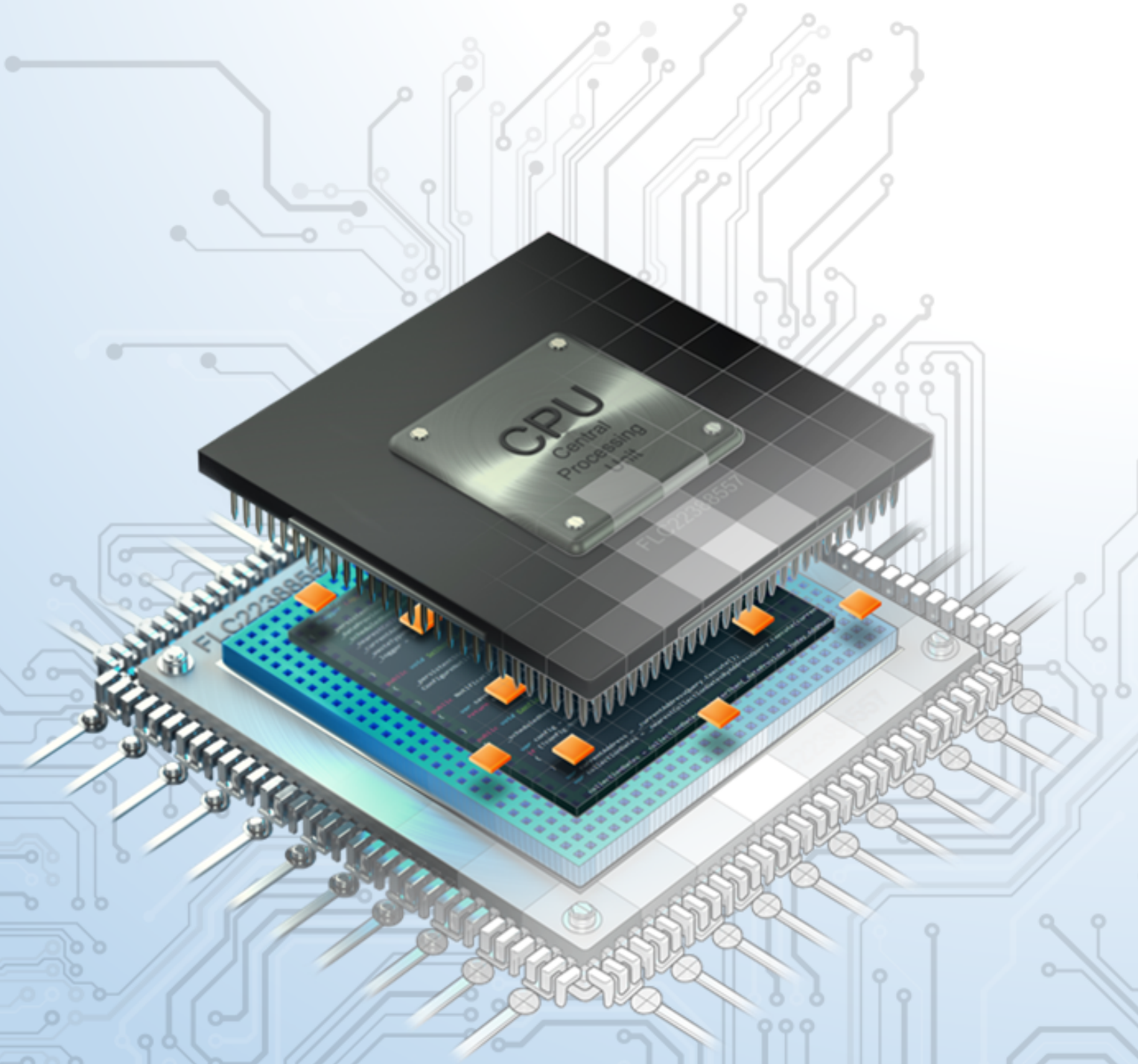
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller

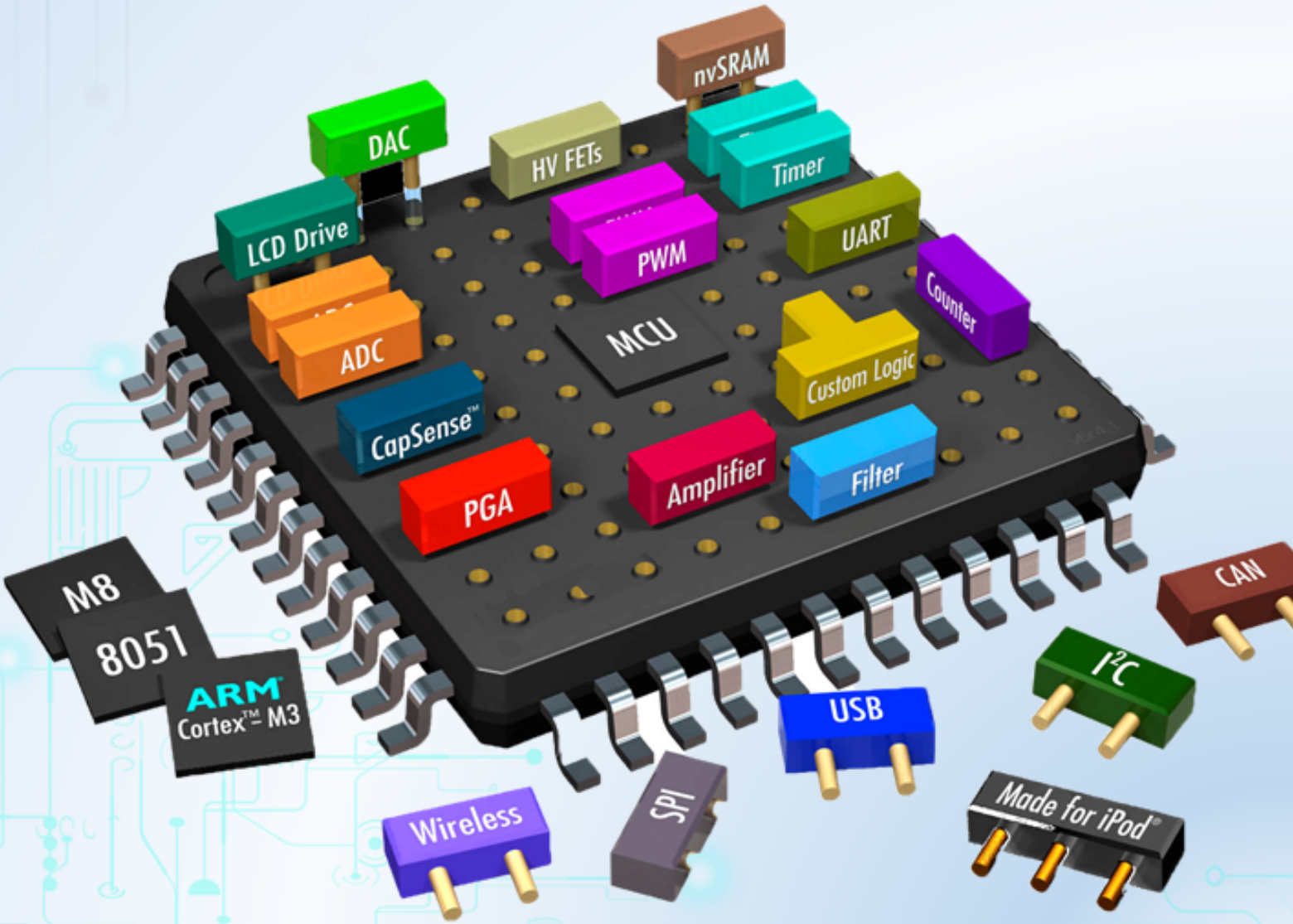


Dr. Le Trong Nhan

Mục lục

CHƯƠNG 1

LED Animations

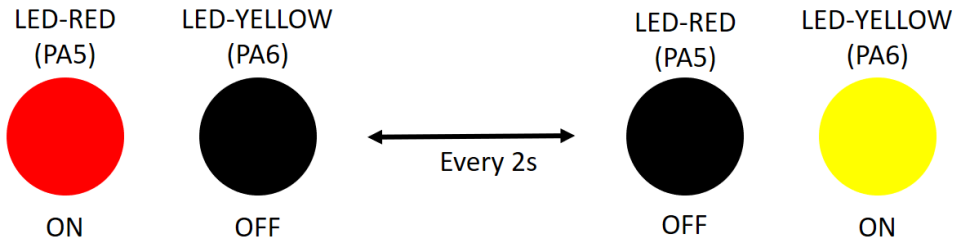


1 Exercise and Report

1.1 Exercise 1

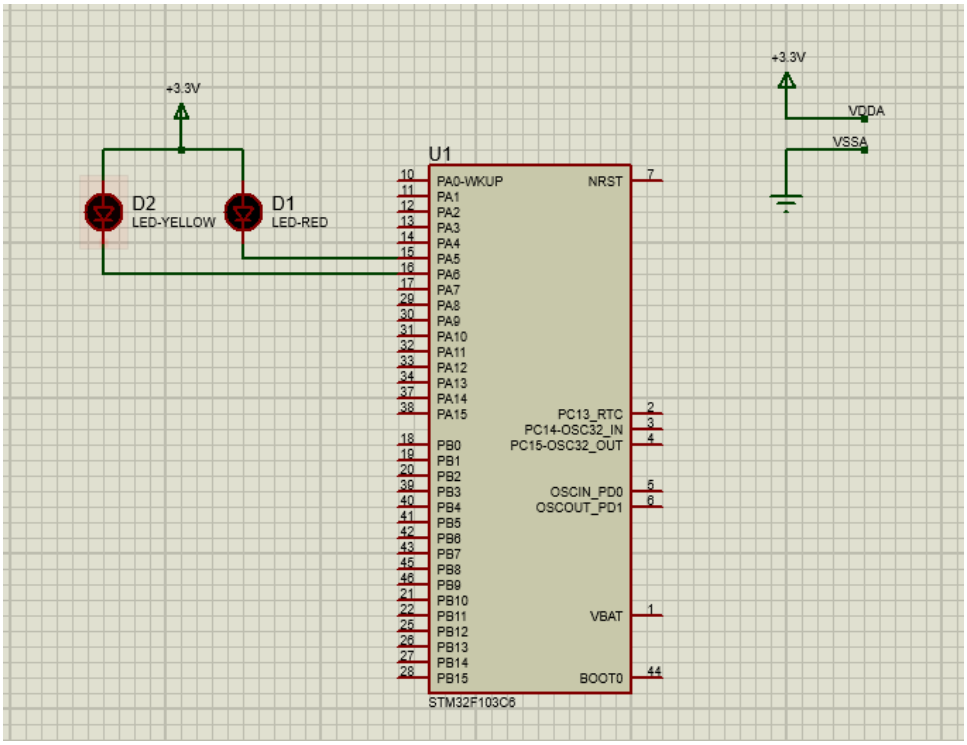
From the simulation on Proteus, one more LED is connected to pin **PA6** of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is **LED-YELLOW**, which can be found from the device list.

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: Schematic from Proteus simulation.



Hình 1.2: https://github.com/DinhHoangDung/MCU_Lab1/blob/main/Simulation/Ex1/Ex1.png

Report 2: Present the source code in the infinite loop while of your project.

```
int counter = 0;
while (1)
{
    if (counter < 2)
    {
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, RESET
   );
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
    SET);
  counter++;
    }
    else
    {
        HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
   SET);
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
    RESET);
  counter++;
        if (counter == 4) counter = 0;
    }
    HAL_Delay(1000);
}
```

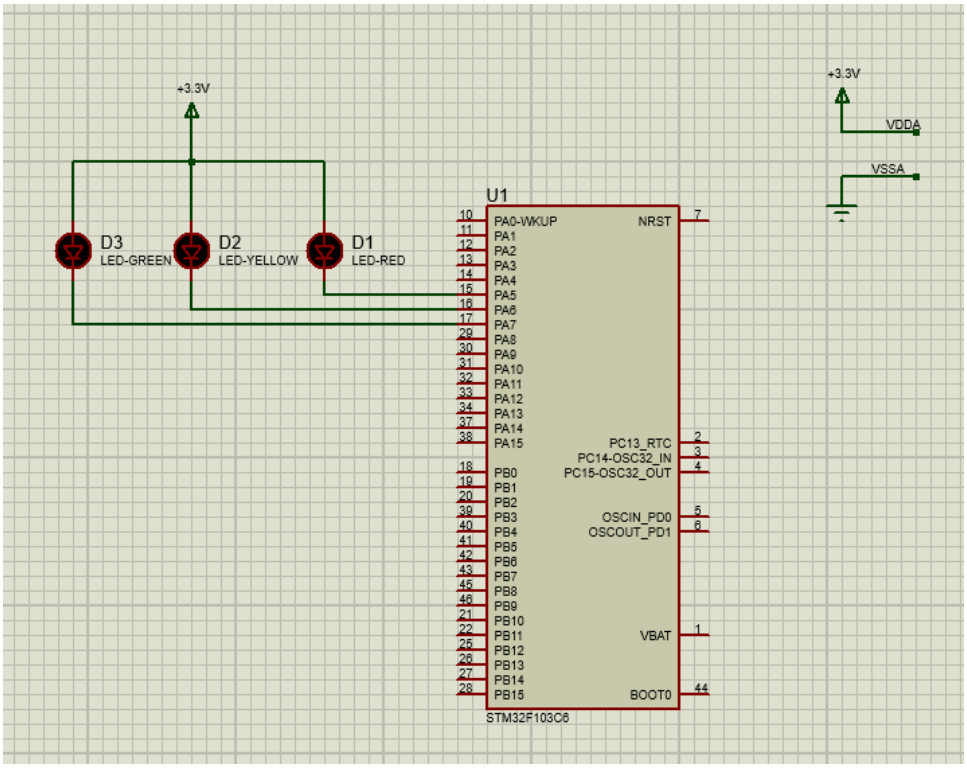
Program 1.1: Source code Ex1

1.2 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: The schematic.



Hình 1.3: https://github.com/DinhHoangDung/MCU_Lab1/blob/main/Simulation/Ex2/Ex2.png

Report 2: Present the source code in while.

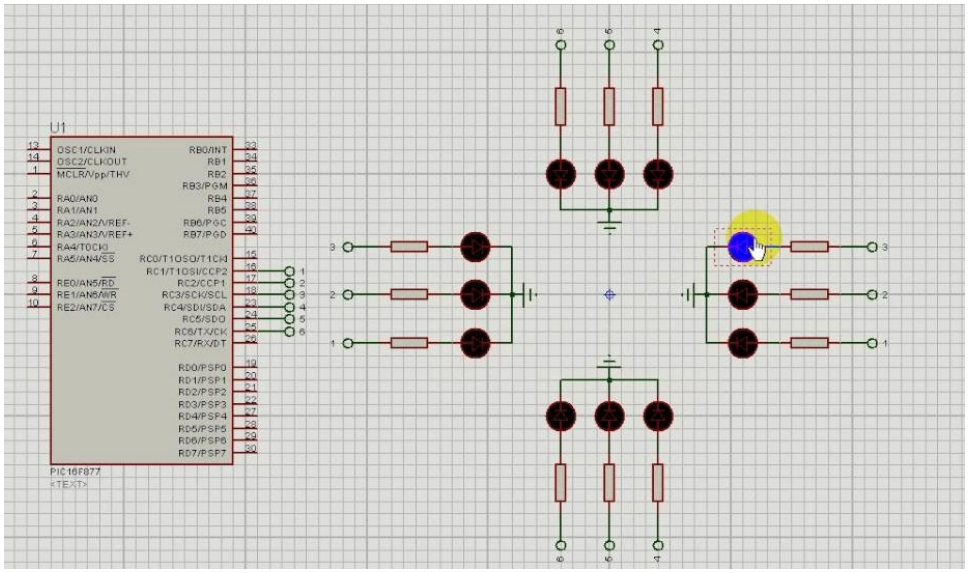
```
int count = 0;
while (1)
{
  if (count == 10) count = 0;
  if (count == 0)
  {
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, RESET);
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, SET);
    HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, SET);
  }
  if (count == 5)
  {
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
    HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, SET);
    HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, RESET);
  }
  if (count == 8)
  {
    HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
```

```
20      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
        RESET);
21      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin,
     SET);
22    }
23    count++;
24    HAL_Delay(1000);
25    }
```

Program 1.2: Source code Ex1

1.3 Exercise 3

Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light. A reference design can be found in the figure bellow.

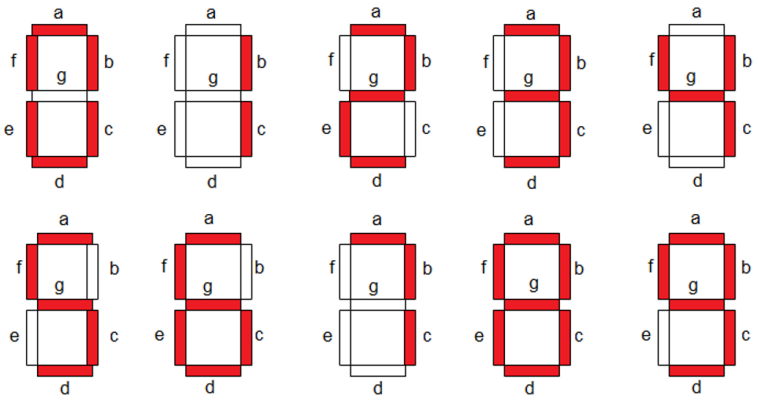


Hình 1.4: *Reference design for a 4 way traffic light*

1.4 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 1.5: Display a number on 7 segment LED

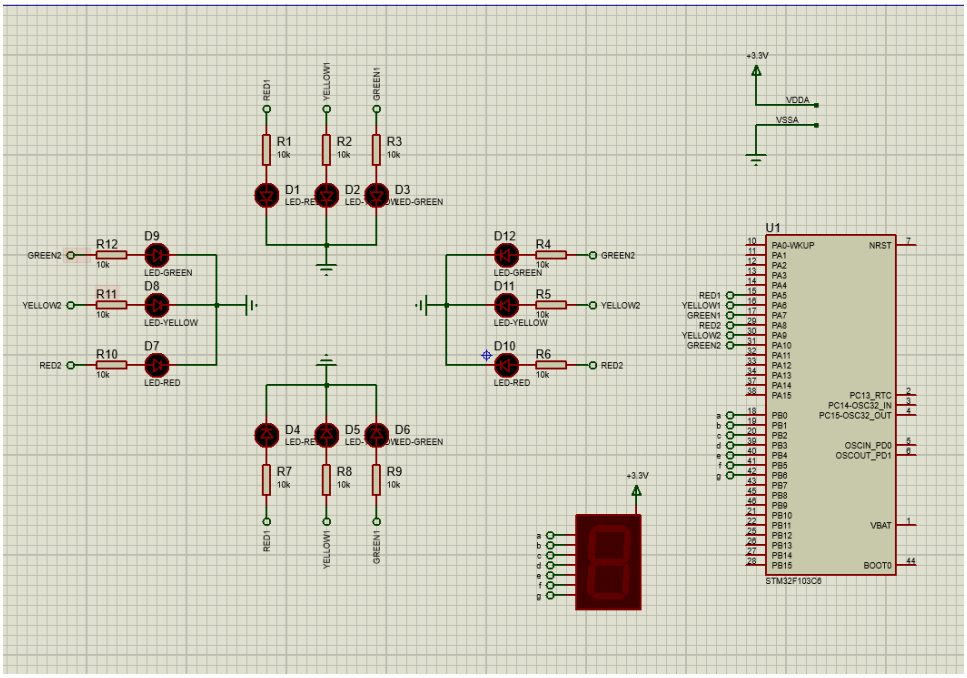
This function is invoked in the while loop for testing as following:

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

}
```

Program 1.3: An example for your source code

Report 1: Present the schematic.



Hình 1.6: https://github.com/DinhHoangDung/MCU_Lab1/blob/main/Simulation/Ex5/Ex5.png

Report 2: Present the function code.

```
void display7SEG(int num)
{
        switch(num)
        {
              case 0:
              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, RESET);
              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, RESET);
              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, RESET);
              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, SET);
              break;
            case 1:
              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, SET);
              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, SET);
              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, SET);
              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, SET);
              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, SET);
              break;
            case 2:
              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, SET);
              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, RESET);
              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, RESET);
              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, SET);
              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, RESET);
              break;
            case 3:
              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, RESET);
              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, SET);
              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, SET);
              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, RESET);
              break;
            case 4:
                HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, SET);
              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, SET);
              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, SET);
              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, RESET);
```

```
48              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, RESET);
49              break;
50          case 5:
51              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
52              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, SET);
53              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
54              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, RESET);
55              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, SET);
56              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, RESET);
57              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, RESET);
58              break;
59          case 6:
60              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
61              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, SET);
62              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
63              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, RESET);
64              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, RESET);
65              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, RESET);
66              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, RESET);
67              break;
68          case 7:
69              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
70              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
71              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
72              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, SET);
73              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, SET);
74              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, SET);
75              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, SET);
76              break;
77          case 8:
78              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
79              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
80              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
81              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, RESET);
82              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, RESET);
83              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, RESET);
84              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, RESET);
85              break;
86          case 9:
87              HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, RESET);
88              HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, RESET);
89              HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, RESET);
90              HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, RESET);
91              HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, SET);
92              HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, RESET);
93              HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, RESET);
94              break;
95
96      }
```

```
}
```

Program 1.4: Function Implementation

1.5 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.
Report: The source code of 4 way traffic light.

```
enum TrafficLightState
{
     RED1_GREEN2,
     RED1_YELLOW2,
     GREEN1_RED2,
     YELLOW1_RED2,
};
    //Initialize the current state of the traffic light
int counter = 3;
int light = 5;
enum TrafficLightState currentState = RED1_GREEN2;
//Set all LEDs off
HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin, RESET);
HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port, LED_YELLOW1_Pin,
   RESET);
HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port, LED_GREEN1_Pin,
   RESET);
HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin, RESET);
HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port, LED_YELLOW2_Pin,
   RESET);
HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port, LED_GREEN2_Pin,
   RESET);

while (1)
{
    display7SEG(light);
  switch (currentState)
  {
    case RED1_GREEN2:
        HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
    SET);
        HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port,
   LED_GREEN2_Pin, SET);
        HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,
   LED_YELLOW1_Pin, RESET);
```

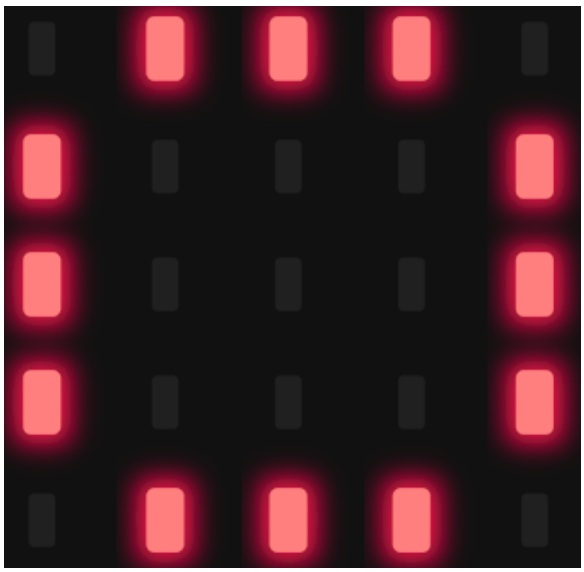
```
29          HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
    RESET);
30          counter--;
31          light--;
32          if (counter <= 0)
33          {
34            currentState = RED1_YELLOW2;
35            counter = 2;
36
37          }
38          break;
39       case RED1_YELLOW2:
40          HAL_GPIO_WritePin(LED_GREEN2_GPIO_Port,
    LED_GREEN2_Pin, RESET);
41          HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,
    LED_YELLOW2_Pin, SET);
42          counter--;
43          light--;
44          if (counter <= 0)
45          {
46            currentState = GREEN1_RED2;
47            counter = 3;
48            light = 3;
49          }
50          break;
51       case GREEN1_RED2:
52          HAL_GPIO_WritePin(LED_RED1_GPIO_Port, LED_RED1_Pin,
    RESET);
53          HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port,
    LED_GREEN1_Pin, SET);
54          HAL_GPIO_WritePin(LED_YELLOW2_GPIO_Port,
    LED_YELLOW2_Pin, RESET);
55          HAL_GPIO_WritePin(LED_RED2_GPIO_Port, LED_RED2_Pin,
    SET);
56          counter--;
57          light--;
58          if (counter <= 0)
59          {
60            currentState = YELLOW1_RED2;
61            counter = 2;
62            light = 2;
63          }
64          break;
65       case YELLOW1_RED2:
66          HAL_GPIO_WritePin(LED_GREEN1_GPIO_Port,
    LED_GREEN1_Pin, RESET);
67          HAL_GPIO_WritePin(LED_YELLOW1_GPIO_Port,
    LED_YELLOW1_Pin, SET);
68          counter--;
```

```
69          light--;
70          if (counter <= 0)
71          {
72            currentState = RED1_GREEN2;
73            counter = 3;
74            light = 5;
75          }
76          break;
77      }
78      HAL_Delay(1000);
79
80  }
```

Program 1.5: Source code

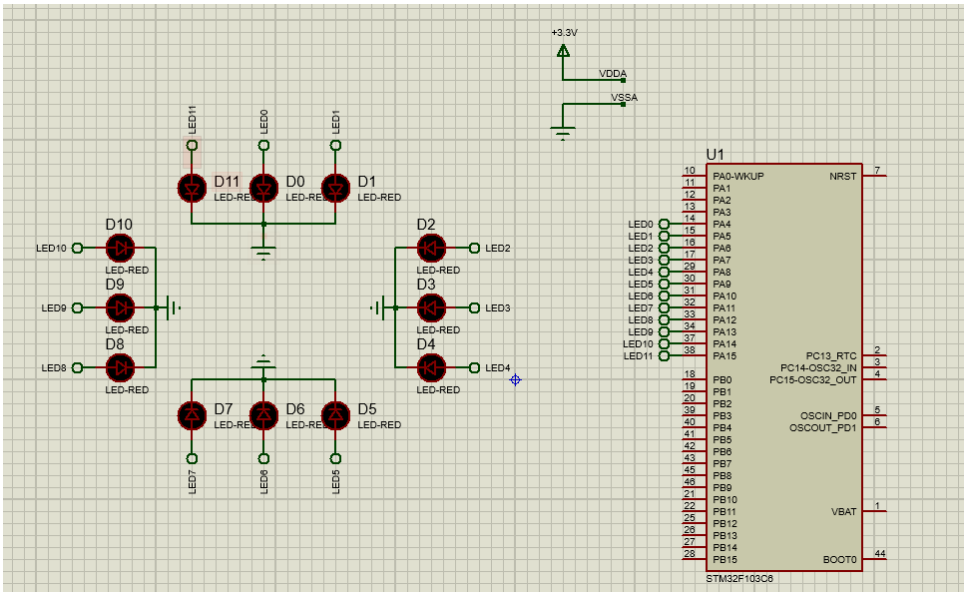
1.6 Exercise 6

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 1.7: 12 LEDs for an analog clock

Report 1: Present the schematic.



Hình 1.8: https://github.com/DinhHoangDung/MCU_Lab1/blob/main/Simulation/Ex6/Ex6.png

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

```
void displayLED (int counter, int *state)
{
  if (*state == 0)
  {
    *state = 1;
    HAL_GPIO_TogglePin(LED0_GPIO_Port, LED0_Pin);
  }
  else if (counter == 0)
  {
    HAL_GPIO_TogglePin(LED0_GPIO_Port, LED0_Pin);
    HAL_GPIO_TogglePin(LED11_GPIO_Port, LED11_Pin);
  }
  else if (counter == 1)
  {
    HAL_GPIO_TogglePin(LED0_GPIO_Port, LED0_Pin);
    HAL_GPIO_TogglePin(LED1_GPIO_Port, LED1_Pin);
  }
  else if (counter == 2)
  {
    HAL_GPIO_TogglePin(LED1_GPIO_Port, LED1_Pin);
    HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
  }
  else if (counter == 3)
  {
    HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
    HAL_GPIO_TogglePin(LED3_GPIO_Port, LED3_Pin);
  }
  else if (counter == 4)
```

```
  {
    HAL_GPIO_TogglePin(LED3_GPIO_Port, LED3_Pin);
    HAL_GPIO_TogglePin(LED4_GPIO_Port, LED4_Pin);
  }
  else if (counter == 5)
  {
    HAL_GPIO_TogglePin(LED4_GPIO_Port, LED4_Pin);
    HAL_GPIO_TogglePin(LED5_GPIO_Port, LED5_Pin);
  }
  else if (counter == 6)
  {
    HAL_GPIO_TogglePin(LED5_GPIO_Port, LED5_Pin);
    HAL_GPIO_TogglePin(LED6_GPIO_Port, LED6_Pin);
  }
  else if (counter == 7)
  {
    HAL_GPIO_TogglePin(LED6_GPIO_Port, LED6_Pin);
    HAL_GPIO_TogglePin(LED7_GPIO_Port, LED7_Pin);
  }
  else if (counter == 8)
  {
    HAL_GPIO_TogglePin(LED7_GPIO_Port, LED7_Pin);
    HAL_GPIO_TogglePin(LED8_GPIO_Port, LED8_Pin);
  }
  else if (counter == 9)
  {
    HAL_GPIO_TogglePin(LED8_GPIO_Port, LED8_Pin);
    HAL_GPIO_TogglePin(LED9_GPIO_Port, LED9_Pin);
  }
  else if (counter == 10)
  {
    HAL_GPIO_TogglePin(LED9_GPIO_Port, LED9_Pin);
    HAL_GPIO_TogglePin(LED10_GPIO_Port, LED10_Pin);
  }
  else if (counter == 11)
  {
    HAL_GPIO_TogglePin(LED10_GPIO_Port, LED10_Pin);
    HAL_GPIO_TogglePin(LED11_GPIO_Port, LED11_Pin);
  }
}
```

Program 1.6: Function to display LED

```
int counter = 0;
int state = 0;
while (1)
{
        if (counter >= 12) counter = 0;
  displayLED(counter++, &state);
  HAL_Delay(1000);
```

```
    /* USER CODE END WHILE */

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

Program 1.7: Code in while() loop

1.7 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, LED0_Pin|LED1_Pin|LED2_Pin|
   LED3_Pin
    |LED4_Pin|LED5_Pin|LED6_Pin|LED7_Pin|LED8_Pin
    |LED9_Pin|LED10_Pin|LED11_Pin, RESET);
}
```

Program 1.8: Function Implementation

1.8 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)
{
  if (num == 0) HAL_GPIO_WritePin(LED0_GPIO_Port, LED0_Pin,
    SET);
  if (num == 1) HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin,
    SET);
  if (num == 2) HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin,
    SET);
  if (num == 3) HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin,
    SET);
  if (num == 4) HAL_GPIO_WritePin(LED4_GPIO_Port, LED4_Pin,
    SET);
  if (num == 5) HAL_GPIO_WritePin(LED5_GPIO_Port, LED5_Pin,
    SET);
  if (num == 6) HAL_GPIO_WritePin(LED6_GPIO_Port, LED6_Pin,
    SET);
  if (num == 7) HAL_GPIO_WritePin(LED7_GPIO_Port, LED7_Pin,
    SET);
  if (num == 8) HAL_GPIO_WritePin(LED8_GPIO_Port, LED8_Pin,
    SET);
  if (num == 9) HAL_GPIO_WritePin(LED9_GPIO_Port, LED9_Pin,
    SET);
```

```
13   if (num == 10) HAL_GPIO_WritePin(LED10_GPIO_Port,
     LED10_Pin, SET);
14   if (num == 11) HAL_GPIO_WritePin(LED11_GPIO_Port,
     LED11_Pin, SET);
15 }
```

Program 1.9: Function Implementation

1.9 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.

```
1  void clearNumberOnClock(int num)
2  {
3    if (num == 0) HAL_GPIO_WritePin(LED0_GPIO_Port, LED0_Pin,
       RESET);
4    if (num == 1) HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin,
       RESET);
5    if (num == 2) HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin,
       RESET);
6    if (num == 3) HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin,
       RESET);
7    if (num == 4) HAL_GPIO_WritePin(LED4_GPIO_Port, LED4_Pin,
       RESET);
8    if (num == 5) HAL_GPIO_WritePin(LED5_GPIO_Port, LED5_Pin,
       RESET);
9    if (num == 6) HAL_GPIO_WritePin(LED6_GPIO_Port, LED6_Pin,
       RESET);
10   if (num == 7) HAL_GPIO_WritePin(LED7_GPIO_Port, LED7_Pin,
       RESET);
11   if (num == 8) HAL_GPIO_WritePin(LED8_GPIO_Port, LED8_Pin,
       RESET);
12   if (num == 9) HAL_GPIO_WritePin(LED9_GPIO_Port, LED9_Pin,
       RESET);
13   if (num == 10) HAL_GPIO_WritePin(LED10_GPIO_Port,
     LED10_Pin, RESET);
14   if (num == 11) HAL_GPIO_WritePin(LED11_GPIO_Port,
     LED11_Pin, RESET);
15 }
```

Program 1.10: Function Implementation

1.10 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.

```
1  int hour = 0;
```

```
int minute = 0;
int second = 0;
while (1)
{
    if (second == 60)
    {
      minute++;
      second = 0;
    }
    if (minute == 60)
    {
      hour++;
      minute = 0;
    }
    if (hour == 12)
    {
      hour = 0;
    }
    clearAllClock();
    setNumberOnClock(second/5);
    setNumberOnClock(minute/5);
    setNumberOnClock(hour);
    second++;
    HAL_Delay(1000);
}
```

Program 1.11: Code in while() loop