# 微處理機 LAB 2.2

Due ： 兩週後 早上 8:00

第 21 組 郭宗諺 林穎沛

PART 1. (10%)

1. Moder 設定:

   RCC -> AHB2ENR |= RCC_AHB2ENR_GPIOBEN;

   gpio -> MODER &= A1;　　//清空 PB0~PB3

   gpio -> MODER |= A2;　　//寫入 PB0~ PB3

   請填入 A1 及 A2 ?

   A1 = 0xFFFFFFCC

   1111 1111 1111 1111 1111 1111 0000 0000

   A2 = 0xFFFFFF55

   1111 1111 1111 1111 1111 1111 0101 0101

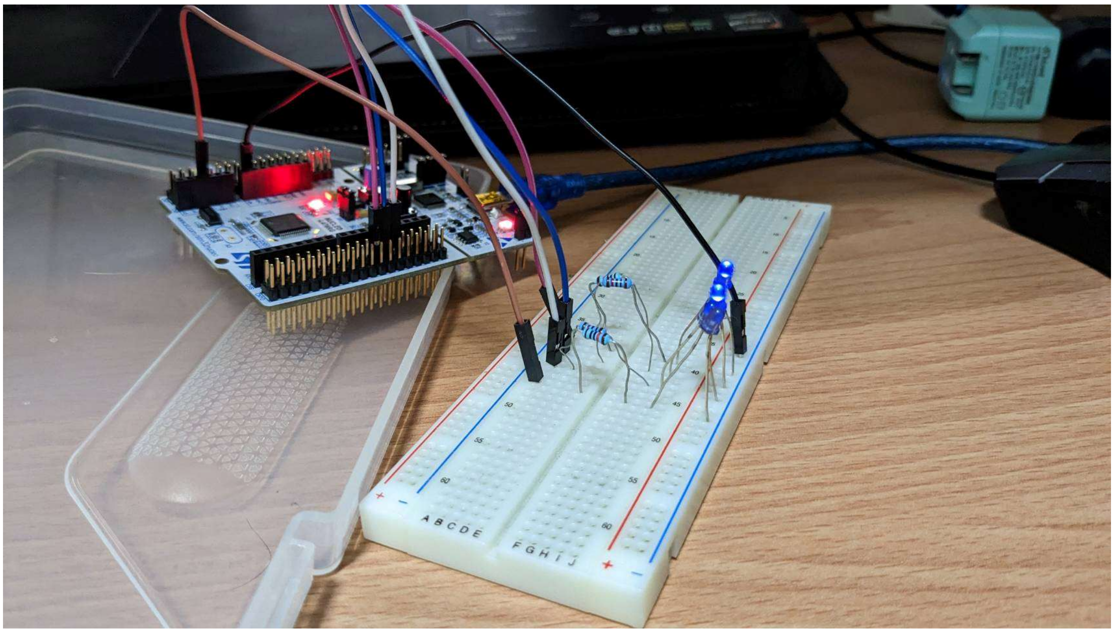
PART 2. (90%) 實作題

請完成實驗 錄影及截圖紀錄實驗結果並附上程式碼(main.s 及 include 之 pin.s 檔案)

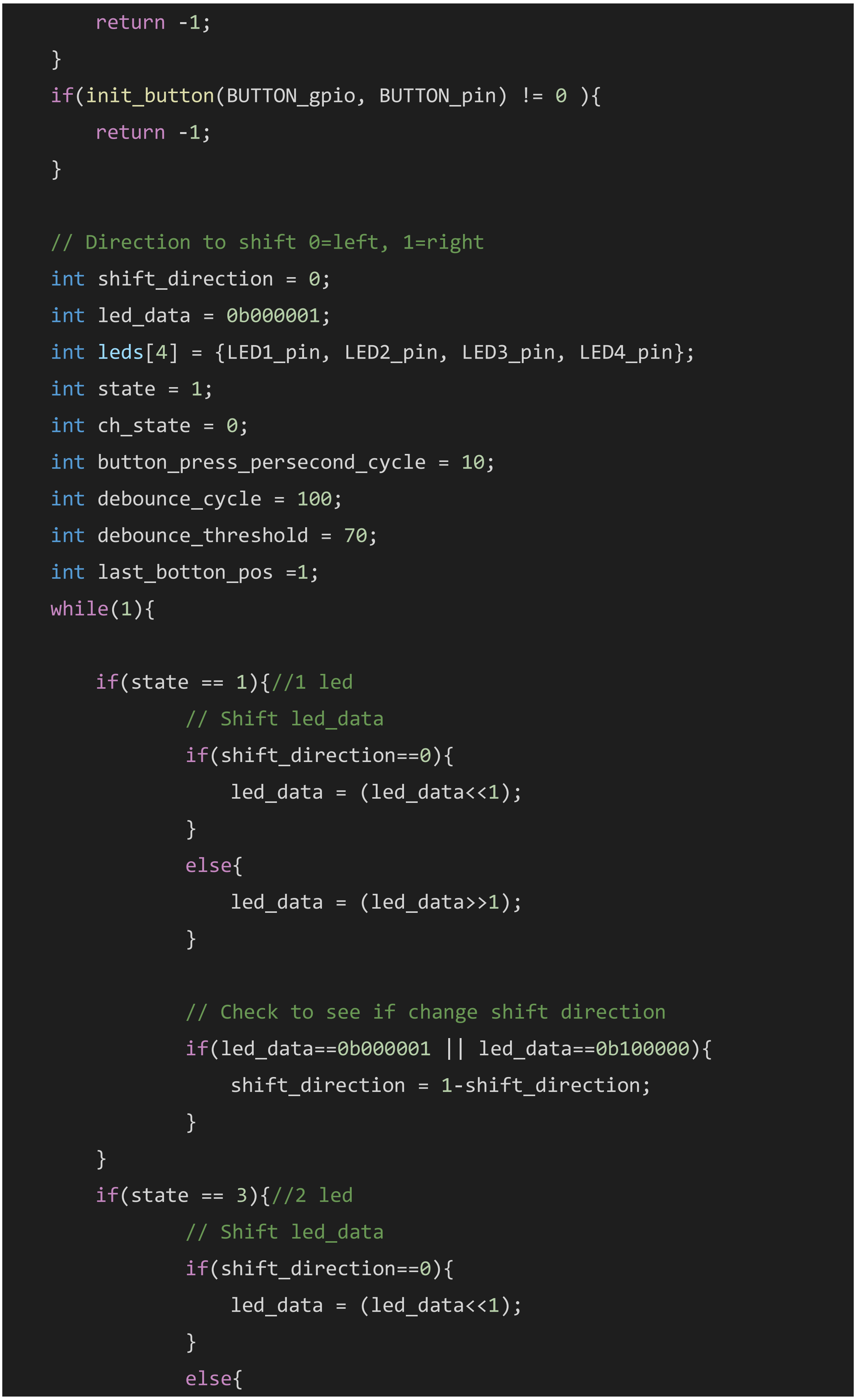
1. LED 跑馬燈+按鍵控制 C 語言版

   功能: 按下版上的藍色按鈕(GPIOC 13)，讓跑馬燈停止/啟動 (30%)
2. 將 GPIOB3 更改為 GPIOA7。程式和 pin 腳都要改才計分。(30%)
3. 然後修改成一顆，兩顆，三顆 LED 左右跑，按按鈕切換。(30%)

   (HINT: led_data)



```
#include "stm32l476xx.h"
#include "helper_functions.h"
#include "led_button.h"
#include "7seg.h"

// Define pins for 4 leds
#define LED_gpio GPIOA
#define LED1_pin 7
#define LED2_pin 6
#define LED3_pin 5
#define LED4_pin 4

// Define pins for button (default use on-board button PC13)
#define BUTTON_gpio GPIOC
#define BUTTON_pin 13

// Define pins for 7seg
//#define SEG_gpio GPIOB
//#define DIN_pin 3
//#define CS_pin 4
//#define CLK_pin 5

// Use to decide which part of the code will run
// Use define & ifdef to control
#define lab_led
//#define lab_led_button
//#define lab_7seg_non_decode
//#define lab_7seg_decode

int main(){

#ifdef lab_led

    if(init_led(LED_gpio, LED1_pin) != 0 || init_led(LED_gpio, LED2_pin) !=
0 || init_led(LED_gpio, LED3_pin) != 0 || init_led(LED_gpio, LED4_pin) !=
0){
        // Fail to init LED
```



```
        return -1;
    }
    if(init_button(BUTTON_gpio, BUTTON_pin) != 0 ){
        return -1;
    }

    // Direction to shift 0=left, 1=right
    int shift_direction = 0;
    int led_data = 0b000001;
    int leds[4] = {LED1_pin, LED2_pin, LED3_pin, LED4_pin};
    int state = 1;
    int ch_state = 0;
    int button_press_persecond_cycle = 10;
    int debounce_cycle = 100;
    int debounce_threshold = 70;
    int last_botton_pos =1;
    while(1){

        if(state == 1){//1 led
                // Shift led_data
                if(shift_direction==0){
                    led_data = (led_data<<1);
                }
                else{
                    led_data = (led_data>>1);
                }

                // Check to see if change shift direction
                if(led_data==0b000001 || led_data==0b100000){
                    shift_direction = 1-shift_direction;
                }
        }
        if(state == 3){//2 led
                // Shift led_data
                if(shift_direction==0){
                    led_data = (led_data<<1);
                }
                else{
                    led_data = (led_data>>1);
                }

                // Check to see if change shift direction
                if(led_data==0b000011 || led_data==0b110000){
                    shift_direction = 1-shift_direction;
                }
        }
        if(state == 5){//3 led
                // Shift led_data
                if(shift_direction==0){
                    led_data = (led_data<<1);
                }
                else{
                    led_data = (led_data>>1);
                }

                // Check to see if change shift direction
                if(led_data==0b000111 || led_data==0b111000){
                    shift_direction = 1-shift_direction;
                }
        }

        for(int a=0;a<4;a++){
            // Active
            if((led_data>>(a+1))&0x1){
                set_gpio(LED_gpio, leds[a]);
            }
            else{
                reset_gpio(LED_gpio, leds[a]);
            }

        }
        for(int i=0; i<button_press_persecond_cycle; i++){
                int pos_cnt = 0; //count
                for(int a=0; a<debounce_cycle; a++){
                    if(read_gpio(BUTTON_gpio, BUTTON_pin) == 0){
                        pos_cnt++;
                    }
                    delay_without_interrupt(1000/(button_press_persecond_cycle*debounce_cycle));
                }
                if(pos_cnt > debounce_threshold){
                    if(last_botton_pos == 1){
                        last_botton_pos = 0;
                    }

                }else{

                    if(last_botton_pos == 0){
                        state++;// state mode: no, 1, no, 2, no, 3
                        if(state == 1){
                            led_data = 0b000001;
                            shift_direction = 0;
                        }else if(state == 3){
                            led_data = 0b000011;
                            shift_direction = 0;
                        }else if(state == 5){
                            led_data = 0b000111;
                            shift_direction = 0;
                        }else if(state == 6){
                            state = 0;
                        }
                        last_botton_pos = 1;
                    }
                }

        }
    }





#endif
```

```
    while(1){}

    return 0;
}
```

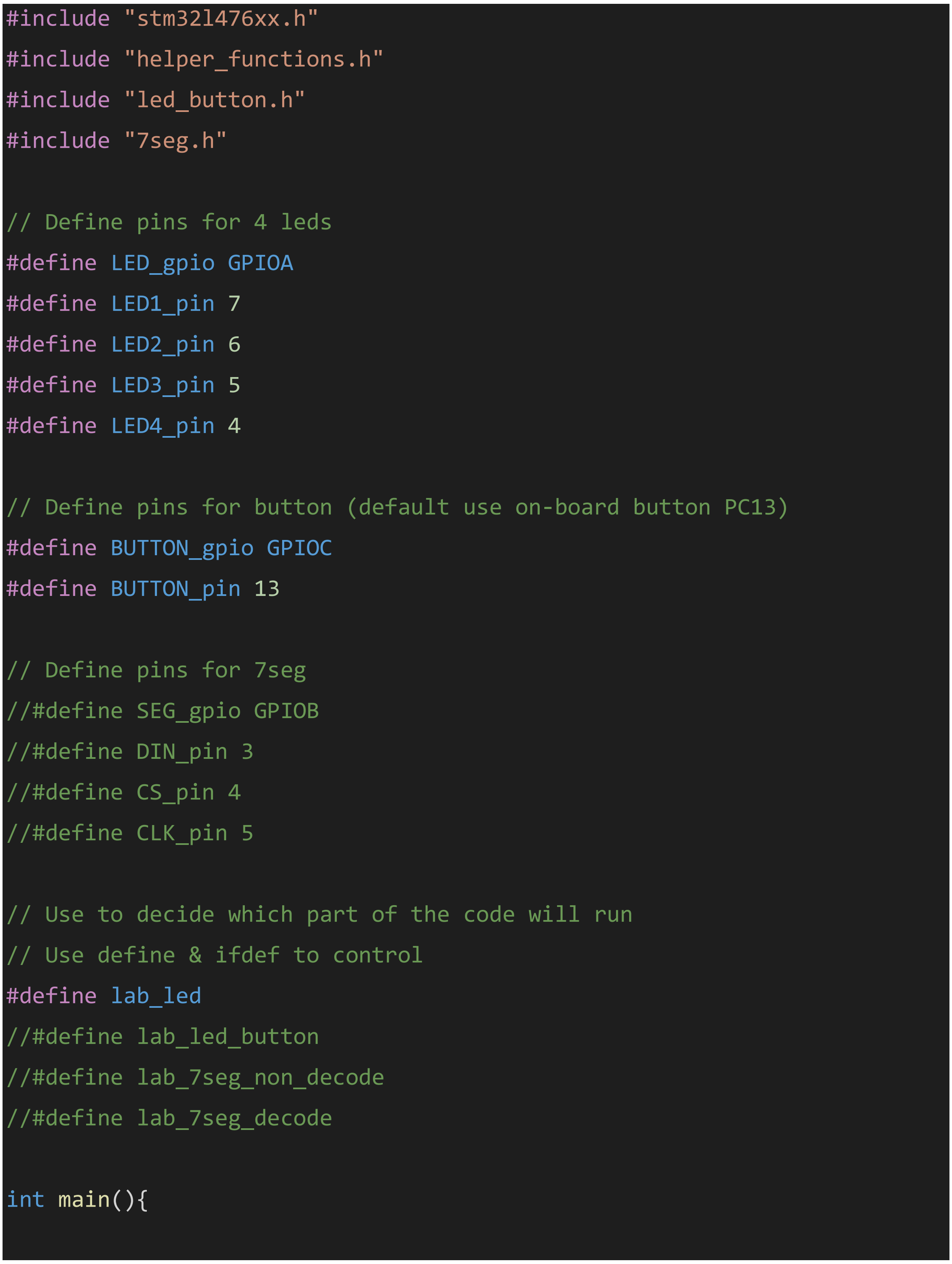
PART 3. 加分題(10%)

將 PART2-1 的跑馬燈從按下按鈕停止改為按下按鈕切換為以下跑馬燈:

●為熄滅　○為亮起

LED 燈腳為由右至左依序為 PB3~PB6

○○○●-> ○○●●-> ○●●○-> ●●○○->●○○○-> ●●○○->
○●●○-> ○○●●-> 循環



```
#include "stm32l476xx.h"
#include "helper_functions.h"
#include "led_button.h"
#include "7seg.h"

// Define pins for 4 leds
#define LED_gpio GPIOA
#define LED1_pin 7
#define LED2_pin 6
#define LED3_pin 5
#define LED4_pin 4

// Define pins for button (default use on-board button PC13)
#define BUTTON_gpio GPIOC
#define BUTTON_pin 13

// Define pins for 7seg
//#define SEG_gpio GPIOB
//#define DIN_pin 3
//#define CS_pin 4
//#define CLK_pin 5

// Use to decide which part of the code will run
// Use define & ifdef to control
#define lab_led
//#define lab_led_button
//#define lab_7seg_non_decode
//#define lab_7seg_decode

int main(){
```

```
#ifdef lab_led

    if(init_led(LED_gpio, LED1_pin) != 0 || init_led(LED_gpio, LED2_pin) !=
0 || init_led(LED_gpio, LED3_pin) != 0 || init_led(LED_gpio, LED4_pin) !=
0){
        // Fail to init LED
        return -1;
    }
    if(init_button(BUTTON_gpio, BUTTON_pin) != 0 ){
        return -1;
    }

    // Direction to shift 0=left, 1=right
    int shift_direction = 0;
    int led_data = 0b00011;
    int leds[4] = {LED1_pin, LED2_pin, LED3_pin, LED4_pin};
    int state = 1;
    int ch_state = 0;
    int button_press_persecond_cycle = 10;
    int debounce_cycle = 100;
    int debounce_threshold = 70;
    int last_botton_pos =1;
    while(1){

        if(state == 1){//normal mode
                // Shift led_data
                if(shift_direction==0){
                    led_data = (led_data<<1);
                }
                else{
                    led_data = (led_data>>1);
                }

                // Check to see if change shift direction
                if(led_data==0b000011 || led_data==0b110000){
                    shift_direction = 1-shift_direction;
                }
        }else{//new mode
```

```
            // Shift led_data
                if(shift_direction==0){
                    led_data = (led_data<<1);
                }
                else{
                    led_data = (led_data>>1);
                }

                // Check to see if change shift direction
                if(led_data==0b000011 || led_data==0b11000){//reduce one 
side's data
                    shift_direction = 1-shift_direction;
                }



        }

        for(int a=0;a<4;a++){
            // Active
            if((led_data>>(a+1))&0x1){
                set_gpio(LED_gpio, leds[a]);
            }
            else{
                reset_gpio(LED_gpio, leds[a]);
            }

        }
        for(int i=0; i<button_press_persecond_cycle; i++){
                int pos_cnt = 0; //count
                for(int a=0; a<debounce_cycle; a++){
                    if(read_gpio(BUTTON_gpio, BUTTON_pin) == 0){
                        pos_cnt++;
                    }
                    delay_without_interrupt(1000/(button_press_persecond
_cycle*debounce_cycle));
                }
                if(pos_cnt > debounce_threshold){
                    if(last_botton_pos == 1){
                        last_botton_pos = 0;
                    }

                }else{

                    if(last_botton_pos == 0){
                        state = 1 - state;
                        led_data = 0b001100;

                        last_botton_pos = 1;
                    }
                }

        }
    }






#endif


    while(1){}

    return 0;
}
```