

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



**THÍ NGHIỆM**

---

**VI XỬ LÝ - VI ĐIỀU KHIỂN (CO 3010)**

---

**BÁO CÁO LAB 1**

**Giảng viên: PHAN VĂN SỸ**

| Họ và tên       | Mã số sinh viên |
|-----------------|-----------------|
| Nguyễn Văn Sang | 2212912         |

TP.Hồ Chí Minh, 24/09/2025



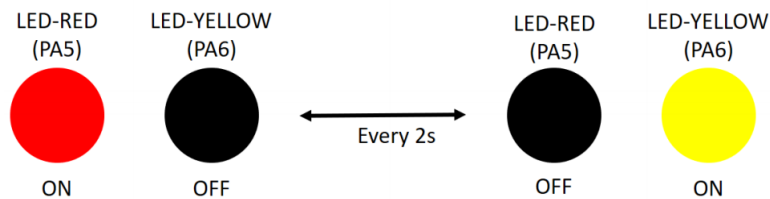
## Mục lục

|                              |    |
|------------------------------|----|
| 1 Bài 1                      | 2  |
| 2 Bài 2                      | 3  |
| 3 Bài 4                      | 4  |
| 4 Bài 5                      | 6  |
| 5 Bài 6                      | 8  |
| 6 Bài 7                      | 10 |
| 7 Bài 8:                     | 11 |
| 8 Bài 9:                     | 12 |
| 9 Bài 10:                    | 13 |
| 10 Link Github toàn bộ LAB 1 | 14 |

## 1 Bài 1

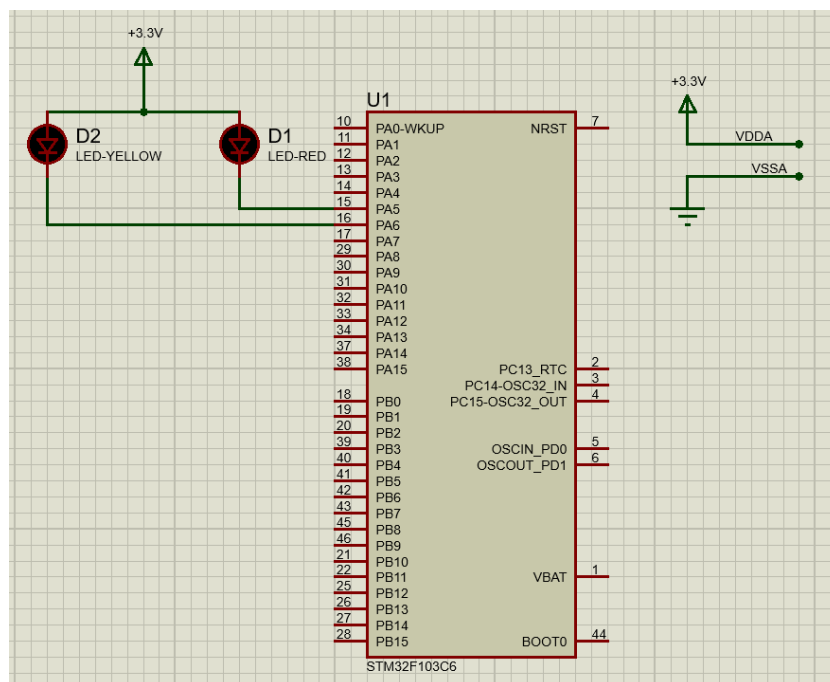
From the simulation on Proteus, one more LED is connected to pin PA6 of the STM32 (negative pin of the LED is connected to PA6). The component suggested in this exercise is LED-YELLOW, which can be found from the device list.

In this exercise, the status of two LEDs are switched every 2 seconds, as demonstrated in the figure bellow.



Hình 1: State transitions for 2 LEDs

### Report 1:



Hình 2: Schematic of exercise 1

### Report 2:

```
1 while (1) {
2     // Led_Do turn on, Led_Vang turn off
3     HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, RESET);
```

```

4  HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, SET);
5  HAL_Delay(2000);
6
7  // Led_Vang turn on, Led_Do turn off
8  HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, SET);
9  HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, RESET);
10  HAL_Delay(2000);
11 }

```

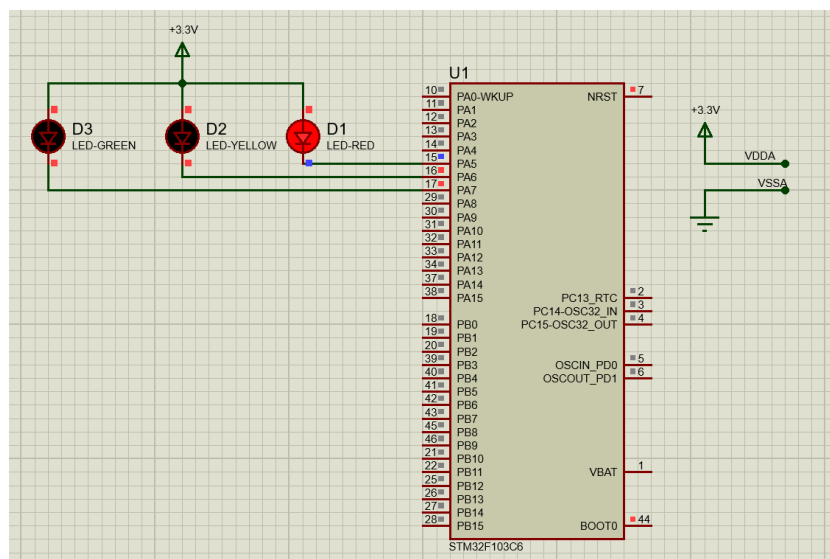
Link Github bài 1: <https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B1>

## 2 Bài 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:



Hình 3: Schematic of exercise 2

### Report 2:

```

1  while (1) {
2      // Led_Xanh turn on, Led_Vang and Led_Do turn off
3      HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, SET);
4      HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, SET);
5      HAL_GPIO_WritePin(Led_Xanh_GPIO_Port, Led_Xanh_Pin, RESET);
6      HAL_Delay(3000);
7
8      // Led_Vang turn on, Led_Xanh and Led_Do turn off

```

```

9  HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, SET);
10 HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, RESET);
11 HAL_GPIO_WritePin(Led_Xanh_GPIO_Port, Led_Xanh_Pin, SET);
12 HAL_Delay(2000);
13
14 // Led_Do turn on, Led_Xanh and Led_Vang turn off
15 HAL_GPIO_WritePin(Led_Do_GPIO_Port, Led_Do_Pin, RESET);
16 HAL_GPIO_WritePin(Led_Vang_GPIO_Port, Led_Vang_Pin, SET);
17 HAL_GPIO_WritePin(Led_Xanh_GPIO_Port, Led_Xanh_Pin, SET);
18 HAL_Delay(5000);
19 }

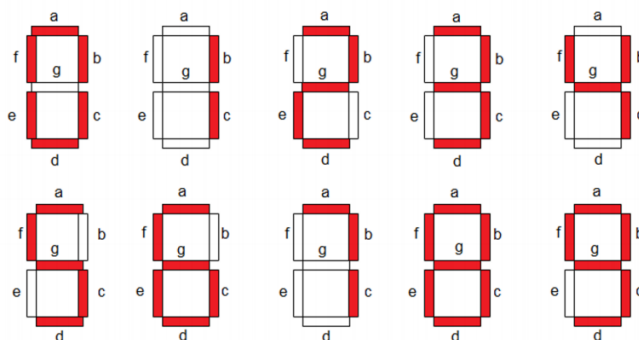
```

Link Github bài 2: <https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B2>

### 3 Bài 4

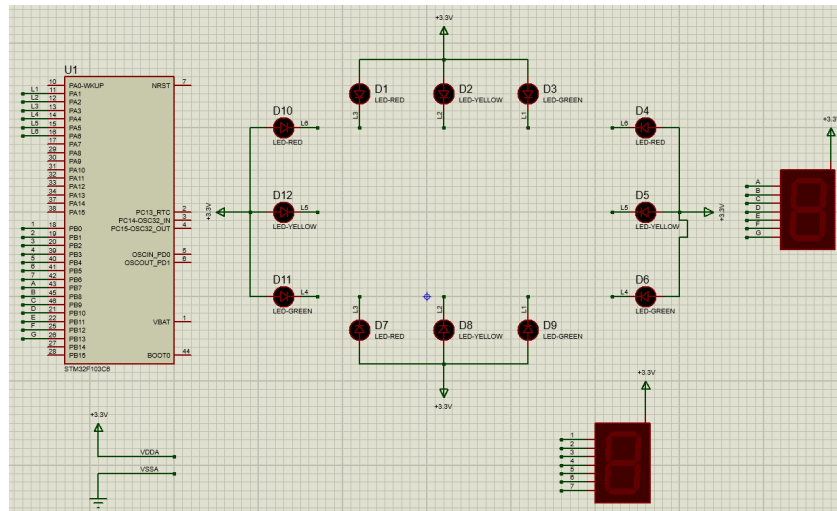
Add only one 7 led segment to the schematic in Exercise 3. This component can be found in Proteus by the keyword 7SEG-COM-ANODE. For this device, the common pin should be connected to the power supply and other pins are supposed to be connected to PB0 to PB6. Therefore, to turn-on a segment in this 7SEG, the STM32 pin should be in logic 0 (0V).

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



Hình 4: Display a number on 7 segment LED

#### Report 1:



Hình 5: Schematic of exercise 4 and 5

## Report 2:

```
1  int main(void) {
2      // Turn off traffic LEDs and clear both 7-seg
3      setNS(0,0,0);
4      setEW(0,0,0);
5      clear7SEG_NS();
6      clear7SEG_EW();
7
8      while (1) {
9          // NS RED, EW GREEN and YELLOW
10         setNS(0,0,1); // NS red on
11         setEW(1,0,0); // EW green on
12         for (int s = 0; s < EW_GREEN_TIME; s++) {
13             display7SEG_NS(NS_RED_TIME - s - 1); // NS red counts 8->4
14             display7SEG_EW(EW_GREEN_TIME - s - 1); // EW green counts 5->1
15             HAL_Delay(1000);
16         }
17         setEW(0,1,0); // EW yellow on
18         for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
19             display7SEG_NS(NS_RED_TIME - EW_GREEN_TIME - s - 1); // NS red
20             counts 3->0
21             display7SEG_EW(EW_YELLOW_TIME - s); // EW yellow counts 2->0
22             HAL_Delay(1000);
23         }
24         // EW RED, North-South GREEN and YELLOW
25         setNS(1,0,0); // NS green on
26         setEW(0,0,1); // EW red on
27         for (int s = 0; s < EW_GREEN_TIME; s++) {
28             display7SEG_EW(NS_RED_TIME - s - 1); // EW red counts 8->4
29             display7SEG_NS(EW_GREEN_TIME - s - 1); // NS green counts 5->1
30             HAL_Delay(1000);
31         }
32         setNS(0,1,0); // NS yellow on
33         for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
34             display7SEG_EW(NS_RED_TIME - EW_GREEN_TIME - s - 1); // EW red
35             counts 3->0
36             display7SEG_NS(EW_YELLOW_TIME - s); // NS yellow counts 2->0
```

```
36         HAL_Delay(1000);  
37     }  
38 }  
39 }
```

## 4 Bài 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 reused.

```
1  // Timing constants  
2  #define NS_RED_TIME 9 // NS red counts 8->0  
3  #define EW_GREEN_TIME 5 // EW green counts 5->0  
4  #define EW_YELLOW_TIME 3 // EW yellow counts 2->0  
5  
6  // GPIO init: configure PA and PB as outputs  
7  static void MX_GPIO_Init(void) {  
8      GPIO_InitTypeDef GPIO_InitStruct = {0};  
9  
10     __HAL_RCC_GPIOA_CLK_ENABLE();  
11     __HAL_RCC_GPIOB_CLK_ENABLE();  
12  
13     // Traffic LEDs (PA) - set pins OFF initially  
14     HAL_GPIO_WritePin(L1_GPIO_Port, L1_Pin, GPIO_PIN_SET);  
15     HAL_GPIO_WritePin(L2_GPIO_Port, L2_Pin, GPIO_PIN_SET);  
16     HAL_GPIO_WritePin(L3_GPIO_Port, L3_Pin, GPIO_PIN_SET);  
17     HAL_GPIO_WritePin(L4_GPIO_Port, L4_Pin, GPIO_PIN_SET);  
18     HAL_GPIO_WritePin(L5_GPIO_Port, L5_Pin, GPIO_PIN_SET);  
19     HAL_GPIO_WritePin(L6_GPIO_Port, L6_Pin, GPIO_PIN_SET);  
20  
21     GPIO_InitStruct.Pin = L1_Pin|L2_Pin|L3_Pin|L4_Pin|L5_Pin|L6_Pin;  
22     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;  
23     GPIO_InitStruct.Pull = GPIO_NOPULL;  
24     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;  
25     HAL_GPIO_Init(L1_GPIO_Port, &GPIO_InitStruct); // all Lx on same port (GPIOA  
26     )  
27  
28     // 7-seg NS and EW on PB - set OFF initially  
29     HAL_GPIO_WritePin(a_GPIO_Port, a_Pin, GPIO_PIN_SET);  
30     HAL_GPIO_WritePin(b_GPIO_Port, b_Pin, GPIO_PIN_SET);  
31     HAL_GPIO_WritePin(c_GPIO_Port, c_Pin, GPIO_PIN_SET);  
32     HAL_GPIO_WritePin(d_GPIO_Port, d_Pin, GPIO_PIN_SET);  
33     HAL_GPIO_WritePin(e_GPIO_Port, e_Pin, GPIO_PIN_SET);  
34     HAL_GPIO_WritePin(f_GPIO_Port, f_Pin, GPIO_PIN_SET);  
35     HAL_GPIO_WritePin(g_GPIO_Port, g_Pin, GPIO_PIN_SET);  
36  
37     HAL_GPIO_WritePin(A_GPIO_Port, A_Pin, GPIO_PIN_SET);  
38     HAL_GPIO_WritePin(B_GPIO_Port, B_Pin, GPIO_PIN_SET);  
39     HAL_GPIO_WritePin(C_GPIO_Port, C_Pin, GPIO_PIN_SET);  
40     HAL_GPIO_WritePin(D_GPIO_Port, D_Pin, GPIO_PIN_SET);  
41     HAL_GPIO_WritePin(E_GPIO_Port, E_Pin, GPIO_PIN_SET);  
42     HAL_GPIO_WritePin(F_GPIO_Port, F_Pin, GPIO_PIN_SET);  
43     HAL_GPIO_WritePin(G_GPIO_Port, G_Pin, GPIO_PIN_SET);  
44  
45     GPIO_InitStruct.Pin = a_Pin|b_Pin|c_Pin|d_Pin|e_Pin|f_Pin|g_Pin  
46     | A_Pin|B_Pin|C_Pin|D_Pin|E_Pin|F_Pin|G_Pin;  
47     GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
```

```
47     GPIO_InitStruct.Pull = GPIO_NOPULL;
48     GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
49     HAL_GPIO_Init(a_GPIO_Port, &GPIO_InitStruct); // all segment pins on GPIOB
50 }
51
52 // Light status control function
53 void setLED(GPIO_TypeDef* port, uint16_t pin, uint8_t on) {
54     HAL_GPIO_WritePin(port, pin, on ? GPIO_PIN_RESET : GPIO_PIN_SET);
55 }
56
57 // Set NS lights
58 void setNS(uint8_t g, uint8_t y, uint8_t r) {
59     setLED(L1_GPIO_Port, L1_Pin, g);
60     setLED(L2_GPIO_Port, L2_Pin, y);
61     setLED(L3_GPIO_Port, L3_Pin, r);
62 }
63
64 // Set EW lights
65 void setEW(uint8_t g, uint8_t y, uint8_t r) {
66     setLED(L4_GPIO_Port, L4_Pin, g);
67     setLED(L5_GPIO_Port, L5_Pin, y);
68     setLED(L6_GPIO_Port, L6_Pin, r);
69 }
70
71 int main(void) {
72     // Turn off traffic LEDs and clear both 7-seg
73     setNS(0,0,0);
74     setEW(0,0,0);
75     clear7SEG_NS();
76     clear7SEG_EW();
77
78     while (1) {
79         // NS RED, EW GREEN and YELLOW
80         setNS(0,0,1); // NS red on
81         setEW(1,0,0); // EW green on
82         for (int s = 0; s < EW_GREEN_TIME; s++) {
83             display7SEG_NS(NS_RED_TIME - s - 1); // NS red counts 8->4
84             display7SEG_EW(EW_GREEN_TIME - s - 1); // EW green counts 5->1
85             HAL_Delay(1000);
86         }
87         setEW(0,1,0); // EW yellow on
88         for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
89             display7SEG_NS(NS_RED_TIME - EW_GREEN_TIME - s - 1); // NS red
90             counts 3->0
91             display7SEG_EW(EW_YELLOW_TIME - s); // EW yellow counts 2->0
92             HAL_Delay(1000);
93         }
94
95         // EW RED, North-South GREEN and YELLOW
96         setNS(1,0,0); // NS green on
97         setEW(0,0,1); // EW red on
98         for (int s = 0; s < EW_GREEN_TIME; s++) {
99             display7SEG_EW(NS_RED_TIME - s - 1); // EW red counts 8->4
100             display7SEG_NS(EW_GREEN_TIME - s - 1); // NS green counts 5->1
101             HAL_Delay(1000);
102         }
103         setNS(0,1,0); // NS yellow on
104         for (int s = 0; s < EW_YELLOW_TIME + 1; s++) {
105             display7SEG_EW(NS_RED_TIME - EW_GREEN_TIME - s - 1); // EW red
106             counts 3->0
107             display7SEG_NS(EW_YELLOW_TIME - s); // NS yellow counts 2->0
108             HAL_Delay(1000);
109         }
110     }
```

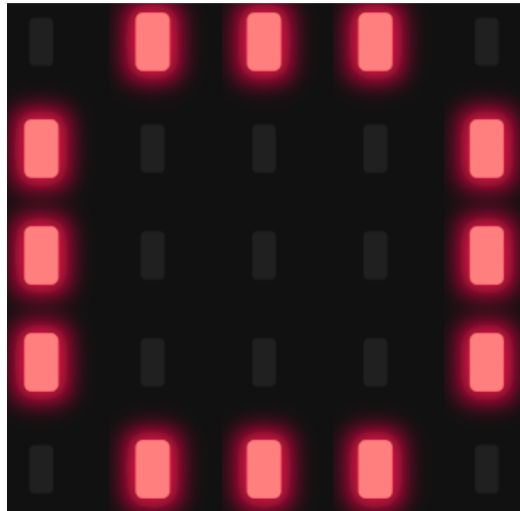


```
107     }
108     }
109 }
110
111 // Display digit on NS LED-7-seg
112 void display7SEG_NS(int num) {
113     clear7SEG_NS();
114     const uint16_t digits[10] = {
115         a_Pin|b_Pin|c_Pin|d_Pin|e_Pin|f_Pin,
116         b_Pin|c_Pin,
117         a_Pin|b_Pin|d_Pin|e_Pin|g_Pin,
118         a_Pin|b_Pin|c_Pin|d_Pin|g_Pin,
119         b_Pin|c_Pin|f_Pin|g_Pin,
120         a_Pin|c_Pin|d_Pin|f_Pin|g_Pin,
121         a_Pin|c_Pin|d_Pin|e_Pin|f_Pin|g_Pin,
122         a_Pin|b_Pin|c_Pin,
123         a_Pin|b_Pin|c_Pin|d_Pin|e_Pin|f_Pin|g_Pin,
124         a_Pin|b_Pin|c_Pin|d_Pin|f_Pin|g_Pin
125     };
126     if (num >= 0 && num <= 9) {
127         HAL_GPIO_WritePin(a_GPIO_Port, digits[num], GPIO_PIN_RESET);
128     }
129 }
130
131 // Display digit on EW LED-7-seg
132 void display7SEG_EW(int num) {
133     clear7SEG_EW();
134     const uint16_t digits[10] = {
135         A_Pin|B_Pin|C_Pin|D_Pin|E_Pin|F_Pin,
136         B_Pin|C_Pin,
137         A_Pin|B_Pin|D_Pin|E_Pin|G_Pin,
138         A_Pin|B_Pin|C_Pin|D_Pin|G_Pin,
139         B_Pin|C_Pin|F_Pin|G_Pin,
140         A_Pin|C_Pin|D_Pin|F_Pin|G_Pin,
141         A_Pin|C_Pin|D_Pin|E_Pin|F_Pin|G_Pin,
142         A_Pin|B_Pin|C_Pin,
143         A_Pin|B_Pin|C_Pin|D_Pin|E_Pin|F_Pin|G_Pin,
144         A_Pin|B_Pin|C_Pin|D_Pin|F_Pin|G_Pin
145     };
146     if (num >= 0 && num <= 9) {
147         HAL_GPIO_WritePin(A_GPIO_Port, digits[num], GPIO_PIN_RESET);
148     }
149 }
```

Link Github bài 4 và 5: <https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B4%2B5>

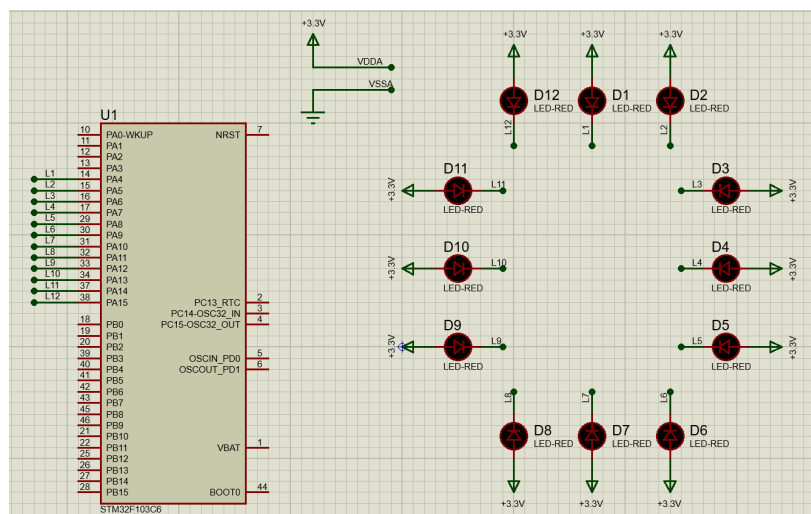
## 5 Bài 6

In this exercise, a new Proteus schematic is designed to simulate an analog clock, with 12 different number. The connections for 12 LEDs are supposed from PA4 to PA15 of the STM32. The arrangement of 12 LEDs is depicted as follows.



Hình 6: 12 LEDs for an analog clock

### Report 1:



Hình 7: Schematic of exercise 6 - 10

### Report 2:

```
1 #define LED_COUNT 12
2 #define LED_SINK 1
3
4 static const uint16_t LED_PINS[LED_COUNT] = {
5     Chan_1_Pin, Chan_2_Pin, Chan_3_Pin, Chan_4_Pin,
6     Chan_5_Pin, Chan_6_Pin, Chan_7_Pin, Chan_8_Pin,
7     Chan_9_Pin, Chan_10_Pin, Chan_11_Pin, Chan_12_Pin
8 };
```

```
9 static GPIO_TypeDef* const LED_PORT = Chan_1_GPIO_Port;
10 static const GPIO_PinState LED_ON = (LED_SINK ? GPIO_PIN_RESET : GPIO_PIN_SET);
11 static const GPIO_PinState LED_OFF = (LED_SINK ? GPIO_PIN_SET : GPIO_PIN_RESET);
12
13 static void AllLEDsOff(void) {
14     for (int i = 0; i < LED_COUNT; ++i) {
15         HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_OFF);
16     }
17 }
18
19 static void AllLEDsOn(void) {
20     for (int i = 0; i < LED_COUNT; ++i) {
21         HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_ON);
22     }
23 }
24
25 #define ALL_ON_MS    1000U
26 #define ALL_OFF_MS   1000U
27 #define STEP_MS      1000U
28
29     AllLEDsOn();
30     HAL_Delay(ALL_ON_MS);
31     AllLEDsOff();
32     HAL_Delay(ALL_OFF_MS);
33
34 while (1) {
35     for (int i = 0; i < LED_COUNT; ++i) {
36         AllLEDsOff();
37         HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_ON);
38         HAL_Delay(STEP_MS);
39     }
40
41     HAL_GPIO_WritePin(LED_PORT, LED_PINS[LED_COUNT - 1], LED_OFF);
42     HAL_Delay(1000);
43
44     AllLEDsOn();
45     HAL_Delay(ALL_ON_MS);
46
47     AllLEDsOff();
48     HAL_Delay(ALL_OFF_MS);
49 }
```

## 6 Bài 7

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

I used the `AllLEDsOff(void)` function to turn off all the LEDs and this is done in conjunction with exercise 6.

```
1 // Main implementation of the function to turn off all 12 LEDs at once
2 static void AllLEDsOff(void) {
3     for (int i = 0; i < LED_COUNT; ++i) {
4         HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_OFF);
5     }
6 }
7
```

```
8 // Initialize or initialize all LEDs to off state before entering main loop,
   ensuring all LEDs are off at the same time.
9 AllLEDsOn();
10 HAL_Delay(ALL_ON_MS);
11 AllLEDsOff();
12 HAL_Delay(ALL_OFF_MS);
13
14 while (1) {
15     // Turn off all LEDs at once before turning on a specific LED
16     for (int i = 0; i < LED_COUNT; ++i) {
17         AllLEDsOff();
18         HAL_GPIO_WritePin(LED_PORT, LED_PINS[i], LED_ON);
19         HAL_Delay(STEP_MS);
20     }
21
22     HAL_GPIO_WritePin(LED_PORT, LED_PINS[LED_COUNT - 1], LED_OFF);
23     HAL_Delay(1000);
24
25     AllLEDsOn();
26     HAL_Delay(ALL_ON_MS);
27
28     //Turn off all LEDs at the same time, used to end an LED running cycle,
       returning the system to the off state before starting a new cycle
29     AllLEDsOff();
30     HAL_Delay(ALL_OFF_MS);
31 }
```

## 7 Bài 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 // LED_ON / LED_OFF macros
2 #define LED_ACTIVE_HIGH 0
3 #if LED_ACTIVE_HIGH
4     #define LED_ON(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_SET)
5     #define LED_OFF(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_RESET)
6 #else
7     #define LED_ON(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_RESET)
8     #define LED_OFF(port,pin) HAL_GPIO_WritePin((port),(pin),GPIO_PIN_SET)
9 #endif
10
11 // LED pin and port mapping array
12 static GPIO_TypeDef* led_ports[12] = {
13     LED1_PORT, LED2_PORT, LED3_PORT, LED4_PORT,
14     LED5_PORT, LED6_PORT, LED7_PORT, LED8_PORT,
15     LED9_PORT, LED10_PORT, LED11_PORT, LED12_PORT
16 };
17 static const uint16_t led_pins[12] = {
18     Led_1_Pin, Led_2_Pin, Led_3_Pin, Led_4_Pin,
19     Led_5_Pin, Led_6_Pin, Led_7_Pin, Led_8_Pin,
20     Led_9_Pin, Led_10_Pin, Led_11_Pin, Led_12_Pin
21 };
22
23 void setNumberOnClock(int num);
24
25 // Cycle 1: Turn ON only the LED corresponding to num
26 void setNumberOnClock(int num) {
27     if (num < 0 || num > 11) return;
28     for (int i = 0; i < 12; ++i) {
```

```
29     if (i == num)
30         LED_ON(led_ports[i], led_pins[i]);
31     else
32         LED_OFF(led_ports[i], led_pins[i]);
33 }
34 }
35
36 int main(void) {
37     // Cycle 1: Run once -> light a single LED
38     setNumberOnClock(3);
39     HAL_Delay(1000);
40     signalCycleEnd(1, 300, 300);
41 }
```

## 8 Bài 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  // LED_ON / LED_OFF macros
2  #define LED_ACTIVE_HIGH 0
3  #if LED_ACTIVE_HIGH
4      #define LED_ON(port, pin) HAL_GPIO_WritePin((port), (pin), GPIO_PIN_SET)
5      #define LED_OFF(port, pin) HAL_GPIO_WritePin((port), (pin), GPIO_PIN_RESET)
6  #else
7      #define LED_ON(port, pin) HAL_GPIO_WritePin((port), (pin), GPIO_PIN_RESET)
8      #define LED_OFF(port, pin) HAL_GPIO_WritePin((port), (pin), GPIO_PIN_SET)
9  #endif
10
11 // LED pin and port mapping array
12 static GPIO_TypeDef* led_ports[12] = {
13     LED1_PORT, LED2_PORT, LED3_PORT, LED4_PORT,
14     LED5_PORT, LED6_PORT, LED7_PORT, LED8_PORT,
15     LED9_PORT, LED10_PORT, LED11_PORT, LED12_PORT
16 };
17 static const uint16_t led_pins[12] = {
18     Led_1_Pin, Led_2_Pin, Led_3_Pin, Led_4_Pin,
19     Led_5_Pin, Led_6_Pin, Led_7_Pin, Led_8_Pin,
20     Led_9_Pin, Led_10_Pin, Led_11_Pin, Led_12_Pin
21 };
22
23 void clearNumberOnClock(int num);
24
25 // Cycle 2: Turn OFF only the LED corresponding to num; turn ON all others.
26
27 void clearNumberOnClock(int num) {
28     if (num < 0 || num > 11) return;
29     for (int i = 0; i < 12; ++i) {
30         if (i == num)
31             LED_OFF(led_ports[i], led_pins[i]);
32         else
33             LED_ON(led_ports[i], led_pins[i]);
34     }
35 }
36
37 int main(void) {
38     // Cycle 2: Run once -> clear a single LED
39     clearNumberOnClock(7);
40     HAL_Delay(1000);
41     signalCycleEnd(1, 300, 300);
```

42 }

## 9 Bài 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 // Cycle 3 step delay in milliseconds
2 #define CYCLE3_STEP_MS 50
3
4 // RTC-like internal time for Cycle 3
5 static int rtc_hour = 0;
6 static int rtc_minute = 0;
7 static int rtc_second = 0;
8
9 void displayClockRealtime(void);
10
11 while (1) {
12     // DisplayClockRealtime includes the delay and time increment
13     displayClockRealtime();
14 }
15
16 void displayClockRealtime(void) {
17     int h = (rtc_hour % 12 + 12) % 12;
18     int m = ((rtc_minute / 5) % 12 + 12) % 12;
19     int s = ((rtc_second / 5) % 12 + 12) % 12;
20
21     // Turn all off first
22     for (int i = 0; i < 12; ++i)
23         LED_OFF(led_ports[i], led_pins[i]);
24
25     // Turn on hour, minute, second LEDs
26     LED_ON(led_ports[h], led_pins[h]);
27     LED_ON(led_ports[m], led_pins[m]);
28     LED_ON(led_ports[s], led_pins[s]);
29
30     // Wait one second (real)
31     HAL_Delay(CYCLE3_STEP_MS);
32
33     // Increment time
34     rtc_second++;
35     if (rtc_second >= 60) {
36         rtc_second = 0;
37         rtc_minute++;
38     }
39     if (rtc_minute >= 60) {
40         rtc_minute = 0;
41         rtc_hour++;
42     }
43     if (rtc_hour >= 24) {
44         rtc_hour = 0;
45     }
46 }
```

Link Github bài 6 - 10: <https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main/B6-10>



## 10 Link Github toàn bộ LAB 1

<https://github.com/SangNguyen-232/VXLVDK-LAB1/tree/main>