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: | *6* | | |
| Due date: | *10/11/2022* | | |
|  |  |  |  |
| 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:  *Complete in necessary* | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | I2C thermometer 1b | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation: We configured the I2C in standard mode, enabling the pins PB8 and PB9. Furthermore, we enabled the TIM2 to use it as interrupt every 1 sec (PR=8400-1 ARR=10000-1).  In the main function we Transmit via I2C the address of the cell that we want to read inside the temperature sensor and start timer 2 in interrupt mode.  In the Elapsed Time Callback function called by the interrupt, we declare an array of two elements, 8 bit signed integers, in which we will store the MSByte and the LSByte given by the temperature Sensor through the I2C. Once we have received the data, we calculate the result through some shifts and multiplication in order to merge the two data (MSByte and LSByte) correctly.  Finally, we send the result via UART and we start again the Timer in interrupt mode.  To verify the correctness of our conversion function, especially with negative or very high temperatures, we did some tests using values taken from the datasheet.  While taking measures of the room temperature, we noticed that most of our temperature sensors had a sensitivity of 0.5°, this means our board are equipped with the LM75 standard version, which uses only one bit for the decimal part. | | | |
| Professor comments: | | | |