

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



**MICROPROCESSORS
MICRO-CONTROLLERS
CO3009 - CC01**

Report

Lab 1 - LED Animations

Advisors: Lê Trọng Nhân
Nguyễn Trần Hữu Nguyên

Student: Bùi Phát Lộc - 1752326

HO CHI MINH CITY, Q4 2022



Contents

1	Exercise 1	3
2	Exercise 2	5
3	Exercise 3	7
4	Exercise 4	10
5	Exercise 5	13
6	Exercise 6, 7, 8, 9, 10	21



This is the drive link contains all file in Lab 1, including STM32 file and Proteus simulation file:
[Github Link](#) and [Back up GG Drive Link](#)

1 Exercise 1

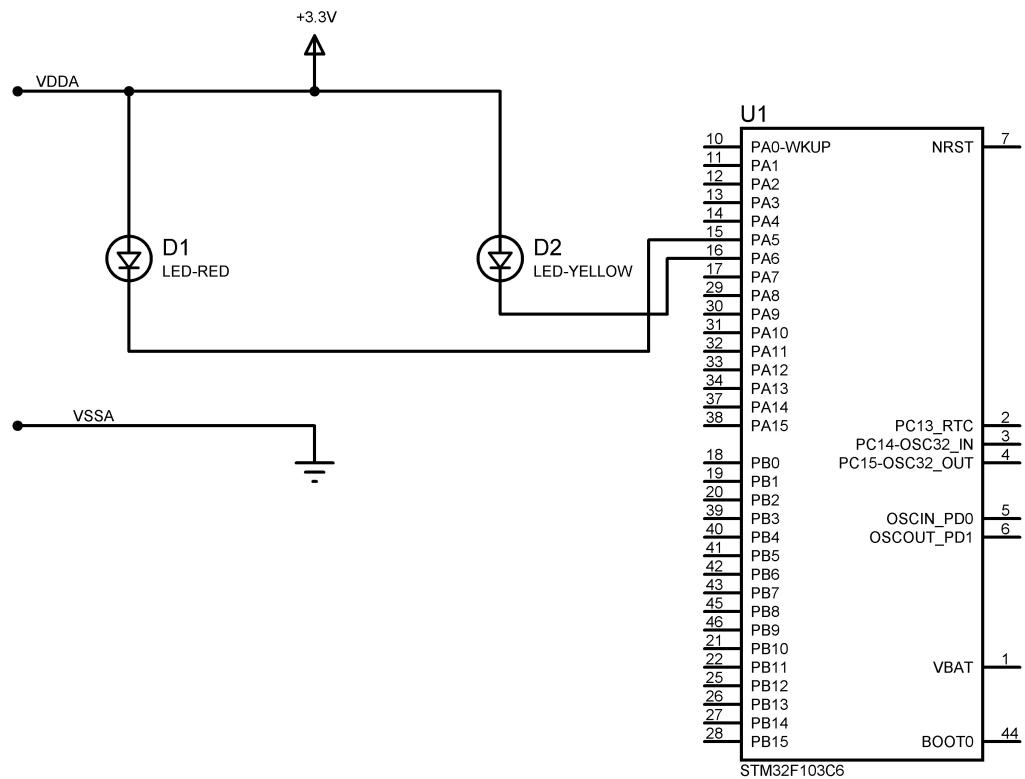


Figure 1.1: [Github Link](#) and [Back up GG Drive Link](#)



```
1  char color = 'N';
2  while (1)
3  {
4      if(color == 'N')
5      {
6          HAL_GPIO_TogglePin(Led_yellow_GPIO_Port, Led_yellow_Pin);
7          color = 'Y';
8      }
9      else if(color == 'R')
10     {
11         HAL_GPIO_TogglePin(Led_yellow_GPIO_Port, Led_yellow_Pin);
12         HAL_GPIO_TogglePin(Led_red_GPIO_Port, Led_red_Pin);
13         color = 'Y';
14     }
15     else if (color == 'Y')
16     {
17         HAL_GPIO_TogglePin(Led_red_GPIO_Port, Led_red_Pin);
18         HAL_GPIO_TogglePin(Led_yellow_GPIO_Port, Led_yellow_Pin);
19         color = 'R';
20     }
21     HAL_Delay(2000);
22
23     /* USER CODE END WHILE */
24
25     /* USER CODE BEGIN 3 */
26 }
```

Program 1.1: Ex1 source code

2 Exercise 2

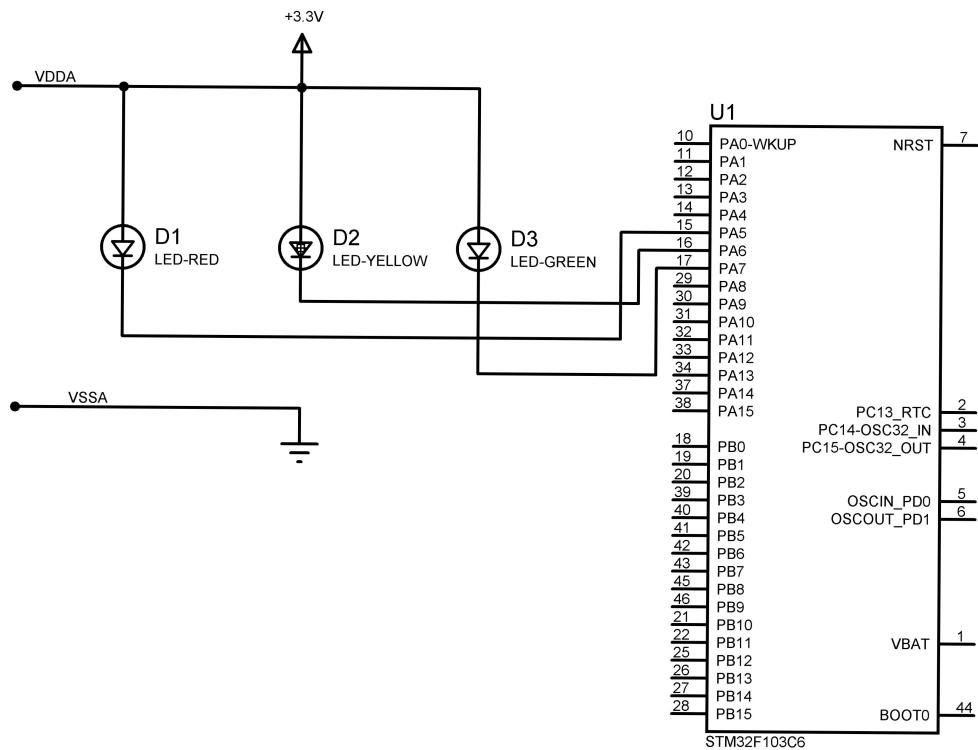


Figure 2.1: [Github Link](#) and [Back up GG Drive Link](#)



```
1  char color = 'N';
2  while (1)
3  {
4      if(color == 'N')
5      {
6          HAL_GPIO_TogglePin(Led_yellow_GPIO_Port, Led_yellow_Pin);
7          HAL_GPIO_TogglePin(Led_green_GPIO_Port, Led_green_Pin);
8          color = 'R';
9      }
10     else if(color == 'R')
11     {
12         HAL_Delay(5000);
13         HAL_GPIO_TogglePin(Led_red_GPIO_Port, Led_red_Pin);
14         HAL_GPIO_TogglePin(Led_green_GPIO_Port, Led_green_Pin);
15         color = 'G';
16     }
17     else if (color == 'Y')
18     {
19         HAL_Delay(2000);
20         HAL_GPIO_TogglePin(Led_yellow_GPIO_Port, Led_yellow_Pin);
21         HAL_GPIO_TogglePin(Led_red_GPIO_Port, Led_red_Pin);
22         color = 'R';
23     }
24     else if (color == 'G')
25     {
26         HAL_Delay(3000);
27         HAL_GPIO_TogglePin(Led_green_GPIO_Port, Led_green_Pin);
28         HAL_GPIO_TogglePin(Led_yellow_GPIO_Port, Led_yellow_Pin);
29         color = 'Y';
30     }
31     /* USER CODE END WHILE */
32
33     /* USER CODE BEGIN 3 */
34 }
```

Program 2.1: Ex2 source code

3 Exercise 3

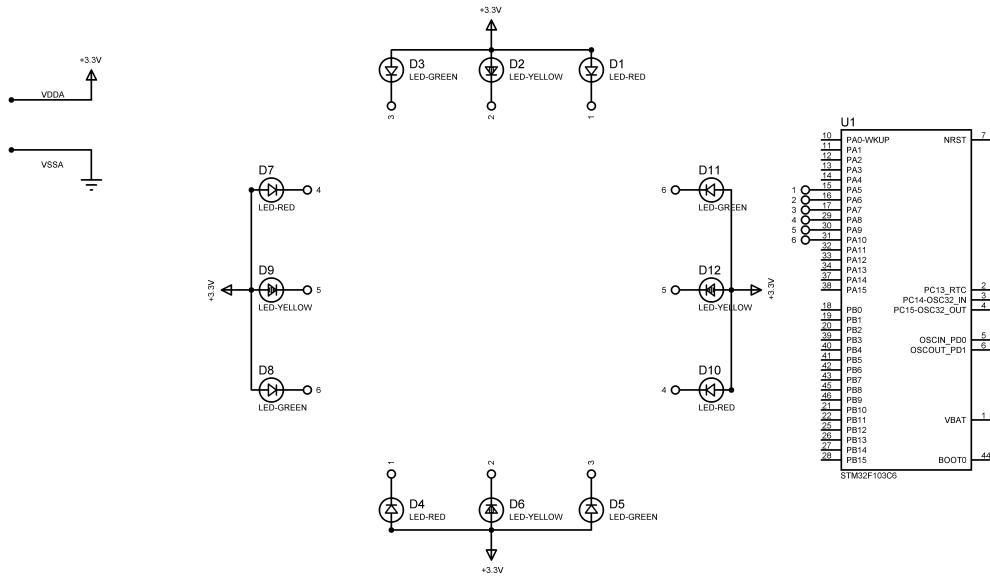


Figure 3.1: [Github Link](#) and [Back up GG Drive Link](#)



```
1  char color_x;
2  char color_y;
3  int first_time = 1;
4  while (1)
5  {
6      if(first_time == 1)
7      {
8          HAL_GPIO_TogglePin(led_yellow_vert_GPIO_Port,
9          led_yellow_vert_Pin);
10         HAL_GPIO_TogglePin(led_green_vert_GPIO_Port,
11         led_green_vert_Pin);
12         color_y = 'R';
13
14         HAL_GPIO_TogglePin(led_yellow_hori_GPIO_Port,
15         led_yellow_hori_Pin);
16         HAL_GPIO_TogglePin(led_red_hori_GPIO_Port, led_red_hori_Pin)
17 ;
18         color_x = 'G';
19
20         first_time = 0;
21     }
22     if(color_y == 'R')
23     {
24         HAL_Delay(3000);
25         color_x = led_toggle(led_red_hori_GPIO_Port,
26         led_red_hori_Pin, led_yellow_hori_GPIO_Port,
27         led_yellow_hori_Pin, led_green_hori_GPIO_Port,
28         led_green_hori_Pin, color_x);
29
30         HAL_Delay(2000);
31         color_x = led_toggle(led_red_hori_GPIO_Port,
32         led_red_hori_Pin, led_yellow_hori_GPIO_Port,
33         led_yellow_hori_Pin, led_green_hori_GPIO_Port,
34         led_green_hori_Pin, color_x);
35
36         color_y = led_toggle(led_red_vert_GPIO_Port,
37         led_red_vert_Pin, led_yellow_vert_GPIO_Port,
38         led_yellow_vert_Pin, led_green_vert_GPIO_Port,
39         led_green_vert_Pin, color_y);
40     }
41     else if (color_x == 'R')
42     {
43         HAL_Delay(3000);
44         color_y = led_toggle(led_red_vert_GPIO_Port,
45         led_red_vert_Pin, led_yellow_vert_GPIO_Port,
46         led_yellow_vert_Pin, led_green_vert_GPIO_Port,
47         led_green_vert_Pin, color_y);
48
49         HAL_Delay(2000);
50     }
51 }
```



```
34     color_y = led_toggle(led_red_vert_GPIO_Port,
35     led_red_vert_Pin, led_yellow_vert_GPIO_Port,
36     led_yellow_vert_Pin, led_green_vert_GPIO_Port,
37     led_green_vert_Pin, color_y);
38
39     color_x = led_toggle(led_red_hori_GPIO_Port,
40     led_red_hori_Pin, led_yellow_hori_GPIO_Port,
41     led_yellow_hori_Pin, led_green_hori_GPIO_Port,
42     led_green_hori_Pin, color_x);
43 }
44
45 /* USER CODE END WHILE */
46
47 /* USER CODE BEGIN 3 */
48 }
```

Program 3.1: Ex3 source code

```
1 char led_toggle(GPIO_TypeDef* red, uint16_t red_pin , GPIO_TypeDef*
2 * yellow, uint16_t yellow_pin, GPIO_TypeDef* green, uint16_t
3 green_pin, char color)
4 {
5     if(color == 'R')
6     {
7         HAL_GPIO_TogglePin(red, red_pin);
8         HAL_GPIO_TogglePin(green, green_pin);
9         color = 'G';
10    }
11    else if (color == 'Y')
12    {
13        HAL_GPIO_TogglePin(yellow, yellow_pin);
14        HAL_GPIO_TogglePin(red, red_pin);
15        color = 'R';
16    }
17    else if (color == 'G')
18    {
19        HAL_GPIO_TogglePin(green, green_pin);
20        HAL_GPIO_TogglePin(yellow, yellow_pin);
21        color = 'Y';
22    }
23
24 }
```

Program 3.2: led_toggle() function

4 Exercise 4

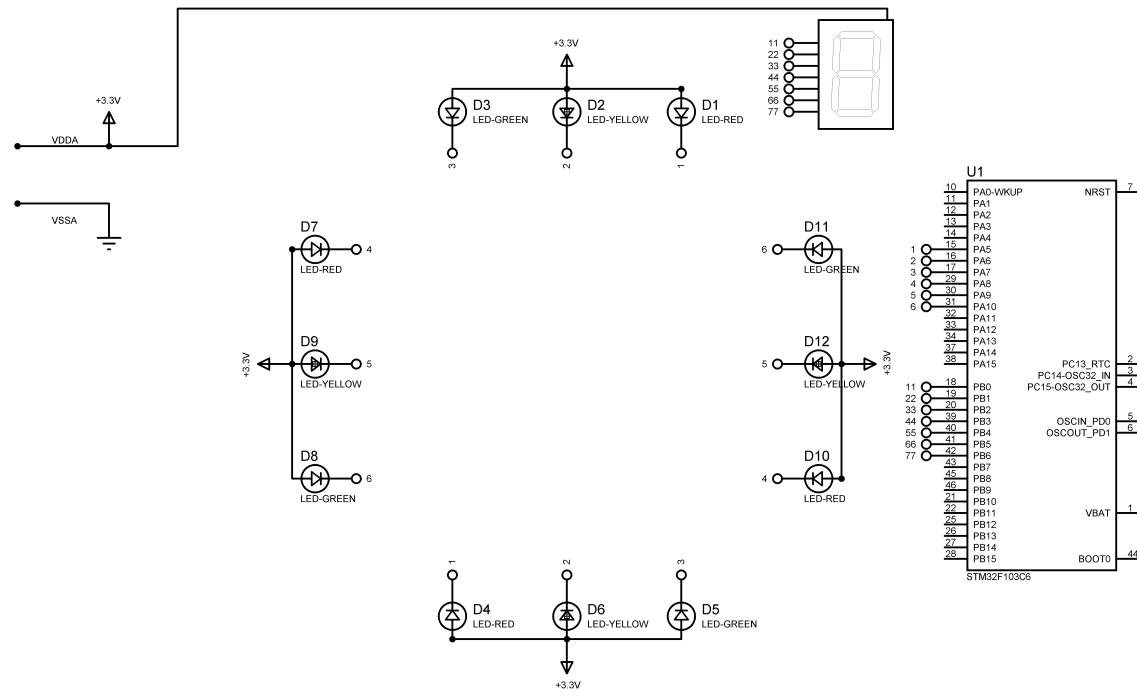


Figure 4.1: [Github Link](#) and [Back up GG Drive Link](#)



```
1 void display7SEG(int num)
2 {
3     switch (num) {
4         case 0:
5             HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
6             HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
7             HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
8             HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
9             HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
10            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
11            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, SET);
12            break;
13        case 1:
14            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, SET);
15            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
16            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
17            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, SET);
18            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
19            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
20            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, SET);
21            break;
22        case 2:
23            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
24            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
25            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, SET);
26            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
27            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
28            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
29            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
30            break;
31        case 3:
32            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
33            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
34            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
35            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
36            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
37            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
38            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
39            break;
40        case 4:
41            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, SET);
42            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
43            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
44            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, SET);
45            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
46            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
47            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
48            break;
49        case 5:
```



```
50     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
51     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, SET);
52     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
53     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
54     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
55     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
56     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
57     break;
58 case 6:
59     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
60     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, SET);
61     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
62     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
63     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
64     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
65     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
66     break;
67 case 7:
68     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
69     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
70     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
71     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, SET);
72     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
73     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
74     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, SET);
75     break;
76 case 8:
77     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
78     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
79     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
80     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
81     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
82     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
83     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
84     break;
85 case 9:
86     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
87     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
88     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
89     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
90     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
91     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
92     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
93     break;
94 default:
95     break;
96 }
97 }
```

Program 4.1: **display7SEG()** function

5 Exercise 5

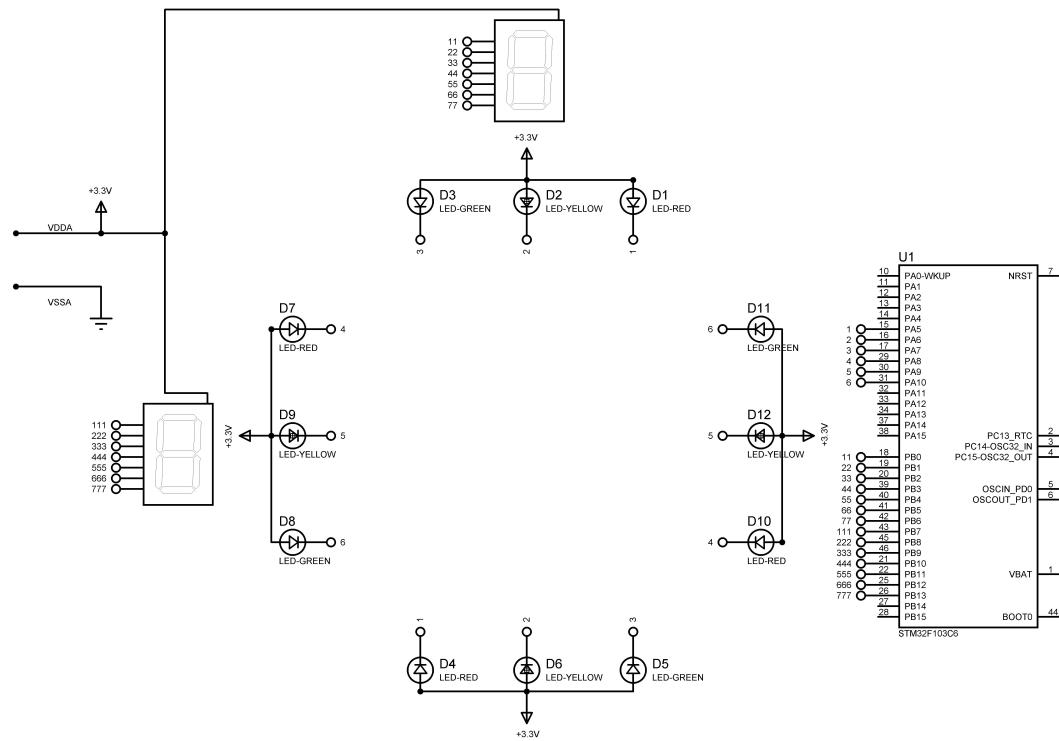


Figure 5.1: [Github Link](#) and [Back up GG Drive Link](#)



I did a little modified to the **display7SEG()** function as the IDE keep give me error and I did not know how to fix

```
1 void display7SEG(int num,
2     GPIO_TypeDef* a_seg_GPIO_Port, uint16_t a_seg_Pin,
3     GPIO_TypeDef* b_seg_GPIO_Port, uint16_t b_seg_Pin,
4     GPIO_TypeDef* c_seg_GPIO_Port, uint16_t c_seg_Pin,
5     GPIO_TypeDef* d_seg_GPIO_Port, uint16_t d_seg_Pin,
6     GPIO_TypeDef* e_seg_GPIO_Port, uint16_t e_seg_Pin,
7     GPIO_TypeDef* f_seg_GPIO_Port, uint16_t f_seg_Pin,
8     GPIO_TypeDef* g_seg_GPIO_Port, uint16_t g_seg_Pin)
9 {
10    switch (num) {
11        case 0:
12            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
13            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
14            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
15            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
16            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
17            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
18            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, SET);
19            break;
20        case 1:
21            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, SET);
22            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
23            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
24            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, SET);
25            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
26            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
27            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, SET);
28            break;
29        case 2:
30            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
31            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
32            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, SET);
33            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
34            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
35            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
36            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
37            break;
38        case 3:
39            HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
40            HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
41            HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
42            HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
43            HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
44            HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
45            HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
46            break;
47        case 4:
```



```
48     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, SET);
49     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
50     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
51     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, SET);
52     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
53     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
54     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
55     break;
56 case 5:
57     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
58     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, SET);
59     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
60     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
61     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
62     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
63     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
64     break;
65 case 6:
66     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
67     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, SET);
68     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
69     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
70     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
71     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
72     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
73     break;
74 case 7:
75     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
76     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
77     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
78     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, SET);
79     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
80     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, SET);
81     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, SET);
82     break;
83 case 8:
84     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
85     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
86     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
87     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
88     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, RESET);
89     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
90     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
91     break;
92 case 9:
93     HAL_GPIO_WritePin(a_seg_GPIO_Port, a_seg_Pin, RESET);
94     HAL_GPIO_WritePin(b_seg_GPIO_Port, b_seg_Pin, RESET);
95     HAL_GPIO_WritePin(c_seg_GPIO_Port, c_seg_Pin, RESET);
96     HAL_GPIO_WritePin(d_seg_GPIO_Port, d_seg_Pin, RESET);
```



```
97     HAL_GPIO_WritePin(e_seg_GPIO_Port, e_seg_Pin, SET);
98     HAL_GPIO_WritePin(f_seg_GPIO_Port, f_seg_Pin, RESET);
99     HAL_GPIO_WritePin(g_seg_GPIO_Port, g_seg_Pin, RESET);
100    break;
101 default:
102    break;
103 }
104 }
105
106 char led_toggle(GPIO_TypeDef* red, uint16_t red_pin ,
107                  GPIO_TypeDef* yellow, uint16_t yellow_pin,
108                  GPIO_TypeDef* green, uint16_t green_pin,
109                  char color)
110 {
111     if(color == 'R')
112     {
113         HAL_GPIO_TogglePin(red, red_pin);
114         HAL_GPIO_TogglePin(green, green_pin);
115         color = 'G';
116     }
117     else if (color == 'Y')
118     {
119         HAL_GPIO_TogglePin(yellow, yellow_pin);
120         HAL_GPIO_TogglePin(red, red_pin);
121         color = 'R';
122     }
123     else if (color == 'G')
124     {
125         HAL_GPIO_TogglePin(green, green_pin);
126         HAL_GPIO_TogglePin(yellow, yellow_pin);
127         color = 'Y';
128     }
129     return color;
130 }
```

Program 5.1: **display7SEG()** and **led_toggle** function

```
1 int first_time_counter = 1;
2 int counter_x = 0;
3 int counter_y = 0;
4 char color_x;
5 char color_y;
6 int first_time_light = 1;
7 while (1)
8 {
9     //Counter set
10    if(first_time_counter == 1)
11    {
12        counter_x = 9;
13        display7SEG(counter_y--,
```



```
14         a_v_seg_GPIO_Port, a_v_seg_Pin,
15         b_v_seg_GPIO_Port, b_v_seg_Pin,
16         c_v_seg_GPIO_Port, c_v_seg_Pin,
17         d_v_seg_GPIO_Port, d_v_seg_Pin,
18         e_v_seg_GPIO_Port, e_v_seg_Pin,
19         f_v_seg_GPIO_Port, f_v_seg_Pin,
20         g_v_seg_GPIO_Port, g_v_seg_Pin
21     );
22
23     counter_y = 6;
24     display7SEG(counter_x--,
25                 a_h_seg_GPIO_Port, a_h_seg_Pin,
26                 b_h_seg_GPIO_Port, b_h_seg_Pin,
27                 c_h_seg_GPIO_Port, c_h_seg_Pin,
28                 d_h_seg_GPIO_Port, d_h_seg_Pin,
29                 e_h_seg_GPIO_Port, e_h_seg_Pin,
30                 f_h_seg_GPIO_Port, f_h_seg_Pin,
31                 g_h_seg_GPIO_Port, g_h_seg_Pin
32     );
33     first_time_counter = 0;
34 }
35 //Light
36 if(first_time_light == 1)
37 {
38     HAL_GPIO_TogglePin(led_yellow_vert_GPIO_Port,
39                         led_yellow_vert_Pin);
40     HAL_GPIO_TogglePin(led_green_vert_GPIO_Port,
41                         led_green_vert_Pin);
42     color_y = 'R';
43
44     HAL_GPIO_TogglePin(led_yellow_hori_GPIO_Port,
45                         led_yellow_hori_Pin);
46     HAL_GPIO_TogglePin(led_red_hori_GPIO_Port, led_red_hori_Pin)
47 ;
48     color_x = 'G';
49
50     first_time_light = 0;
51 }
52 if(color_y == 'R')
53 {
54     counter_y = 9;
55     counter_x = 6;
56     for (int i = 0; i < 7; i++)
57     {
58         display7SEG(counter_y--,
59                     a_v_seg_GPIO_Port, a_v_seg_Pin,
60                     b_v_seg_GPIO_Port, b_v_seg_Pin,
61                     c_v_seg_GPIO_Port, c_v_seg_Pin,
62                     d_v_seg_GPIO_Port, d_v_seg_Pin,
```



```
59         e_v_seg_GPIO_Port, e_v_seg_Pin,
60         f_v_seg_GPIO_Port, f_v_seg_Pin,
61         g_v_seg_GPIO_Port, g_v_seg_Pin
62     );
63     display7SEG(counter_x--,
64         a_h_seg_GPIO_Port, a_h_seg_Pin,
65         b_h_seg_GPIO_Port, b_h_seg_Pin,
66         c_h_seg_GPIO_Port, c_h_seg_Pin,
67         d_h_seg_GPIO_Port, d_h_seg_Pin,
68         e_h_seg_GPIO_Port, e_h_seg_Pin,
69         f_h_seg_GPIO_Port, f_h_seg_Pin,
70         g_h_seg_GPIO_Port, g_h_seg_Pin
71     );
72     HAL_Delay(1000);
73 }
74 color_x = led_toggle(
75     led_red_hori_GPIO_Port, led_red_hori_Pin,
76     led_yellow_hori_GPIO_Port, led_yellow_hori_Pin,
77     led_green_hori_GPIO_Port, led_green_hori_Pin,
78     color_x);
79 counter_x = 2;
80 for (int i = 0; i < 3; i++)
81 {
82     display7SEG(counter_y--,
83         a_v_seg_GPIO_Port, a_v_seg_Pin,
84         b_v_seg_GPIO_Port, b_v_seg_Pin,
85         c_v_seg_GPIO_Port, c_v_seg_Pin,
86         d_v_seg_GPIO_Port, d_v_seg_Pin,
87         e_v_seg_GPIO_Port, e_v_seg_Pin,
88         f_v_seg_GPIO_Port, f_v_seg_Pin,
89         g_v_seg_GPIO_Port, g_v_seg_Pin
90     );
91     display7SEG(counter_x--,
92         a_h_seg_GPIO_Port, a_h_seg_Pin,
93         b_h_seg_GPIO_Port, b_h_seg_Pin,
94         c_h_seg_GPIO_Port, c_h_seg_Pin,
95         d_h_seg_GPIO_Port, d_h_seg_Pin,
96         e_h_seg_GPIO_Port, e_h_seg_Pin,
97         f_h_seg_GPIO_Port, f_h_seg_Pin,
98         g_h_seg_GPIO_Port, g_h_seg_Pin
99     );
100    HAL_Delay(1000);
101 }
102 color_x = led_toggle(
103     led_red_hori_GPIO_Port, led_red_hori_Pin,
104     led_yellow_hori_GPIO_Port, led_yellow_hori_Pin,
105     led_green_hori_GPIO_Port, led_green_hori_Pin,
106     color_x);
107 color_y = led_toggle(
```



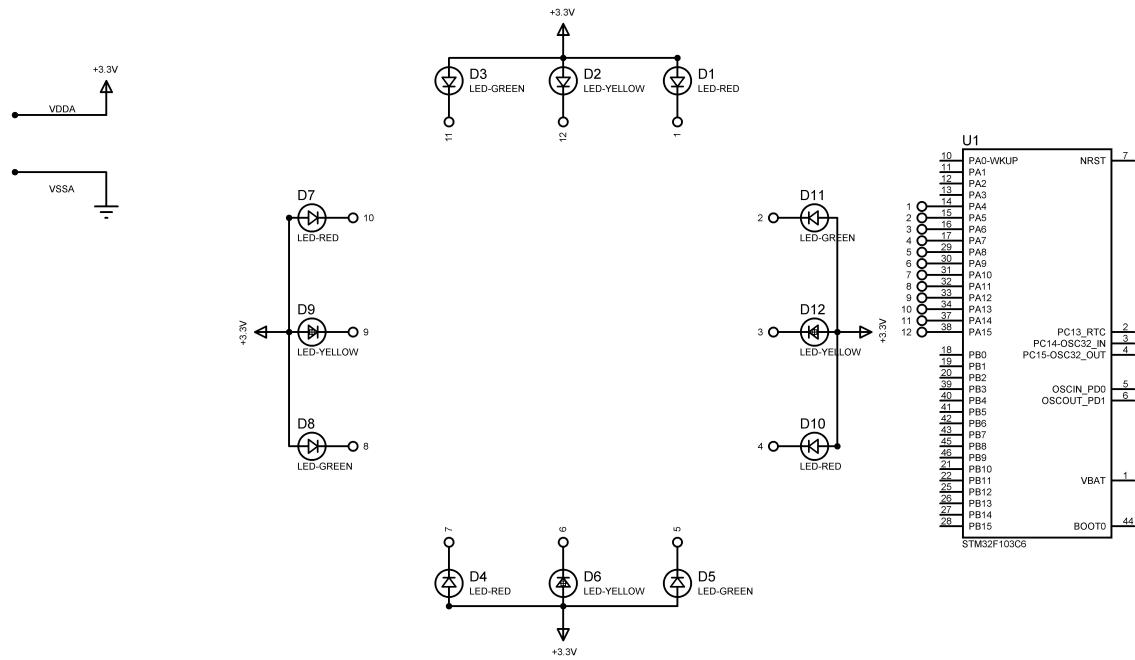
```
108     led_red_vert_GPIO_Port, led_red_vert_Pin,
109     led_yellow_vert_GPIO_Port, led_yellow_vert_Pin,
110     led_green_vert_GPIO_Port, led_green_vert_Pin,
111     color_y);
112 }
113 else if (color_x == 'R')
114 {
115     counter_x = 9;
116     counter_y = 6;
117     for (int i = 0; i < 7; i++)
118     {
119         display7SEG(counter_y--,
120             a_v_seg_GPIO_Port, a_v_seg_Pin,
121             b_v_seg_GPIO_Port, b_v_seg_Pin,
122             c_v_seg_GPIO_Port, c_v_seg_Pin,
123             d_v_seg_GPIO_Port, d_v_seg_Pin,
124             e_v_seg_GPIO_Port, e_v_seg_Pin,
125             f_v_seg_GPIO_Port, f_v_seg_Pin,
126             g_v_seg_GPIO_Port, g_v_seg_Pin
127         );
128         display7SEG(counter_x--,
129             a_h_seg_GPIO_Port, a_h_seg_Pin,
130             b_h_seg_GPIO_Port, b_h_seg_Pin,
131             c_h_seg_GPIO_Port, c_h_seg_Pin,
132             d_h_seg_GPIO_Port, d_h_seg_Pin,
133             e_h_seg_GPIO_Port, e_h_seg_Pin,
134             f_h_seg_GPIO_Port, f_h_seg_Pin,
135             g_h_seg_GPIO_Port, g_h_seg_Pin
136         );
137         HAL_Delay(1000);
138     }
139     color_y = led_toggle(
140         led_red_vert_GPIO_Port, led_red_vert_Pin,
141         led_yellow_vert_GPIO_Port, led_yellow_vert_Pin,
142         led_green_vert_GPIO_Port, led_green_vert_Pin,
143         color_y);
144     counter_y = 2;
145     for (int i = 0; i < 3; i++)
146     {
147         display7SEG(counter_y--,
148             a_v_seg_GPIO_Port, a_v_seg_Pin,
149             b_v_seg_GPIO_Port, b_v_seg_Pin,
150             c_v_seg_GPIO_Port, c_v_seg_Pin,
151             d_v_seg_GPIO_Port, d_v_seg_Pin,
152             e_v_seg_GPIO_Port, e_v_seg_Pin,
153             f_v_seg_GPIO_Port, f_v_seg_Pin,
154             g_v_seg_GPIO_Port, g_v_seg_Pin
155         );
156         display7SEG(counter_x--,
```



```
157         a_h_seg_GPIO_Port, a_h_seg_Pin,
158         b_h_seg_GPIO_Port, b_h_seg_Pin,
159         c_h_seg_GPIO_Port, c_h_seg_Pin,
160         d_h_seg_GPIO_Port, d_h_seg_Pin,
161         e_h_seg_GPIO_Port, e_h_seg_Pin,
162         f_h_seg_GPIO_Port, f_h_seg_Pin,
163         g_h_seg_GPIO_Port, g_h_seg_Pin
164     );
165     HAL_Delay(1000);
166 }
167 color_y = led_toggle(
168     led_red_vert_GPIO_Port, led_red_vert_Pin,
169     led_yellow_vert_GPIO_Port, led_yellow_vert_Pin,
170     led_green_vert_GPIO_Port, led_green_vert_Pin,
171     color_y);
172 color_x = led_toggle(
173     led_red_hori_GPIO_Port, led_red_hori_Pin,
174     led_yellow_hori_GPIO_Port, led_yellow_hori_Pin,
175     led_green_hori_GPIO_Port, led_green_hori_Pin,
176     color_x);
177 }
178 /* USER CODE END WHILE */
179 /* USER CODE BEGIN 3 */
180 }
181 }
```

Program 5.2: Ex5 source code

6 Exercise 6, 7, 8, 9, 10





```
1 void clearAllClock()
2 {
3     HAL_GPIO_WritePin(clk_I_GPIO_Port, clk_I_Pin, RESET);
4     HAL_GPIO_WritePin(clk_II_GPIO_Port, clk_II_Pin, RESET);
5     HAL_GPIO_WritePin(clk_III_GPIO_Port, clk_III_Pin, RESET);
6     HAL_GPIO_WritePin(clk_IV_GPIO_Port, clk_IV_Pin, RESET);
7     HAL_GPIO_WritePin(clk_V_GPIO_Port, clk_V_Pin, RESET);
8     HAL_GPIO_WritePin(clk_VI_GPIO_Port, clk_VI_Pin, RESET);
9     HAL_GPIO_WritePin(clk_VII_GPIO_Port, clk_VII_Pin, RESET);
10    HAL_GPIO_WritePin(clk_VIII_GPIO_Port, clk_VIII_Pin, RESET);
11    HAL_GPIO_WritePin(clk_IX_GPIO_Port, clk_IX_Pin, RESET);
12    HAL_GPIO_WritePin(clk_X_GPIO_Port, clk_X_Pin, RESET);
13    HAL_GPIO_WritePin(clk_XI_GPIO_Port, clk_XI_Pin, RESET);
14    HAL_GPIO_WritePin(clk_XII_GPIO_Port, clk_XII_Pin, RESET);
15 }
16
17 void setNumberOnClock(int num)
18 {
19     if(num == 0)
20     {
21         num = 12;
22     }
23     switch (num) {
24         case 1:
25             HAL_GPIO_WritePin(clk_I_GPIO_Port, clk_I_Pin, SET);
26             break;
27         case 2:
28             HAL_GPIO_WritePin(clk_II_GPIO_Port, clk_II_Pin, SET);
29             break;
30         case 3:
31             HAL_GPIO_WritePin(clk_III_GPIO_Port, clk_III_Pin, SET);
32             break;
33         case 4:
34             HAL_GPIO_WritePin(clk_IV_GPIO_Port, clk_IV_Pin, SET);
35             break;
36         case 5:
37             HAL_GPIO_WritePin(clk_V_GPIO_Port, clk_V_Pin, SET);
38             break;
39         case 6:
40             HAL_GPIO_WritePin(clk_VI_GPIO_Port, clk_VI_Pin, SET);
41             break;
42         case 7:
43             HAL_GPIO_WritePin(clk_VII_GPIO_Port, clk_VII_Pin, SET);
44             break;
45         case 8:
46             HAL_GPIO_WritePin(clk_VIII_GPIO_Port, clk_VIII_Pin, SET);
47             break;
48         case 9:
49             HAL_GPIO_WritePin(clk_IX_GPIO_Port, clk_IX_Pin, SET);
```



```
50     break;
51 case 10:
52     HAL_GPIO_WritePin(clk_X_GPIO_Port, clk_X_Pin, SET);
53     break;
54 case 11:
55     HAL_GPIO_WritePin(clk_XI_GPIO_Port, clk_XI_Pin, SET);
56     break;
57 case 12:
58     HAL_GPIO_WritePin(clk_XII_GPIO_Port, clk_XII_Pin, SET);
59     break;
60 default:
61     break;
62 }
63 }

64 void clearNumberOnClock(int num)
65 {
66     if(num == 0)
67     {
68         num = 12;
69     }
70     switch (num) {
71     case 1:
72         HAL_GPIO_WritePin(clk_I_GPIO_Port, clk_I_Pin, RESET);
73         break;
74     case 2:
75         HAL_GPIO_WritePin(clk_II_GPIO_Port, clk_II_Pin, RESET);
76         break;
77     case 3:
78         HAL_GPIO_WritePin(clk_III_GPIO_Port, clk_III_Pin, RESET);
79         break;
80     case 4:
81         HAL_GPIO_WritePin(clk_IV_GPIO_Port, clk_IV_Pin, RESET);
82         break;
83     case 5:
84         HAL_GPIO_WritePin(clk_V_GPIO_Port, clk_V_Pin, RESET);
85         break;
86     case 6:
87         HAL_GPIO_WritePin(clk_VI_GPIO_Port, clk_VI_Pin, RESET);
88         break;
89     case 7:
90         HAL_GPIO_WritePin(clk_VII_GPIO_Port, clk_VII_Pin, RESET);
91         break;
92     case 8:
93         HAL_GPIO_WritePin(clk_VIII_GPIO_Port, clk_VIII_Pin, RESET);
94         break;
95     case 9:
96         HAL_GPIO_WritePin(clk_IX_GPIO_Port, clk_IX_Pin, RESET);
97         break;
```



```
99     case 10:
100        HAL_GPIO_WritePin(clk_X_GPIO_Port, clk_X_Pin, RESET);
101        break;
102    case 11:
103        HAL_GPIO_WritePin(clk_XI_GPIO_Port, clk_XI_Pin, RESET);
104        break;
105    case 12:
106        HAL_GPIO_WritePin(clk_XII_GPIO_Port, clk_XII_Pin, RESET);
107        break;
108    default:
109        break;
110    }
111}
112}
```

Program 6.1: Ex7 and Ex8 and Ex9 function

```
1   while (1)
2   {
3       HAL_GPIO_TogglePin(clk_XII_GPIO_Port, clk_XII_Pin);
4       HAL_Delay(1000);
5       HAL_GPIO_TogglePin(clk_I_GPIO_Port, clk_I_Pin);
6       HAL_Delay(1000);
7       HAL_GPIO_TogglePin(clk_II_GPIO_Port, clk_II_Pin);
8       HAL_Delay(1000);
9       HAL_GPIO_TogglePin(clk_III_GPIO_Port, clk_III_Pin);
10      HAL_Delay(1000);
11      HAL_GPIO_TogglePin(clk_IV_GPIO_Port, clk_IV_Pin);
12      HAL_Delay(1000);
13      HAL_GPIO_TogglePin(clk_V_GPIO_Port, clk_V_Pin);
14      HAL_Delay(1000);
15      HAL_GPIO_TogglePin(clk_VI_GPIO_Port, clk_VI_Pin);
16      HAL_Delay(1000);
17      HAL_GPIO_TogglePin(clk_VII_GPIO_Port, clk_VII_Pin);
18      HAL_Delay(1000);
19      HAL_GPIO_TogglePin(clk_VIII_GPIO_Port, clk_VIII_Pin);
20      HAL_Delay(1000);
21      HAL_GPIO_TogglePin(clk_IX_GPIO_Port, clk_IX_Pin);
22      HAL_Delay(1000);
23      HAL_GPIO_TogglePin(clk_X_GPIO_Port, clk_X_Pin);
24      HAL_Delay(1000);
25      HAL_GPIO_TogglePin(clk_XI_GPIO_Port, clk_XI_Pin);
26      HAL_Delay(1000);
27      /* USER CODE END WHILE */
28
29      /* USER CODE BEGIN 3 */
30  }
```

Program 6.2: Ex6 source code



```
1 int hour = -1;
2 int minute = -1;
3 int second = -1;
4 int first_time = 1;
5 int index_sec = -1;
6 int index_min = -1;
7 while (1)
{
8     if(first_time == 1)
9     {
10         first_time = 0;
11         hour = 3;
12         minute = 25;
13         second = 45;
14         index_min = 0;
15         index_sec = 0;
16         clearAllClock();
17     }
18     initialize_hms(&hour, &minute, &second);
19
20 //hour
21 if(hour > 12)
22 {
23     hour = 1;
24 }
25 setNumberOnClock(hour);
26
27 if(hour - 1 != index_sec && hour - 1 != index_min)
28 {
29     clearNumberOnClock(hour -1);
30 }
31
32 //minute
33 if(minute > 60)
34 {
35     minute = 1;
36     index_min = 0;
37 }
38 while(minute < 60)
39 {
40     //second
41     if(second > 60)
42     {
43         second = 1;
44         index_sec = 0;
45     }
46     if(minute % 5 == 0)
47     {
48         index_min = minute/5;
```



```
50     setNumberOnClock(index_min);
51     if(index_min - 1 != index_sec && index_min - 1 != hour)
52     {
53         clearNumberOnClock(index_min - 1);
54     }
55 }
56 while(second <= 60)
57 {
58     if(second % 5 == 0)
59     {
60         index_sec = second/5;
61         setNumberOnClock(index_sec);
62         if(index_sec - 1 != index_min && index_sec - 1 != hour)
63         {
64             clearNumberOnClock(index_sec - 1);
65         }
66     }
67     HAL_Delay(1000);
68     second++;
69 }
70 minute++;
71 }
72 hour++;
73 /* USER CODE END WHILE */
74
75 /* USER CODE BEGIN 3 */
76 }
```

Program 6.3: Ex10 source code

```
1 void clearAllClock()
2 {
3     for(int i = 0; i < 12 ; i++)
4     {
5         clearNumberOnClock(i);
6     }
7 }
```

Program 6.4: Bonus **clearAllClock** function in Ex10