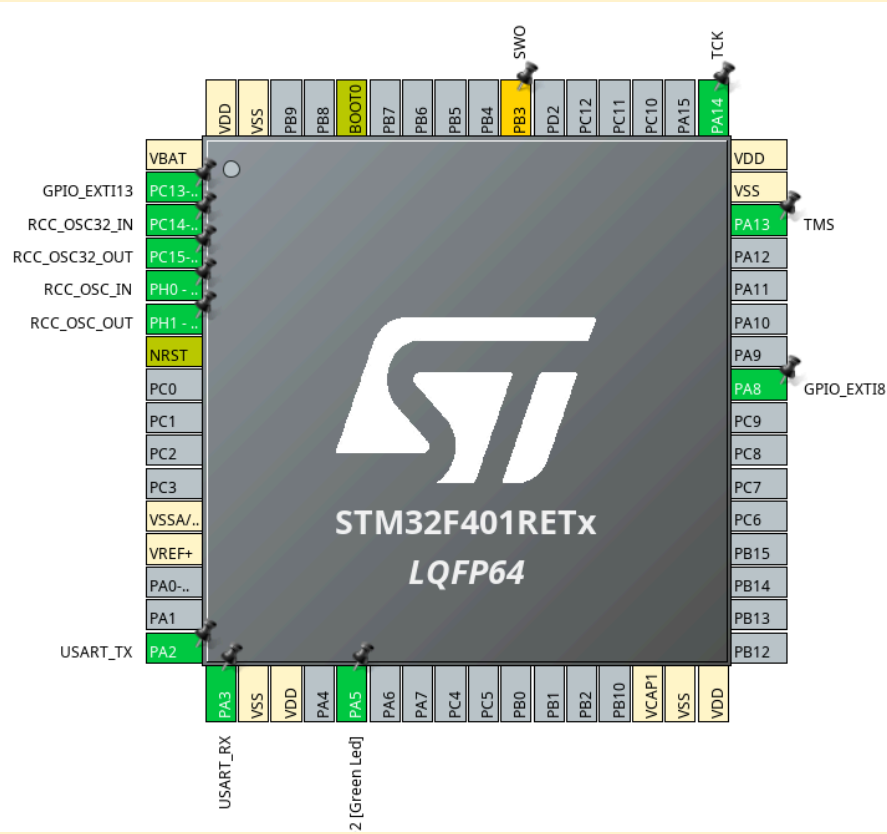


Mark	1/11
------	------

Team name:	B5		
Homework number:	HOMEWORK 02		
Due date:	01/10/24		
Contribution	NO	Partial	Full
Marenghi Manuela			x
Fellegara Tommaso			x
Giammusso Samuele			x
Cattani Luca			x
Csata Dániel			x
Notes: none			

Project name	Test		
Not done	Partially done (major problems)	Partially done (minor problems)	Completed
			x
<p>Part 1 a - Modify the status (switch on / off) of the NUCLEO green LED, every time you snap your fingers</p> <ul style="list-style-type: none"> To find the pin of the microphone: <p>we opened the pdf "Hands on lab schematic": search "microphone" and find that is connected to SND_DETECT search SND_DETECT and find that is connected to SND_IN search SND_IN and find that the pin is PA8.</p> <p>So the pin is PA8</p> <p>Another way is to search directly for SND_IN (as specified in the homework assignment) and follow the trace of the circuit until PA8</p>			

- We first set the PA8 pin to GPIO_EXTI8



- We then enable the EXTI line 15 to 10 and also the EXTI line 9 to 5 interrupts line in the NVIC

NVIC Mode and Configuration

Configuration

☒ NVIC
 ☒ Code generation

Priority Group ☒
☐ Sort by Preemption Priority and Sub Priority
 ☐ Sort by ir

Search Show
☒ Force DM

NVIC Interrupt Table	Enabled	Preemption P
Non maskable interrupt	<input checked="" type="checkbox"/>	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0
Memory management fault	<input checked="" type="checkbox"/>	0
Pre-fetch fault, memory access fault	<input checked="" type="checkbox"/>	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0
Debug monitor	<input checked="" type="checkbox"/>	0
Pendable request for system service	<input checked="" type="checkbox"/>	0
Time base: System tick timer	<input checked="" type="checkbox"/>	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0
Flash global interrupt	<input type="checkbox"/>	0
RCC global interrupt	<input type="checkbox"/>	0
EXTI line[9:5] interrupts	<input checked="" type="checkbox"/>	0
USART2 global interrupt	<input type="checkbox"/>	0
EXTI line[15:10] interrupts	<input checked="" type="checkbox"/>	0
FPU global interrupt	<input type="checkbox"/>	0

- Then we set the GPIO mode of the pin PA8 to External Interrupt mode with Falling edge trigger detection, we decided to use the falling edge to toggle the state of the LED when snapping our fingers (instead of making the LED blink)

Pin ...	Signal o...	GPIO ou...	GPIO m...	GPIO Pu...	Maximu...	User La...	Modified
PA5	n/a	Low	Output ...	No pull-...	Low	LD2 [Gr...	✓
PA8	n/a	n/a	External...	No pull-...	n/a		✓
PC13-A...	n/a	n/a	External...	No pull-...	n/a		✓

PA8 Configuration :

GPIO mode: External Interrupt Mode with Falling edge trigger det

GPIO Pull-up/Pull-down: No pull-up and no pull-down

User Label:

- Saving the previous controller configuration on the IDE creates the code for the ISR, which calls the HAL_GPIO_EXTI_Callback function, passing as argument GPIO_Pin == GPIO_PIN_8, this allows us to extend the EXTI callback code developed in class to toggle the status of the LED on microphone input as follows (added the second switch case):

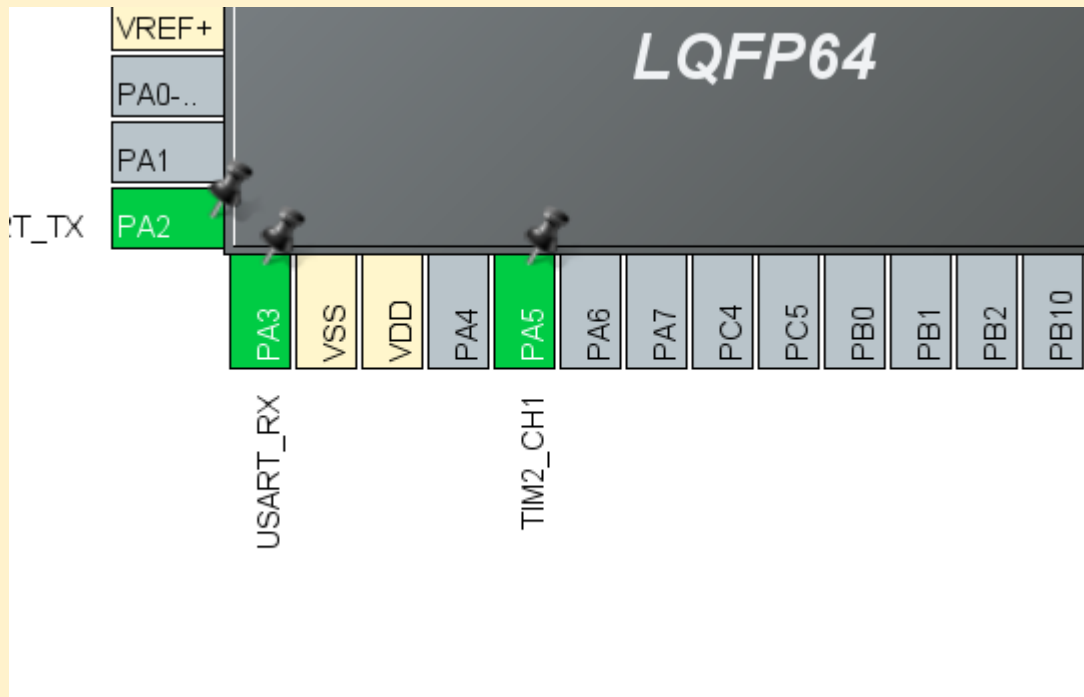
```

68 /* USER CODE BEGIN 0 */
69 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
70 {
71     switch(GPIO_Pin){
72     case GPIO_PIN_13: // keep the led on (or off) while the button is pressed
73         GPIO_PinState pushbutton = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13);
74         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, !pushbutton);
75         break;
76     case GPIO_PIN_8: // toggle the status of the led
77         GPIO_PinState ledStatus = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_5);
78         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, !ledStatus);
79         break;
80     default:
81         break;
82     }
83 }

```

Part 1 b - Make the NUCLEO green LED blink at a 1 Hz rate using PWM generation on the corresponding channel

- Set the PA5 pin (the green led pin) to TIM2_CH1



- To achieve a 1 Hz, use this formula:

$$f_{PWM} = \frac{f_{TIM} \overset{84 \text{ MHz}}{\nearrow}}{(\underset{\curvearrowright 10'000-1}{ARR+1}) \cdot (\underset{\curvearrowright 8400-1}{PSC+1})} = 1 \text{ Hz}$$

- Then set up the timer as seen in class:

The screenshot shows the STM32CubeMX Configuration tool. At the top, several dropdown menus are set to 'Disable' (Slave Mode, Trigger Source, Channel2, Channel3, Channel4) or 'Internal Clock' (Clock Source). Channel1 is set to 'PWM Generation CH1'. Below these is a 'Configuration' bar with a 'Reset Configuration' button. A row of tabs includes 'Parameter Settings' (selected), 'User Constants', 'NVIC Settings', 'DMA Settings', and 'GPIO Settings'. Under 'Parameter Settings', a search bar is present. The 'Counter Settings' section is expanded, showing: Prescaler (PSC - 16 bits value) set to 8400-1, Counter Mode set to Up, Counter Period (AutoReload Register - 32 bits value) set to 10000-1, Internal Clock Division (CKD) set to No Division, and auto-reload preload set to Disable. The 'Trigger Output (TRGO) Parameters' section is also expanded, showing Master/Slave Mode (MSM bit) set to Disable (Trigger input effect not delayed) and Trigger Event Selection set to Reset (UG bit from TIMx_EGR). The 'PWM Generation Channel 1' section is expanded, showing Mode set to PWM mode 1, Pulse (32 bits value) set to 5000-1, and Output compare preload set to Enable.

- Then add this HAL function to activate the PWM

```

96  /* USER CODE BEGIN 2 */
97
98  HAL_TIM_PWM_Start(&htim2,TIM_CHANNEL_1);
99
100 /* USER CODE END 2 */

```

- Then build and run the code to see the led blinking

Professor comments: