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

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A2* | | |
| Homework number: | *09* | | |
| Due date: | 24/11/2024 | | |
|  |  |  |  |
| 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:  **Part 1a:**  First of all, we configure the board pinout for the SPI communication protocol enabling SCK (PA5), MISO (PA6) and MOSI (PA7), then we set the SPI1 communication to Full-Duplex Master. We configure also the PB6 pin as GPIO\_Output to handle the SET and RESET of the SS (RCLK for the shift registers) pin.    From “connectivity*”*, in addition to the Mode, we set the Prescaler to “4” to avoid the bug:    We enable the DMA2 as shown in image:    We set the TIM2 that will be used to print in series each different column of the LED Matrix (timeout parameterized by the constant TEMPO):    Finally, we enable the SPI1 and TIM2 interrupts.    In the “main.c” file we declared a struct named *rowcolumn*. This struct contains two integer values: the first one indicates the LEDs of the column to turn on and the second one specifies the column.    Then we declared two arrays of struct to recreate the letters ‘A’ and ‘H’.    We defined the variables:   * *i* indicates the current column that we are controlling; * *size* is the number of bytes we send through the SPI; * *letter\_flag* is a flag used to select which letter to write: A = 0, H = 1; * *letter\_timer* is a counter that keeps track of the time a letter has been showed * *showing\_time* is the time a letter has to stay visible on the LED matrix;     In the main() we initialize the TIM2:    The TIM2 timer triggers the callback every 4 ms. When this happens the GPIO\_PIN\_6 (RCLK) is set to ‘0’ to not copy the value of the column in the output latches yet (it will be done afterward) and depending on the flag(A = 0, H = 1), a column of the letters is transmitted via SPI using DMA (the column represented by the index *i*). Since we have 5 columns with a delay of 4ms per column we have that the whole writing procedure of a letter takes place every 20ms (50Hz).  *Letter\_timer* is increased and the letters are displayed for the correct amount of time before switching between them.    *HAL\_SPI\_TxCpltCallback* is a callback function for the SPI transmission completion interrupt, when the *HAL\_SPI\_Transmit\_DMA* function completes the transmission the GPIO PIN is set to ‘1’ in order to turn on the LEDs and a check on *i* is done to verify if the last column has been reached, if so we restart from the first one, otherwise, we increment *i*.      **Finally, here our result in the physical environment, obtainable thanks to the high frame rate video function of our camera, where we can experience the values printing on each column at a time, every 4 milliseconds:** | | | |
| Professor comments: | | | |