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

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 :

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

The schematic for the exercises from 1 to 5 is located here:
https://github.com/KennyLe298/MPU-MCU/tree/main/schematics

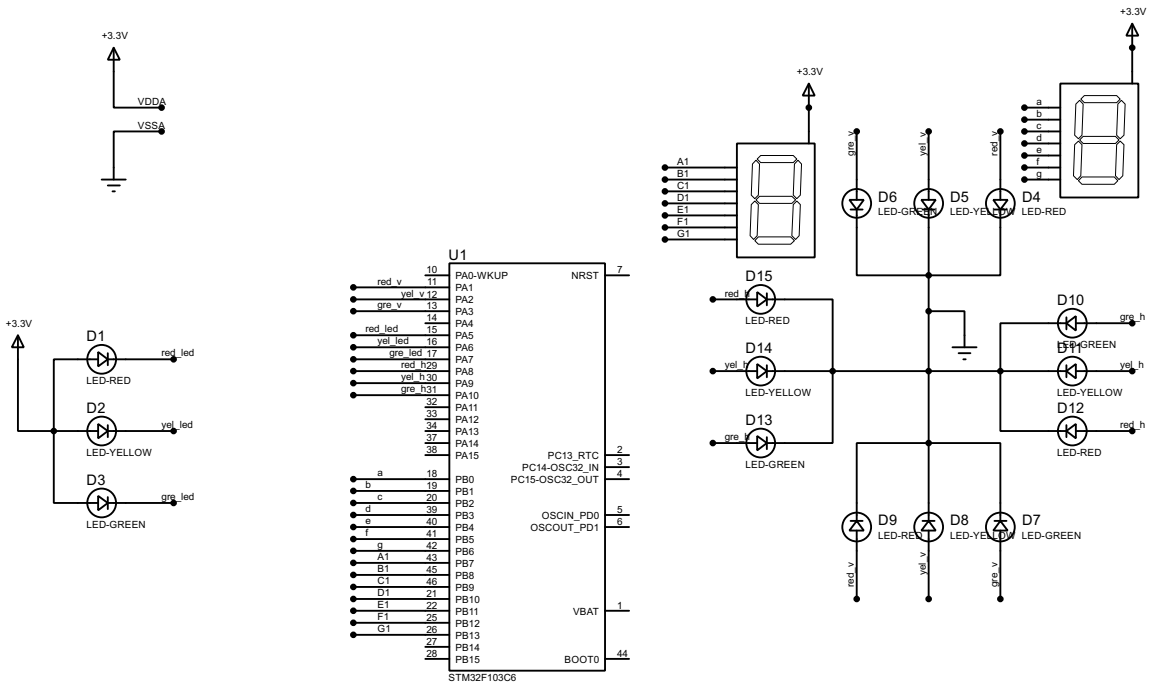

Figure 1.1: Exercise 1 to 5 schematic

The schematic for the exercises from 6 to 10 is located here:
https://github.com/KennyLe298/MPU-MCU/tree/main/schematics

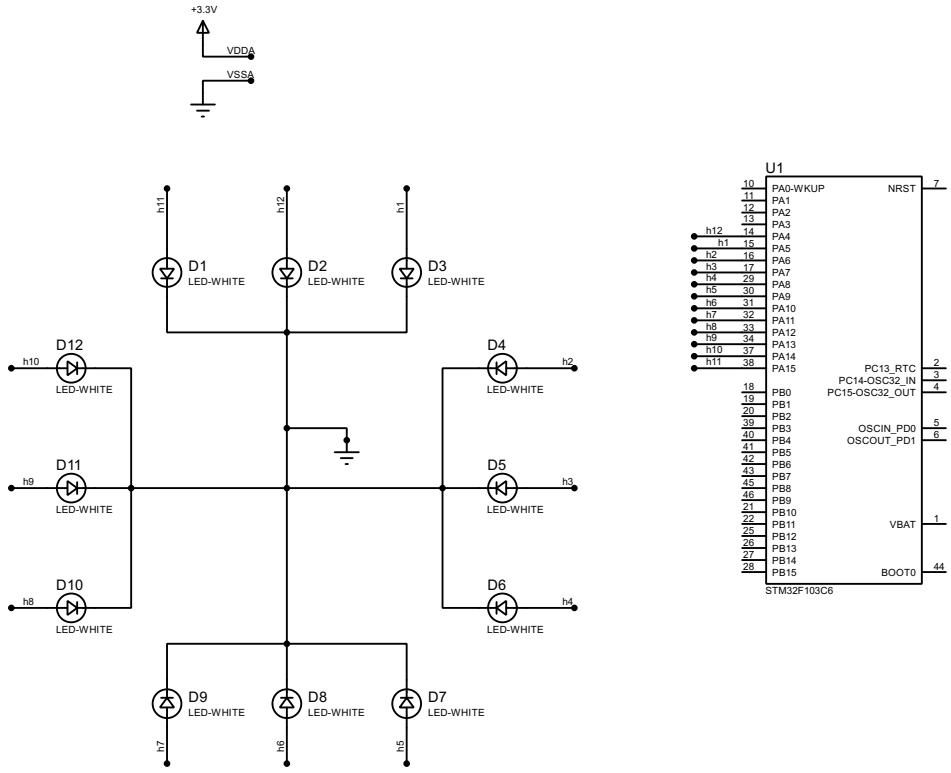

Figure 1.2: Exercise 6 to 10 schematic

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:

```
void LED_toggle(){
HAL_GPIO_TogglePin(LED_RED_GPIO_Port,LED_RED_Pin);
HAL_GPIO_TogglePin(YELLOW_LED_GPIO_Port,YELLOW_LED_Pin);
}
//this is a function to toggle the leds

void initStage(){
HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, GPIO_PIN_RESET);
```

```
HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port, YELLOW_LED_Pin,
    GPIO_PIN_SET);
}
//this is a function to set the first stage before starting to
    toggle

void ex1(){
static unsigned int init = 0; //a variable to initialize
static unsigned int count = 0; //a variable to count seconds

if(init == 0){
    initStage();
    init++;
}
if(count >= 3){ //toggle after 2 seconds so count >=3
    count = 0;
    LED_toggle();
}
else ++count;
}
```

3 Exercise 2

3.1 Report 1

Schematic is in **Overall** section.

3.2 Report 2

```
void initStage(){
HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, GPIO_PIN_RESET);
HAL_GPIO_WritePin(YELLOW_LED_GPIO_Port, YELLOW_LED_Pin,
    GPIO_PIN_SET);
HAL_GPIO_WritePin(LED_GREEN_GPIO_Port, LED_GREEN_Pin, GPIO_PIN_SET
    );
} //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.

```
Initial:
init = 0
ver_s, hor_s = 0
ver_c, hor_c = 0

function ex3():
IF init == 0:
    init = 1
    Vertical = GREEN ON
    Horizontal = RED ON

// Vertical state machine
if ver_s == 1 and ver_c >= 3: (GREEN -> YELLOW)
    ver_s = 2; ver_c = 0
else if ver_s == 2 and ver_c >= 2: (YELLOW -> RED)
    ver_s = 3; ver_c = 0
else if ver_s == 3 and ver_c >= 5: (RED -> GREEN)
    ver_s = 1; ver_c = 0

// Horizontal state machine
if hor_s == 1 and hor_c >= 5: (RED -> GREEN)
    hor_s = 2; hor_c = 0
else if hor_s == 2 and hor_c >= 3: (GREEN -> YELLOW)
    hor_s = 3; hor_c = 0
else if hor_s == 3 and hor_c >= 2: (YELLOW -> RED)
    hor_s = 1; hor_c = 0

ver_c++, hor_c++
```

5 Exercise 4

5.1 Report 1

Schematic is in **Overall** section.

5.2 Report 2

```
int segmentmap[10][7] = {
    {0,0,0,0,0,0,1}, // 0
    {1,0,0,1,1,1,1}, // 1
    {0,0,1,0,0,1,0}, // 2
    {0,0,0,0,1,1,0}, // 3
    {1,0,0,1,1,0,0}, // 4
    {0,1,0,0,1,0,0}, // 5
    {0,1,0,0,0,0,0}, // 6
    {0,0,0,1,1,1,1}, // 7
    {0,0,0,0,0,0,0}, // 8
    {0,0,0,0,1,0,0}  // 9
};

void display7SEG(int num){
if (num < 0 || num > 9){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_All, GPIO_PIN_SET);
        //turn of all segment if wrong input
        return;
}
for (int i = 0; i < 7; i++){
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0 << i, segmentmap[num][
           i]);
}
    // inputs a,b,c,d,e,f,g of the 7SEG LED is from PB0-6 so we
       can shift right for simplification
}
```

6 Exercise 5

6.1 Report 1

Schematic is in **Overall** section.

6.2 Report 2

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

```
//initialize the clock pins
uint16_t hourPins[12] = { h12_Pin, h1_Pin, h2_Pin, h3_Pin, h4_Pin,
    h5_Pin,
h6_Pin, h7_Pin, h8_Pin, h9_Pin, h10_Pin, h11_Pin };

void testLEDs(void){
for(int i = 0; i < 12; i++){
    for(int j = 0; j < 12; j++){
        HAL_GPIO_WritePin(GPIOA, hourPins[j], GPIO_PIN_RESET);
    }
    HAL_GPIO_WritePin(GPIOA, hourPins[i], GPIO_PIN_SET);
    HAL_Delay(5000); //change every 5s
    }
}
```

8 Exercise 7

8.1 Report 1

Schematic is in **Overall** section.

8.2 Report 2

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

9 Exercise 8

9.1 Report 1

Schematic is in **Overall** section.

9.2 Report 2

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

10 Exercise 9

10.1 Report 1

Schematic is in **Overall** section.

10.2 Report 2

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

11 Exercise 10

11.1 Report 1

Schematic is in **Overall** section.

11.2 Report 2

```
//global time variables
int sec = 0;
int min = 0;
int hrs = 0;

// clock simulation
void Clock(void) {
    ++sec;
    if (sec > 59) {
        sec = 0;
        ++min;
    }
    if (min > 59) {
        min = 0;
        ++hrs;
    }
    if (hrs > 23) {
        hrs = 0;
    }
    clearAllClock();

    int s_idx = sec / 5;
    int m_idx = min / 5;
    int h_idx = hrs % 12;
    //changing hours, minute, second so that it shows on the clock
    setNumberOnClock(s_idx);
    setNumberOnClock(m_idx);
    setNumberOnClock(h_idx);
}
```