

Toggling a LED using interrupt in CUBE IDE

Objective:

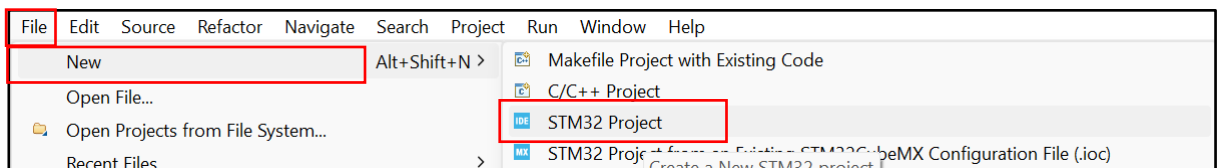
The main objective of this experiment is to understand the basics of GPIOs, how a pin can be configured as and output, input or an interrupt pin etc. By understanding these, we will configure the **PA 5** pin of the microcontroller as an **GPIO Output** pin, because the LD2 which is in-built on the microcontroller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



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

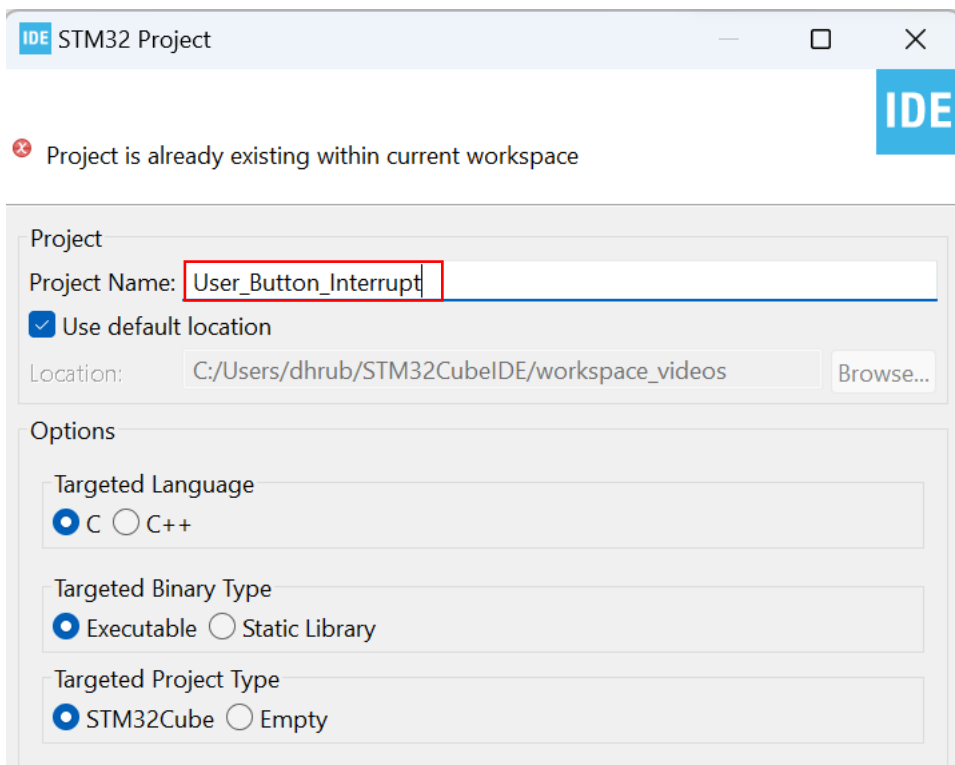


3. After this on the right-hand side of the window, under **Board List** you will see the board you have selected. Click on the board and then click on **Next**.

Boards List: 1 item Export

	Commercial Part No
	NUCLEO-F411RE

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



IDE STM32 Project

Project is already existing within current workspace

Project

Project Name: User_Button_Interrupt

☒ Use default location

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

Options

Targeted Language

☒ C ☐ C++

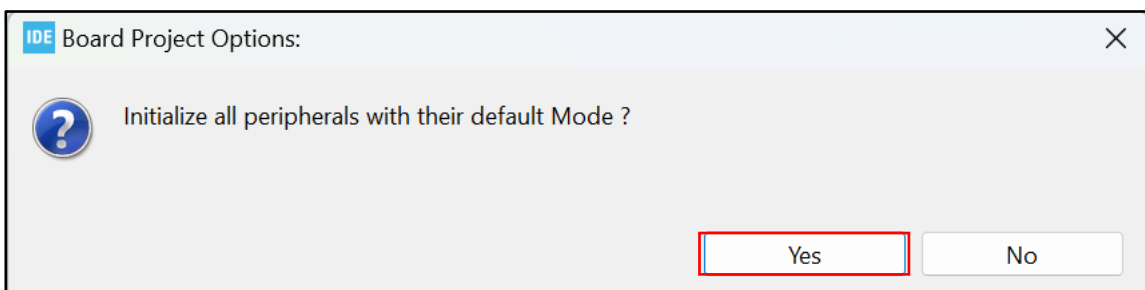
Targeted Binary Type

☒ Executable ☐ Static Library


Targeted Project Type

☒ STM32Cube ☐ Empty

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

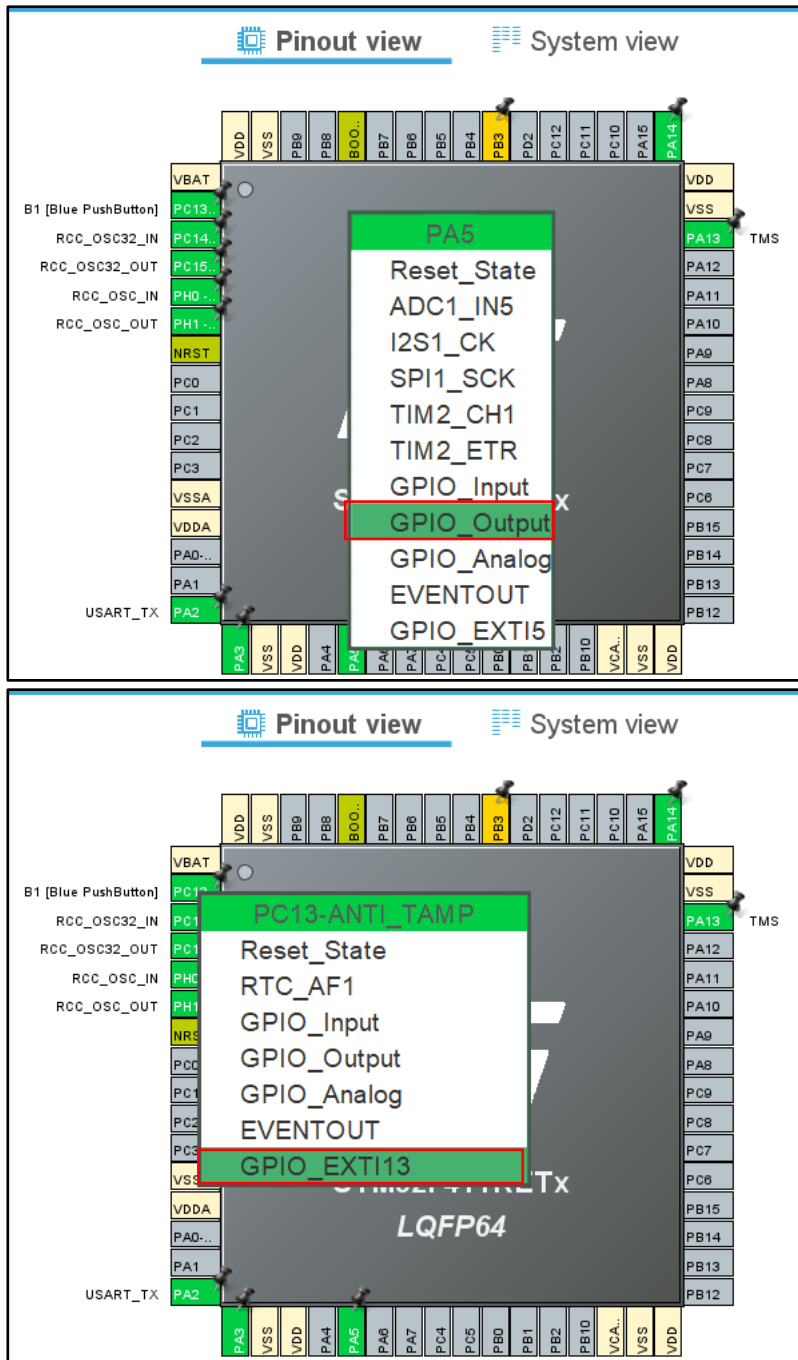


IDE Board Project Options:

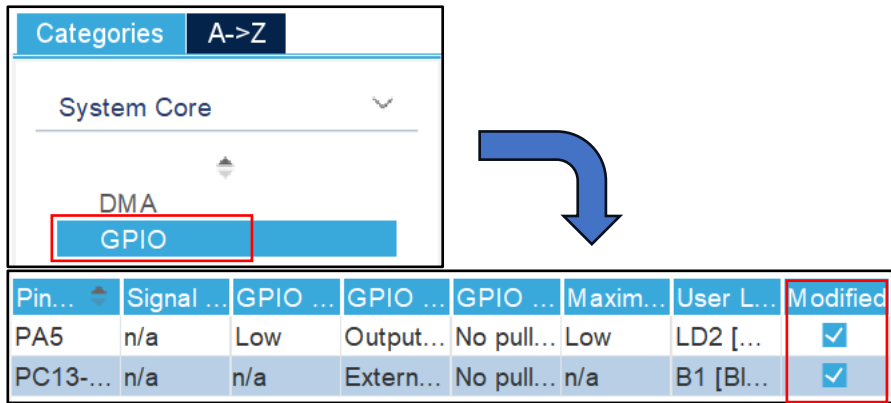
 Initialize all peripherals with their default Mode ?

Yes No

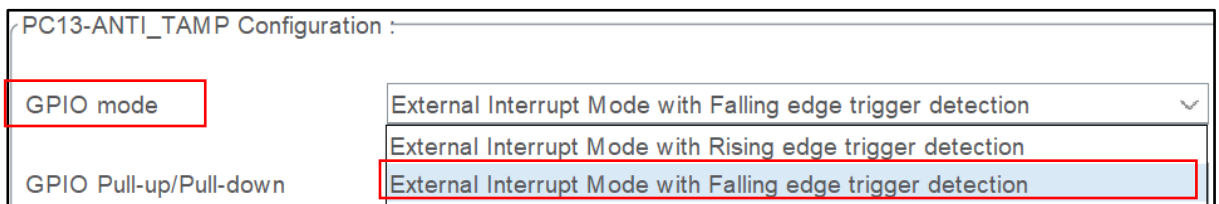
- In the **Pinout& Configuration** tab, click on **PA5**pin (LED 2 is connected to this pin) and select it as **GPIO_OUTPUT**, if it is not selected by default. Next click on **PC13** pin (User Button is connected to this pin) and select it as **GPIO_EXTI13**.



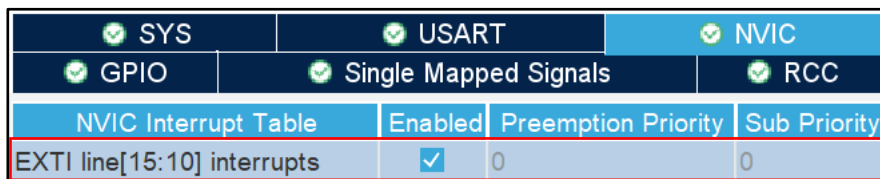
7. Next on the left-hand side under **Categories** → **System Core**, select GPIO to configure the GPIO functions. In the GPIO tab select the boxes in **PA5** and **PC13** as shown below.



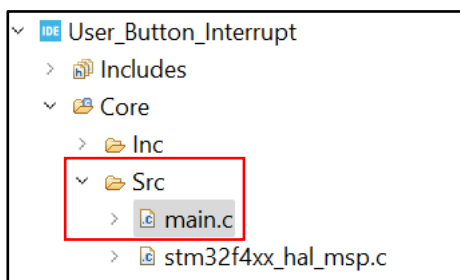
8. Next click on **PC13** and in **GPIO mode** select **External Interrupt Mode with Falling edge trigger detection** from the list of options. Let everything else remain as default for now.



9. Next go to **NVIC** settings and enable the **EXTI line [15:10] interrupts** line.







10. Press **Ctrl+S** to 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_Interrupt) **User_Button_Interrupt** → **Core** → **Src** → **main.c** (double click to load the code).



11. 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.

```
57 /* Private user code -----  
58 /* USER CODE BEGIN 0 */  
59 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)  
60 {  
61     HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);  
62 }  
63 /* USER CODE END 0 */
```

12. 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.
13. 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.
14. In debugging mode click on the **Resume** icon , to run your code. You should be able to see the led on-board your STM32 changing with every press of the **User Button**.
15. 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.