|  |  |
| --- | --- |
| **Mark** | **/11** |

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A5* | | |
| Homework number: | *HOMEWORK 02* | | |
| Due date: | 29/09/2024 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| Crespi Emma |  |  | *x* |
| Cecchetto Sofia |  |  | *x* |
| De Togni Annamaria |  |  | *x* |
| Spineto Alessio |  |  | *x* |
| Lamarca Riccardo |  |  | *x* |
| Notes: none | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | GPIOs and Timers | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Completed |
|  |  |  |  |
| Explanation:  The homework consisted of two separate parts: the first (a) to understand how to setup GPIOs and handle exceptions; the second part (b) is focused on timers and PWMs.    **Part 1a:**  In the first part the goal was to toggle a LED when the microphone senses a finger snap. First, we searched for “SND\_IN” in the “PMDB16 Schematics” and discovered that the microphone is wired to the GPIO “PA8”.  Hence, we configured “PA8” as GPIO\_EXTI8 to use the external interrupt generated by the microphone:    We selected *Rising edge trigger detection* because we wanted the LED to turn on and off when the noise starts but do nothing when the noise stops.   We also need to enable the 8th line of the EXTI:    We did not need to reconfigure the PA5 pin (the green LED on the NUCLEO board), because the default configuration is already GPIO\_output, the one we need to set the state of the pin.  Now, to toggle the state of “PA5” we need to first define “GREEN\_LED” in the *main.c* file as follows:  Then we define a callback function that executes every time an external interrupt occurs:  In the HAL\_GPIO\_EXTI\_Callback function we check if the GPIO\_Pin that called the said function is the microphone’s Pin (GPIO\_PIN\_8). If this condition is verified, we toggle the GREEN\_LED value.  Now, whenever the microphone hears a sound the pin PA8 generates an interrupt on line 8 (rising edge detection) and the current state of the LED is changed until a new sound is detected.  **Part 1b:**  The second part of the homework required us to make the NUCLEO green LED blink at 1Hz rate using PWM generation on the corresponding channel.  First, we found on the “Nucleo schematic” that the green LED is wired to “PIN5” of “PORT A”.  We defined it into the *main.c*:    And we also configured it into the Pinout view as GPIO\_OUTPUT:    We enabled the global interrupt on the overflow of the timer TIM2  .    We then computed the values of the prescaler and the Auto Reload Register to obtain a frequency of the timer of 1Hz:  Knowing that = frequency of the internal clock, and that , the only two parameters left to configure are the ARR (Auto Reload Register) and the PSC (Prescaler).  Considering the limitations on the value of the two parameters we decided to set the prescaler at 0:  And the ARR at 83999999:    So that  Inide the main, but outside the while(1) loop we used the function HAL\_TIM\_Base\_Start\_IT to start the timer.    We then used the callback function HAL\_TIM\_PeriodElapsedCallback that executes every time the external timer TIM2 generates a trigger. In this way with a frequency of 1 Hz, we change the output of the LED, using a function that toggles the value of the selected PIN. | | | |
|  | | | |
| Professor comments: | | | |