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

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A5* | | |
| Homework number: | *HOMEWORK 07* | | |
| Due date: | 3/11/2024 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| Alessio Spineto |  |  | *x* |
| Riccardo Lamarca | *x* |  |  |
| Sofia Cecchetto |  |  | *x* |
| Annamaria De Togni |  |  | *x* |
| Emma Crespi |  |  | *x* |
| Notes: none | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name |  | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Completed |
|  |  |  |  |
| **Complete the I2C thermometer project 1b:** read the temperature measured by the LM75 and send it to a remote terminal every 1 second.  The LM75 temperature sensor uses the I2C protocol on PB8 for the clock (SCL) and PB9 for the data (SDA) lines. We enabled I2C1 for the two pins using the pinout view and set I2C1 mode to I2C.    Since the I2C bus is connected to different peripherals, we set two global variables with the address of the LM75. Both addresses are left-shifted by 1: *LM75\_ADDRESS\_W* has the R/W bit equal to 0, while *LM75\_ADDRESS\_R* has the R/W bit equal to 1.  We also set LM75\_TEMP\_R with the address of the Pointer register of the sensor, which will allow us to select the temperature register.    We also need to include <string.h>.    Since we decided to manage I2C communication in interrupt mode, we enabled the I2C1 interrupts.    We used USART in DMA mode with transfer mode USART2\_TX and high priority.  We also set TIM2 to 1Hz and enabled its global interrupt, along with USART global interrupt.    OLD //Inside the main(), we initialize TIM2 in interrutpt mode and set the pointer register of the LM75 to the temperature register.  ALESSIO //Inside the main(), we start TIM2 in interrupt mode, then write 0x00 (LM75\_TEMP\_ADDRESS) to the temperature sensor, this way we select the value we want to get, in this case the temperature.    We also need to create two callback functions, one for the I2C communication and one for the timer:        If we read the value only one time, a peculiar bug can occur, where the LSB are updated but not the MSB, so in the timer callback we call the I2C receive function 2 times.  For the same reason, we check with a flag how many times has the callback function been called.  If it’s the second time, we convert the value from two’s complementary to float, separating the case for positive and negative numbers.  If the number is negative, since we are getting a value in 2C, it will be > 127 when stored in a uint16\_t format.  Then we just need to format a string and send it with UART and DMA.  The conversion works for the LM75B, which uses 2 more bits to store decimal values than the LM75, resulting in more accuracy. | | | |
| Professor comments: | | | |