

HƯỚNG DẪN SỬ DỤNG BOARD GRAPE32 UNLEASHED VÀ GRAPINI32

(dành cho Windows)

1. Giới thiệu

Tài liệu này sẽ hướng dẫn bạn sử dụng hai board Grape32 Unleashed và Grapini32 do Grapetech phát triển. Nội chính của tài liệu gồm:

- Giới thiệu phần mềm.
- Hướng dẫn tạo một project.
- Hướng dẫn biên dịch (build/compile) và dò lỗi (debug).

2. Giới thiệu phần mềm

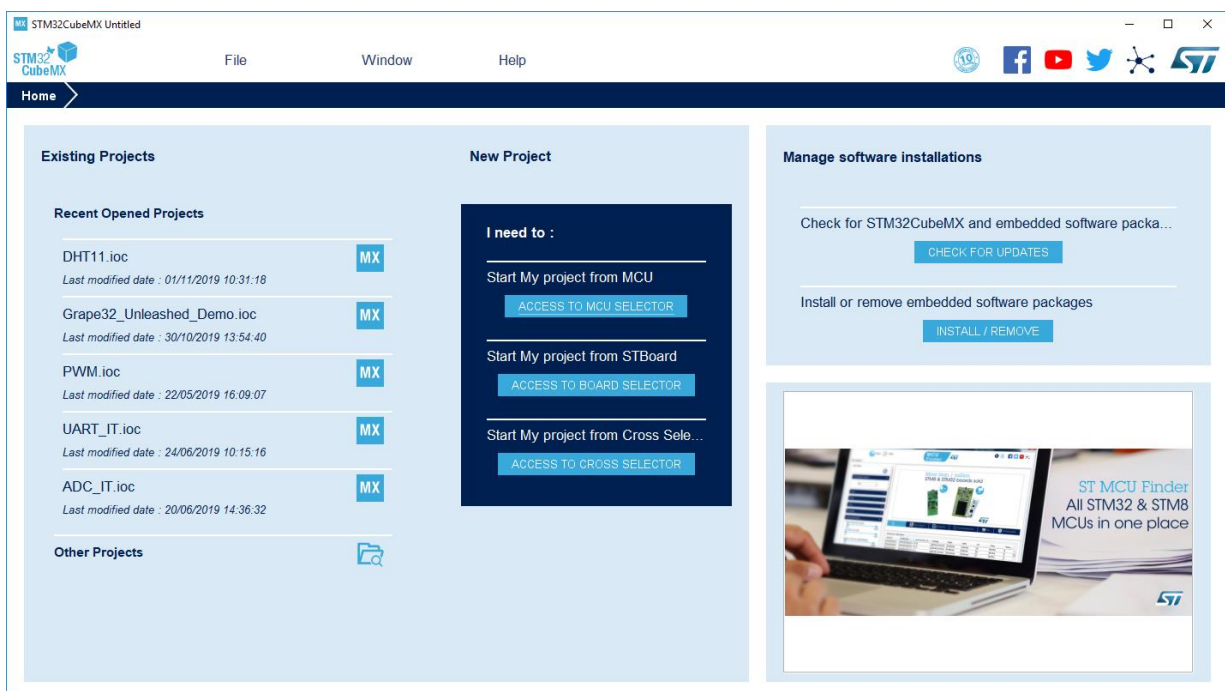
2.1. STM32CubeMX

Để cấu hình cho vi điều khiển (VĐK) một cách trực quan nhất, chúng tôi khuyến khích bạn sử dụng phần mềm STM32CubeMX.

Link tải phần mềm STM32CubeMX:

<https://www.st.com/en/development-tools/stm32cubemx.html>

Giao diện chính của phần mềm:



Hình 2.1. Giao diện khởi động phần mềm STM32CubeMX.

2.2. Atollic TrueSTUDIO

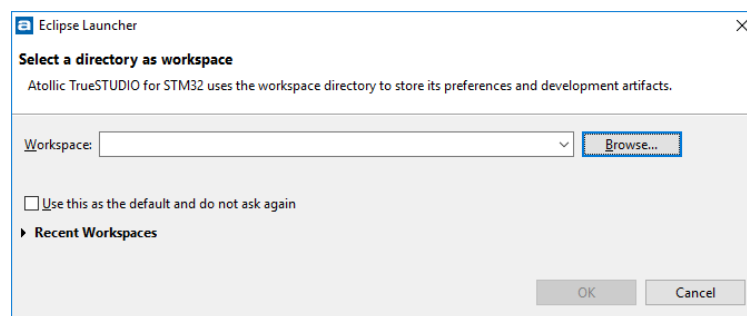
Atollic TrueSTUDIO là một phần mềm hỗ trợ cho việc lập trình, gỡ lỗi, phát triển ứng dụng trên VĐK STM32. Một số ưu điểm của TrueSTUDIO:

- Dựa trên nền những thành phần mã nguồn mở như phần mềm ECLIPSE và GNU.
- Có đa dạng các tính năng quản lý code và phân tích hệ thống.
- Dễ dàng đi sâu vào quản lý phần cứng, tương tác giữa phần cứng và phần mềm.
- Ngoài ra còn nhiều tính năng hữu ích đang chờ bạn khám phá.

Link tải phần mềm Atollic TrueSTUDIO:

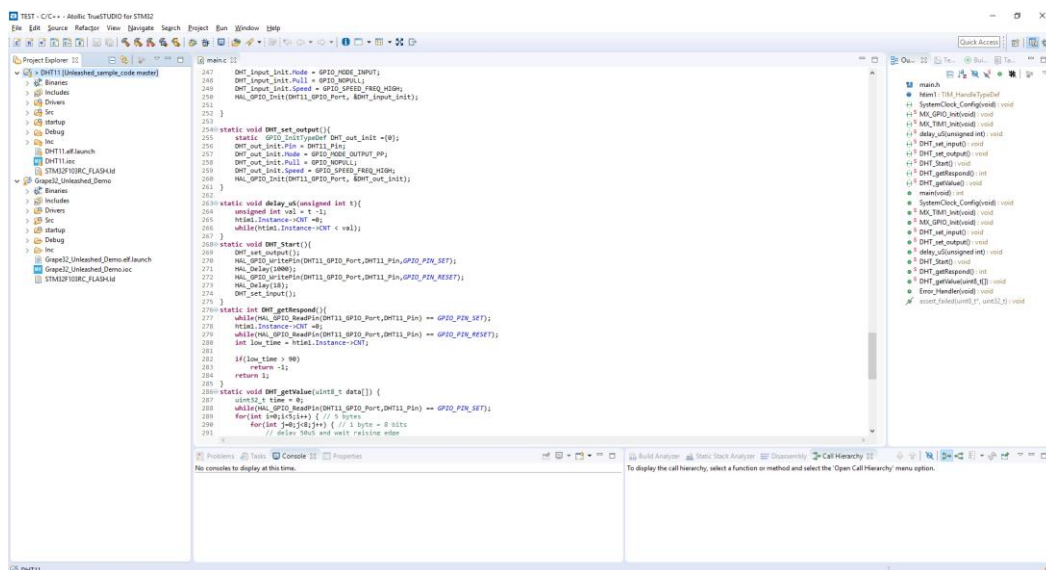
<https://www.st.com/en/development-tools/truestudio.html#get-software>

Khi vừa cài đặt, TrueSTUDIO sẽ yêu cầu bạn chọn đường dẫn của workspace, nơi bạn dùng để quản lý các project bạn thực hiện. Bạn có thể tạo workspace bất kỳ trên máy tính của mình và nhấn OK để tiếp tục.



Hình 2.2. Chọn đường dẫn workspace.

Dưới đây là giao diện chính của phần mềm khi vừa khởi động vào (workspace đã tồn tại project mở sẵn).



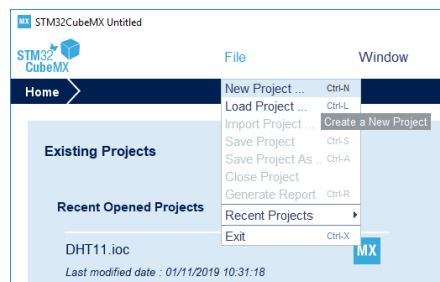
Hình 2.3. Giao diện chính của Atollic TrueSTUDIO.

3. Hướng dẫn tạo một project

Trong phần này, chúng tôi sẽ hướng dẫn bạn các bước tạo một project bước đầu trước khi bắt tay vào xây dựng và phát triển ứng dụng trên board Grape32 Unleashed hoặc Grapini32.

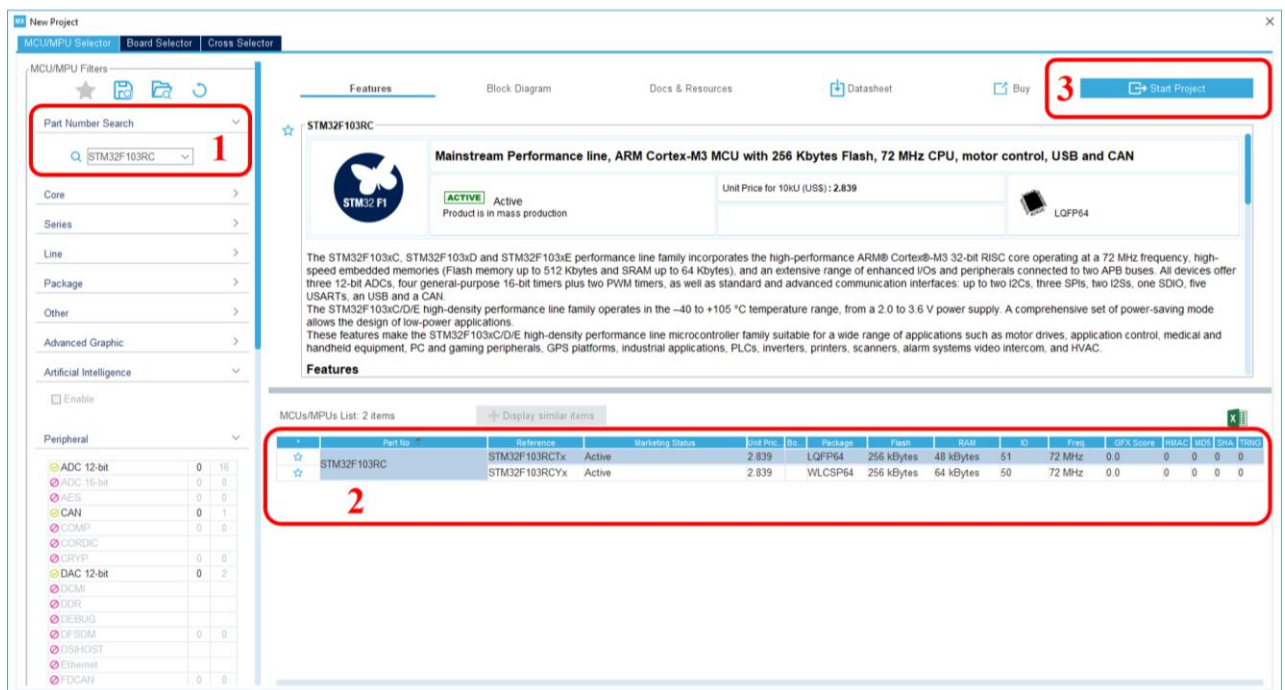
3.1. Cấu hình

Để thuận tiện cho việc cấu hình VDK trên board, chúng ta sẽ bắt đầu với STM32CubeMX trước. Đầu tiên là tạo một project CubeMX. Bạn có thể nhấn tổ hợp phím **Ctrl+N** hoặc chọn **File > New Project...** như Hình 3.1.



Hình 3.1. Tạo project CubeMX mới.

Phần mềm sẽ tự động kết nối đến server để tải những nội dung cần thiết do việc tạo một project mới. Sau khi kết nối xong sẽ ra được giao diện lựa chọn VDK như Hình 3.2. Ở đây, bạn có thể gõ tên VDK bạn đang dùng vào ô **Part Number Search** (vùng số 1) hoặc tìm nó trong danh sách có sẵn (vùng số 2). VDK sử dụng trên board Grape32 Unleashed và Grapini32 là **STM32F103RCT6**. Sau khi tìm và chọn đúng tên VDK, bạn chọn **Start Project** (vùng số 3).



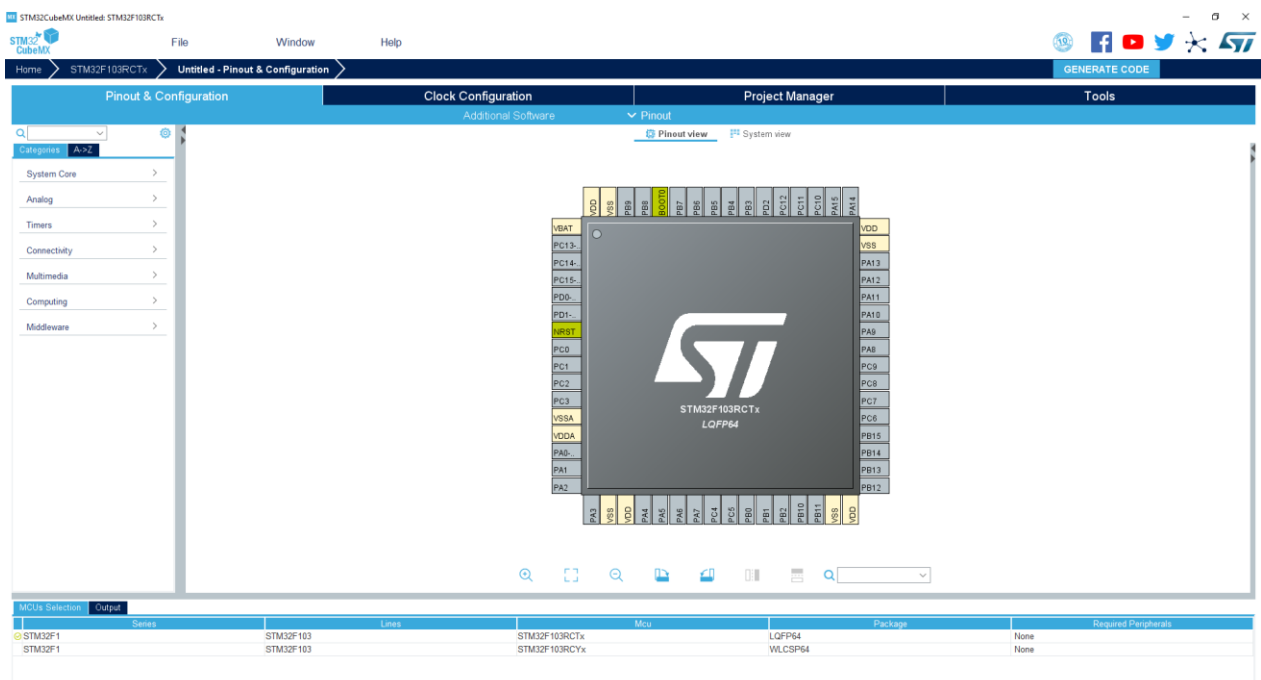
Hình 3.2. Giao diện lựa chọn VDK.

Sau khi nhấn **Start Project**, phần mềm sẽ dẫn chúng ta đến giao diện cấu hình vô cùng trực quan. Nội dung phần này chỉ là hướng dẫn bạn tạo một project đơn giản, nên chúng ta sẽ không đi chi tiết vào phần cấu hình (sẽ được đề cập ở những phần sau). Các bạn chỉ cần để mặc định các thông số là được.

Giao diện cấu hình này gồm 4 tab chính – **Pinout & Configuration**, **Clock Configuration**, **Project Manager** và **Tools**. Sau đây sẽ là những **thiết lập bắt buộc** cho từng tab để chương trình có thể hoạt động trên board Grape32 Unleashed và Grapini32.

3.1.1. Pinout & Configuration

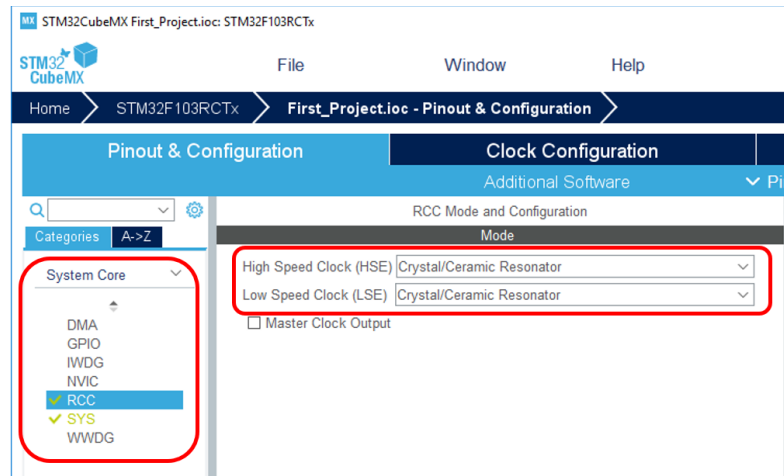
Tab **Pinout & Configuration** cho phép người dùng thiết lập và cấu hình các chân của VDK tùy vào yêu cầu sử dụng.



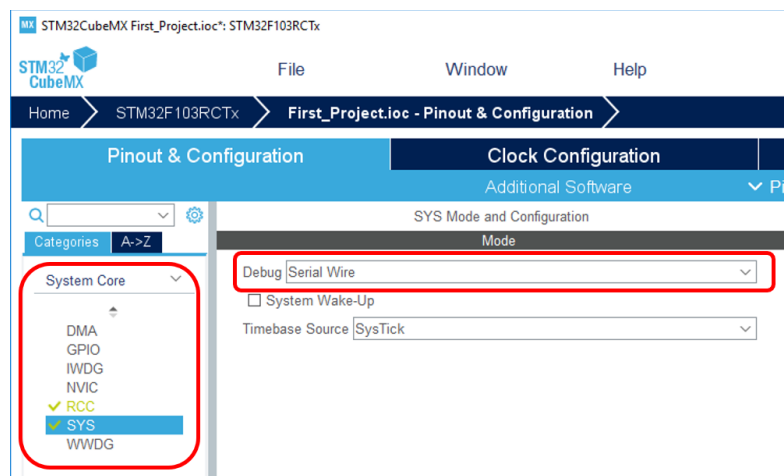
Hình 3.3. Pinout & Configuration.

Lưu ý quan trọng – Bạn cần thực hiện một số thiết lập như sau:

- **Thiết lập RCC** (Hình 3.4): Lựa chọn 2 loại clock tốc độ cao (HSE) và tốc độ thấp (LSE) là **Crystal/Ceramic Resonator** (Thạch anh dao động đã tích hợp sẵn trên board sẵn trên board).
- **Thiết lập SYS** (Hình 3.5): Lựa chọn **Debug** là **Serial Wire**.



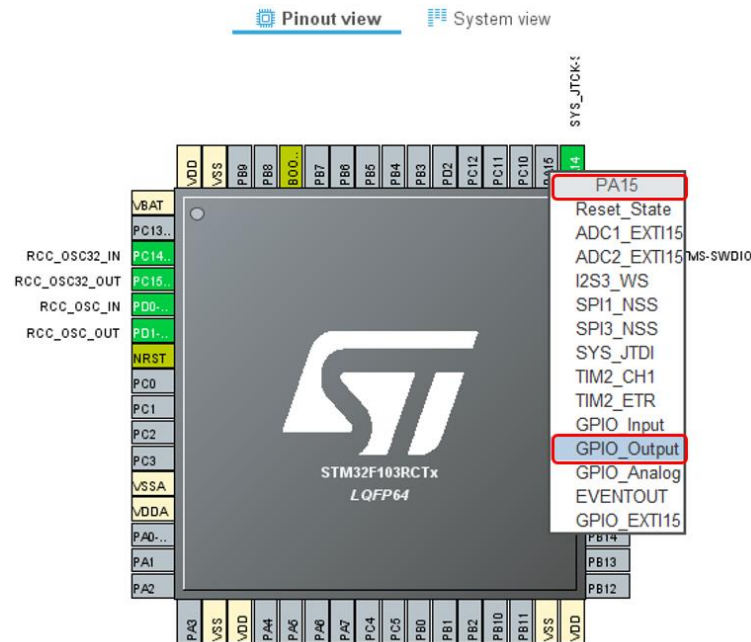
Hình 3.4. Thiết lập RCC.



Hình 3.5. Thiết lập SYS.

Trong bài hướng dẫn này, chúng ta sẽ không đi chi tiết vào giải thích các vấn đề kỹ thuật nên chỉ cần thực hiện một project đơn giản như bật, tắt các LED trên board. Bạn chỉ cần thực hiện theo các bước dưới đây:

- Bước 1: Nhấn chuột trái vào chân PA15 > Chọn **GPIO_Output** (Hình 3.6) > Mục **GPIO Mode and Configuration** sẽ xuất hiện chân vừa chọn, nhấn vào đó và thiết lập như Hình 3.7.
- Bước 2: Thực hiện tương tự đối với các chân **PB10, PB3, PB12, PB13, PB14, PB15** sao cho kết quả thu được như Hình 3.8. Lưu ý là **vùng được khoanh đỏ** là những vùng mà bạn cần chú ý thay đổi thiết lập nếu cần.



Hình 3.6. Pinout view.

GPIO Mode and Configuration

Configuration

☐ Group By Peripherals

☒ GPIO ☒ RCC ☒ SYS

Search Signals

☐ Show only Modified Pins

Pin N...	Signal o...	GPIO ou...	GPIO m...	GPIO Pu...	Maximu...	User Label	Modified
PA15	n/a	High	Output P...	No pull-u...	Low	LED_GR...	<input checked="" type="checkbox"/>

PA15 Configuration :

GPIO output level:

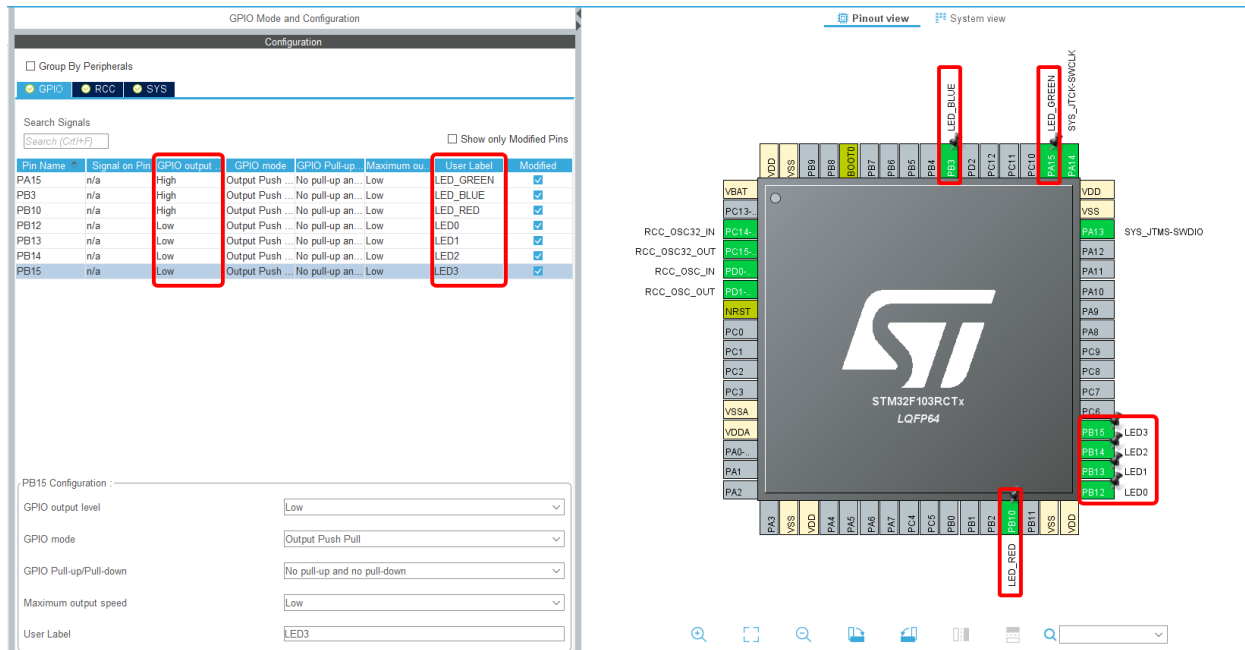
GPIO mode:

GPIO Pull-up/Pull-down:

Maximum output speed:

User Label:

Hình 3.7. GPIO Mode and Configuration.



Hình 3.8. Cấu hình LED GPIO.

3.1.2. Clock Configuration

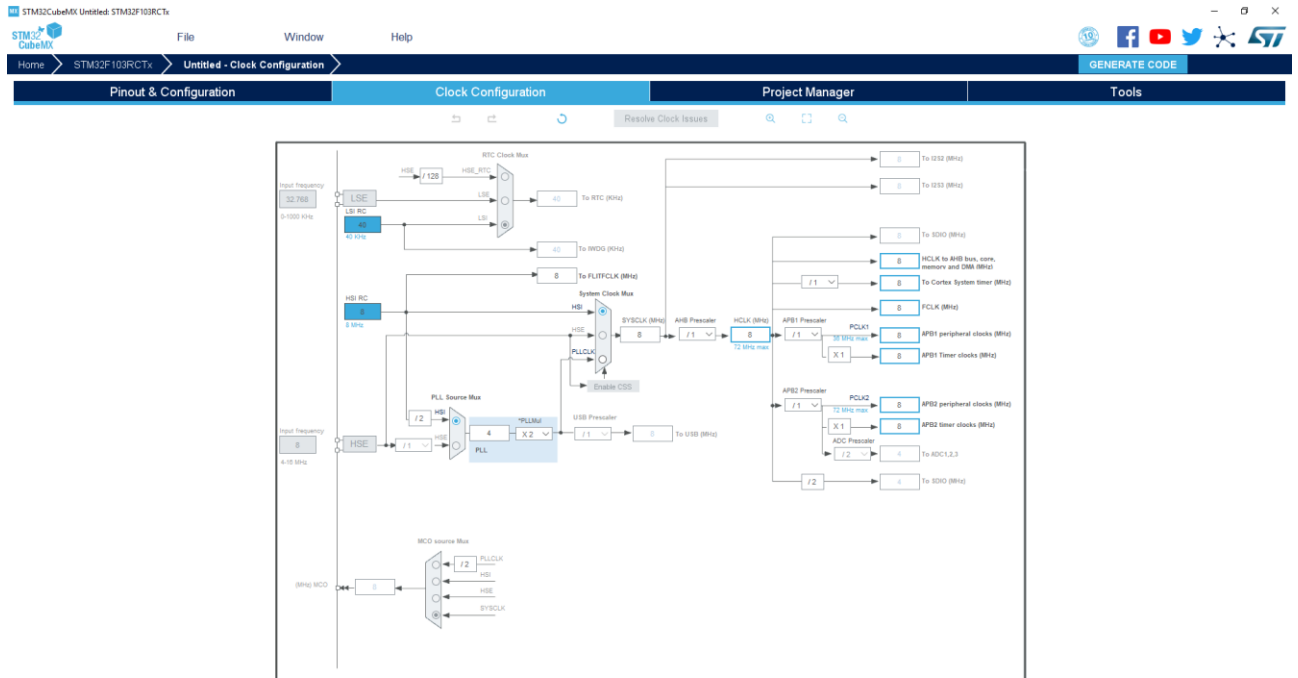
Tab **Clock Configuration** (Hình 3.9) cho phép người dùng thiết lập tần số hoạt động cho từng thành phần của VDK. Tab này bạn cứ giữ nguyên thiết lập mặc định.

3.1.3. Project Manager

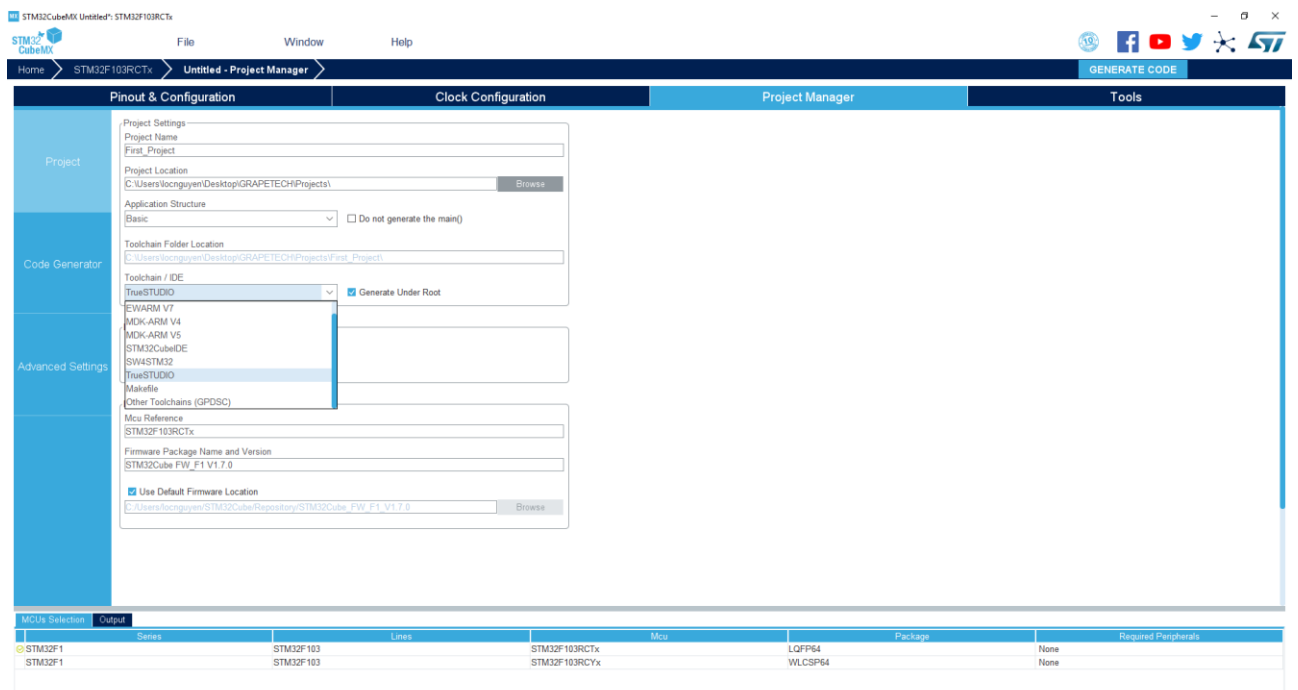
Ở tab **Project Manager** (Hình 3.10), người dùng sẽ thiết lập một số thông tin liên quan đến project. Chúng ta chỉ cần quan tâm đến các thiết lập sau:

- **Project Name:** Khai báo tên project.
- **Project Location:** Khai báo đường dẫn nơi lưu project.
- **Toolchain / IDE*:** Lựa chọn phần mềm - môi trường (IDE) phát triển ứng dụng mà bạn sử dụng.

* Bạn có thể chọn sử dụng bất kỳ phần mềm lập trình nào có trong danh sách mà bạn muốn. Ở đây, chúng tôi chọn sử dụng TrueSTUDIO để hướng dẫn.



Hình 3.9. Clock Configuration.



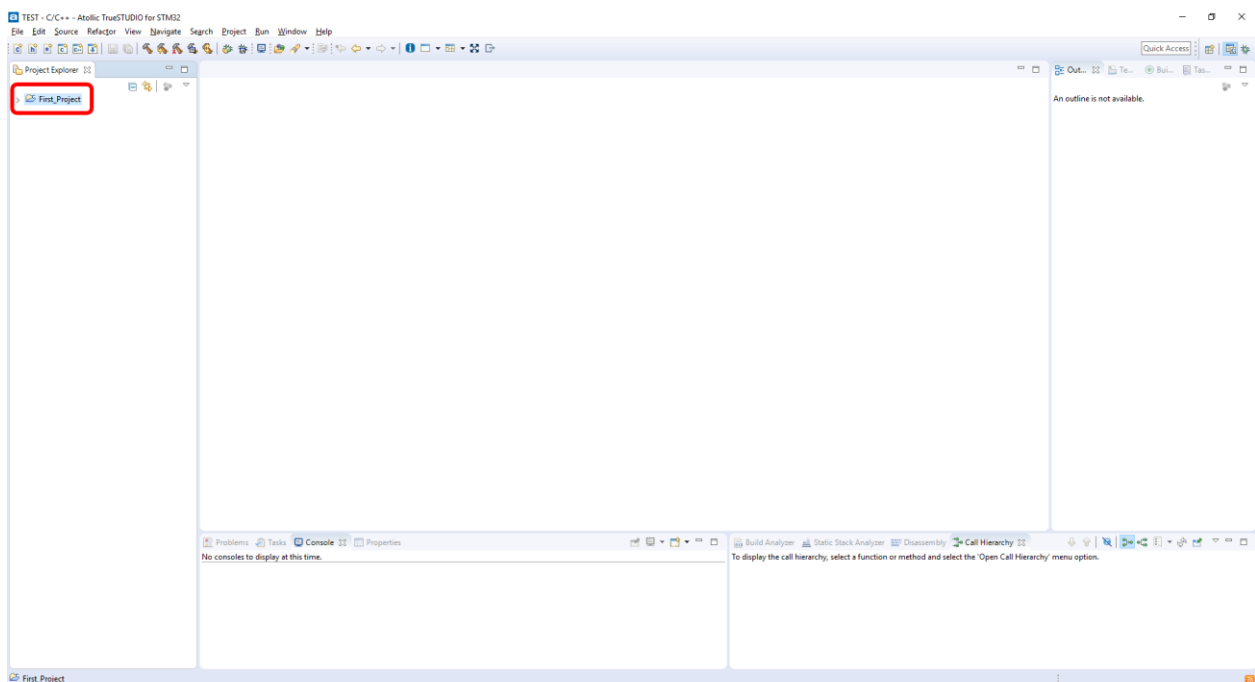
Hình 3.10. Project Manager.

3.1.4. Tools

Tab **Tools** sẽ chứa những công cụ hỗ trợ liên quan đến VĐK. Chúng ta không cần phải quan tâm đến tab này.

3.2. Tạo project

Sau khi cấu hình xong mọi thứ, các bạn chọn **GENERATE CODE** ở góc trên bên phải chương trình. Cửa sổ **Code Generation** sẽ xuất hiện để thông báo rằng việc xuất nội dung code theo như bạn đã cấu hình trước đó đã hoàn tất. Bạn có những tùy chọn như **Open Folder** – mở thư mục chứa project, **Open Project** – mở trực tiếp project vừa tạo ra (tùy vào **Toolchain / IDE** mà bạn chọn, phần mềm tương ứng sẽ dùng để mở project đó lên), **Close** để tắt cửa sổ **Code Generation**. Ở đây, chúng ta chọn **Open Project**. Phần mềm TrueSTUDIO sẽ được khởi động để mở project bạn vừa tạo lên (Hình 3.11). Project của chúng ta sẽ nằm trong cửa sổ **Project Explorer**.



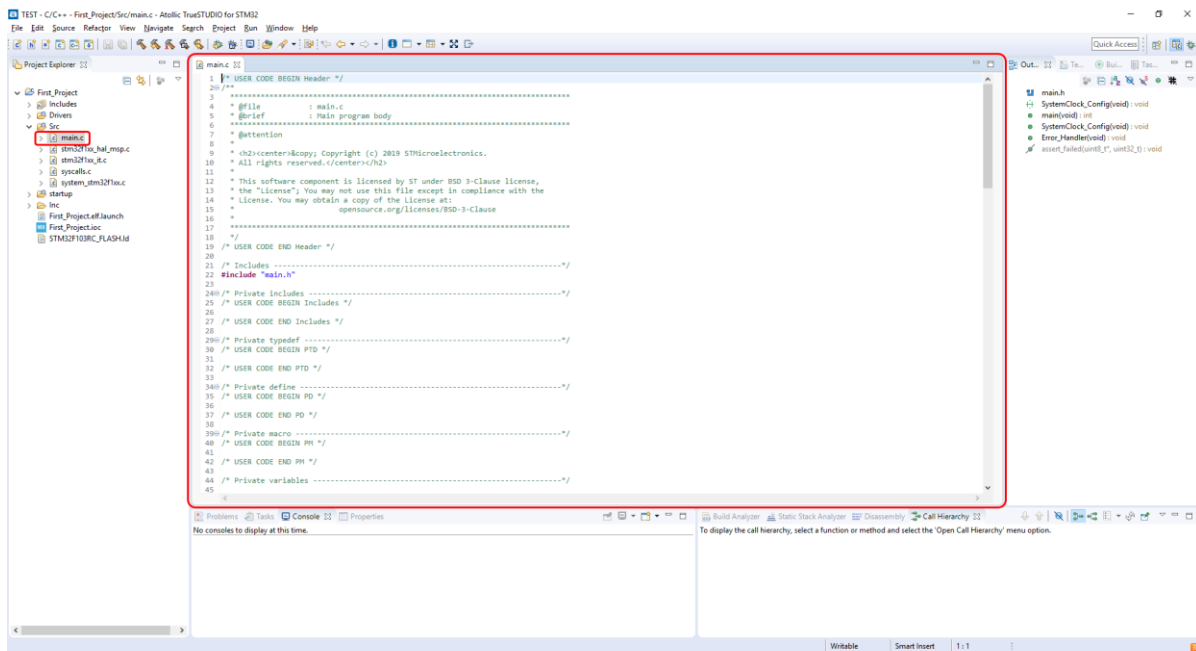
Hình 3.11. Mở project bằng TrueSTUDIO.

Mở rộng thẻ project ra, ta sẽ xem được các thành phần tạo nên project. Nội dung của chương trình chính của chúng ta sẽ nằm trong **file main.c** (Hình 3.12). Các bạn **lưu ý khi code**, chúng ta nên viết vào vùng bao bọc bởi chú thích có định dạng như sau:

```
/* USER CODE BEGIN ... */

<nội dung code>

/* USER CODE END ... */
```



Hình 3.12. File main.c.

3.3. Thiết lập công cụ dò lỗi (Debug Configurations)

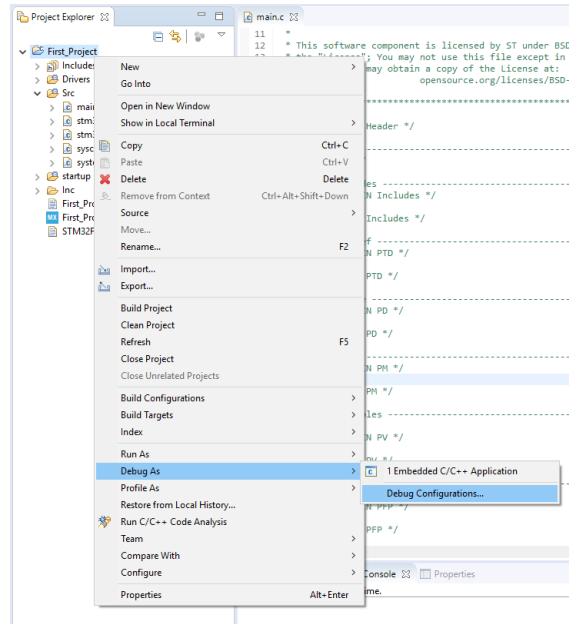
Board Grape32 Unleashed và Grapini32 có tích hợp khối dò lỗi dựa trên SEGGER J-LINK, nó giúp người lập trình kiểm tra hoạt động của chương trình cũng như dò lỗi. Để thiết lập project có thể sử dụng đúng khối dò lỗi trên board, ta cần truy cập vào cửa sổ **Debug Configurations**. Có hai cách:

- Cách 1: Lựa chọn **biểu tượng Configure Debug** trên thanh công cụ.



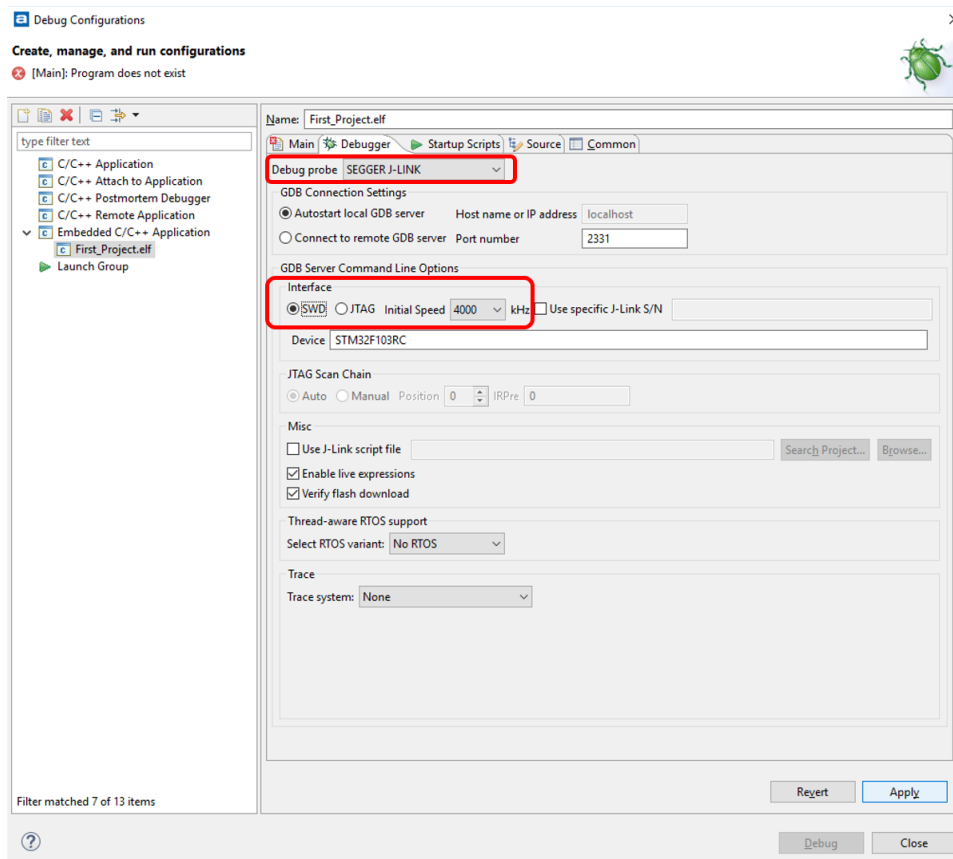
Hình 3.13. Configure Debug.

- Cách 2: Chuột phải vào tên project (**First_Project**) > Chọn **Debug As** > Chọn **Debug Configurations...**



Hình 3.14. Debug As.

Trong cửa sổ **Debug Configurations**, chọn mục **Embedded C/C++ Application** > Chọn tên-**project.elf** (ở đây là First_Project.elf) > Chọn **tab Debugger** > Thiết lập như Hình 3.15 > Chọn **Apply** > Chọn **Close**.



Hình 3.15. Thiết lập Debugger.

4. Hướng dẫn biên dịch và dò lỗi

Sau khi hoàn tất mọi thiết lập, bạn có thể bắt tay vào xây dựng chương trình của riêng mình. Nên nhớ là chương trình của bạn phải luôn đặt trong vùng bao bọc bởi chú thích có định dạng:

```
/* USER CODE BEGIN ... */  
  
<nội dung code>  
  
/* USER CODE END ... */
```

4.1. Chương trình mẫu

Dưới đây là một chương trình mẫu để điều khiển LED (mà bạn đã cấu hình trên CubeMX theo mục 3.1.1. Bạn có thể **copy và paste vào đúng vị trí trong file main.c (nội dung copy là phần in đậm)** hoặc tự viết lại nếu muốn.

```
/**  
 * @brief The application entry point.  
 * @retval int  
 */  
int main(void)  
{  
    /* USER CODE BEGIN 1 */  
  
    /* USER CODE END 1 */  
  
    /* MCU Configuration-----*/  
-----*/  
  
    /* Reset of all peripherals, Initializes the Flash interface and the  
Systick. */  
    HAL_Init();  
  
    /* USER CODE BEGIN Init */  
  
    /* USER CODE END Init */  
  
    /* Configure the system clock */  
    SystemClock_Config();  
  
    /* USER CODE BEGIN SysInit */  
  
    /* USER CODE END SysInit */  
  
    /* Initialize all configured peripherals */  
    MX_GPIO_Init();
```



```
/* USER CODE BEGIN 2 */

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, GPIO_PIN_SET);
    HAL_Delay(100);

    HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin,
GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin,
GPIO_PIN_SET);
    HAL_Delay(100);

    HAL_GPIO_WritePin(LED_BLUE_GPIO_Port, LED_BLUE_Pin,
GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED_BLUE_GPIO_Port, LED_BLUE_Pin,
GPIO_PIN_SET);
    HAL_Delay(100);

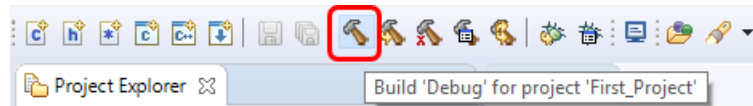
    HAL_GPIO_WritePin(LED0_GPIO_Port, LED0_Pin, GPIO_PIN_SET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_SET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, GPIO_PIN_SET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin, GPIO_PIN_SET);
    HAL_Delay(100);

    HAL_GPIO_WritePin(LED0_GPIO_Port, LED0_Pin, GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin, GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin, GPIO_PIN_RESET);
    HAL_Delay(100);
    HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin, GPIO_PIN_RESET);
    HAL_Delay(100);
}
/* USER CODE END 3 */
}
```

4.2. Biên dịch (Build)

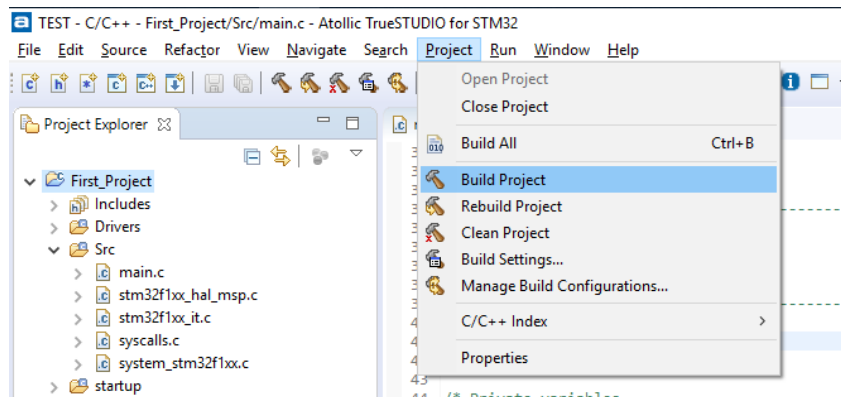
Để biên dịch và kiểm tra lỗi cú pháp của chương trình, bạn sử dụng công cụ **Build**, có hai cách:

- Cách 1: Lựa chọn **biểu tượng Build** (hình chiếc búa) trên thanh công cụ.



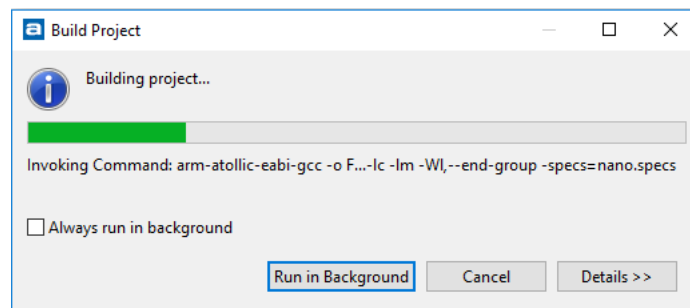
Hình 4.1. Build – cách 1.

- Cách 2: Chọn **Project** > Chọn **Build Project**.

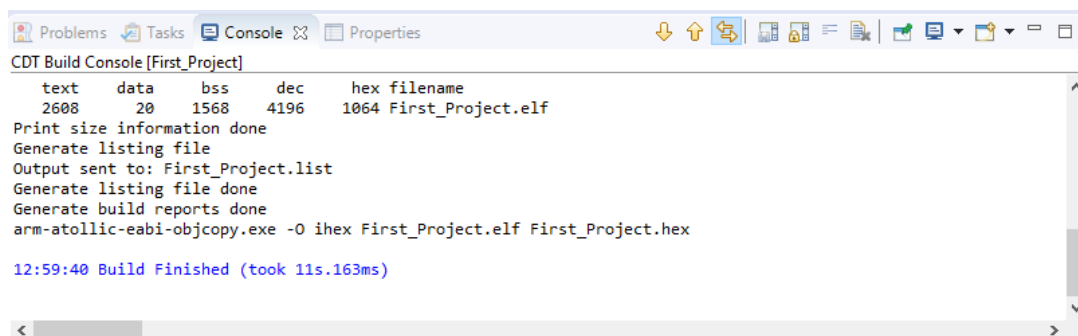


Hình 4.2. Build – cách 2.

Chương trình được biên dịch và báo kết quả ở cửa sổ **Console**.



Hình 4.3. Chương trình đang được biên dịch.

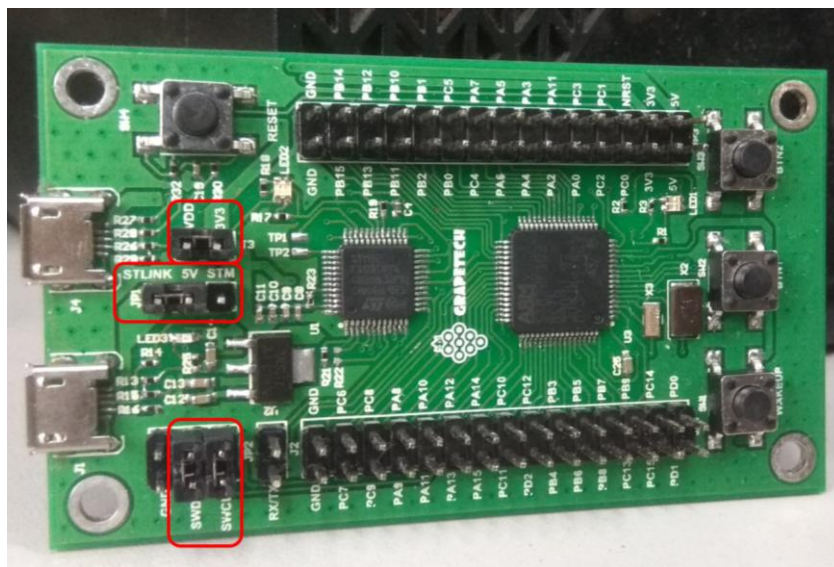


Hình 4.4. Kết quả biên dịch.

4.3. Kết nối board với máy tính

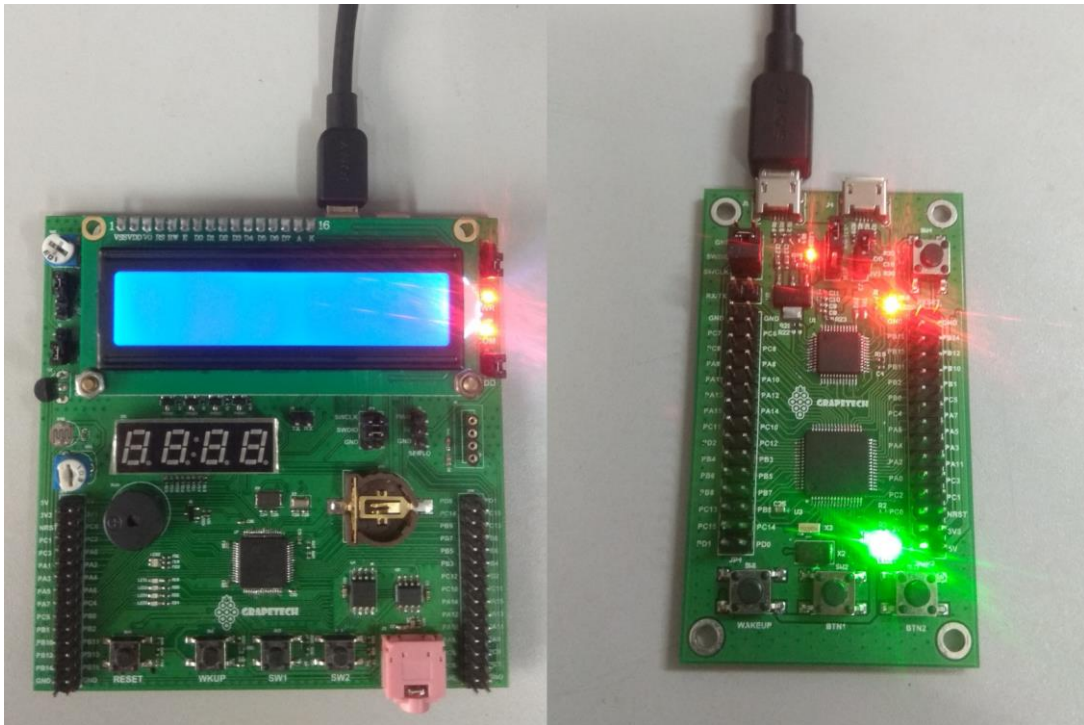
Trước khi nạp chương trình vào, chúng ta phải kiểm tra kỹ lại xem board đã thiết lập phần cứng – **gắn jumper** đúng chưa, nếu sai sẽ nạp không thành công. Các jumper cần được gắn lên board bao gồm những vị trí sau (Hình 4.5):

- Jumper nguồn: Nối VDD và 3V3.
- Jumper lựa chọn cổng micro-USB: Nối STLINK và 5V.
- Jumper hỗ trợ chuẩn nối dây truyền dữ liệu và dò lỗi (để đưa dữ liệu chương trình nạp xuống VĐK): SWDIO và SWCLK.



Hình 4.5. Kết nối jumper.

Sau khi gắn jumper lên board đầy đủ, board đã sẵn sàng để kết nối với máy tính. Bạn chỉ cần sử dụng dây cáp cắm vào cổng micro-USB ở phía bên trái trên board (J1 đối với Grapini32 và J4 đối với Grape32 Unleashed). Nếu kết nối thành công, trên board sẽ có hai đèn báo đỏ sáng lên – một đèn báo nguồn và một đèn báo firmware nạp chương trình vẫn hoạt động bình thường. (Hình 4.6).

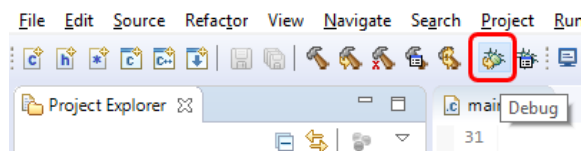


Hình 4.6. Kết nối board với máy tính.

4.4. Nạp và dò lỗi (Debug)

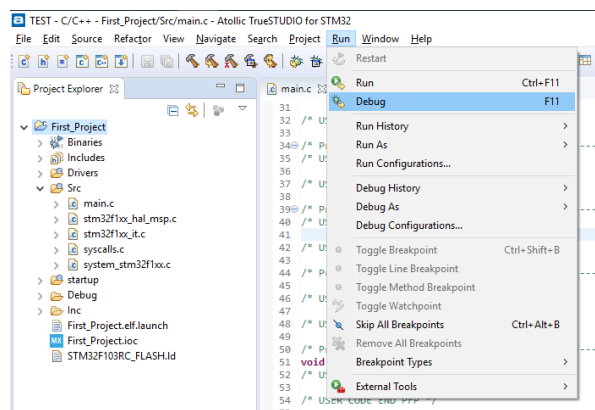
Sau khi biên dịch và kết nối board xong, chúng ta tiến hành **nạp chương trình** vào board, **kiểm tra** và **dò lỗi**. Có hai cách debug:

- Cách 1: Lựa chọn **biểu tượng Debug** (hình con bọ) trên thanh công cụ.



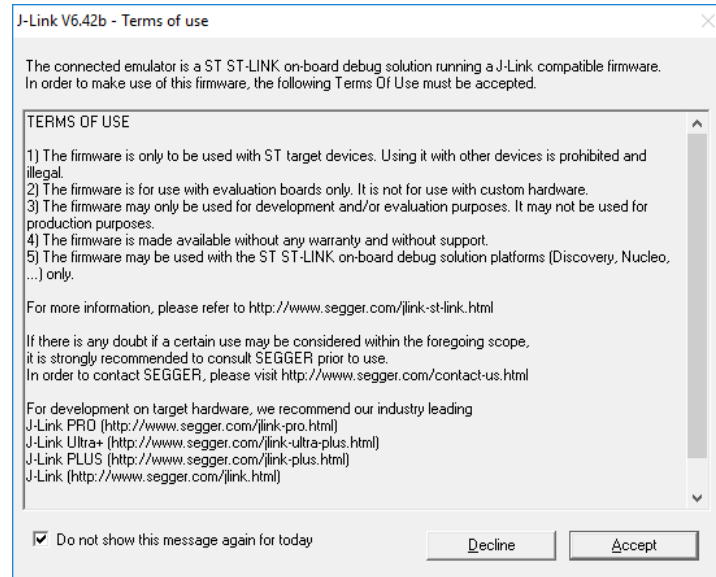
Hình 4.7. Debug – cách 1.

- Cách 2: Chọn **Run > Chọn Debug**.



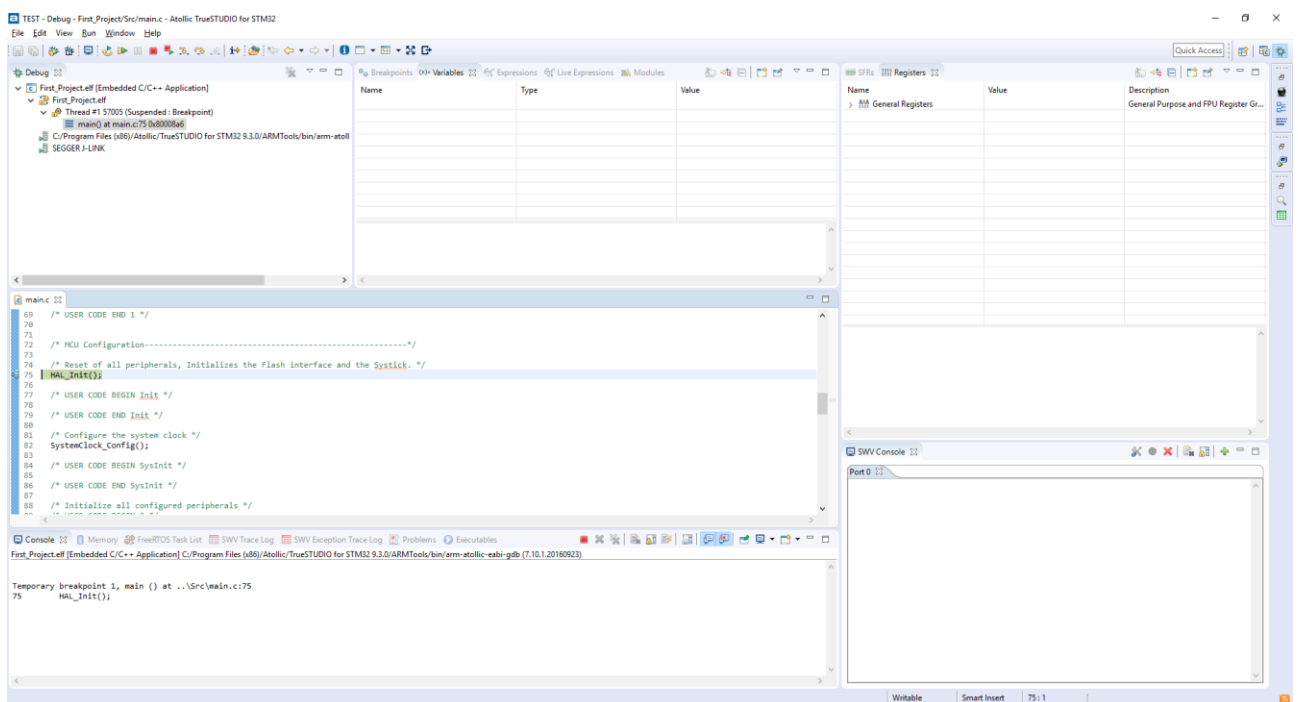
Hình 4.8. Debug – cách 2.

Nếu nạp lần đầu, TrueSTUDIO sẽ cho hiện một cửa sổ như Hình 4.9. Cửa sổ này liệt kê những điều khoản khi sử dụng mạch nạp J-Link (được tích hợp sẵn trên board). Bạn chỉ cần chọn **Accept** (có thể chọn **Do not show this message again for today**) để tắt thông báo này cho những lần nạp sau trong ngày.



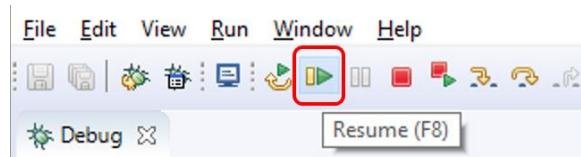
Hình 4.9. Điều khoản sử dụng của J-Link.

Sau khi chấp nhận điều khoản trên, TrueSTUDIO sẽ thay đổi giao diện để phù hợp với việc nạp và dò lỗi chương trình (Hình 4.10).



Hình 4.10. Giao diện nạp và dò lỗi.

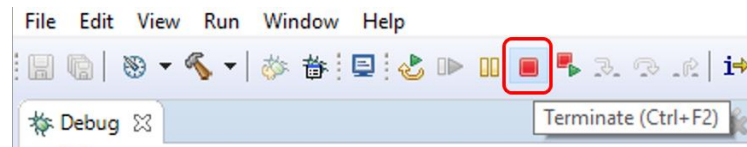
Để nạp chương trình, bạn chỉ cần nhấp vào **biểu tượng Resume** hoặc nhấn **F8**.



Hình 4.11. Resume.

Nếu biểu tượng Resume trở thành màu xám sau khi nhấn thì nghĩa là chương trình đã nạp xong. Đối với đoạn chương trình mẫu ở mục 4.1. , kết quả sau khi nạp là LED RGB sẽ sáng lần lượt màu đỏ, xanh lá, xanh dương, sau đó 4 LED đơn sắc (xanh dương) sẽ sáng theo thứ tự và tắt theo thứ tự.

Nếu muốn dừng chương trình lại, bạn chỉ cần nhấp vào **biểu tượng Terminate** hoặc nhấn **Ctrl+F2**.



Hình 4.12. Terminate.

Và như thế là bạn đã nắm được quy trình cơ bản cách sử dụng board Grape32 Unleashed và Grapini32 – từ việc cấu hình, thiết lập phần cứng cho đến việc tạo project, biên dịch, nạp chương trình và dò lỗi.

Lưu ý: Project mẫu của mục 3 và 4 nằm trong đường dẫn

Grapini32_Grape32Unleashed_Boards/Sample_code/First_Project/, bạn có thể tải repository các chương trình mẫu của chúng tôi tại:

https://github.com/grapetechvn/Grapini32_Grape32Unleashed_Boards