

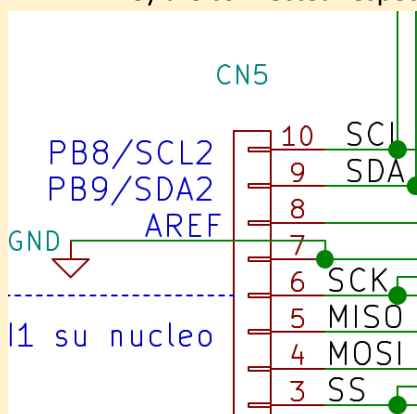
Mark	1/11
------	------

Team name:	B5		
Homework number:	HOMEWORK 09		
Due date:	26/11/24		
Contribution	NO	Partial	Full
Marenghi Manuela			x
Fellegara Tommaso			x
Giammusso Samuele			x
Cattani Luca			x
Csata Dániel			x
Notes: none			

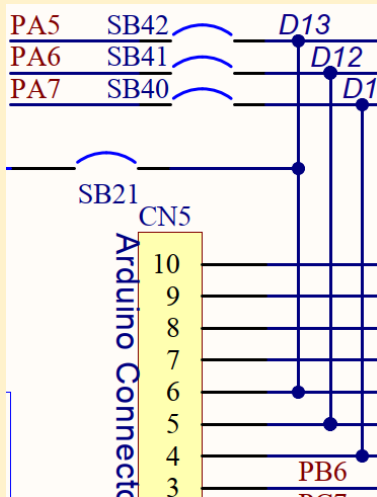
Project name	Test		
Not done	Partially done (major problems)	Partially done (minor problems)	Completed
			x

### LED Matrix (SPI) project:

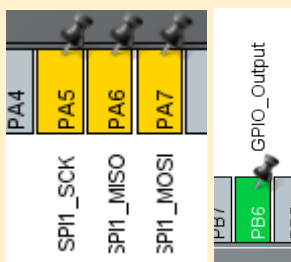
- Setup the pins:
  - First of all we search for SCK, MISO, MOSI, SS in the Hands-on lab schematic  
They are connected respectively to the pins 6, 5, 4, 3 of the arduino connector CN5



- Then we look for them in the Nucleo Schematic:  
They are connected respectively to pin PA5, PA6, PA7, PB6



-Then we set them in the IDE:



- Enable the SPI1:
  - Enable it in half-duplex Master mode (because it only needs to send and not receive).
  - Set the prescaler to 4
  - Enable the DMA, by adding the SPI1\_TX
- Setup the Timers:
  - Enable Timer 2: (used for changing the column of led of the matrix every 4ms)
    - Set the clock source as 'Internal'.
    - Then, we want to obtain a delay of 4ms, this means a frequency of  $1/4\text{ms} = 250\text{Hz}$ .
    - So we set the Prescaler at 8400-1, and the counter at **40-1** to obtain that frequency.
    - Then, enable the TIM2 global interrupt
  - Enable Timer 3: (used for changing the letter every 1s)
    - Set the clock source as 'Internal'.
    - Then, we want to obtain an interrupt every 1s,
    - so we set the Prescaler at 8400-1, and the counter at 10000-1 to obtain that frequency.
    - Then, enable the TIM3 global interrupt
- How we implemented the code:
  - In the main, enable the Timers in interrupt mode and initialize the PIN\_6 to SET state.

```

159  /* USER CODE BEGIN 2 */
160      HAL_TIM_Base_Start_IT(&htim2);
161      HAL_TIM_Base_Start_IT(&htim3);
162      HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_SET);

```

-Then declare this private variables:

- coord: we choosed a three-dimensional matrix to store the values of each letter, because with this solution it's easier to iterate and it's more extensible in the future.
- i and j are used later respectively to cycle over the columns and the letter.
- numElements: is useful to know how many characters are in coord.
- num\_col: is useful to know the number of columns, which is the same of the led matrix.

```

51 /* USER CODE BEGIN PV */
52
53 uint8_t coord[4][5][2]={ //letter, row, column
54     {{31,16}, //A
55      {36,8},
56      {68,4},
57      {36,2},
58      {31,1}},
59
60     {{127,16}, //B
61      {73,8},
62      {73,4},
63      {73,2},
64      {54,1}},
65
66     {{62,16}, //O
67      {65,8},
68      {65,4},
69      {65,2},
70      {62,1}},
71
72     {{99,16}, //X
73      {20,8},
74      {8,4},
75      {20,2},
76      {99,1}},
77
78 };
79
80 int i=0,j=0;
81 int numElements = sizeof(coord) / sizeof(coord[0]);
82 int num_col= sizeof(coord[0])/sizeof(coord[0][0]);

```

-Then we defined the timer callback:

-if it's a timer 2 interrupt: we firstly stop (and then restart) the timer in order to obtain a precise 4ms delay. Then, if the DMA is Ready, we put the SS signal to the RESET state of the peripheral and then the DMA sends the led values of a column. When the DMA has finished the callback is executed and the cpu puts the SS signal to SET state.

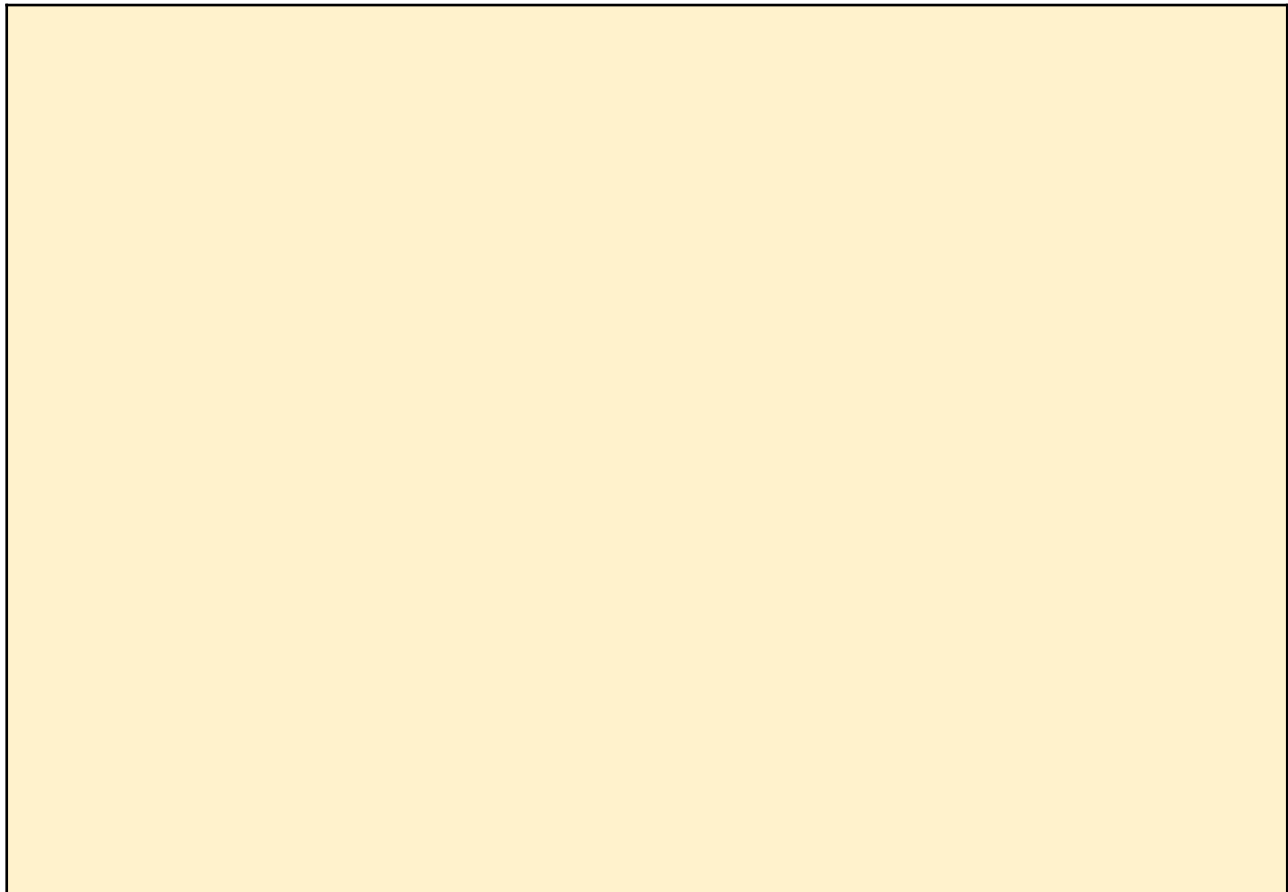
In the end we use the module operation to iterate over the columns.

-if it's a timer 3 interrupt: we use the module operation to iterate over the letters of coord.

```

99 void HAL_SPI_TxCpltCallback(SPI_HandleTypeDef* hspi) {
100     if(hspi->Instance == SPI1) {
101         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_SET);
102     }
103 }
104
105 void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) {
106     if (htim->Instance == TIM2) { //tim2 every 4ms, change column
107         HAL_TIM_Base_Stop_IT(&htim2);
108
109         if( HAL_SPI_GetState(&hspi1)==HAL_SPI_STATE_READY ) {
110             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_6, GPIO_PIN_RESET); //pulse the RCLK, PIN PB6
111             HAL_SPI_Transmit_DMA(&hspi1, coord[j][i], 2); //transmit the column
112         }
113
114         i=(i+1)%num_col; //iterate over the 5 columns
115
116         HAL_TIM_Base_Start_IT(&htim2);
117     }
118
119     if (htim->Instance == TIM3) { //tim3 every 1s, change letter
120         j=(j+1)%numElements;
121     }
122 }

```



Professor comments:

You don't need to stop and restart the TIM in the TIM interrupt routine.