

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): Lương Đức Hiếu 2352324

HO CHI MINH CITY, SEPTEMBER 2025



Contents

1 Overall	4
2 Exercise 1	5
2.1 Report 1	5
2.2 Report 2	5
3 Exercise 2	6
3.1 Report 1	6
3.2 Report 2	6
4 Exercise 3	8
4.1 Report 1	8
4.2 Report 2	8
5 Exercise 4	10
5.1 Report 1	10
5.2 Report 2	10
6 Exercise 5	11
6.1 Report 1	11
6.2 Report 2	11
7 Exercise 6	13
7.1 Report 1	13
7.2 Report 2	13
8 Exercise 7	14
8.1 Report 1	14
8.2 Report 2	14
9 Exercise 8	15
9.1 Report 1	15
9.2 Report 2	15
10 Exercise 9	16
10.1 Report 1	16
10.2 Report 2	16



11 Exercise 10	17
11.1 Report 1	17
11.2 Report 2	17

1 Overall

Lab schematics are submitted via GitHub link: <https://github.com/hieuld1003/MPU-MCU>

The default while(1) code for most of the exercise is:

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

The schematic for the exercises from 1 to 5 and 6 to 10:

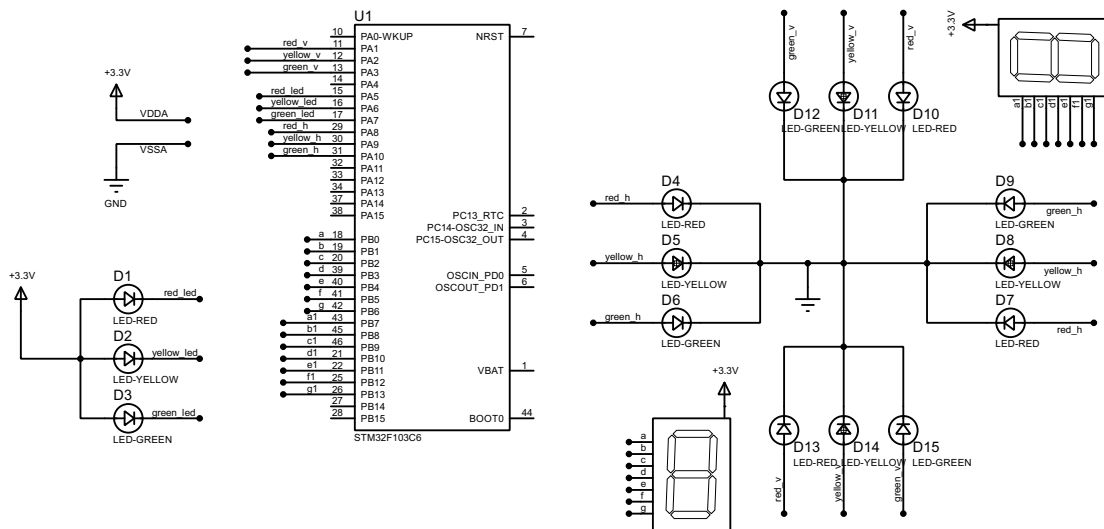


Figure 1.1: Exercise 1 to 5 schematic

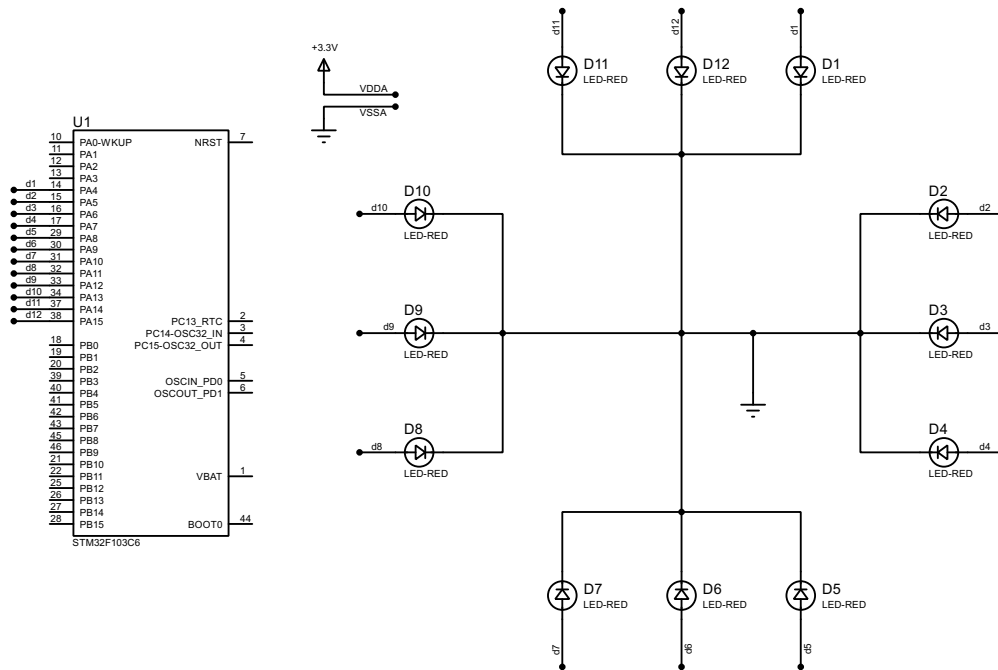


Figure 1.2: Exercise 6 to 10 schematic

2 Exercise 1

2.1 Report 1

Schematic is in **Overall** section.

2.2 Report 2

This is the pseudocode from exercise 1 header file:

```
1 Initialize:
2 setInit = 0
3 count = 0
4
5 Main Loop:
6 while(1) {
7     if (setInit == 0) {
8         // Initialize LEDs: red OFF, yellow ON
```

```
9      setInitLED()
10      setInit = 1
11  }
12
13  if (count == 1) {
14      count = 0
15      // Toggle both red and yellow LEDs
16      toggleLED()
17  } else {
18      count = count + 1
19  }
20
21  // Wait for next iteration
22 }
```

3 Exercise 2

3.1 Report 1

Schematic is in **Overall** section.

3.2 Report 2

This is the pseudocode from exercise 2 header file:

```
1 Initialize:
2 init = 0
3 count = 0
4 phase = 0
5
6 Main Loop:
7 while(1) {
8     if (init == 0) {
9         // Initial state: red ON, yellow OFF, green OFF
10        initial()
11        phase = 1
12        init = 1
13    }
```



```
14
15  switch(phase) {
16      case 1: // Red phase (5 time units)
17          if (count == 5) {
18              Phase2() // Toggle red OFF, yellow ON
19              phase = 2
20              count = 0
21          }
22          break
23
24      case 2: // Yellow phase (3 time units)
25          if (count == 3) {
26              Phase3() // Toggle yellow OFF, green ON
27              phase = 3
28              count = 0
29          }
30          break
31
32      case 3: // Green phase (2 time units)
33          if (count == 2) {
34              Phase1() // Toggle green OFF, red ON
35              phase = 1
36              count = 0
37          }
38          break
39  }
40
41  count = count + 1
42  // Wait for next iteration
43 }
```



4 Exercise 3

4.1 Report 1

Schematic is in **Overall** section.

4.2 Report 2

This is the pseudocode from exercise 3 header file:

```
1 Initialize:
2 init = 0
3 ver_phase = 0, hor_phase = 0
4 ver_count = 0, hor_count = 0
5
6 Main Loop:
7 while(1) {
8     // Initial setup (runs only once)
9     if (init == 0) {
10         Set vertical: green ON, others OFF
11         Set horizontal: red ON, others OFF
12         ver_phase = 1, hor_phase = 1
13         init = 1
14     }
15
16     // Vertical light state machine
17     if (ver_count reaches threshold for current phase) {
18         ver_count = 0
19         Switch to next phase:
20         - If phase 1 (green, 3s): change to phase 2 (yellow)
21         - If phase 2 (yellow, 2s): change to phase 3 (red)
22         - If phase 3 (red, 5s): change to phase 1 (green)
23     }
24
25     // Horizontal light state machine
26     if (hor_count reaches threshold for current phase) {
27         hor_count = 0
28         Switch to next phase:
29         - If phase 1 (red, 5s): change to phase 2 (green)
```




```
30     - If phase 2 (green, 3s): change to phase 3 (yellow)
31     - If phase 3 (yellow, 2s): change to phase 1 (red)
32 }
33
34 ver_count++
35 hor_count++
36 }
```

5 Exercise 4

5.1 Report 1

Schematic is in **Overall** section.

5.2 Report 2

This is the pseudocode from exercise 4 header file:

```
1 int arr[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// a, b, c, d, e, f, g
14
15
16Function display7SEG(num):
17if (0 <= num <= 9) {
18    for (state = 0 to 6) {
19        // Set each segment according to the pattern in arr
20        Set GPIO pin (GPIOB_PIN_state) to arr[num][state]
21    }
22} else {
23    // Invalid number, turn off all segments
24    Turn off all pins on GPIOB
25}
```



6 Exercise 5

6.1 Report 1

Schematic is in **Overall** section.

6.2 Report 2

This is the pseudocode from exercise 5 header file:

```
1 Initialize:
2 // Same as Exercise 3
3
4 Main Loop:
5 while(1) {
6     // Run the traffic light control system
7     exercise_3()
8
9     // Display countdown on first 7-segment display for vertical
10    switch(ver_phase) {
11        case 1: // Green phase
12            display7SEG(3 - ver_count) // Countdown from 3 to 0
13            break
14
15        case 2: // Yellow phase
16            display7SEG(2 - ver_count) // Countdown from 2 to 0
17            break
18
19        case 3: // Red phase
20            display7SEG(5 - ver_count) // Countdown from 5 to 0
21            break
22    }
23
24    // Display countdown on second 7-segment display for
25    horizontal
26    switch(hor_phase) {
27        case 1: // Red phase
28            display7SEG_2(5 - hor_count) // Countdown from 5 to
29            0
```



```
28         break
29
30     case 2: // Green phase
31         display7SEG_2(3 - hor_count) // Countdown from 3 to
32         0
33         break
34
35     case 3: // Yellow phase
36         display7SEG_2(2 - hor_count) // Countdown from 2 to
37         0
38         break
39
40 }
```

```
37 }
38
39 // Wait for next iteration
40 }
```



7 Exercise 6

7.1 Report 1

Schematic is in **Overall** section.

7.2 Report 2

This is the pseudocode from exercise 6 header file:

```
1 //initialize the clock pins
2 uint16_t clockPins[12] = {
3     d12_Pin, d1_Pin, d2_Pin, d3_Pin, d4_Pin, d5_Pin,
4     d6_Pin, d7_Pin, d8_Pin, d9_Pin, d10_Pin, d11_Pin
5 };
6
7 void testLEDs() {
8     for (int i = 0; i < 12; i++) {
9         HAL_GPIO_WritePin(GPIOA, clockPins[i], 1);
10        HAL_Delay(500);
11    }
12 }
```



8 Exercise 7

8.1 Report 1

Schematic is in **Overall** section.

8.2 Report 2

This is the pseudocode from exercise 7 header file:

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



9 Exercise 8

9.1 Report 1

Schematic is in **Overall** section.

9.2 Report 2

This is the pseudocode from exercise 8 header file:

```
1 void setNumberOnClock(int num) {  
2     if (num >= 0 && num < 12) {  
3         HAL_GPIO_WritePin(GPIOA, clockPins[num], 1);  
4     }  
5 }
```



10 Exercise 9

10.1 Report 1

Schematic is in **Overall** section.

10.2 Report 2

This is the pseudocode from exercise 9 header file:

```
1 void clearNumberOnClock(int num) {  
2     if (num >= 0 && num < 12) {  
3         HAL_GPIO_WritePin(GPIOA, clockPins[num], 0);  
4     }  
5 }
```




11 Exercise 10

11.1 Report 1

Schematic is in **Overall** section.

11.2 Report 2

This is the pseudocode from exercise 10 header file

```
1 int second = 0,
2 minute = 0,
3 hour = 0;
4 void displayClock() {
5
6     ++second;
7     if (second > 59){
8         second = 0;
9         ++minute;
10    }
11    if (minute > 59){
12        minute = 0;
13        ++hour;
14    }
15    if (hour > 23){
16        hour = 0;
17    }
18
19    clearAllClock();
20    hour = hour % 12;
21    int minuteLED = minute / 5;
22    int secondLED = second / 5;
23
24    setNumberOnClock(hour);
25    setNumberOnClock(minuteLED);
26    setNumberOnClock(secondLED);
27 }
```