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

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A2* | | |
| Homework number: | *HOMEWORK 03* | | |
| Due date: |  | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| Hui Jiang |  |  | *x* |
| Mattia Sironi |  |  | *x* |
| Gabriele Landi |  |  | *x* |
| Arturo Caliandro |  |  | *x* |
| Luigi Lizzini |  |  | *x* |
| Notes:  none | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name |  | | |
| Not done | Partially done   (major problems) | Partially done   (minor problems) | Completed |
|  |  |  |  |

|  |
| --- |
| We have done the whole homework, here is our explanation:  **Part 1:**  Firstly, we have configured the board as shown below:    As USART\_TX and USART\_RX by default are configured properly, in this project, we haven’t done any modification to the board.  Secondly, we have configured the UART3 and DMA as shown in the slides:    Thirdly, in order to control the sending frecuency to 1Hz, we have configured the timer 2 instead of using the HAL\_Delay() function:  We just set the Clock Source to Internal Clock and in order to get a sending frequency 1Hz, we set the prescaler to 8399 and Period to 9999.(As (8399+1)\*(9999+1) = 84MHz)  Fourthly, we have enabled the TIM2 global interrupt(in order to control the sending frequency) and USART2 global interrupt as shown below:    Fifthly, we move to the “main.c”, we firstly define a global variable to save the message that the board is going to send:  Sixthly, we set the valur of string to “Hui 2001” using snprintf , and send the first message using the function HAL\_UART\_Transmit\_DMA. Then we set the TIM2 to generate a interrupt when the timer reaches its maximum value and we start the timer in a secure way:    The descriptions of HAL functions we have use in this step are shown below:      Finally, we created our callback function as shown below:  As in this project we have just used one single timer, we don’t need to check if it’s the proper timer in the callback function and the only thing the callback function does is to send the message.  And, we have proved it using the board:  **Part 2:**  Firstly, we have added “PMDB16\_LCD.h” and “PMDB16\_LCD.c” in Inc and Src respectively:    Secondly, we have configured the board as the description in the slide:  Thirdly, we have used the timer a control the changing frequency on the LCD:    And once again, Prescaler 8399 and Period to 9999 to make the LCD change the content every second.  Fourthly, we enabled the TIM2 global interrupt in order to change the LCD content every second in the NVIC table as shown below:  Then, in the main.c, we firstly add the PMDB16\_LCD.h to be able to use those functions in the slides:  Later, we defined a series of global variables as shown below:  Where the name is the variable to save the name of every partner in the group, our\_index (we firstly defined as index, but the compiler complains that in the PMDB16\_LCD.c there is already a variable which name is index and type char, so we redefined as our\_index) to indicate the name is displaying on the bottom of the LCD. And the variable last\_name\_ptr to indicate the last name which was displaying in the bottom of the LCD.  After that, we have defined some functions:    The function increment\_index() is just to increment the variable our\_index, and thenit reaches 4, reset to 0 again. The function printname() is to print the name of our partners: first in the top part of the LCD, we print the previous name which was at the bottom of the LCD and then print the “new” name at the bottom, then save the “new name” to the last\_name\_ptr in order to print it on the top of the LCD in the next iteration. And increment the index in order to print the next “new” name on the bottom of the LCD in the next iteration.  Then, in the main:    We first initialize the LCD, turn on the LCD backlight and clear the previous content of the LCD. Then we call the printname. As you can see, the variable last\_name\_ptr in this point is empty, therefore, the function printname doesn’t print anything on the top of the LCD this time, but will print the first element in the variable name on the bottom of the LCD. Then as the part1, we set the timer to generate a interrupt everytime the timer reaches its maxium value and we start the timer in a safe way.  Finally, we created the callback function:    Which just clear the previous content of the previous iteration and call the printname().  And we have proved it using the board. |
| Professor comments: |