Toggling a LED using Timers

Objective:

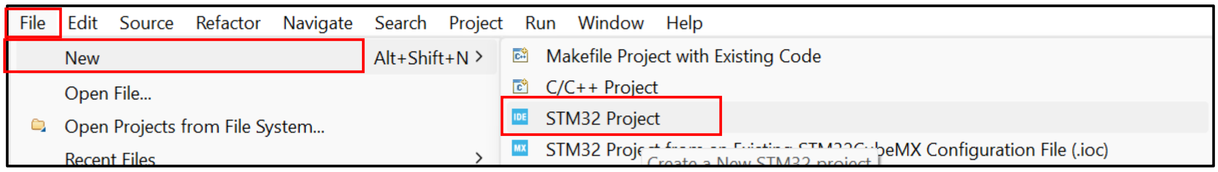
The main objective of this experiment is to understand the basics of GPIOs, how a pin can be configured as an output or an input pin etc. By understanding these, we will configure the **PA 5** pin of the microcontroller as a timer channel pin, because the LD2, which is in-built on the microcontroller, is connected to this pin. Then with the help of timers we will change the state of the LED.

Requirements:

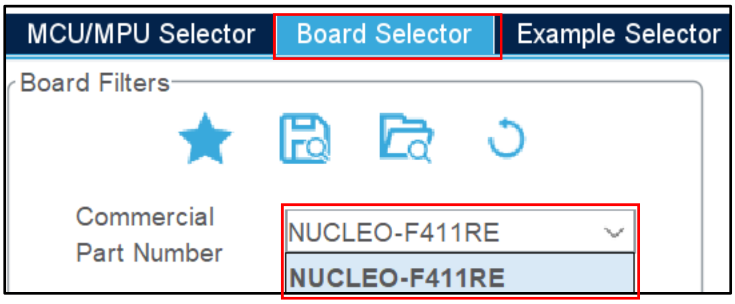
- STM32 Cube IDE software.
- STM32 Microcontroller.
- USB Cable for the microcontroller.
- PC or Laptop.

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

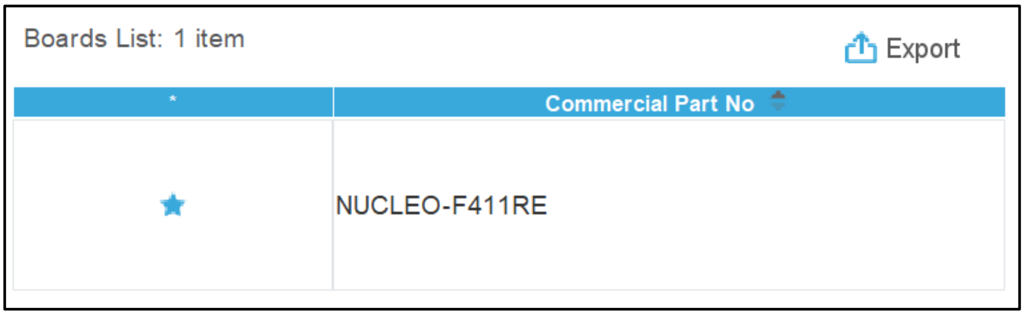
2. A **Target Selection** window will open. Click on **Board Selector**, where you need to select the microcontroller board you are working with.



Commercial Part Number: NUCLEO-F411RE

NUCLEO-F411RE

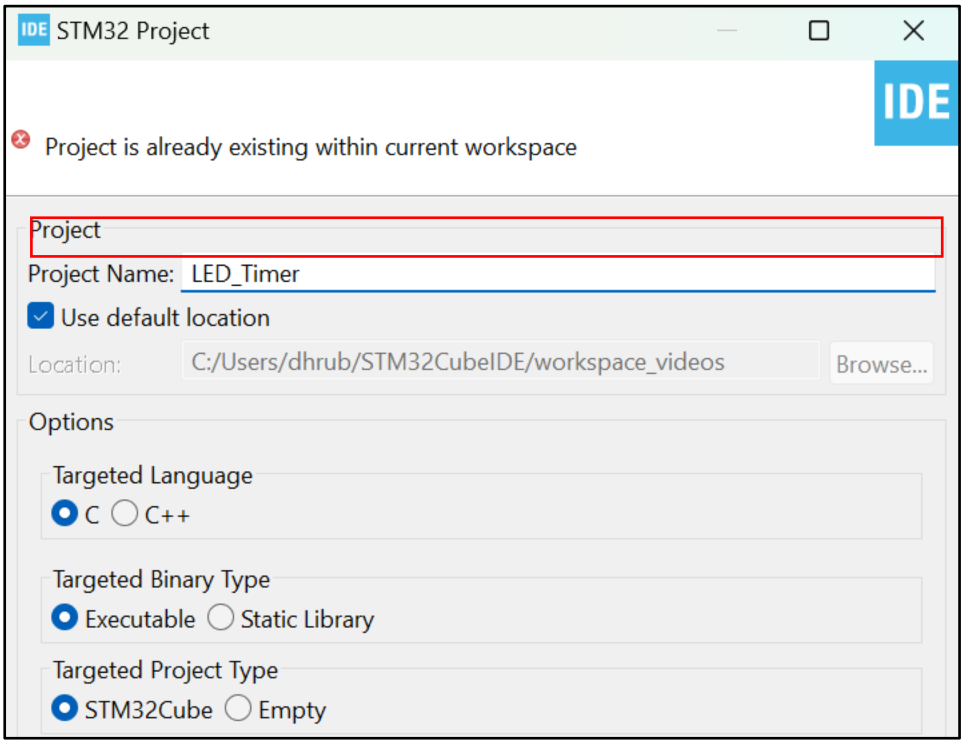
3. After this on the right-hand side of the window, under **Board List** you will see the board you have selected. Click on the board and then click on **Next.**



Boards List: 1 item

*	Commercial Part No
★	NUCLEO-F411RE

4. Give your project a name, rest of the things will remain as default. Click on **Finish.**



STM32 Project

Project is already existing within current workspace

Project

Project Name: LED_Timer

☑ Use default location

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

Options

Targeted Language

● C ○ C++

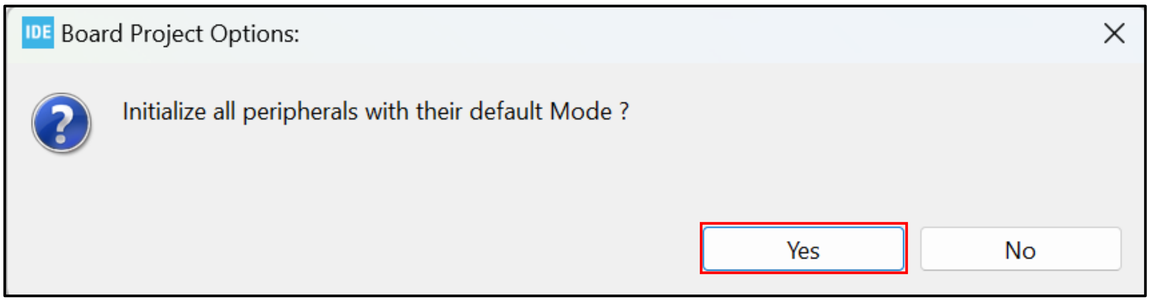
Targeted Binary Type

● Executable ○ Static Library

Targeted Project Type

● STM32Cube ○ Empty

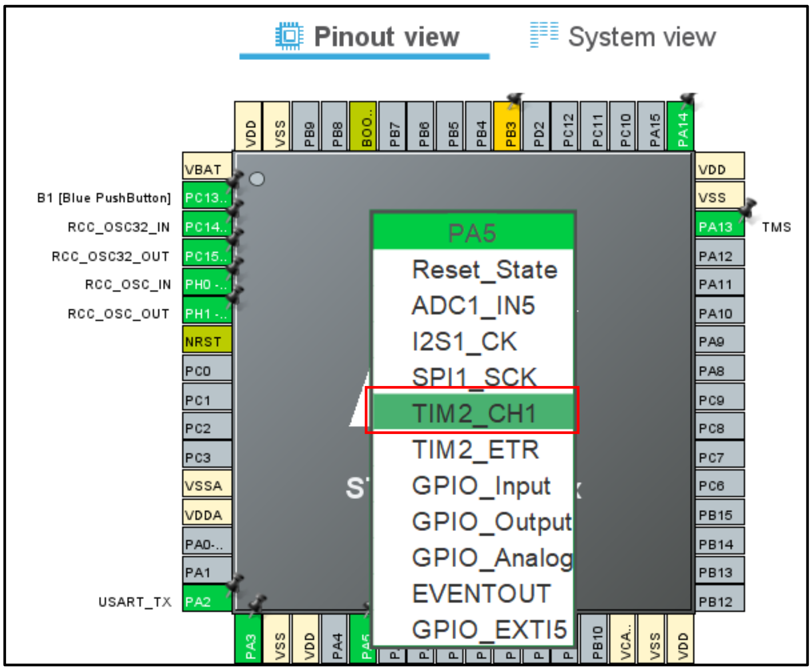
5. Cube IDE will ask if you want to initialize all peripherals with their default mode, click on **Yes.**



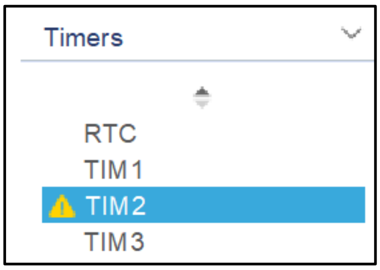
Board Project Options:

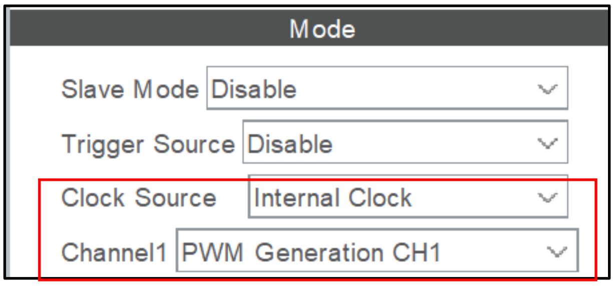
Initialize all peripherals with their default Mode ?

6. In the **Pinout & Configuration** tab, click on **PA5** pin and select it as **TIM2_CH1**.



7. Next go to **Timer**, select **TIM2** and select **Clock Source** as **Internal Clock**, and **Channel 1** as **PWM Generation CH1**. Under **Configuration** update the **Prescaler** as**124** and **Counter Period** value as **63999**. Next update the value of **Pulse** as **32000**.



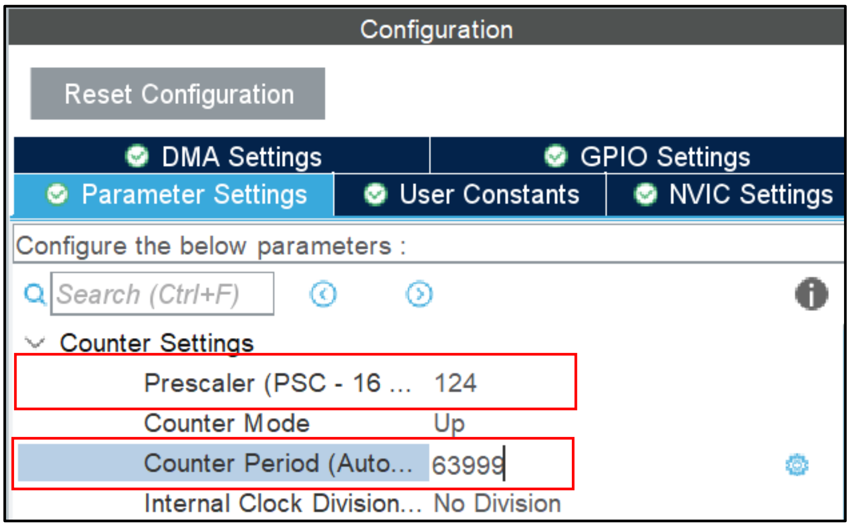

Mode

Slave Mode Disable

Trigger Source Disable

Clock Source Internal Clock

Channel1 PWM Generation CH1



Configuration

Reset Configuration

DMA Settings | GPIO Settings | Parameter Settings | User Constants | NVIC Settings

Configure the below parameters :

Search (Ctrl+F)

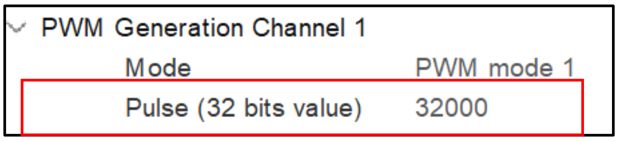
Counter Settings

Prescaler (PSC - 16 ... 124

Counter Mode Up

Counter Period (Auto... 63999

Internal Clock Division... No Division

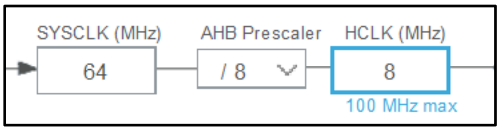


PWM Generation Channel 1

Mode PWM mode 1

Pulse (32 bits value) 32000

8. Go to **Clock Configuration**, enter 8 as a value in **HCLK** and press enter to configure the peripheral and timer clocks.



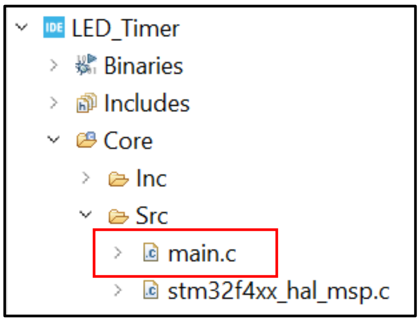
SYSCLK (MHz) 64

AHB Prescaler / 8

HCLK (MHz) 8

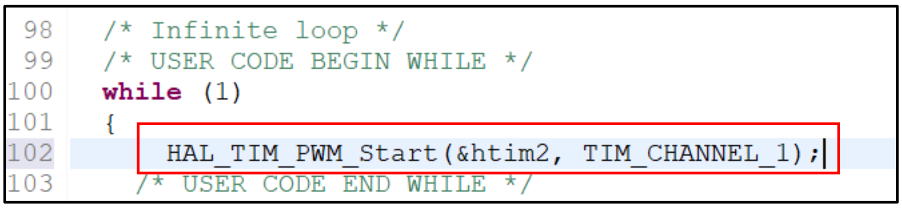
100 MHz max

9. Press C**trl+S** to generate your code or you can go to **Project →Generate 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 LED_Timer) **LED_Timer→Core →Src →main.c (**double click to load the code).



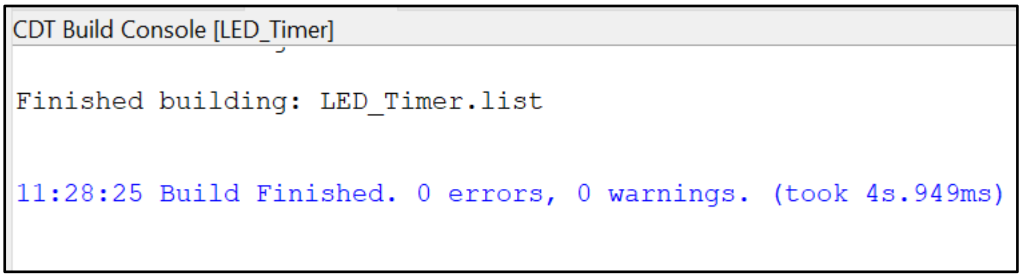
```
LED_Timer
  Binaries
  Includes
  Core
    Inc
    Src
      main.c
      stm32f4xx_hal_msp.c
```

10. 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)
  {
      HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1);
    /* USER CODE END WHILE */
```

11. 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 [LED_Timer]

```
Finished building: LED_Timer.list


11:28:25 Build Finished. 0 errors, 0 warnings. (took 4s.949ms)
```

12. 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.
13. In debugging mode click on the **Resume** icon to run your code. You should be able to see the led on-board your STM32 board blinking.
14. To move out of the debugging mode, click on the **Terminate** icon . You will be moved out of the debugging mode. Congratulations you have just completed your first project on STM32 Cube IDE!

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.