



實驗四 STM32 GPIO System

1. 實驗目的

- 了解 STM32 基本輸出入 I/O port 使用原理
- 設計簡易 LED 跑馬燈程式
- 了解按鈕與指撥開關使用原理

2. 實驗原理

請參考上課 GPIO 講義與 STM32L4x6 Reference manual。

3. 實驗步驟

3.1. Lab4.1: LED pattern displayer

參考講義上的教學完成 4 個 GPIO output 初始化，並在麵包板上完成 4 個為 **Active Low** 的 LED 燈電路(當 GPIO 輸出 '1' 時燈暗，'0' 代表燈亮)。

Note: LED 需連接至實驗板上的 PB3, PB4, PB5, PB6

完成依以下 Pattern 閃爍的跑馬燈程式。

3.1.1. Pattern requirement

1 代表 LED 亮，0 代表 LED 暗

初始狀態：最右邊的 LED 亮

0	0	0	1
---	---	---	---

接著**每一秒鐘** LED 依序往左位移，此時會有 2 個 LED 亮

0	0	1	1
---	---	---	---



0	1	1	0
---	---	---	---

1	1	0	0
---	---	---	---

當 LED 亮至最左邊時的下一秒改變位移方向，由左至右



1	0	0	0
---	---	---	---



1	1	0	0
---	---	---	---

當回至初始狀態後在改變位移方向，並重複以上步驟

完成以下程式碼，並利用 leds 這個變數紀錄目前位移數值，與 DisplayLED 函式輸出 leds 數值顯示至 4 個 LED 上。

Note: 需用位移指令 LSL 或 LSR 進行數值位移

```
.data
    leds .byte 0

.text
    .global main

main:
    BL    GPIO_init
    MOVS R1, #1
    LDR   R0, =leds
    STRB R1, [R0]

Loop:
    //TODO: Write the display pattern into leds variable

    BL    DisplayLED
    BL    delay
    B     loop

GPIO_init:
    //TODO: Initial LED GPIO pins as output
    BX LR

DisplayLED:

    BX LR

Delay:
    //TODO: Write a delay 1sec function
    BX LR
```



3.2. Lab4.2 Push button

初始化 GPIO PC13 為 Pull-up input，並設計一程式 Polling 實驗板上的 User button 狀態，當 button 按下再放開可以控制 Lab4.1 跑馬燈的**停止與啟動(按一次停止再按一次啟動...)**。

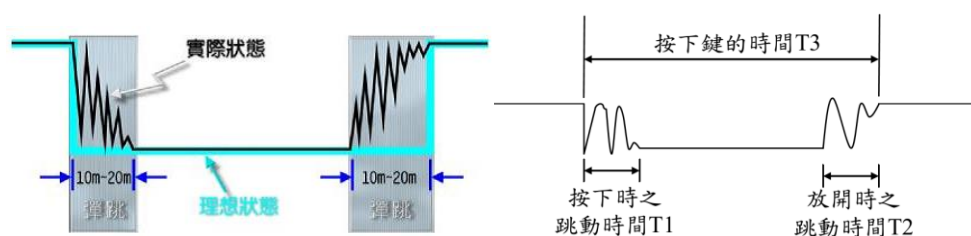
Note: 開發板上的 User button 是連接在 PC13 上，請自行參考講義或 STM32L476 datasheet 完成 GPIOC 初始化。

3.2.1. 開關彈跳

使用軟體方式，解決按鍵彈跳問題。

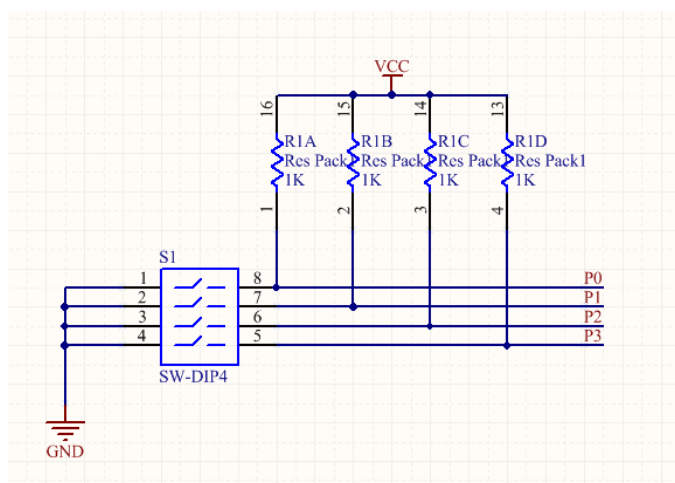
按鍵開關之機械彈跳現象：

按鍵是機械裝置，按壓後，在穩定之前，內部連結會在幾毫秒間來回彈跳。在消除彈跳的這段時間裡，low 和 high 的訊號都會偵測到，造成誤判。

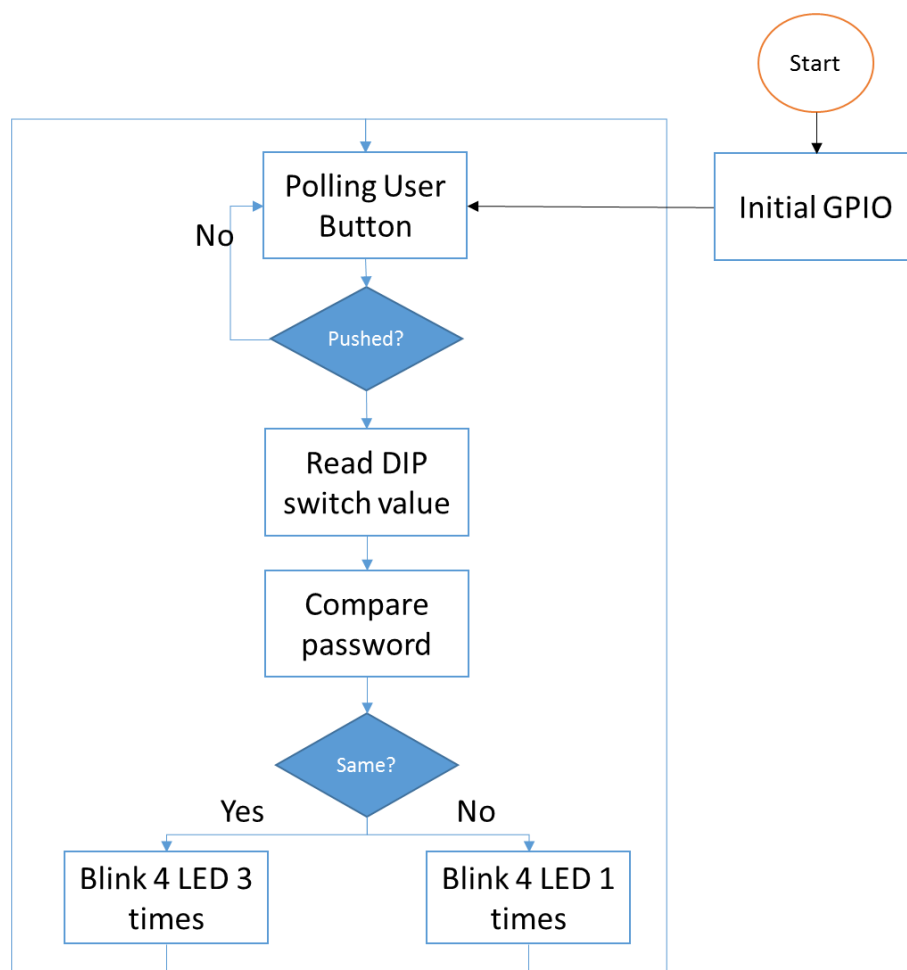


3.3. Lab4.3 密碼鎖

利用麵包板連接 DIP switch 的 active low 電路並連接 P0~P3 至實驗板的 GPIO Pin (同學可自行決定連接的 Pin)



在程式中宣告一個 password 1byte 全域變數並實做一個簡易的 4bit 密碼鎖程式，其流程如下：



Note: DIP switch ON 代表 '1', OFF 代表 '0', 若使用者輸入“ON ON OFF OFF”則代表“1100”。Blink 時間間隔 0.5s