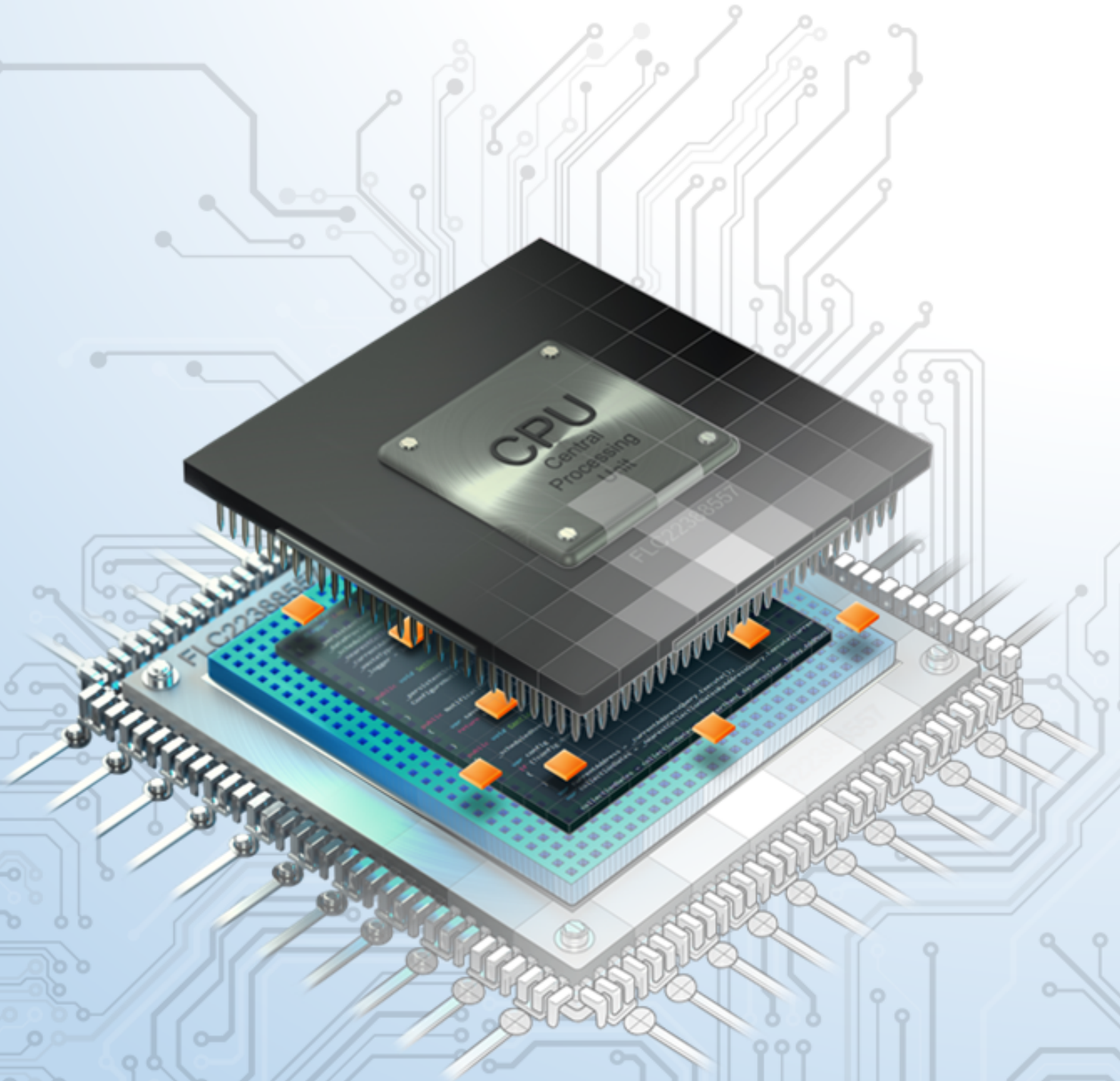




HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller



Dr. Le Trong Nhan

Mục lục

Chapter 1. LED Animations	5
1 Exercise and Report	6
1.1 Exercise 1	6
1.2 Exercise 2	7
1.3 Exercise 3	8
1.4 Exercise 4	9
1.5 Exercise 5	11
1.6 Exercise 6	13
1.7 Exercise 7	14
1.8 Exercise 8	14
1.9 Exercise 9	15
1.10 Exercise 10	15

CHƯƠNG 1

LED Animations



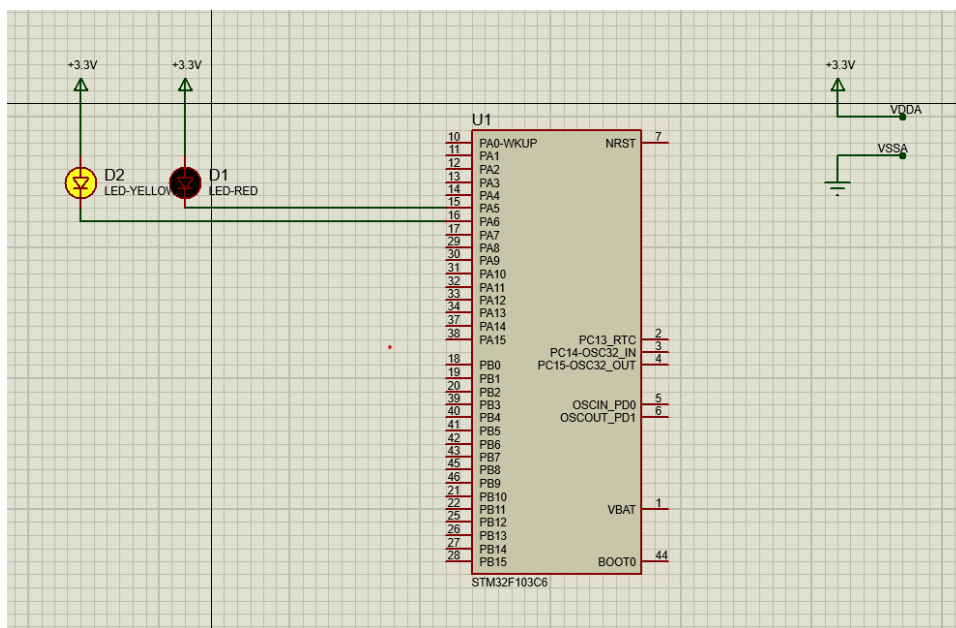
1 Exercise and Report

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.

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).//



Hình 1.1: https://github.com/duyhoangbk23/Microcontroller_Lab/blob//Proteus_EX1/EX1.pdsprj

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  while (1)
2  {
3      /* USER CODE END WHILE */
4
5      /* USER CODE BEGIN 3 */
6      HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
7      GPIO_PIN_RESET);
8      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
9      LED_YELLOW_Pin, GPIO_PIN_SET);
10     HAL_Delay(2000);
```

```

10 // LED-RED OFF, LED-YELLOW ON
11 HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
GPIO_PIN_SET);
12 HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
LED_YELLOW_Pin, GPIO_PIN_RESET);
13 HAL_Delay(2000);
14 }

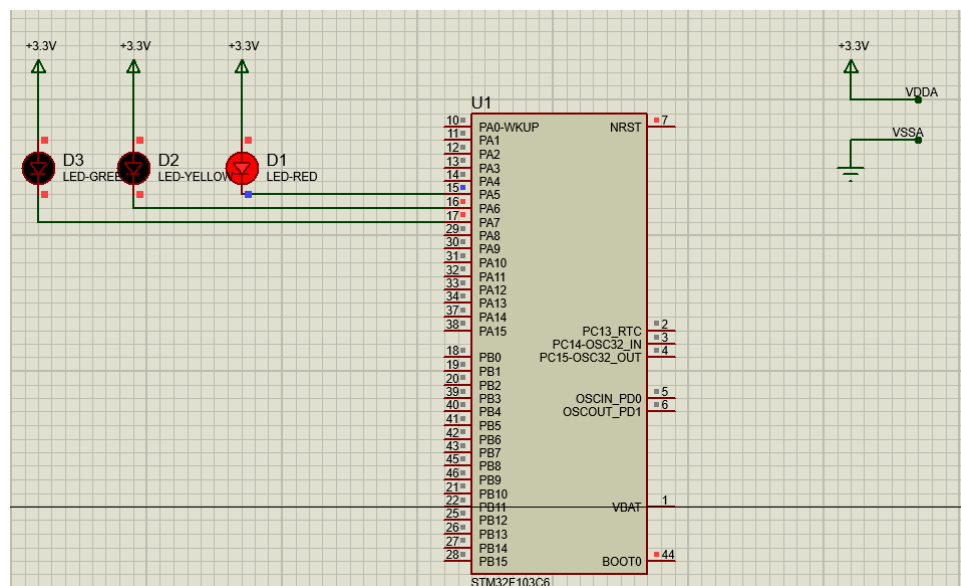
```

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: Present the schematic.



Report 2: Present the source code in while.

```

1 while (1)
2 {
3     /* USER CODE END WHILE */
4
5     /* USER CODE BEGIN 3 */
6     // LED-RED ON, LED-YELLOW OFF, LED-GREEN OFF
7     HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin,
GPIO_PIN_RESET);
8     HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port,
LED_YELLOW_Pin, GPIO_PIN_SET);

```



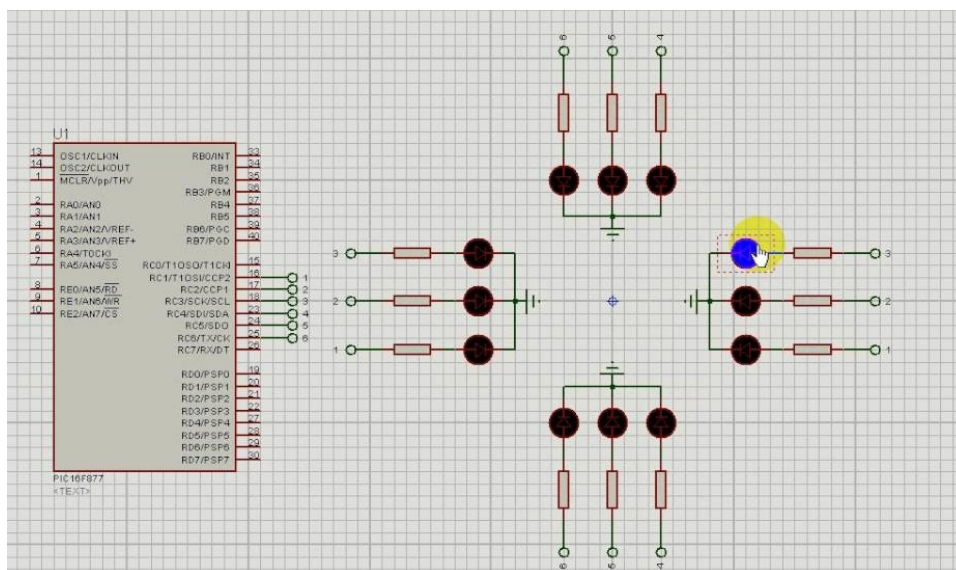
```

9      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port ,
LED_GREEN_Pin , GPIO_PIN_SET);
10      HAL_Delay(5000);
11
12      // LED-RED OFF, LED-YELLOW ON, LED-GREEN OFF
13      HAL_GPIO_WritePin(LED_RED_GPIO_Port , LED_RED_Pin ,
GPIO_PIN_SET);
14      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port ,
LED_YELLOW_Pin , GPIO_PIN_RESET);
15      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port ,
LED_GREEN_Pin , GPIO_PIN_SET);
16      HAL_Delay(2000);
17
18      // LED-RED OFF, LED-YELLOW OFF, LED-GREEN ON
19      HAL_GPIO_WritePin(LED_RED_GPIO_Port , LED_RED_Pin ,
GPIO_PIN_SET);
20      HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port ,
LED_YELLOW_Pin , GPIO_PIN_SET);
21      HAL_GPIO_WritePin(LED_GREEN_GPIO_Port ,
LED_GREEN_Pin , GPIO_PIN_RESET);
22      HAL_Delay(3000);
23  }

```

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.

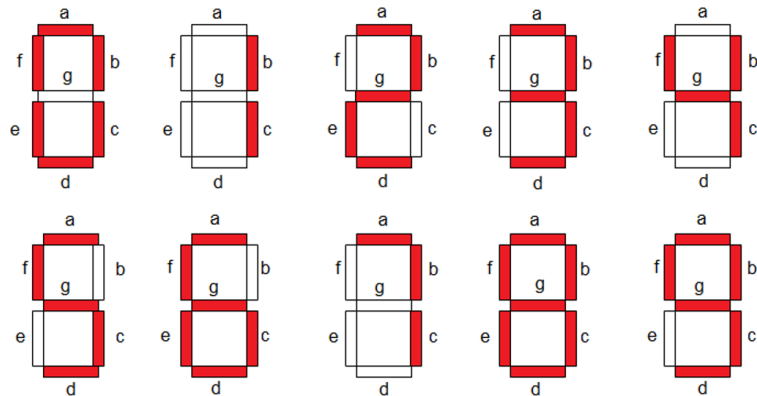


Hình 1.2: Reference design for a 4 way traffic light

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 be connected to PB0 to PB6. Therefore, to turn-on a segment in this 7SEG, the STM32 pin should be in logic 0 (0V).

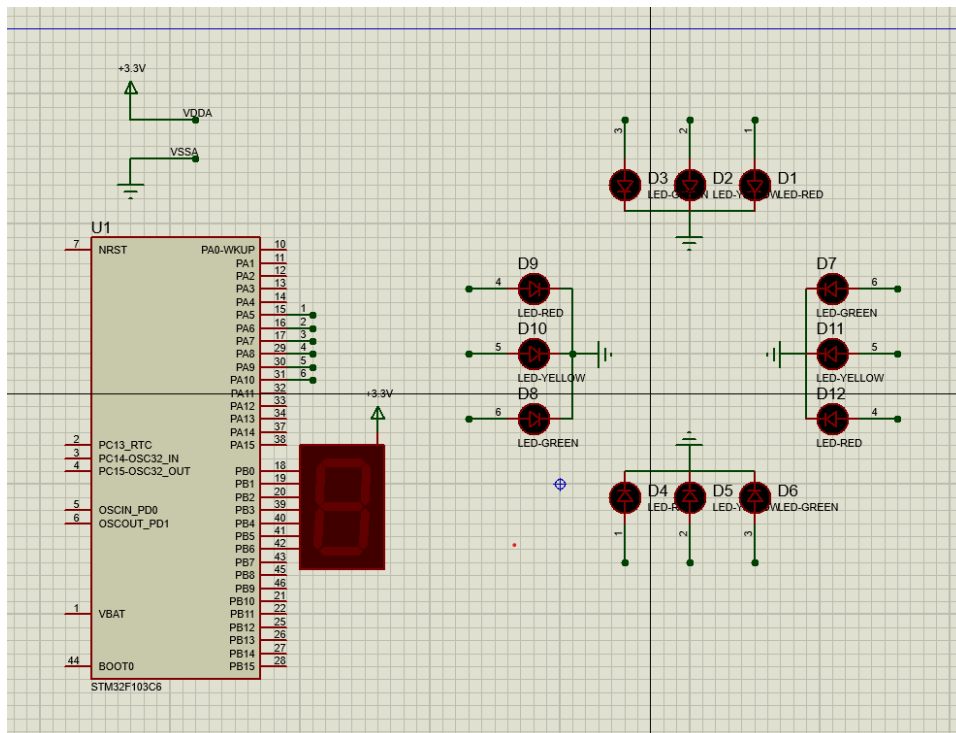
Implement a function named **display7SEG(int num)**. The input for this function is from 0 to 9 and the outputs are listed as following:



Hình 1.3: Display a number on 7 segment LED

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

Report 1: Present the schematic.



```
1 /* Private function prototypes
   -----*/
2 void SystemClock_Config(void);
3 static void MX_GPIO_Init(void);
4 /* USER CODE BEGIN PFP */
5 static const uint8_t segCode[10] = {
6     0b1000000, // 0
7     0b1111001, // 1
8     0b0100100, // 2
9     0b0110000, // 3
10    0b0011001, // 4
11    0b0010010, // 5
12    0b0000010, // 6
13    0b1111000, // 7
14    0b0000000, // 8
```

```

15     0b0010000 // 9
16 };
17
18 void display7SEG(int num) {
19     if (num < 0 || num > 9) num = 0;
20
21     for (int i = 0; i < 7; i++) {
22         HAL_GPIO_WritePin(GPIOB, (1 << i),
23             (segCode[num] & (1 << i)) ? GPIO_PIN_SET :
24             GPIO_PIN_RESET);
25     }
26 }

```

Program 1.2: 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.

```

1  /* USER CODE BEGIN 2 */
2  static const uint8_t segCode[10] = {
3      0b1000000, // 0
4      0b1111001, // 1
5      0b0100100, // 2
6      0b0110000, // 3
7      0b0011001, // 4
8      0b0010010, // 5
9      0b0000010, // 6
10     0b1111000, // 7
11     0b0000000, // 8
12     0b0010000 // 9
13 };
14
15 void display7SEG(int num) {
16     if (num < 0 || num > 9) num = 0;
17
18     for (int i = 0; i < 7; i++) {
19         HAL_GPIO_WritePin(GPIOB, (1 << i),
20             (segCode[num] & (1 << i)) ? GPIO_PIN_SET :
21             GPIO_PIN_RESET);
22     }
23 }
24
25 /* USER CODE END 2 */
26
27 /* Infinite loop */

```

```

27  /* USER CODE BEGIN WHILE */
28  int counter = 9;
29  int phase = 0; // 0: Road1 RED, Road2 GREEN->YELLOW
30                // 1: Road2 RED, Road1 GREEN->YELLOW
31
32  while (1)
33  {
34      display7SEG(counter);
35
36      // Reset all LEDs first
37      HAL_GPIO_WritePin(GPIOA, RED1_Pin|YELLOW1_Pin|
38      GREEN1_Pin|
39                                RED2_Pin|YELLOW2_Pin|
40      GREEN2_Pin, GPIO_PIN_RESET);
41
42      if (phase == 0) {
43          // Road1 always RED
44          HAL_GPIO_WritePin(GPIOA, RED1_Pin, GPIO_PIN_SET);
45
46          if (counter > 4) HAL_GPIO_WritePin(GPIOA,
47      GREEN2_Pin, GPIO_PIN_SET); // Green
48          else if (counter > 1) HAL_GPIO_WritePin(GPIOA,
49      YELLOW2_Pin, GPIO_PIN_SET); // Yellow
50          else HAL_GPIO_WritePin(GPIOA, RED2_Pin,
51      GPIO_PIN_SET); // Switch RED
52      } else {
53          // Road2 always RED
54          HAL_GPIO_WritePin(GPIOA, RED2_Pin, GPIO_PIN_SET);
55
56          if (counter > 4) HAL_GPIO_WritePin(GPIOA,
57      GREEN1_Pin, GPIO_PIN_SET); // Green
58          else if (counter > 1) HAL_GPIO_WritePin(GPIOA,
59      YELLOW1_Pin, GPIO_PIN_SET); // Yellow
60          else HAL_GPIO_WritePin(GPIOA, RED1_Pin,
61      GPIO_PIN_SET); // Switch RED
62      }
63
64      HAL_Delay(1000);
65
66      // Countdown
67      if (counter <= 0) {
68          counter = 9;
69          phase = !phase; // change road
70      } else {
71          counter--;
72      }
73
74      /* USER CODE END WHILE */
75
76      /* USER CODE BEGIN 3 */

```

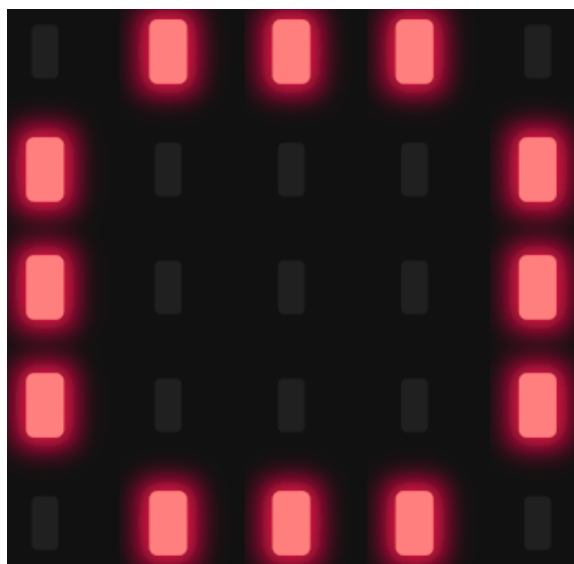
68
69

```
}
```

Program 1.3: Function Implementation

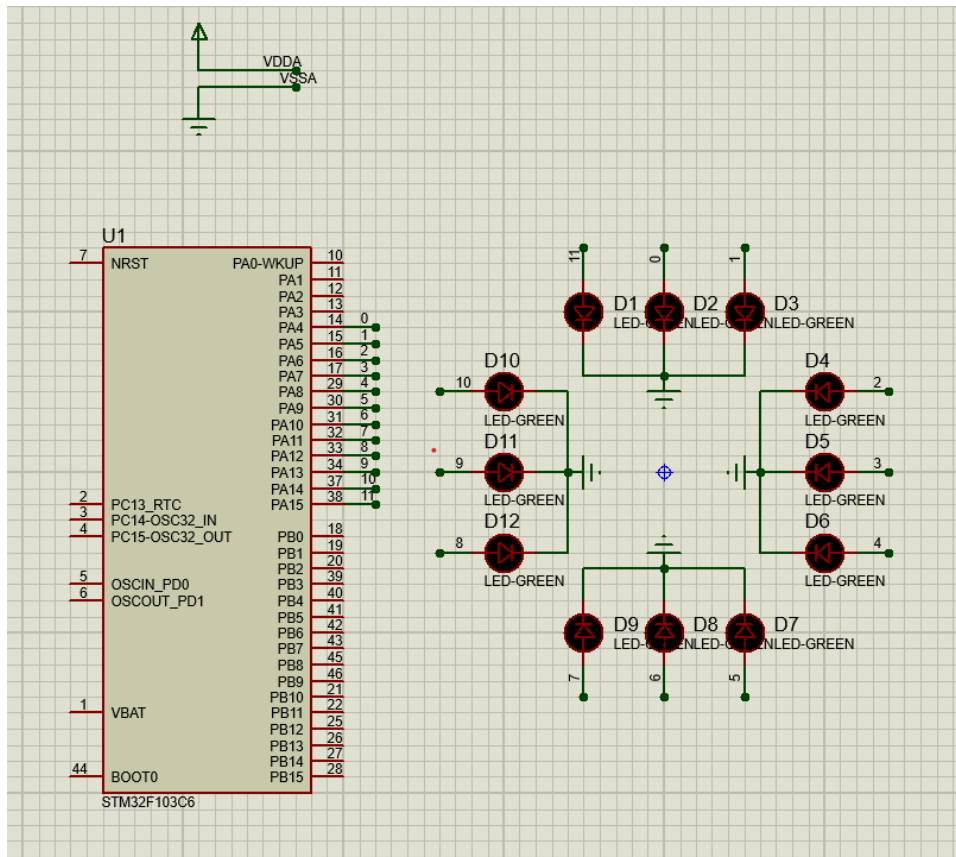
1.6 Exercise 6

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 1.4: 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. In order to turn on a LED, the corresponding pin should be in logic 1 (3.3V). The source code for this testing program is presented as following:

```

1
2 int first_time=1;
3 while (1)
4 {
5     if(first_time){
6         for (int i = 4; i <= 15; i++)
7         {
8             // Turn ON current LED
9             AL_GPIO_WritePin(GPIOA, (1 << i), GPIO_PIN_SET);
10            HAL_Delay(200); // delay 200ms
11            // Turn OFF current LED
12            HAL_GPIO_WritePin(GPIOA, (1 << i), GPIO_PIN_RESET);
13        }
14        first_time=0;
15    }
16 }

```

Program 1.4: Testing program for Exercise 6

1.7 Exercise 7

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

```
1 void clearAllClock() {  
2     for (int i = 4; i <= 15; i++) {  
3         HAL_GPIO_WritePin(GPIOA, (1 << i), GPIO_PIN_RESET); //  
4         Turn off all leds  
5     }  
6 }
```

Program 1.5: ClearAllClock function

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.

```
1 void setNumberOnClock(int num) {  
2     if (num < 0 || num > 11) return; // safety check  
3  
4     // Each LED corresponds to PA(num+4) due to PA4 is the  
5     begin Pin  
6     HAL_GPIO_WritePin(GPIOA, (1 << (num + 4)), GPIO_PIN_SET  
7     ); //Turn on led on Num  
8 }
```

Program 1.6: Set Number on Clock function

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     if (num < 0 || num > 11) return;  
3  
4     HAL_GPIO_WritePin(GPIOA, (1 << (num + 4)),  
5     GPIO_PIN_RESET); // Turn off led on Num  
6 }
```

Program 1.7: Clear Number on Clock function

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.

Source code for EX6-10: https://github.com/duyhoangbk23/Microcontroller_Lab/blob/main/EX6_10/EX6_10.c

Proteus schematic for EX6-10: https://github.com/duyhoangbk23/Microcontroller_Lab/blob/main/EX6_10/Proteus_EX6_10/EX6_10.pdsprj