



AMERICAN INTERNATIONAL UNIVERSITY – BANGLADESH(AIUB)

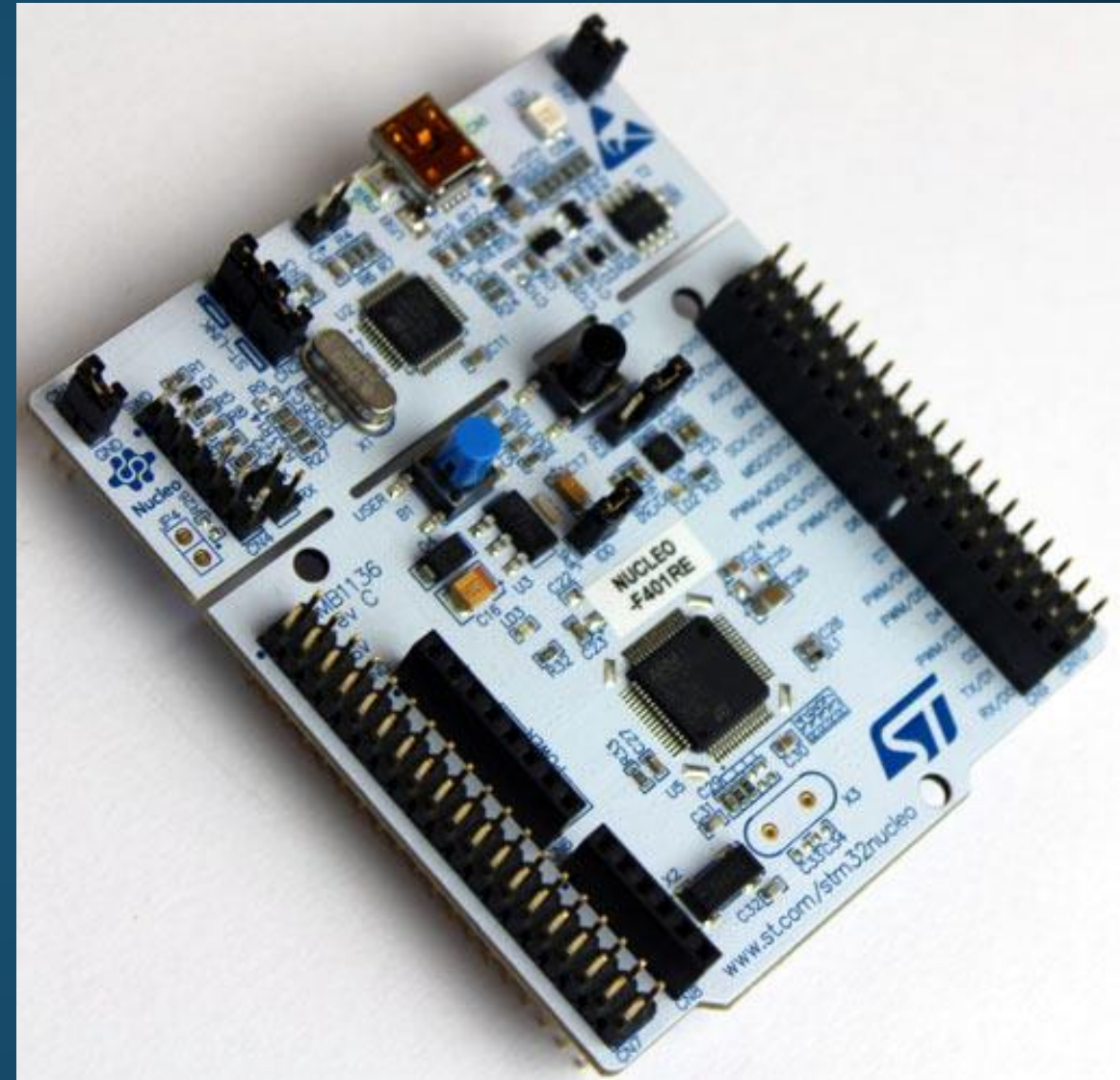
Where leaders are created



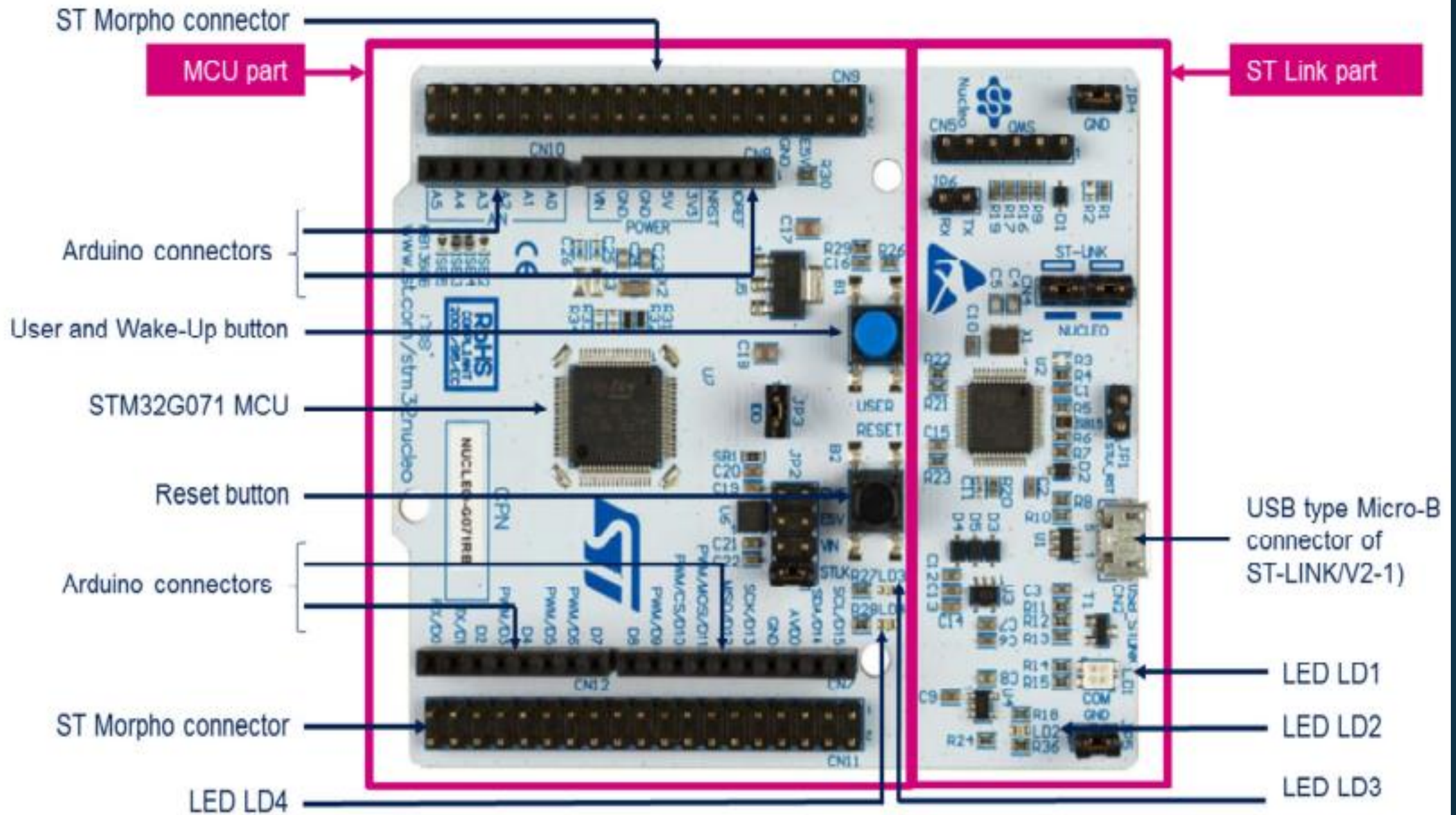
# Introduction to STM32

# Specifications:

- Part number : STM32F401RE
- Core Processor: high-performance ARM® Cortex® -M4
- Operating Frequency of upto 84 MHz.
- The STM32F401xD/xE incorporate high-speed embedded memories (512 Kbytes of Flash memory, 96 Kbytes of SRAM)
- The STM32F401xD/xE operate in the  $-40$  to  $+105$  °C temperature range from a 1.7 (PDR OFF) to 3.6 V power supply. A comprehensive set of power-saving mode allows the design of low-power applications.
- Upto 12 communication interfaces
- Upto 81 I/O ports
- Upto 11 timers

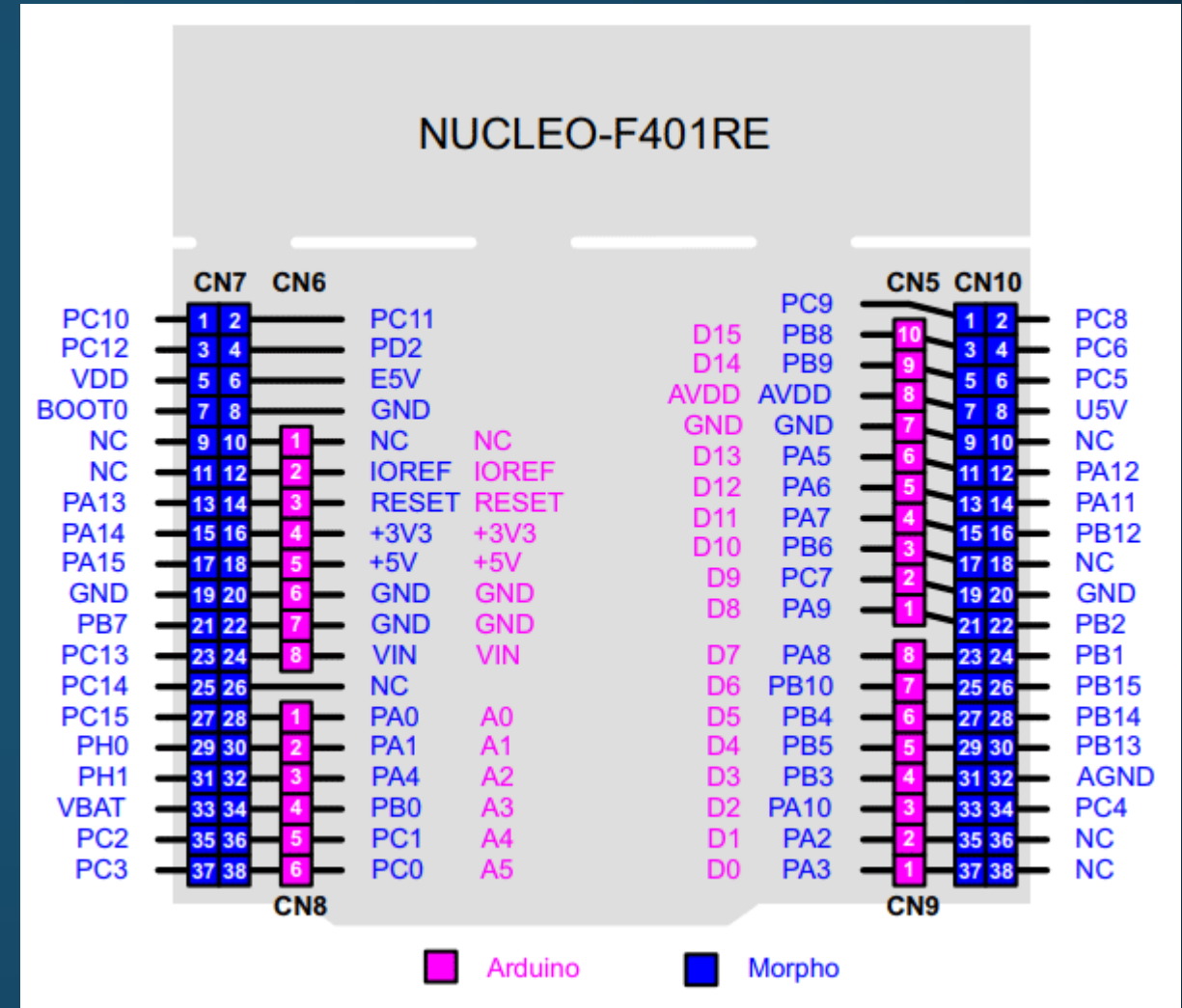






These features make the STM32F401xD/xE microcontrollers suitable for a wide range of applications:

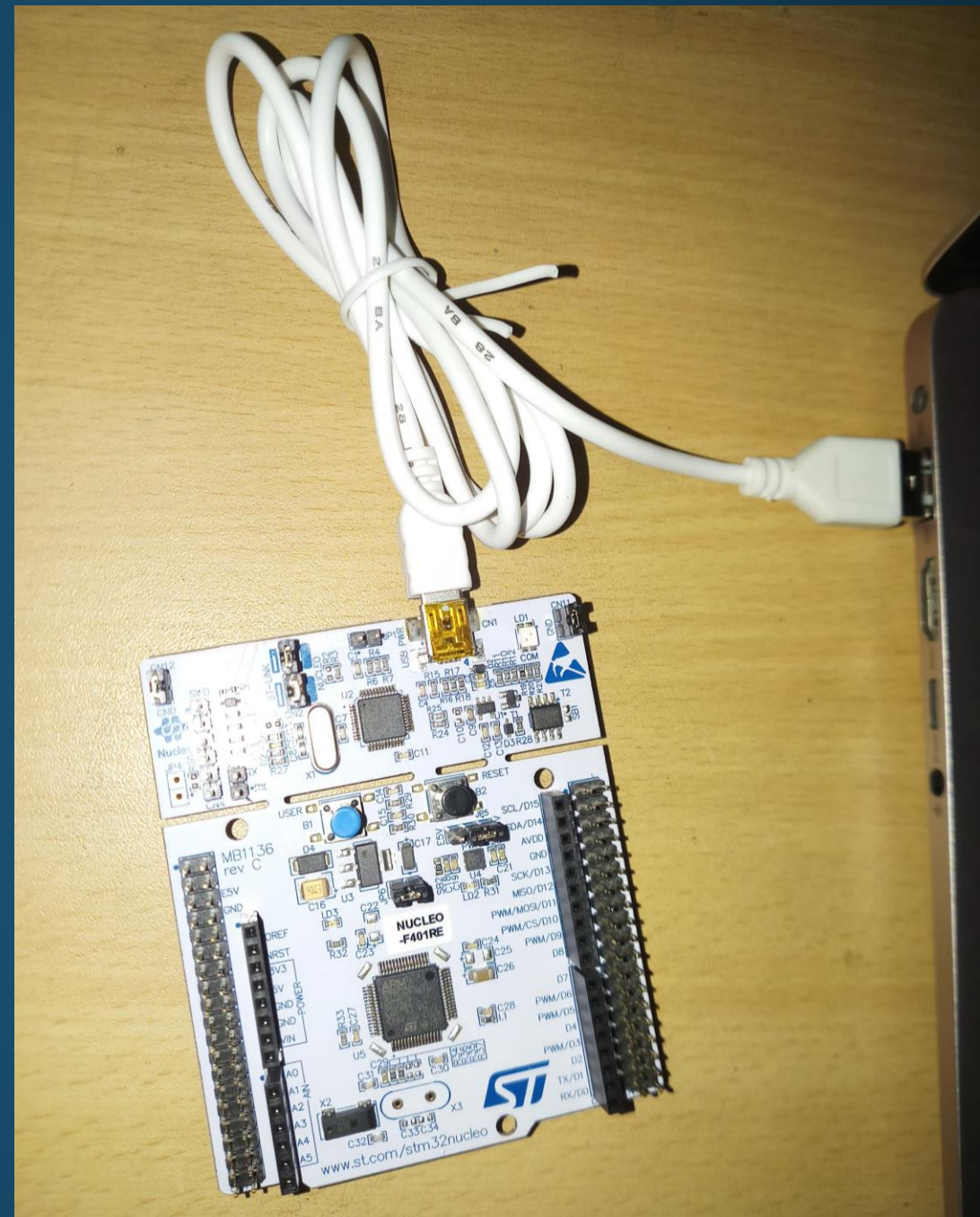
- Motor drive and application control
- Medical equipment
- Industrial applications: PLC, inverters, circuit breakers
- Printers, and scanners
- Alarm systems, video intercom, and HVAC
- Home audio appliances
- Mobile phone sensor hub





Now , lets learn how to start a basic system with this board:

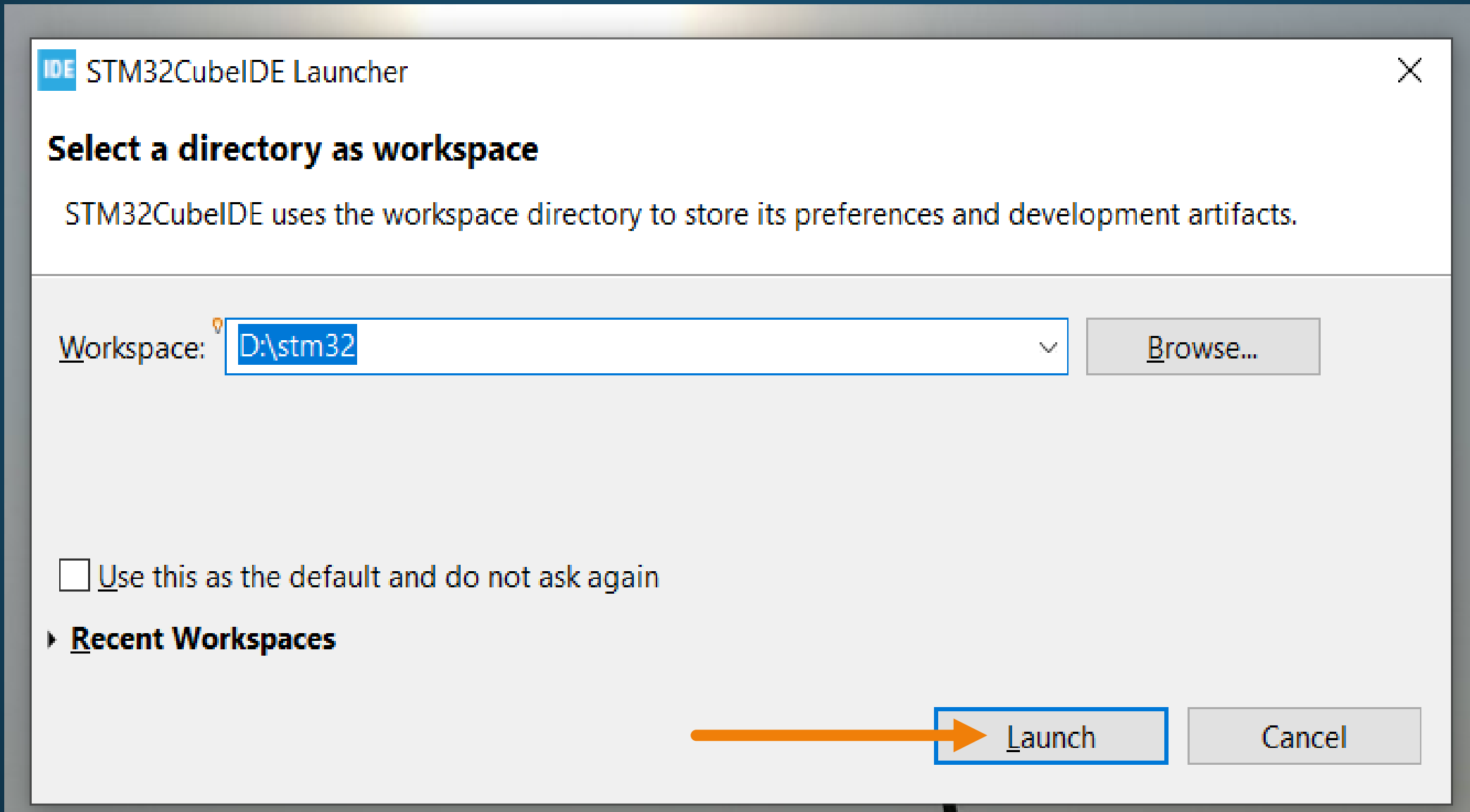
Connect the board to a USB port:



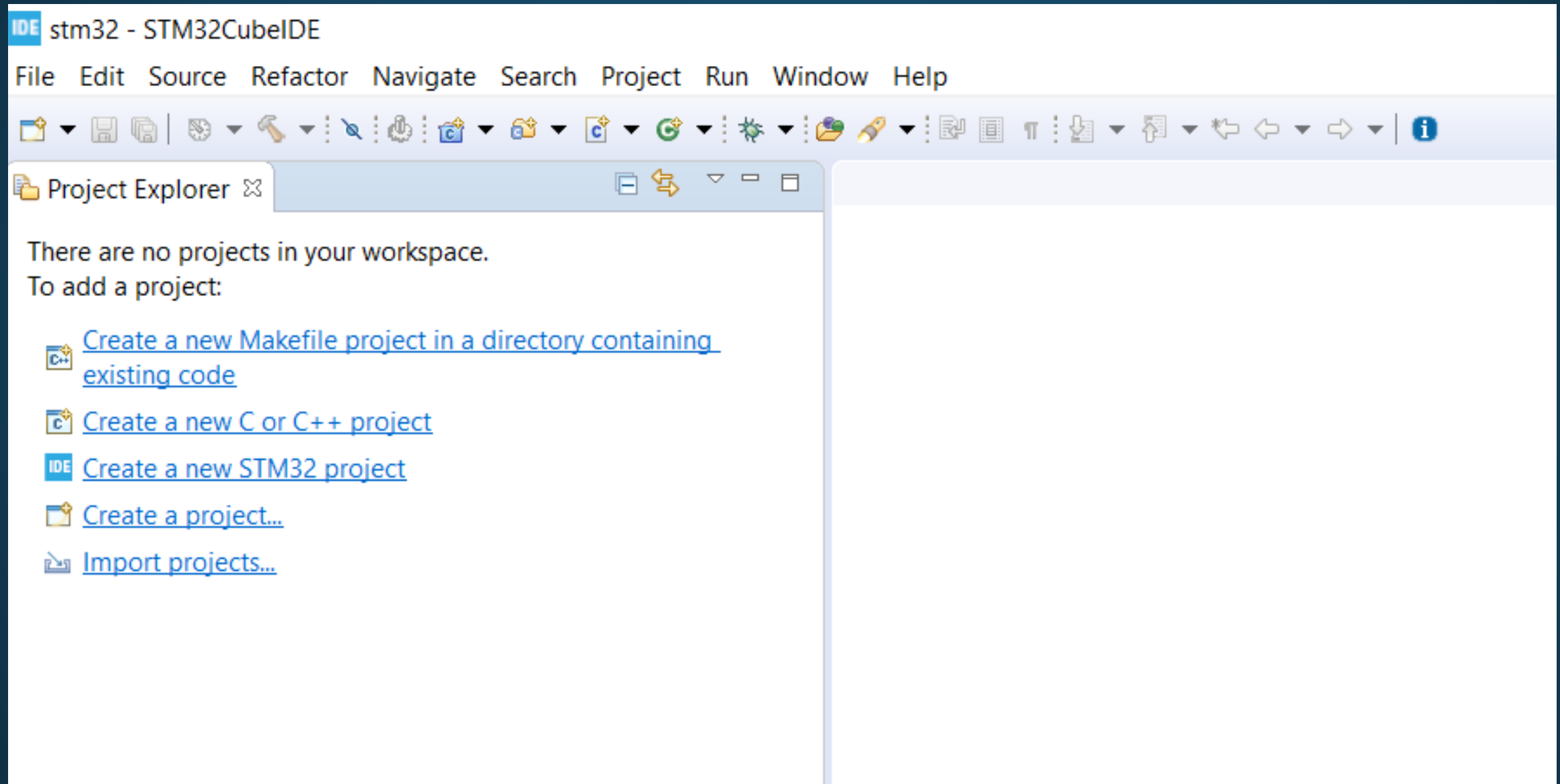
# Open the Software STM32 Cube IDE



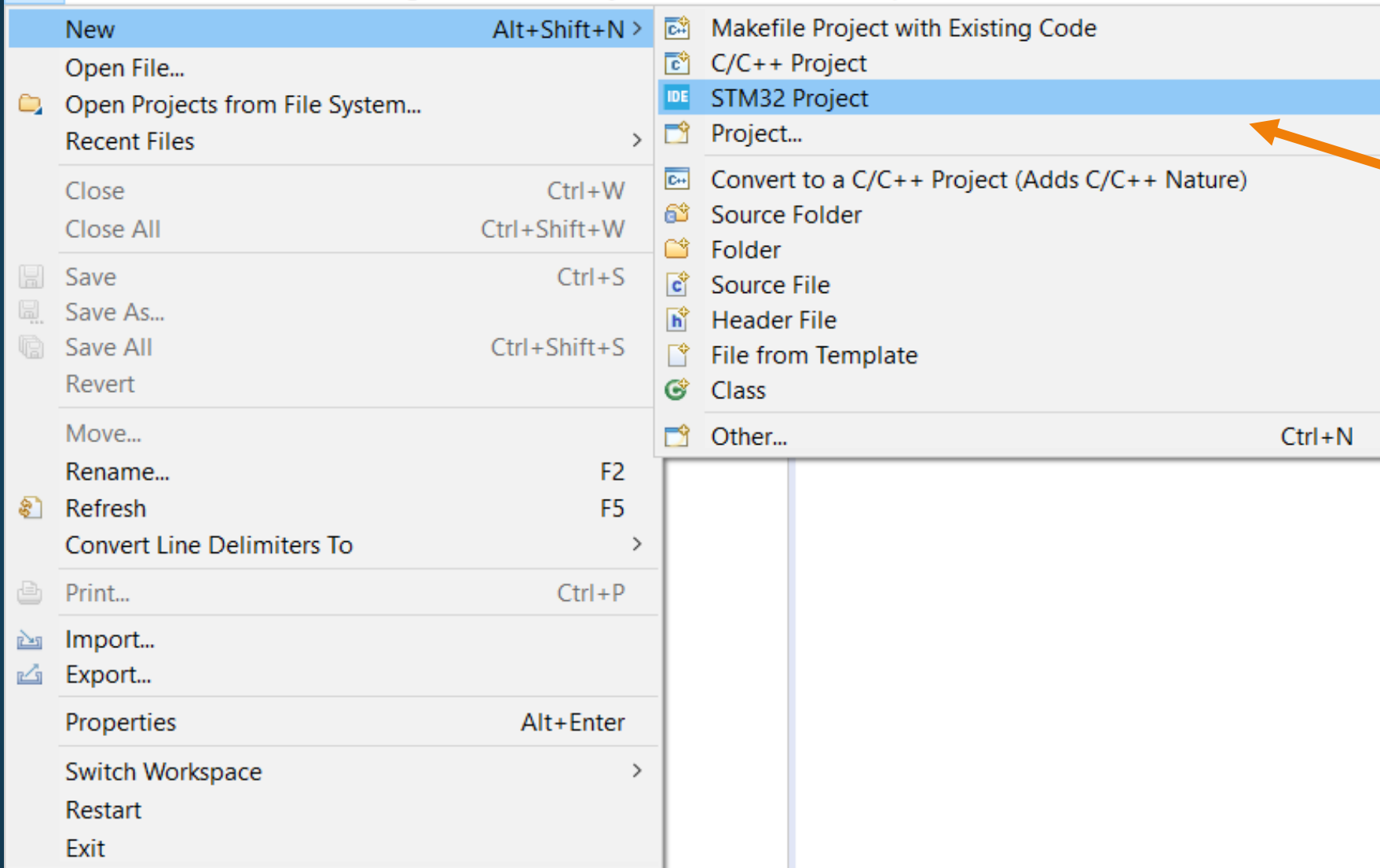
# Click Launch to Select a Workspace in a Specific Directory:



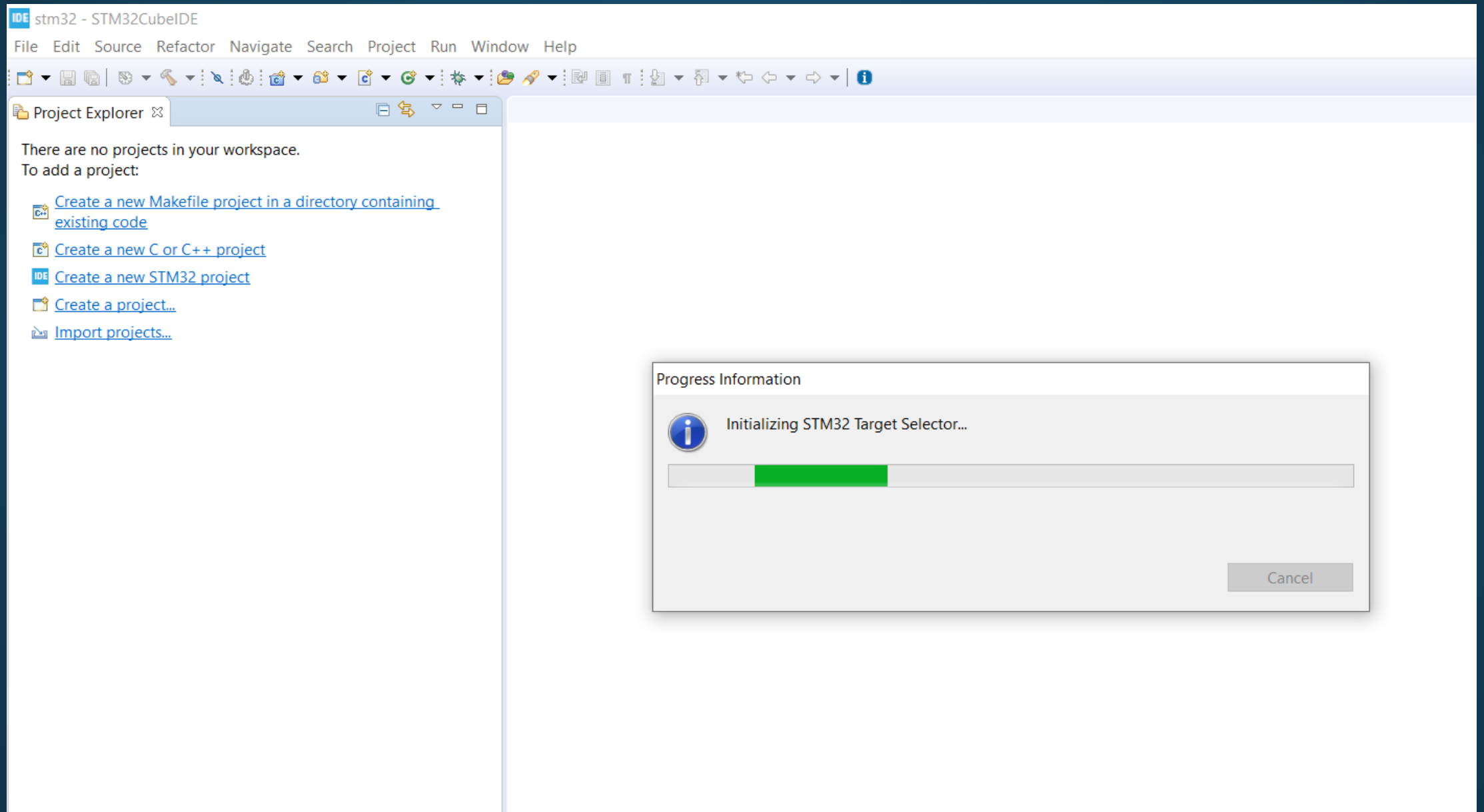
# The following workspace window will pop up:







# This will eventually activate a target selector window:



# A window will pop up:

stm32 - STM32CubeIDE

File Edit Source Refactor Navigate

Project Explorer

There are no projects in your workspace. To add a project:

Create a new Makefile project in a directory with existing code

Create a new C or C++ project

Create a new STM32 project

Create a project...

Import projects...

STM32 Project

Target Selection

Select STM32 target

MCU/MPU Selector Board Selector Cross Selector

MCU/MPU Filters

Part Number Search

Core

Series

Line

Package

Other

Price From 0.0 to 11.543

IO From 0 to 176

Eeprom From 0 to 16384 (Bytes)

Flash From 0 to 2048 (kBytes)

Ram From 0 to 1184 (kBytes)

Freq. From 24 to 800 (MHz)

Advanced Graphic

Features

Block Diagram

Docs & Resources

Datasheet

Buy

STM32Cube

STM32U5 ultra-low-power MCU series with comprehensive STM32Cube ecosystem

ST

MCUs/MPUs List: 1904 items

Display similar items

Part No	Reference	Marketing Status	Unit Price for 10kU (US\$)	Board	Package	Flash	RAM	ID	Freq.	GFX Score
☆ STM32F030C6	STM32F030C6Tx	Active	0.722		LQFP48	32 kBytes	4 kBytes	39	48 MHz	0.0
☆ STM32F030C8	STM32F030C8Tx	Active	0.874		LQFP48	64 kBytes	8 kBytes	39	48 MHz	0.0
☆ STM32F030CC	STM32F030CCTx	Active	1.331		LQFP48	256 kBytes	32 kBytes	37	48 MHz	0.0
☆ STM32F030F4	STM32F030F4Px	Active	0.513		TSSOP20	16 kBytes	4 kBytes	15	48 MHz	0.0
☆ STM32F030K6	STM32F030K6Tx	Active	0.627		LQFP32	32 kBytes	4 kBytes	25	48 MHz	0.0
☆ STM32F030R8	STM32F030R8Tx	Active	0.912	N. 3	LQFP64	64 kBytes	8 kBytes	55	48 MHz	0.0
☆ STM32F030RC	STM32F030RCTx	Active	1.464		LQFP64	256 kBytes	32 kBytes	51	48 MHz	0.0
☆ STM32F031C4	STM32F031C4Tx	Active	1.174		LQFP48	16 kBytes	4 kBytes	39	48 MHz	0.0
☆ STM32F031C6	STM32F031C6Tx	Active	1.226		LQFP48	32 kBytes	4 kBytes	39	48 MHz	0.0
☆ STM32F031E6	STM32F031E6Yx	Active	0.939		WLCSP25	32 kBytes	4 kBytes	20	48 MHz	0.0
☆ STM32F031F4	STM32F031F4Px	Active	0.861		TSSOP20	16 kBytes	4 kBytes	15	48 MHz	0.0
☆ STM32F031F6	STM32F031F6Px	Active	0.913		TSSOP20	32 kBytes	4 kBytes	15	48 MHz	0.0

Back

Next

Finish

Cancel

# Click on board selector and notice blank space of part number search:

The screenshot shows the STM32 Project IDE interface. The 'Board Selector' tab is active, and an orange arrow points to the 'Part Number Search' field, which is currently blank. The interface includes filters for Vendor, Type, MCU/MPU Series, and Other, as well as a list of boards with columns for Overview, Part No., Type, Marketing Status, Unit Price, and Mounted Device.

**Board Filters:**

- Part Number Search:
- Vendor:
- Type:
- MCU/MPU Series:
- Other:
- Price: From 0.0 to 560.0
- Oscillator Freq. From 0 to 25 (MHz)
- Peripheral:

**Boards List: 164 items**

*	Overview	Part No.	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		32F030R8DISCOVERY	Discovery	Obsolete	8.9	<a href="#">STM32F030R8Tx</a>
☆		32F072R8DISCOVERY	Discovery	Active	10.4	<a href="#">STM32F072R8Tx</a>
☆		32F3348DISCOVERY	Discovery	Active	17.0	<a href="#">STM32F3348Tx</a>



stm32 - STM32CubeIDE

File Edit Source Refactor Navigate

Project Explorer

There are no projects in your workspace.  
To add a project:

Create a new Makefile project in a directory with existing code

Create a new C or C++ project

Create a new STM32 project

Create a project...

Import projects...

STM32 Project

Target Selection

Select STM32 target

MCU/MPU Selector Board Selector Cross Selector

Board Filters

Part Number Search

Vendor

Type

MCU/MPU Series

Other

Price = 13.0

Oscillator Freq. = 0 (MHz)

Peripheral

Accelerometer

Analog I/O

Arduino Form Factor

Audio Line In

Audio Line Out

Battery

Button

CAN

Camera

Compass

Custom Form Factor

Features


Large Picture

Docs & Resources

Datasheet

Buy

NUCLEO-F401RE




**STMicroelectronics NUCLEO-F401RE Board Support and Examples**

ACTIVE Active

Product is in mass production


Unit Price (US\$) : 13.0

Mounted device: [STM32F401RETx](#)



The STM32 Nucleo-64 board provides an affordable and flexible way for users to try out new concepts and build prototypes by choosing from the various combinations of performance and power consumption features, provided by the STM32 microcontroller. For the compatible boards, the external SMPS significantly reduces power consumption in Run mode.  
The ARDUINO® Uno V3 connectivity support and the ST morpho headers allow the easy

Boards List: 1 item

	Overview	Part No	Type	Marketing Status	Unit Price (US\$)	Mounted Device
☆		NUCLEO-F401RE	Nucleo64	Active	13.0	STM32F401RETx

< Back

Next >

Finish

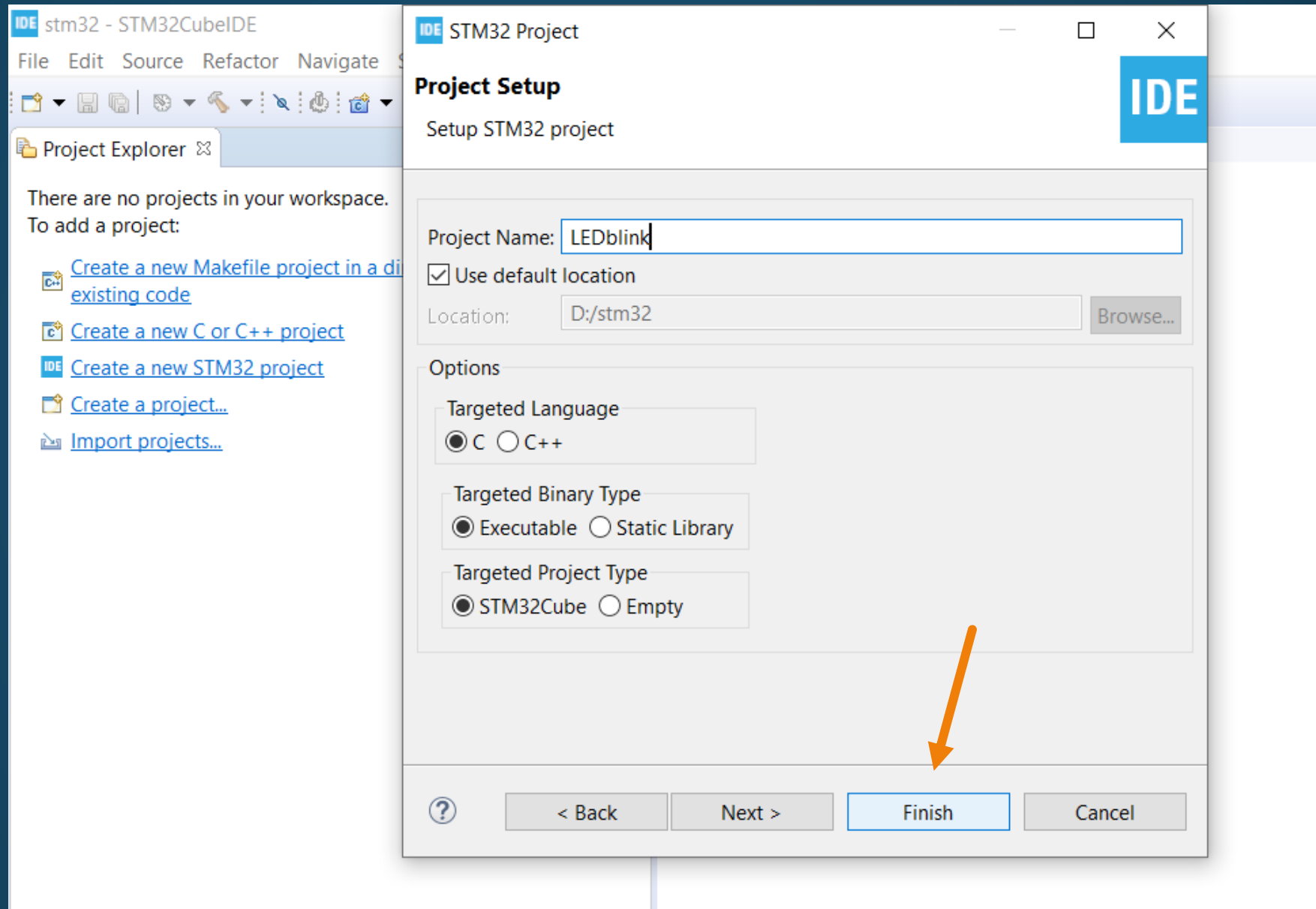
Cancel

Quick Access

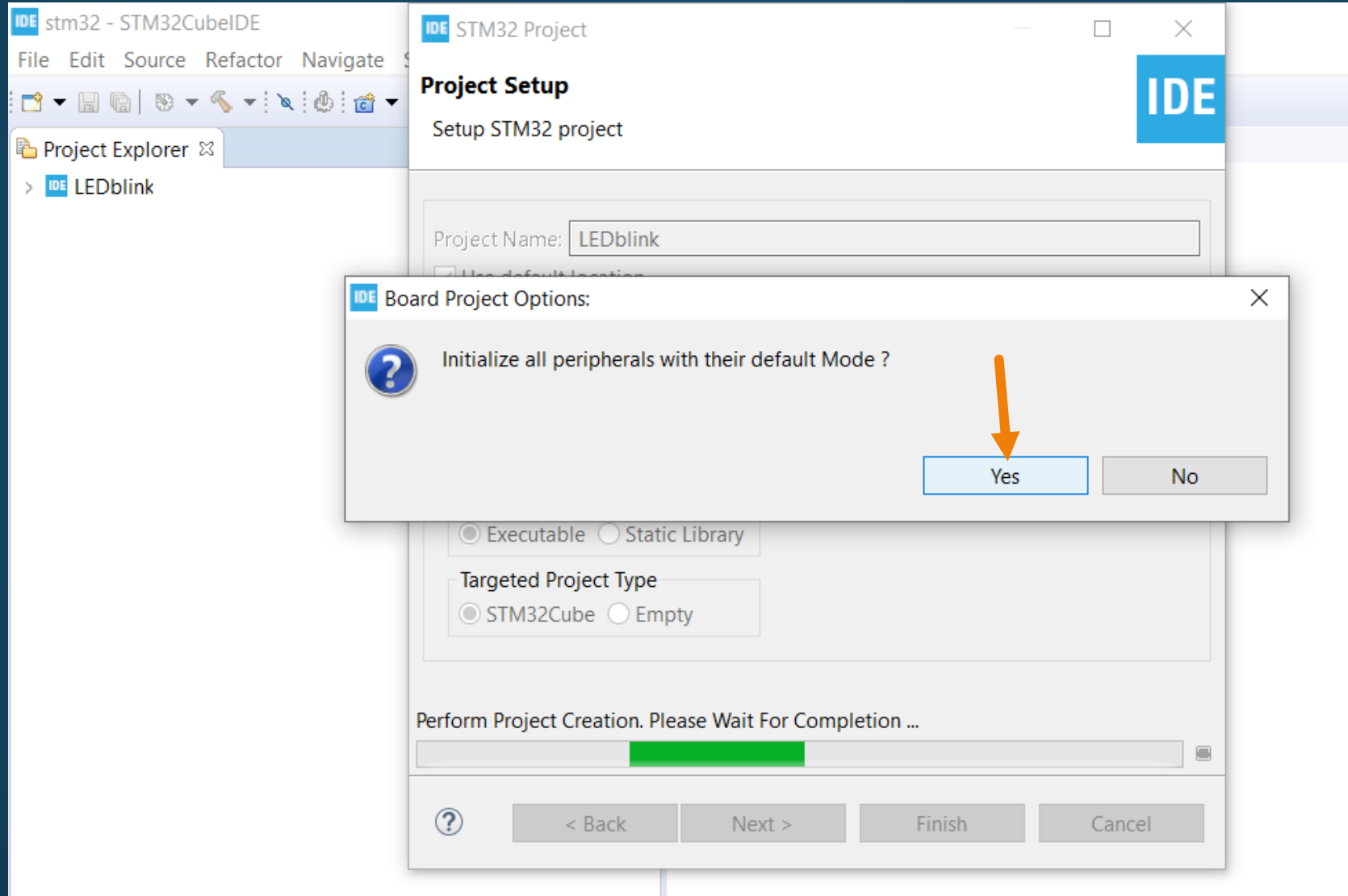
IDE

13

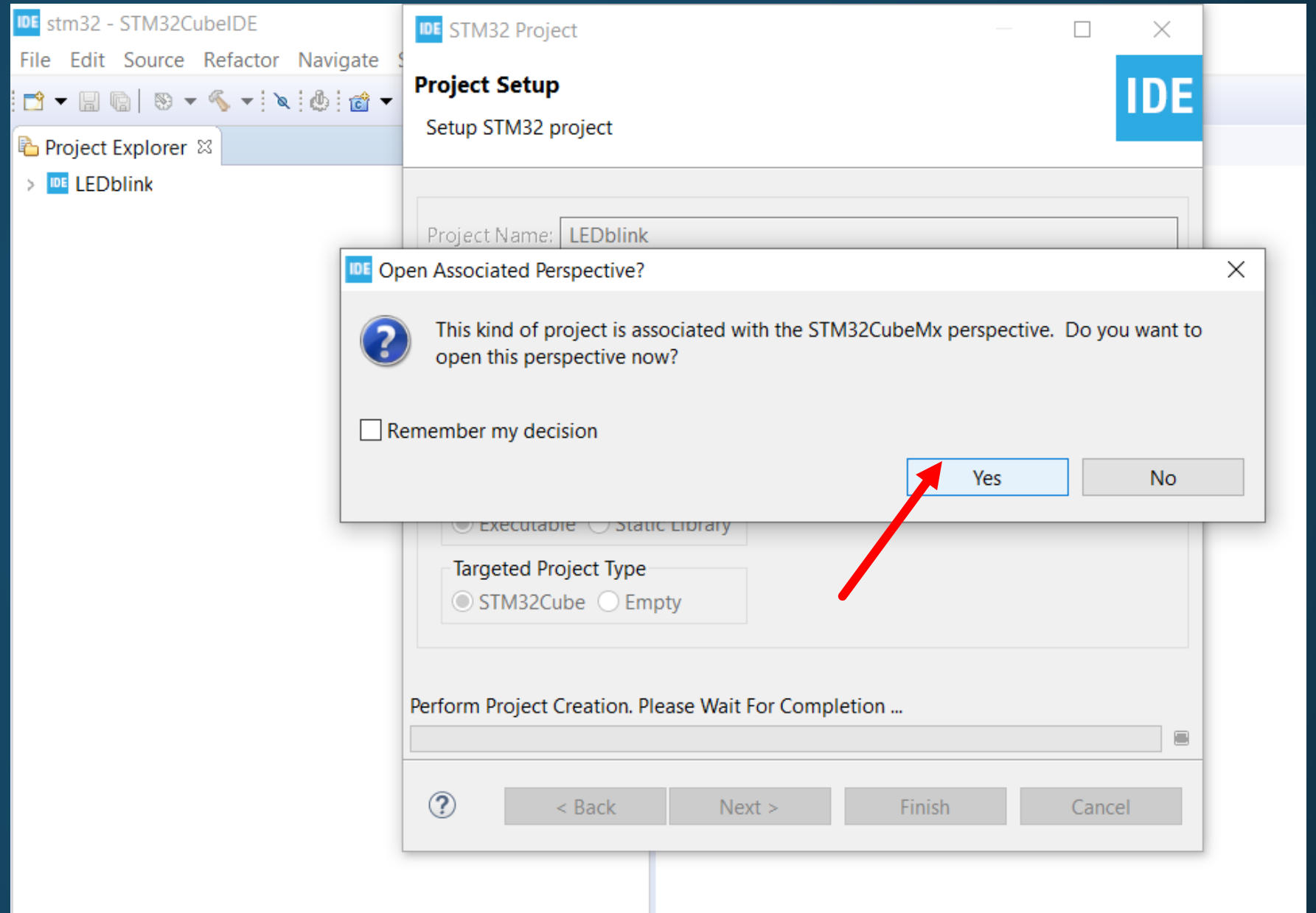
# Write the name of the program in project setup window:



# Select the option yes to initialize all the peripherals with default mode:

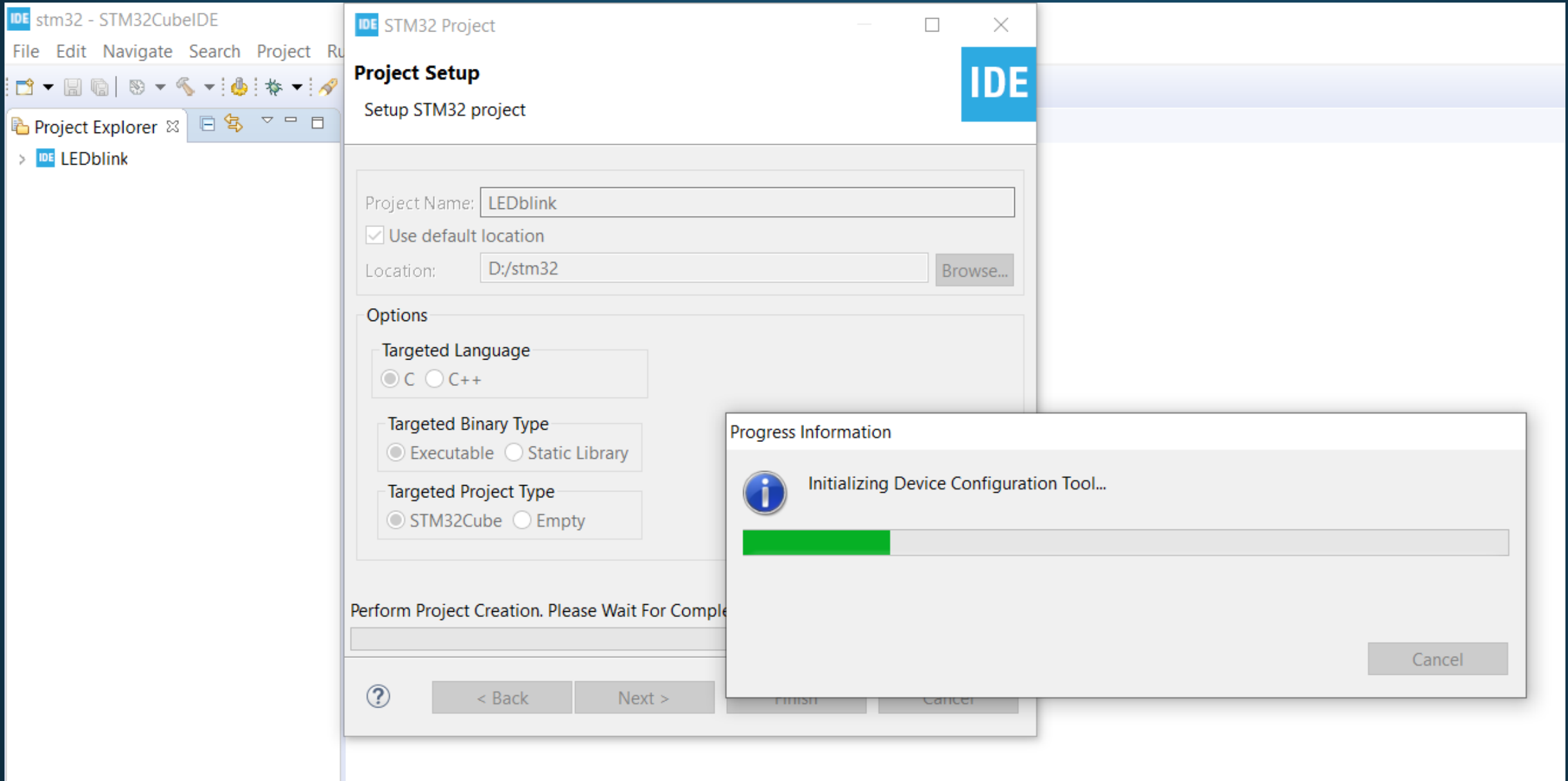


# Select **Yes** to open perspective:

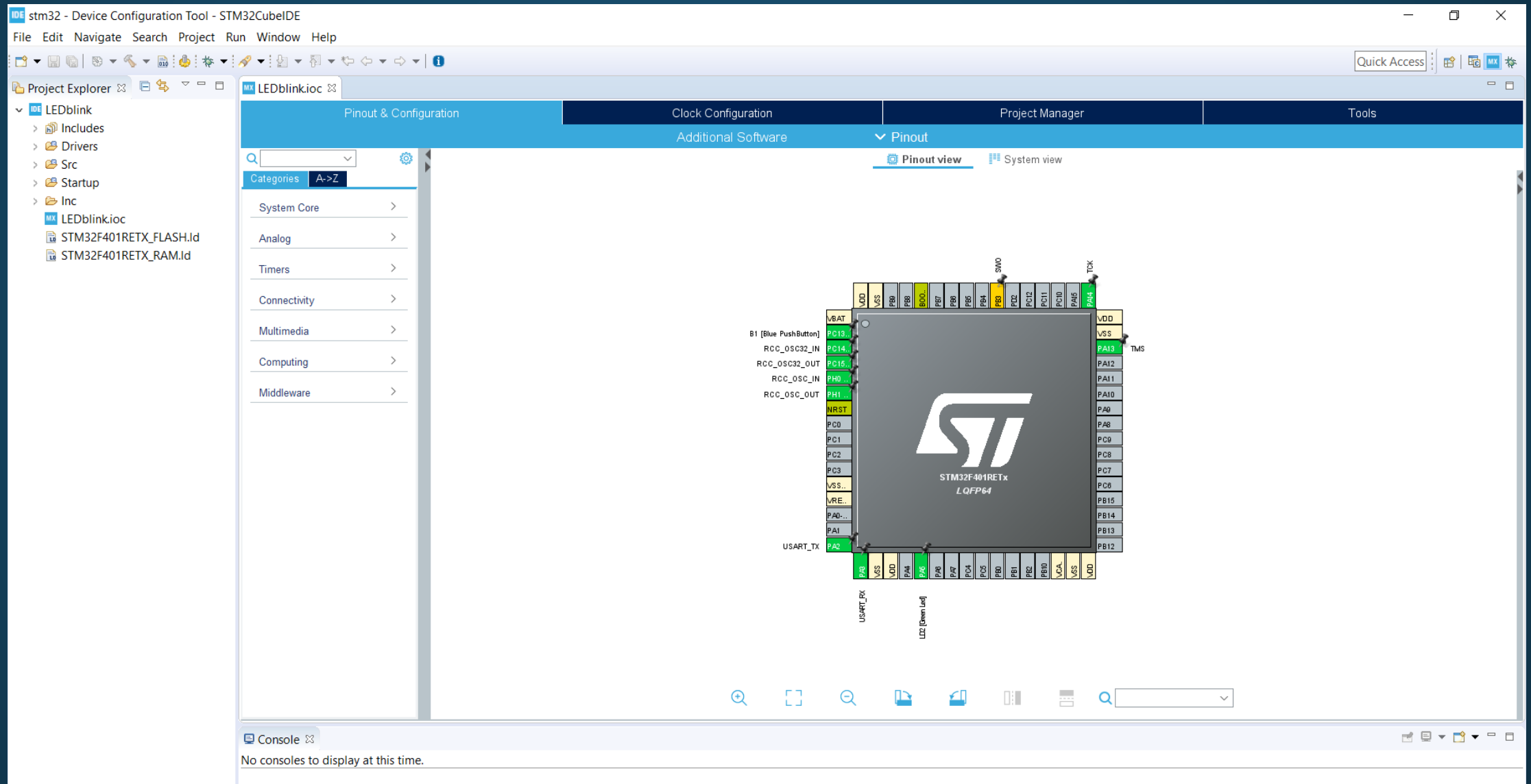




# Device configuration tool is initializing:



# Wait for the packages to be downloaded: it will be easy to navigate through the ports and pins, clock frequency, and project manager.



# Clock Configuration:

stm32 - Device Configuration Tool - STM32CubeIDE

File Edit Navigate Search Project Run Window Help

Project Explorer

LEDblink

Includes

Drivers

Src

Startup

Inc

LEDblink.ioc

STM32F401RET\_X\_FLASH.Id

STM32F401RET\_X\_RAM.Id

LEDblink.ioc

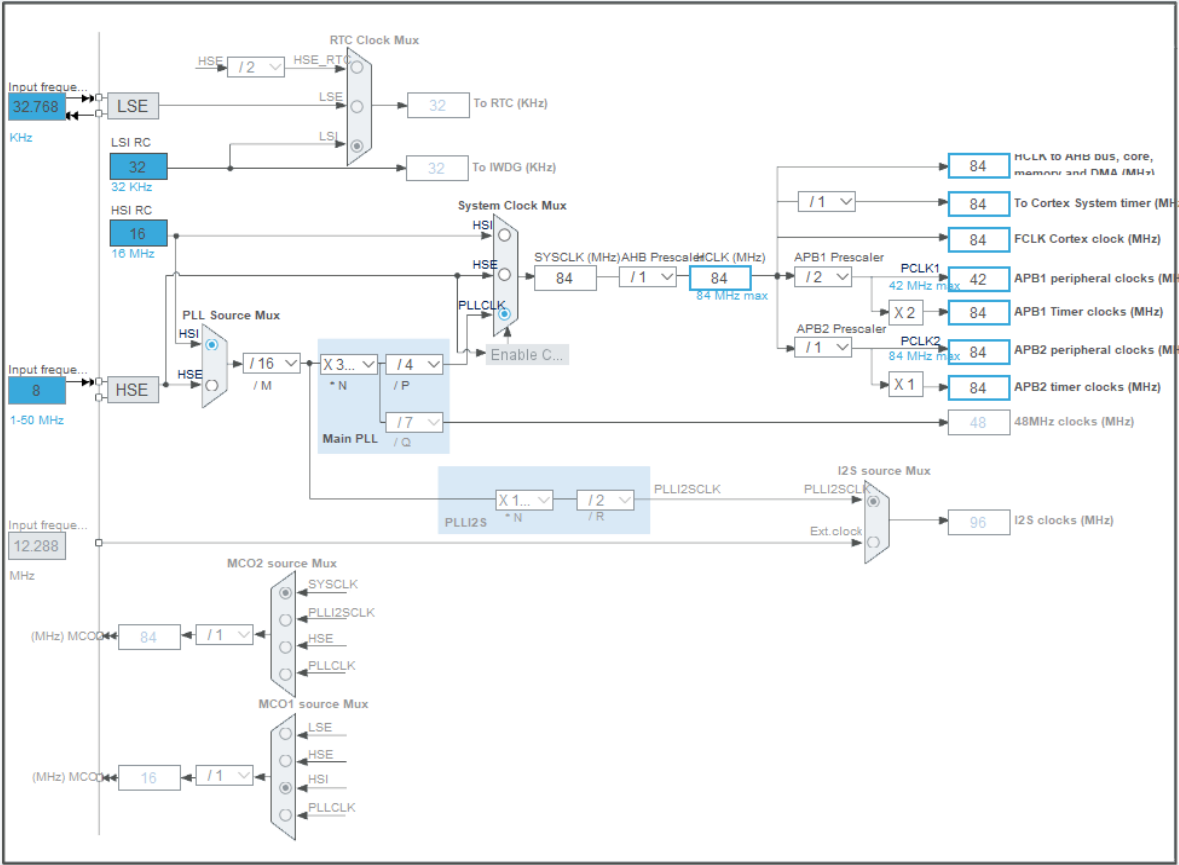
Pinout & Configuration

Clock Configuration

Project Manager

Tools

Resolve Clock Issues



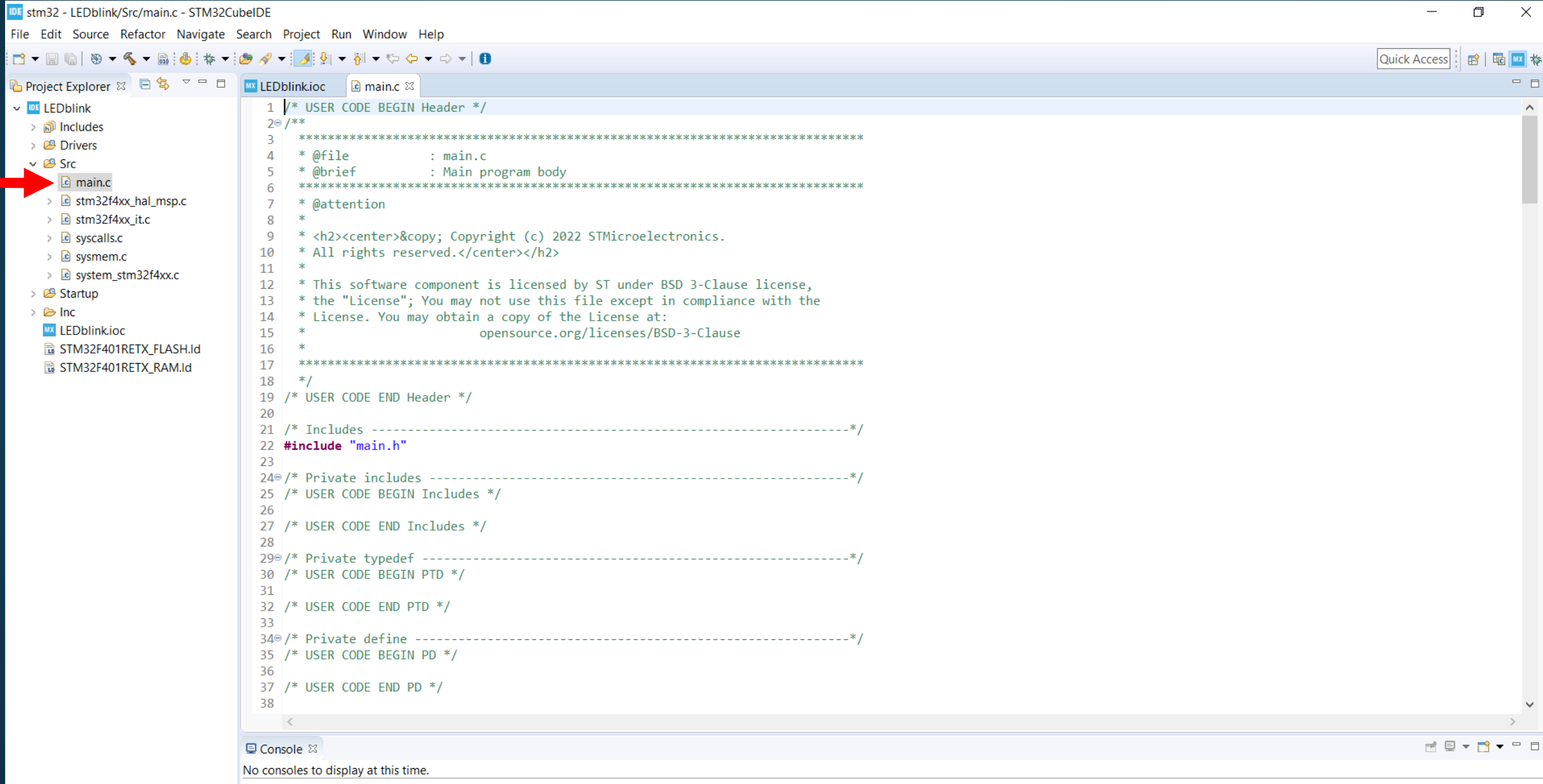
The diagram illustrates the clock configuration for the STM32F401RET. It shows the flow from input frequencies to various system and peripheral clocks. Key components include:

- Input Frequencies:** 32.768 KHz (LSE), 16 MHz (HSI), 1-50 MHz (HSE), and 12.288 MHz (MCO).
- PLL Source Mux:** Selects between HSI and HSE for the PLL.
- Main PLL:** Configured with a multiplier of 3, divider of 4, and divider of 7. It outputs PLLCLK (84 MHz) and PLLI2SCLK (96 MHz).
- System Clock Mux:** Selects between HSI and HSE for the system clock (SYSCLK).
- APB1 and APB2 Prescalers:** Configure the prescalers for the APB1 and APB2 buses.
- I2S Source Mux:** Selects between SYSCLK and PLLI2SCLK for the I2S clock.

Console

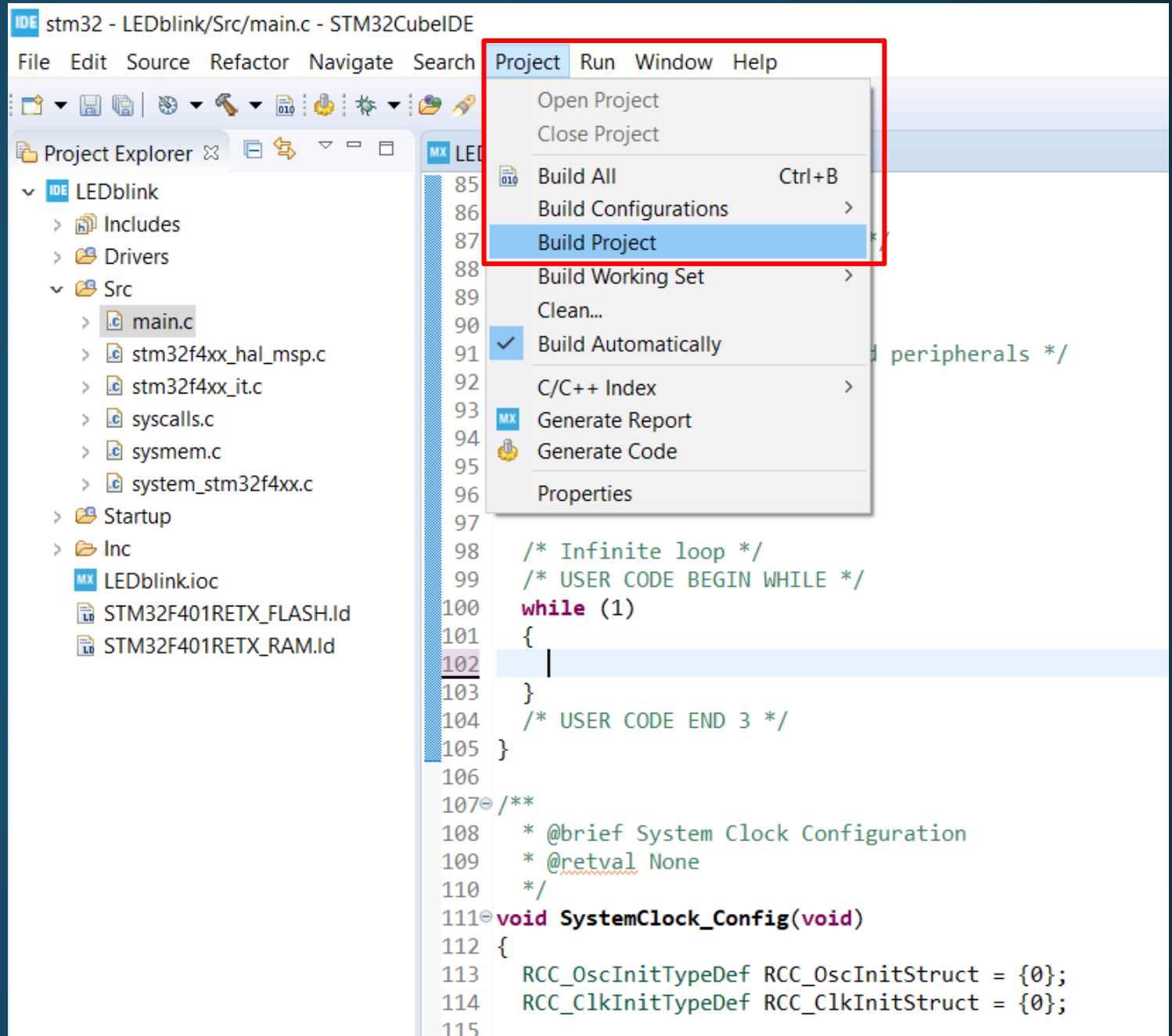
No consoles to display at this time.

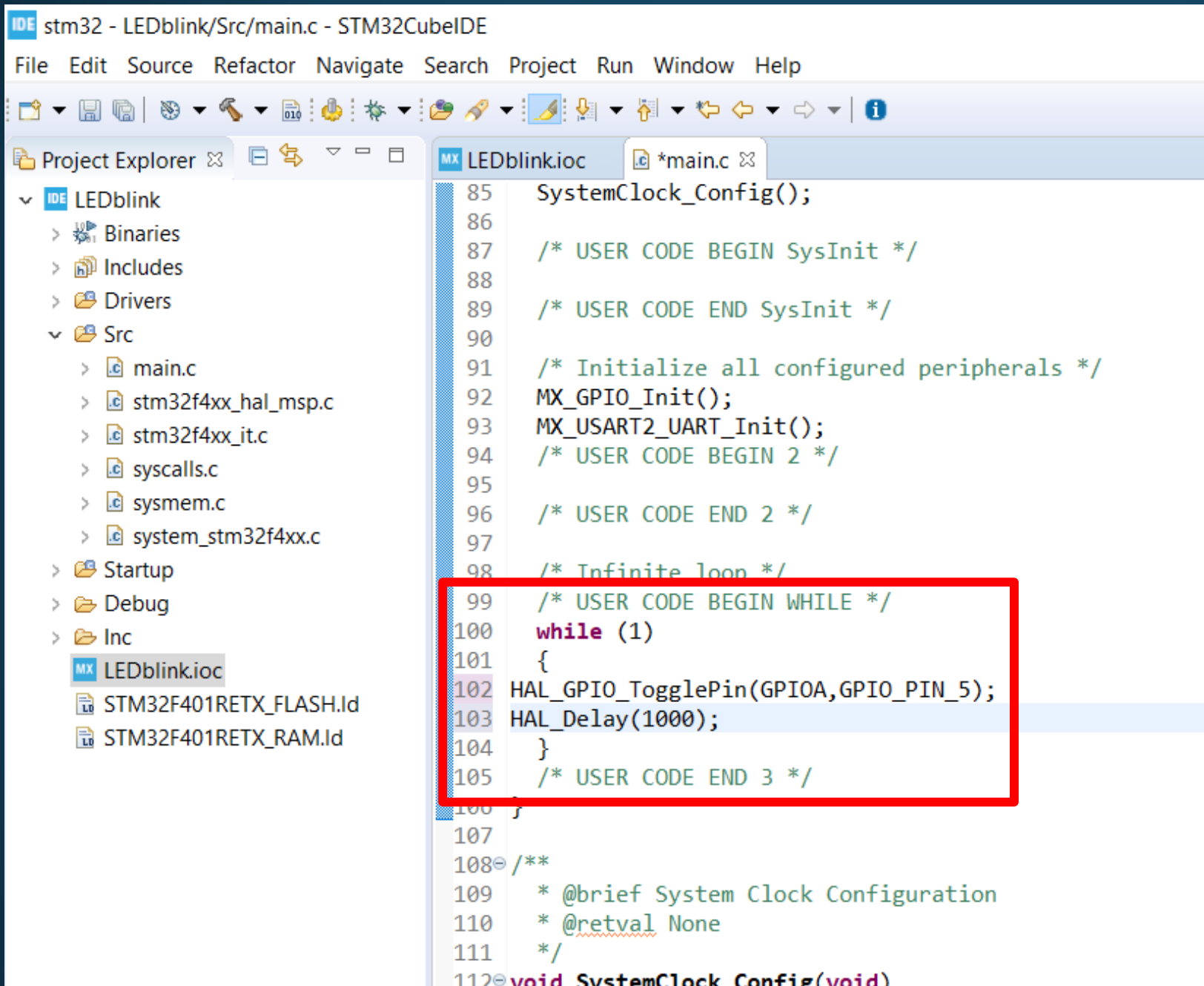
# How timer works:





# Click Project Menu > Build Project





The screenshot shows the STM32CubeIDE interface. The Project Explorer on the left shows the project structure for 'LEDblink'. The main editor displays the 'main.c' file. A red rectangle highlights a while loop starting at line 99. The code in the while loop calls 'HAL\_GPIO\_TogglePin' and 'HAL\_Delay'.

```
85  SystemClock_Config();
86
87  /* USER CODE BEGIN SysInit */
88
89  /* USER CODE END SysInit */
90
91  /* Initialize all configured peripherals */
92  MX_GPIO_Init();
93  MX_USART2_UART_Init();
94  /* USER CODE BEGIN 2 */
95
96  /* USER CODE END 2 */
97
98  /* Infinite loop */
99  /* USER CODE BEGIN WHILE */
100 while (1)
101 {
102     HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
103     HAL_Delay(1000);
104 }
105 /* USER CODE END 3 */
106
107
108 /**
109  * @brief System Clock Configuration
110  * @retval None
111  */
112 void SystemClock_Config(void)
```

**Writing instructions:**

Nucleo board comes with the STM32 comprehensive software **Hardware Abstraction Layer (HAL)** drivers together with various packaged software examples.

It also embeds a debugger that helps you to develop your own applications



Project Explorer

- IDE LEDblink
  - > Binaries
  - > Includes
  - > Drivers
  - > Src
    - main.c
    - stm32f4xx\_hal\_msp.c
    - stm32f4xx\_it.c
    - syscalls.c
    - sysmem.c
    - system\_stm32f4xx.c
  - > Startup
  - > Debug
  - > Inc
  - LEDblink.ioc
  - STM32F401RETX\_FLASH.Id
  - STM32F401RETX\_RAM.Id

- Open Project
- Close Project
- Build All Ctrl+B
- Build Configurations >
- Build Project
- Build Working Set >
- Clean...
- Build Automatically
- C/C++ Index >
- Generate Report
- Generate Code
- Properties

```
85
86
87
88
89
90
91
92
93
94
95
96
97
98 /* Infinite loop */
99 /* USER CODE BEGIN WHILE */
100 while (1)
101 {
102     HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5)
103     HAL_Delay(1000);
104 }
105 /* USER CODE END WHILE */
106 }
107
108 /**
109  * @brief System Clock Configuration
110  * @retval None
```



Project Explorer

- IDE LEDblink
  - > Binaries
  - > Includes
  - > Drivers
  - > Src
    - main.c
    - stm32f4xx\_hal\_msp.c
    - stm32f4xx\_it.c
    - syscalls.c
    - systemem.c
    - system\_stm32f4xx.c
  - > Startup
  - > Debug
  - > Inc
- MX LEDblink.ioc
- STM32F401RETX\_FLASH.Id
- STM32F401RETX\_RAM.Id

```
85  Syst
86
87  /* U
88
89  /* USER CODE END SysInit */
90
91  /* Initialize all configured peripherals */
92  MX_GPIO_Init();
93  MX_USART2_UART_Init();
94  /* USER CODE BEGIN 2 */
95
96  /* USER CODE END 2 */
97
98  /* Infinite loop */
99  /* USER CODE BEGIN WHILE */
100 while (1)
101 {
102   HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
103   HAL_Delay(1000);|
104 }
105 /* USER CODE END 3 */
106 }
107
108 /**
109  * @brief System Clock Configuration
110  * @retval None
111  */
112 void SystemClock_Config(void)
113 {
114   RCC_OscInitTypeDef RCC_OscInitStruct = {0};
115   RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
```

- Reset
- Debug F11
- Debug History >
- Debug As >
- Debug Configurations...

- 1 Local C/C++ Application
- IDE 2 STM32 MCU C/C++ Application





## Project Explorer

- IDE LEDblink
  - > Binaries
  - > Includes
  - > Drivers
  - > Src
    - > main.c
    - > stm32f4xx\_hal\_msp.c
    - > stm32f4xx\_it.c
    - > syscalls.c
    - > sysmem.c
    - > system\_stm32f4xx.c
  - > Startup
  - > Debug
  - > Inc
    - LEDblink.elf.launch
    - LEDblink.ioc
    - STM32F401RETX\_FLASH.Id
    - STM32F401RETX\_RAM.Id

```
85  SystemClock_Config();
86
87  /* USER CODE BEGIN SysInit */
88
89  /* USER CODE END SysInit */
90
91  /* Initialize all configured
92  MX_GPIO_Init();
93  MX_USART2_UART_Init();
94  /* USER CODE BEGIN 2 */
95
96  /* USER CODE END 2 */
97
98  /* Infinite loop */
99  /* USER CODE BEGIN WHILE */
100 while (1)
101 {
102 HAL_GPIO_TogglePin(GPIOA,GPIO_
103 HAL_Delay(1000);
104 }
105 /* USER CODE END 3 */
106 }
107
108 /**
109  * @brief System Clock Config
110  * @retval None
111  */
112 void SystemClock_Config(void)
113 {
114 RCC_OscInitTypeDef RCC_OscIn
115 RCC_ClkInitTypeDef RCC_ClkIn
116
117 /** Configure the main internal regulator output voltage
118  */
119 __HAL_RCC_PWR_CLK_ENABLE();
```

## Edit Configuration

## Edit launch configuration properties

Name: LEDblink.elf

Main Debugger Startup Source Common

C/C++ Application:

Debug/LEDblink.elf

Search Project...

Browse...

Project:

LEDblink

Browse...

Build (if required) before launching

Build Configuration: Use Active

☐ Enable auto build☐ Disable auto build☒ Use workspace settings[Configure Workspace Settings...](#)

Revert

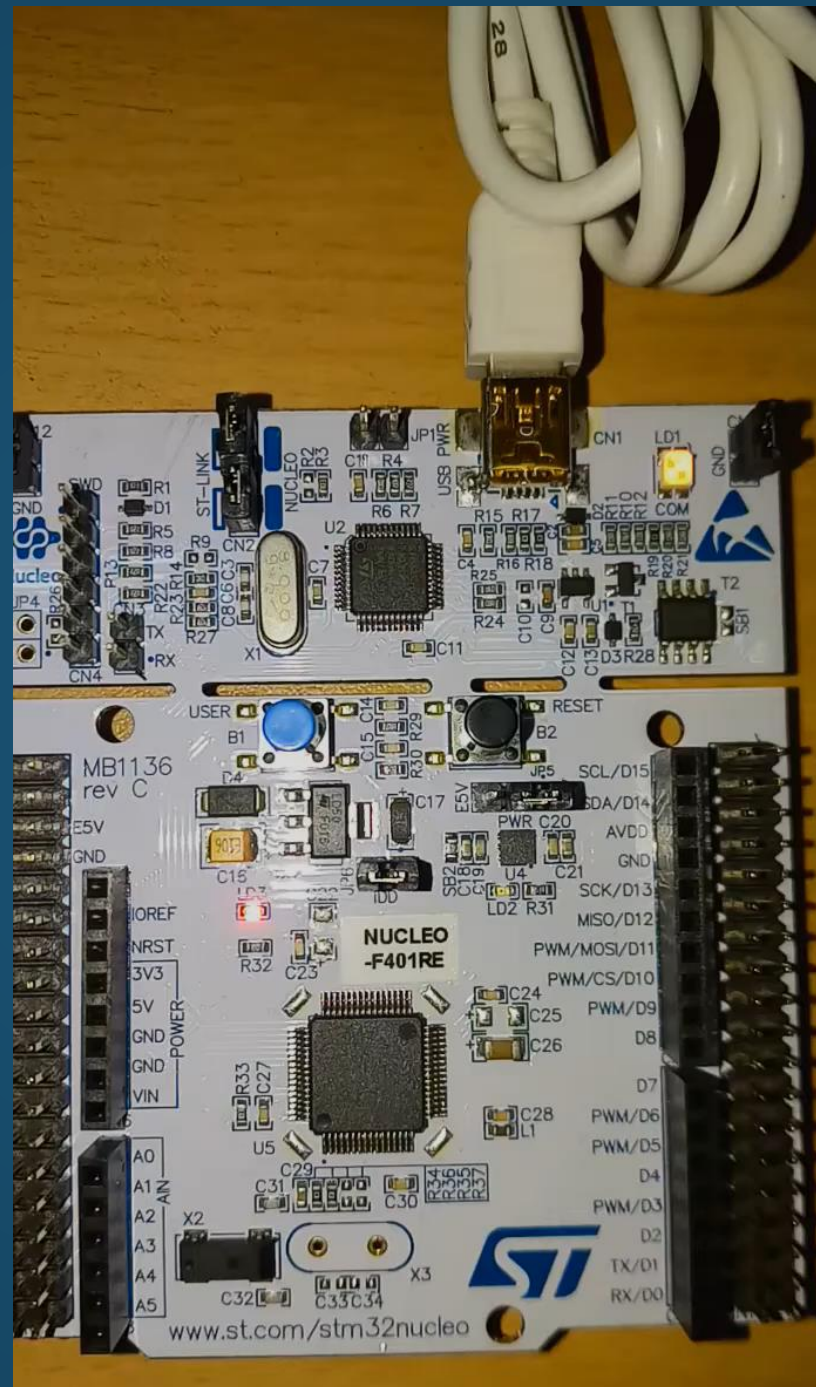
Apply



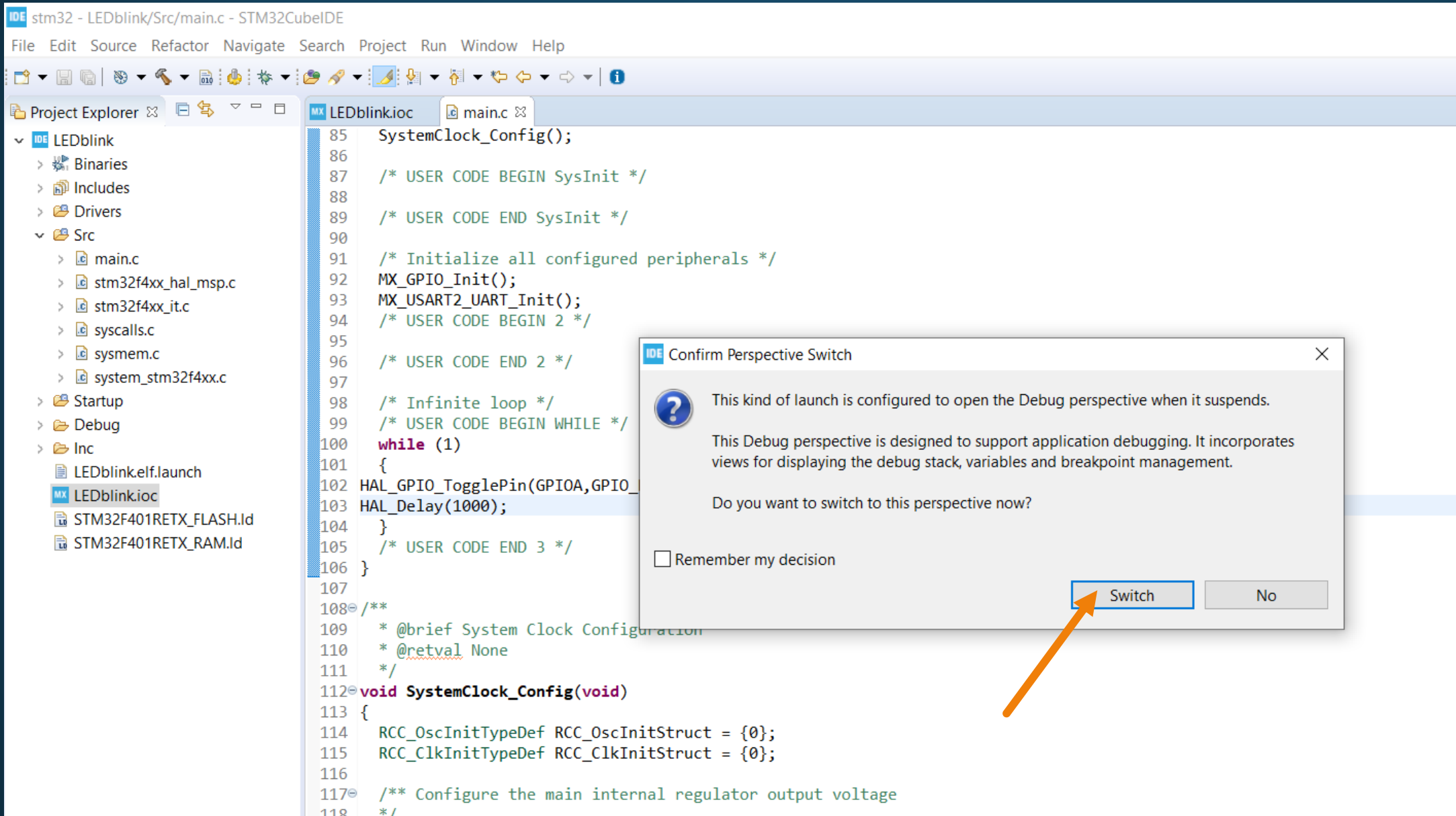
OK

Cancel

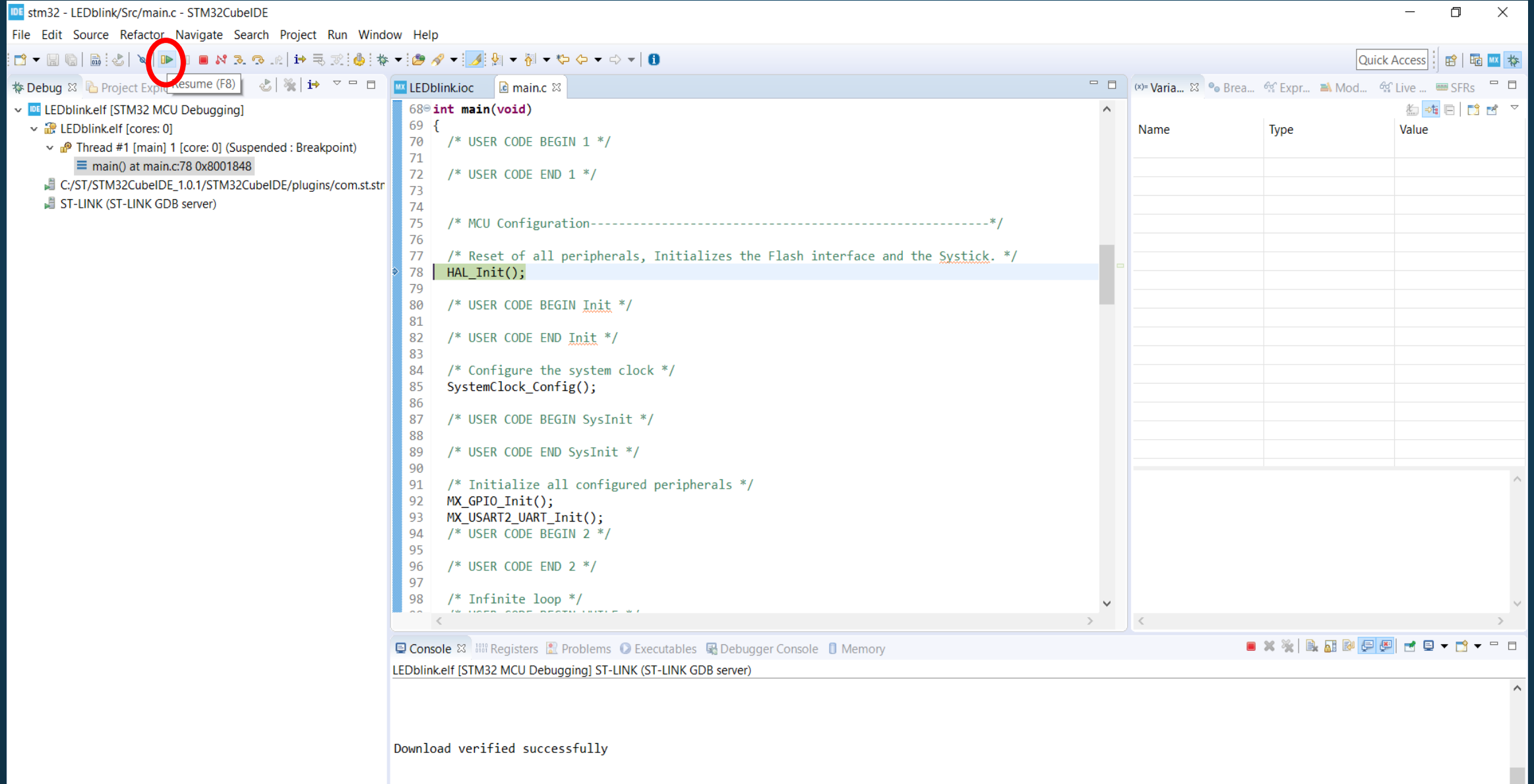
# After debugging:



# Select **Switch** to the current perspective:



# Select **Resume** and next see the LED glowing up as per program



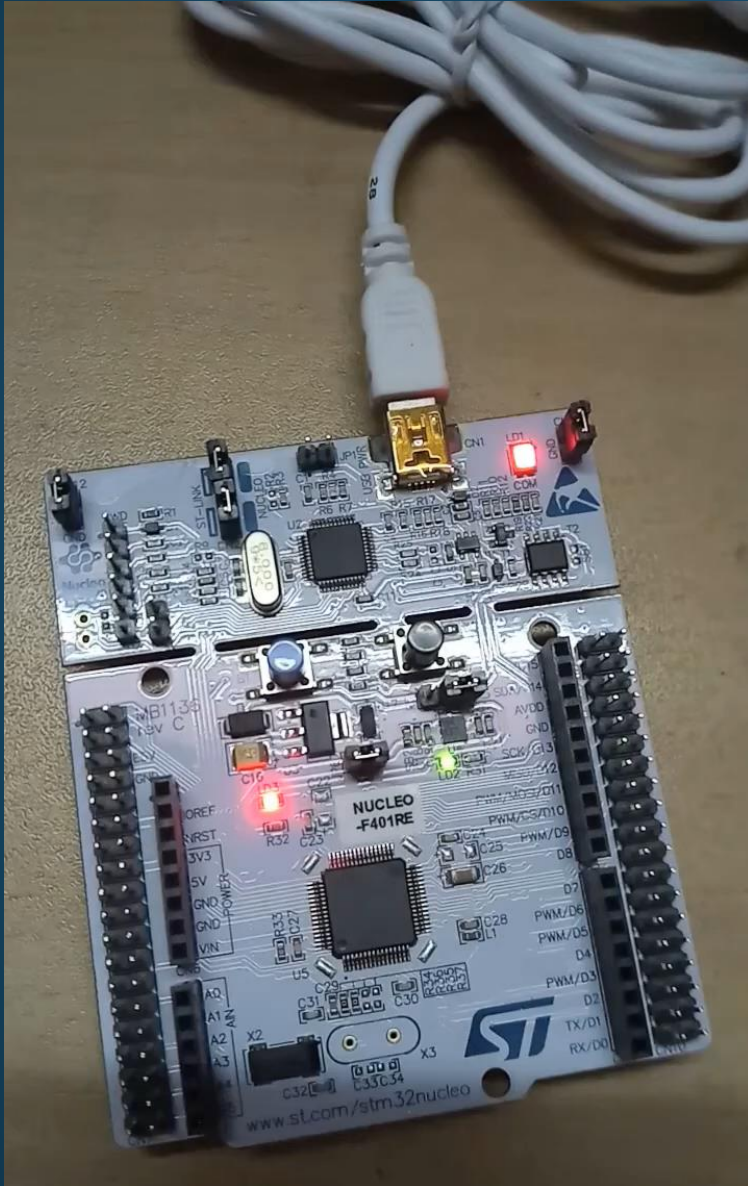
The screenshot shows the STM32CubeIDE interface. The top toolbar contains various icons, with the 'Resume' button (a green play icon) circled in red. The left sidebar shows the project structure, including 'LEDblink.elf [STM32 MCU Debugging]' and 'Thread #1 [main] 1 [core: 0] (Suspended : Breakpoint)'. The main editor window displays the 'main.c' file, which contains the following code:

```
68 int main(void)
69 {
70     /* USER CODE BEGIN 1 */
71
72     /* USER CODE END 1 */
73
74     /* MCU Configuration-----*/
75
76     /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
77     HAL_Init();
78
79     /* USER CODE BEGIN Init */
80
81     /* USER CODE END Init */
82
83     /* Configure the system clock */
84     SystemClock_Config();
85
86     /* USER CODE BEGIN SysInit */
87
88     /* USER CODE END SysInit */
89
90     /* Initialize all configured peripherals */
91     MX_GPIO_Init();
92     MX_USART2_UART_Init();
93     /* USER CODE BEGIN 2 */
94
95     /* USER CODE END 2 */
96
97     /* Infinite loop */
98     /* USER CODE BEGIN 3 */
99 }
```

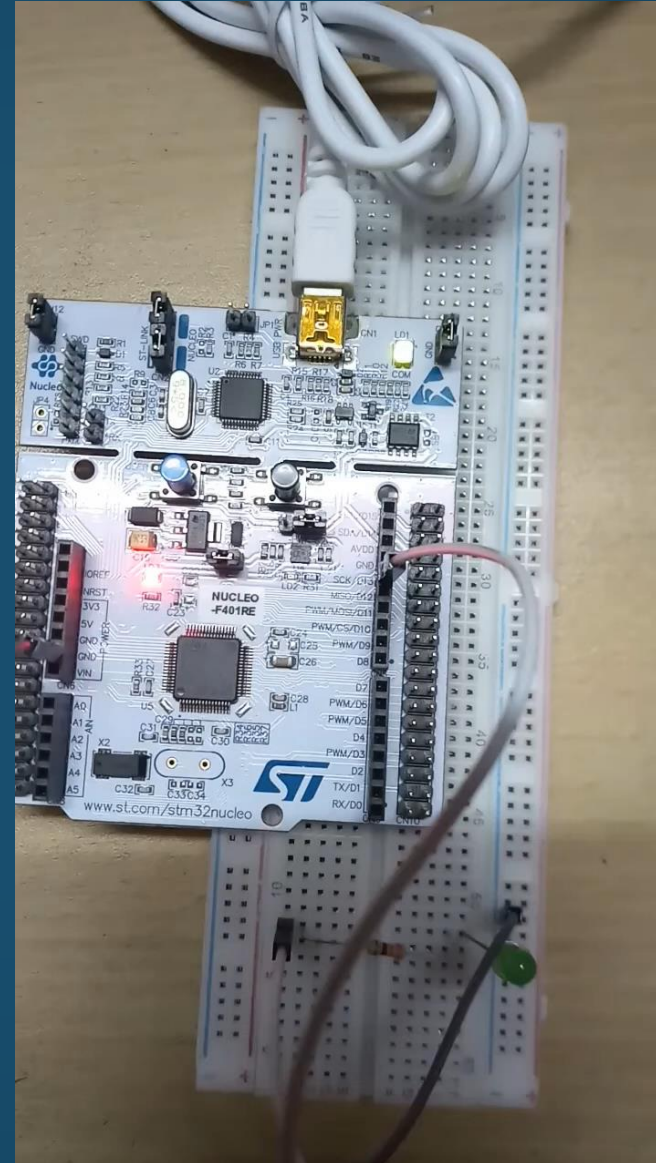
The right sidebar shows the 'Variables' window, which is currently empty. The bottom status bar indicates 'Download verified successfully'.



The built in LED blinks:



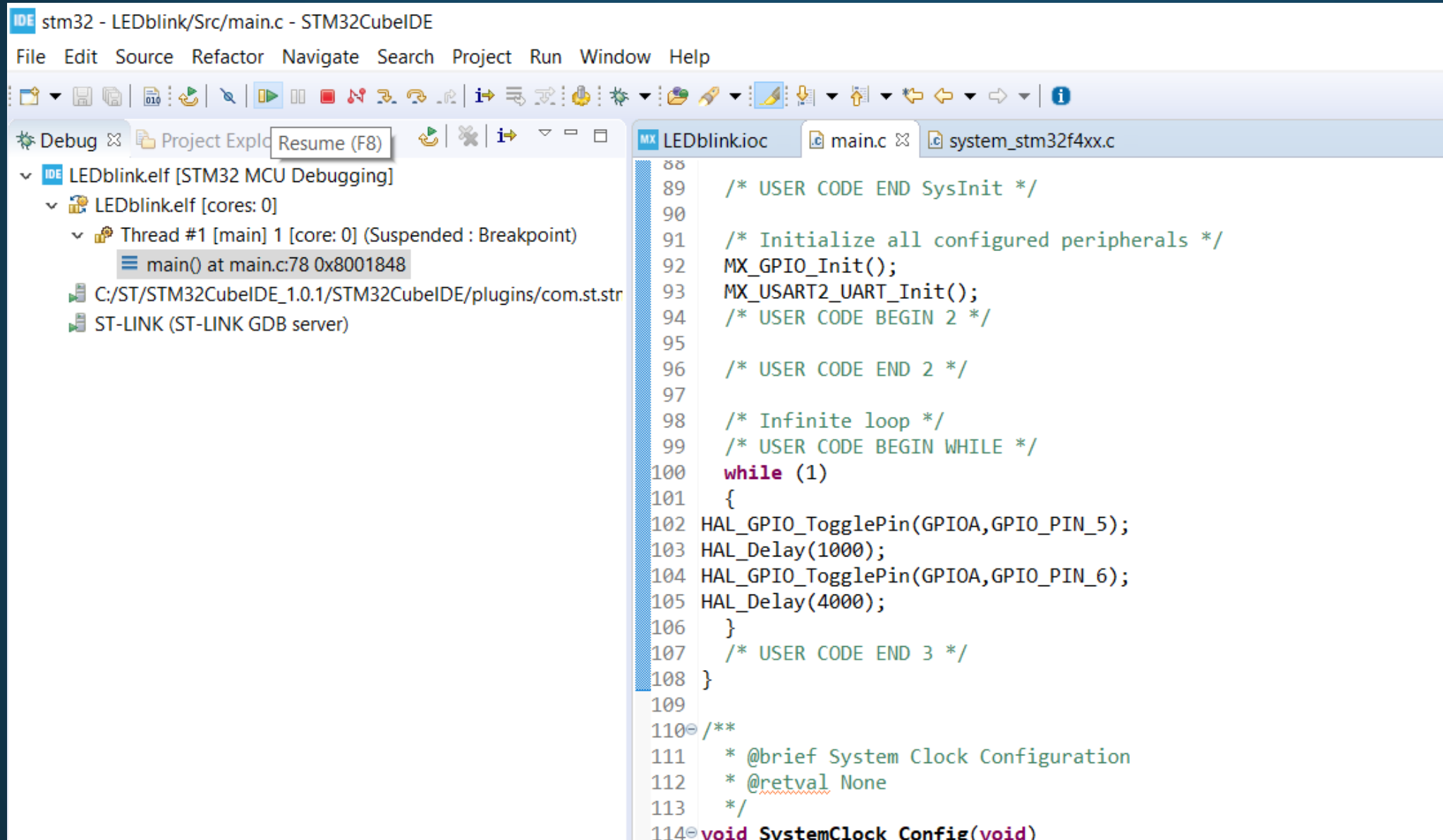
LED blinks simultaneously:







# Write the instructions as follows:



stm32 - LEDblink/Src/main.c - STM32CubeIDE

File Edit Source Refactor Navigate Search Project Run Window Help

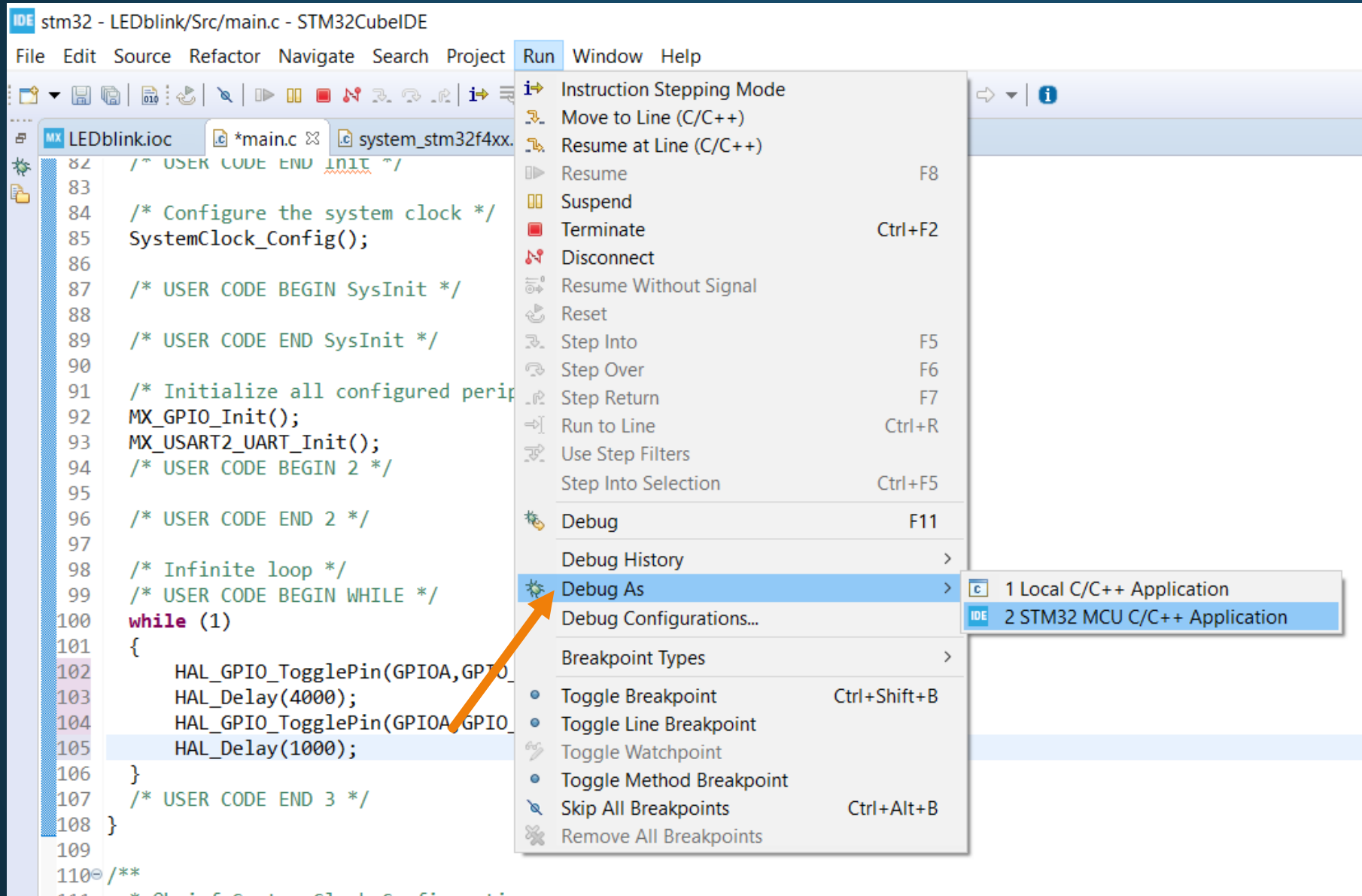
Debug Project Explorer Resume (F8)

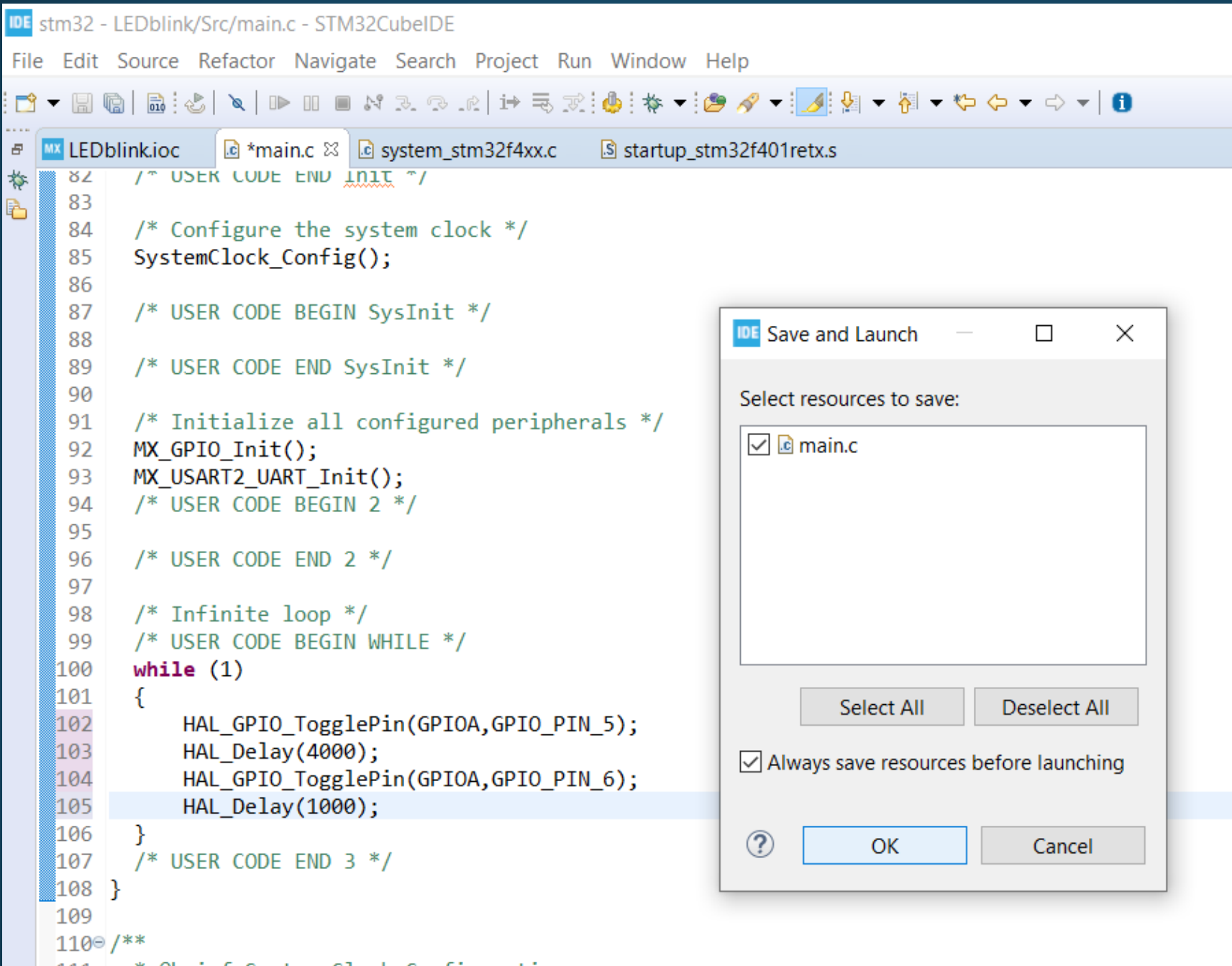
LEDblink.elf [STM32 MCU Debugging]

- LEDblink.elf [cores: 0]
  - Thread #1 [main] 1 [core: 0] (Suspended : Breakpoint)
    - main() at main.c:78 0x8001848
- C:/ST/STM32CubeIDE\_1.0.1/STM32CubeIDE/plugins/com.st.stm32cube.ide.mcu.debug.stlink
- ST-LINK (ST-LINK GDB server)

```
88
89  /* USER CODE END SysInit */
90
91  /* Initialize all configured peripherals */
92  MX_GPIO_Init();
93  MX_USART2_UART_Init();
94  /* USER CODE BEGIN 2 */
95
96  /* USER CODE END 2 */
97
98  /* Infinite loop */
99  /* USER CODE BEGIN WHILE */
100 while (1)
101 {
102 HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
103 HAL_Delay(1000);
104 HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_6);
105 HAL_Delay(4000);
106 }
107 /* USER CODE END 3 */
108 }
109
110 /**
111  * @brief System Clock Configuration
112  * @retval None
113  */
114 void SystemClock_Config(void)
```

# Then build the project and debug again as before .







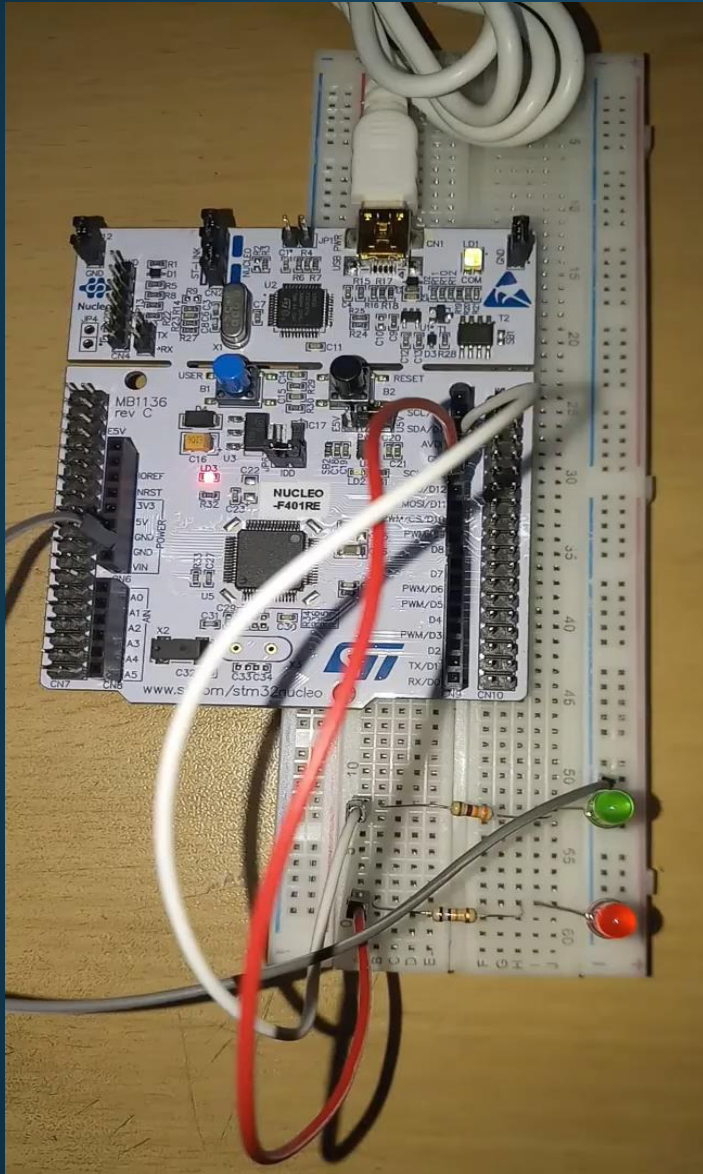
LEDblink.ioc    ma Resume (F8) tem\_stm32f4xx.c

```
94  /* USER CODE BEGIN 2 */
95
96  /* USER CODE END 2 */
97
98  /* Infinite loop */
99  /* USER CODE BEGIN WHILE */
100 while (1)
101 {
102     HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
103     HAL_Delay(4000);
104     HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_6);
105     HAL_Delay(1000);
106 }
107 /* USER CODE END 3 */
108 }
109
110 /**
111  * @brief System Clock Configuration
112  * @retval None
113  */
114 void SystemClock_Config(void)
115 {
116     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
117     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
```



# Then the LEDs will display the output accordingly:

Now, try to implement a traffic system using the basic knowledge of STM32F401RE:



# References

- <https://www.st.com/en/evaluation-tools/nucleo-f401re.html> for STM32F401RE,datasheet
- [www.st.com](http://www.st.com)
- [https://www.st.com/resource/en/user\\_manual/dm00105879-description-of-stm32f4-hal-and-ll-drivers-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/dm00105879-description-of-stm32f4-hal-and-ll-drivers-stmicroelectronics.pdf)
- [www.st.com/en/development-tools/stm32cubeide.html](http://www.st.com/en/development-tools/stm32cubeide.html)

# THANKS FOR ATTENDING....

