STM32G4 Technique Training

life.augmented

STM32G4 hands-on
LAB2

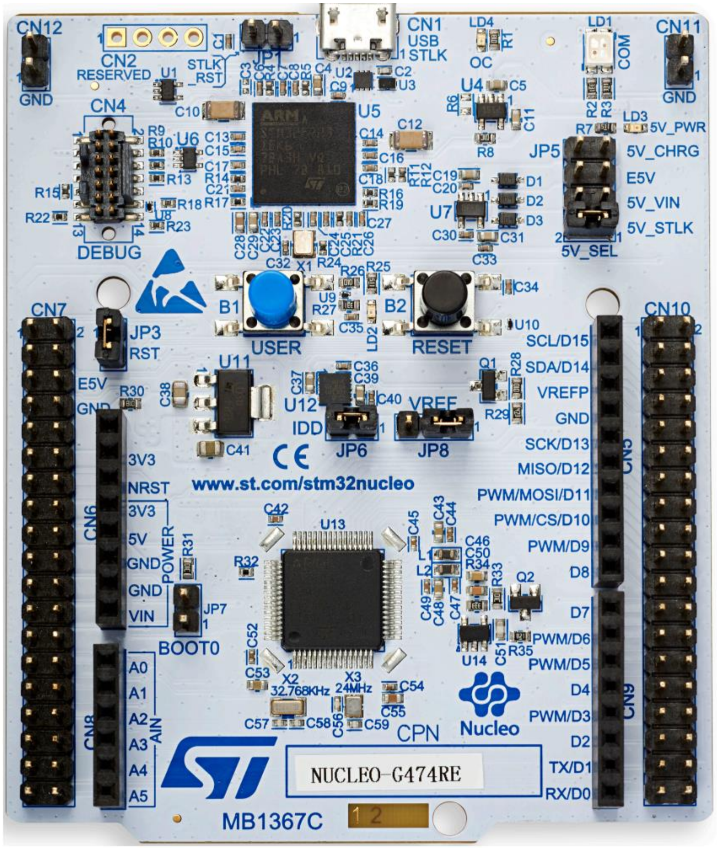
STM32G4 hands-on

- Objective
  - Toggle the LED of the NUCLEO-G474RE
  - Control LED by EXTI
  - Enable TIM3 feature : 1Hz interrupt and control LED

- Step 1
  - Create a project in STM32CubeMX
    - Configure Clocks, GPIO
  - Generate the CubeIDE project and initialization code
  - Add the user code, compile and run

- Step 2
  - Enable GPIO-EXTI
  - Generate, compile again and run

- Step 3
  - Enable TIM3 and global interrupt
  - Generate, compile again and run



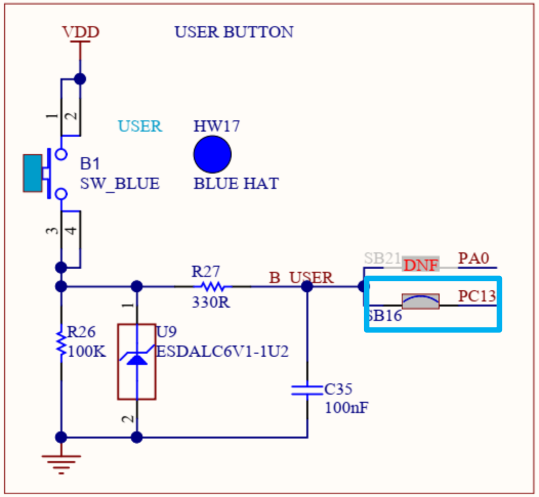
LAB2 GPIO

NUCLEO-G474RE Board

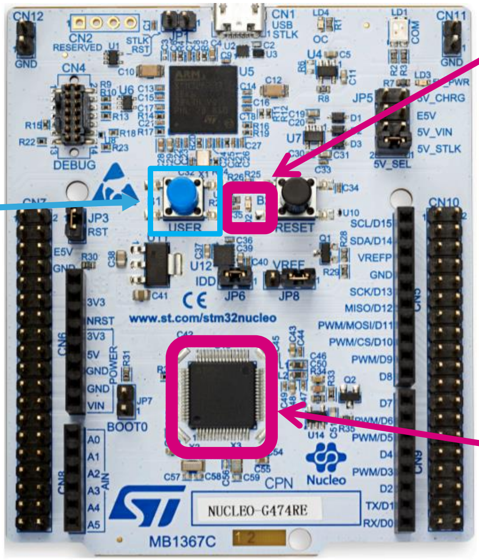
- MCU – **STM32G474RET6**

- LED
  - LD2 (**GREEN**) – GPIO **PA5** pin

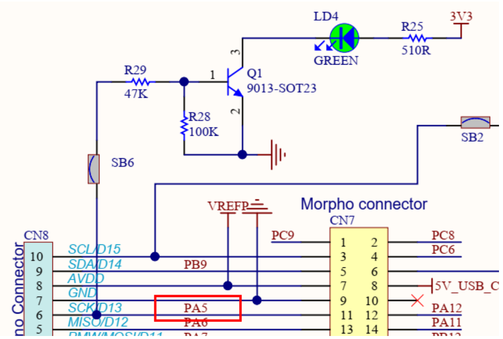
- USER button
  - GPIO **PC13** pin



User button



Green user LED

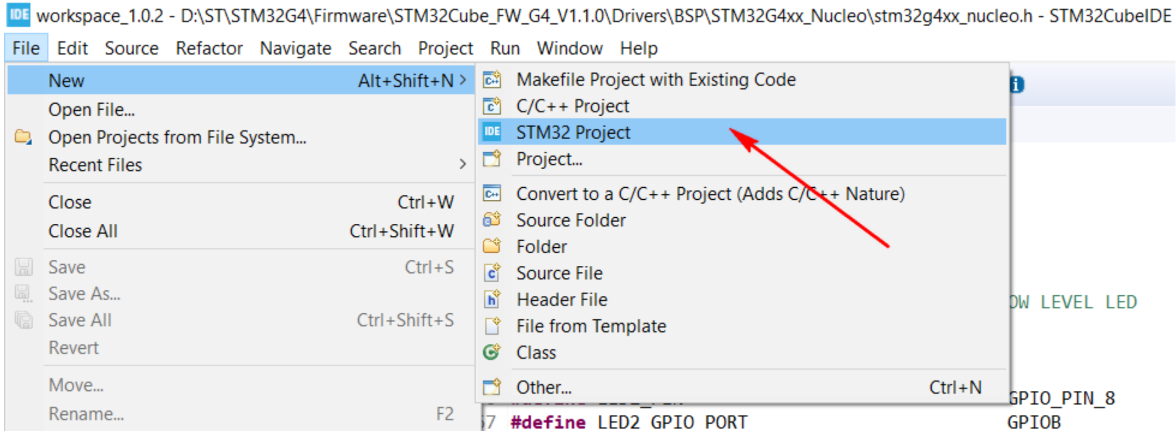


STM32G4 MCU

- ARM Cortex-M4 core 170MHz
- 512KBytes of Flash
- 128 KB of RAM

Create a new project(1/2)

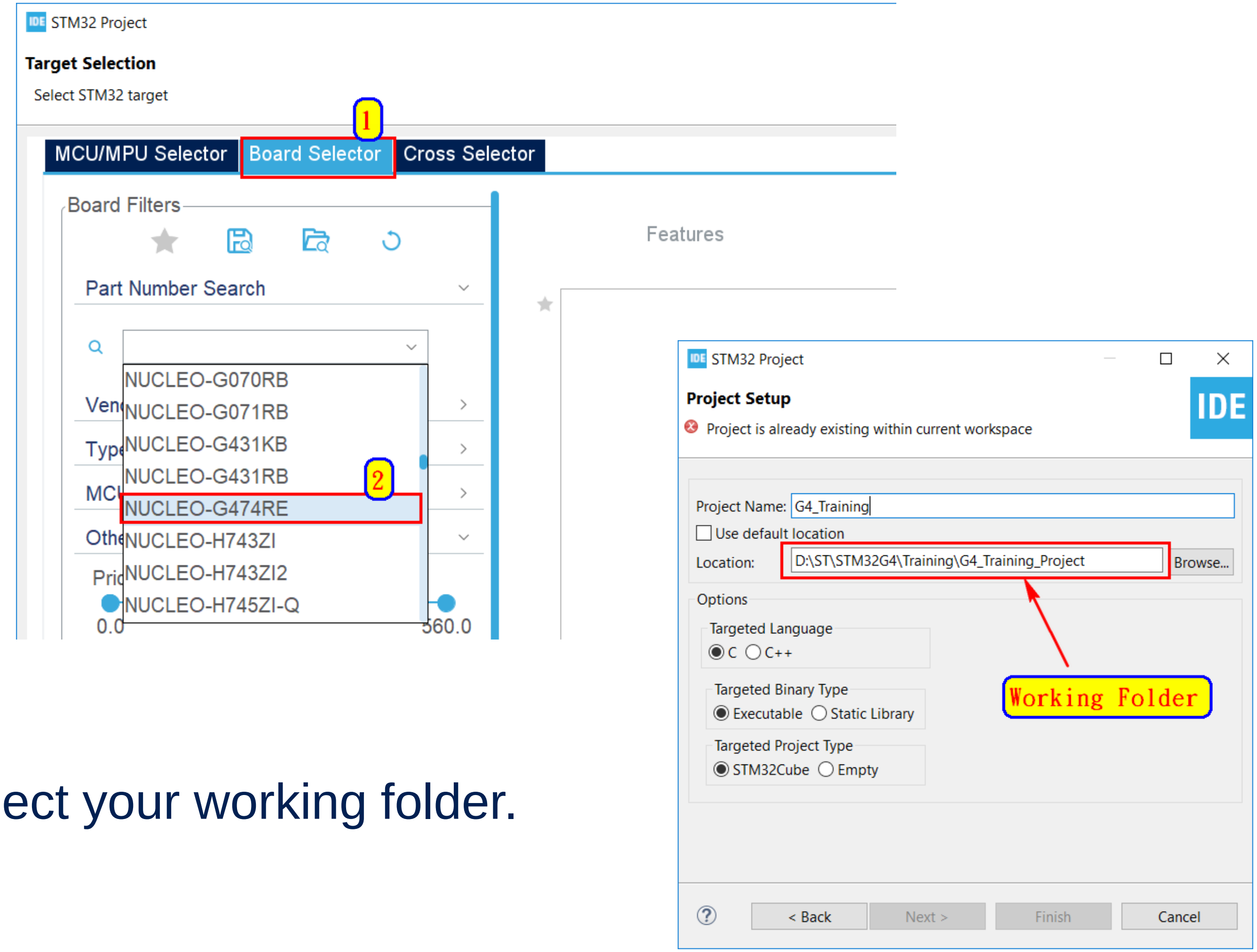
- New a STM32 Project.



```
OW LEVEL LED
GPIO_PIN_8
#define LED2 GPIO PORT                                 GPIOB
```

Create a new project(2/2)

- Selector NUCLEO-G474RE Board for use



STM32 Project

Target Selection

Select STM32 target

- NUCLEO-G070RB
- NUCLEO-G071RB
- NUCLEO-G431KB
- NUCLEO-G431RB
- NUCLEO-G474RE
- NUCLEO-H743ZI
- NUCLEO-H743ZI2
- NUCLEO-H745ZI-Q

STM32 Project

Project Setup

Project is already existing within current workspace

Project Name: G4_Training

□ Use default location

Location: D:\ST\STM32G4\Training\G4_Training_Project

Options

Targeted Language: ● C ○ C++

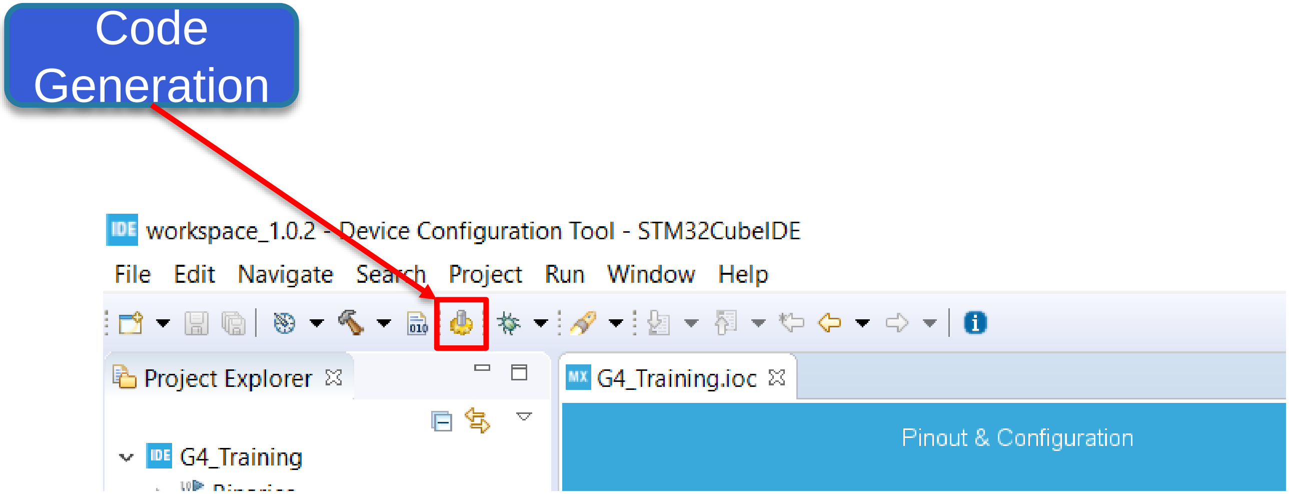
Targeted Binary Type: ● Executable ○ Static Library

Targeted Project Type: ● STM32Cube ○ Empty

- Select your working folder.

Generate Code

- STM32CubeIDE
  - Click “Code Generation” button to generate source code.


Code Generation
workspace_1.0.2 - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
Project Explorer
G4_Training
G4_Training.ioc
Pinout & Configuration

Add user code

- Add application code in porjcet
  - Open main.c then add HAL_GPIO_TogglePin( ) and HAL_Delay( )API in Infinite loop.

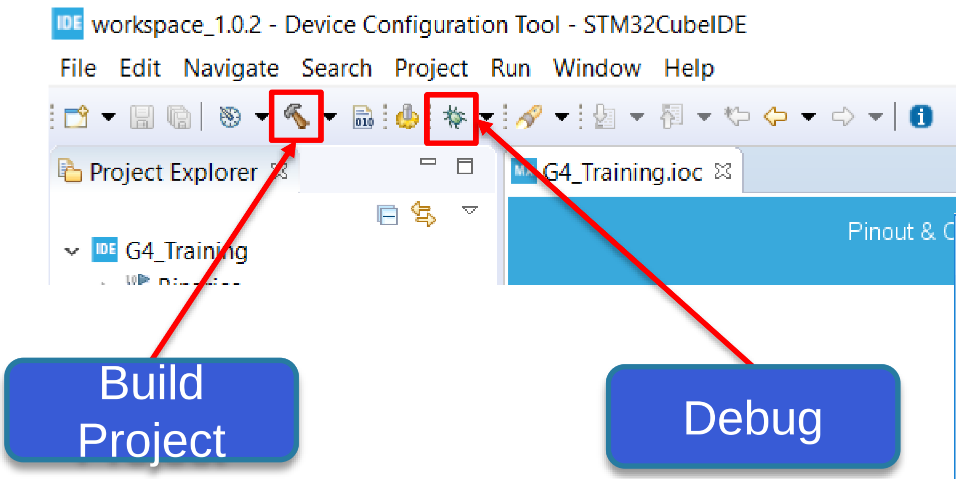
- Browse the main.c file to review the STM32CubeMX generated initialization code.
- Add the user code to toggle GPIOs driving the LED to the specific **USER CODE BEGIN \ END** section in the main.c
- Make sure to save the changes.

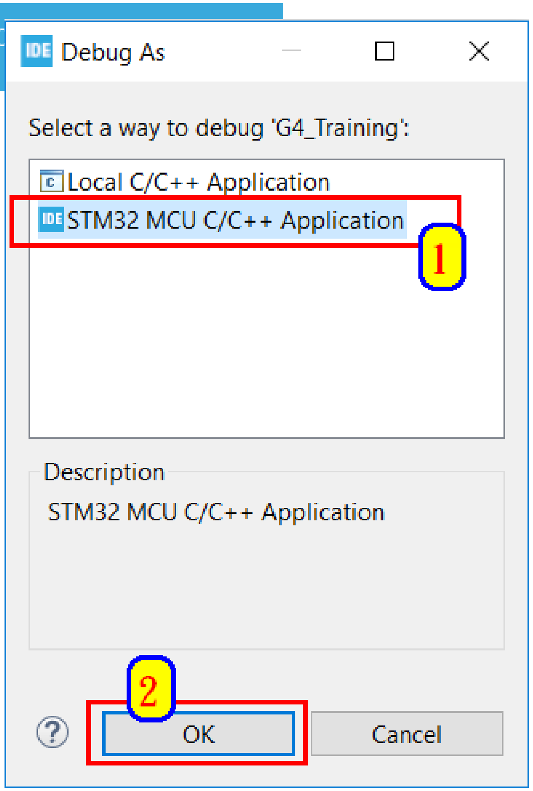
```
/* Infinite loop */
 /* USER CODE BEGIN WHILE */
 while (1)
 {
   /* USER CODE END WHILE */

   /* USER CODE BEGIN 3 */
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
   /* Insert delay 100 ms */
   HAL_Delay(100);

 }
 /* USER CODE END 3 */
```

life.augmented

Build and Debug project

- STM32CubeIDE
  - Build the project by click “Make” and then “Download and Debug”.


workspace_1.0.2 - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
Project Explorer
G4_Training
G4_Training.ioc
Pinout & C
Build Project
Debug



Select a way to debug 'G4_Training':

STM32 MCU C/C++ Application

- Selet STM32 Application for debug

LAB2 GPIO-EXTI

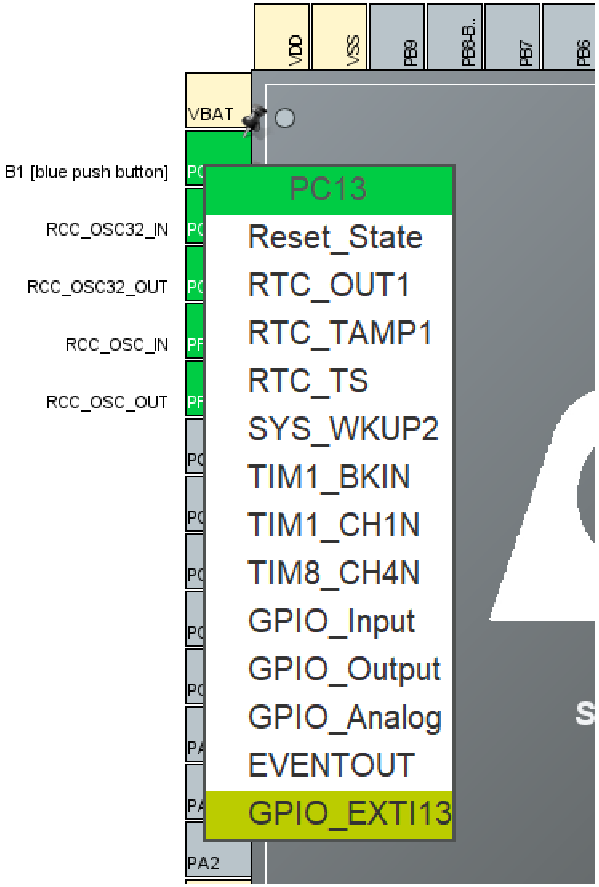
Regenerating the initialization code

- It is possible to regenerate the initialization code via the STM32CubeMX tool

- Open the STM32CubeMX project if you already closed it.

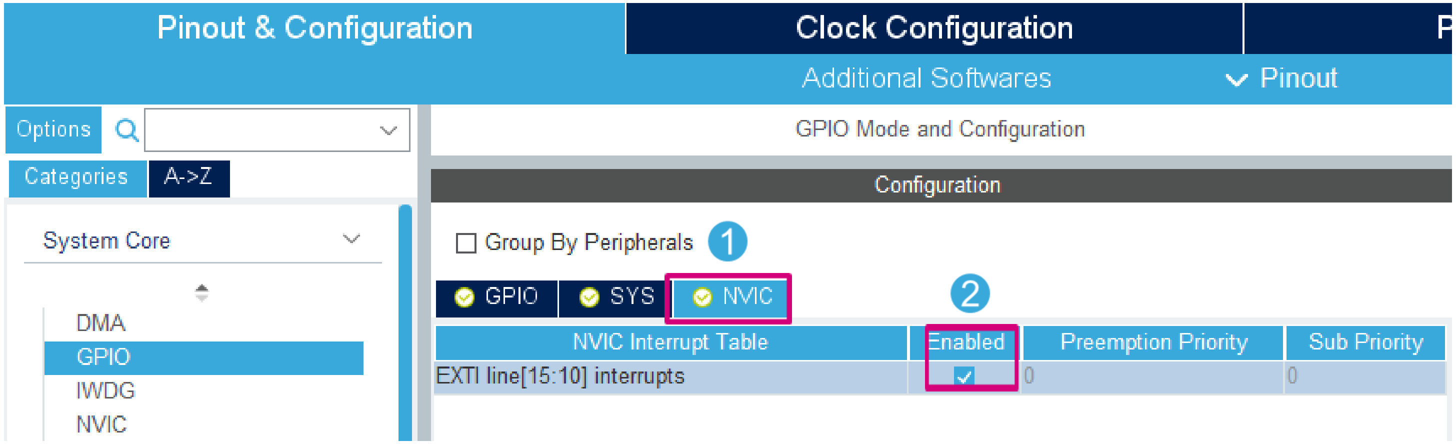
- Make the necessary changes to the peripheral settings, and then regenerate the project.

- In this case, configure PC13, which connect to blue button, as GPIO_EXTI13



Enable EXTI Interrupt

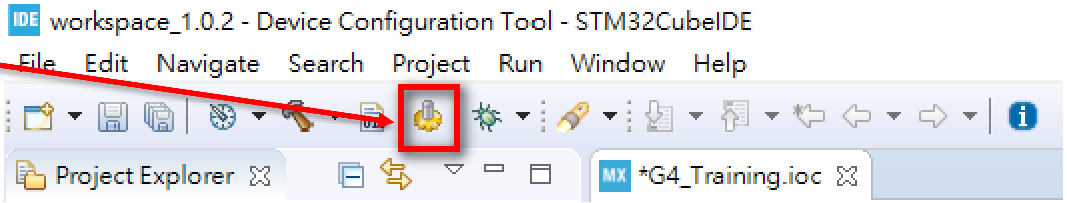
- Select NVIC tab, and enable the interrupt.



□ Group By Peripherals

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
EXTI line[15:10] interrupts	☑	0	0

- That all, now click the GENERATE CODE.


workspace_1.0.2 - Device Configuration Tool - STM32CubeIDE

Complete the new code

- Complete the code for EXTI.
- 1. Need a variable.
- 2. Code in while loop.
- 3. Interrupt function.

1

```
/* USER CODE BEGIN PV */
_Bool SpeedVar = 0;

/* USER CODE END PV */
```

2

```
/* USER CODE BEGIN 3 */
                    if(SpeedVar == 0)
                    {
                      HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
                      /* Insert delay 500 ms */
                      HAL_Delay(500);
                    }
                    else
                    {
                      HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
                      /* Insert delay 100 ms */
                      HAL_Delay(100);
                    }
  }
  /* USER CODE END 3 */
```

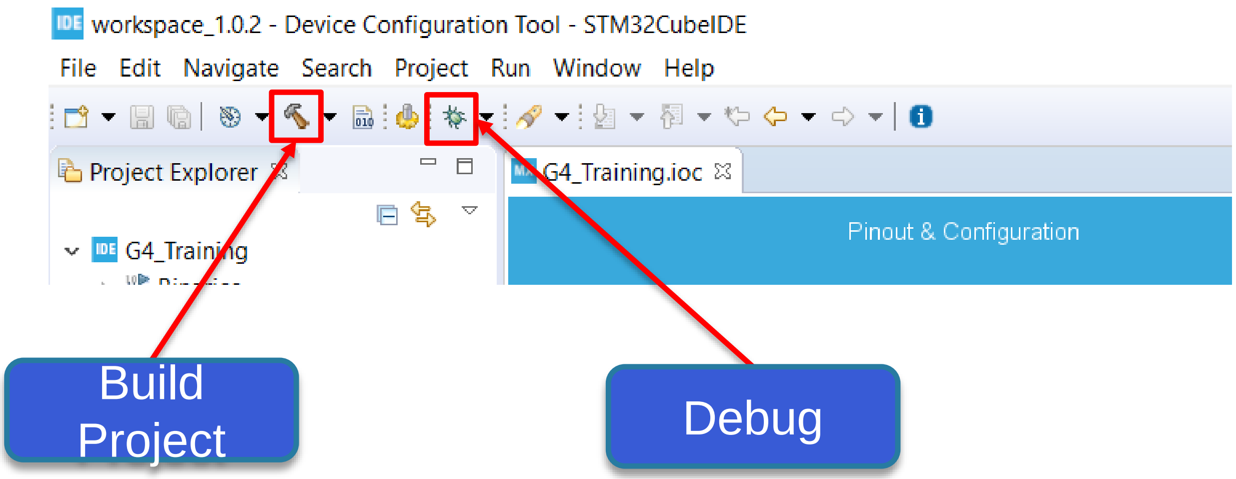
3

```
/* USER CODE BEGIN 4 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
  SpeedVar ^= 1;
}
/* USER CODE END 4 */
```

life.augmented

Build and Debug project

- STM32CubeIDE
  - Build the project by click “Make” and then “Download and Debug”.



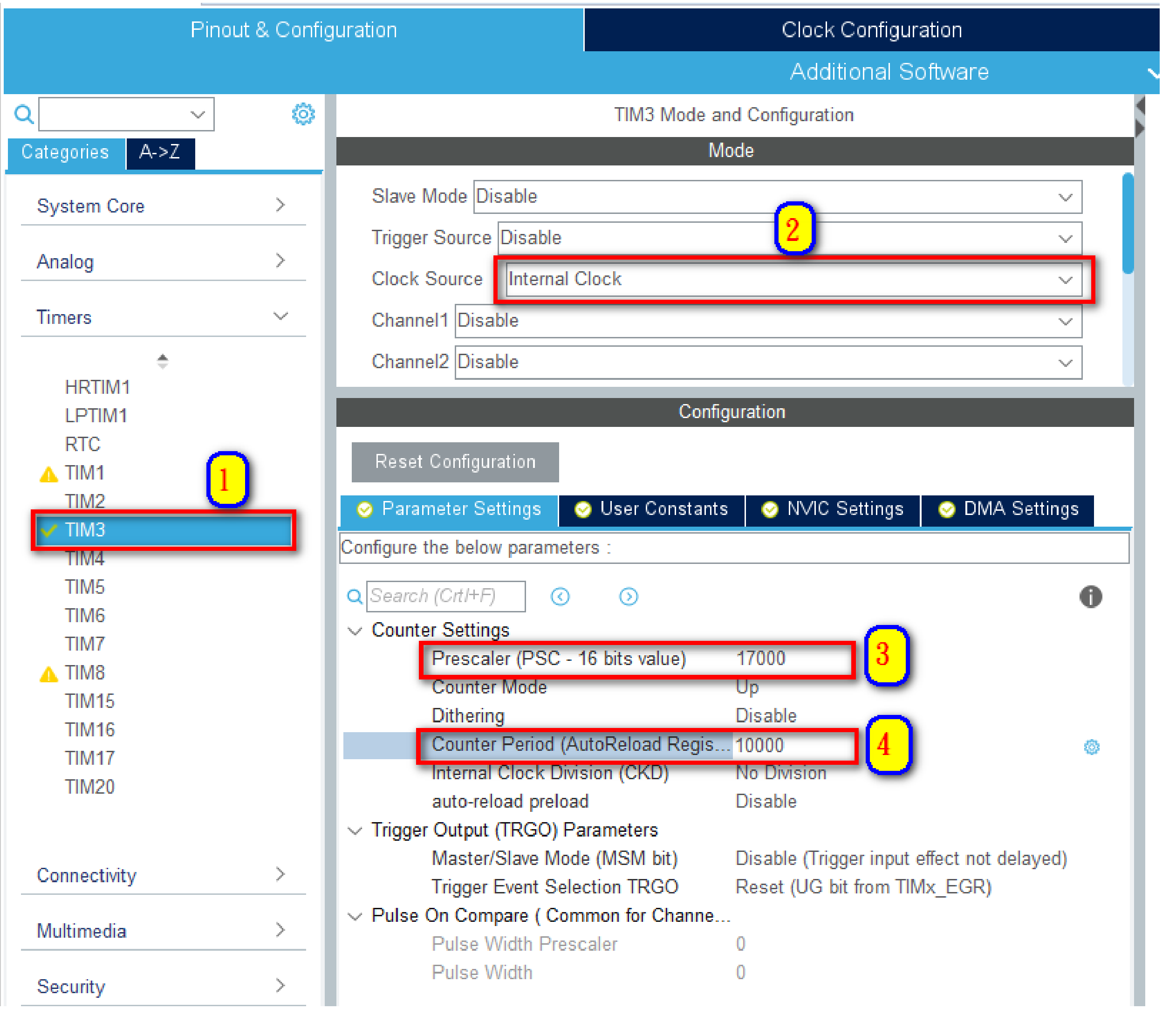
LAB2 Timer

Regenerating the initialization code

- Make the necessary changes to the peripheral settings, and then regenerate the project.

- In this case, configure TIM3.

- The Timer period is 1Hz.
  - 170MHz/170000/1000 = 1Hz



TIM3 Mode and Configuration

Mode

Slave Mode Disable
Trigger Source Disable
Clock Source Internal Clock
Channel1 Disable
Channel2 Disable

Configuration

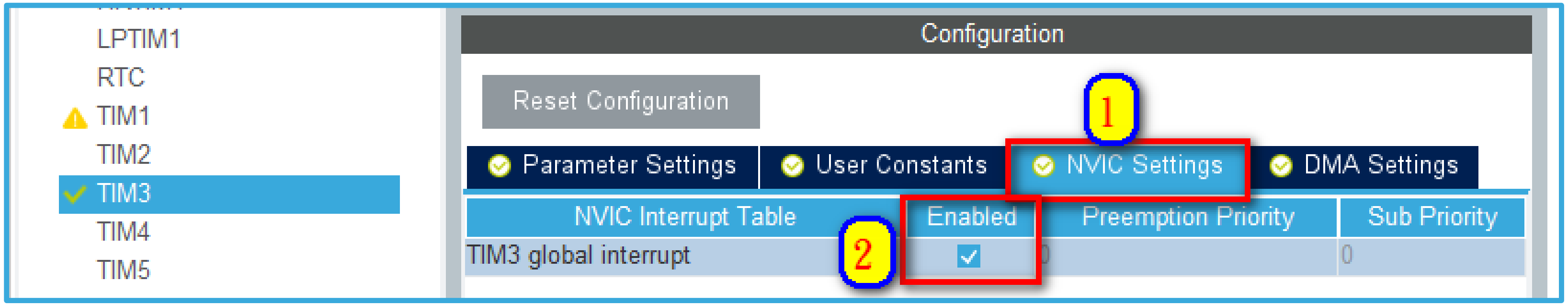
Configure the below parameters :

Counter Settings
Prescaler (PSC - 16 bits value) 17000
Counter Mode Up
Dithering Disable
Counter Period (AutoReload Regis... 10000
Internal Clock Division (CKD) No Division
auto-reload preload Disable
Trigger Output (TRGO) Parameters
Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)
Trigger Event Selection TRGO Reset (UG bit from TIMx_EGR)
Pulse On Compare (Common for Channe...
Pulse Width Prescaler 0
Pulse Width 0

life.augmented

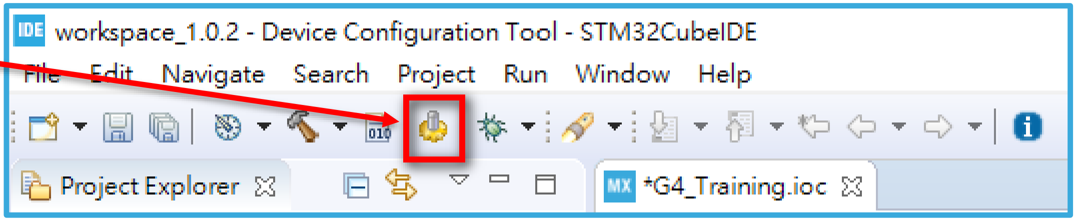
Enable TIM3 Interrupt

- Select NVIC tab, and enable the interrupt.



NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
TIM3 global interrupt	☑	0	0

- That all, now click the GENERATE CODE.



Complete the new code

- Mark code in while loop of main.c file to prevent clash.

①

```
/* USER CODE BEGIN 3 */
                         //  if(SpeedVar == 0)
                         //  {
                          //   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
                             /* Insert delay 500 ms */
                          //   HAL_Delay(500);
                          // }
                          // else
                          // {
                          //    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
                              /* Insert delay 100 ms */
                          //    HAL_Delay(100);
                          // }
 }
 /* USER CODE END 3 */
```

- Complete the code for TIM3.

②

```
/* USER CODE BEGIN 2 */
 HAL_TIM_Base_Start_IT(&htim3);
 /* USER CODE END 2 */
```

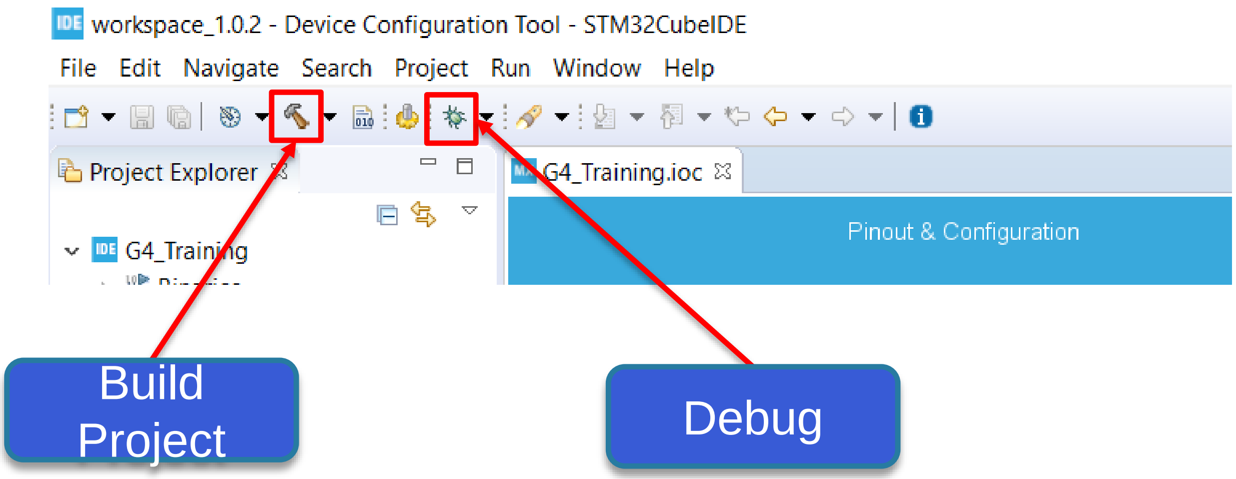
- Interrupt Call Back function.

③

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
   HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
}
```

Build and Debug project

- STM32CubeIDE
  - Build the project by click “Make” and then “Download and Debug”.



workspace_1.0.2 - Device Configuration Tool - STM32CubeIDE
File Edit Navigate Search Project Run Window Help
Project Explorer
G4_Training
G4_Training.ioc
Pinout & Configuration
Build Project
Debug

Releasing Your Creativity

/STM32 @ST_World community.st.com stm32g4-online-training

www.st.com/STM32G4