VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
Faculty of Computer Science and Engineering

CC02 — Lab Report

Microprocessor - Microcontroller
Lab 1

Supervisors: Nguyen Thien An
Students: Nguyen Dang Duy 2252116

Ho Chi Minh City, September 17, 2024

Contents

1 Exercise 2
1.1 Exercise 1 4
1.1.1 Report 1 4
1.1.2 Report 2 4
1.2 Exercise 2 5
1.2.1 Report 1 5
1.2.2 Report 2 5
1.3 Exercise 3 6
1.3.1 Report 1 6
1.3.2 Report 2 6
1.4 Exercise 4 7
1.4.1 Report 1 7
1.4.2 Report 2 7
1.5 Exercise 5 8
1.5.1 Report 1 8
1.5.2 Report 2 8
1.6 Exercise 6 9
1.6.1 Report 1 9
1.6.2 Report 2 9
1.7 Exercise 7 10
1.8 Exercise 8 11
1.9 Exercise 9 12
1.10 Exercise 10 13
References 14

1 Exercise

The GitHub link for the lab schematics is at here or in this link: https://github.com/l1ttled1no/mcu-mpu.

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

```c
while(1) {
    // THE FUNCTION INPUT INSERTED HERE
    HAL_Delay(1000);
}
```

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

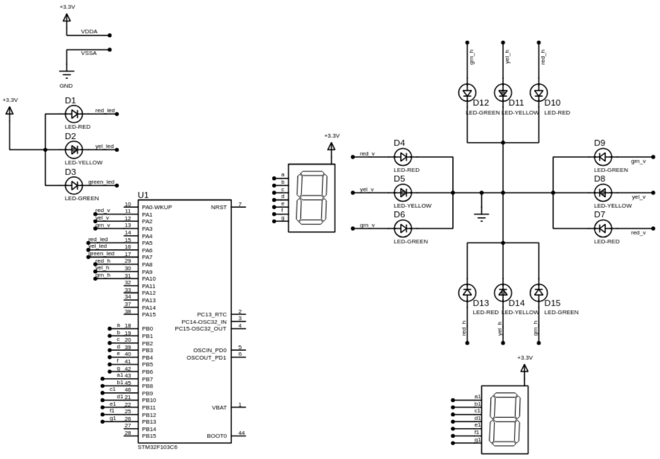

Figure 1: *The schematic for the exercises from 1 to 5.*

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

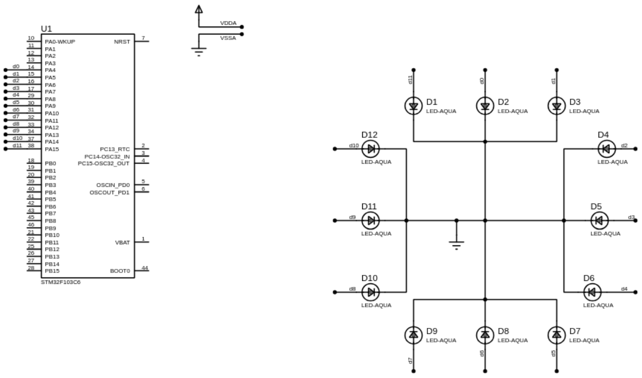

Figure 2: *The schematic for the exercises from 6 to 10.*

1.1 Exercise 1

1.1.1 Report 1

Can be found at 1.

1.1.2 Report 2

This is the header file library for the exercise 1 :

```
function initial_stage:
    set led_red on
    set led_yellow off

function toggle_led:
    toggle led_red
    toggle led_yellow

while true:
    if (stage is initial):
        initial_stage
    if (count to the end):
        toggle led
        reset counter
```

1.2 Exercise 2

1.2.1 Report 1

Can be found at 1.

1.2.2 Report 2

This is the pseudocode for the exercise 2:

```
function stage1:
    reset clock
    set stage
    set led_red on
    set led_yel off
    set led_green off

function stage2:
    reset clock
    set stage
    set led_red off
    set led_yel on
    set led_green off

function stage3:
    reset clock
    set stage
    set led_red on
    set led_yel off
    set led_green off

while true:
    set initial_stage (skip when stage been set)
    switch stage:
        case stage1:
        if (clock = stage_counter) stage2
        case stage2:
        if (clock = stage_counter) stage3
        case stage3:
        if (clock = stage_counter) stage1
```

1.3 Exercise 3

1.3.1 Report 1

Can be found at 1.

1.3.2 Report 2

This is the pseudocode for the exercise 3:

```
Use the pseudocode from exercise2, and add another counter for each LEDs
/*****************************************************/
Vertical LED:
Stage 1: GREEN - 3 sec
Stage 2: YELLOw - 2 sec
Stage 3: RED - 5 sec

Horizontal LED:
Stage 1: RED - 5 sec
Stage 2: GREEN - 3 sec
Stage 3: YELLOW - 2 sec
/*****************************************************/
```

1.4 Exercise 4

1.4.1 Report 1

Can be found at 1.

1.4.2 Report 2

This is the pseudocode for the exercise 4:

```c
//Set an array for 7-Segment LED
int arr[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
};
//   a, b, c, d, e, f, g

void display7SEG(int num){
  if (num >= 0 && num <= 9){
    for (int state = 0; state < 7; state++){
      HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0 << state, arr[num][state]);
    }
  }
 // Since the 7 Segment LED is from PB0 - PB6, so we can use shift right to the location
  else HAL_GPIO_WritePin(GPIOB, GPIO_PIN_All, 0); // Turn off the LED if the number exceed 0-9
}
```

1.5 Exercise 5

1.5.1 Report 1

Can be found at 1.

1.5.2 Report 2

This is the pseudocode for the exercise 5:

```
//Using the pseudocode from the Exercise 3 and 4. In this case, we use 7 Segment
    to see the number of vertical LED
//Add another LED for the 2-dimension of the Traffic

void display7SEGHor(int num);
void display7SEGVer(int num);
void traffic();
void 7SegWithTraffic(){
    traffic();
    switch(ver_stage){ //Make the counter countdown
    case 1:
        display7SEGVer(3-ver_timer);
        break;
    case 2:
      display7SEG(2-ver_timer);
      break;
    case 3:
      display7SEG(5-ver_timer);
      break;
    }
    switch(hor_stage){ //Make the counter countdown
    case 1:
        display7SEGHor(5-hor_timer);
        break;
    case 2:
        display7SEGHor(3-hor_timer);
        break;
    case 3:
        display7SEGHor(2-hor_timer);
        break;
    }
}
```

1.6 Exercise 6

1.6.1 Report 1

Can be found at 2.

1.6.2 Report 2

This is the pseudocode for the exercise 6:

```
int clkPin = 4; //First Pin is at PA4
void turnOnClockSimu(){
  if (clkPin > 15) clkPin = 4;
  if (led[clkPin - 4] < 1){ //If the LED status = OFF -> toggle
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_0 << clkPin); //Shifting right to get the PIN value
    ++led[clkPin - 4]; //Set LED status to on
  }
  ++clkPin;
}
```

1.7 Exercise 7

This is the code for the exercise 7 `void clearAllClock()`:

```c
// This setAllClock is for testing
void setAllClock(){
  int clearClk = 4;
  while (clearClk <= 15){
    if (led[clearClk - 4] == 0){
      HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_0 << clearClk); //Shift right to get location
      ++led[clearClk - 4]; //Set status
    }
    clearClk++;
  }
}

void clearAllClock(){
  int clearClk = 4;
  while (clearClk <= 15){ loop PA4->PA15
    if (led[clearClk - 4] == 1){
      HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_0 << clearClk); //Shift right to get location
      --led[clearClk - 4]; //Set status
    }
    clearClk++;
  }
}
```

1.8 Exercise 8

This is the code for the exercise 8 `void setNumberOnClock()`:

```c
void setNumberOnClock(int num){
    //LED is OFF, num is from 0-11
    if (led[num] < 1 && num >= 0 && num <= 11){
     ++led[num]; //Change status
     HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_0 << (num + 4)); //Shift right to get the location, then toggle
  }
  else return;
}
```

1.9 Exercise 9

This is the code for the exercise 8 void clearNumberOnClock(int num):

```
void clearNumberOnClock(int num){
    //LED is ON, num is from 0-11
    if (led[num] > 0 && num >= 0 && num <= 11){
    --led[num]; //Change status
    HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_0 << (num + 4)); //Shift right to get the location, then toggle
  }
  else return;
}
```

1.10 Exercise 10

This is the code for the exercise 8 `void clock12h()`:

```
void Clock12h(){
    /*
    Clock works:
    if (sec get to 60): sec reset && min adds 1
    if (min get to 60): min reset && min adds 1
    if (hrs get to 23): hrs reset -> 00:00:00
    */
    ++sec; //Count up the clock
    if (sec > 59){
        sec = 0;
        ++min;
    }
    if (min > 59){
        min = 0;
        ++hrs;
    }
    if (hrs > 23){
        hrs = 0;
    }
    clearAllClock();
    setNumberOnClock(sec/5);
    setNumberOnClock(min/5);
    if (hrs < 12) setNumberOnClock(hrs);
    else setNumberOnClock(hrs - 12);
}
```

References