|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A1* | | |
| Homework number: | *8* | | |
| Due date: | *27/11/2022* | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| 1 *Giorgio Donato Carlo* |  |  | *x* |
| 2 *Lenzi Francesco* |  |  | *x* |
| 3 *Lodari Gianmarco* |  |  | *x* |
| 4 *Lanzini Alessio* |  |  | *x* |
| 5 *Chiapparo Lenn* |  |  | *x* |
| Notes:  *Complete in necessary* | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | Alternate between two letters 1a | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation: We set all the pins for the SPI, we used SPI1 in Transmit Only Master mode. We set TIM2 with the following parameters in order to obtain an interrupt signal every 4 ms: PSC = 84-1, ARR = 4000-1.  To represent each character that could be printed in the led matrix we used an array of “letter” struct, where each of them is composed of two uint8\_t variables, row and column.  In our case, we printed “F” on the led matrix (**struct** matrix letter[] = {{127, COL1},{104, COL2},{72, COL3},{72, COL4},{64, COL5}};  When HAL\_TIM\_PeriodElapsedCallback is called we transmit row and column value and we set the leds through HAL\_GPIO\_WritePin function.  After that we increment a variable that permit us to send the next row and callback at the next interrupt. | | | |
| Professor comments:  Good! | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | Alternate between two letters 1b | | |
| Not done | Partially done  (major problems) | Partially done  (minor problems) | Successfully completed |
|  |  |  | *x* |
| Explanation: We set all the pins for the SPI, we used SPI1 in Transmit Only Master mode and we enabled its DMA in normal mode. As in the class session, we set TIM2 with the following parameters in order to obtain an interrupt signal every 4 ms: PSC = 84-1, ARR = 4000-1.  To represent each character that could be printed in the led matrix we used an array of “character” struct, where each of them is composed of two uint8\_t variables, row and column.  In order to have a more flexible and scalable solution, we decided to implement a program that can print a sequence of not only two, but n symbols, where n is selected by the user: this can be done by modifying the length of the sequence in the define section of the program, and then, as we did in the main with ‘A’ and ‘B’, insert all the encoded symbols in “word”, an array of pointer to characters.  In the while(1) of the main, using HAL\_Delay(1000), we iterate thorough the symbols present in ‘word’ by changing the value of the global variable characterIndex.  In the HAL\_TIM\_PeriodElapsedCallback, when triggered by TIM2, we send the correct data using HAL\_SPI\_Transmit\_DMA, in order to turn on the right column of the right symbol.  Then, in the HAL\_SPI\_TxCpltCallback, we toggle PIN\_B6,update columnIndex, a global variable that indicates which column should be enabled in the next 4 ms, and finally we toggle again PB6. | | | |
| Professor comments:  Very good! | | | |