



HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
COMPUTER ENGINEERING

Microcontroller



Dr. Le Trong Nhan



Microprocessor - Microcontroller (CO3009)

Laboratory Report

LAB 1

Teacher: Le Trong Nhan
Huynh Phuc Nghi
Class: L03
Student: Pham Van Nhat Vu (2110676)

HO CHI MINH CITY, 2023

Contents

Chapter 1. LED Animations	5
1 Exercise and Report	6
1.1 Exercise 1	6
1.2 Exercise 2	8
1.3 Exercise 3	10
1.4 Exercise 4	13
1.5 Exercise 5	19
1.6 Exercise 6	22
1.7 Exercise 7	24
1.8 Exercise 8	25
1.9 Exercise 9	26
1.10 Exercise 10	28

CHAPTER 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.

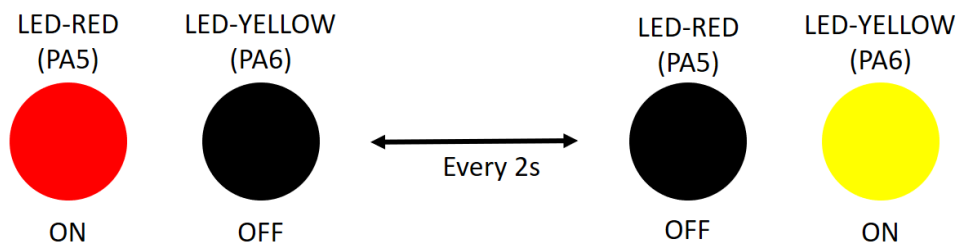


Figure 1.1: State transitions for 2 LEDs

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).

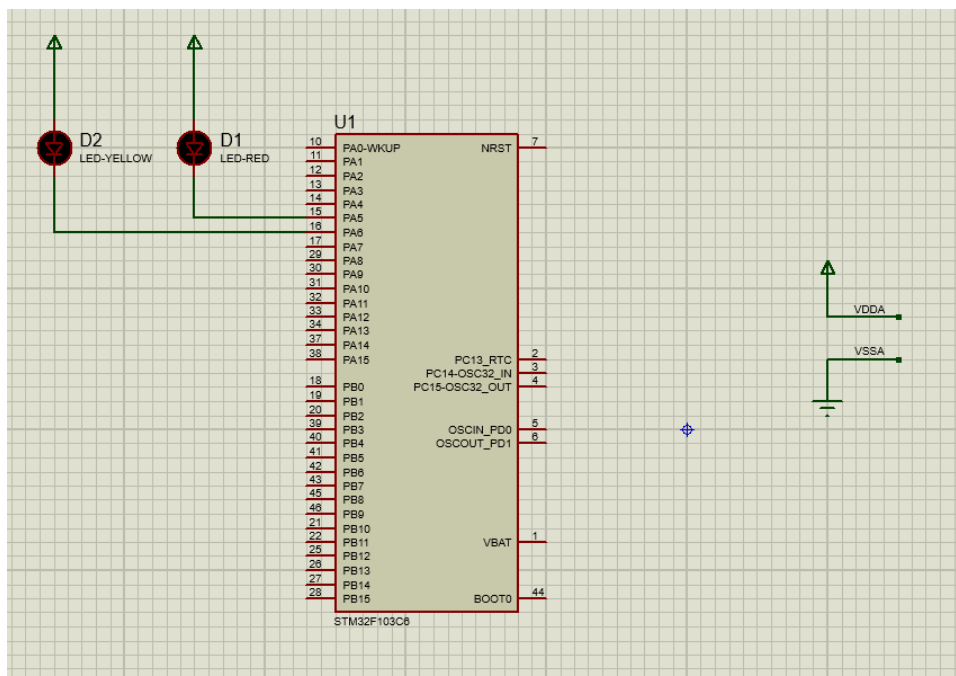


Figure 1.2: Schematic of [Proteus simulation project](#)

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).


```

1 #define TRANSITION_TIME 2
2 ...
3 HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin,
4   GPIO_PIN_RESET);
5 HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port, YELLOW_LED_Pin,
6   GPIO_PIN_SET);
7 uint8_t count = TRANSITION_TIME;
8 while (1)
9 {
10     if (count <= 0)
11     {
12         count = TRANSITION_TIME;
13         HAL_GPIO_TogglePin(RED_LED_GPIO_Port, RED_LED_Pin);
14         HAL_GPIO_TogglePin(YELLOW_LED_GPIO_Port,
15           YELLOW_LED_Pin);
16     }
17     --count;
18     HAL_Delay(1000);
19 /* USER CODE END WHILE */
20
21 /* USER CODE BEGIN 3 */
22 }

```

Program 1.1: Source code for exercise 1

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.

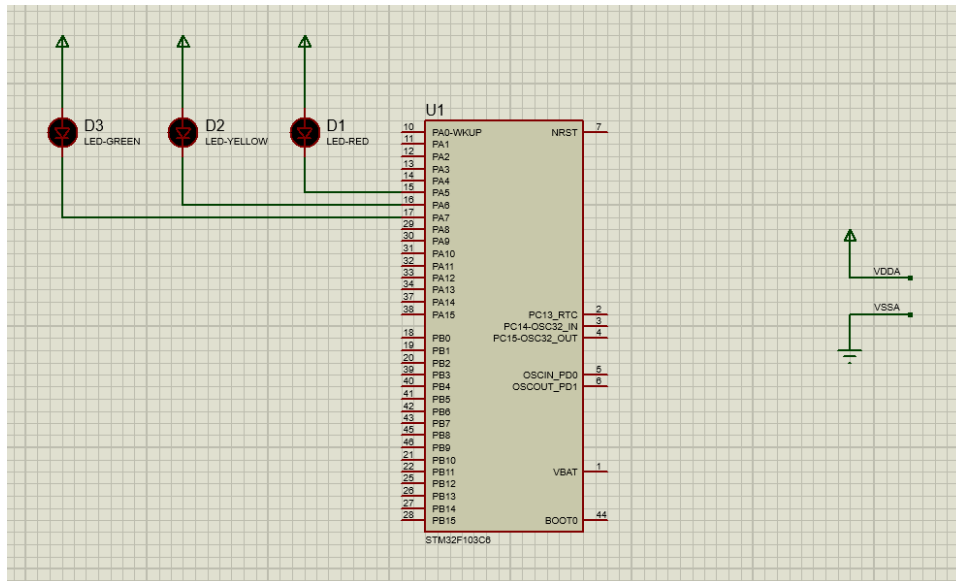


Figure 1.3: Schematic of *Proteus simulation project*

Report 2: Present the source code in while.

```
1 #define RED_DURATION 5
2 #define YELLOW_DURATION 2
3 #define GREEN_DURATION 3
4 #define RED_STATE 0
5 #define GREEN_STATE 1
6 #define YELLOW_STATE 2
7 ...
8 HAL_GPIO_WritePin(RED_LED_GPIO_Port , RED_LED_Pin ,
9   GPIO_PIN_RESET);
10 HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port , YELLOW_LED_Pin ,
11   GPIO_PIN_SET);
12 HAL_GPIO_WritePin(GREEN_LED_GPIO_Port , GREEN_LED_Pin ,
13   GPIO_PIN_SET);
14 uint8_t count = RED_DURATION;
15 uint8_t state = RED_STATE;
16 while (1)
17 {
18     switch (state)
```

```

16 {
17 case RED_STATE:
18     HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin,
19     GPIO_PIN_RESET);
20     if (count <= 0)
21     {
22         count = GREEN_DURATION;
23         state = GREEN_STATE;
24         HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin,
25         GPIO_PIN_SET);
26         HAL_GPIO_WritePin(GREEN_LED_GPIO_Port,
27         GREEN_LED_Pin, GPIO_PIN_RESET);
28     }
29     break;
30 case GREEN_STATE:
31     HAL_GPIO_WritePin(GREEN_LED_GPIO_Port, GREEN_LED_Pin,
32     GPIO_PIN_RESET);
33     if (count <= 0)
34     {
35         count = YELLOW_DURATION;
36         state = YELLOW_STATE;
37         HAL_GPIO_WritePin(GREEN_LED_GPIO_Port,
38         GREEN_LED_Pin, GPIO_PIN_SET);
39         HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port,
40         YELLOW_LED_Pin, GPIO_PIN_RESET);
41     }
42     break;
43 case YELLOW_STATE:
44     HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port,
45     YELLOW_LED_Pin, GPIO_PIN_RESET);
46     if (count <= 0)
47     {
48         count = RED_DURATION;
49         state = RED_STATE;
50         HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port,
51         YELLOW_LED_Pin, GPIO_PIN_SET);
52         HAL_GPIO_WritePin(RED_LED_GPIO_Port, RED_LED_Pin,
53         GPIO_PIN_RESET);
54     }
55     break;
56 }
57 --count;
58 HAL_Delay(1000);
59 }

```

Program 1.2: Source code for exercise 2

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.

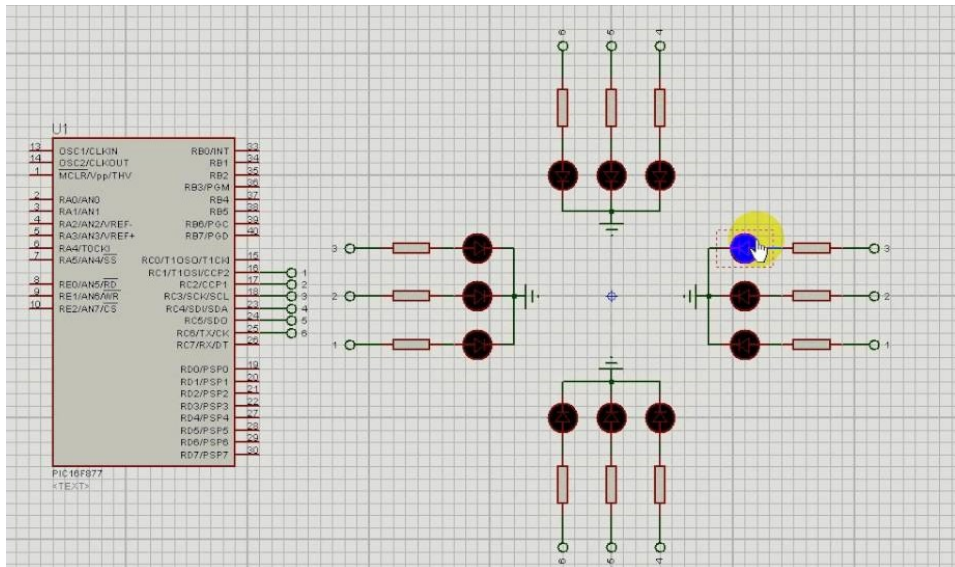


Figure 1.4: Reference design for a 4 way traffic light

Report 1: Present the schematic.

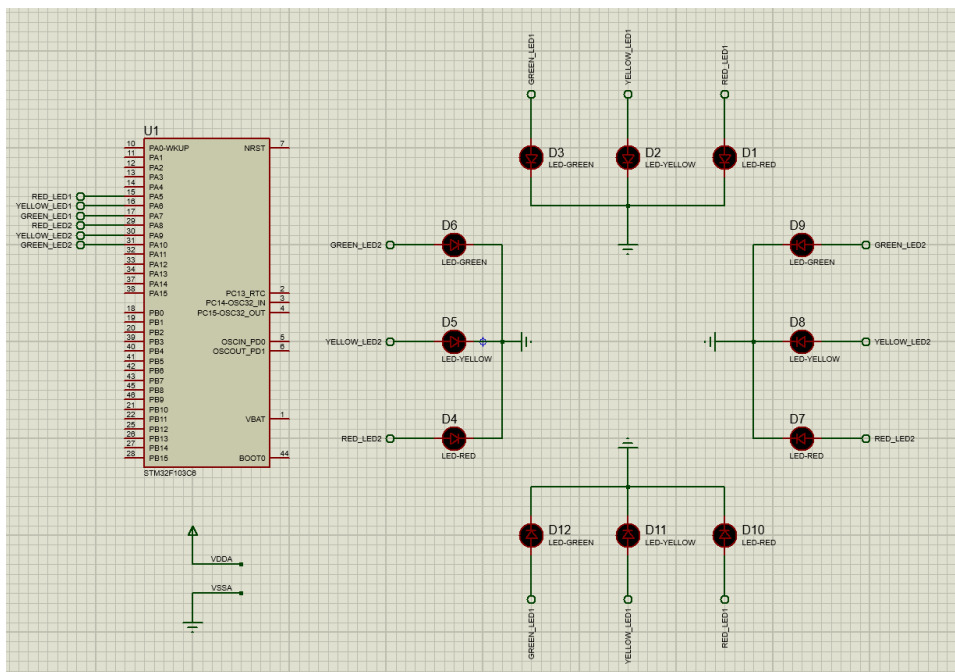


Figure 1.5: Schematic of Proteus simulation project

Report 2: Present the source code in while.

```
1 enum TrafficLightState
2 {
3     RED1_GREEN2,
```

```

4   RED1_YELLOW2 ,
5   GREEN1_RED2 ,
6   YELLOW1_RED2
7 };
8
9 enum Led
10 {
11     RED1 ,
12     RED2 ,
13     GREEN1 ,
14     GREEN2 ,
15     YELLOW1 ,
16     YELLOW2
17 };
18
19 void turnOnLight(enum Led);
20 void clearAllLed();
21
22 int main(void)
23 {
24     HAL_Init();
25     SystemClock_Config();
26     MX_GPIO_Init();
27
28     turnOnLight(RED1);
29     turnOnLight(GREEN1);
30     uint8_t count = 3;
31     uint8_t light = 5;
32     uint8_t state = RED1_GREEN2;
33     while (1)
34     {
35         switch (state)
36         {
37             case RED1_GREEN2:
38                 clearAllLed();
39                 turnOnLight(RED1);
40                 turnOnLight(GREEN2);
41                 --count;
42                 --light;
43                 if (count <= 0)
44                 {
45                     count = 2;
46                     state = RED1_YELLOW2;
47                 }
48                 break;
49             case RED1_YELLOW2:
50                 clearAllLed();
51                 turnOnLight(RED1);
52                 turnOnLight(YELLOW2);

```

```

53         --count;
54         --light;
55         if (count <= 0)
56         {
57             count = 3;
58             light = 3;
59             state = GREEN1_RED2;
60         }
61         break;
62     case GREEN1_RED2:
63         clearAllLed();
64         turnOnLight(GREEN1);
65         turnOnLight(RED2);
66         --count;
67         --light;
68         if (count <= 0)
69         {
70             count = 2;
71             light = 2;
72             state = YELLOW1_RED2;
73         }
74         break;
75     case YELLOW1_RED2:
76         clearAllLed();
77         turnOnLight(YELLOW1);
78         turnOnLight(RED2);
79         --count;
80         --light;
81         if (count <= 0)
82         {
83             count = 3;
84             light = 5;
85             state = RED1_GREEN2;
86         }
87         break;
88     }
89     HAL_Delay(1000);
90     /* USER CODE END WHILE */
91
92     /* USER CODE BEGIN 3 */
93 }
94
95 void clearAllLed()
96 {
97     HAL_GPIO_WritePin(RED_LED1_GPIO_Port, RED_LED1_Pin,
98     GPIO_PIN_RESET);
99     HAL_GPIO_WritePin(GREEN_LED1_GPIO_Port, GREEN_LED1_Pin,
    GPIO_PIN_RESET);

```

```

100     HAL_GPIO_WritePin(YELLOW_LED1_GPIO_Port ,
YELLOW_LED1_Pin , GPIO_PIN_RESET);
101     HAL_GPIO_WritePin(RED_LED2_GPIO_Port , RED_LED2_Pin ,
GPIO_PIN_RESET);
102     HAL_GPIO_WritePin(GREEN_LED2_GPIO_Port , GREEN_LED2_Pin ,
GPIO_PIN_RESET);
103     HAL_GPIO_WritePin(YELLOW_LED2_GPIO_Port ,
YELLOW_LED2_Pin , GPIO_PIN_RESET);
104 }
105
106 void turnOnLight(enum Led led)
107 {
108     switch (led)
109     {
110         case RED1:
111             HAL_GPIO_WritePin(RED_LED1_GPIO_Port , RED_LED1_Pin ,
GPIO_PIN_SET);
112             break;
113         case GREEN1:
114             HAL_GPIO_WritePin(GREEN_LED1_GPIO_Port ,
GREEN_LED1_Pin , GPIO_PIN_SET);
115             break;
116         case YELLOW1:
117             HAL_GPIO_WritePin(YELLOW_LED1_GPIO_Port ,
YELLOW_LED1_Pin , GPIO_PIN_SET);
118             break;
119         case RED2:
120             HAL_GPIO_WritePin(RED_LED2_GPIO_Port , RED_LED2_Pin ,
GPIO_PIN_SET);
121             break;
122         case GREEN2:
123             HAL_GPIO_WritePin(GREEN_LED2_GPIO_Port ,
GREEN_LED2_Pin , GPIO_PIN_SET);
124             break;
125         case YELLOW2:
126             HAL_GPIO_WritePin(YELLOW_LED2_GPIO_Port ,
YELLOW_LED2_Pin , GPIO_PIN_SET);
127             break;
128     }
129 }

```

Program 1.3: Source code for exercise 3

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:

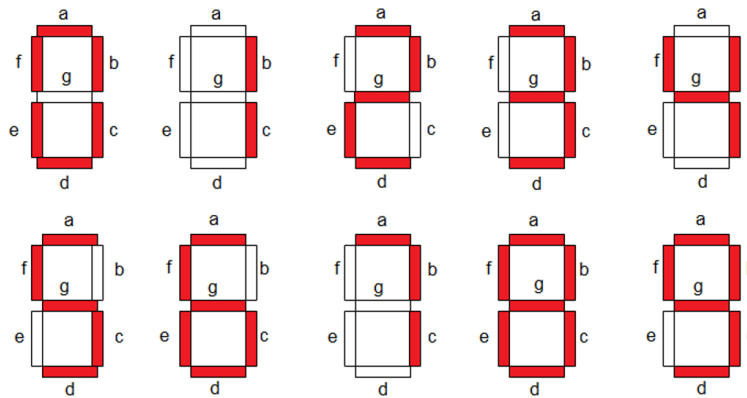
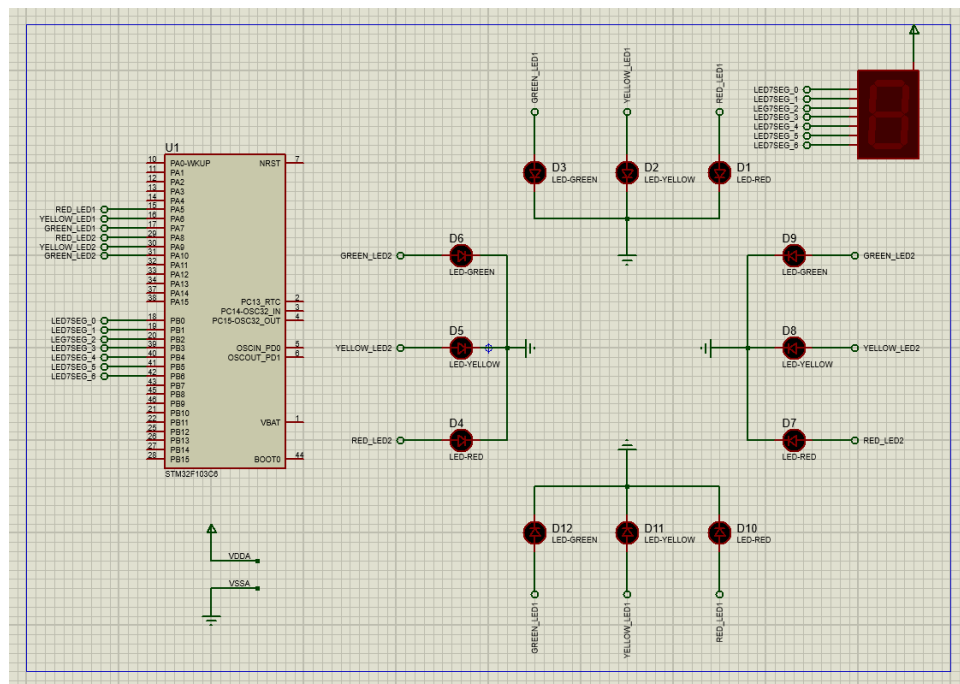


Figure 1.6: Display a number on 7 segment LED


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

Report 1: Present the schematic.



Report 2: Present the source code for display7SEG function.

```
1 void display7SEG(int number)
2 {
3     switch (number)
4     {
5     case 0:
6         HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port, LED7SEG_0_Pin,
7                             GPIO_PIN_RESET);
8         HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port, LED7SEG_1_Pin,
9                             GPIO_PIN_RESET);
10        HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port, LED7SEG_2_Pin,
11                            GPIO_PIN_RESET);
12        HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port, LED7SEG_3_Pin,
13                            GPIO_PIN_RESET);
14        HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port, LED7SEG_4_Pin,
15                            GPIO_PIN_RESET);
```

```

11     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port , LED7SEG_5_Pin ,
    GPIO_PIN_RESET);
12     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port , LED7SEG_6_Pin ,
    GPIO_PIN_SET);
13     break;
14 case 1:
15     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port , LED7SEG_0_Pin ,
    GPIO_PIN_SET);
16     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port , LED7SEG_1_Pin ,
    GPIO_PIN_RESET);
17     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port , LED7SEG_2_Pin ,
    GPIO_PIN_RESET);
18     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port , LED7SEG_3_Pin ,
    GPIO_PIN_SET);
19     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port , LED7SEG_4_Pin ,
    GPIO_PIN_SET);
20     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port , LED7SEG_5_Pin ,
    GPIO_PIN_SET);
21     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port , LED7SEG_6_Pin ,
    GPIO_PIN_SET);
22     break;
23 case 2:
24     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port , LED7SEG_0_Pin ,
    GPIO_PIN_RESET);
25     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port , LED7SEG_1_Pin ,
    GPIO_PIN_RESET);
26     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port , LED7SEG_2_Pin ,
    GPIO_PIN_SET);
27     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port , LED7SEG_3_Pin ,
    GPIO_PIN_RESET);
28     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port , LED7SEG_4_Pin ,
    GPIO_PIN_RESET);
29     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port , LED7SEG_5_Pin ,
    GPIO_PIN_SET);
30     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port , LED7SEG_6_Pin ,
    GPIO_PIN_RESET);
31     break;
32 case 3:
33     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port , LED7SEG_0_Pin ,
    GPIO_PIN_RESET);
34     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port , LED7SEG_1_Pin ,
    GPIO_PIN_RESET);
35     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port , LED7SEG_2_Pin ,
    GPIO_PIN_RESET);
36     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port , LED7SEG_3_Pin ,
    GPIO_PIN_RESET);
37     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port , LED7SEG_4_Pin ,
    GPIO_PIN_SET);
38     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port , LED7SEG_5_Pin ,

```

```

GPIO_PIN_SET);
39     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port, LED7SEG_6_Pin,
GPIO_PIN_RESET);
40     break;
41 case 4:
42     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port, LED7SEG_0_Pin,
GPIO_PIN_SET);
43     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port, LED7SEG_1_Pin,
GPIO_PIN_RESET);
44     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port, LED7SEG_2_Pin,
GPIO_PIN_RESET);
45     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port, LED7SEG_3_Pin,
GPIO_PIN_SET);
46     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port, LED7SEG_4_Pin,
GPIO_PIN_SET);
47     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port, LED7SEG_5_Pin,
GPIO_PIN_RESET);
48     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port, LED7SEG_6_Pin,
GPIO_PIN_RESET);
49     break;
50 case 5:
51     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port, LED7SEG_0_Pin,
GPIO_PIN_RESET);
52     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port, LED7SEG_1_Pin,
GPIO_PIN_SET);
53     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port, LED7SEG_2_Pin,
GPIO_PIN_RESET);
54     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port, LED7SEG_3_Pin,
GPIO_PIN_RESET);
55     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port, LED7SEG_4_Pin,
GPIO_PIN_SET);
56     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port, LED7SEG_5_Pin,
GPIO_PIN_RESET);
57     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port, LED7SEG_6_Pin,
GPIO_PIN_RESET);
58     break;
59 case 6:
60     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port, LED7SEG_0_Pin,
GPIO_PIN_RESET);
61     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port, LED7SEG_1_Pin,
GPIO_PIN_SET);
62     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port, LED7SEG_2_Pin,
GPIO_PIN_RESET);
63     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port, LED7SEG_3_Pin,
GPIO_PIN_RESET);
64     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port, LED7SEG_4_Pin,
GPIO_PIN_RESET);
65     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port, LED7SEG_5_Pin,
GPIO_PIN_RESET);

```

```

66     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port , LED7SEG_6_Pin ,
    GPIO_PIN_RESET);
67     break;
68 case 7:
69     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port , LED7SEG_0_Pin ,
    GPIO_PIN_RESET);
70     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port , LED7SEG_1_Pin ,
    GPIO_PIN_RESET);
71     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port , LED7SEG_2_Pin ,
    GPIO_PIN_RESET);
72     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port , LED7SEG_3_Pin ,
    GPIO_PIN_SET);
73     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port , LED7SEG_4_Pin ,
    GPIO_PIN_SET);
74     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port , LED7SEG_5_Pin ,
    GPIO_PIN_SET);
75     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port , LED7SEG_6_Pin ,
    GPIO_PIN_SET);
76     break;
77 case 8:
78     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port , LED7SEG_0_Pin ,
    GPIO_PIN_RESET);
79     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port , LED7SEG_1_Pin ,
    GPIO_PIN_RESET);
80     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port , LED7SEG_2_Pin ,
    GPIO_PIN_RESET);
81     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port , LED7SEG_3_Pin ,
    GPIO_PIN_RESET);
82     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port , LED7SEG_4_Pin ,
    GPIO_PIN_RESET);
83     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port , LED7SEG_5_Pin ,
    GPIO_PIN_RESET);
84     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port , LED7SEG_6_Pin ,
    GPIO_PIN_RESET);
85     break;
86 case 9:
87     HAL_GPIO_WritePin(LED7SEG_0_GPIO_Port , LED7SEG_0_Pin ,
    GPIO_PIN_RESET);
88     HAL_GPIO_WritePin(LED7SEG_1_GPIO_Port , LED7SEG_1_Pin ,
    GPIO_PIN_RESET);
89     HAL_GPIO_WritePin(LED7SEG_2_GPIO_Port , LED7SEG_2_Pin ,
    GPIO_PIN_RESET);
90     HAL_GPIO_WritePin(LED7SEG_3_GPIO_Port , LED7SEG_3_Pin ,
    GPIO_PIN_RESET);
91     HAL_GPIO_WritePin(LED7SEG_4_GPIO_Port , LED7SEG_4_Pin ,
    GPIO_PIN_SET);
92     HAL_GPIO_WritePin(LED7SEG_5_GPIO_Port , LED7SEG_5_Pin ,
    GPIO_PIN_RESET);
93     HAL_GPIO_WritePin(LED7SEG_6_GPIO_Port , LED7SEG_6_Pin ,

```

```

    GPIO_PIN_RESET);
94     break;
95 }
96 }

```

Program 1.5: Source code of function display7SEG() for exercise 2

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.

The source code of this exercise is almost the same as the source code of exercise 3, only an invocation of function display7SEG(light) is added at line **35** to display the countdown value of the up and down traffic light.

```

1  enum TrafficLightState
2  {
3      RED1_GREEN2 ,
4      RED1_YELLOW2 ,
5      GREEN1_RED2 ,
6      YELLOW1_RED2
7  };
8
9  enum Led
10 {
11     RED1 ,
12     RED2 ,
13     GREEN1 ,
14     GREEN2 ,
15     YELLOW1 ,
16     YELLOW2
17 };
18
19 void turnOnLight(enum Led);
20 void clearAllLed();
21
22 int main(void)
23 {
24     HAL_Init();
25     SystemClock_Config();
26     MX_GPIO_Init();
27
28     turnOnLight(RED1);
29     turnOnLight(GREEN1);
30     uint8_t count = 3;
31     uint8_t light = 5;

```

```

32     uint8_t state = RED1_GREEN2;
33     while (1)
34     {
35         display7SEG(light);
36         switch (state)
37         {
38             case RED1_GREEN2:
39                 clearAllLed();
40                 turnOnLight(RED1);
41                 turnOnLight(GREEN2);
42                 --count;
43                 --light;
44                 if (count <= 0)
45                 {
46                     count = 2;
47                     state = RED1_YELLOW2;
48                 }
49                 break;
50             case RED1_YELLOW2:
51                 clearAllLed();
52                 turnOnLight(RED1);
53                 turnOnLight(YELLOW2);
54                 --count;
55                 --light;
56                 if (count <= 0)
57                 {
58                     count = 3;
59                     light = 3;
60                     state = GREEN1_RED2;
61                 }
62                 break;
63             case GREEN1_RED2:
64                 clearAllLed();
65                 turnOnLight(GREEN1);
66                 turnOnLight(RED2);
67                 --count;
68                 --light;
69                 if (count <= 0)
70                 {
71                     count = 2;
72                     light = 2;
73                     state = YELLOW1_RED2;
74                 }
75                 break;
76             case YELLOW1_RED2:
77                 clearAllLed();
78                 turnOnLight(YELLOW1);
79                 turnOnLight(RED2);
80                 --count;

```

```

81         --light;
82         if (count <= 0)
83         {
84             count = 3;
85             light = 5;
86             state = RED1_GREEN2;
87         }
88         break;
89     }
90     HAL_Delay(1000);
91     /* USER CODE END WHILE */
92
93     /* USER CODE BEGIN 3 */
94 }
95 }
96
97 void clearAllLed()
98 {
99     HAL_GPIO_WritePin(RED_LED1_GPIO_Port, RED_LED1_Pin,
100     GPIO_PIN_RESET);
101     HAL_GPIO_WritePin(GREEN_LED1_GPIO_Port, GREEN_LED1_Pin,
102     GPIO_PIN_RESET);
103     HAL_GPIO_WritePin(YELLOW_LED1_GPIO_Port,
104     YELLOW_LED1_Pin, GPIO_PIN_RESET);
105     HAL_GPIO_WritePin(RED_LED2_GPIO_Port, RED_LED2_Pin,
106     GPIO_PIN_RESET);
107     HAL_GPIO_WritePin(GREEN_LED2_GPIO_Port, GREEN_LED2_Pin,
108     GPIO_PIN_RESET);
109     HAL_GPIO_WritePin(YELLOW_LED2_GPIO_Port,
110     YELLOW_LED2_Pin, GPIO_PIN_RESET);
111 }
112
113 void turnOnLight(enum Led led)
114 {
115     switch (led)
116     {
117     case RED1:
118         HAL_GPIO_WritePin(RED_LED1_GPIO_Port, RED_LED1_Pin,
119         GPIO_PIN_SET);
120         break;
121     case GREEN1:
122         HAL_GPIO_WritePin(GREEN_LED1_GPIO_Port,
123         GREEN_LED1_Pin, GPIO_PIN_SET);
124         break;
125     case YELLOW1:
126         HAL_GPIO_WritePin(YELLOW_LED1_GPIO_Port,
127         YELLOW_LED1_Pin, GPIO_PIN_SET);
128         break;
129     case RED2:

```

```

121     HAL_GPIO_WritePin(RED_LED2_GPIO_Port , RED_LED2_Pin ,
    GPIO_PIN_SET);
122     break;
123     case GREEN2:
124         HAL_GPIO_WritePin(GREEN_LED2_GPIO_Port ,
    GREEN_LED2_Pin , GPIO_PIN_SET);
125         break;
126     case YELLOW2:
127         HAL_GPIO_WritePin(YELLOW_LED2_GPIO_Port ,
    YELLOW_LED2_Pin , GPIO_PIN_SET);
128         break;
129     }
130 }

```

Program 1.6: Source code for exercise 5

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.

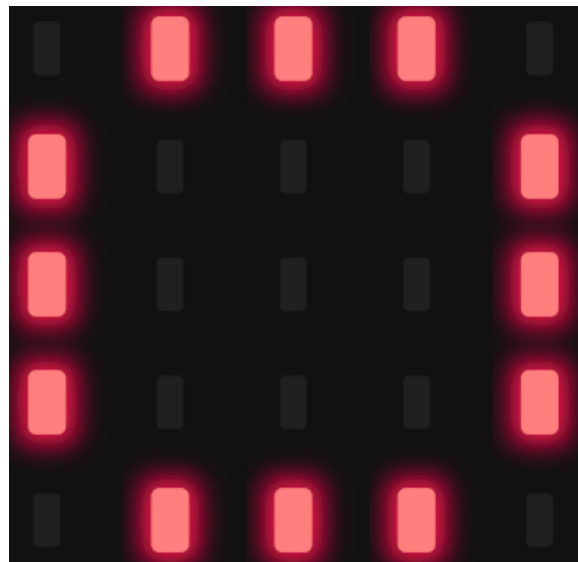


Figure 1.8: 12 LEDs for an analog clock

Report 1: Present the schematic.

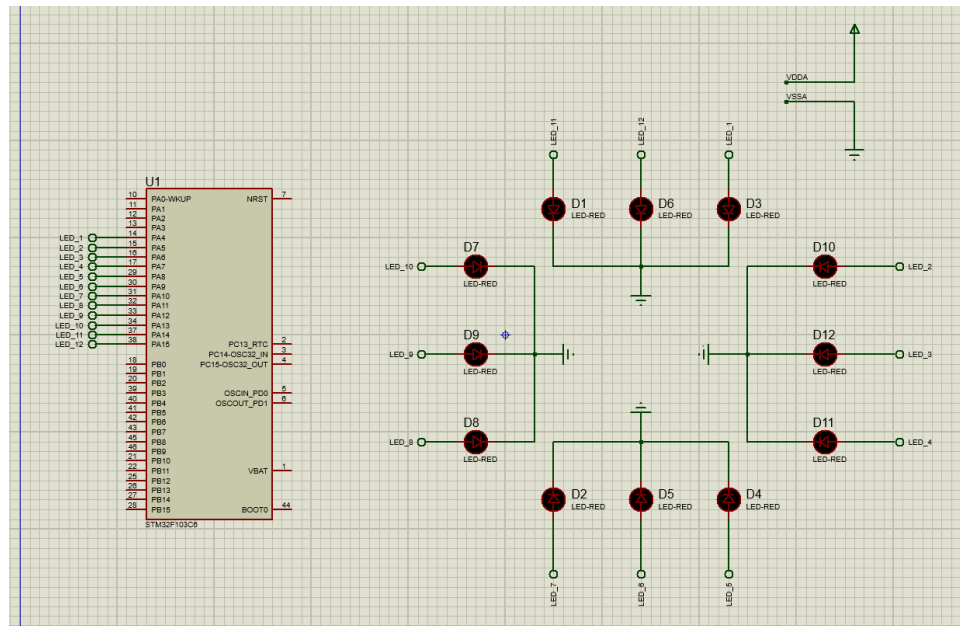


Figure 1.9: Schematic of *Proteus simulation project*

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 uint8_t count = 0;
2 while (1)
3 {
4     ++count;
5     switch (count)
6     {
7     case 1:
8         HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin,
9                             GPIO_PIN_SET);
10        break;
11     case 2:
12        HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin,
13                            GPIO_PIN_SET);
14        break;
15     case 3:
16        HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin,
17                            GPIO_PIN_SET);
18        break;
19     case 4:
20        HAL_GPIO_WritePin(LED4_GPIO_Port, LED4_Pin,
21                            GPIO_PIN_SET);
22        break;
23     case 5:
24        HAL_GPIO_WritePin(LED5_GPIO_Port, LED5_Pin,
25                            GPIO_PIN_SET);
26        break;
27     case 6:
28        HAL_GPIO_WritePin(LED6_GPIO_Port, LED6_Pin,
29                            GPIO_PIN_SET);
30        break;
31     }
32 }

```

```

    GPIO_PIN_SET);
24     break;
25 case 7:
26     HAL_GPIO_WritePin(LED7_GPIO_Port, LED7_Pin,
    GPIO_PIN_SET);
27     break;
28 case 8:
29     HAL_GPIO_WritePin(LED8_GPIO_Port, LED8_Pin,
    GPIO_PIN_SET);
30     break;
31 case 9:
32     HAL_GPIO_WritePin(LED9_GPIO_Port, LED9_Pin,
    GPIO_PIN_SET);
33     break;
34 case 10:
35     HAL_GPIO_WritePin(LED10_GPIO_Port, LED10_Pin,
    GPIO_PIN_SET);
36     break;
37 case 11:
38     HAL_GPIO_WritePin(LED11_GPIO_Port, LED11_Pin,
    GPIO_PIN_SET);
39     break;
40 case 12:
41     HAL_GPIO_WritePin(LED12_GPIO_Port, LED12_Pin,
    GPIO_PIN_SET);
42     break;
43 default:
44     count = 13;
45     break;
46 }
47 HAL_Delay(1000);
48 /* USER CODE END WHILE */
49
50 /* USER CODE BEGIN 3 */
51 }

```

Program 1.7: Source code for testing the connection of every single LED

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 {
3     HAL_GPIO_WritePin(LED1_GPIO_Port, LED1_Pin,
    GPIO_PIN_RESET);
4     HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin,
    GPIO_PIN_RESET);

```

```

5  HAL_GPIO_WritePin(LED3_GPIO_Port , LED3_Pin ,
    GPIO_PIN_RESET);
6  HAL_GPIO_WritePin(LED4_GPIO_Port , LED4_Pin ,
    GPIO_PIN_RESET);
7  HAL_GPIO_WritePin(LED5_GPIO_Port , LED5_Pin ,
    GPIO_PIN_RESET);
8  HAL_GPIO_WritePin(LED6_GPIO_Port , LED6_Pin ,
    GPIO_PIN_RESET);
9  HAL_GPIO_WritePin(LED7_GPIO_Port , LED7_Pin ,
    GPIO_PIN_RESET);
10 HAL_GPIO_WritePin(LED8_GPIO_Port , LED8_Pin ,
    GPIO_PIN_RESET);
11 HAL_GPIO_WritePin(LED9_GPIO_Port , LED9_Pin ,
    GPIO_PIN_RESET);
12 HAL_GPIO_WritePin(LED10_GPIO_Port , LED10_Pin ,
    GPIO_PIN_RESET);
13 HAL_GPIO_WritePin(LED11_GPIO_Port , LED11_Pin ,
    GPIO_PIN_RESET);
14 HAL_GPIO_WritePin(LED12_GPIO_Port , LED12_Pin ,
    GPIO_PIN_RESET);
15 }

```

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.

```

1  void setNumberOnClock(int num)
2  {
3      switch (num)
4      {
5          case 1:
6              HAL_GPIO_WritePin(LED1_GPIO_Port , LED1_Pin ,
                GPIO_PIN_SET);
7              break;
8          case 2:
9              HAL_GPIO_WritePin(LED2_GPIO_Port , LED2_Pin ,
                GPIO_PIN_SET);
10             break;
11          case 3:
12              HAL_GPIO_WritePin(LED3_GPIO_Port , LED3_Pin ,
                GPIO_PIN_SET);
13             break;
14          case 4:
15              HAL_GPIO_WritePin(LED4_GPIO_Port , LED4_Pin ,
                GPIO_PIN_SET);
16             break;
17          case 5:

```

```

18     HAL_GPIO_WritePin(LED5_GPIO_Port , LED5_Pin ,
19     GPIO_PIN_SET);
20     break;
21 case 6:
22     HAL_GPIO_WritePin(LED6_GPIO_Port , LED6_Pin ,
23     GPIO_PIN_SET);
24     break;
25 case 7:
26     HAL_GPIO_WritePin(LED7_GPIO_Port , LED7_Pin ,
27     GPIO_PIN_SET);
28     break;
29 case 8:
30     HAL_GPIO_WritePin(LED8_GPIO_Port , LED8_Pin ,
31     GPIO_PIN_SET);
32     break;
33 case 9:
34     HAL_GPIO_WritePin(LED9_GPIO_Port , LED9_Pin ,
35     GPIO_PIN_SET);
36     break;
37 case 10:
38     HAL_GPIO_WritePin(LED10_GPIO_Port , LED10_Pin ,
39     GPIO_PIN_SET);
40     break;
41 case 11:
42     HAL_GPIO_WritePin(LED11_GPIO_Port , LED11_Pin ,
43     GPIO_PIN_SET);
44     break;
45 case 0:
46     HAL_GPIO_WritePin(LED12_GPIO_Port , LED12_Pin ,
47     GPIO_PIN_SET);
48     break;
49 }
50 }

```

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     switch (num)
4     {
5     case 1:
6         HAL_GPIO_WritePin(LED1_GPIO_Port , LED1_Pin ,
7         GPIO_PIN_RESET);
8         break;
9     case 2:

```

```

9     HAL_GPIO_WritePin(LED2_GPIO_Port, LED2_Pin,
GPIO_PIN_RESET);
10     break;
11 case 3:
12     HAL_GPIO_WritePin(LED3_GPIO_Port, LED3_Pin,
GPIO_PIN_RESET);
13     break;
14 case 4:
15     HAL_GPIO_WritePin(LED4_GPIO_Port, LED4_Pin,
GPIO_PIN_RESET);
16     break;
17 case 5:
18     HAL_GPIO_WritePin(LED5_GPIO_Port, LED5_Pin,
GPIO_PIN_RESET);
19     break;
20 case 6:
21     HAL_GPIO_WritePin(LED6_GPIO_Port, LED6_Pin,
GPIO_PIN_RESET);
22     break;
23 case 7:
24     HAL_GPIO_WritePin(LED7_GPIO_Port, LED7_Pin,
GPIO_PIN_RESET);
25     break;
26 case 8:
27     HAL_GPIO_WritePin(LED8_GPIO_Port, LED8_Pin,
GPIO_PIN_RESET);
28     break;
29 case 9:
30     HAL_GPIO_WritePin(LED9_GPIO_Port, LED9_Pin,
GPIO_PIN_RESET);
31     break;
32 case 10:
33     HAL_GPIO_WritePin(LED10_GPIO_Port, LED10_Pin,
GPIO_PIN_RESET);
34     break;
35 case 11:
36     HAL_GPIO_WritePin(LED11_GPIO_Port, LED11_Pin,
GPIO_PIN_RESET);
37     break;
38 case 0:
39     HAL_GPIO_WritePin(LED12_GPIO_Port, LED12_Pin,
GPIO_PIN_RESET);
40     break;
41 }
42 }

```

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 #define SECONDS_PER_TICK 5
2 #define SECOND_CYCLES_PER_MINUTE 5
3 #define NUM_OF_TICKS 12
4 #define SECOND_DURATION 1
5 ...
6 uint8_t counter_second_led = 0;
7 uint8_t counter_minute_led = 0;
8 uint8_t counter_hour_led = 0;
9 uint8_t counter_second_cycles = SECOND_CYCLES_PER_MINUTE;
10 uint8_t counter_second = SECONDS_PER_TICK;
11 uint8_t timer_counter = SECOND_DURATION;
12 clearAllClock();
13 setNumberOnClock(counter_second_led);
14 setNumberOnClock(counter_minute_led);
15 setNumberOnClock(counter_hour_led);
16 while (1)
17 {
18     if (timer_counter <= 0)
19     {
20         timer_counter = SECOND_DURATION;
21         --counter_second;
22         if (counter_second <= 0)
23         {
24             counter_second = SECONDS_PER_TICK;
25             clearNumberOnClock(counter_second_led);
26             clearNumberOnClock(counter_minute_led);
27             clearNumberOnClock(counter_hour_led);
28
29             ++counter_second_led;
30             if (counter_second_led == NUM_OF_TICKS)
31             {
32                 counter_second_led = 0;
33                 --counter_second_cycles;
34                 if (counter_second_cycles <= 0)
35                 {
36                     counter_second_cycles =
37                     SECOND_CYCLES_PER_MINUTE;
38                     ++counter_minute_led;
39                 }
40             }
41             if (counter_minute_led == NUM_OF_TICKS)
42             {
43                 counter_minute_led = 0;
44                 ++counter_hour_led;
```

```

45         }
46
47         if (counter_hour_led == NUM_OF_TICKS)
48         {
49             counter_hour_led = 0;
50         }
51
52
53         setNumberOnClock(counter_second_led);
54         setNumberOnClock(counter_minute_led);
55         setNumberOnClock(counter_hour_led);
56     }
57 }
58 --timer_counter;
59 HAL_Delay(1000);
60 /* USER CODE END WHILE */
61
62 /* USER CODE BEGIN 3 */
63 }

```

Program 1.11: Source code for exercise 10