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

|  |  |  |  |
| --- | --- | --- | --- |
| Team name: | *A2* | | |
| Homework number: | *HOMEWORK 03* | | |
| Due date: | *15/10/23* | | |
|  |  |  |  |
| Contribution | NO | Partial | Full |
| Hui Jiang |  |  | *x* |
| Mattia Sironi |  |  | *x* |
| Gabriele Landi |  |  | *x* |
| Arturo Caliandro |  |  | *x* |
| Luigi Lizzini |  |  | *x* |
| Notes:  none | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Project name | DMA and LCD | | |
| Not done | Partially done   (major problems) | Partially done   (minor problems) | Completed |
|  |  |  | *x* |

|  |
| --- |
| We have done the whole homework, here is our explanation:  **Part 1:**   1. We have configured the board as shown below:     As USART\_TX and USART\_RX are configured properly by default, in this project, we haven’t done any modification to the board.  2. We have configured the UART3 and DMA as shown in the slides:    3. In order to set the transmission frequency 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 (84MHz / (8399+1)\*(9999+1) = 1Hz).  4. We have enabled the TIM2 global interrupt (in order to control the transmission frequency) and USART2 global interrupt as shown below:    5. In “main.c”, we firstly define a global variable to save the message that the board is going to send:    6. We set the value 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:      7. 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:**   1. We have imported the LCD library inside the project:      1. We have configured the board as the description in the slide: 2. We have used TIM2 to control the update frequency of the LCD:     As in part 1, Prescaler is set to 8399 and Period to 9999 to make the LCD change the content every second.   1. We enabled the TIM2 global interrupt in order to change the LCD content in the NVIC table as shown below: 2. In the main.c, we included the PMDB16\_LCD.h to be able to use those functions in the slides:      1. We defined a series of global variables as shown below:   Where *name* contains the name of every partner in the group, *our\_index* is the position in *name* ofthe name to display the LCD, and *last\_name\_ptr* to point to the last name which was displayed on the bottom of the LCD.   1. we have defined some functions:   *print\_name* prints the previous name (i.e. *last\_name\_ptr*) on the upper part of the LCD and the next name in *name* on the lower part, then saves the latter in *last\_name\_ptr* and updates the index.   1. In the main:   We first initialize the LCD, turn on the LCD backlight and clear the previous content of the LCD, then we call *print\_name*. The next part is the same as the part1.   1. We created the callback function:     Which just clears the content of the previous iteration and executes *print\_name*.  The code works as expected. |
| Professor comments: |