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

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A5* | | |
| Homework number: | *HOMEWORK 09* | | |
| Due date: | 24/11/2024 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| Alessio Spineto |  |  | *x* |
| Riccardo Lamarca | *x* |  |  |
| Sofia Cecchetto |  |  | *x* |
| Annamaria De Togni |  |  | *x* |
| Emma Crespi |  |  | *x* |
| Notes: none | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | Matrix of LEDs | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Completed |
|  |  |  |  |
| The objective of the homework was to transmit a letter to the LED matrix using SPI, using a timer interrupt to alternate between two letters(or one letter and one symbol).  We decided to transmit the letters P and C.  The first step was to locate and set the correct pins on the GUI.  On the schematic we identified the pins corresponding to SCK, MISO and MOSI, respectively PA5, PA6 and PA7. RCLK is the pin corresponding to the SS line in the SPI communication.  Inserimento dell'immagine in corso...  We configured the SPI communication in the GUI By setting CPOL on low, and CPHA on 1 Edge (value 0), meaning that the sampling is done on the falling edge of the SCK clock.    We set also 2 timers. TIM2 settings to obtain a frequency of 2 kHz:    TIM3 settings to obtain a frequency of 1 Hz:    In the main function we started the two timers, TIM2 for sending the data to configure the LED matrix, and TIM3 for changing the letter/symbol to display.  We also enabled TIM2 and TIM3 global interrupts.    Next, before defining the callbacks, we need to declare a variable to store the data to send via SPI.  We created a struct called *intersection\_s* made of 2 uint8\_t elements, to define the column and the row to turn on.  Since each letter is displayed thanks to multiple intersections of rows and columns, we created an array made of multiple *intersection\_s*.  To know the exact intersections of rows and columns to display each letter, we referred to the excel file on Webeep.    To allow the transmission of the data coding for the intersection of rows and columns, we used the callback of TIM2.  We first store the current pair column-row of the letter we are displaying in the variable *data*. We use the variables “flag” to alternate between letters and “counter” to cycle through columns.  Then we transmit with SPI in DMA mode the data to both shift registers (one controlling the columns, the other one the rows). The corresponding data are initially stored in the *shifting* part of both shift registers. To allow the data to reach the LED matrix, and so to effectively turn on the LEDs, we need to transfer the data from the shifting register to the storage one (both inside each shift register of column and rows). To do so, we need to enable the line connecting the two, called RCLK (slave selector line), which corresponds to pin PB6. | | | |
| Professor comments: | | | |