Toggling a LED using a push buttonin CUBE IDE

Objective:

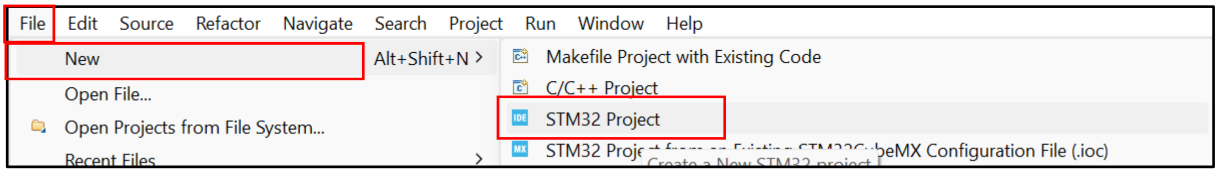
The main objective of this experiment is to understand the basics of GPIOs, how a pin can be configured as and output, input or an interrupt pin etc. By understanding these, we will configure the **PA 5** pin of the microcontroller as an **GPIO Output** pin, because the LD2 which in in-built on the microcontroller is connected to this pin. The blue user push button connected to the **PC 13** pin will be configured as**GPIO Input**. Then with the help of code we will change the state of the LED with the press of the user push button.

Requirements:

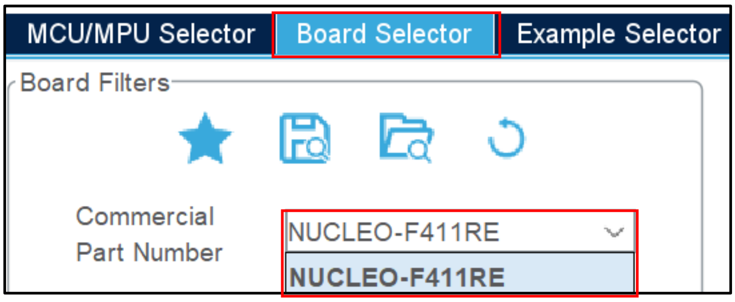
- STM32 Cube IDE software.
- STM32 Microcontroller.
- USB Cable for the microcontroller.
- PC or Laptop

Procedure:

1. Click on **File→New→STM32 Project**to start your project on Cube IDE.


File
Edit
Source
Refactor
Navigate
Search
Project
Run
Window
Help
New
Alt+Shift+N >
Open File...
Open Projects from File System...
Recent Files
Makefile Project with Existing Code
C/C++ Project
STM32 Project
STM32 Project from an Existing STM32CubeMX Configuration File (.ioc)
Create a New STM32 project

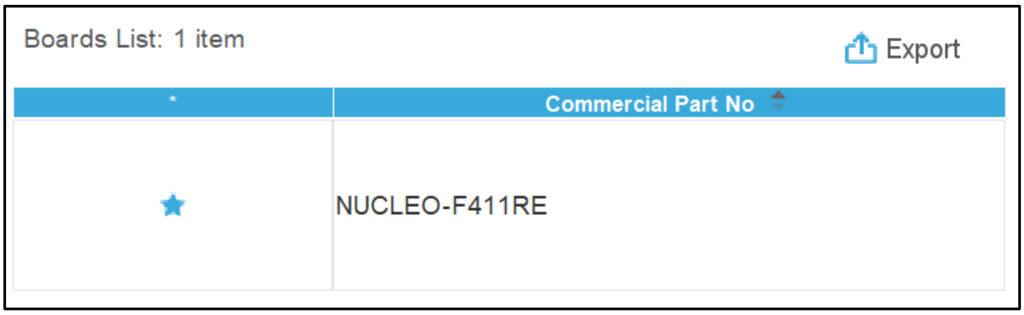
2. A **Target Selection** window will open. Click on **Board Selector**, where you need to select the microcontroller board you are working with.



Commercial Part Number: NUCLEO-F411RE

NUCLEO-F411RE

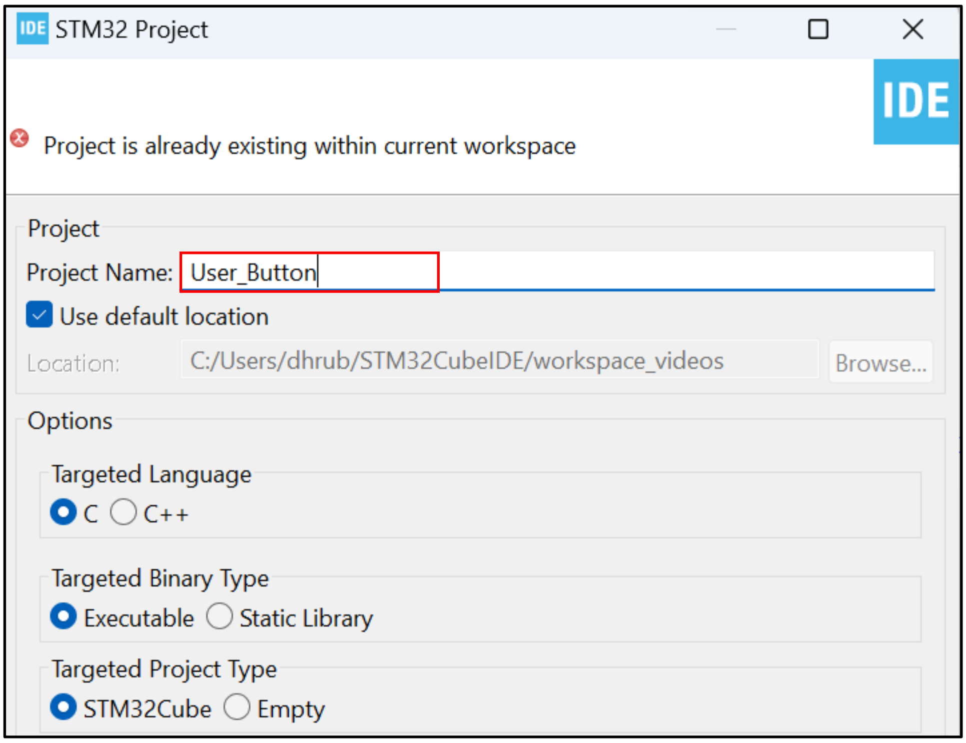
3. After this on the right-hand side of the window, under**Board List** you will see the board you have selected. Click on the board and then click on **Next.**



Boards List: 1 item

*	Commercial Part No
★	NUCLEO-F411RE

4. Give your project a name, rest of the things will remain by default as it is for now. Click on **Next.**



STM32 Project

Project is already existing within current workspace

Project

Project Name: User_Button

☑ Use default location

Location: C:/Users/dhrub/STM32CubeIDE/workspace_videos

Options

Targeted Language

● C ○ C++

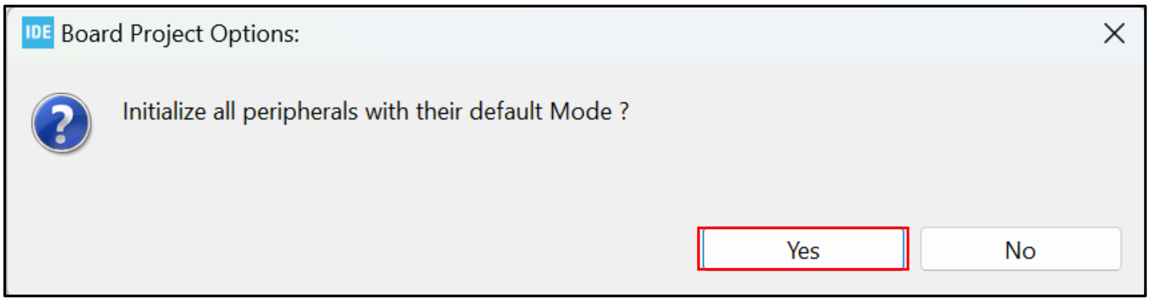
Targeted Binary Type

● Executable ○ Static Library

Targeted Project Type

● STM32Cube ○ Empty

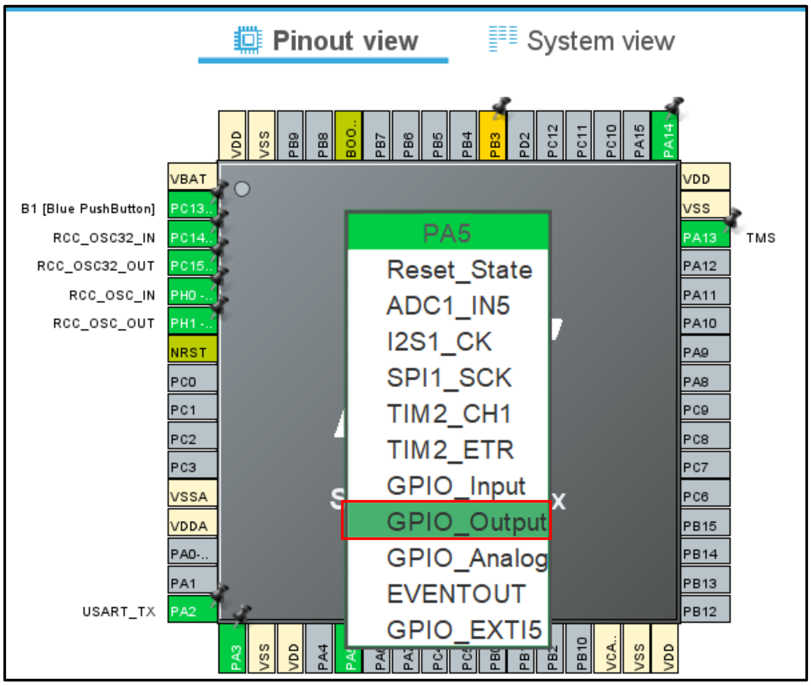
5. Cube IDE will ask if you want to initialize all peripherals with their default mode, click on **Yes.**

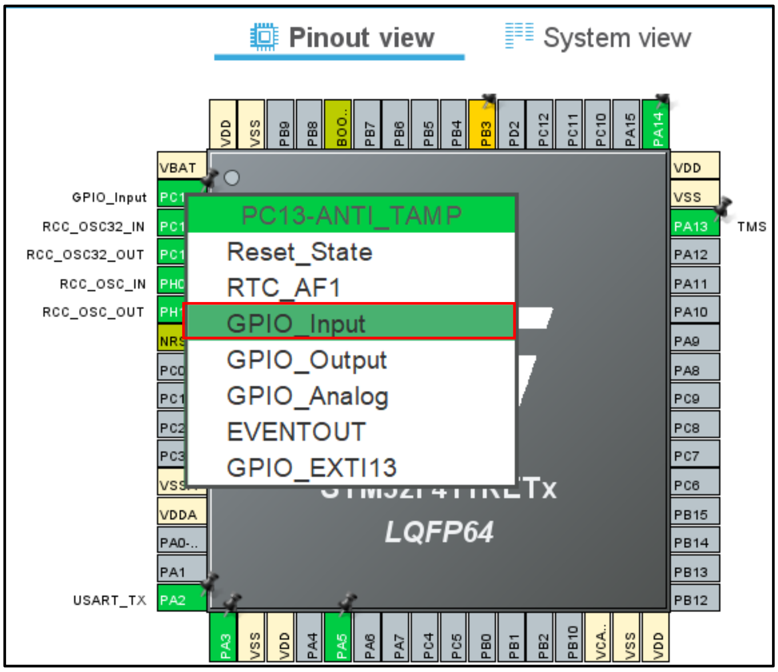


Board Project Options:

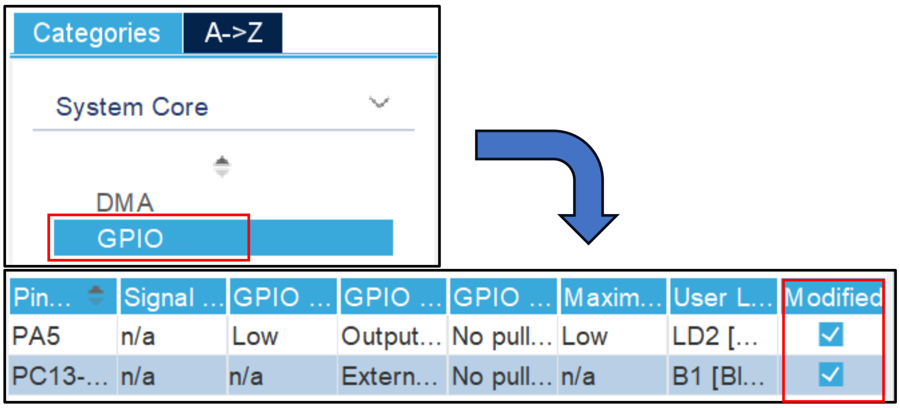
Initialize all peripherals with their default Mode ?

6. In the **Pinout& Configuration** tab, click on **PA5**pin (LED 2 is connected to this pin) and select it as **GPIO_OUTPUT**, if it is not selected by default. Next click on **PC13** pin (User Button is connected to this pin)and select it as **GPIO_Input**.



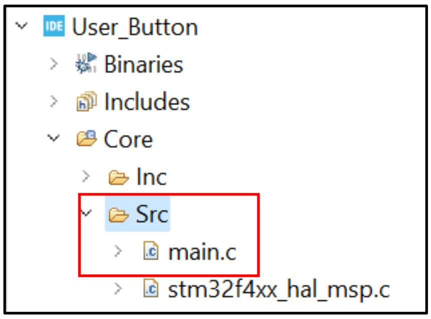


7. Next on the left-hand side under **Categories →System Core**, select GPIO to configure the GPIO functions. In the GPIO tab select the boxes in **PA5**and **PC13** as shown below.

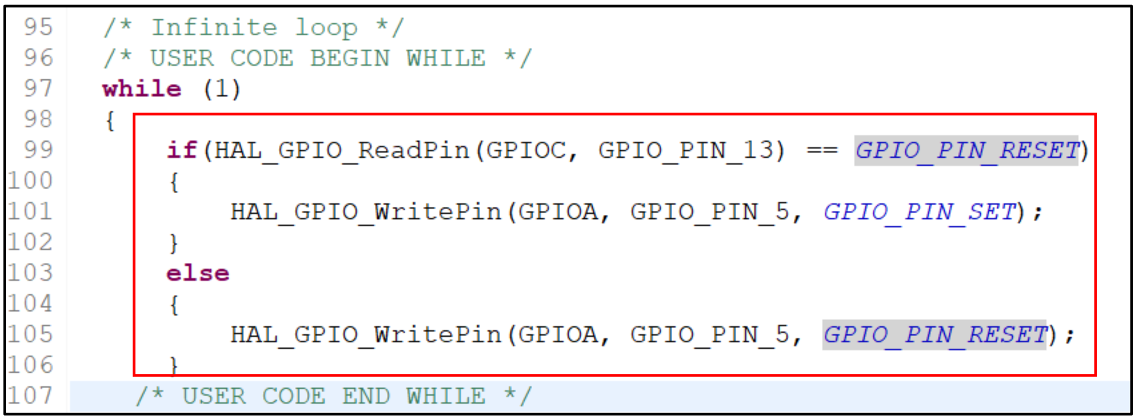


Pin...	Signal ...	GPIO ...	GPIO ...	GPIO ...	Maxim...	User L...	Modified
PA5	n/a	Low	Output...	No pull...	Low	LD2 [...	☑
PC13-...	n/a	n/a	Extern...	No pull...	n/a	B1 [Bl...	☑

8. Press Ctrl+Sto generate your code. On the left-hand side of the Cube IDE, under **Project Explorer** go to the project you have created (For example I have named my project as User_Button)**User_Button→Core →Src→main.c (**double click to load the code).

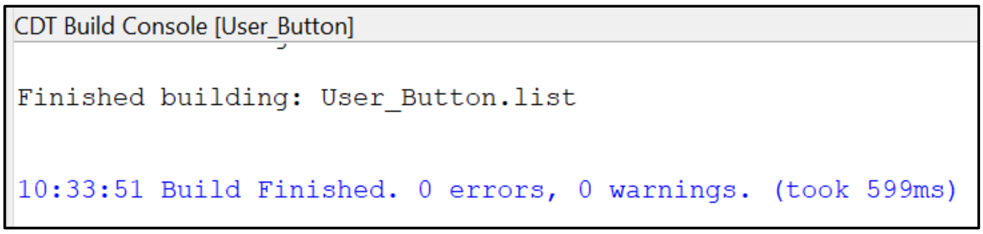


9. Cube IDE automatically generates a code format based on the configurations you have done. Cube IDE used HAL libraries. Below are the code snippets, put your code in the appropriate places in the **main.c** file.



```
/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
    if(HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET)
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
    }
    else
    {
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
    }
  /* USER CODE END WHILE */
```

10. Now click on the build 🔨▾ symbol on the top left corner on your Cube IDE. If you have done everything correctly your code should be built without any errors.



CDT Build Console [User_Button]

```
Finished building: User_Button.list

10:33:51 Build Finished. 0 errors, 0 warnings. (took 599ms)
```

11. Next connect your STM32 board to your PC and click on the **Debug** icon to start the Debugging process. An**Edit Configuration** window will open, click on **OK,** without making any changes.
12. In debugging mode click on the **Resume** icon to run your code. You should be able to see the led on-board your STM32 get **ON** when you press the **User Button**, and **OFF** when you release it.
13. To move out of the debugging mode, click on the **Terminate** icon . You will be moved out of the debugging mode.

Note: All important steps and parts are highlighted with a red colour box for the proper understanding of the user. This document is for the use of education purpose only.