|  |  |
| --- | --- |
| **Mark** | **A** |

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A10* | | |
| Homework number: | *HOMEWORK 7* | | |
| Due date: | 21/11/2022 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| 1 Team member |  |  | *x* |
| 2 Team member |  |  | *x* |
| 3 Team member |  |  | *x* |
| 4 Team member |  |  | *x* |
| 5 Team member |  |  | *x* |
| 6 Team member |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | MEMS Accelerometer | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Completed |
|  |  |  | *x* |
| Explanation:  We successfully completed the homework.  **Part 1b:**  We set the I2C and USART settings as we did in part 1a, adding in DMA settings a USART2\_TX DMA request. To perform acquisitions every 1 second, we needed a timer interrupt. We used TIM2, setting the Prescaler as 8400-1 and the counter period as 10000-1. We then activated the USART2 and TIM2 global interrupts.  We started the timer using the HAL\_TIM\_Base\_Start\_IT function, paying attention to doing this after the HAL\_I2C\_Master\_Transmit operations to configure the accelerometer (to avoid conflicts with timer interrupts).  In the “HAL\_TIM\_PeriodElapsedCallback” timer callback, we used the HAL\_I2C\_Master\_Transmit function to transmit a specific sub-address and the HAL\_I2C\_Master\_Receive function to read the data and store them in the acc\_x, acc\_y, acc\_z variables. We did this to respect the sensor protocol.  Finally, we converted the values and transmitted them in a string using the HAL\_UART\_Transmit\_DMA function.  **Part 1c:**  Starting from the same settings of part 1b, we added an I2C1\_RX DMA request and we enabled the I2C1 event and error interrupts in the NVIC settings.  Instead of using three different reading operations, we used the auto-increment function of the sensor, by setting to 1 the MSB of the first sub-address in a single HAL\_I2C\_Master\_Transmit function.  We then read 5 subsequent registers providing a 5 elements buffer to the HAL\_I2C\_Master\_Receive\_DMA function. We had to read 5 values because the distance between the OUT\_X and the OUT\_Z registers is equal to 5. We called these two functions inside the timer callback function.  Eventually, in the HAL\_I2C\_MasterRxCpltCallback callback (called when all the five values are read), we used the populated buffer data to make the requested conversions and to send them to the PC by using the HAL\_UART\_Transmit\_DMA function. | | | |
| Professor comments:  OK! Good! | | | |