

VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY  
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY  
Faculty of Computer Science and Engineering



---

CC02 — Lab Report

# Microprocessor - Microcontroller Lab 3

---

**Supervisors:** Nguyen Thien An  
**Students:** Vu Trinh Thanh Binh 2252085

Ho Chi Minh City, November 6, 2024



## Contents

<b>1</b>	<b>Exercise</b>	<b>2</b>
1.1	Exercise 1 . . . . .	3
1.2	Exercise 2 . . . . .	3
1.3	Exercise 3 . . . . .	4
1.4	Exercise 4 . . . . .	4
1.4.1	Key Equations: . . . . .	4
1.5	Exercise 5 . . . . .	5
1.5.1	Report . . . . .	5
1.6	Exercise 6 . . . . .	6
1.6.1	7SEGMENT-LED source code . . . . .	6
1.6.2	Trafficlight source code . . . . .	11
1.7	Exercise 7 . . . . .	12
1.8	Exercise 8 . . . . .	14
1.9	Exercise 9 . . . . .	15
1.10	Exercise 10 . . . . .	16
	<b>References</b>	<b>17</b>



## 1 Exercise

The GitHub link for the lab schematics is at [here](#) or in this link: <https://github.com/thanhbinh0710/VXL.git>.

## 1.1 Exercise 1

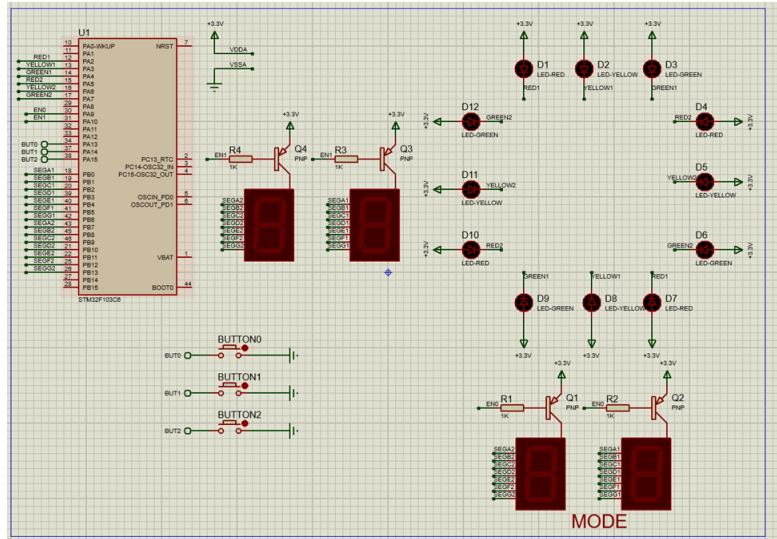
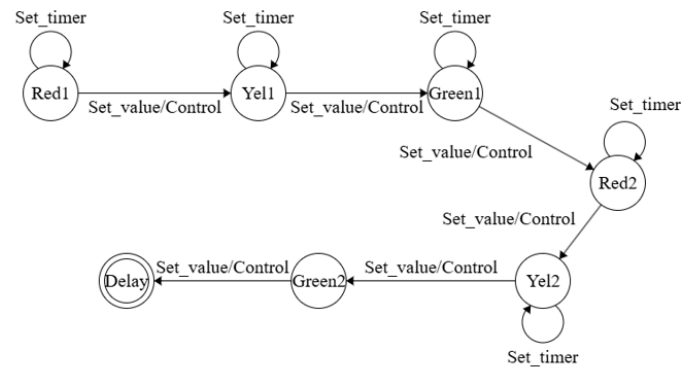


Figure 1: Proteus

## 1.2 Exercise 2



\*Where:

Control = Button 1 – use for choosing mode

Set\_timer = Button 2 – use for increasing time

Set\_value = Button 3 – use for accepting the current time and move to next mode

Figure 2: Proteus

### 1.3 Exercise 3

- Already include in Github.

### 1.4 Exercise 4

- To maintain a stable LED blinking frequency of 2 Hz, regardless of changes in the timer interrupt duration, by modifying the `MAX_COUNTER` parameter. This approach avoids floating-point calculations for accuracy.

#### 1.4.1 Key Equations:

1. **Basic Timer Duration Equation:**  $T = 1/f$

2. **Timer Duration with Prescaler and Counter Period:** Given a new timer frequency  $f'$  after downscaling:

$$T_0 = \frac{(\text{prescaler} + 1) \times (\text{counterPeriod} + 1)}{f}$$

Figure 3

3. **Calculation of MAXCOUNTER:** To maintain the LED blinking at 2 Hz:

$$\text{MAX\_COUNTER} = \frac{\text{time\_duration\_led}}{T_0}$$

Figure 4



## 1.5 Exercise 5

### 1.5.1 Report

```
1 #include "input_reading.h"
2
3 int Max_press = 500;
4 int Max_hold = 500;
5 int button_flag[NO_OF_BUTTONS];
6
7 //define 4-stages buffer for debouncing
8 static GPIO_PinState debounceButtonBuffer1[NO_OF_BUTTONS];
9 static GPIO_PinState debounceButtonBuffer2[NO_OF_BUTTONS];
10 static GPIO_PinState debounceButtonBuffer3[NO_OF_BUTTONS];
11 static GPIO_PinState debounceButtonBuffer4[NO_OF_BUTTONS];
12
13 static uint8_t press1s_flag[NO_OF_BUTTONS];
14 static uint16_t press1s_counter[NO_OF_BUTTONS];
15
16 #define INPUT_PORT GPIOA
17 uint16_t BUTTON_PIN[NO_OF_BUTTONS] = {0x2000, 0x4000, 0x6000};
18 }
19 void setButton(){
20     for (unsigned char i = 0; i < NO_OF_BUTTONS; i++){
21         debounceButtonBuffer1[i] = BUTTON_IS_RELEASED;
22         debounceButtonBuffer2[i] = BUTTON_IS_RELEASED;
23         debounceButtonBuffer3[i] = BUTTON_IS_RELEASED;
24         debounceButtonBuffer4[i] = BUTTON_IS_RELEASED;
25         button_flag[i] = BUTTON_FLAG_RESET;
26         press1s_flag[i] = BUTTON_FLAG_RESET;
27         press1s_counter[i] = 0;
28     }
29 }
30
31 void readButton(void){
32     for (unsigned char i = 0; i < NO_OF_BUTTONS; i++){
33         debounceButtonBuffer3[i] = debounceButtonBuffer2[i];
34         debounceButtonBuffer2[i] = debounceButtonBuffer1[i];
35         debounceButtonBuffer1[i] = HAL_GPIO_ReadPin(INPUT_PORT, BUTTON_PIN[i]);
36         if ((debounceButtonBuffer2[i] == debounceButtonBuffer1[i]) && (
37             debounceButtonBuffer2[i] == debounceButtonBuffer3[i])){
38             if (debounceButtonBuffer3[i] != debounceButtonBuffer4[i]){
39                 debounceButtonBuffer4[i] = debounceButtonBuffer3[i];
40                 if (debounceButtonBuffer4[i] == BUTTON_IS_PRESSED){
41                     Max_press = 500;
42                     button_flag[i] = BUTTON_FLAG_SET;
43                 }else{
44                     Max_press--;
45                     press1s_counter[i] = 0;
46                     debounceButtonBuffer4[i] = BUTTON_IS_RELEASED;
47                 }
48             }
49         }
50     }
51 }
```

```
16     }
17     }else{
18         if (press1s_counter[i] < DURATION_FOR_AUTO_INCREASING){
19             press1s_counter[i]++;
20         }else press1s_flag[i] = BUTTON_FLAG_SET;
21     }
22 }
23 }
24 }
25 unsigned char Press_button(unsigned char index){
26     if (index >= NO_OF_BUTTONS) return 0;
27     if (button_flag[index] == BUTTON_FLAG_SET){
28         button_flag[index] = BUTTON_FLAG_RESET;
29         return 1;
30     }else return 0;
31 }
32
33 unsigned char Press1s_button(unsigned char index){
34     if (index >= NO_OF_BUTTONS) return 0;
35     if (press1s_flag[index] == BUTTON_FLAG_SET){
36         press1s_flag[index] = BUTTON_FLAG_RESET;
37         press1s_counter[index] = 0;
38         return 1;
39     }else return 0;
40 }
```

## 1.6 Exercise 6

### 1.6.1 7SEGMENT-LED source code

```
1     #include "segment_display.h"
2     #include "global.h"
3
4     void display7SEG1 (int num){
5         switch (num){
6             case 0:
7                 HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
8                 HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
9                 HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
10                HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, RESET);
11                HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, RESET);
12                HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, RESET);
13                HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET);
14                break;
15            case 1:
16                HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, SET);
17                HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
18                HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
19                HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET);
```

```
20     HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
21     HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET);
22     HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET);
23     break;
24     case 2:
25         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
26         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
27         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, SET);
28         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, RESET);
29         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, RESET);
30         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET);
31         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, RESET);
32         break;
33     case 3:
34         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
35         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
36         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
37         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, RESET);
38         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
39         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET);
40         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, RESET);
41         break;
42     case 4:
43         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, SET);
44         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
45         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
46         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET);
47         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
48         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, RESET);
49         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, RESET);
50         break;
51     case 5:
52         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
53         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, SET);
54         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
55         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, RESET);
56         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
57         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, RESET);
58         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, RESET);
59         break;
60     case 6:
61         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
62         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, SET);
63         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
64         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, RESET);
65         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, RESET);
66         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, RESET);
67         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, RESET);
68         break;
```



```
69     case 7:
70         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
71         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
72         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
73         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET);
74         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
75         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET);
76         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET);
77         break;
78     case 8:
79         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
80         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
81         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
82         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, RESET);
83         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, RESET);
84         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, RESET);
85         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, RESET);
86         break;
87     case 9:
88         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, RESET);
89         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, RESET);
90         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, RESET);
91         HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, RESET);
92         HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
93         HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, RESET);
94         HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, RESET);
95         break;
96     default:
97         HAL_GPIO_WritePin(SEGA1_GPIO_Port, SEGA1_Pin, SET);
98         HAL_GPIO_WritePin(SEGB1_GPIO_Port, SEGB1_Pin, SET);
99         HAL_GPIO_WritePin(SEGC1_GPIO_Port, SEGC1_Pin, SET);
100        HAL_GPIO_WritePin(SEGD1_GPIO_Port, SEGD1_Pin, SET);
101        HAL_GPIO_WritePin(SEGE1_GPIO_Port, SEGE1_Pin, SET);
102        HAL_GPIO_WritePin(SEGF1_GPIO_Port, SEGF1_Pin, SET);
103        HAL_GPIO_WritePin(SEGG1_GPIO_Port, SEGG1_Pin, SET);
104        break;
105    }
106 }
107 void display7SEG2(int num) {
108     switch (num){
109     case 0:
110         HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, RESET);
111         HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, RESET);
112         HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
113         HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, RESET);
114         HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, RESET);
115         HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, RESET);
116         HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, SET);
117         break;
```

```
118     case 1:
119         HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, SET);
120         HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, RESET);
121         HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
122         HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, SET);
123         HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, SET);
124         HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, SET);
125         HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, SET);
126         break;
127     case 2:
128         HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, RESET);
129         HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, RESET);
130         HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGD2_Pin, SET);
131         HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, RESET);
132         HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, RESET);
133         HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, SET);
134         HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, RESET);
135         break;
136     case 3:
137         HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA1_Pin, RESET);
138         HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, RESET);
139         HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
140         HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, RESET);
141         HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, SET);
142         HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, SET);
143         HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, RESET);
144         break;
145     case 4:
146         HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, SET);
147         HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, SET);
148         HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
149         HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, SET);
150         HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, SET);
151         HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, RESET);
152         HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, RESET);
153         break;
154     case 5:
155         HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, RESET);
156         HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, SET);
157         HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
158         HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, RESET);
159         HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, SET);
160         HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, RESET);
161         HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, RESET);
162         break;
163     case 6:
164         HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, RESET);
165         HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, SET);
166         HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
```



```
167     HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, RESET);
168     HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, RESET);
169     HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, RESET);
170     HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, RESET);
171     break;
172 case 7:
173     HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, RESET);
174     HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, RESET);
175     HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
176     HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, SET);
177     HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, SET);
178     HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, SET);
179     HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, SET);
180     break;
181 case 8:
182     HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, RESET);
183     HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, RESET);
184     HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
185     HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, RESET);
186     HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, RESET);
187     HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, RESET);
188     HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, RESET);
189     break;
190 case 9:
191     HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, RESET);
192     HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, RESET);
193     HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, RESET);
194     HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, RESET);
195     HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, SET);
196     HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, RESET);
197     HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, RESET);
198     break;
199 default:
200     HAL_GPIO_WritePin(SEGA2_GPIO_Port, SEGA2_Pin, SET);
201     HAL_GPIO_WritePin(SEGB2_GPIO_Port, SEGB2_Pin, SET);
202     HAL_GPIO_WritePin(SEGC2_GPIO_Port, SEGC2_Pin, SET);
203     HAL_GPIO_WritePin(SEGD2_GPIO_Port, SEGD2_Pin, SET);
204     HAL_GPIO_WritePin(SEGE2_GPIO_Port, SEGE2_Pin, SET);
205     HAL_GPIO_WritePin(SEGF2_GPIO_Port, SEGF2_Pin, SET);
206     HAL_GPIO_WritePin(SEGG2_GPIO_Port, SEGG2_Pin, SET);
207     break;
208 }
209 }
210 int * led1;
211 int * led2;
212 int led_index = 0;
213 void update7SEG(){
214     if (endTimer0()){
215         setTimer0(10);
```

```
216     switch(led_index){
217     case 0:
218         //Turn on count led
219         SEG_PORT->ODR = 0xffff;
220         display7SEG1(*led1 % 10);
221         display7SEG2(*led1 / 10);
222         HAL_GPIO_WritePin(EN0_GPIO_Port, EN0_Pin, RESET);
223         HAL_GPIO_WritePin(EN1_GPIO_Port, EN1_Pin, SET);
224         break;
225     case 1:
226         //Turn on mode led
227         SEG_PORT->ODR = 0xffff;
228         display7SEG1(*led2 % 10);
229         display7SEG2(*led2 / 10);
230         HAL_GPIO_WritePin(EN0_GPIO_Port, EN0_Pin, SET);
231         HAL_GPIO_WritePin(EN1_GPIO_Port, EN1_Pin, RESET);
232         break;
233     default:
234         break;
235     }
236     led_index = (led_index + 1) % 2;
237 }
238 }
```

### 1.6.2 Trafficlight source code

```
1     #include "led_display.h"
2 void setRed1(){
3     HAL_GPIO_WritePin(RED1_GPIO_Port, RED1_Pin, RESET);
4     HAL_GPIO_WritePin(GREEN1_GPIO_Port, GREEN1_Pin, SET);
5     HAL_GPIO_WritePin(YELLOW1_GPIO_Port, YELLOW1_Pin, SET);
6 }
7
8 void setYellow1(){
9     HAL_GPIO_WritePin(RED1_GPIO_Port, RED1_Pin, SET);
10    HAL_GPIO_WritePin(GREEN1_GPIO_Port, GREEN1_Pin, SET);
11    HAL_GPIO_WritePin(YELLOW1_GPIO_Port, YELLOW1_Pin, RESET);
12 }
13
14 void setGreen1(){
15    HAL_GPIO_WritePin(RED1_GPIO_Port, RED1_Pin, SET);
16    HAL_GPIO_WritePin(GREEN1_GPIO_Port, GREEN1_Pin, RESET);
17    HAL_GPIO_WritePin(YELLOW1_GPIO_Port, YELLOW1_Pin, SET);
18 }
19
20 void setRed2(){
21    HAL_GPIO_WritePin(RED2_GPIO_Port, RED2_Pin, RESET);
22    HAL_GPIO_WritePin(GREEN2_GPIO_Port, GREEN2_Pin, SET);
23    HAL_GPIO_WritePin(YELLOW2_GPIO_Port, YELLOW2_Pin, SET);
```

```
24 }
25
26 void setYellow2(){
27     HAL_GPIO_WritePin(RED2_GPIO_Port, RED2_Pin, SET);
28     HAL_GPIO_WritePin(GREEN2_GPIO_Port, GREEN2_Pin, SET);
29     HAL_GPIO_WritePin(YELLOW2_GPIO_Port, YELLOW2_Pin, RESET);
30 }
31
32 void setGreen2(){
33     HAL_GPIO_WritePin(RED2_GPIO_Port, RED2_Pin, SET);
34     HAL_GPIO_WritePin(GREEN2_GPIO_Port, GREEN2_Pin, RESET);
35     HAL_GPIO_WritePin(YELLOW2_GPIO_Port, YELLOW2_Pin, SET);
36 }
37
38 void toggleRed1(){
39     HAL_GPIO_TogglePin(RED1_GPIO_Port, RED1_Pin);
40 }
41 void toggleYellow1(){
42     HAL_GPIO_TogglePin(YELLOW1_GPIO_Port, YELLOW1_Pin);
43 }
44 void toggleGreen1(){
45     HAL_GPIO_TogglePin(GREEN1_GPIO_Port, GREEN1_Pin);
46 }
47 void toggleRed2(){
48     HAL_GPIO_TogglePin(RED2_GPIO_Port, RED2_Pin);
49 }
50 void toggleYellow2(){
51     HAL_GPIO_TogglePin(YELLOW2_GPIO_Port, YELLOW2_Pin);
52 }
53 void toggleGreen2(){
54     HAL_GPIO_TogglePin(GREEN2_GPIO_Port, GREEN2_Pin);
55 }
56
57 void clearLed(){
58     HAL_GPIO_WritePin(RED1_GPIO_Port, RED1_Pin, SET);
59     HAL_GPIO_WritePin(GREEN1_GPIO_Port, GREEN1_Pin, SET);
60     HAL_GPIO_WritePin(YELLOW1_GPIO_Port, YELLOW1_Pin, SET);
61     HAL_GPIO_WritePin(RED2_GPIO_Port, RED2_Pin, SET);
62     HAL_GPIO_WritePin(GREEN2_GPIO_Port, GREEN2_Pin, SET);
63     HAL_GPIO_WritePin(YELLOW2_GPIO_Port, YELLOW2_Pin, SET);
64 }
```

## 1.7 Exercise 7

```
1 void fsm_modify(){
2     switch(status3){
3         case RED1_MODIFY:
4             //set mode to display
5             mode = 2;
```

```
6     if(endTimer1()) {
7         setTimer1(100);
8         toggleLEDs(mode);
9     }
10    if(Press_button(SET_VALUE)) {
11        clearLed();
12        red1_duration = time_input * 100;
13        time_input = 1;
14        status3 = YELLOW1_MODIFY;
15    }
16    if(Press_button(SET_TIMER)) {
17        time_input = (time_input + 1) % 100;
18        if(time_input == 0) time_input = 1;
19    }
20    if(Press_button(CONTROL)){
21        clearLed();
22        time_input = 1;
23        status3 = YELLOW1_MODIFY;
24    }
25    break;
26    case RED2_MODIFY:
27        mode = 5;
28        if(endTimer1()) {
29            setTimer1(100);
30            toggleLEDs(mode);
31        }
32        if(Press_button(SET_VALUE)) {
33            clearLed();
34            red2_duration = time_input * 100;
35            time_input = 1;
36            status3 = YELLOW2_MODIFY;
37        }
38        if(Press_button(SET_TIMER)) {
39            time_input = (time_input + 1) % 100;
40            if(time_input == 0) time_input = 1;
41        }
42        if(Press_button(CONTROL)) {
43            clearLed();
44            time_input = 1;
45            status3 = YELLOW2_MODIFY;
46        }
47        break;
```

## 1.8 Exercise 8

```
1  case YELLOW1_MODIFY:
2  mode = 3;
3  if(endTimer1()) {
4      setTimer1(100);
5      toggleLEDs(mode);
6  }
7  if(Press_button(SET_VALUE)){
8      clearLed();
9      yellow1_duration = time_input * 100;
10     time_input = 1;
11     status3 = GREEN1_MODIFY;
12 }
13 if(Press_button(SET_TIMER)) {
14     time_input = (time_input + 1) % 100;
15     if(time_input == 0) time_input = 1;
16 }
17 if(Press_button(CONTROL)){
18     clearLed();
19     time_input = 1;
20     status3 = GREEN1_MODIFY;
21 }
22 break;
23 case YELLOW2_MODIFY:
24 mode = 6;
25 if(endTimer1()) {
26     setTimer1(100);
27     toggleLEDs(mode);
28 }
29 if(Press_button(SET_VALUE)){
30     clearLed();
31     yellow2_duration = time_input * 100;
32     time_input = 1;
33     status3 = GREEN2_MODIFY;
34 }
35 if(Press_button(SET_TIMER)) {
36     time_input = (time_input + 1) % 100;
37     if(time_input == 0) time_input = 1;
38 }
39 if(Press_button(CONTROL)) {
40     clearLed();
41     time_input = 1;
42     status3 = GREEN2_MODIFY;
43 }
44 break;
```

## 1.9 Exercise 9

```
1  void updateLEDMatrix(int index) case GREEN1_MODIFY:
2  mode = 4;
3  if(endTimer1()) {
4      setTimer1(100);
5      toggleLEDs(mode);
6  }
7  if(Press_button(SET_VALUE)) {
8      clearLed();
9      green1_duration = time_input * 100;
10     time_input = 1;
11     status3 = RED2_MODIFY;
12 }
13 if(Press_button(SET_TIMER)) {
14     time_input = (time_input + 1) % 100;
15     if(time_input == 0) time_input = 1;
16 }
17 if(Press_button(CONTROL)) {
18     clearLed();
19     time_input = 1;
20     status3 = RED2_MODIFY;
21 }
22 break;
23 case GREEN2_MODIFY:
24     mode = 7;
25     if (endTimer1()){
26         setTimer1(100);
27         toggleLEDs(mode);
28     }
29     if(Press_button(SET_VALUE)) {
30         clearLed();
31         green2_duration = time_input * 100;
32         time_input = 1;
33         return_flag1 = return_flag2 = 1;
34         status3 = DELAY;
35         led1 = &countdown1;
36         led2 = &countdown2;
37     }
38
39     if(Press_button(SET_TIMER)) {
40         time_input = (time_input % 99) + 1;
41     }
42
43     if(Press_button(CONTROL)) {
44         clearLed();
45         time_input = 1;
46         return_flag1 = return_flag2 = 1;
47         status3 = DELAY;
48         led1 = &countdown1;
```





```
49     led2 = &countdown2;  
50     }  
51     break;
```

## 1.10 Exercise 10

- You can get the source code in the link Github before.



## References