

# HW1 Mbed OS Semaphore and Button/LED IO

R11921008 電機碩二 羅恩至

## I. Discussion of main.cpp

程式改動部分：

1. 將 Semaphore 的 initial value 從 0 改為 1，讓程式在編譯之後便能直接 acquire 成功。

```
20 Semaphore led_sem(1);
21 Thread t2;
22 Thread t3;
23
24 DigitalOut led1(LED1);
25 DigitalOut led2(LED2);
```

2. 在 void led\_thread(void const \*name)函式當中，進行參數修改，把原先對應 led2 的部分改成 led1，led3 的部分改 led2。

```
void led_thread(void const *name) {
    while (1) {
        led_sem.acquire();
        while (1) {
            if (*((int*)name) == 1) {
                LD1_TOG;
                ThisThread::sleep_for(LED_DELAY);
                printf("led1\n");
                if(button_switch % 2 == 0) {
                    break;
                }
                // break;
            }
            else if (*((int*)name) == 2) {
                LD2_TOG;
                ThisThread::sleep_for(LED_DELAY);
                printf("led2\n");
                if (button_switch % 2 == 1) {
                    break;
                }
            }
        }
    }
}
```

3. 在 main 函式中，將 button.fall(&button\_pressed)那行註解掉，並改變 a2, a3 的數值為 1 與 2，分別對應 led1 以及 led2。

```
int main()
{
    LD1_OFF;
    LD2_OFF;
    LD3_OFF;
    LD4_OFF;

    // button.fall(&button_pressed);
    button.rise(&button_released); // switch led
    const int a2 = 1;
    const int a3 = 2;
    t2.start(callback(led_thread, (void *)&a2));
    t3.start(callback(led_thread, (void *)&a3));
    while (1);
}
```

討論：

1. 執行原先的 code，編譯完之後需按一下 button 後 led2 才會亮起(週期 1 秒的閃爍)，再按一次變成 led3 閃爍，形成 led2 和 led3 兩燈交互閃爍。而此次作業需求為 led1 與 led2 兩燈交互閃爍，因此 trace code 將原先與 led2 有關的變數改成 led1，led3 的部分則改為 led2。
2. 另一要求為編譯完後一開始 led1 燈即需閃爍，因此需要將 semaphore 的 initial value 改為 1，直接 acquire 成功以讓 led1 閃爍。不過實際操作發現，若只將 initial value 改為 1，則在按下 button 之後會出現其中一盞燈閃爍、而另一盞燈恆亮的情況，後來才花了許久時間才瞭解到需要將 main 函式中 `button.fall(&button_pressed)` 註解，因為程式一開始已經有 acquire，故不需要再切換至 falling edge。

## II. Review questions.

### 1. What is the effect or the meaning of the initial value of the semaphore?

若 Semaphore 的初始值為 0，代表資源尚不可獲取，需要等到呼叫玩 release 之後，再去執行 acquire 後才會成功；而若初始值為 1，代表資源可以取得，不須 acquire 即能執行成功。

### 2. What is the purpose of the C keyword volatile in the program?

在 C 中使用 volatile 的目的在於對於某些變數，不去 allocate CPU 的 register，以讓其他 thread 也可以看到最新的 value。

一般來說，當改變某個變數的 value，編譯器並不會馬上將更新過後的值寫進記憶體當中，而是會先將變動的結果暫存在 CPU 的 register 裡，等 process 結束過後才寫入記憶體中。而若此變數在 multi-thread 中需要用到時，可能就會因為變動後的數值未寫入記憶體，導致存取到錯誤的值。因此在此狀況就需要使用 volatile 宣告變數，來讓編譯器不要暫存變數，這樣當該變數有任何更動時，其他 thread 也能夠隨時存取到最新的變數。

### 3. What are the roles played by the parameters of methods fall() and rise() in class InterruptIn?

fall(): Attach a function to call when a falling edge occurs on the input Interrupts are enabled for the pin.

rise(): Attach a function to call when a rising edge occurs on the input Interrupts are enabled for the pin.