

Ho Chi Minh City University of Technology
MICROPROCESSORS-MICROCONTROLLERS
LAB 1 REPORT

Vương Quốc Anh

November 15, 2023

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.

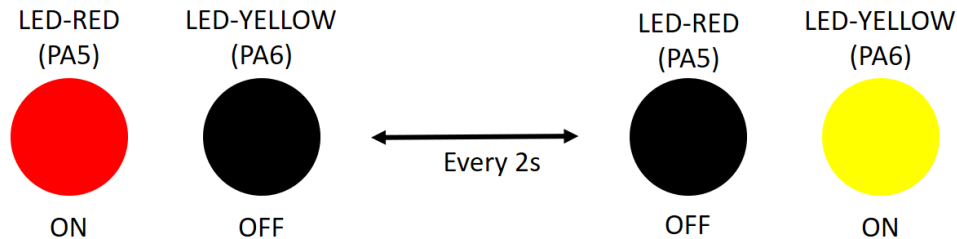


Figure 1: *State transitions for 2 LEDs*

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).

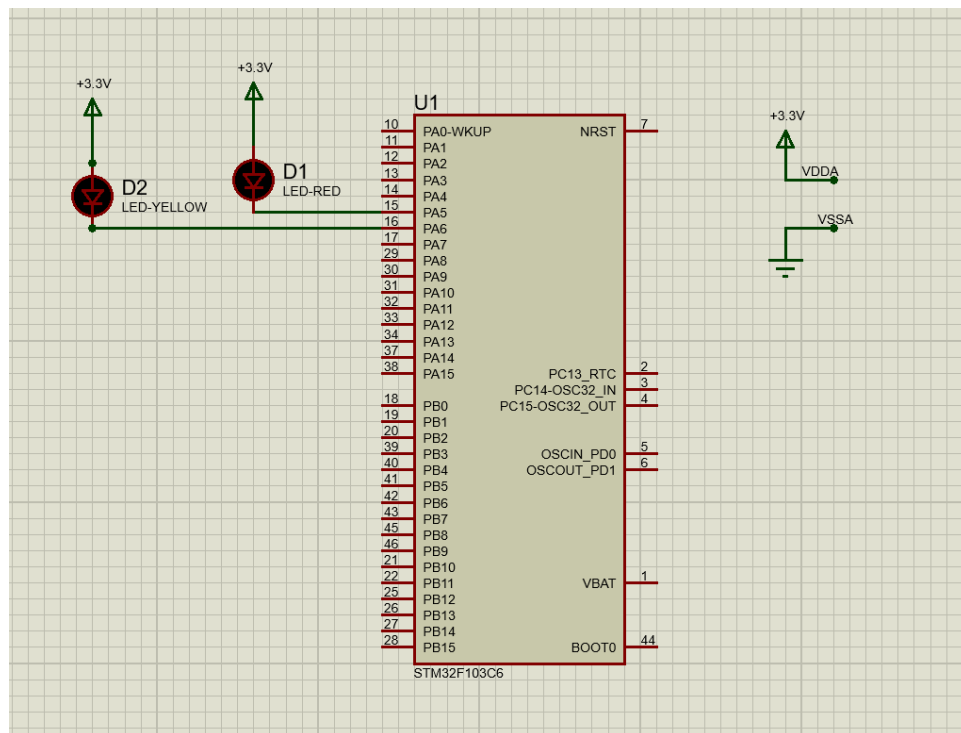


Figure 2: <https://github.com/VgQ8Auk/ProteusPro/blob/main/Ex1.pdsprj>

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     HAL_GPIO_WritePin( LED_RED_GPIO_Port , LED_RED_Pin , GPIO_PIN_SET);
```

```

4  HAL_GPIO_WritePin( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin , GPIO_PIN_SET);
5  HAL_GPIO_TogglePin ( LED_RED_GPIO_Port , LED_RED_Pin );
6  HAL_Delay (2000) ;
7  HAL_GPIO_TogglePin ( LED_RED_GPIO_Port , LED_YELLOW_Pin );
8  HAL_GPIO_TogglePin ( LED_RED_GPIO_Port , LED_RED_Pin );
9  HAL_Delay (2000) ;
10 }

```

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.

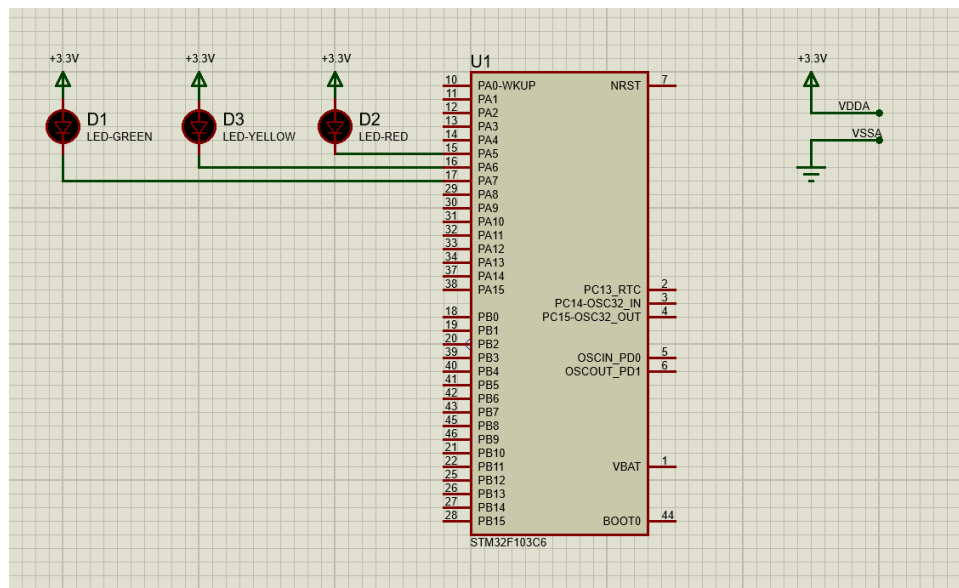


Figure 3: <https://github.com/VgQ8Auk/ProteusPro/blob/main/Ex2.pdsprj>

Report 2: Present the source code in while.

```

1  while (1)
2  {
3      HAL_GPIO_WritePin( LED_RED_GPIO_Port , LED_RED_Pin , GPIO_PIN_SET);
4      HAL_GPIO_WritePin( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin , GPIO_PIN_SET);
5      HAL_GPIO_WritePin( LED_GREEN_GPIO_Port , LED_GREEN_Pin , GPIO_PIN_SET);
6      HAL_GPIO_TogglePin( LED_RED_GPIO_Port , LED_RED_Pin);
7      HAL_Delay (5000);
8      HAL_GPIO_TogglePin ( LED_GREEN_GPIO_Port , LED_GREEN_Pin );
9      HAL_GPIO_TogglePin( LED_RED_GPIO_Port , LED_RED_Pin);
10     HAL_Delay (3000);
11     HAL_GPIO_TogglePin ( LED_GREEN_GPIO_Port , LED_GREEN_Pin );
12     HAL_GPIO_TogglePin ( LED_YELLOW_GPIO_Port , LED_YELLOW_Pin );
13     HAL_Delay (2000);
14 }

```

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.

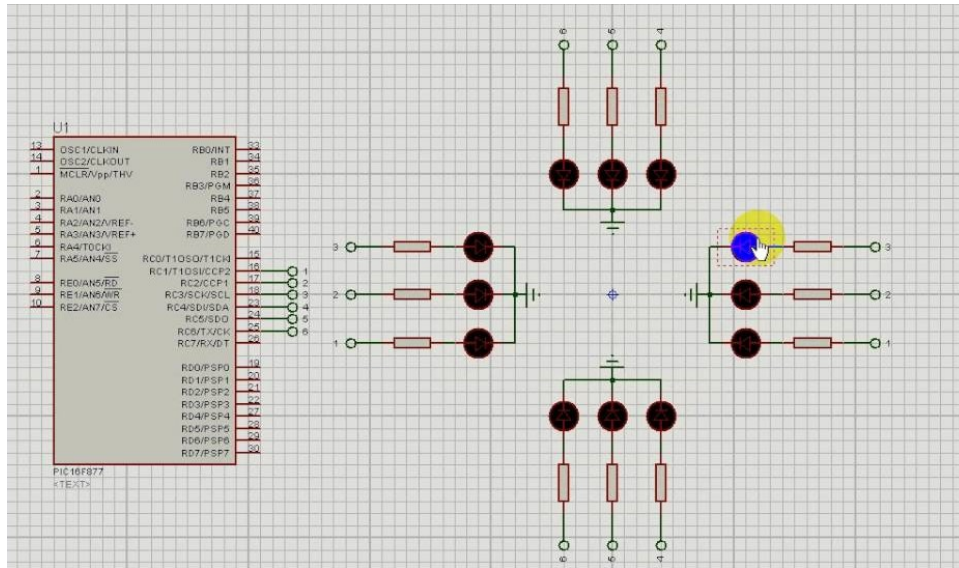


Figure 4: Reference design for a 4 way traffic light

<https://github.com/VgQ8Auk/LAB1MCU/tree/Ex3>

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:

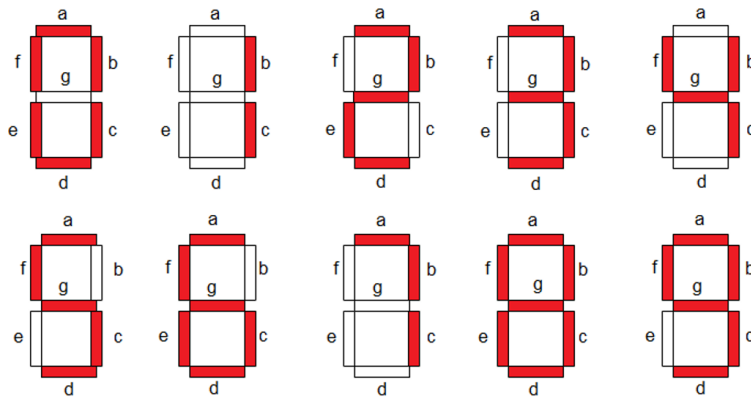


Figure 5: Display a number on 7 segment LED

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

```

1 int counter = 0;
2 while (1){
3     if(counter >= 10) counter = 0;
4     display7SEG(counter++);
5     HAL_Delay(1000);
6 }
7

```

Listing 1: An example for your source code

Report 1: Present the schematic.

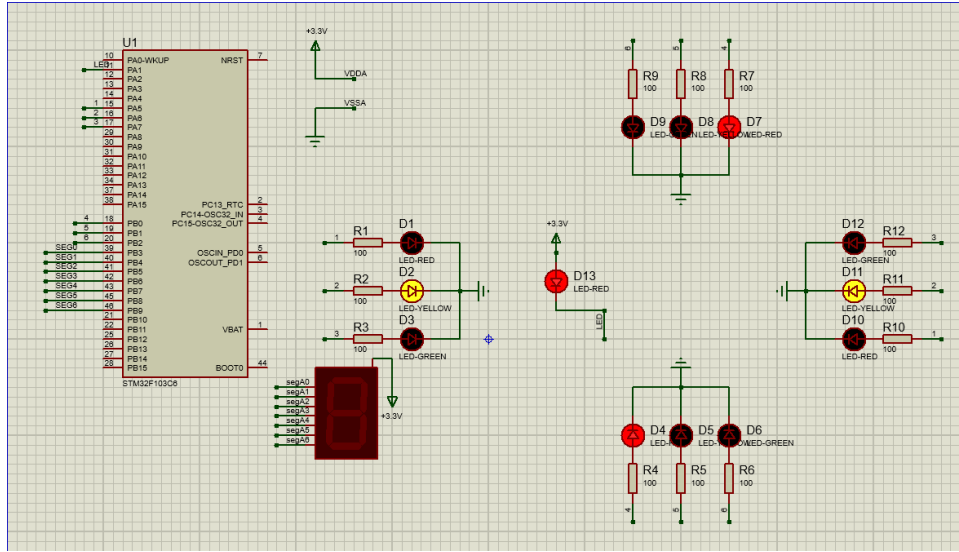


Figure 6:

Report 2: Present the source code for display7SEG function.

```

1 void SetAndDisplay7Segments(const struct Sevenled* led, unsigned short number)
2 {
3     switch(number)
4     {
5     case 0:
6         HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
7         HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
8         HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
9         HAL_GPIO_WritePin(led->gpio, led->pin_d, 0);
10        HAL_GPIO_WritePin(led->gpio, led->pin_e, 0);
11        HAL_GPIO_WritePin(led->gpio, led->pin_f, 0);
12        HAL_GPIO_WritePin(led->gpio, led->pin_g, 1);
13        break;
14    case 1:
15        HAL_GPIO_WritePin(led->gpio, led->pin_a, 1);
16        HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
17        HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
18        HAL_GPIO_WritePin(led->gpio, led->pin_d, 1);
19        HAL_GPIO_WritePin(led->gpio, led->pin_e, 1);
20        HAL_GPIO_WritePin(led->gpio, led->pin_f, 1);
21        HAL_GPIO_WritePin(led->gpio, led->pin_g, 1);
22        break;
23    case 2:
24        HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
25        HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
26        HAL_GPIO_WritePin(led->gpio, led->pin_c, 1);
27        HAL_GPIO_WritePin(led->gpio, led->pin_d, 0);
28        HAL_GPIO_WritePin(led->gpio, led->pin_e, 0);
29        HAL_GPIO_WritePin(led->gpio, led->pin_f, 1);

```

```

30     HAL_GPIO_WritePin(led->gpio, led->pin_g, 0);
31     break;
32 case 3:
33     HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
34     HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
35     HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
36     HAL_GPIO_WritePin(led->gpio, led->pin_d, 0);
37     HAL_GPIO_WritePin(led->gpio, led->pin_e, 1);
38     HAL_GPIO_WritePin(led->gpio, led->pin_f, 1);
39     HAL_GPIO_WritePin(led->gpio, led->pin_g, 0);
40     break;
41 case 4:
42     HAL_GPIO_WritePin(led->gpio, led->pin_a, 1);
43     HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
44     HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
45     HAL_GPIO_WritePin(led->gpio, led->pin_d, 1);
46     HAL_GPIO_WritePin(led->gpio, led->pin_e, 1);
47     HAL_GPIO_WritePin(led->gpio, led->pin_f, 0);
48     HAL_GPIO_WritePin(led->gpio, led->pin_g, 0);
49     break;
50 case 5:
51     HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
52     HAL_GPIO_WritePin(led->gpio, led->pin_b, 1);
53     HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
54     HAL_GPIO_WritePin(led->gpio, led->pin_d, 0);
55     HAL_GPIO_WritePin(led->gpio, led->pin_e, 1);
56     HAL_GPIO_WritePin(led->gpio, led->pin_f, 0);
57     HAL_GPIO_WritePin(led->gpio, led->pin_g, 0);
58     break;
59 case 6:
60     HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
61     HAL_GPIO_WritePin(led->gpio, led->pin_b, 1);
62     HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
63     HAL_GPIO_WritePin(led->gpio, led->pin_d, 0);
64     HAL_GPIO_WritePin(led->gpio, led->pin_e, 0);
65     HAL_GPIO_WritePin(led->gpio, led->pin_f, 0);
66     HAL_GPIO_WritePin(led->gpio, led->pin_g, 0);
67     break;
68 case 7:
69     HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
70     HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
71     HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
72     HAL_GPIO_WritePin(led->gpio, led->pin_d, 1);
73     HAL_GPIO_WritePin(led->gpio, led->pin_e, 1);
74     HAL_GPIO_WritePin(led->gpio, led->pin_f, 1);
75     HAL_GPIO_WritePin(led->gpio, led->pin_g, 1);
76     break;
77 case 8:
78     HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
79     HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
80     HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
81     HAL_GPIO_WritePin(led->gpio, led->pin_d, 0);
82     HAL_GPIO_WritePin(led->gpio, led->pin_e, 0);
83     HAL_GPIO_WritePin(led->gpio, led->pin_f, 0);
84     HAL_GPIO_WritePin(led->gpio, led->pin_g, 0);
85     break;
86 case 9:
87     HAL_GPIO_WritePin(led->gpio, led->pin_a, 0);
88     HAL_GPIO_WritePin(led->gpio, led->pin_b, 0);
89     HAL_GPIO_WritePin(led->gpio, led->pin_c, 0);
90     HAL_GPIO_WritePin(led->gpio, led->pin_d, 0);
91     HAL_GPIO_WritePin(led->gpio, led->pin_e, 1);
92     HAL_GPIO_WritePin(led->gpio, led->pin_f, 0);
93     HAL_GPIO_WritePin(led->gpio, led->pin_g, 0);
94     break;
95 default:
96     HAL_GPIO_WritePin(led->gpio, led->pin_a, 1);
97     HAL_GPIO_WritePin(led->gpio, led->pin_b, 1);
98     HAL_GPIO_WritePin(led->gpio, led->pin_c, 1);
99     HAL_GPIO_WritePin(led->gpio, led->pin_d, 1);
100    HAL_GPIO_WritePin(led->gpio, led->pin_e, 1);

```

```

101     HAL_GPIO_WritePin(led->gpio, led->pin_f, 1);
102     HAL_GPIO_WritePin(led->gpio, led->pin_g, 1);
103     break;
104 }
105 }

```

Exercise 5

Integrate the 7SEG-LED to the 4 way traffic light. In this case, the 7SEG-LED is used to display countdown value.

```

1  while (1)
2  {
3      /* USER CODE END WHILE */
4      if (timerflag0 == 1) {
5          HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_1);
6          setTimer0(100);
7      }
8      switch(status)
9      {
10         case 1:
11             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
12             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
13             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
14             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
15             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 0);
16             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 1);
17             if (timerflag == 1) {
18                 status = 2;
19                 setTimer(200);
20             }
21             break;
22         case 2:
23             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
24             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
25             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
26             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
27             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 1);
28             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 0);
29             if (timerflag == 1) {
30                 status = 3;
31                 setTimer(300);
32             }
33             break;
34         case 3:
35             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
36             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
37             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
38             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
39             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 0);
40             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 0);
41             if (timerflag == 1) {
42                 status = 4;
43                 setTimer(200);
44             }
45             break;
46         case 4:
47             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
48             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
49             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
50             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 1);
51             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 0);
52             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 0);
53             if (timerflag == 1) {
54                 status = 1;
55                 setTimer(300);
56             }
57             break;

```

```

58     default:
59         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
60         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
61         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
62         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, 0);
63         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, 0);
64         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_2, 1);
65         if (timerflag == 1) {
66             status = 2;
67             setTimer(200);
68         }
69         break;
70     }
71     switch(lednumA)
72     {
73     case 0:
74         SetAndDisplay7Segments(&ledA, 0);
75         if (timerflag1 == 1)
76         {
77             if (trigger5A) {lednumA = 5; trigger5A = 0; trigger3A = 1; setTimer1(80);
78             break;}
79             if (trigger3A) {lednumA = 3; trigger3A = 0; trigger2A = 1; setTimer1(80);
80             break;}
81             if (trigger2A) {lednumA = 2; trigger2A = 0; trigger5A = 1; setTimer1(80);
82             break;}
83             break;
84         case 1:
85             SetAndDisplay7Segments(&ledA, 1);
86             if (timerflag1 == 1)
87             {
88                 lednumA = 0;
89                 setTimer1(20);
90             }
91             break;
92         case 2:
93             SetAndDisplay7Segments(&ledA, 2);
94             if (timerflag1 == 1)
95             {
96                 lednumA = 1;
97                 setTimer1(100);
98             }
99             break;
100         case 3:
101             SetAndDisplay7Segments(&ledA, 3);
102             if (timerflag1 == 1)
103             {
104                 lednumA = 2;
105                 setTimer1(100);
106             }
107             break;
108         case 4:
109             SetAndDisplay7Segments(&ledA, 4);
110             if (timerflag1 == 1)
111             {
112                 lednumA = 3;
113                 setTimer1(100);
114             }
115             break;
116         case 5:
117             SetAndDisplay7Segments(&ledA, 5);
118             if (timerflag1 == 1)
119             {
120                 lednumA = 4;
121                 setTimer1(100);
122             }
123             break;
124         case 6:
125             SetAndDisplay7Segments(&ledA, 6);
126             if (timerflag1 == 1)
127             {

```



```

126         lednumA = 5;
127         setTimer1(100);
128     }
129     break;
130 case 7:
131     SetAndDisplay7Segments(&ledA, 7);
132     if (timerflag1 == 1)
133     {
134         lednumA = 6;
135         setTimer1(100);
136     }
137     break;
138 case 8:
139     SetAndDisplay7Segments(&ledA, 8);
140     if (timerflag1 == 1)
141     {
142         lednumA = 7;
143         setTimer1(100);
144     }
145     break;
146 case 9:
147     SetAndDisplay7Segments(&ledA, 9);
148     if (timerflag1 == 1)
149     {
150         lednumA = 8;
151         setTimer1(100);
152     }
153     break;
154 default:
155     SetAndDisplay7Segments(&ledA, 0);
156     setTimer1(50);
157     break;
158 }

159 switch(lednumB)
160 {
161 case 0:
162     SetAndDisplay7Segments(&ledB, 0);
163     if (timerflag2 == 1)
164     {
165         if (trigger5B) {lednumB = 5; trigger5B = 0; trigger3B = 1; setTimer2(80);
166 break;}
167         if (trigger3B) {lednumB = 3; trigger3B = 0; trigger2B = 1; setTimer2(80);
168 break;}
169         if (trigger2B) {lednumB = 2; trigger2B = 0; trigger5B = 1; setTimer2(80);
170 break;}
171         setTimer2(80);
172     }
173     break;
174 case 1:
175     SetAndDisplay7Segments(&ledB, 1);
176     if (timerflag2 == 1)
177     {
178         lednumB = 0;
179         setTimer2(20);
180     }
181     break;
182 case 2:
183     SetAndDisplay7Segments(&ledB, 2);
184     if (timerflag2 == 1)
185     {
186         lednumB = 1;
187         setTimer2(100);
188     }
189     break;
190 case 3:
191     SetAndDisplay7Segments(&ledB, 3);
192     if (timerflag2 == 1)
193     {
194         lednumB = 2;
195         setTimer2(100);

```

```

194     }
195     break;
196 case 4:
197     SetAndDisplay7Segments(&ledB, 4);
198     if (timerflag2 == 1)
199     {
200         lednumB = 3;
201         setTimer2(100);
202     }
203     break;
204 case 5:
205     SetAndDisplay7Segments(&ledB, 5);
206     if (timerflag2 == 1)
207     {
208         lednumB = 4;
209         setTimer2(100);
210     }
211     break;
212 case 6:
213     SetAndDisplay7Segments(&ledB, 6);
214     if (timerflag2 == 1)
215     {
216         lednumB = 5;
217         setTimer2(100);
218     }
219     break;
220 case 7:
221     SetAndDisplay7Segments(&ledB, 7);
222     if (timerflag2 == 1)
223     {
224         lednumB = 6;
225         setTimer2(100);
226     }
227     break;
228 case 8:
229     SetAndDisplay7Segments(&ledB, 8);
230     if (timerflag2 == 1)
231     {
232         lednumB = 7;
233         setTimer2(100);
234     }
235     break;
236 case 9:
237     SetAndDisplay7Segments(&ledB, 9);
238     if (timerflag2 == 1)
239     {
240         lednumB = 8;
241         setTimer2(100);
242     }
243     break;
244 default:
245     SetAndDisplay7Segments(&ledB, 0);
246     setTimer2(50);
247     break;
248 }
249 /* USER CODE BEGIN 3 */
250 }
251 /* USER CODE END 3 */

```

<https://github.com/VgQ8Auk/LAB1MCU/tree/Ex4&5>

In this exercise, only source code is required to present. The function display7SEG in previous exercise can be re-used.

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.

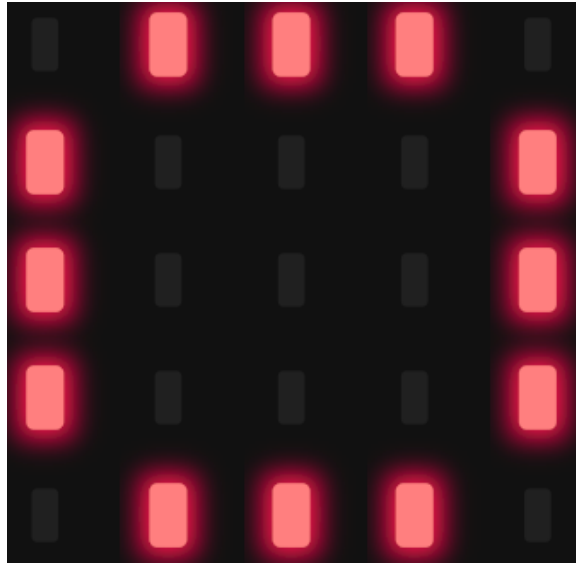
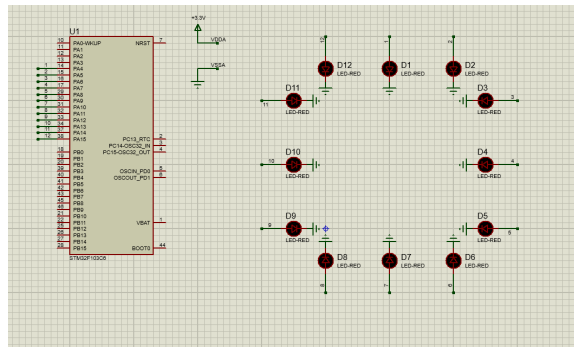


Figure 7: 12 LEDs for an analog clock

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.

Exercise 7

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

```
1 void clearAllClock(){
2     //TODO
3 }
```

Listing 2: Function Implementation

```

1 void clearAllClock(void){
2     //TODO
3     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);
4     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
5     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
6     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
7     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, 0);
8     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 0);
9     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 0);
10    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 0);
11    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 0);
12    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, 0);
13    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, 0);
14    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);
15 }

```

void clearAllClock(void)

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     //TODO
3     switch(num)
4     {
5         case 0:
6             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);
7             break;
8         case 1:
9             HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
10            break;
11        case 2:
12            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
13            break;
14        case 3:
15            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
16            break;
17        case 4:
18            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, 1);
19            break;
20        case 5:
21            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 1);
22            break;
23        case 6:
24            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 1);
25            break;
26        case 7:
27            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 1);
28            break;
29        case 8:
30            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 1);
31            break;
32        case 9:
33            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, 1);
34            break;
35        case 10:
36            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, 1);
37            break;
38        case 11:
39            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1);
40            break;
41        default:
42            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 1);
43            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 1);
44            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
45            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
46            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, 1);

```

```

47     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 1);
48     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 1);
49     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 1);
50     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 1);
51     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, 1);
52     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, 1);
53     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 1);
54     break;
55 }
56 }

```

```
void setNumberOnClock(int num)
```

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  //TODO
3  switch(num)
4  {
5      case 0:
6          HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);
7          break;
8      case 1:
9          HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
10         break;
11     case 2:
12         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
13         break;
14     case 3:
15         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
16         break;
17     case 4:
18         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, 0);
19         break;
20     case 5:
21         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 0);
22         break;
23     case 6:
24         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 0);
25         break;
26     case 7:
27         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 0);
28         break;
29     case 8:
30         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 0);
31         break;
32     case 9:
33         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, 0);
34         break;
35     case 10:
36         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, 0);
37         break;
38     case 11:
39         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);
40         break;
41     default:
42         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, 0);
43         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, 0);
44         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
45         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 0);
46         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_8, 0);
47         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 0);
48         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_10, 0);
49         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, 0);
50         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, 0);
51         HAL_GPIO_WritePin(GPIOA, GPIO_PIN_13, 0);

```

```
52     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_14, 0);  
53     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, 0);  
54     break;  
55 }  
56 }
```

```
void clearNumberOnClock(int num)
```

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.

<https://github.com/VgQ8Auk/LAB1MCU/tree/Ex7%268%269%2610>