ERTC - Laboratory 0

29-03-2022

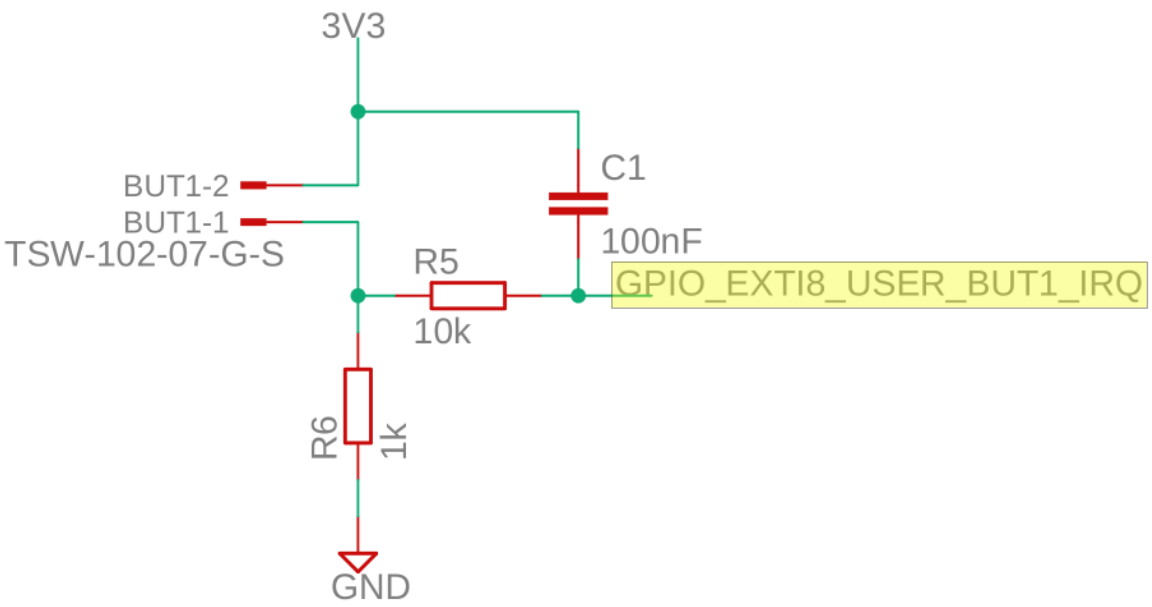
1 Introduction

This experience is focused on the use of GPIOs and interrupts.

The TurtleBot has several buttons, switches, and LEDs connected to the STM32F767 GPIOs.

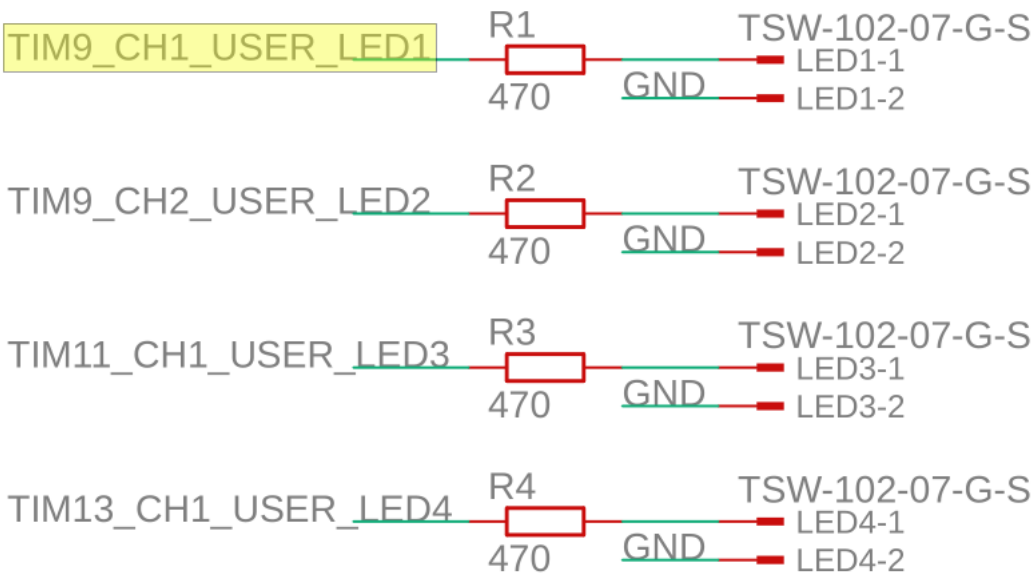
1.1 Button

One of the buttons is connected to the pin `PF8 (Pin 8, Port F)` and it is labeled as `GPIO_EXTI8_USER_BUT1_IRQ`. Normally the GPIO line is forced to ground by using the series of pull-down resistors `R6 and R5`. The capacitor `C1` combined with `R6 and R5`, form a low-pass filter (de-bounce circuit).



1.2 Led

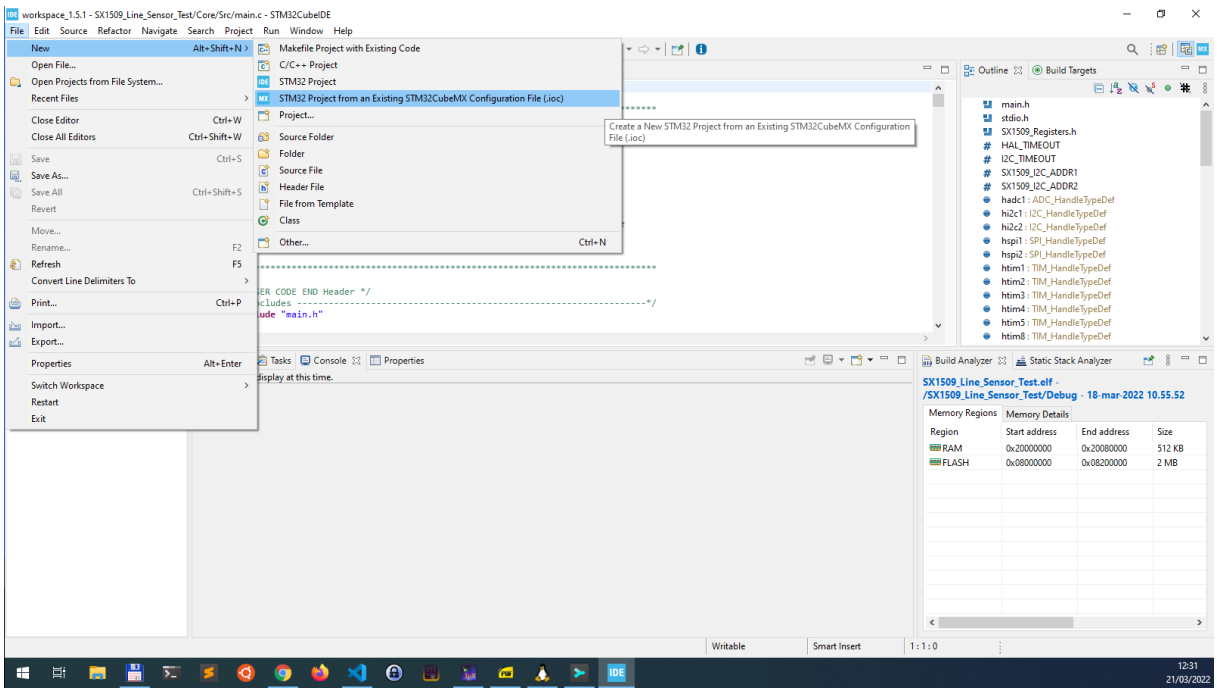
One of the LEDs is connected to the pin `PE5 (Pin 5, Port E)` and it is labeled as `TIM9_CH1_USER_LED1`. The led is connected directly to the GPIO using a current limiting resistor `R1`.



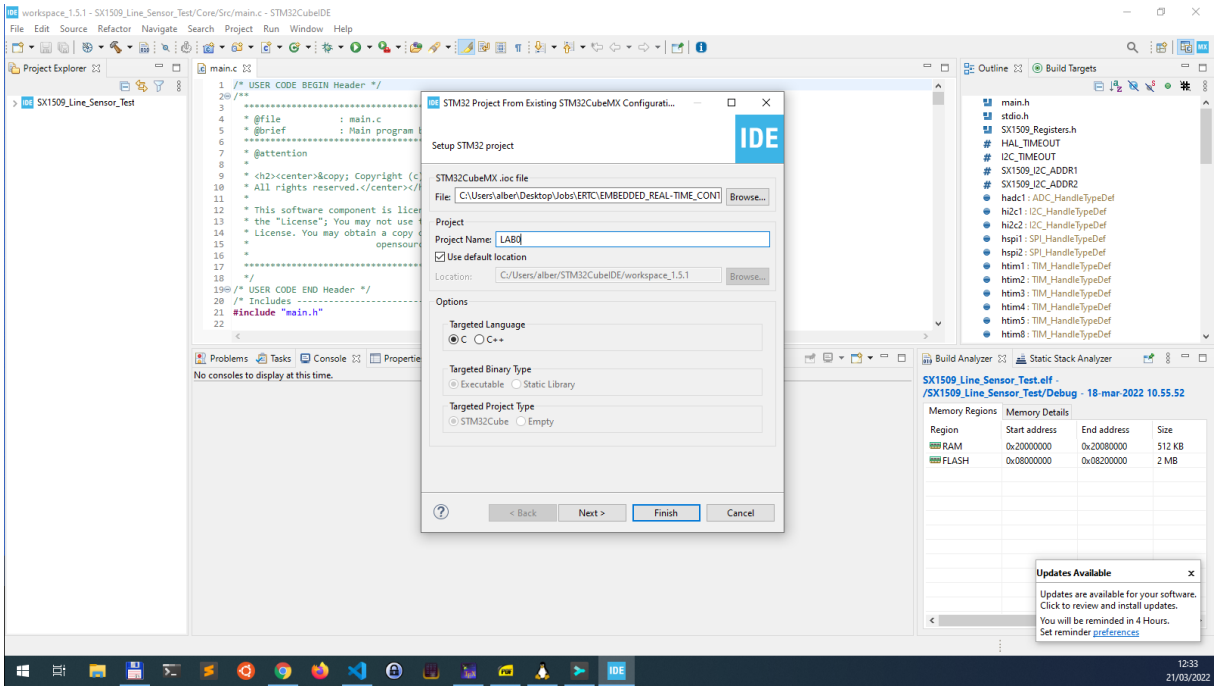
2 Preliminary operations

2.1 Create a new project

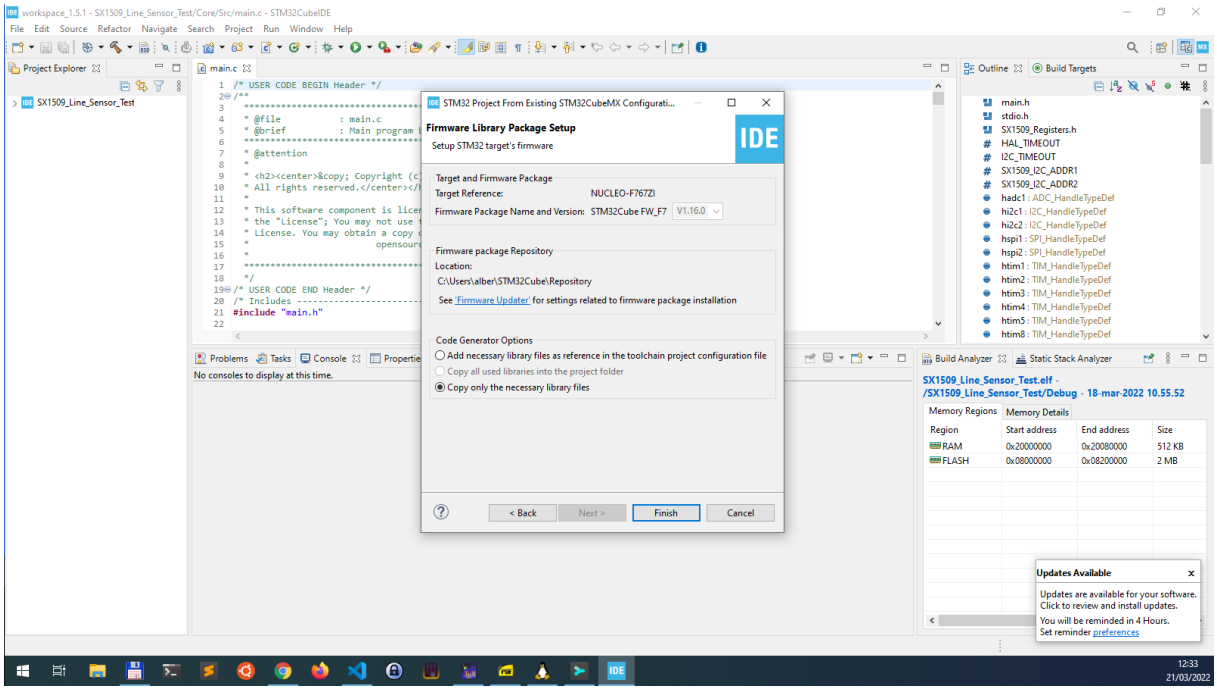
1. Open STM32CubeIDE
2. File -> New -> STM32 Project from an existing STM32CubeMX Configuration File



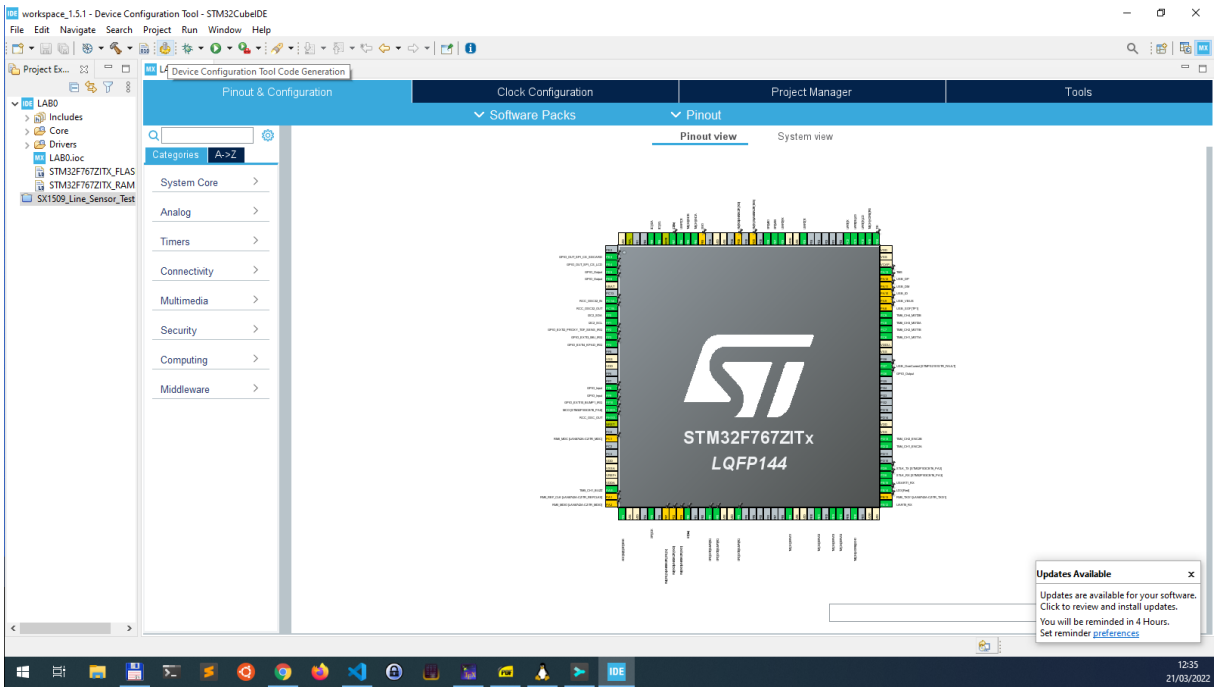
3. Select the provided `*.ioc` file and give a name to the project. Select `C` as target language. Click `Next`.



4. Select `Copy only the necessary library files` and click `Finish`.



5. You will see a screen similar to the following one. Click the gear icon on the tool bar. In this way the tool will generate all the necessary HAL functions an helpers.



2.2 HAL functions

Following are useful HAL functions for this lab. More detailed information here.

1. `HAL_GPIO_WritePin(GPIOx, GPIO_Pin, GPIO_PinState)` Set the state of a GPIO pin given the PORT, pin number and state.
2. `HAL_GPIO_ReadPin(GPIOx, GPIO_Pin)` Read the state of a GPIO pin given the PORT and pin number.
3. `HAL_GPIO_TogglePin(GPIOx, GPIO_Pin)` Toggle the state of a GPIO pin given the PORT and pin number.
4. `HAL_Delay(uint32_t Delay)` Delay execution for a given number of milliseconds.

3 Exercises

3.1 Exercise 1

Run this sequence **once**:

1. Turn the led **ON**;
2. Wait for 1[s] (you can use the function `HAL_Delay(uint32_t delay)` to wait for delay milliseconds);
3. Turn the led **OFF**;

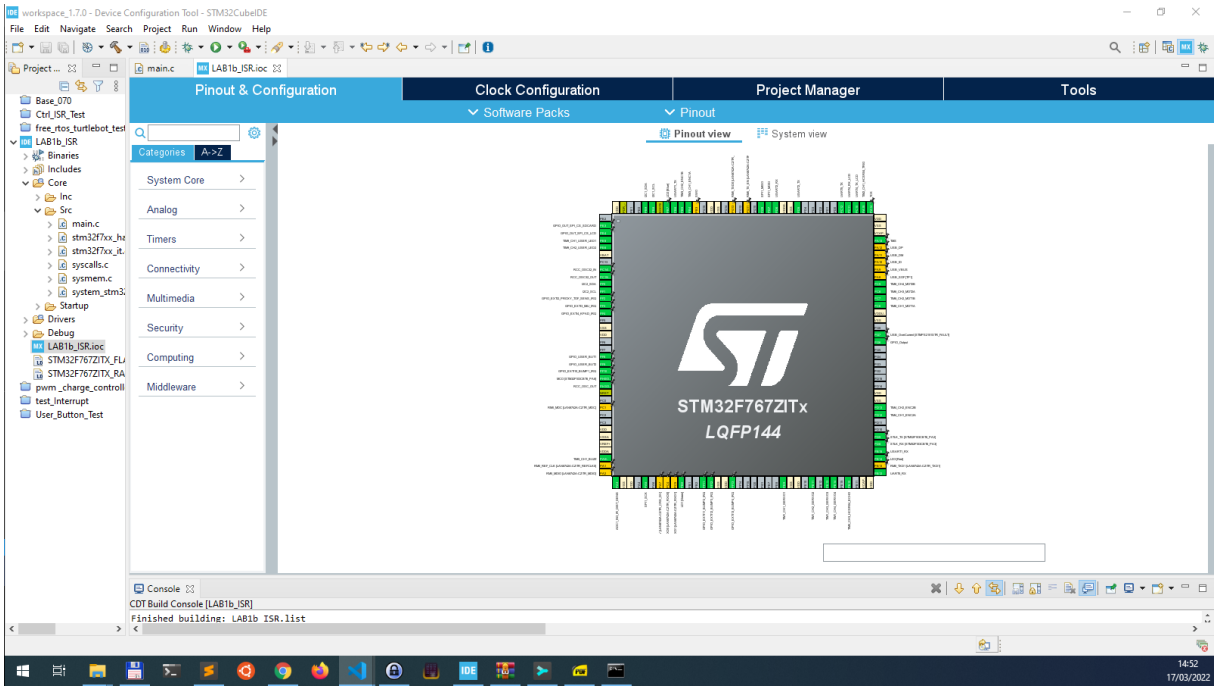
3.2 Exercise 2

Run this sequence **forever**:

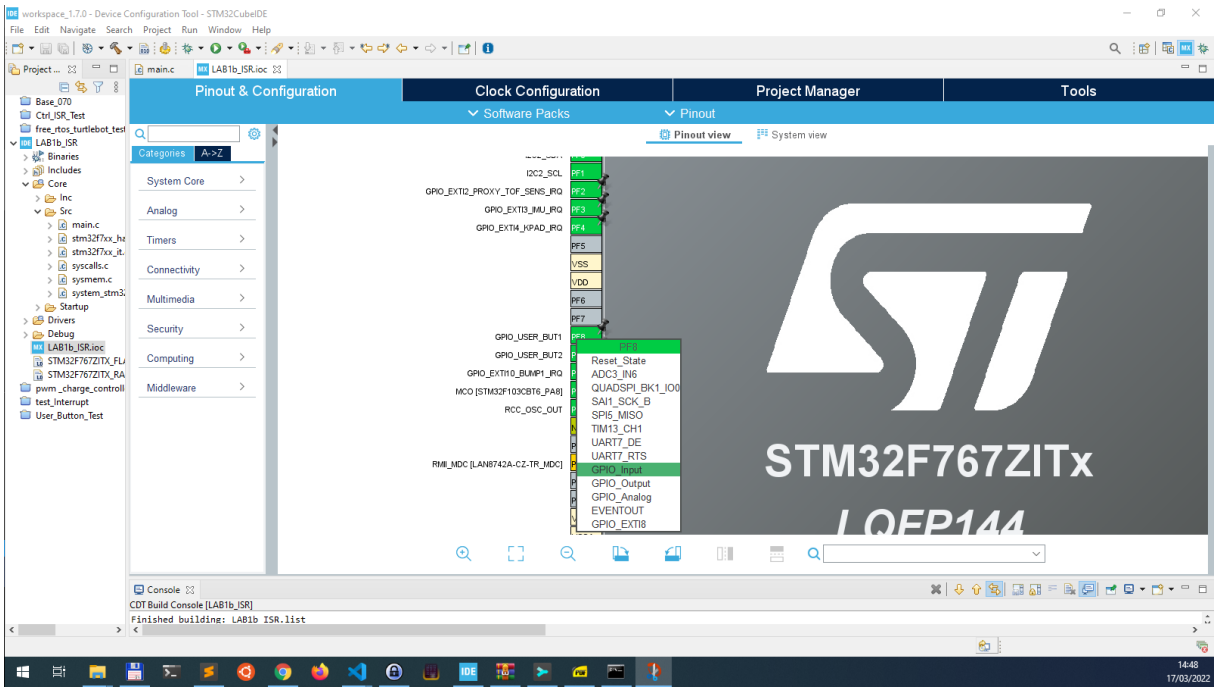
1. Read the state of the button;
2. If it is pressed, turn **ON** the led, **OFF** otherwise.
3. To simulate the time required to run other portions of the code, try different delays between two readings of the button state; Which is the effect?

3.3 Exercise 3

1. Configure the GPIO associated with the button, as an interrupt source; you can do this by opening the *.ioc file from the “Project Explorer” bar; You will have a screen similar to this

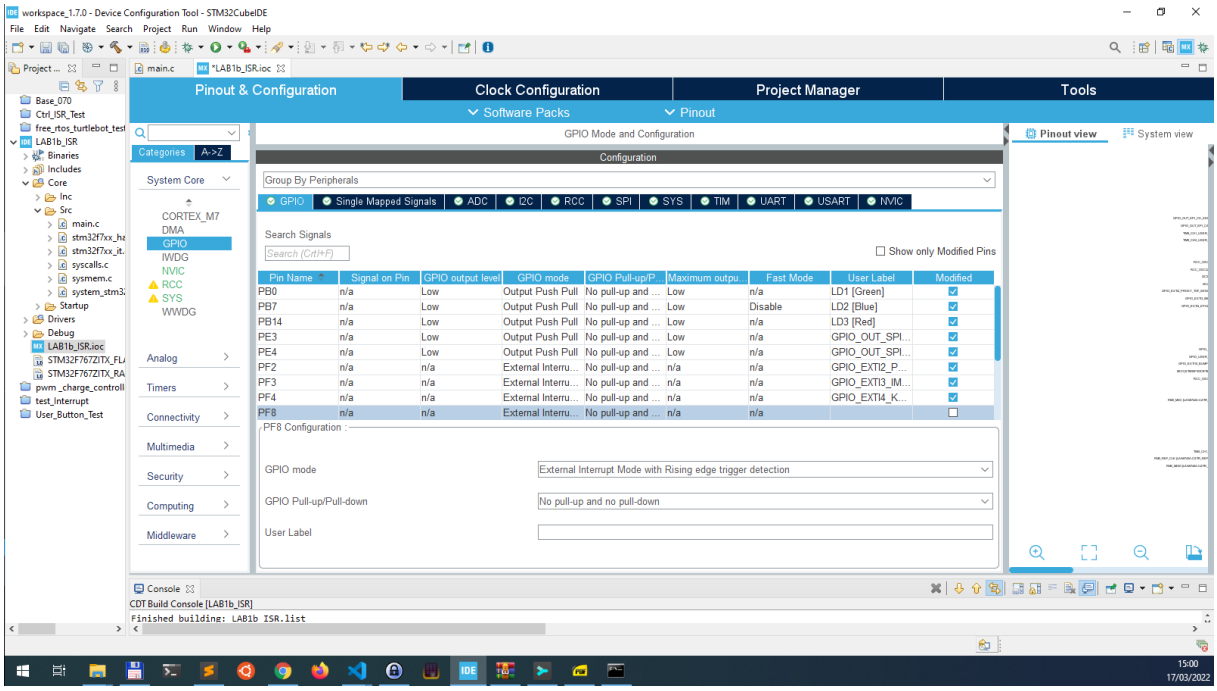


2. Search for pin PF8; Click on that and than click `GPIO_EXTI8`. In this way the GPIO is connected to the interrupt line 8.

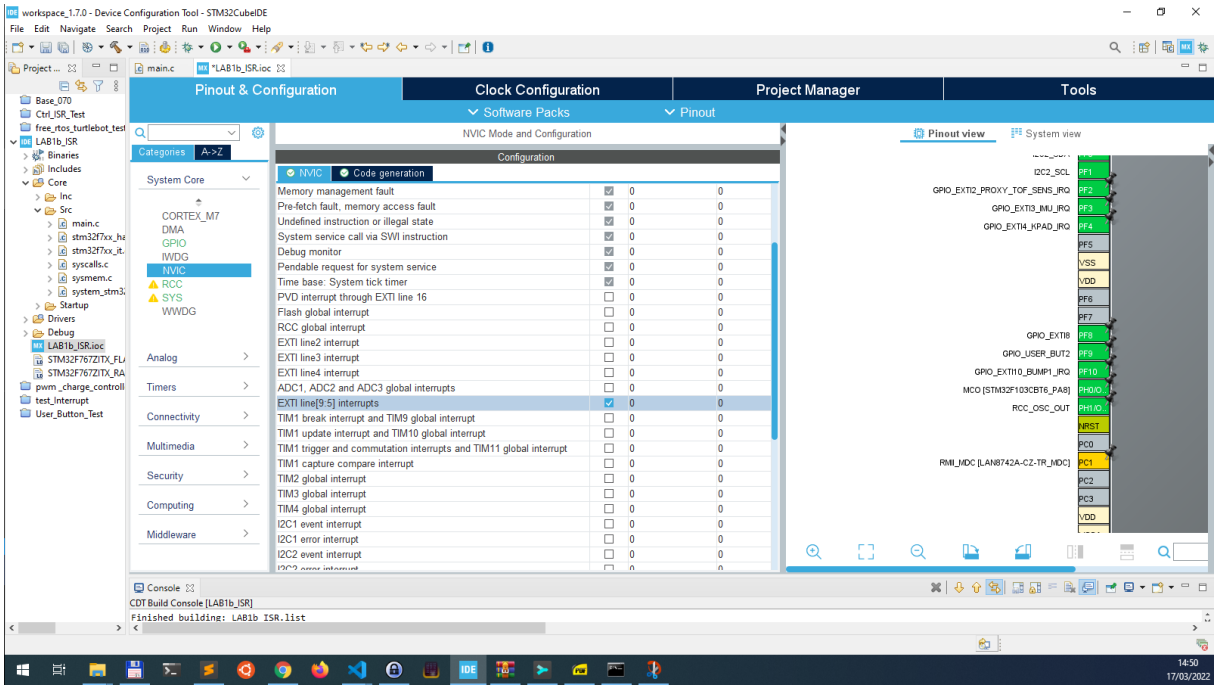


3. Click on System Core > GPIO. Search for pin PF8 and click on that. Here you can set the trigger mode for the interrupt and the pull/push up/down configuration. Select `External Interrupt`

`Mode with Rising edge trigger detection` and `No pull-up and no pull-down`.



4. Click on System Core > NVIC. Search for `EXTI line[9:5]` and enable it. In this way you have enabled all the interrupts for line 5 to 9. Here you can also change the priority.



5. Save (CTRL-S)

6. Remove the code you wrote inside main.c for exercise 1 and 2. Make a backup of the main.c file in case you need it in the future
7. Inside the main.c file, define a function called void HAL_GPIO_EXTI_Callback(uint16_t pin); this function is automatically called when the interrupt occurs;
8. Toggle the state of the led every time the button changes state from “not pressed” to “pressed”

3.4 Exercise 4:

Assuming that the button is configured as an interrupt source, as in Exercise 3:

1. Apply the sequence (led ON -> wait -> led OFF) from Exercise 1, but instead of doing that once, repeat the sequence forever;
2. Every time the button changes state from “not pressed” to “pressed”, decrease the wait time by 100[ms]. If the wait time reaches 0, reset it to 1[s];

3.5 Bonus

The TurtleBot has another led and button connected respectively to `PE6` and `PF9`. Try to use also these ones. Have fun!