



# 微算機實驗報告

Lab #02

姓名：洪巧芸

系級：資工 114

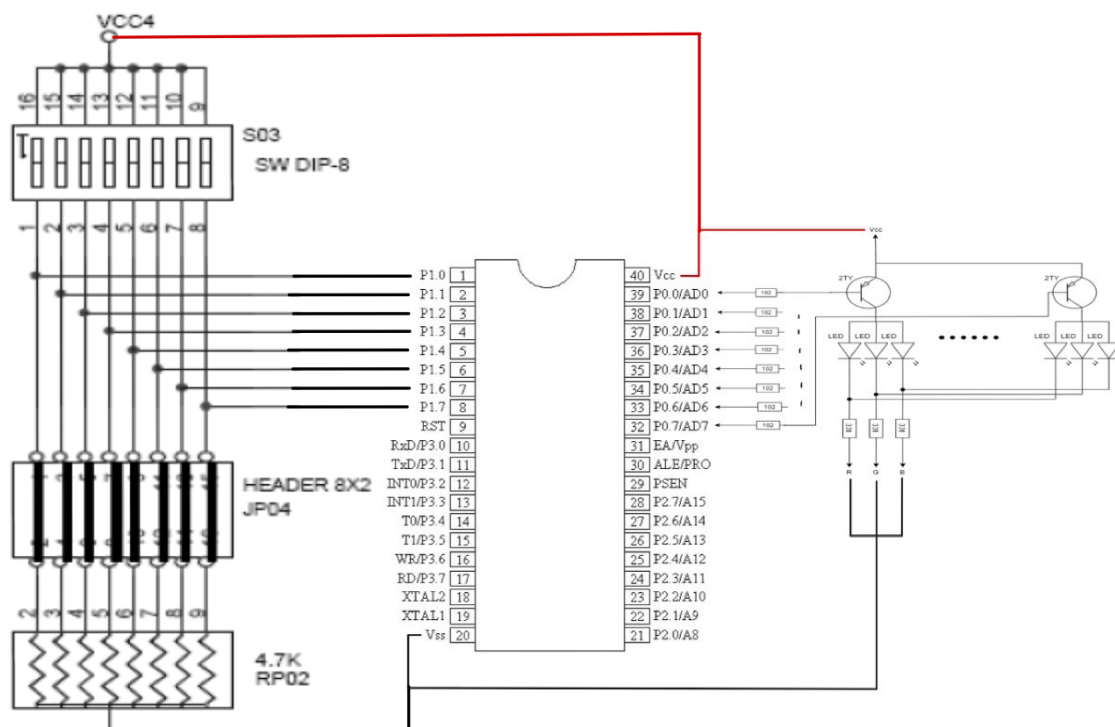
學號：110550143

上課時間：2023/9/26

## 一、實驗目的：

了解開關的電路結構，並配合 Lab1 的 RGB LED 模組，進一步探討微算機 I/O 腳位於輸入輸出控制的相關方法。同時運用到兩個 Port，從其中一個 Port 讀取開關訊號後經過處理再從另一個 Port 輸出，並藉由組合語言的處理調整 LED 的燈亮順序與速度。

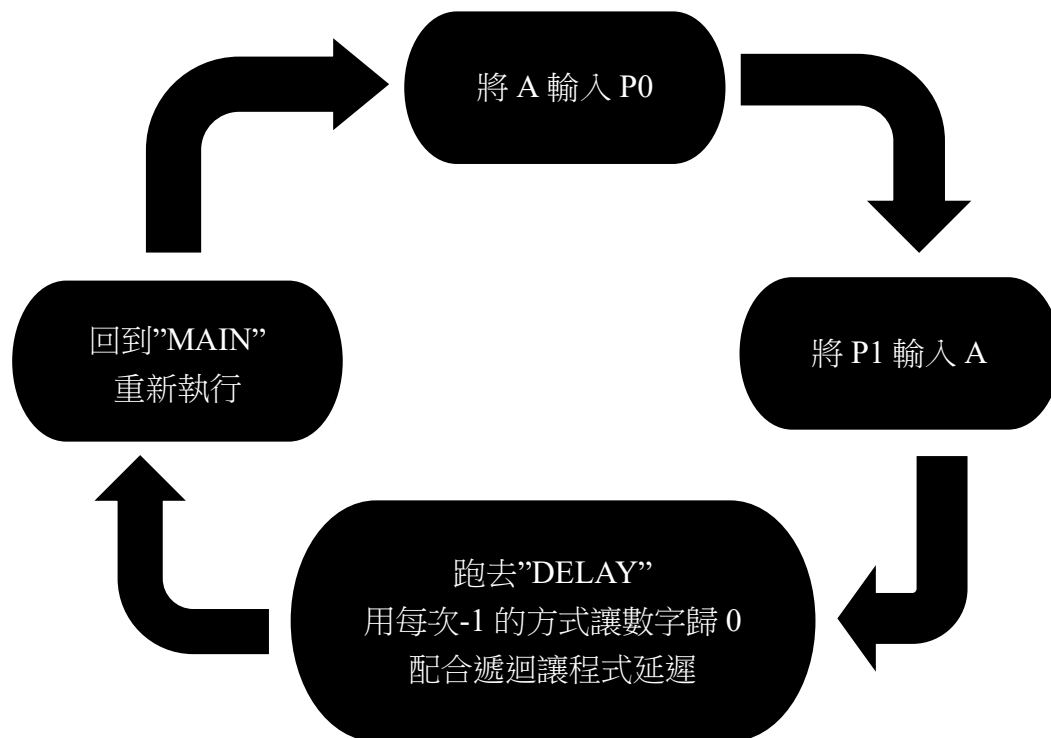
## 二、硬體架構：



### 三、程式流程圖：

兩個題目大致上的流程想法皆是先從連接指撥開關的 Port 讀取並儲存數值，再將其輸入至 A 中進行處理，最後再將 A 的結果輸入回 LED 的 Port。基本題中因為只有要求指撥開關與 LED 的亮暗配合，因此可以直接將指撥開關的數值不經處理輸入到 LED。而進階題中指撥開關的數值代表 LED 燈的跑馬速度，因此從指撥開關讀取的數值需要跟輸入到 LED 的數值分開處理。

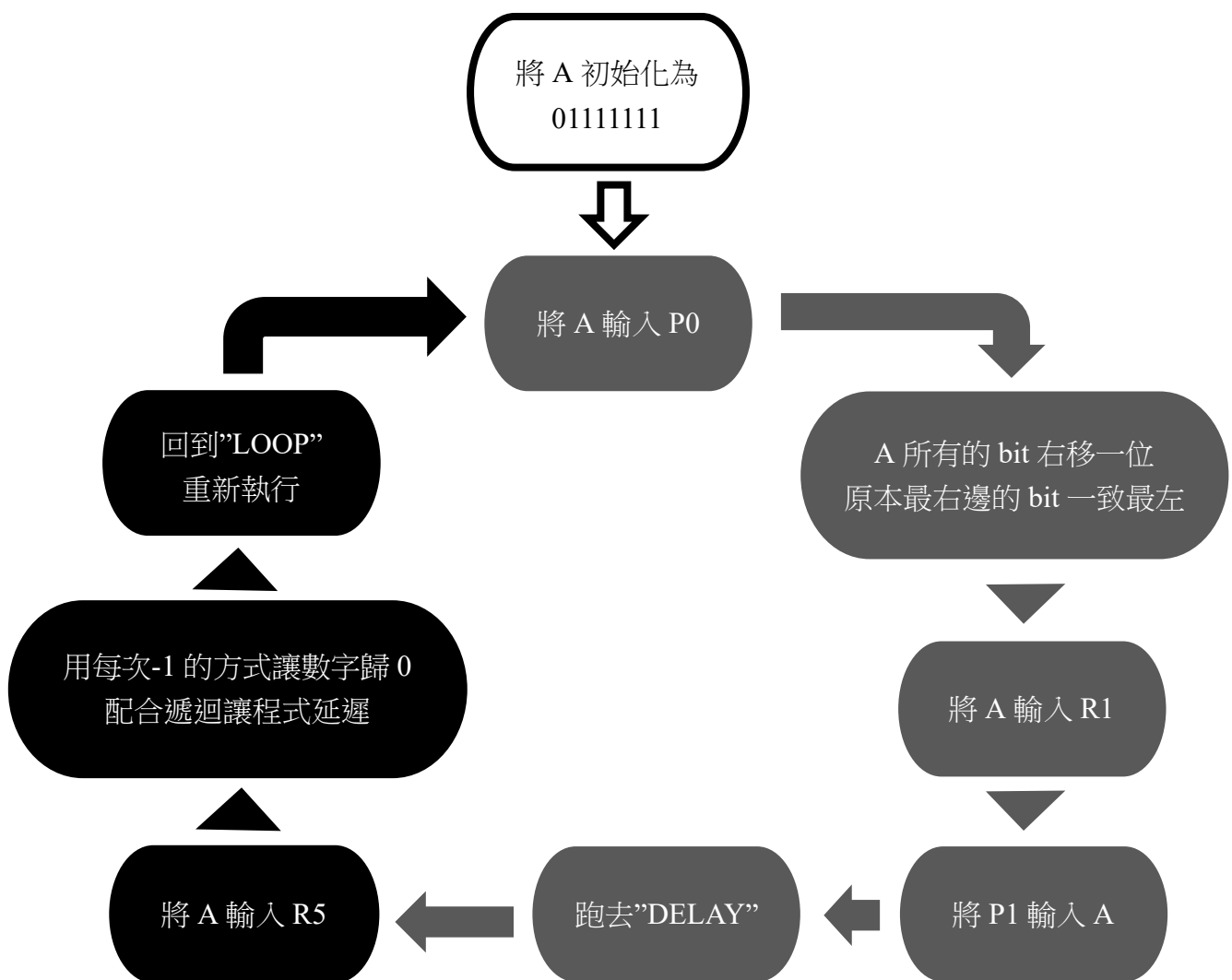
#### A. 基本題



## B. 進階題

因為 LOOP 中會直接將 A 輸入 P0，又因為 P0 代表 LED 的燈亮位置，所以最一開始時需要先決定 LED 的亮度位置，也就是給 A 一個初始值(這邊設定是 01111111)。

因為 A 同時要代表 LED 的燈亮位置及 DELAY 的速度變換，因此運用到另一個暫存器去紀錄 A 再轉換為調整 DELAY 速度前紀錄的 LED 燈亮位置，也就是下圖中的 R1。



#### 四、問題與討論：

(1) 如果將 0x55 寫入 Port 0，請問 Port 0 的輸出為何？

0x55 是 16 進位的表示方式，而我們在產生 Port 的輸出訊號時是以二進位的方式表示： $(0x55)_H = (85)_{10} = (1010101)_b$ ，因此 Port 0 輸出為 1010101。

(2) 在基本題中，假設指撥開關由左至右相對應 RGB LED 模組的 D8-D0，當 LED 由左至右分別呈現「亮暗亮亮亮亮暗亮」，指撥開關對應狀態為何？

開關狀態為「關開關關關關開關」。(因為題目要求開關開時燈不亮，因此兩者狀態相反)

(3) 接續題(2)，假設指撥開關狀態，關為 0，開為 1，將八個指撥開關狀態由左至右轉換成對應的二進制數值，並將 8bit 的二進制轉換成相對應的十六進制。

開關狀態為「關開關關關關開關」時輸出訊號為  $(01000010)_b = (42)_H$

## 五、程式碼與註解：

### A、基本題

```
1 // 初始設定
2 ORG 0000H;
3 AJMP MAIN;
4 ORG 0050H;
5
6 MAIN :
7     MOV A, P1; //把P1讀取到的訊號輸入至A暫存
8     MOV P0, A; //把A的值輸入P0, 也就是把燈亮信號輸入到LED中
9     ACALL DELAY; //跳出迴圈, 跑到 "DELAY"這個副程式
10    JMP MAIN; //跳回 "MAIN" 也就是現在這個程式
11
12 // 和Lab1一樣
13 DELAY:
14     MOV R5, #0FFH; //令R5的初始值為255
15     DELAY1:
16     MOV R6, #0FFH; //令R6的初始值為255
17     DELAY2:
18     MOV R7, #05H; //令R7的初始值為5
19     DELAY3:
20     DJNZ R7, DELAY3; //每次都將R7的數值-1, 如果減完之後R7≠0, 就跳到 "DELAY3" 繼續執行
21     |   |   |   |   |   如果減完之後R7=0, 就執行下一行
22     DJNZ R6, DELAY2; //每次都將R6的數值-1, 如果減完之後R6≠0, 就跳到 "DELAY3" 繼續執行
23     |   |   |   |   |   如果減完之後R6=0, 就執行下一行
24     DJNZ R5, DELAY1; //每次都將R5的數值-1, 如果減完之後R5≠0, 就跳到 "DELAY3" 繼續執行
25     |   |   |   |   |   如果減完之後R5=0, 就執行下一行
26     RET; //回到ACALL的下一行, 也就是JMP MAIN
27 END;
```

### B、進階題

```
1 // 初始設定
2 ORG 0000H
3 AJMP MAIN
4 ORG 0050H
5
6 MAIN:
7     MOV A, #07FH; //把A初始化為01111111 (讓最左邊的燈亮)
8     LOOP:
9     MOV P0, A; //把A的值輸入P0, 也就是把燈亮信號輸入到LED中
10    RR A; //讓A所有的bit都右移一位, 原本在最右邊的bit移到最左邊
11    MOV R1, A; //因為A等一下要當作指撥開關的數值紀錄, 但下一個LOOP的LED燈亮位置要接續這次的, 所以先將A輸入R1暫存
12    MOV A, P1; //把P1的值輸入A (此時A紀錄的數值為指撥開關的數值)
13    ACALL DELAY; //跳出迴圈, 跑到 "DELAY"這個副程式
14    MOV A, R1; //把R1, 也就是剛剛算完下一次LED要亮燈的位子的數值輸回A
15    JMP LOOP; //跳到 "LOOP", 也就是重新執行現在這個程式
16
17 // 和Lab1一樣
18 DELAY:
19     MOV R5, A; //把A的值輸入R5, 這樣LED的速度就會隨著A值上升而變慢
20     DELAY1:
21     MOV R6, #0FFH; //令R6的初始值為255
22     DELAY2:
23     MOV R7, #05H; //令R7的初始值為5
24     DELAY3:
25     DJNZ R7, DELAY3; //每次都將R7的數值-1, 如果減完之後R7≠0, 就跳到 "DELAY3" 繼續執行
26     |   |   |   |   |   如果減完之後R7=0, 就執行下一行
27     DJNZ R6, DELAY2; //每次都將R6的數值-1, 如果減完之後R6≠0, 就跳到 "DELAY3" 繼續執行
28     |   |   |   |   |   如果減完之後R6=0, 就執行下一行
29     DJNZ R5, DELAY1; //每次都將R5的數值-1, 如果減完之後R5≠0, 就跳到 "DELAY3" 繼續執行
30     |   |   |   |   |   如果減完之後R5=0, 就執行下一行
31     RET; //回到ACALL的下一行, 也就是JMP LOOP
32 END;
```

## 六、心得：

### 1. 對於上課內容的心得感想：

這次的課程介紹了指撥開關的板子，原本以為板子上的兩組開關的電路接線會是一樣的，但在老師介紹之後才知道兩者的接線不同。因此如果在實驗時不小心接錯跳線位置的話程式碼就需要改變，是在實驗時需要特別注意的事情。

指撥開關	ON	OFF
左邊	輸出訊號為 0 (燈暗)	輸出訊號為 1 (燈亮)
右邊	輸出訊號為 1 (燈亮)	輸出訊號為 0 (燈暗)

### 2. 對於實驗內容的心得感想：

這次的程式內容跟以往學到的 C，C++ 比較相似，尤其是解題的思路與其實際的書寫方式，因此在實作時上手較為迅速，並沒有遇到太大的困難，也在實驗結束後協助隔壁同學進行 Debug。