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

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *B5* | | |
| Homework number: | *07* | | |
| Due date: | Tuesday, 21st November, 08:30 | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| Ghidini Alessandro |  |  | *x* |
| Latino Francesco |  |  | *x* |
| Luppi Eleonora |  |  | *x* |
| Bravin Riccardo |  |  | *x* |
| Feltrin Elia |  |  | *x* |
| Notes: | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | *LED matrix – Single and multiple letters* | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Completed |
|  |  |  | *x* |
| Explanation:  The homework goal is showing some symbols or letters on the embedded LED matrix.  In the first part we used the SPI protocol to communicate with the LED matrix and send one single letter which is then shown by turning the LED on the correct point of the matrix.  The second part is quite similar, in addition it is requested to make two or more letters alternating and being showed on the LED matrix.  **Part 1a:**  In this project we need to use SPI protocol communication to draw a letter using the LED matrix.  Firstly, we enable PA5 as SPI1\_CLK and PA7 as SPI1\_MOSI.  PB6 is set as GPIO\_Output to use it as shift register clock to correctly send the values to the LED matrix.  To be able to communicate with the LED matrix we need to set up the SPI1 as a *Transmit Only Master* with a prescaler of 4. Then, to light the columns of the LED matrix sequentially and repeatedly we set up a timer (TIM3) with a prescaler value of 8400-1 and a counter period of 40-1, to make the five columns alternate with a frequency in a way that the human eye is not capable to perceive when each of the columns is turned on, but only the whole columns light.  Once the code has been generated, in the main.c file we define an index to shift the columns and one matrix containing the value pairs calculated through the excel file.  We start the timer in interrupt mode.  In the callback function of the timer, *HAL\_TIM\_PeriodElapsedCallback*, we transmit the data using *HAL\_SPI\_Transmit(&hspi1, matrix\_A[column\_shift], 2, 10)*, we toggle the SRCLOCK pin and we increment the value of the shift column.  Once the shift columns reach the value of 5 it is initialized by zero again.  **Part 1b:**  The setting of the GUI is the same as the previous project except for the DMA: in the SPI tab we also enable *SPI1\_TX DMA* request and the related interrupt. Once we generate the code, we define again a shift counter and a counter for the time as well. Within the same matrix we write the values calculated from the excel file to display two different letters.  We start the timer in interrupt mode.  In the callback of the timer we use the function *HAL\_SPI\_Transmit\_DMA()* to transmit the data for each column and increment the shift column counter as previously done. This time we need to maintain one letter for a sufficient time in order to be perceived, but also we need to check when to switch each letter. To do this, each time the shift column reaches the value of 5 we increment a counter that takes the number of cycles.  By doing this, when this repetition reaches the value of 50 it means the letters has been displayed for 1 second and the code can restart by displaying the other letter which means starting from the shift column counter equal to 5. This cycle is then repeated for the second letter and once finished it restarts again by displaying the first one.  Finally, we use the *HAL\_SPI\_TxCpltCallback* to make toggle the RSCLK once data are transferred. | | | |
| Professor comments:  The report us too poor in details to be fully evaluated. You should add some screenshot of the code for clarity. | | | |