

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



## Microprocessor - Microcontroller

---

### Lab Report - CO3010

# Lab 1

---

Advisor(s): Phan Văn Sỹ

Student(s): Chu Le Hoang 2352346

HO CHI MINH CITY, SEPTEMBER 2025





## Contents

<b>1</b>	<b>Overall</b>	<b>5</b>
<b>2</b>	<b>Exercise 1</b>	<b>6</b>
2.1	Report 1 . . . . .	6
2.2	Report 2 . . . . .	6
<b>3</b>	<b>Exercise 2</b>	<b>7</b>
3.1	Report 1 . . . . .	7
3.2	Report 2 . . . . .	7
<b>4</b>	<b>Exercise 3</b>	<b>10</b>
4.1	Report 1 . . . . .	10
4.2	Report 2 . . . . .	10
<b>5</b>	<b>Exercise 4</b>	<b>11</b>
5.1	Report 1 . . . . .	11
5.2	Report 2 . . . . .	11
<b>6</b>	<b>Exercise 5</b>	<b>12</b>
6.1	Report 1 . . . . .	12
6.2	Report 2 . . . . .	12
<b>7</b>	<b>Exercise 6</b>	<b>14</b>
7.1	Report 1 . . . . .	14
7.2	Report 2 . . . . .	14
<b>8</b>	<b>Exercise 7</b>	<b>15</b>
8.1	Report 1 . . . . .	15
8.2	Report 2 . . . . .	15
<b>9</b>	<b>Exercise 8</b>	<b>16</b>
9.1	Report 1 . . . . .	16
9.2	Report 2 . . . . .	16
<b>10</b>	<b>Exercise 9</b>	<b>17</b>
10.1	Report 1 . . . . .	17
10.2	Report 2 . . . . .	17



<b>11 Exercise 10</b>	<b>18</b>
11.1 Report 1 . . . . .	18
11.2 Report 2 . . . . .	18

# 1 Overall

Lab schematics are submitted via GitHub link: <https://github.com/KennyLe298/MPU-MCU>

For most of the exercises, the default while(1) code segment is as follows :

```
1 while(1){  
2     //Insert function for each exercise  
3     HAL_Delay(1000);  
4 }
```

The schematic for the exercises from 1 to 5 is located here:

<https://github.com/KennyLe298/MPU-MCU/tree/main/schematics>

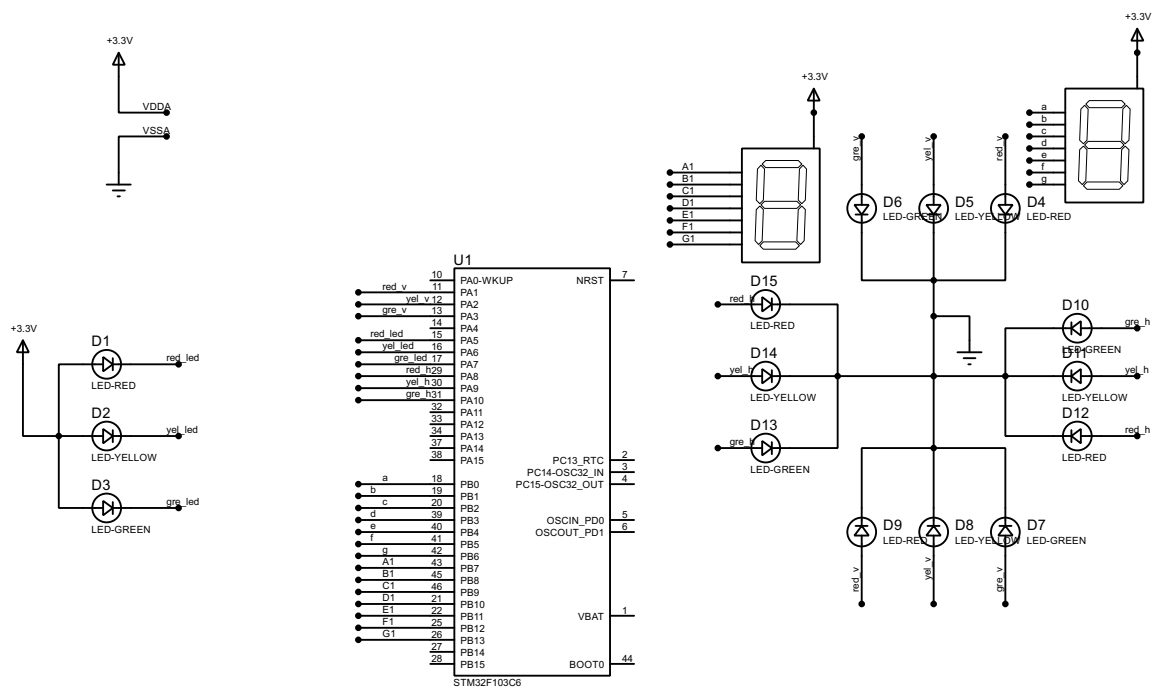


Figure 1.1: Exercise 1 to 5 schematic

The schematic for the exercises from 6 to 10 is located here:

<https://github.com/KennyLe298/MPU-MCU/tree/main/schematics>

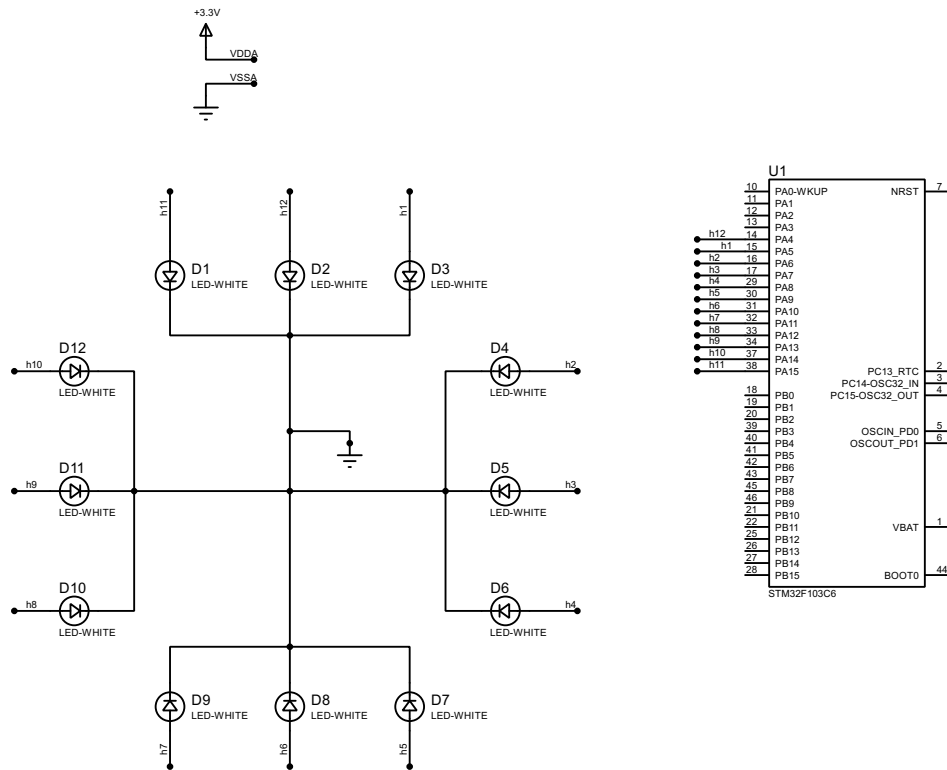


Figure 1.2: Exercise 6 to 10 schematic

## 2 Exercise 1

### 2.1 Report 1

Schematic is in **Overall** section.

### 2.2 Report 2

The following code segment is from exercise 1 header file:

```
1 void LED_toggle(){
2 HAL_GPIO_TogglePin(LED_RED_GPIO_Port,LED_RED_Pin);
3 HAL_GPIO_TogglePin(YELLOW_LED_GPIO_Port,YELLOW_LED_Pin);
4 }
5 //this is a function to toggle the leds
6
7 void initStage(){
8 HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, GPIO_PIN_RESET);
```

```
9 HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port, YELLOW_LED_Pin,
    GPIO_PIN_SET);
10 }
11 //this is a function to set the first stage before starting to
    toggle
12
13 void ex1(){
14     static unsigned int init = 0; //a variable to initialize
15     static unsigned int count = 0; //a variable to count seconds
16
17     if(init == 0){
18         initStage();
19         init++;
20     }
21     if(count >= 3){ //toggle after 2 seconds so count >=3
22         count = 0;
23         LED_toggle();
24     }
25     else ++count;
26 }
```

## 3 Exercise 2

### 3.1 Report 1

Schematic is in **Overall** section.

### 3.2 Report 2

```
1 void initStage(){
2     HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, GPIO_PIN_RESET);
3     HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port, YELLOW_LED_Pin,
        GPIO_PIN_SET);
4     HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, GPIO_PIN_SET
        );
5 } //this function is for initial stage, setting red led on, yellow
    and green off
```



```
6
7 void StageYellow(){
8 HAL_GPIO_TogglePin(LED_RED_GPIO_Port , LED_RED_Pin);
9 HAL_GPIO_TogglePin(YELLOW_LED_GPIO_Port , YELLOW_LED_Pin);
10 }
11
12 // next stage turn off red and turn on yellow
13
14 void StageGreen(){
15 HAL_GPIO_TogglePin(YELLOW_LED_GPIO_Port , YELLOW_LED_Pin);
16 HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port , LED_GREEN_Pin);
17 }
18
19 // next stage turn off yellow and turn on green
20
21 void StageRed(){
22 HAL_GPIO_TogglePin(LED_GREEN_GPIO_Port , LED_GREEN_Pin);
23 HAL_GPIO_TogglePin(LED_RED_GPIO_Port , LED_RED_Pin);
24 }
25
26 //next stage turn off green and turn on red
27
28 void ex2(){
29 static unsigned init = 0; //initialize variable
30 static unsigned count = 0; //keep track of second variable
31 static unsigned stage = 0; //keep track of stage variable
32
33 if(init == 0) {
34     initStage();
35     stage = 1;
36     init++;
37 }
38
39 switch(stage){
40 case 1:
41 {
42     if(count == 5){
43         stage = 2;
```





```
44     StageYellow();
45     count = 0;
46 }
47 break;
48 }
49 case 2:
50 {
51     if(count == 3){
52         stage = 3;
53         StageGreen();
54         count = 0;
55     }
56     break;
57 }
58 case 3:
59 {
60     if(count == 2){
61         stage = 1;
62         StageRed();
63         count = 0;
64     }
65     break;
66 }
67 }
68 ++count;
69 }
```



## 4 Exercise 3

### 4.1 Report 1

Schematic is in **Overall** section.

### 4.2 Report 2

Because the coding of this exercise is quite long and reuse the same logic from Exercise 2 then for this one, I will use pseudocode for general idea.

```
1 Initial:
2 init = 0
3 ver_s, hor_s = 0
4 ver_c, hor_c = 0
5
6 function ex3():
7 IF init == 0:
8     init = 1
9     Vertical = GREEN ON
10    Horizontal = RED ON
11
12 // Vertical state machine
13 if ver_s == 1 and ver_c >= 3: (GREEN -> YELLOW)
14     ver_s = 2; ver_c = 0
15 else if ver_s == 2 and ver_c >= 2: (YELLOW -> RED)
16     ver_s = 3; ver_c = 0
17 else if ver_s == 3 and ver_c >= 5: (RED -> GREEN)
18     ver_s = 1; ver_c = 0
19
20 // Horizontal state machine
21 if hor_s == 1 and hor_c >= 5: (RED -> GREEN)
22     hor_s = 2; hor_c = 0
23 else if hor_s == 2 and hor_c >= 3: (GREEN -> YELLOW)
24     hor_s = 3; hor_c = 0
25 else if hor_s == 3 and hor_c >= 2: (YELLOW -> RED)
26     hor_s = 1; hor_c = 0
27
28 ver_c++, hor_c++
```



## 5 Exercise 4

### 5.1 Report 1

Schematic is in **Overall** section.

### 5.2 Report 2

```
1 int segmentmap[10][7] = {
2     {0,0,0,0,0,0,1}, // 0
3     {1,0,0,1,1,1,1}, // 1
4     {0,0,1,0,0,1,0}, // 2
5     {0,0,0,0,1,1,0}, // 3
6     {1,0,0,1,1,0,0}, // 4
7     {0,1,0,0,1,0,0}, // 5
8     {0,1,0,0,0,0,0}, // 6
9     {0,0,0,1,1,1,1}, // 7
10    {0,0,0,0,0,0,0}, // 8
11    {0,0,0,0,1,0,0}  // 9
12 };
13
14 void display7SEG(int num){
15     if (num < 0 || num > 9){
16         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_All, GPIO_PIN_SET);
17         //turn of all segment if wrong input
18         return;
19     }
20     for (int i = 0; i < 7; i++){
21         HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0 << i, segmentmap[num][
22             i]);
23     }
24     // inputs a,b,c,d,e,f,g of the 7SEG LED is from PB0-6 so we
25     // can shift right for simplification
```



## 6 Exercise 5

### 6.1 Report 1

Schematic is in **Overall** section.

### 6.2 Report 2

```
1 //We can reuse the display7SEG function from exercise 4 for one
  route
2 //Therefore we have to make another display7SEG function for the
  perpendicular route
3 void display7SEG2(int num){
4   if(num >= 0 && num <= 9){
5     for(int i = 7; i < 14; i++){
6       HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0 << i, segmentmap[num -
          1][i]);
7     }
8   }
9 }
10 //Implementing both functions and reuse the traffic lights from
    ex3:
11 void ex5(){
12   ex3();
13   switch(ver_s){
14   case 1:
15     display7SEG(3 - ver_c); //use subtraction here is for counting
        down the timer
16     break;
17   case 2:
18     display7SEG(2 - ver_c);
19     break;
20   case 3:
21     display7SEG(5 - ver_c);
22     break;
23   }
24   switch(hor_s){
25   case 1:
```



```
26     display7SEG2(5 - hor_c);
27     break;
28 case 2:
29     display7SEG2(3 - hor_c);
30     break;
31 case 3:
32     display7SEG2(2 - hor_c);
33     break;
34 }
35 }
```



## 7 Exercise 6

### 7.1 Report 1

Schematic is in **Overall** section.

### 7.2 Report 2

```
1 //initialize the clock pins
2 uint16_t hourPins[12] = { h12_Pin, h1_Pin, h2_Pin, h3_Pin, h4_Pin,
   h5_Pin,
3 h6_Pin, h7_Pin, h8_Pin, h9_Pin, h10_Pin, h11_Pin };
4
5 void testLEDs(void){
6   for(int i = 0; i < 12; i++){
7     for(int j = 0; j < 12; j++){
8       HAL_GPIO_WritePin(GPIOA, hourPins[j], GPIO_PIN_RESET);
9     }
10    HAL_GPIO_WritePin(GPIOA, hourPins[i], GPIO_PIN_SET);
11    HAL_Delay(5000); //change every 5s
12  }
13 }
```



## 8 Exercise 7

### 8.1 Report 1

Schematic is in **Overall** section.

### 8.2 Report 2

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



## 9 Exercise 8

### 9.1 Report 1

Schematic is in **Overall** section.

### 9.2 Report 2

```
1 void setNumberOnClock(int num){  
2     if(num < 0 || num > 11) return;  
3     HAL_GPIO_WritePin(GPIOA, hourPins[num], GPIO_PIN_SET);  
4 }
```





## 10 Exercise 9

### 10.1 Report 1

Schematic is in **Overall** section.

### 10.2 Report 2

```
1 void clearNumberOnClock(int num){  
2     if (num < 0 || num > 11) return;  
3     HAL_GPIO_WritePin(GPIOA, hourPins[num], GPIO_PIN_RESET);  
4 }
```



## 11 Exercise 10

### 11.1 Report 1

Schematic is in **Overall** section.

### 11.2 Report 2

```
1 //global time variables
2 int sec = 0;
3 int min = 0;
4 int hrs = 0;
5
6 // clock simulation
7 void Clock(void) {
8     ++sec;
9     if (sec > 59) {
10         sec = 0;
11         ++min;
12     }
13     if (min > 59) {
14         min = 0;
15         ++hrs;
16     }
17     if (hrs > 23) {
18         hrs = 0;
19     }
20     clearAllClock();
21
22     int s_idx = sec / 5;
23     int m_idx = min / 5;
24     int h_idx = hrs % 12;
25     //changing hours, minute, second so that it shows on the clock
26     setNumberOnClock(s_idx);
27     setNumberOnClock(m_idx);
28     setNumberOnClock(h_idx);
29 }
```