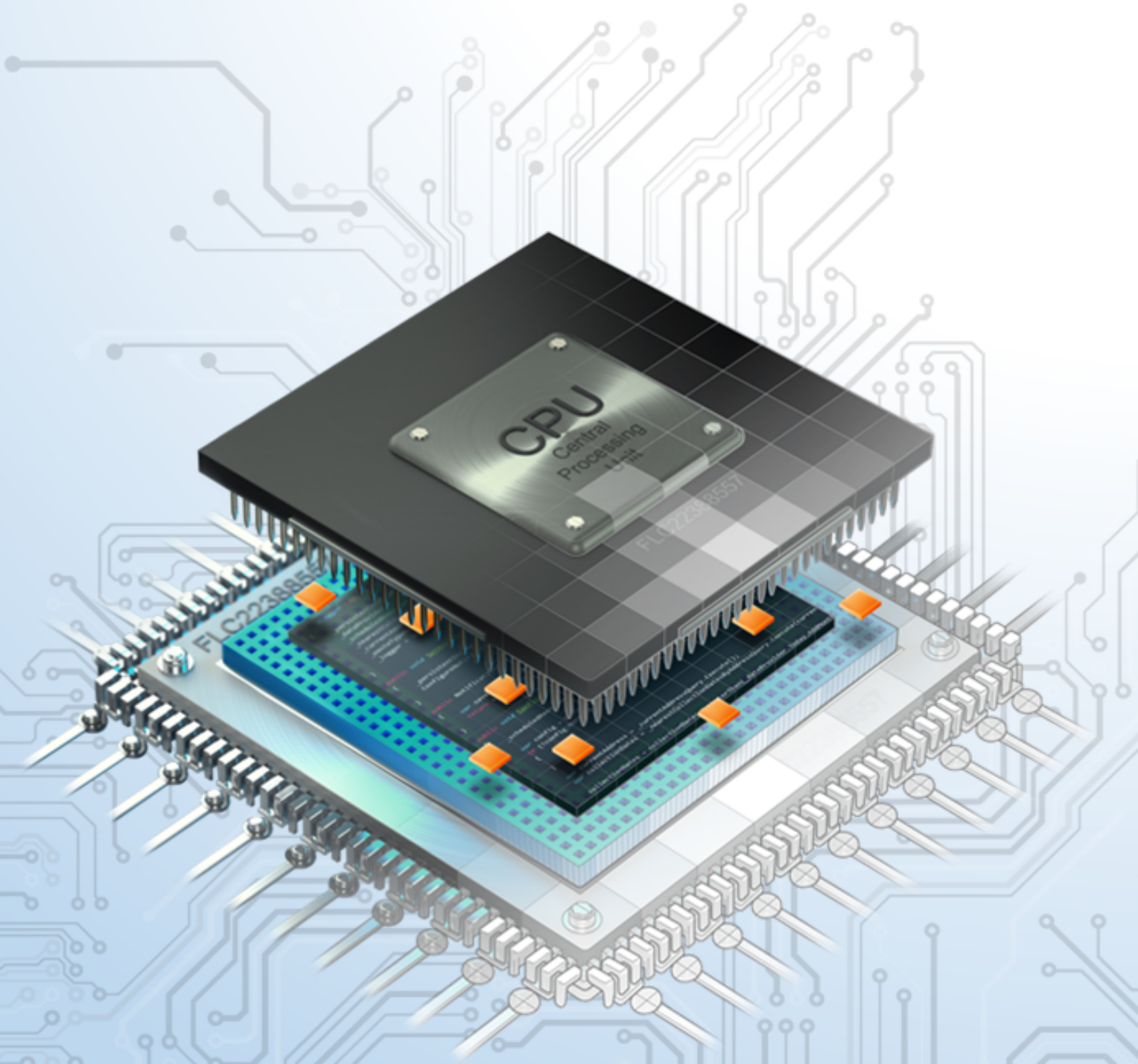
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller

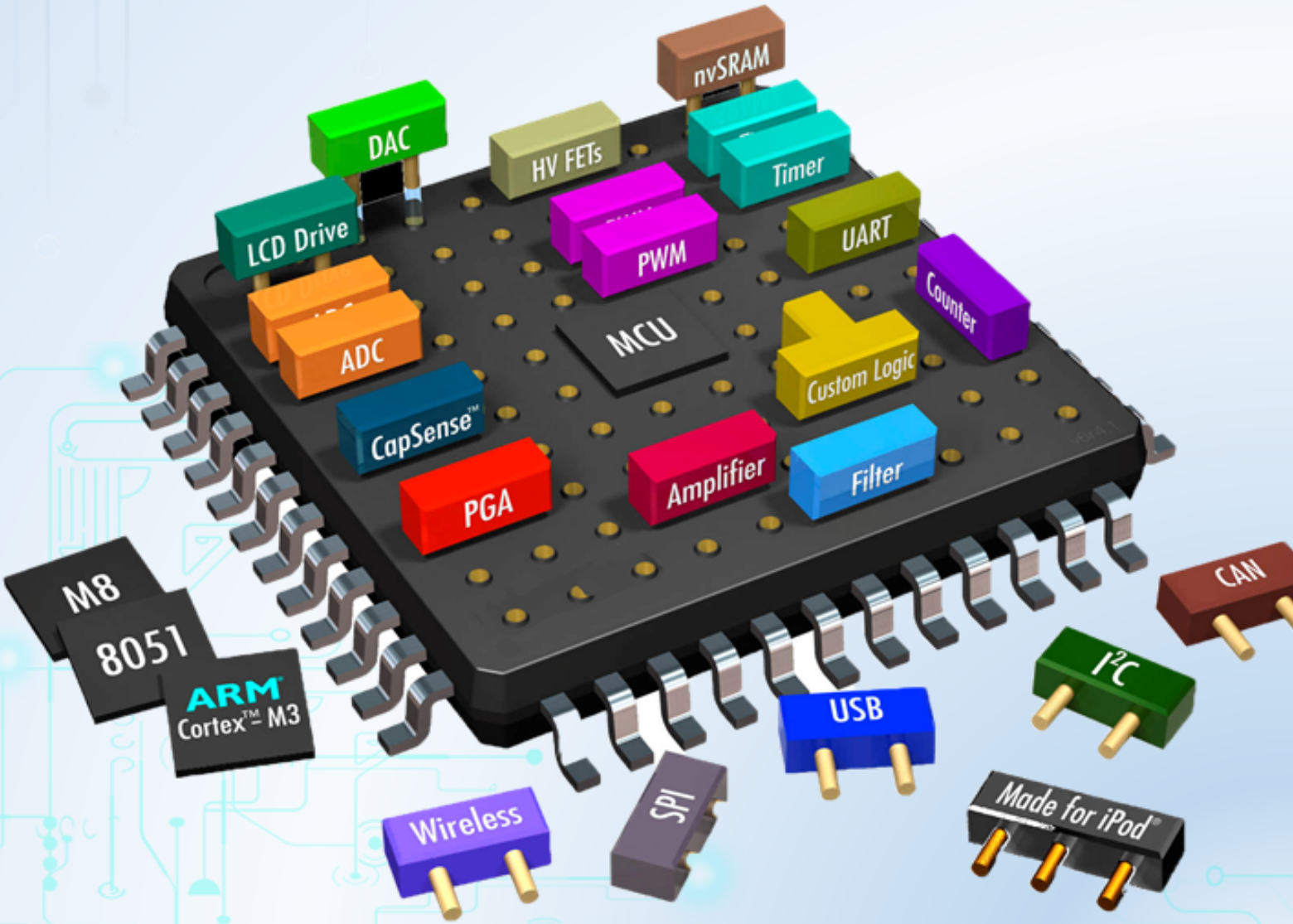


Dr. Le Trong Nhan

Mục lục

CHƯƠNG 1

LED Animations

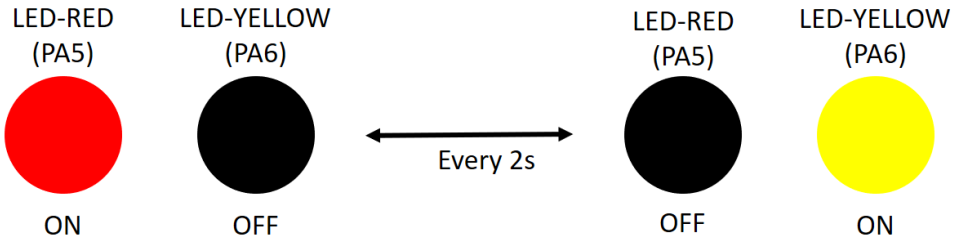


Exercise and Report

1 Exercise 1

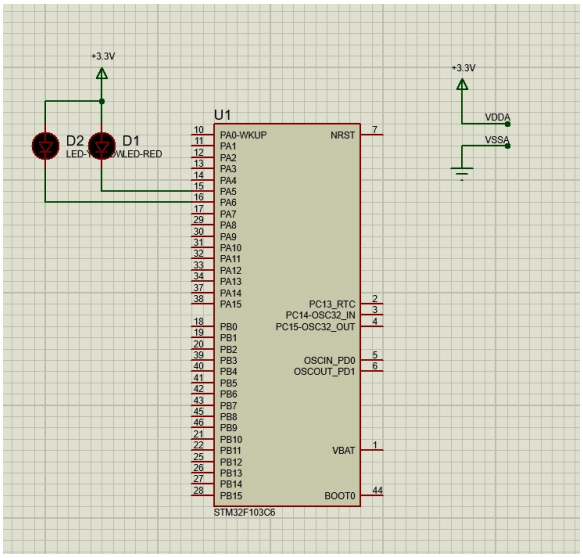
From the simulation on Proteus, one more LED is connected to pin **PA6** of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is **LED-YELLOW**, which can be found from the device list.

In this exercise, the status of two LEDs are switched every 2 seconds, as demonstrated in the figure bellow.



Hình 1.1: State transitions for 2 LEDs

Report 1: Depict the schematic from Proteus simulation in this report. The caption of the figure is a downloadable link to the Proteus project file (e.g. a github link).



Report 2: Present the source code in the infinite loop while of your project. If a user-defined functions is used, it is required to present in this part. A brief description can be added for this function (e.g. using comments). A template to present your source code is presented bellow.

```
1  int led_status = 1; /*1 = red, 2 = yellow)*/
2  int count = 2;
```

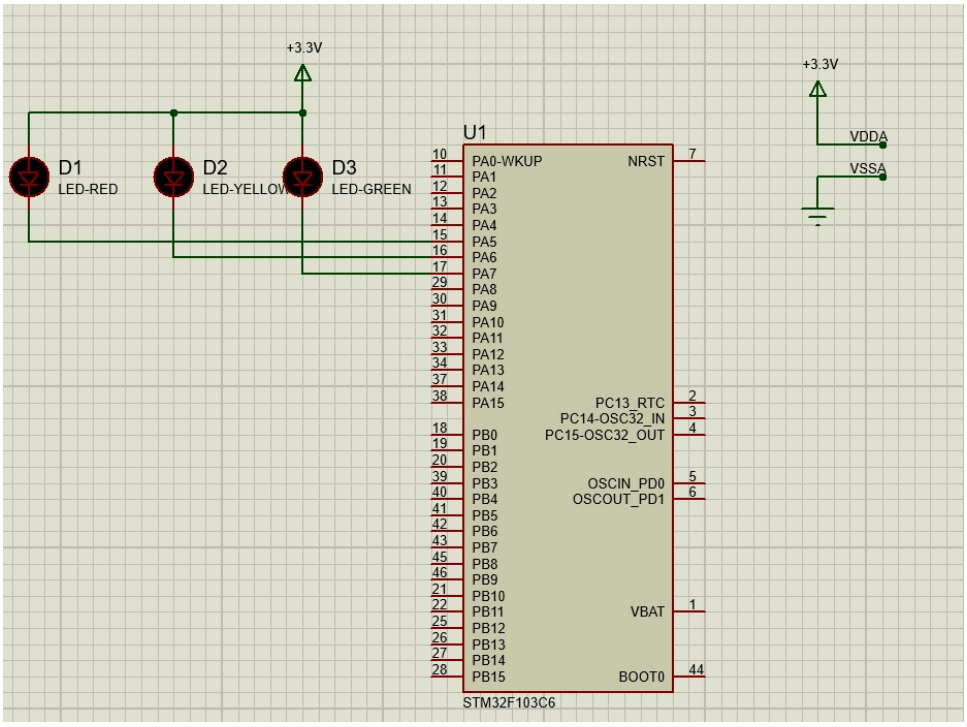
```
  while (1)
  {
  switch(led_status){
  case 1:
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
   GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
    GPIO_PIN_SET);
    count--;
    if(count == 0){
      led_status = 2;
      count = 2;
    }
    break;
  case 2:
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
    GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
   GPIO_PIN_SET);
    count--;
    if(count == 0){
      led_status = 1;
      count = 2;
    }
    break;
  default:
    break;
  }
  HAL_Delay(1000);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
```

2 Exercise 2

Extend the first exercise to simulate the behavior of a traffic light. A third LED, named **LED-GREEN** is added to the system, which is connected to **PA7**. A cycle in this traffic light is 5 seconds for the RED, 2 seconds for the YELLOW and 3 seconds for the GREEN. The LED-GREEN is also controlled by its negative pin.

Similarly, the report in this exercise includes the schematic of your circuit and a your source code in the while loop.

Report 1: Present the schematic.



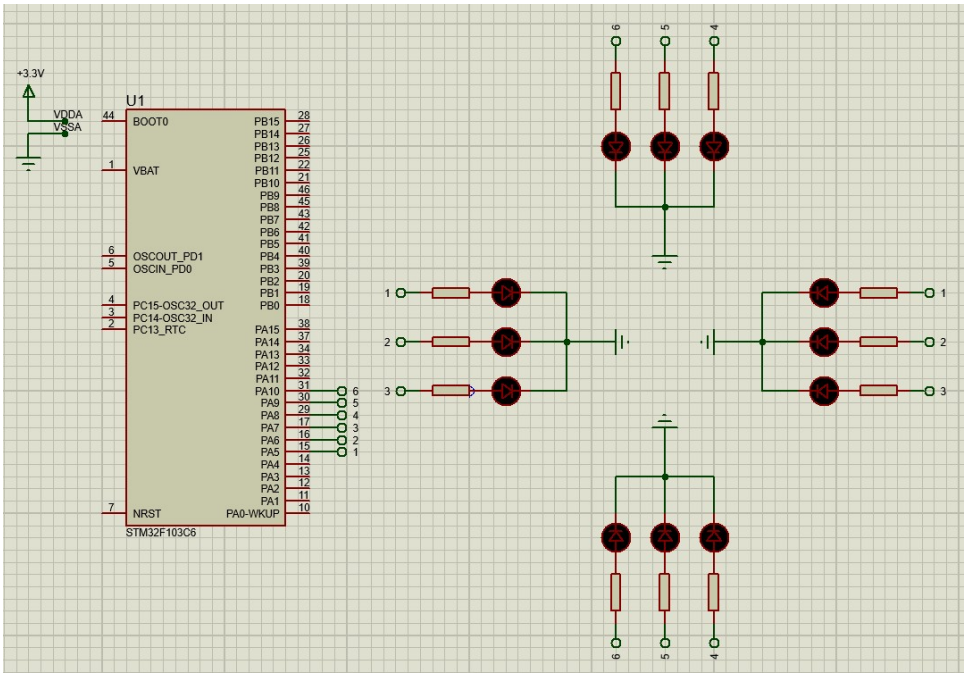
Report 2: Present the source code in while.

```
int led_status = 1; /*1 = red, 2 = yellow, 3 = green*/
int red_count = 5;
int yellow_count = 2;
int green_count = 3;
while (1)
{
switch(led_status) {
  case 1: // RED -> GREEN
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
 GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
 LED_YELLOW_Pin, GPIO_PIN_SET);

    red_count--;
    if(red_count == 0){
      led_status = 3;
      red_count = 5;
    }
    break;
  case 2: // YELLOW -> RED
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
 LED_YELLOW_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin,
  GPIO_PIN_SET);

    yellow_count--;
    if(yellow_count == 0){
      led_status = 1;
```

```
25          yellow_count = 2;
26        }
27        break;
28      case 3: // GREEN -> YELLOW
29        HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin,
    GPIO_PIN_RESET);
30        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
   GPIO_PIN_SET);
31
32        green_count--;
33        if(green_count == 0){
34          led_status = 2;
35          green_count = 3;
36        }
37        break;
38      default:
39        break;
40    }
41    HAL_Delay(1000);
42      /* USER CODE END WHILE */
43
44      /* USER CODE BEGIN 3 */
45    }
```

3 Exercise 3

Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light.
Schematic:



Source code:

```
int led_status = 1; /*1 = red, 2 = yellow, 3 = green*/
int red_count = 5;
int yellow_count = 2;
int green_count = 3;
while (1)
{
switch(led_status) {
  case 1: // RED -> GREEN
    // WEST - EAST
    red_count--;
    HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port, LED_YELLOW1_Pin, GPIO_PIN_RESET);

    if(red_count > 1){
      // NORTH - SOUTH
      HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin, GPIO_PIN_SET);
      HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin, GPIO_PIN_RESET);
    }
    else if(red_count == 1){
      // NORTH - SOUTH
      HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port, LED_YELLOW2_Pin, GPIO_PIN_SET);
      HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin, GPIO_PIN_RESET);
```

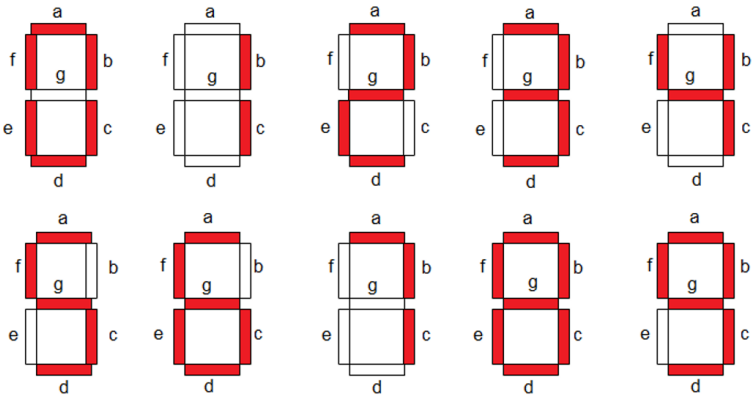
```
23        }
24        else{
25          led_status = 3;
26          red_count = 5;
27        }
28
29        break;
30      case 2: // YELLOW -> RED
31        yellow_count--;
32        // WEST - EAST
33        HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,
    LED_YELLOW1_Pin, GPIO_PIN_SET);
34        HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port,
    LED_GREEN1_Pin, GPIO_PIN_RESET);
35        // NORTH - SOUTH
36        HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
    GPIO_PIN_SET);
37
38        if(yellow_count == 0){
39          led_status = 1;
40          yellow_count = 2;
41        }
42        break;
43      case 3: // GREEN -> YELLOW
44        // WEST - EAST
45        green_count--;
46        HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port,
    LED_GREEN1_Pin, GPIO_PIN_SET);
47        HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
    GPIO_PIN_RESET);
48
49        // NORTH - SOUTH
50        HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
    GPIO_PIN_SET);
51        HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,
    LED_YELLOW2_Pin, GPIO_PIN_RESET);
52
53        if(green_count == 0){
54          led_status = 2;
55          green_count = 3;
56        }
57        break;
58      default:
59        break;
60    }
61    HAL_Delay(1000);
62      /* USER CODE END WHILE */
63
64      /* USER CODE BEGIN 3 */
```

```
65  }
```

4 Exercise 4

Add **only one 7 led segment** to the schematic in Exercise 3. This component can be found in Proteus by the keyword **7SEG-COM-ANODE**. For this device, the common pin should be connected to the power supply and other pins are supposed to connected to PB0 to PB6. Therefore, to turn-on a segment in this 7SEG, the STM32 pin should be in logic 0 (0V).

Implement a function named **display7SEG(int num)**. The input for this function is from 0 to 9 and the outputs are listed as following:



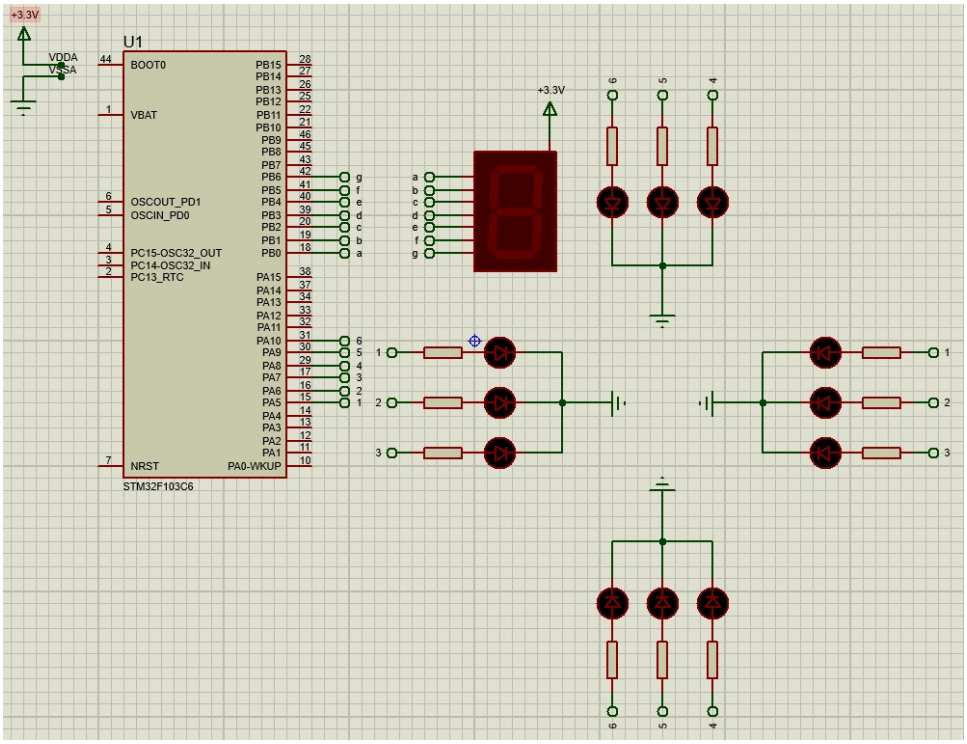
Hình 1.2: Display a number on 7 segment LED

This function is invoked in the while loop for testing as following:

```
int counter = 0;
while (1){
    if(counter >= 10) counter = 0;
    display7SEG(counter++);
    HAL_Delay(1000);

}
```

Program 1.1: An example for your source code

Report 1: Present the schematic.



Hình 1.3: https://github.com/yshic/microcontroller/blob/main/IMGS/ex4schematic.jpg

Report 2: Present the source code for display7SEG function.

```
void display7SEG(int num){
  // 0/RESET = ON, 1/SET = OFF
  switch(num){
  case 0:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_SET);
```

```
    break;
  case 1:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_SET);
    break;
  case 2:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  case 3:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  case 4:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  case 5:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  case 6:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  case 7:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  case 8:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  case 9:
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_RESET);
    break;
  default: //OFF
    HAL_GPIO_WritePin(GPIOB, a_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, b_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, c_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, d_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, e_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, f_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOB, g_Pin, GPIO_PIN_SET);
    break;
  }
}
```

5 Exercise 5

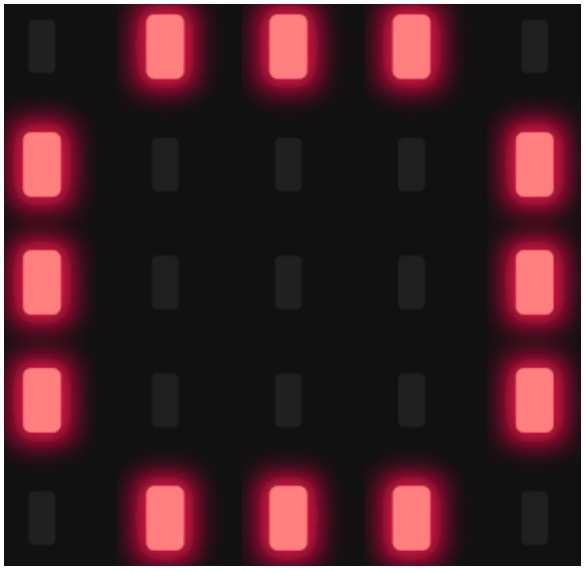
Integrate the 7SEG-LED to the 4 way traffic light. In this case, the 7SEG-LED is used to display countdown value.
In this exercise, only source code is required to present. The function display7SEG in previous exercise can be re-used.

```
int led_status = 1; /*1 = red, 2 = yellow, 3 = green*/
int red_count = 5;
int yellow_count = 2;
int green_count = 3;
while (1)
{
switch(led_status) {
  case 1: // RED -> GREEN
    display7SEG(red_count);
    red_count--;
    // WEST - EAST
    HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port, LED_YELLOW1_Pin, GPIO_PIN_RESET);

    if(red_count > 1){ //GREEN
      // NORTH - SOUTH
      HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin, GPIO_PIN_SET);
      HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin, GPIO_PIN_RESET);
    }
    else if(red_count == 1){ //YELLOW
      // NORTH - SOUTH
      HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port, LED_YELLOW2_Pin, GPIO_PIN_SET);
      HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin, GPIO_PIN_RESET);
    }
    else{
      led_status = 3;
      red_count = 5;

    }
    break;
  case 2: // YELLOW -> RED
    // WEST - EAST
    display7SEG(yellow_count);
    yellow_count--;

```

```
36        HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,
    LED_YELLOW1_Pin, GPIO_PIN_SET);
37        HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port,
    LED_GREEN1_Pin, GPIO_PIN_RESET);
38        // NORTH - SOUTH
39        HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
    GPIO_PIN_SET);
40
41        if(yellow_count == 0){
42          led_status = 1;
43          yellow_count = 2;
44          break;
45        }
46        break;
47      case 3: // GREEN -> YELLOW
48        display7SEG(green_count);
49        green_count--;
50
51        // WEST - EAST
52        HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port,
    LED_GREEN1_Pin, GPIO_PIN_SET);
53        HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
    GPIO_PIN_RESET);
54
55        // NORTH - SOUTH
56        HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
    GPIO_PIN_SET);
57        HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,
    LED_YELLOW2_Pin, GPIO_PIN_RESET);
58
59        if(green_count == 0){
60          led_status = 2;
61          green_count = 3;
62          break;
63        }
64        break;
65      default:
66        break;
67    }
68    HAL_Delay(1000);
69      /* USER CODE END WHILE */
70
71      /* USER CODE BEGIN 3 */
72    }
```

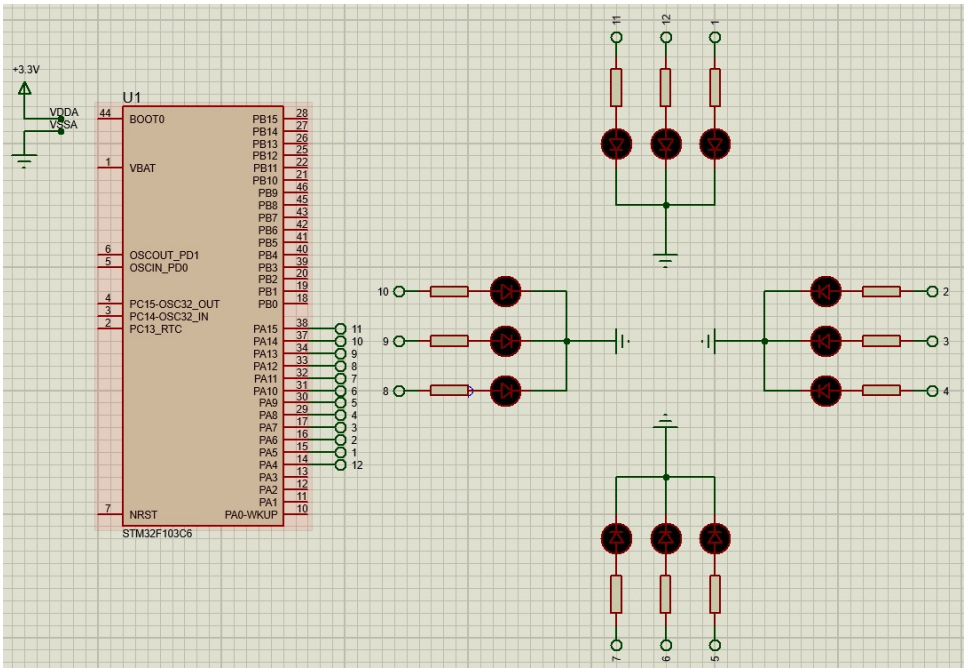
6 Exercise 6

In this exercise, a new Proteus schematic is designed to simulate an analog clock, with 12 different number. The connections for 12 LEDs are supposed from PA4 to PA15 of the STM32. The arrangement of 12 LEDs is depicted as follows.



Hình 1.4: 12 LEDs for an analog clock

Report 1: Present the schematic.



Report 2: Implement a simple program to test the connection of every single LED. This testing program should turn every LED in a sequence.

```
1   int count = 0;
```

```
  while (1)
  {
  switch(count){
  case 0:
    HAL_GPIO_WritePin(GPIOA, CLK_12_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_11_Pin, GPIO_PIN_RESET);
    break;
  case 1:
    HAL_GPIO_WritePin(GPIOA, CLK_1_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_12_Pin, GPIO_PIN_RESET);
    break;
  case 2:
    HAL_GPIO_WritePin(GPIOA, CLK_2_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_1_Pin, GPIO_PIN_RESET);
  break;
  case 3:
    HAL_GPIO_WritePin(GPIOA, CLK_3_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_2_Pin, GPIO_PIN_RESET);
    break;
  case 4:
    HAL_GPIO_WritePin(GPIOA, CLK_4_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_3_Pin, GPIO_PIN_RESET);
    break;
  case 5:
    HAL_GPIO_WritePin(GPIOA, CLK_5_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_4_Pin, GPIO_PIN_RESET);
    break;
  case 6:
    HAL_GPIO_WritePin(GPIOA, CLK_6_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_5_Pin, GPIO_PIN_RESET);
    break;
  case 7:
    HAL_GPIO_WritePin(GPIOA, CLK_7_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_6_Pin, GPIO_PIN_RESET);
    break;
  case 8:
    HAL_GPIO_WritePin(GPIOA, CLK_8_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_7_Pin, GPIO_PIN_RESET);
    break;
  case 9:
    HAL_GPIO_WritePin(GPIOA, CLK_9_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_8_Pin, GPIO_PIN_RESET);
    break;
  case 10:
    HAL_GPIO_WritePin(GPIOA, CLK_10_Pin, GPIO_PIN_SET);
    HAL_GPIO_WritePin(GPIOA, CLK_9_Pin, GPIO_PIN_RESET);
    break;
  case 11:
    HAL_GPIO_WritePin(GPIOA, CLK_11_Pin, GPIO_PIN_SET);
```

```
    HAL_GPIO_WritePin(GPIOA, CLK_10_Pin, GPIO_PIN_RESET);
    break;
  default:
    break;
  }
  count++;
  if(count == 12) count = 0;
  HAL_Delay(1000);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
```

7 Exercise 7

Implement a function named **clearAllClock()** to turn off all 12 LEDs. Present the source code of this function.

```
void clearAllClock(){
  HAL_GPIO_WritePin(GPIOA, CLK_12_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_1_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_2_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_3_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_4_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_5_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_6_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_7_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_8_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_9_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_10_Pin, GPIO_PIN_RESET);
  HAL_GPIO_WritePin(GPIOA, CLK_11_Pin, GPIO_PIN_RESET);
}
```

7.1 Exercise 8

Implement a function named **setNumberOnClock(int num)**. The input for this function is from **0 to 11** and an appropriate LED is turn on. Present the source code of this function.

```
void setNumberOnClock(int num){
  switch(num){
  case 0:
    HAL_GPIO_WritePin(GPIOA, CLK_12_Pin, GPIO_PIN_SET);
    break;
  case 1:
    HAL_GPIO_WritePin(GPIOA, CLK_1_Pin, GPIO_PIN_SET);
    break;
```

```
  case 2:
    HAL_GPIO_WritePin(GPIOA, CLK_2_Pin, GPIO_PIN_SET);
    break;
  case 3:
    HAL_GPIO_WritePin(GPIOA, CLK_3_Pin, GPIO_PIN_SET);
    break;
  case 4:
    HAL_GPIO_WritePin(GPIOA, CLK_4_Pin, GPIO_PIN_SET);
    break;
  case 5:
    HAL_GPIO_WritePin(GPIOA, CLK_5_Pin, GPIO_PIN_SET);
    break;
  case 6:
    HAL_GPIO_WritePin(GPIOA, CLK_6_Pin, GPIO_PIN_SET);
    break;
  case 7:
    HAL_GPIO_WritePin(GPIOA, CLK_7_Pin, GPIO_PIN_SET);
    break;
  case 8:
    HAL_GPIO_WritePin(GPIOA, CLK_8_Pin, GPIO_PIN_SET);
    break;
  case 9:
    HAL_GPIO_WritePin(GPIOA, CLK_9_Pin, GPIO_PIN_SET);
    break;
  case 10:
    HAL_GPIO_WritePin(GPIOA, CLK_10_Pin, GPIO_PIN_SET);
    break;
  case 11:
    HAL_GPIO_WritePin(GPIOA, CLK_11_Pin, GPIO_PIN_SET);
    break;
  default:
    break;
  }
}
```

8 Exercise 9

Implement a function named **clearNumberOnClock(int num)**. The input for this function is from **0 to 11** and an appropriate LED is turn off.

```
void clearNumberOnClock(int num){
  switch(num){
  case 0:
    HAL_GPIO_WritePin(GPIOA, CLK_12_Pin, GPIO_PIN_RESET);
    break;
  case 1:
    HAL_GPIO_WritePin(GPIOA, CLK_1_Pin, GPIO_PIN_RESET);
    break;
```

```
  case 2:
    HAL_GPIO_WritePin(GPIOA, CLK_2_Pin, GPIO_PIN_RESET);
    break;
  case 3:
    HAL_GPIO_WritePin(GPIOA, CLK_3_Pin, GPIO_PIN_RESET);
    break;
  case 4:
    HAL_GPIO_WritePin(GPIOA, CLK_4_Pin, GPIO_PIN_RESET);
    break;
  case 5:
    HAL_GPIO_WritePin(GPIOA, CLK_5_Pin, GPIO_PIN_RESET);
    break;
  case 6:
    HAL_GPIO_WritePin(GPIOA, CLK_6_Pin, GPIO_PIN_RESET);
    break;
  case 7:
    HAL_GPIO_WritePin(GPIOA, CLK_7_Pin, GPIO_PIN_RESET);
    break;
  case 8:
    HAL_GPIO_WritePin(GPIOA, CLK_8_Pin, GPIO_PIN_RESET);
    break;
  case 9:
    HAL_GPIO_WritePin(GPIOA, CLK_9_Pin, GPIO_PIN_RESET);
    break;
  case 10:
    HAL_GPIO_WritePin(GPIOA, CLK_10_Pin, GPIO_PIN_RESET);
    break;
  case 11:
    HAL_GPIO_WritePin(GPIOA, CLK_11_Pin, GPIO_PIN_RESET);
    break;
  default:
    break;
  }
}
```

9 Exercise 10

Integrate the whole system and use 12 LEDs to display a clock. At a given time, there are only 3 LEDs are turn on for hour, minute and second information.

```
  int second = 0;
  int minute = 0;
  int hour = 0;
  while (1)
  {
  clearAllClock();
  if(second == 60) {
    minute++;
```

```
    second = 0;
  }
  if(minute == 60){
    hour++;
    minute = 0;
    if(hour == 13) hour = 1;
  }
  setNumberOnClock(second / 5);
  setNumberOnClock(minute / 5);
  setNumberOnClock(hour);
  second++;
  HAL_Delay(1000);
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
```