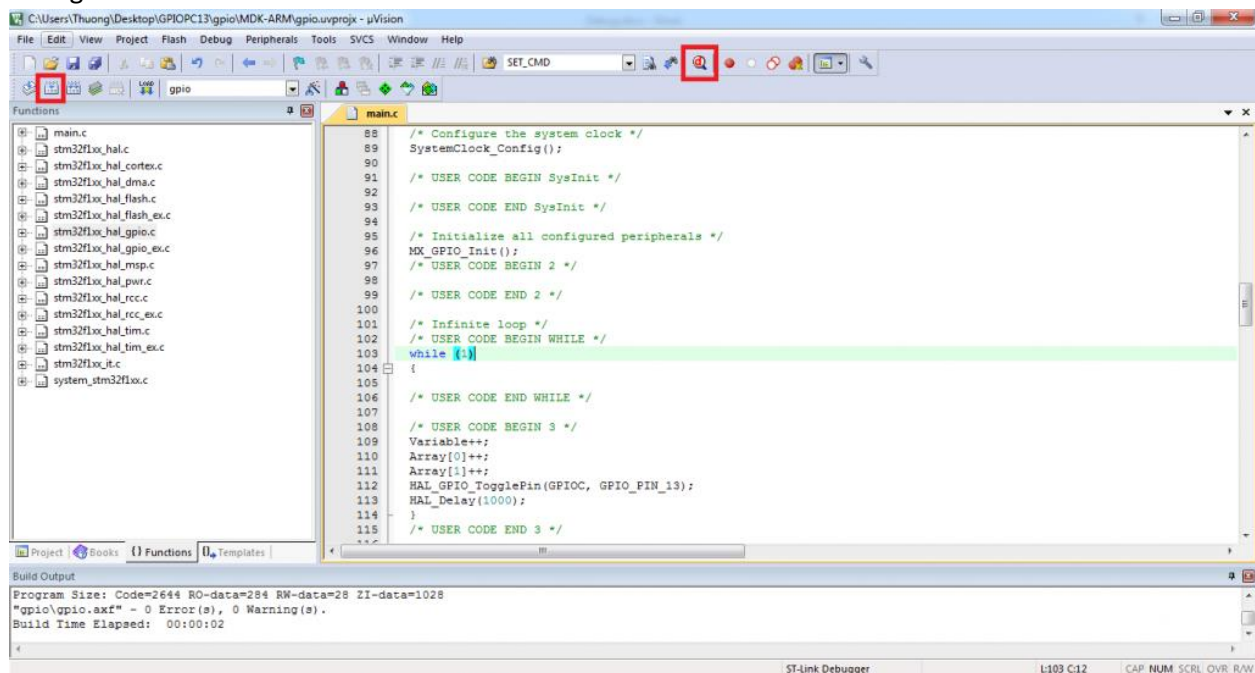
- Tăng giá trị của biến Variable cùng với những phần tử trong mảng Array lên 1
- Đảo trạng thái led PC13 với tần số 1s 1 lần.



#### Bước 4:

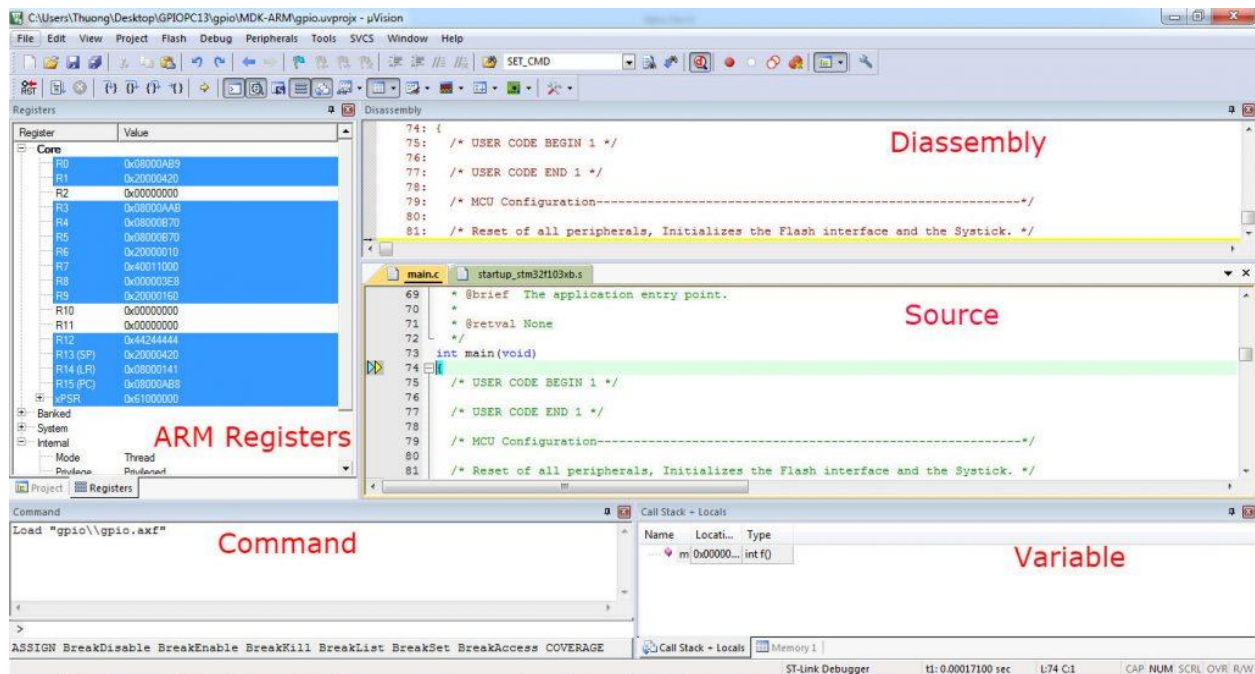
- Biên dịch chương trình
- Nhấn vào nút debug

**Chú ý:** Cứ mỗi lần có thay đổi gì ở source thì nên biên dịch lại rồi mới nạp hoặc biên dịch lại code rồi mới debug.



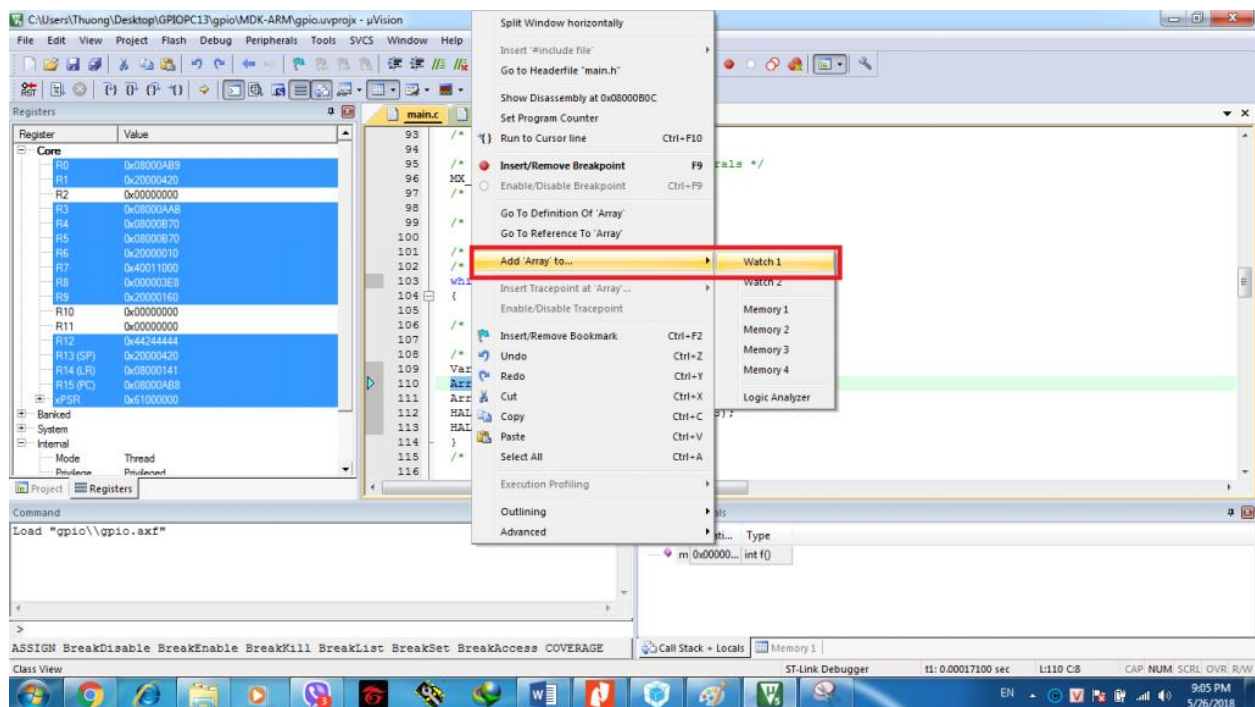
#### Bước 5:

Hiển thị giao diện debug và một số chức năng của window:



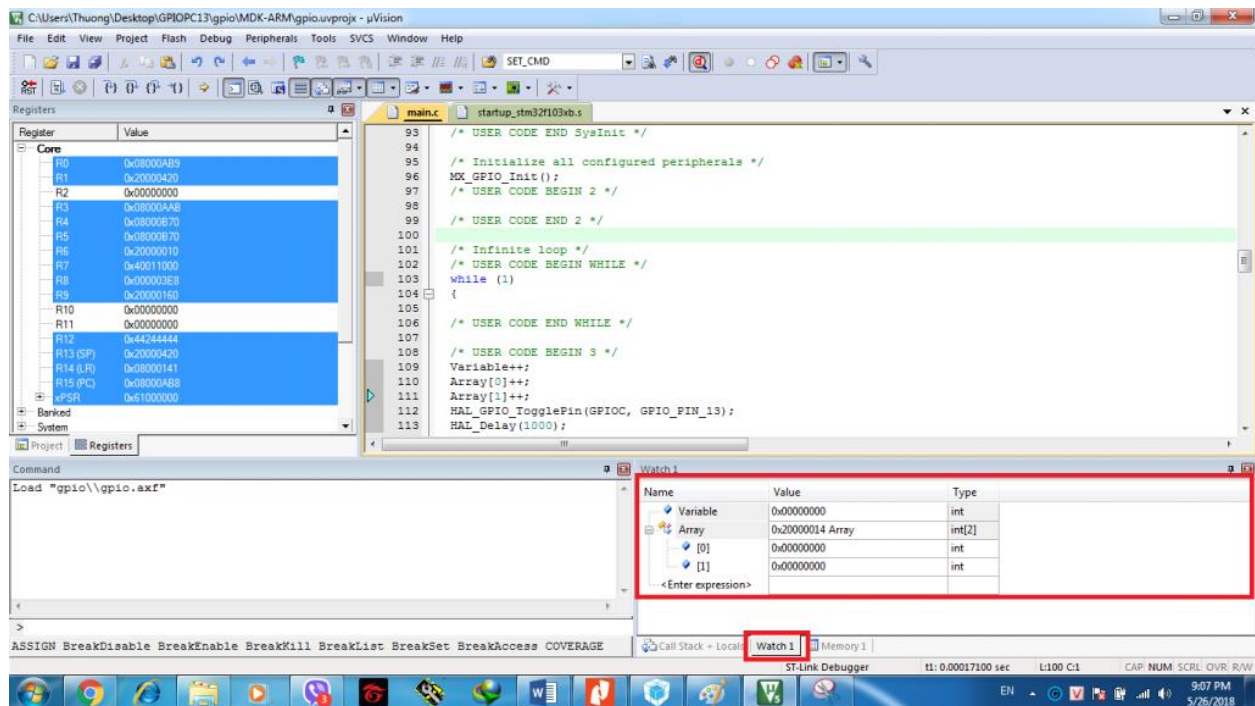
#### Bước 6:

- Bôi đen biến Variable
- Nhấn chuột phải và chọn Add "Variable" vào cửa sổ Watch1 để theo dõi sự thay đổi của biến
- Tương tự ta thực hiện cho các phần tử mảng Array



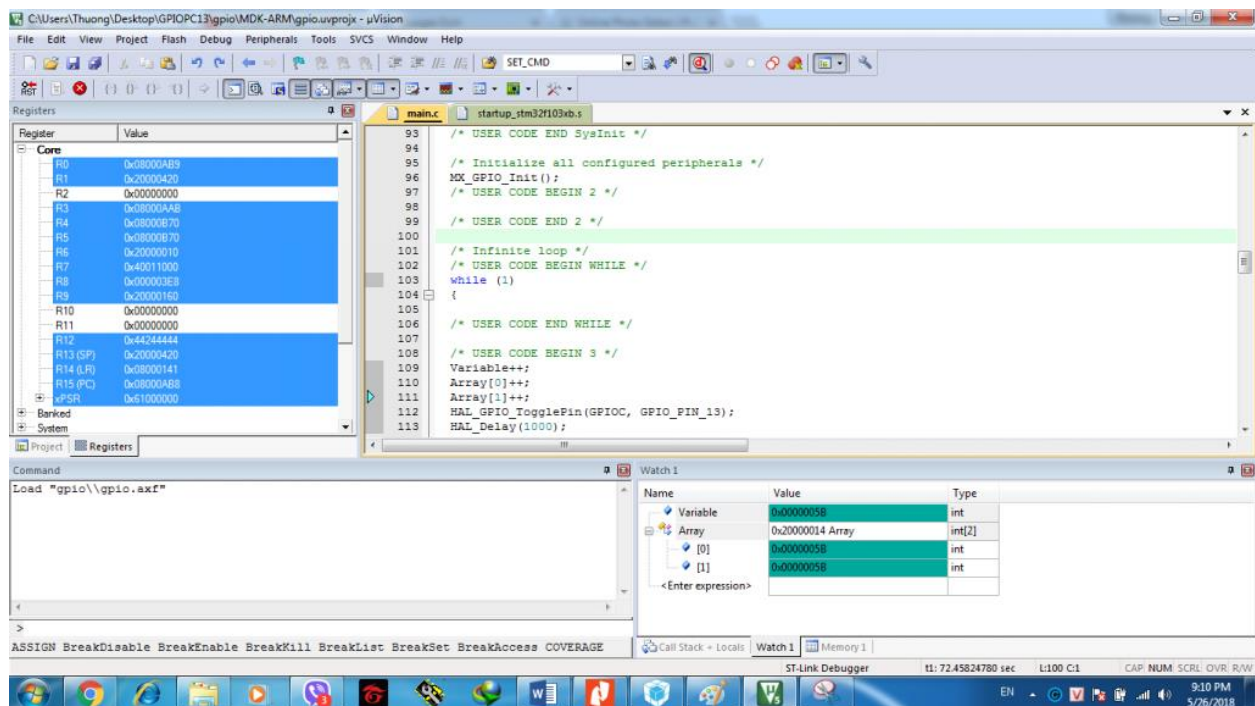
#### Bước 7:

Sau khi Add các biến vào Watch 1, chọn vào ô cửa sổ để quan sát các biến. Ở đây, chúng ta thấy các biến đã được Add vào Watch 1



#### Bước 8:

Quan sát được sự thay đổi của các biến, tương ứng với 1s nó sẽ tăng lên 1 lần.

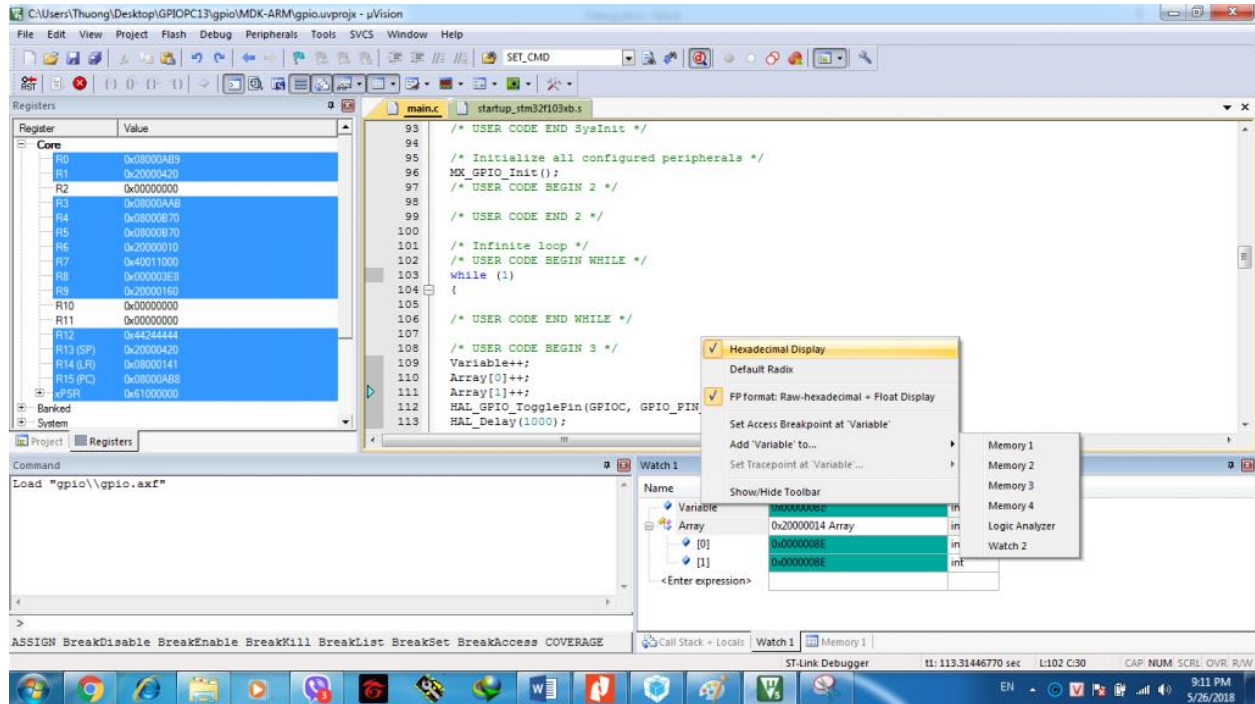


#### Bước 9:



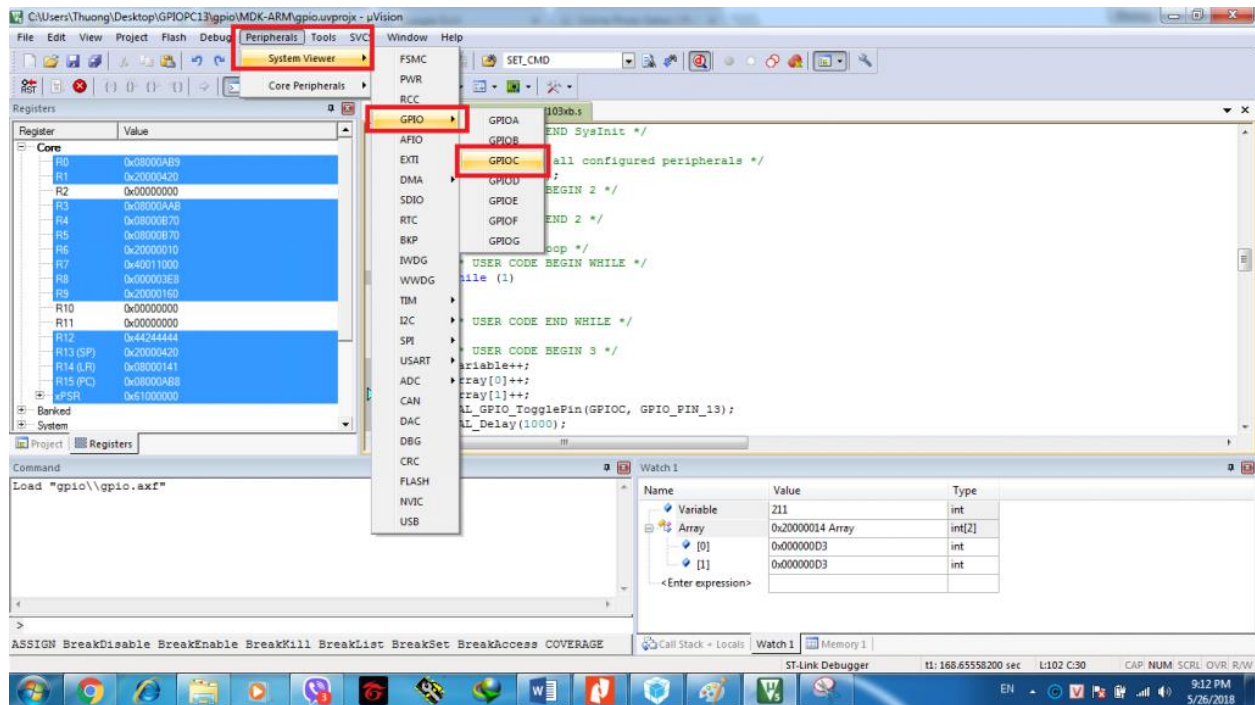
Quan sát biến ở dạng thập phân:

- Nhấn chuột phải vào biến
- Bỏ chọn *Hexadecimal Display*



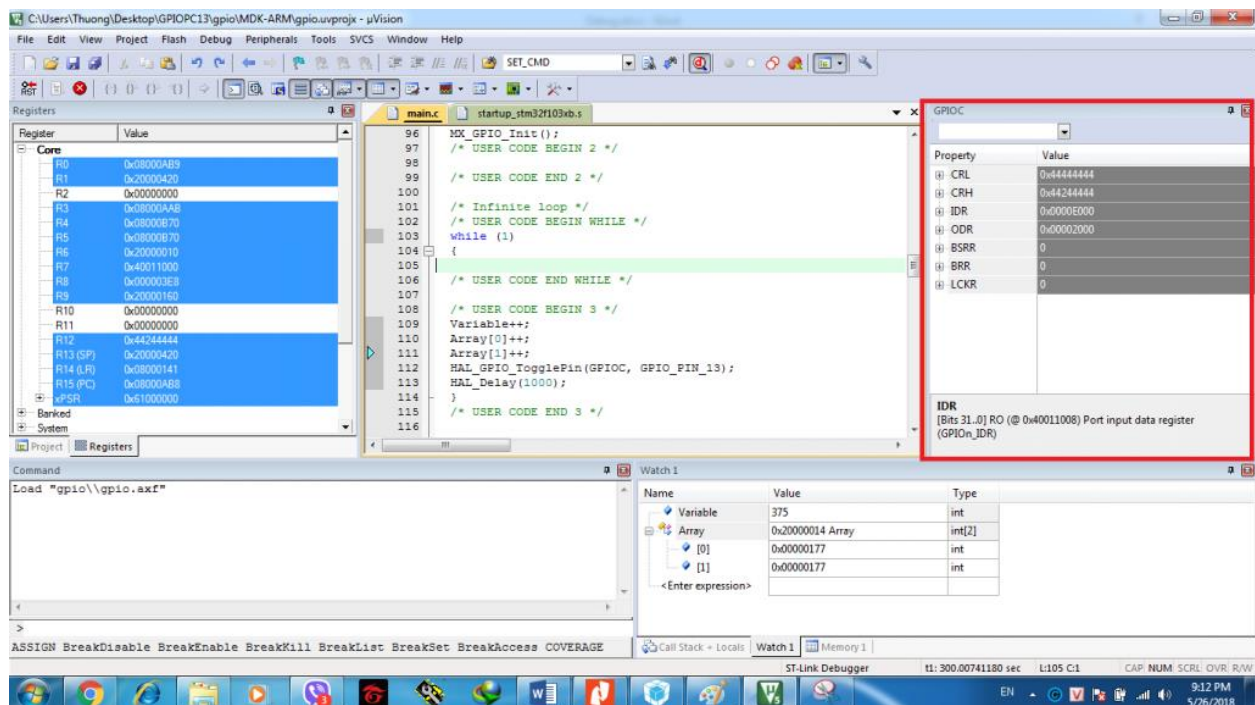
**Bước 10:**

- Vào Peripheral và chọn System Viewer
- Theo dõi các bit thay đổi trên những thanh ghi này

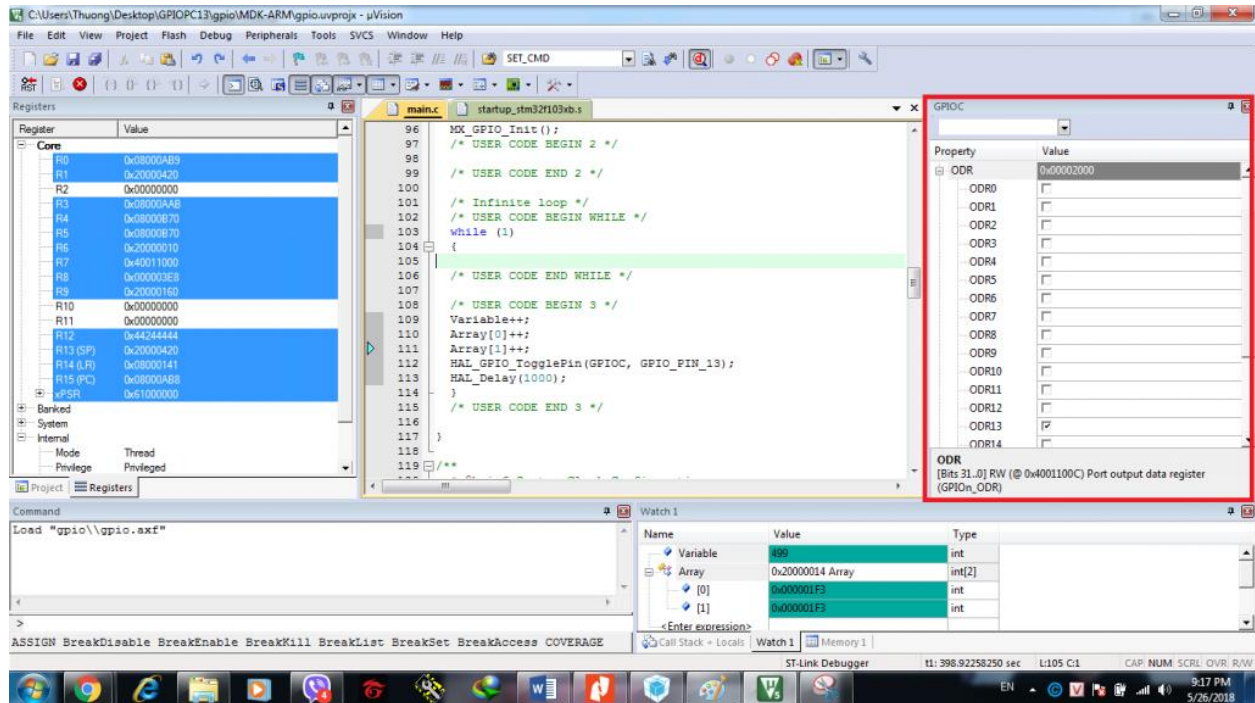


## Bước 11:

Sau đó, nó sẽ hiện ra các thanh ghi của ngoại vi IO ở Port C.



Ở đây chúng ta sẽ thấy bit ODR13 thanh ghi Output Data Register ở bit 13 sẽ thay đổi:



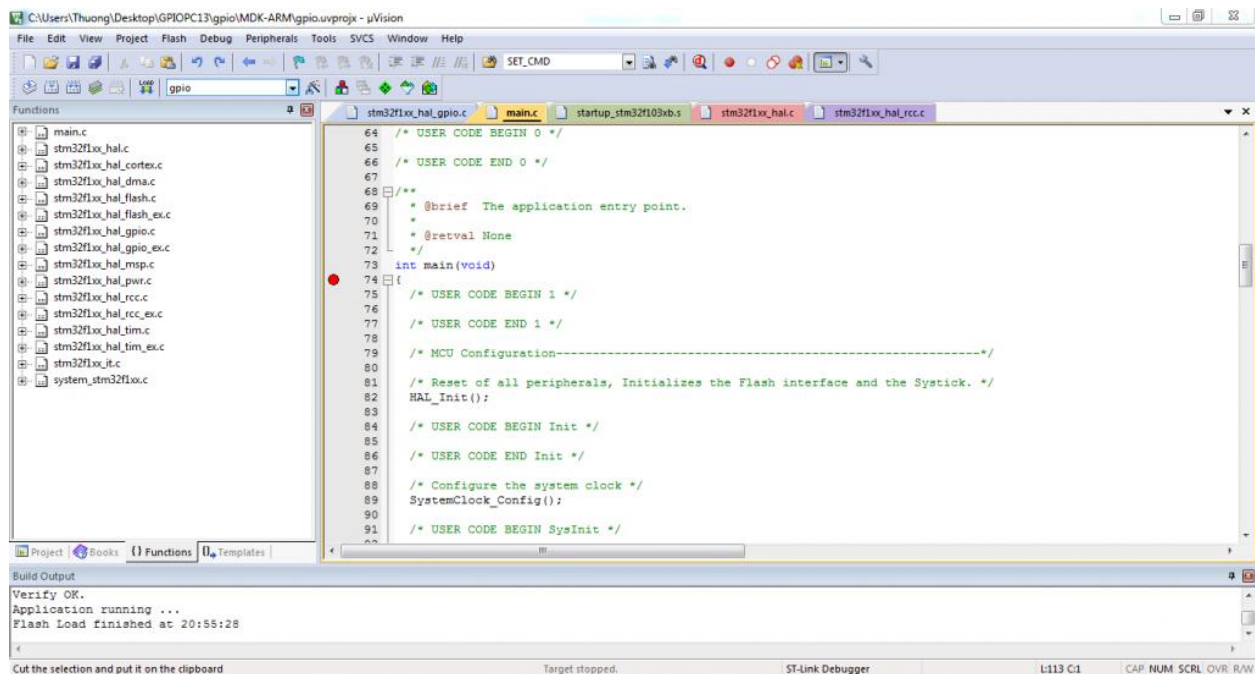
## 2/ Dùng breakpoint để debug:

**Breakpoint** là những điểm (vị trí) trong chương trình được người lập trình chọn, tại điểm đó chương trình thực thi sẽ dừng lại, breakpoint được sử dụng trong debug mode. Breakpoint rất hữu hiệu trong việc phát triển chương trình và gỡ lỗi.

- Sử dụng breakpoint để kiểm tra chương trình của bạn có thực hiện được đến câu lệnh mà bạn đặt breakpoint hay không. Nếu chương trình bạn chạy mãi không dừng, thì chứng tỏ lệnh tại breakpoint không bao giờ được thực hiện. Như vậy chương trình của bạn đã có lỗi.
- Sử dụng breakpoint để theo dõi, kiểm tra và thay đổi giá trị các biến, bộ nhớ, các thanh ghi ngoại vi, thanh ghi CPU, thanh ghi ngoại vi tại thời điểm dừng câu lệnh để quan sát chương trình chạy có đúng hay không, xem xét các giá trị có theo dự kiến hoặc để tác động trực tiếp vào chương trình.
- Sử dụng breakpoint tạm dừng chương trình để người sử dụng tương tác vào thông qua các ngoại vi, từ đó quan sát luồng chương trình có thực hiện đúng hay không sau khi tác động.

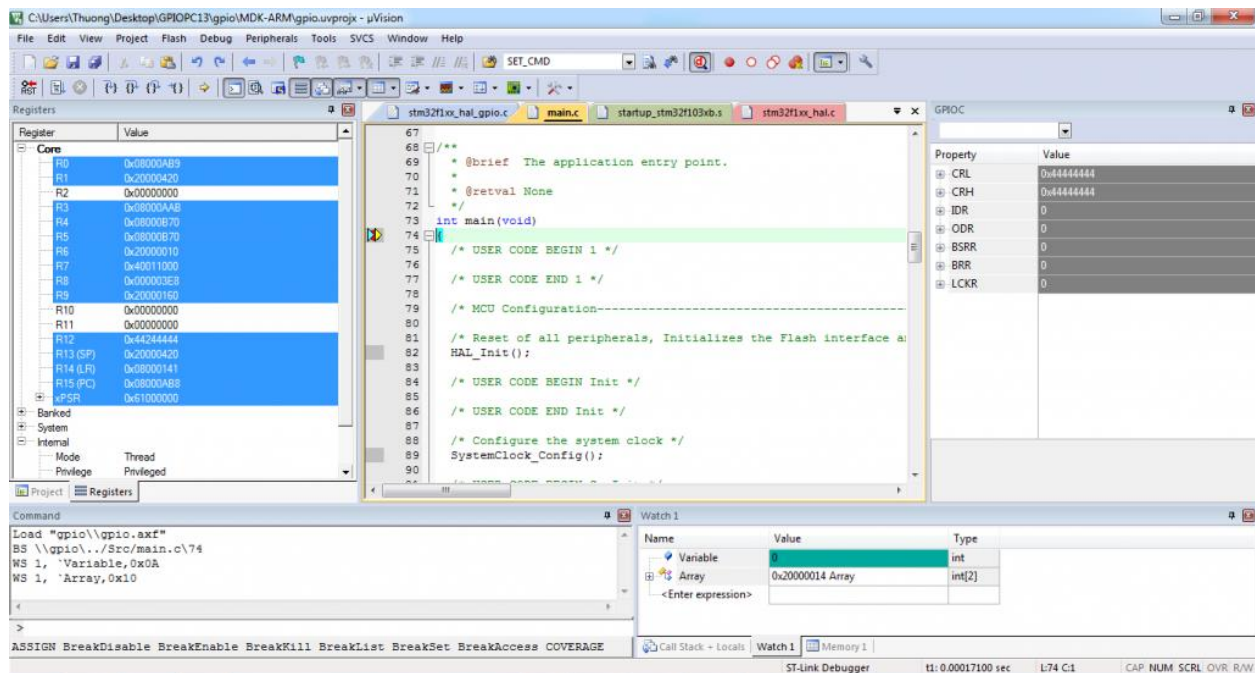
### Bước 1:

- Nhấn chuột chọn vị trí đặt breakpoint
- Nhấn biểu tượng trên thanh công cụ để thiết lập breakpoint. Ở đây, chúng ta đặt breakpoint khi vào hàm main.c
- Bật chế độ debug



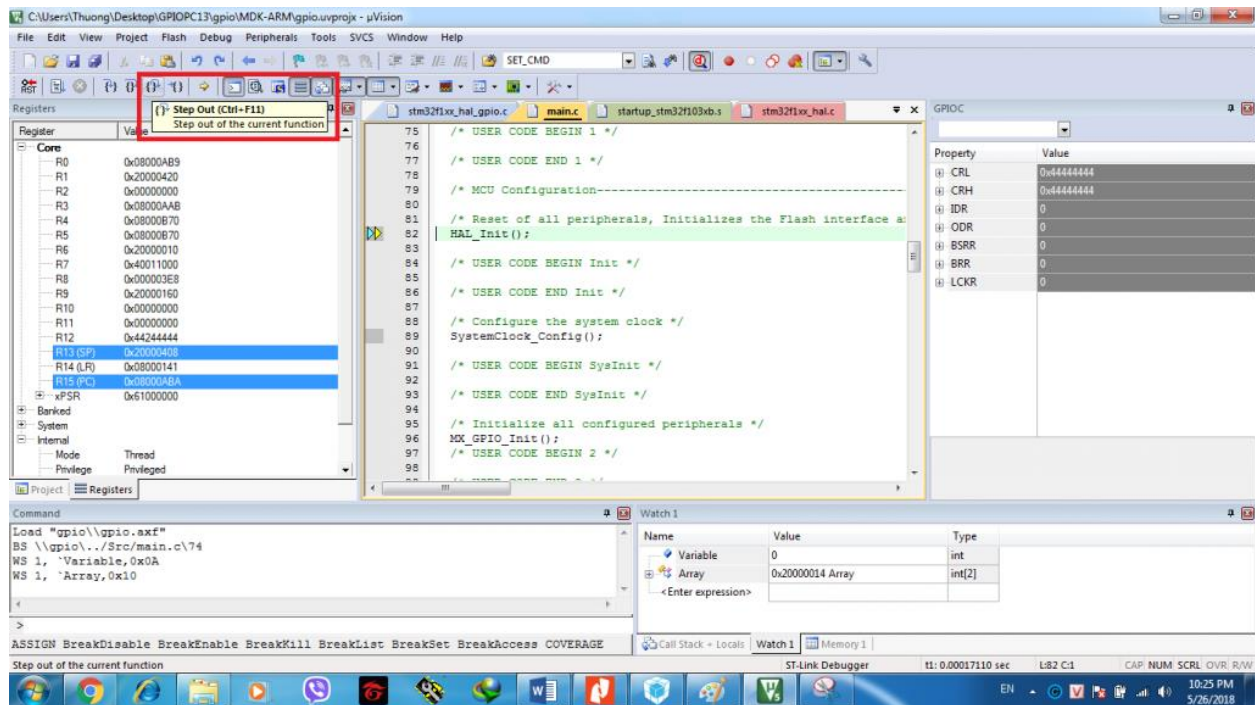
## Bước 2:

Sau khi bật debug thì chương trình đã dừng tại dòng 74



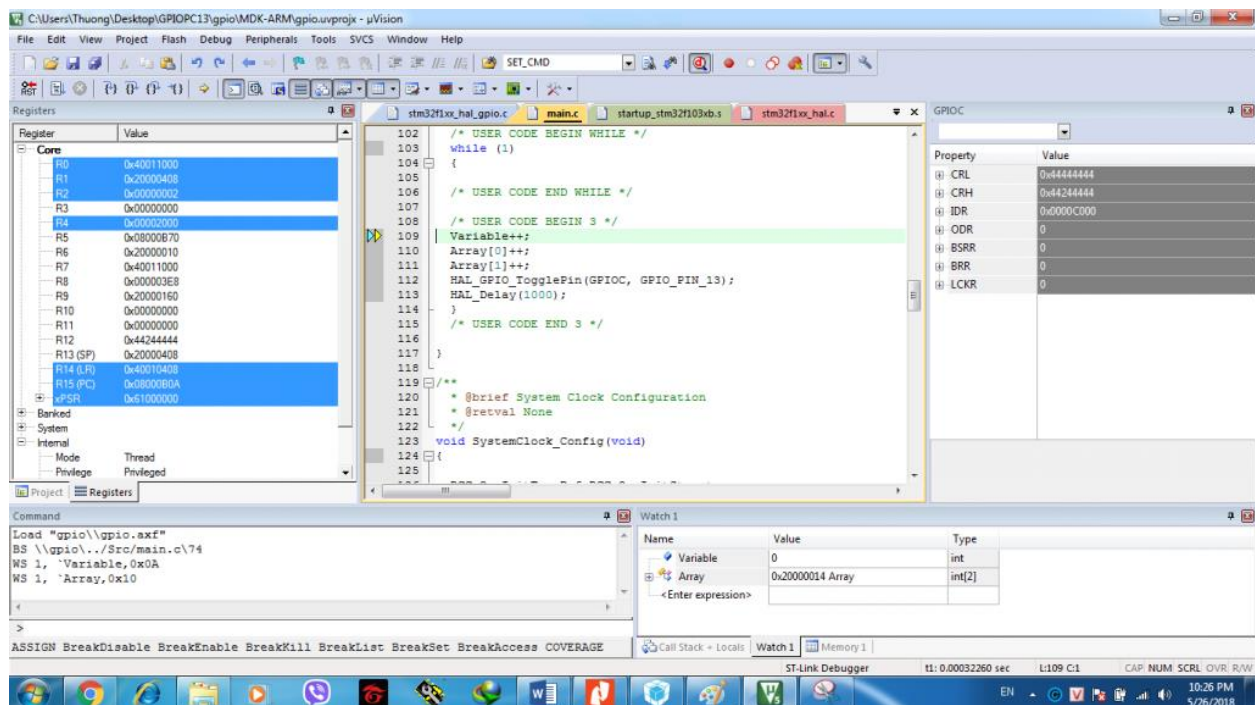
Chúng ta sử dụng *Step out* để bỏ qua thực thi các hàm cấu hình trước khi vào hàm *while* để theo dõi các biến





### Bước 3:

Chương trình sau khi vào vòng while



- Thực thi từng câu lệnh để quan sát từng sự thay đổi ở các biến bằng Step (F11)
- Quan sát sự thay đổi sau khi thực thi từng câu lệnh