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

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *B01* | | |
| Homework number: | *HOMEWORK 7* | | |
| Due date: | 21/11/2023 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| Francesco Scroccarello |  |  | *x* |
| Paolo Salvatore Galfano |  |  | *x* |
| William Stucchi |  |  | *x* |
| Giada Silvestrini |  |  | *x* |
| Francesco Maria Tranquillo |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | SPI LED Matrix | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Completed |
|  |  |  | *x* |
| Explanation:  We successfully completed the homework.  **Part 7a:**  In the GUI, we set pins: PB6 to GPIO\_Output, PA5 to SPI1\_SCK, PA7 to SPI1\_MOSI (PA6 is not required since we’re only transmitting from the Master to the Slave). Then we moved to Connectivity -> SPI1 ->Mode: Transmit Only Master, prescaler to 4. We moved to Timers -> TIM10: prescaler to 84-1 and period to 4000-1 in order to have a frequency of 250Hz, and we enabled the TIM1 update interrupt and TIM10 global interrupt from the NVIC tab.  In the code we defined a matrix as a global variable containing the tuples that allow to draw the letter A. Then we started the timer in interrupt mode in the main (outside the while loop) and we redefined the callback in order to print the correct values at the frequency defined by the timer. In particular, we transmitted the right values relative to the column of the matrix. With the write pin function we enabled the RCLK pin in order to write the values of the matrix and soon after we reset the pins to store permanently the values. Notice that in order to have the letter correctly displayed we have to send to the matrix one column at a time according to the refresh rate imposed by the timer.  **Part 7b:**  Starting from our work previously done, we enabled an additional timer (TIM1) with clock source: internal clock, prescaler 8400-1 and counter period: 10000-1 (this is to have a 1Hz period to change the shown letter). Interrupt flags were already set by the previous work. In the Connectivity -> SPI we created a new DMA configuration in SPI\_TX.  In the code we created the 3 global matrices storing the letters A, B and C and an additional global matrix to store the right value to display at runtime. We defined 2 callbacks, respectively for period elapsed of the timers and for the completion of the SPI transmit. In the first, according to the timer that raised the interrupt, we reacted in 2 ways:   * If the timer was the one of the refresh rate of the matrix, we transmitted in DMA mode the column of the right matrix copied in the global matrix. * If the timer was the one for alternating letters, we rised the value of a flag that, read in the while loop of the main (in non-blocking mode), allows the switch of the letter at runtime by copying the memory content of the right global variable into the general displayed matrix.   Then, in the SPI\_TxCpltCallback we set and reset the RCLK pins as above in order to store and permanently save the values of the column of the matrix. Notice that this callback is necessary because we’re transmitting in DMA mode so we have to correctly work on the RCLK signal only when the writing process is completed and so the callback allows us to synchronize all the operations. | | | |
| Professor comments:  Ok, but better to show some code screenshots. | | | |