

TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC QUỐC GIA TP.HCM  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



**VI XỬ LÝ - VI ĐIỀU KHIỂN (TN) (CO3010)**

---

**Lab 1**

## **LED Animations**

---

GVHD: Lê Trọng Nhân  
Cao Tiến Đạt

Sinh viên: Huỳnh Bùi Ngọc Khoa - 2211589

TP.Hồ Chí Minh, 09/2024



## Contents

1	Exercise 1	2
2	Exercise 2	3
3	Exercise 3	4
4	Exercise 4	6
5	Exercise 5	8
6	Exercise 6	10
7	Exercise 7	11
8	Exercise 8	11
9	Exercise 9	11
10	Exercise 10	12

## 1 Exercise 1

**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).

*Note:* For this Report 1, I will combine the schematic in exercise 1 and exercise 2 in one figure below.

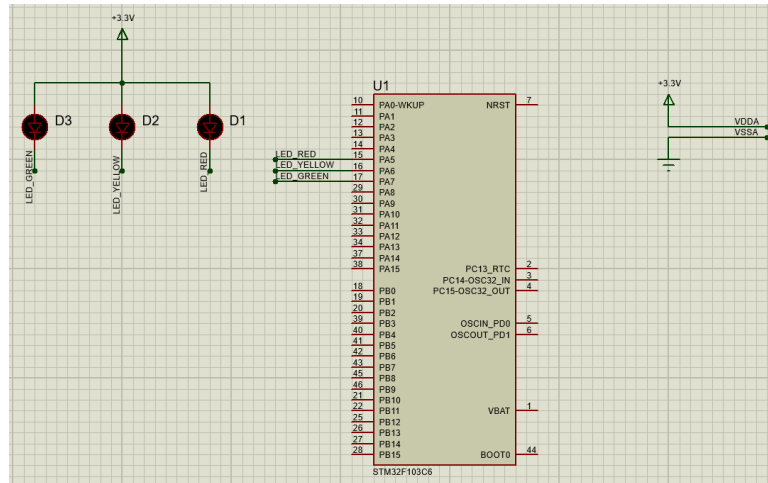


Figure 1: [Link to the figure](#)

**Report 2:** Present the source code in the infinite 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 int cnt = 0;
2 while (1) {
3     switch (cnt) {
4         case 4:
5             cnt = 0;
6         case 0:
7             HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, RESET);
8             HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, SET);
9             break;
10        case 2:
11            HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
12            HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, RESET);
13            break;
14        default:
15            break;
16    }
17    cnt++;
18    HAL_Delay(1000);
19 }
```

Listing 1: Source code of while function

## 2 Exercise 2

**Report 1:** Present the schematic.

*Checkout Report 1 of Exercise 1.*

**Report 2:** Present the source code in while.

```
1 void set_LEDS(GPIO_PinState redstate, GPIO_PinState yellowstate, GPIO_PinState  
   greenstate) {  
2     HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, redstate);  
3     HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, yellowstate);  
4     HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, greenstate);  
5 }
```

Listing 2: Source code of set\_LEDS function

```
1 int cnt = 10;  
2 while (1) {  
3     switch (cnt) {  
4         case 10:  
5             set_LEDS(1, 1, 0);  
6             break;  
7         case 7:  
8             set_LEDS(1, 0, 1);  
9             break;  
10        case 5:  
11            set_LEDS(0, 1, 1);  
12            break;  
13        case 1:  
14            cnt = 11;  
15        default:  
16            break;  
17    }  
18    cnt--;  
19    HAL_Delay(1000);  
20 }
```

Listing 3: Source code of while function

### 3 Exercise 3

Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light. A reference design can be found in the figure bellow.

*Note: The below schematic is used for Exercise 3, 4 and 5.*

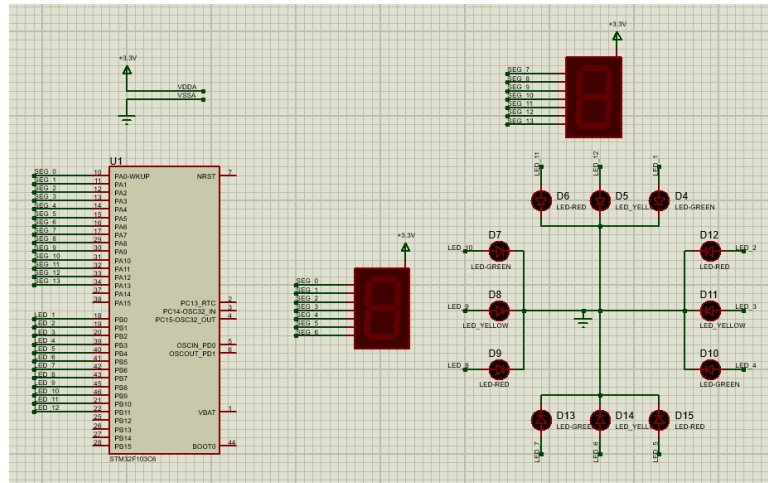


Figure 2: [Link to the figure](#)

```
1 GPIO_TypeDef *LED_PORTS[12] = { LED_1_GPIO_Port, LED_2_GPIO_Port,
2   LED_3_GPIO_Port, LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
3   LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port, LED_10_GPIO_Port,
4   LED_11_GPIO_Port, LED_12_GPIO_Port };
5 uint16_t LED_PINS[12] = { LED_1_Pin, LED_2_Pin, LED_3_Pin, LED_4_Pin,
6   LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin, LED_9_Pin, LED_10_Pin,
7   LED_11_Pin, LED_12_Pin };
8 GPIO_PinState LEDState[4][12] = { { 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0 },
9   { 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1 }, { 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
10   1, 1, 0 }, { 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0 } };
11 void set_LEDS(const GPIO_PinState *state) {
12     for (int i = 0; i < 12; i++) {
13         HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], state[i]);
14     }
15 }
```

Listing 4: Source code of set\_LEDS function

```
1 int cnt = 10;
2 while (1) {
3     switch (cnt) {
4     case 10:
5         set_LEDS(LEDState[0]);
6         break;
7     case 7:
8         set_LEDS(LEDState[1]);
9         break;
10    case 5:
11        set_LEDS(LEDState[2]);
12        break;
13    case 2:
```

```
14         set_LEDS(LEDState[3]);  
15         break;  
16     case 1:  
17         cnt = 11;  
18         break;  
19     default:  
20         break;  
21     }  
22     cnt--;  
23     HAL_Delay(1000);  
24 }
```

Listing 5: Source code of while function

## 4 Exercise 4

**Report 1:** Present the schematic.

*Checkout the schematic of Exercise 3*

**Report 2:** Present the source code for display7SEG function.

```
1 GPIO_TypeDef *SEG_PORTS[12] = { SEG_0_GPIO_Port, SEG_1_GPIO_Port,
2     SEG_2_GPIO_Port, SEG_3_GPIO_Port, SEG_4_GPIO_Port, SEG_5_GPIO_Port,
3     SEG_6_GPIO_Port };
4 uint16_t SEG_PINS[12] = { SEG_0_Pin, SEG_1_Pin, SEG_2_Pin, SEG_3_Pin,
5     SEG_4_Pin, SEG_5_Pin, SEG_6_Pin };
6 GPIO_PinState LEDS_state[10][7] = { { 0, 0, 0, 0, 0, 0, 1 }, { 1, 0, 0, 1,
7     1, 1, 1 }, { 0, 0, 1, 0, 0, 1, 0 }, { 0, 0, 0, 0, 1, 1, 0 }, { 1, 0,
8     0, 1, 1, 0, 0 }, { 0, 1, 0, 0, 1, 0, 0 }, { 0, 1, 0, 0, 0, 0, 0 }, {
9     0, 0, 0, 1, 1, 1, 1 }, { 0, 0, 0, 0, 0, 0, 0 }, { 0, 0, 0, 0, 1, 0,
10    0 } };
11 void set_LEDS(GPIO_PinState *L_LEDS_state) {
12     for (int i = 0; i < 7; i++) {
13         HAL_GPIO_WritePin(SEG_PORTS[i], SEG_PINS[i], L_LEDS_state[i]);
14     }
15 }
16 void display7SEG(int number) {
17     switch (number) {
18     case 0:
19         set_LEDS(LEDState[0]);
20         break;
21     case 1:
22         set_LEDS(LEDState[1]);
23         break;
24     case 2:
25         set_LEDS(LEDState[2]);
26         break;
27     case 3:
28         set_LEDS(LEDState[3]);
29         break;
30     case 4:
31         set_LEDS(LEDState[4]);
32         break;
33     case 5:
34         set_LEDS(LEDState[5]);
35         break;
36     case 6:
37         set_LEDS(LEDState[6]);
38         break;
39     case 7:
40         set_LEDS(LEDState[7]);
41         break;
42     case 8:
43         set_LEDS(LEDState[8]);
44         break;
45     case 9:
46         set_LEDS(LEDState[9]);
47         break;
48     default:
49         break;
50     }
51 }
```

Listing 6: Source code of set\_LEDS function and display7SEG function



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

Listing 7: Source code of while function



## 5 Exercise 5

```
1 GPIO_TypeDef *LED_PORTS[12] = { LED_1_GPIO_Port, LED_2_GPIO_Port,
2 LED_3_GPIO_Port, LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
3 LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port, LED_10_GPIO_Port,
4 LED_11_GPIO_Port, LED_12_GPIO_Port };
5 uint16_t LED_PINS[12] = { LED_1_Pin, LED_2_Pin, LED_3_Pin, LED_4_Pin,
6 LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin, LED_9_Pin, LED_10_Pin,
7 LED_11_Pin, LED_12_Pin };
8 GPIO_PinState LEDS_state[4][12] = { { 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0 },
9     { 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1 }, { 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
10     1, 1, 0 }, { 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0 } };
11 void set_LEDS(const GPIO_PinState *state) {
12     for (int i = 0; i < 12; i++) {
13         HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], state[i]);
14     }
15 }
16 GPIO_TypeDef *SEG_PORTS[14] = { SEG_0_GPIO_Port, SEG_1_GPIO_Port,
17 SEG_2_GPIO_Port, SEG_3_GPIO_Port, SEG_4_GPIO_Port, SEG_5_GPIO_Port,
18 SEG_6_GPIO_Port, SEG_7_GPIO_Port, SEG_8_GPIO_Port, SEG_9_GPIO_Port,
19 SEG_10_GPIO_Port, SEG_11_GPIO_Port, SEG_12_GPIO_Port,
20 SEG_13_GPIO_Port };
21 uint16_t SEG_PINS[14] = { SEG_0_Pin, SEG_1_Pin, SEG_2_Pin, SEG_3_Pin,
22 SEG_4_Pin, SEG_5_Pin, SEG_6_Pin, SEG_7_Pin, SEG_8_Pin, SEG_9_Pin,
23 SEG_10_Pin, SEG_11_Pin, SEG_12_Pin, SEG_13_Pin };
24 GPIO_PinState LEDS_7SEG_state[6][7] = { { 0, 0, 0, 0, 0, 0, 1 }, { 1, 0, 0,
25     1, 1, 1, 1 }, { 0, 0, 1, 0, 0, 1, 0 }, { 0, 0, 0, 0, 1, 1, 0 }, { 1,
26     0, 0, 1, 1, 0, 0 }, { 0, 1, 0, 0, 1, 0, 0 } };
27 void set_7SEG_X(const GPIO_PinState *L_LEDS_X_state) {
28     for (int i = 0; i < 7; i++) {
29         HAL_GPIO_WritePin(SEG_PORTS[i], SEG_PINS[i], L_LEDS_X_state[i]);
30     }
31 }
32 void set_7SEG_Y(const GPIO_PinState *L_LEDS_Y_state) {
33     for (int i = 0; i < 7; i++) {
34         HAL_GPIO_WritePin(SEG_PORTS[i + 7], SEG_PINS[i + 7],
35             L_LEDS_Y_state[i]);
36     }
37 }
```

Listing 8: Source code of set\_LEDS, set\_7SEG\_X and set\_7SEG\_Y function

```
1 int cnt = 10;
2 while (1) {
3     switch (cnt) {
4         case 10:
5             set_LEDS(LEDS_state[0]);
6             set_7SEG_Y(LEDS_7SEG_state[3]);
7             set_7SEG_X(LEDS_7SEG_state[5]);
8             break;
9         case 9:
10            set_7SEG_Y(LEDS_7SEG_state[2]);
11            set_7SEG_X(LEDS_7SEG_state[4]);
12            break;
13        case 8:
14            set_7SEG_Y(LEDS_7SEG_state[1]);
15            set_7SEG_X(LEDS_7SEG_state[3]);
16            break;
17        case 7:
18            set_LEDS(LEDS_state[1]);
19            set_7SEG_Y(LEDS_7SEG_state[2]);
```

```
20     set_7SEG_X(LEDState[2]);
21     break;
22 case 6:
23     set_7SEG_Y(LEDState[1]);
24     set_7SEG_X(LEDState[1]);
25     break;
26 case 5:
27     set_LED(LEDState[2]);
28     set_7SEG_Y(LEDState[5]);
29     set_7SEG_X(LEDState[3]);
30     break;
31 case 4:
32     set_7SEG_Y(LEDState[4]);
33     set_7SEG_X(LEDState[2]);
34     break;
35 case 3:
36     set_7SEG_Y(LEDState[3]);
37     set_7SEG_X(LEDState[1]);
38     break;
39 case 2:
40     set_LED(LEDState[3]);
41     set_7SEG_Y(LEDState[2]);
42     set_7SEG_X(LEDState[2]);
43     break;
44 case 1:
45     set_7SEG_Y(LEDState[1]);
46     set_7SEG_X(LEDState[1]);
47     cnt = 11;
48     break;
49 default:
50     break;
51 }
52 cnt--;
53 HAL_Delay(1000);
54 }
```

Listing 9: Source code of while function

## 6 Exercise 6

**Report 1:** Present the schematic.

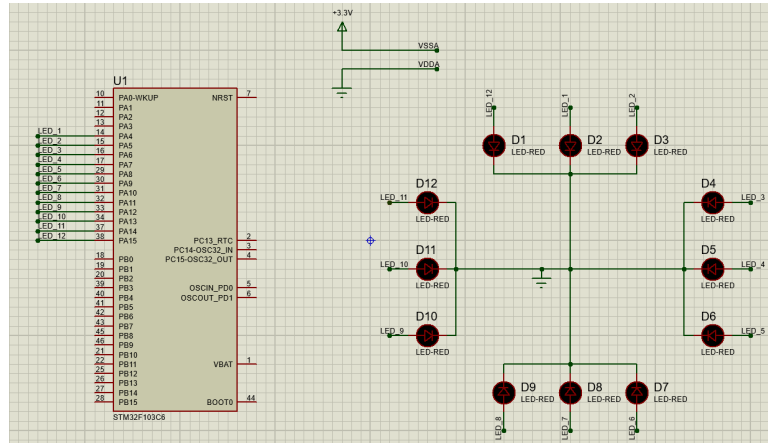


Figure 3: [Link to the figure](#)

**Report 2:** Implement a simple program to test the connection of every single LED. This testing program should turn every LED in a sequence.

```
1 GPIO_TypeDef *LED_PORTS[12] = { LED_1_GPIO_Port, LED_2_GPIO_Port,
2   LED_3_GPIO_Port, LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
3   LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port, LED_10_GPIO_Port,
4   LED_11_GPIO_Port, LED_12_GPIO_Port };
5 uint16_t LED_PINS[12] = { LED_1_Pin, LED_2_Pin, LED_3_Pin, LED_4_Pin,
6   LED_5_Pin, LED_6_Pin, LED_7_Pin, LED_8_Pin, LED_9_Pin, LED_10_Pin,
7   LED_11_Pin, LED_12_Pin };
8 }
```

Listing 10: Source code of variables declaration

```
1 int cnt = 0;
2 while (1) {
3     if (cnt >= 12) {
4         cnt = 0;
5     }
6     HAL_GPIO_WritePin(LED_PORTS[cnt], LED_PINS[cnt], 1);
7     HAL_Delay(1000);
8     HAL_GPIO_WritePin(LED_PORTS[cnt], LED_PINS[cnt], 0);
9     cnt++;
10 }
```

Listing 11: Source code of while function



## 7 Exercise 7

```
1 void clearAllClock() {  
2     for (int i = 0; i < 12; i++) {  
3         HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], 0);  
4     }  
5 }
```

Listing 12: Source code of clearAllClock function

## 8 Exercise 8

```
1 void setNumberOnClock(int num) {  
2     HAL_GPIO_WritePin(LED_PORTS[num], LED_PINS[num], 1);  
3 }
```

Listing 13: Source code of setNumberOnClock function

## 9 Exercise 9

```
1 void clearNumberOnClock(int num) {  
2     HAL_GPIO_WritePin(LED_PORTS[num], LED_PINS[num], 0);  
3 }
```

Listing 14: Source code of clearNumberOnClock

## 10 Exercise 10

```
1 int cnt_hour = 0;
2 int cnt_min = 0;
3 int cnt_sec = 0;
4 while (1) {
5     setNumberOnClock(cnt_hour);
6     setNumberOnClock(cnt_min / 5);
7     setNumberOnClock(cnt_sec / 5);
8     HAL_Delay(1000);
9     cnt_sec++;
10    if (cnt_sec == 60) {
11        cnt_sec = 0;
12        cnt_min++;
13    }
14    if (cnt_min == 60) {
15        cnt_min = 0;
16        cnt_hour++;
17    }
18    if (cnt_hour == 12) {
19        cnt_hour = 0;
20    }
21    clearAllClock();
22 }
```

Listing 15: Source code of while function