Mark	**0/1**

Team name:	*A1*		
Homework number:	*HOMEWORK 3*		
Due date:	15/10/23		
Contribution	NO	Partial	Full
Monti Pietro			*x*
Moretto Alessia			*x*
Pallotto Francesco			*x*
Perna Alessandro			*x*
Ventura Ludovico			*x*
Notes:			

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

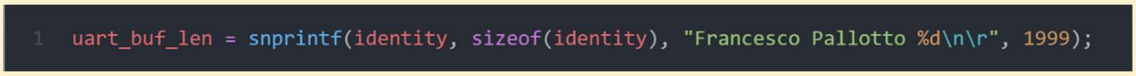
Explanation:
We successfully completed the homework.

Part 3a:

To complete the UART project adding the exploitation of the DMA, we enabled the USART2 global interrupt from the NVIC interrupt table in the GUI, as the DMA operates on data by means of it.

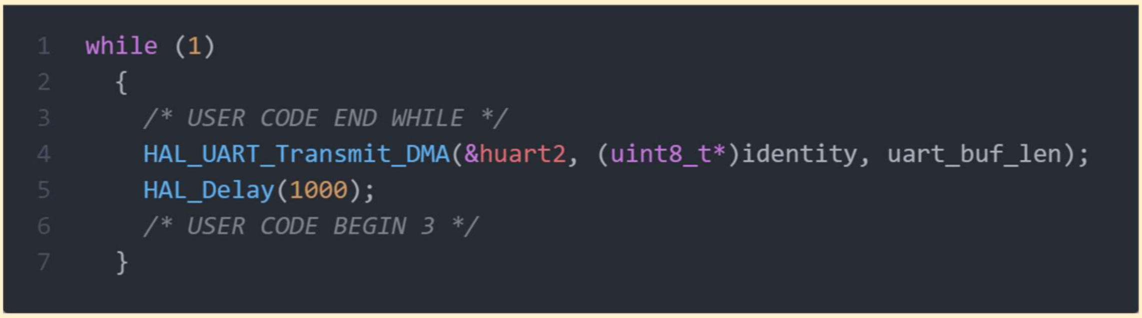
A string called “identity” is defined to store the required information, as well as an int16_t variable (“uart_buf_len”) used to store the length of the data to be transmitted via UART.

We used *snprintf* to format the identity information into the identity buffer, ensuring that it doesn't exceed the buffer size:



```c
uart_buf_len = snprintf(identity, sizeof(identity), "Francesco Pallotto %d\n\r", 1999);
```

Finally, within the while loop we transmitted the identity string over UART by means of *HAL_UART_Transmit_DMA*, providing the necessary parameters. After each send, *HAL_Delay* is used to achieve the desired time interval:



```c
while (1)
  {
    /* USER CODE END WHILE */
    HAL_UART_Transmit_DMA(&huart2, (uint8_t*)identity, uart_buf_len);
    HAL_Delay(1000);
    /* USER CODE BEGIN 3 */
  }
```

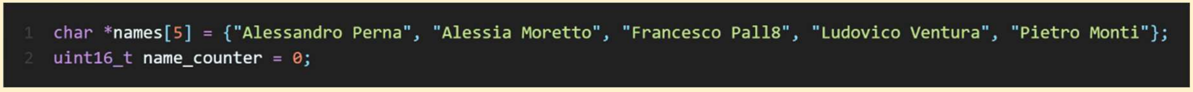
Part 3b:

Here we detail the process of implementing a name scrolling feature on an LCD screen. The implementation involves the use of an array of strings containing group names, a uint16_t variable named "name_counter," and the TIM2 timer. The purpose of this implementation is to display names from the array in a scrolling fashion on the LCD screen.

Implementation Steps:

Initialization of Array and Variables:

First, we declare a simple array of strings containing all the names of the group.
We also initialize a `uint16_t` variable called "name_counter" to facilitate the scrolling through the array on the LCD screen. The "name_counter" variable keeps track of the current name to display.

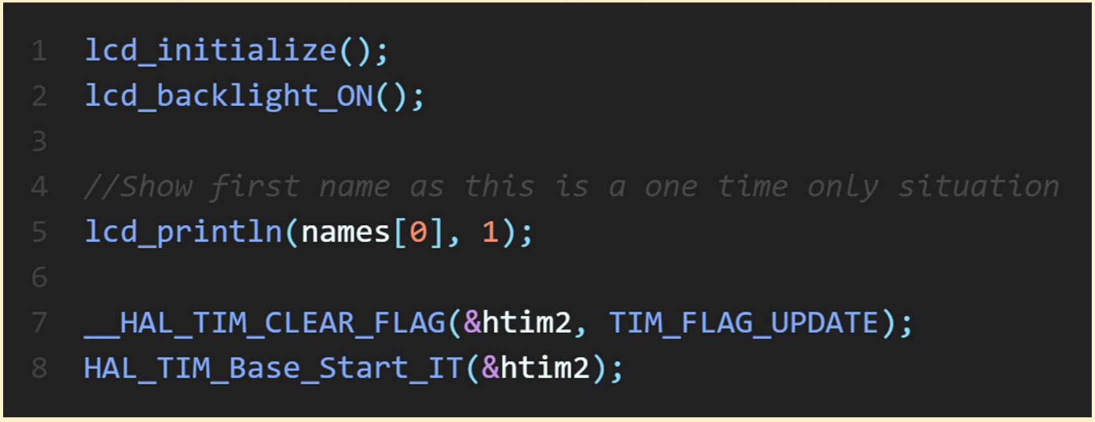


```c
char *names[5] = {"Alessandro Perna", "Alessia Moretto", "Francesco Pall8", "Ludovico Ventura", "Pietro Monti"};
uint16_t name_counter = 0;
```

Note: Regrettably, Francesco Pallotto's complete name exceeds the 16-character limit for one row on the LCD screen, necessitating the abbreviation "Pall8."

Initialization of LCD & TIM2 Timer:

The functions **`lcd_initialize()`** and **`lcd_backlight_ON()`** are used for initializing the LCD and turning on its backlight.
These functions are declared in the "PMDB16_LCD" file, which was imported into the project files and then included in the main.c file.
To create an initial display where the first row is empty and the first name is shown on the second row, we print the first name from the names array.
After that, we initialize the TIM2 timer. The TIM2 timer is configured to trigger once every second.
A prescaler of 8400-1 and a period of 10000-1 are set to achieve this timing.
The timer's flag is cleared to prevent it from triggering immediately upon activation.



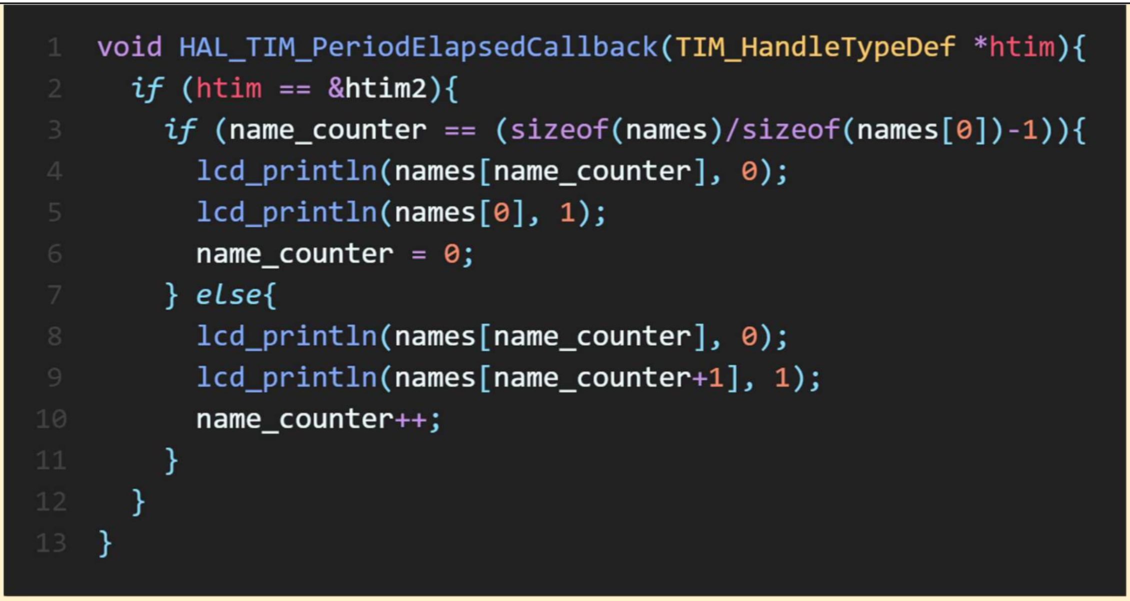
```c
lcd_initialize();
lcd_backlight_ON();

//Show first name as this is a one time only situation
lcd_println(names[0], 1);

__HAL_TIM_CLEAR_FLAG(&htim2, TIM_FLAG_UPDATE);
HAL_TIM_Base_Start_IT(&htim2);
```

Scrolling Logic in TIM2 Callback:

The logic to scroll names on the LCD is implemented in the TIM2 callback.
During each timer loop, the name corresponding to the "name_counter" position in the array is printed on the first row, and the following name is printed on the second row.
The "name_counter" is then updated to point to the next name in the array.
When the last name is reached, it is printed on the first row, and the name in position 0 of the array is printed on the second row. The "name_counter" is reset to 0.
This logic creates the desired scrolling effect on the LCD screen, where names continuously move from one row to the other.



```c
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim){
  if (htim == &htim2){
    if (name_counter == (sizeof(names)/sizeof(names[0])-1)){
      lcd_println(names[name_counter], 0);
      lcd_println(names[0], 1);
      name_counter = 0;
    } else{
      lcd_println(names[name_counter], 0);
      lcd_println(names[name_counter+1], 1);
      name_counter++;
    }
  }
}
```

Professor comments: