

LAB06 digital I/O Sample Code

DELAY

將延遲寫成一個 macro function 來解決 bouncing problem。透過消耗大量的 cycles 來製造 delay。由於預設的 clock rate 是 1 MHz，所以 252360 個 cycles 約為 0.25 sec。

Initialize

設定 adcon1 將 adcon 設定成 digital IO (因為有些腳位預設是接收類比訊號)。用 BSF 將 TRISB 的 bit 0 設定成 input，用 BCF 將 TRISA 的 bit 0 設為 output。

check_process

用來檢查 input 訊號是否改變的迴圈。Sample code 使用的電阻為上拉電阻，所以在沒有按下 button 時，input 讀到的值會是 1。

使用 BTFSC PORTB, 0 判斷 RB0 的值為 1 時，則繼續在 check_process 內重複判斷。直到 RB0 的值變為 0 時 (button 被按下)，跳到 lightup fuction。

lightup

使用 BTG 來翻轉 bit 的值 (0 -> 1, 1 -> 0) 來改變輸出，藉此改變 LED 燈的狀態。
跳到 DELAY macro 後再回到 check_process 繼續讀取 input 訊號。

Sample Code

LAB06 digital I/O ...



HackMD

(https://hackmd.io?utm_source=view-page&utm_medium=logo-nav).


```

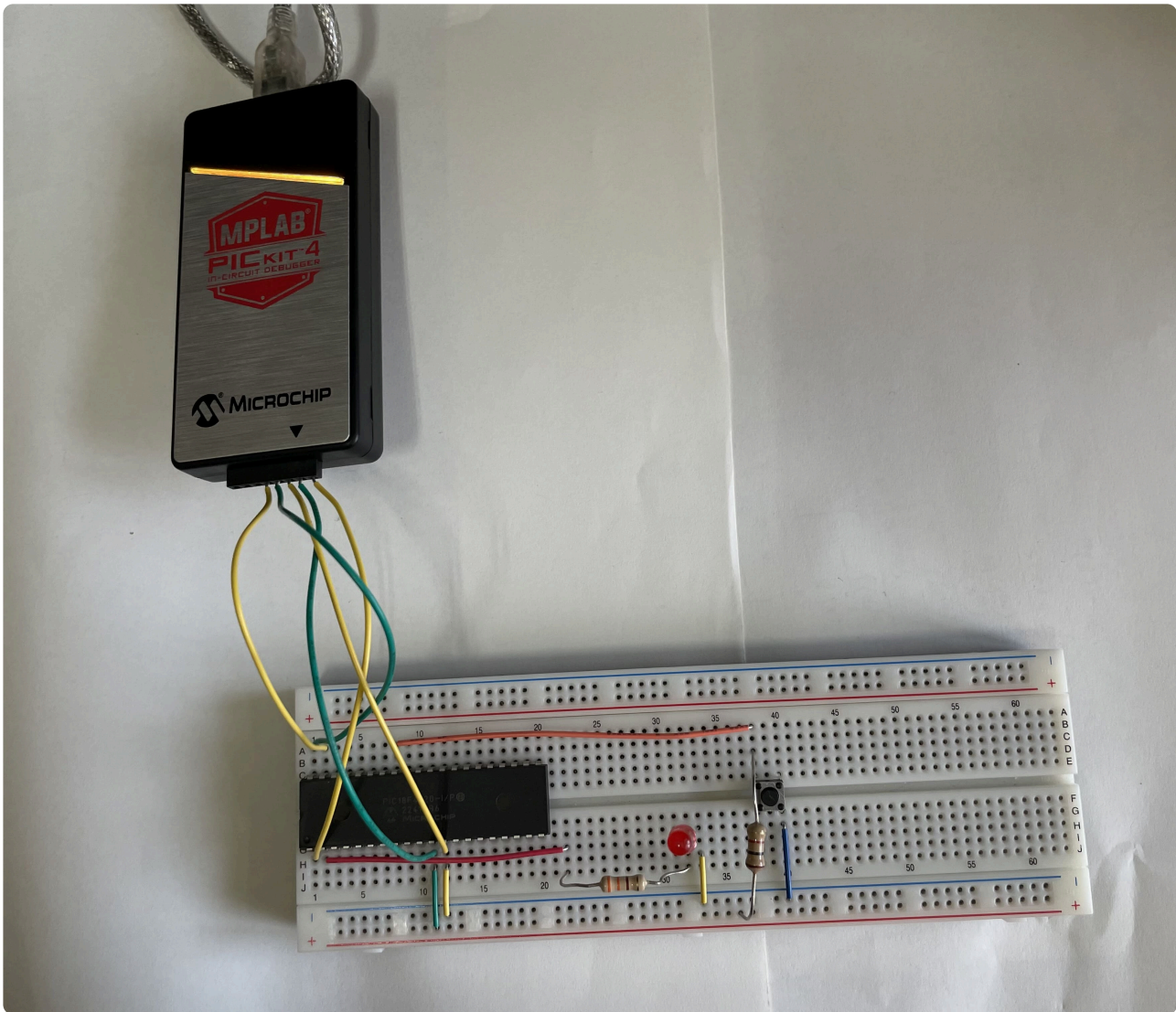
1  LIST p=18f4520
2  #include<p18f4520.inc>
3
4      CONFIG OSC = INTIO67 ; Set internal oscillator to 1 MHz
5      CONFIG WDT = OFF      ; Disable Watchdog Timer
6      CONFIG LVP = OFF      ; Disable Low Voltage Programming
7
8      L1 EQU 0x14           ; Define L1 memory location
9      L2 EQU 0x15           ; Define L2 memory location
10     org 0x00              ; Set program start address to 0x00
11
12     ; instruction frequency = 1 MHz / 4 = 0.25 MHz
13     ; instruction time = 1/0.25 = 4 µs
14     ; Total_cycles = 2 + (2 + 8 * num1 + 3) * num2 cycles
15     ; num1 = 111, num2 = 70, Total_cycles = 62512 cycles
16     ; Total_delay ~= Total_cycles * instruction time = 0.25 s
17     DELAY macro num1, num2
18         local LOOP1      ; Inner loop
19         local LOOP2      ; Outer loop
20
21         ; 2 cycles
22         MOVLW num2        ; Load num2 into WREG
23         MOVWF L2          ; Store WREG value into L2
24
25         ; Total_cycles for LOOP2 = 2 cycles
26         LOOP2:
27         MOVLW num1
28         MOVWF L1
29
30         ; Total_cycles for LOOP1 = 8 cycles
31         LOOP1:
32         NOP               ; busy waiting
33         NOP
34         NOP
35         NOP
36         NOP
37         DECFSZ L1, 1
38         BRA LOOP1        ; BRA instruction spends 2 cycles
39
40         ; 3 cycles
41         DECFSZ L2, 1      ; Decrement L2, skip if zero
42         BRA LOOP2
43     endm
44
45     start:
46     int:
47     ; let pin can receive digital signal
48     MOVLW 0x0f           ; Set ADCON1 register for digital mode
49     MOVWF ADCON1         ; Store WREG value into ADCON1 register
50     CLRF PORTB           ; Clear PORTB
51     BSF TRISB, 0         ; Set RB0 as input (TRISB = 0000 0001)
52     CLRF LATA            ; Clear LATA
53     BCF TRISA, 0         ; Set RA0 as output (TRISA = 0000 0000)
54
55     ; Button check
56     check_process:
57         BTFSC PORTB, 0    ; Check if PORTB bit 0 is low (button pressed)
58         BRA check_process ; If button is not pressed, branch back to check_pi

```

```
59     BRA lightup           ; If button is pressed, branch to lightup
60
61     lightup:
62         BTG LATA, 0         ; Toggle RA0 state (change LED state)
63         DELAY d'111', d'70' ; Call delay macro to delay for about 0.25 seconds
64         BRA check_process
65     end
```

Demo

實際接線圖如下



[Lab06_Demo] Turn on Light



備註:

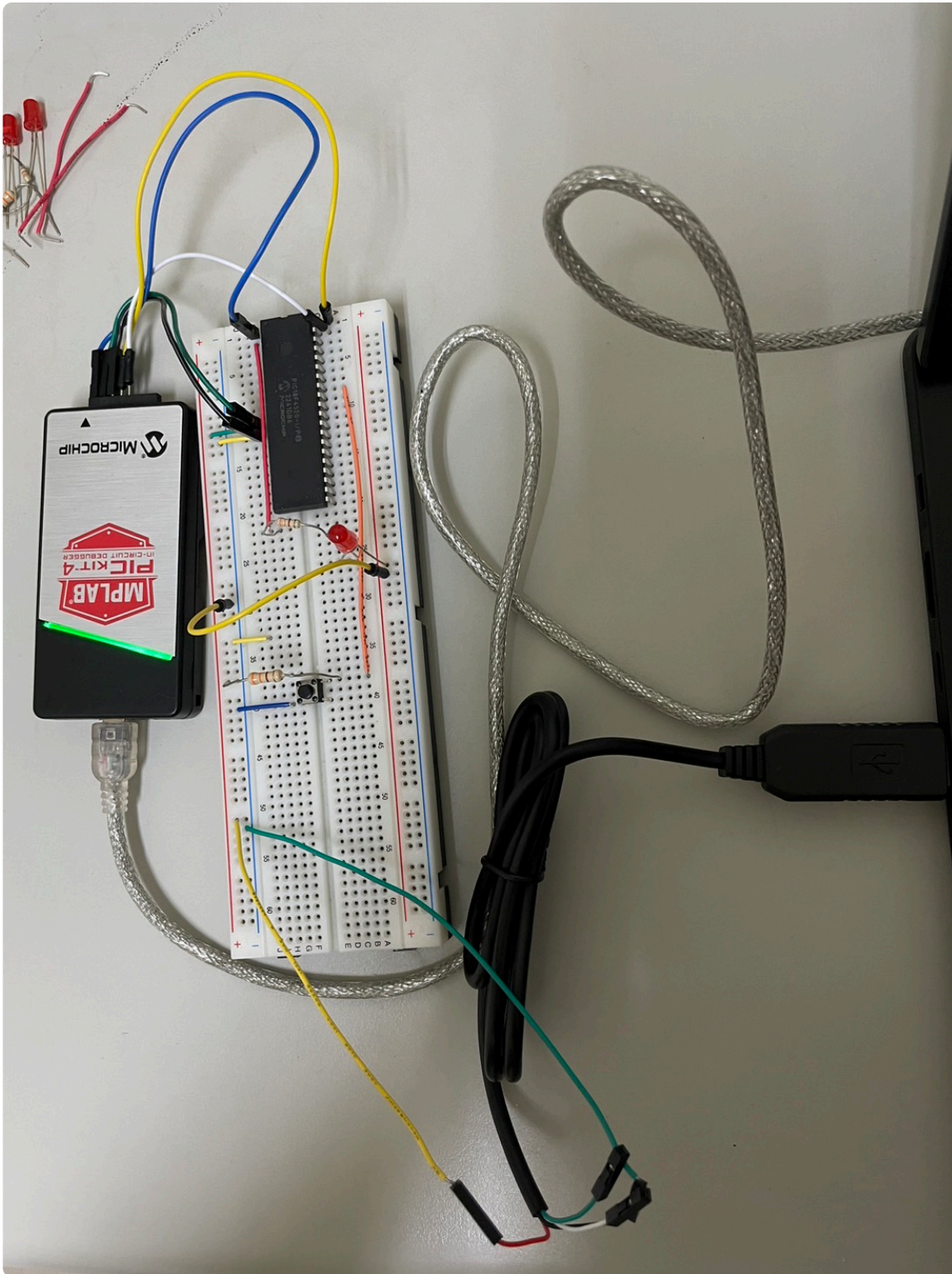
PICkit 4 燒錄器啟動順序為紫色 (常亮約 4 秒) -> 藍色 (常亮，表示燒錄器已就緒)

以下為常見指示燈顏色:

Color	Description
Blue	Power is connected; debugger in standby.
Orange	Power target circuit from PICkit 4 checked.
Green	Power target circuit from PICkit 4 unchecked.
Red	Lit when the debugger has failed.

更新(2024/10/29)

以下為接 TTL 線 (即外部電源) 的電路圖，與前一個電路圖相同，只是多接了一個 TTL 線，由此供電，且 PICkit 4 指示燈為綠色，表示 Power 並非來自 PICkit 4，供同學參考。



Conclusion

將模式設定為數位 I/O，設定 RB0 為輸入，RA0 為輸出，利用 `check_process` 迴圈檢查 button 狀態，如果 button 被按下，就跳到 `lightup`，`lightup` 部分會 toggle LED 狀態，並利用 `DELAY` 來達到延遲效果，然後返回 `check_process` 重新檢查 button 狀態。

