|  |  |
| --- | --- |
| Mark |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A2* | | |
| Homework number: | *10* | | |
| Due date: | 1/12/2024 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| La Barbera Marco |  |  | *x* |
| Lotto Giulio |  |  | *x* |
| Majocchi Tommaso |  |  | *x* |
| Maffezzini Andrea |  |  | *x* |
| Pompilio Matteo |  |  | *x* |
| Notes: none | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | SPI LED MATRIX | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Completed |
|  |  |  | *x* |
| We successfully completed the homework.  Next, we will explain all the steps for accomplishing our goals:  **First part (Keyboard):**  First of all, we configure the board pinout for the keyboard: PC2/3/12/13 for the rows to be read, as GPIO\_Input; PC8/9/10/11 for the columns to be scanned, and so written, as GPIO\_Output.    From “Timers*”*, we enable TIM2 and TIM3, as we’ll both need a timer to scan the next column, and one for the debouncing timeout (timeouts parameterized by the constants TEMPO and TEMPO2). Finally, the UART interface in DMA mode:    And here our interrupt table:    In the “main.c” file we declared a struct type to store all ports and pins of the keyboard, this for easiness of use, by accessing the data structures *ROWS* and *COLS* with indexing:    To conclude the declaration part, out global variables (all set to 0 except for *c\_old*, initialized to -1 to be different from *c*, and for *buttons[]*, array of characters set as indicated on our physical board):    …  Finally, … | | | |
| Professor comments: | | | |