Complete the “yellow” tabs and delate the phrases in italics.  
You can duplicate the table “Project”, if more than one project are due for the homework.

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A1* | | |
| Homework number: | HW07 | | |
| Due date: | 21/11 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| 1 *Giorgio Donato Carlo* |  |  | *x* |
| 2 *Lenzi Francesco* |  |  | *x* |
| 3 *Lodari Gianmarco* |  |  | *x* |
| 4 *Lanzini Alessio* |  |  | *x* |
| 5 *Chiapparo Lenn* |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | Accelerometer pt.b | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation: The goal of the project is to acquire the data from the accelerometer through I2C protocol, once every 1s , and send it to the PC through UART.  First of all we set up TIM2 in time base with the usual prescaler and period settings to get an interrupt every one second. Afterwards, we enabled the USART in DMA normal mode, making sure to enable the corresponding interrupt to avoid the known issue with the USART. Then we proceeded as described during the Monday lesson: we enabled the I2C in the .ioc. We checked the version of the accelerometer trough a simple conditional statement in which we checked the correct peripheral address and then configured the accelerometer’s control registers. In the timer interrupt service routine, we acquired the x, y and z registers and sent it through the UART in DMA mode. | | | |
| Professor comments:  ok!  Did you performed the conversion to obtain a number expressed in g? Yes  Did you configured the accelerometer registers? Yes | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | Accelerometer pt.c | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation: in the main we set up the peripherals and recognize the right address, as the previous program, and started the timer. Then, in HAL\_TIM\_PeriodElapsedCallback, we took the first output register (the X-axis) and set the MSB to 1 to enable auto increment read, as shown in the slides. Afterwards, we call the transmit and receive DMA functions, saving the values in the global viable "axel\_val", declared as a 5 element int8\_t array. Eventually, in HAL\_I2C\_MasterRxCpltCallback we converted to strings the even index elements of the “axel\_val” array (which are the X, Y and Z axis values, from the sensor’s datasheet) and sent them to PC using Uart\_transmit\_DMA, then the timer is restarted. | | | |
| Professor comments:  OK! | | | |