|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *B7* | | |
| Homework number: | *HW01* | | |
| Due date: | *October 4th* | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| *1 Giampà Simone* |  |  | *x* |
| *2 Massa Giacomo* |  |  | *x* |
| *3 Raduzzi Lucafrancesco* |  |  | *x* |
| *4 Micelli Johanna* |  |  | *x* |
| *5 Galimberti Claudio* |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | HW 01: led switch with microphone input | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation:  We’ve firstly looked for the ‘SDN\_IN’ signal input (that’s linked to the audio microphone) in the “Microcontrollers Hands On” PDF file, finding out that it’s connected to the morpho connector 23; then in the Nucleo Schematic file we found the connection to the PA8 Pin.  Secondly, once created the project, we initialized the PA8 pin as an EXTI8 to let it be sensible to the external interrupt signal. After that, we enabled EXTI lines [9:5] interrupts (in the NVIC section of the configuration pane) and set the GPIO mode of the PA8 configuration as external interrupt mode with rising edge trigger detection (in the GPIO section).  Once set the environment settings related to various pins, we’ve generated, before the main function, a callback function (taken from the declaration of HAL\_GPIO\_EXTI\_IRQHandler(GPIO\_PIN\_8) placed in the stm32f4xx\_it.c file). The callback handles only the interrupts from the PIN8 and toggles the PIN5 status in order to switch on/off the led (since LED2 is connected to the PIN5).  When the microphone will sense a sound which intensity is higher than a certain threshold, an interrupt will be generated and the callback function will be executed, toggling the PIN5 and switching on/off the led. This is possible because the microphone is connected to a comparator between 3.3v and ground, with a digital output signal (binary output). | | | |
| Professor comments: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | HW 01: blinking LED with PWM generation | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation:  We set the LED2 (PA5) as TIM2\_CH1 output. Then in the pinout and configuration we set in the timer2 the output for channel 1 as the PWM generation CH1.  The PWM generation was achieved using the edge aligned PWM, and we used the formulas to compute the values to set to the constants such that the resulting timer frequency would be 1Hz. Inside the configuration of the timer2 we declared the user constants PSC (prescaler) = 83999, ARR (auto reload register) = 999 and CCR = 41999.  The pulse generation is set equal to CCR, the period to the ARR and the prescaler value to PSC. The clock cycle was left to the default value of 84MHz.  Then in the main.c file the line HAL\_TIM\_PWM\_Start(&htim2, TIM\_CHANNEL\_1) is used to start the timer at channel 1 after the initializations of the standard functions. | | | |
| Professor comments: | | | |