



微算機實驗報告

Lab #05

姓名：洪巧芸

系級：資工 114

學號：110550143

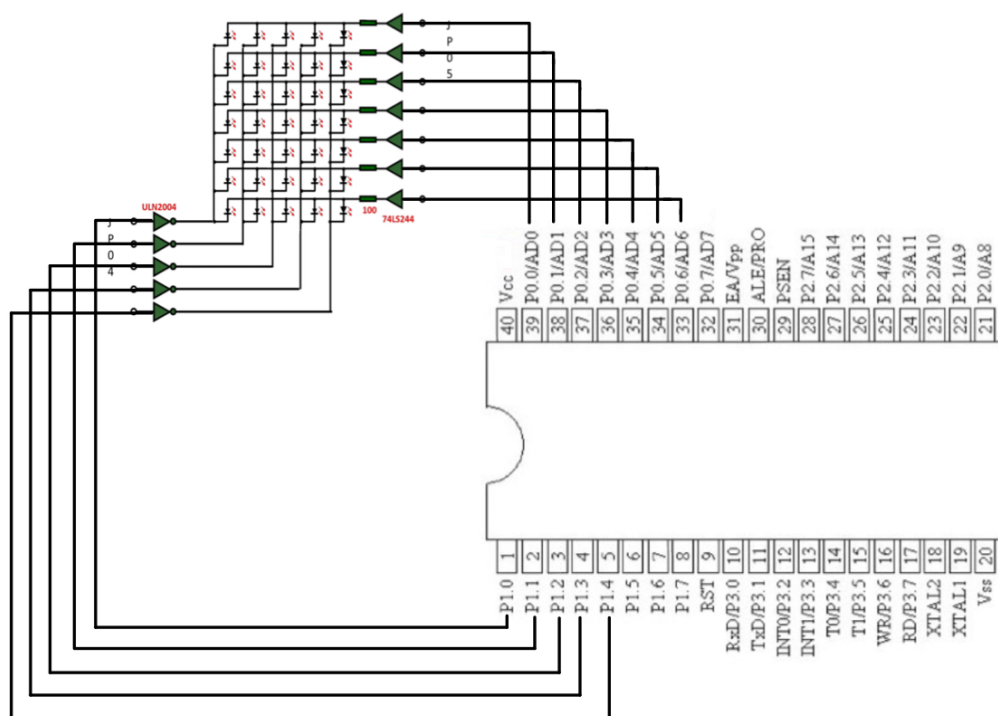
上課時間：2023/10/24

一、實驗目的：

了解 5*7 點矩陣 LED 之電路架構及其工作原理，並學習其驅動方法，同時考慮到直行與橫列的組合決定當前燈亮位置，並配合延遲時間，讓眼睛產生視覺暫留，藉此產生不同的圖形。

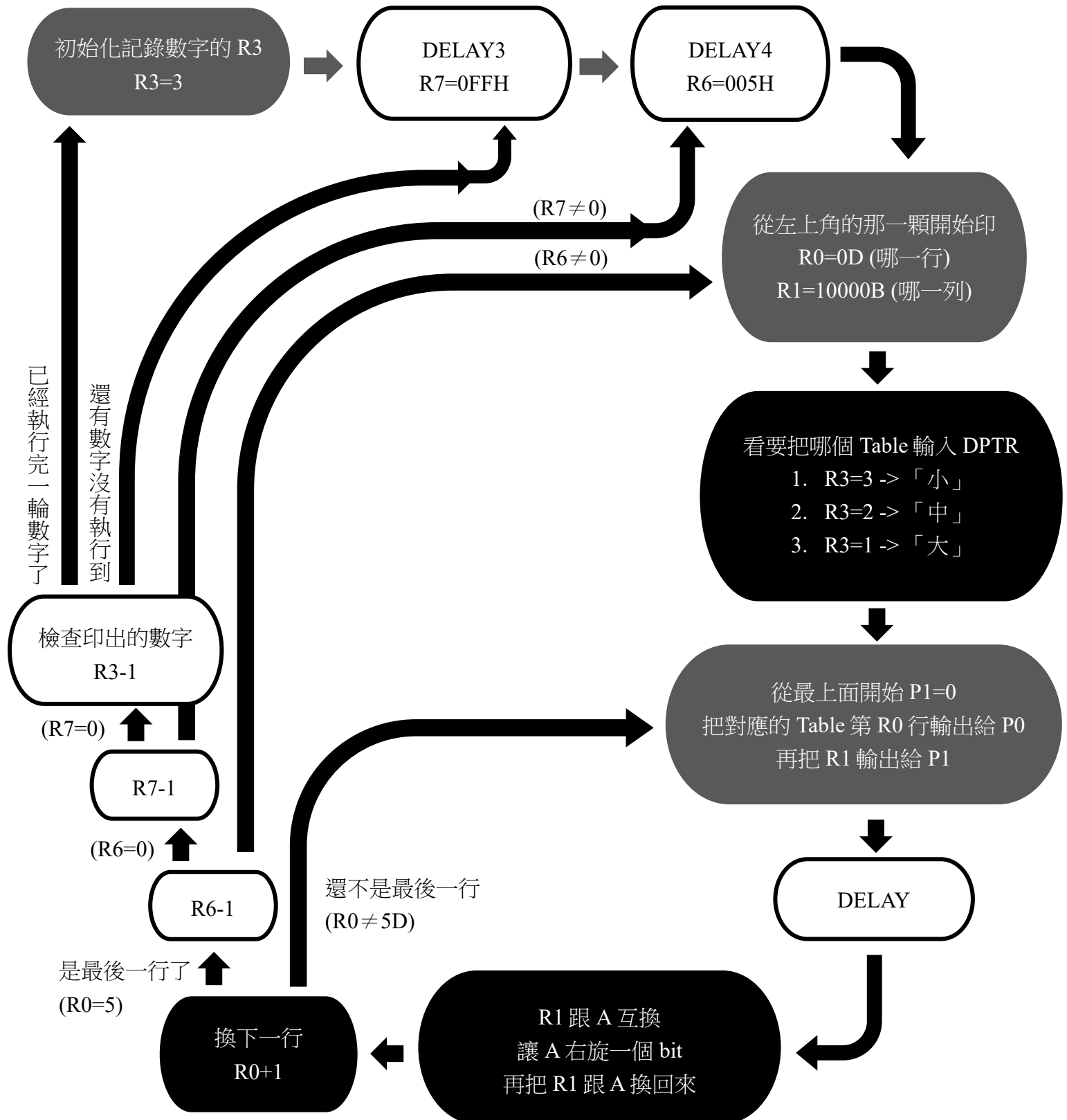
二、硬體架構：

P0 控制橫列(JP05)，P1 控制直行(JP04)



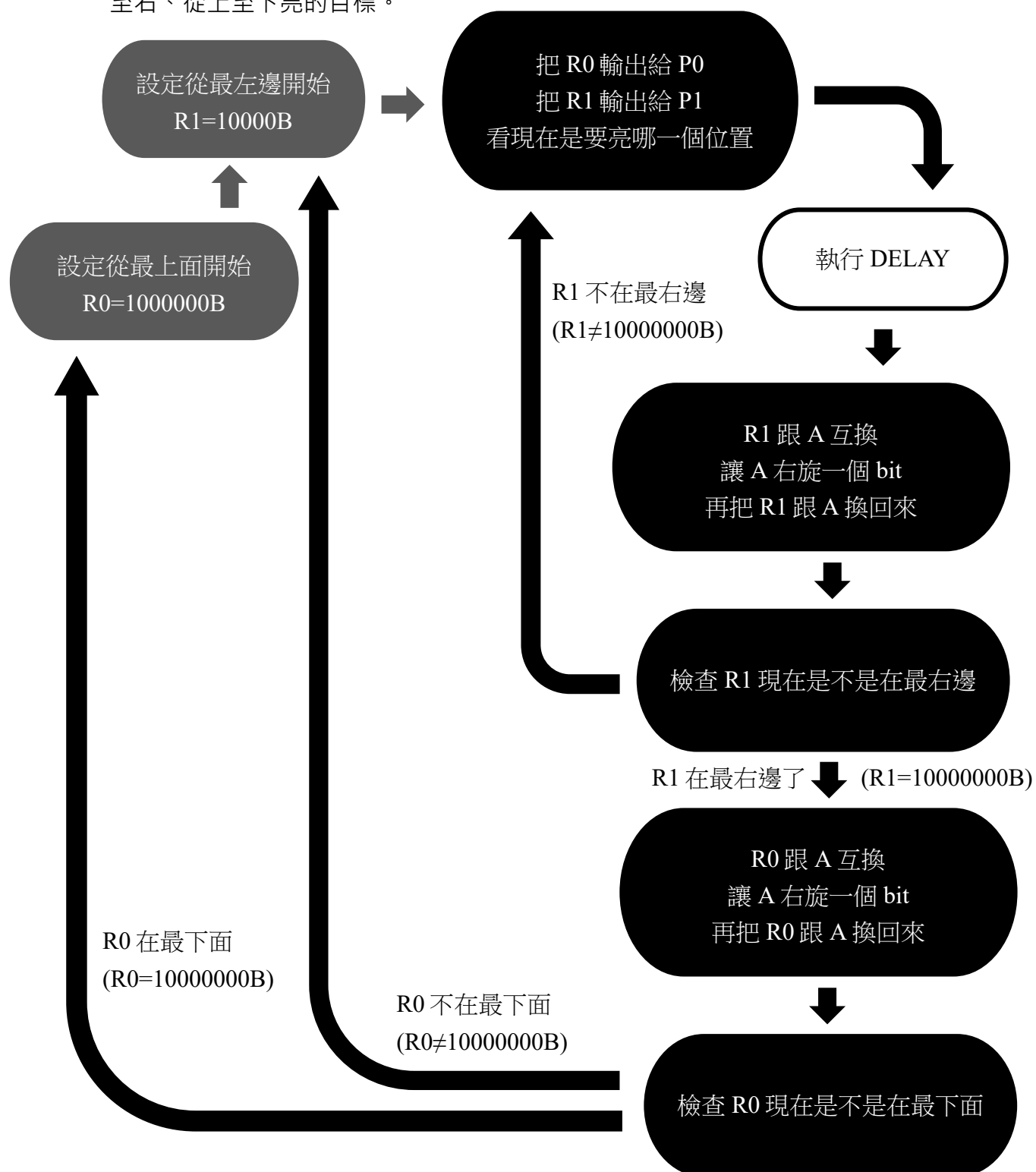
三、程式流程圖：

A. 基本題



B. 進階題

設定兩個變數 R0、R1 分別表示當前顯示的列跟行，先讓 R0 從最左邊一
到最右邊再更新 R1 的數值，也就是換下一橫列亮，藉此達到讓 LED 燈從左
至右、從上至下亮的目標。



四、問題與討論：

(1) 程式中自行設計的 delay time，時間增加或減少會有什麼影響？

	Delay time 影響	Delay time 減少	Delay time 增加
第一題	燈亮位置改變的速度，並直接影響肉眼看見的圖形	速度加快，還沒有超過肉眼的視覺暫留，因此看起來的國字顯示會變成全部的 LED 都亮起來	速度減慢，每次看就會是國字的一個部分，不會同時顯示一整個國字 (ex. 只有「小」的左邊一點)
第二題	燈亮位置改變的速度	速度加快	速度減慢

(2) 請舉例生活中，與 LED 點矩陣具有相同驅動方式的產品。

跑馬燈、捷運上的到站提醒、紅綠燈的小人。

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 R3, #3D //記錄當前要顯示哪個字 共3個所以初始化為3
8
9 //延遲
10 DELAY3:
11     MOV R7, #0FFH
12 DELAY4:
13     MOV R6, #005H
14
15 //重新設置為左上角那個位置
16 RESET:
17     MOV R0, #0D //顯示到哪一行
18     MOV R1, #10000B //哪一顆亮
19
20 /*目標：
21     用R3的數字決定現在要顯示哪個數，
22     就把那個數的table移到DPTR中
23     1. R3=3: 顯示「小」
24     2. R3=2: 顯示「中」
25     3. R3=1: 顯示「大」*/
26 WORD1:
27     CJNE R3, #3, WORD2
28     MOV DPTR, #SMALL //把SMALL的table移到DPTR
29     AJMP LOOP
30 WORD2:
31     CJNE R3, #2, WORD3
32     MOV DPTR, #MIDDLE //把MIDDLE的table移到DPTR
33     AJMP LOOP
34 WORD3:
35     MOV DPTR, #BIG //把BIG的table移到DPTR
36
```

```

37 LOOP:
38     MOV P1, #0 //從第0列開始(i.e.最左邊)
39     MOV A, R0 //當前在哪一行
40     MOVC A, @A+DPTR
41     MOV P0, A //把當行要亮的燈號輸出
42     MOV P1, R1 //看是亮哪一顆
43
44     //呼叫"DELAY"副程式進行延遲
45     ACALL DELAY
46
47     //R1右旋一位
48     XCH A, R1
49     RR A
50     XCH A, R1
51     //R0+1 下次換看下一行
52     INC R0
53
54     //目標：跑完一整個點矩陣之後才換下一個數字
55     CJNE R0, #5, LOOP
56
57     //延遲完重新初始化成左上角那個點(RESET)
58     DJNZ R6, RESET
59     DJNZ R7, DELAY4
60     DJNZ R3, DELAY3
61     AJMP MAIN
62

```

```

63 //延遲
64 DELAY:
65     MOV R5, #0FFH
66 DELAY1:
67     DJNZ R5, DELAY1
68     RET
69
70 //建立「大中小」的Table
71 BIG:
72     DB 0100010B
73     DB 0100100B
74     DB 1111000B
75     DB 0100100B
76     DB 0100010B
77 MIDDLE:
78     DB 0111110B
79     DB 0100010B
80     DB 1111111B
81     DB 0100010B
82     DB 0111110B
83 SMALL:
84     DB 0111100B
85     DB 0000001B
86     DB 1111111B
87     DB 0000000B
88     DB 0111100B
89
90 END

```

B、進階題

```

1  //初始設定
2  ORG 0000H
3  AJMP DOWN
4  ORG 0050H
5
6  DOWN:
7  | MOV R0, #1000000B //jp05往下
8  RIGHT:
9  | MOV R1, #10000B //jp04往右
10 |
11 LOOP:
12 | MOV P0, R0 //把R0輸出給jp05
13 | MOV P1, R1 //把R1輸出給jp04
14 | ACALL DELAY //延遲
15
16 //目標:讓R1向右移一個(燈往右亮)
17 XCH A, R1 //讓R1的數值先跟A交換
18 RR A //A 右旋 1 bit
19 XCH A, R1 //再把A跟R1的數值換回來
20
21 /*目標:
22 1. R1≠10000000B(還沒跑到最右邊) -> 重新執行LOOP(繼續向右跑)
23 2. R1=10000000B(跑到最右邊) -> 執行下面的程式(換下一橫列)*/
24 CJNE R1, #10000000B, LOOP
25
26 //目標:讓R0向右移一個(換下一橫列的燈亮)
27 XCH A, R0 //讓R0的數值先跟A交換
28 RR A //A 右旋 1 bit
29 XCH A, R0 //再把A跟R0的數值換回來
30
31 /*目標:
32 1. R0≠10000000B(還沒跑到最下面) -> 重新初始化R1(從最左邊的開始亮)
33 2. R0=10000000B(跑到最下面) -> 重新初始化R0(從低一系列開始亮)*/
34 CJNE R0, #10000000B, RIGHT
35 AJMP DOWN

```

```

36
37 //同Lab1的延遲
38 DELAY:
39 | MOV R5, #0FFH
40 DELAY1:
41 | MOV R6, #0FFH
42 DELAY2:
43 | MOV R7, #005H
44 DELAY3:
45 | DJNZ R7, DELAY3
46 | DJNZ R6, DELAY2
47 | DJNZ R5, DELAY1
48 RET
49
50 END

```

六、心得：

1. 對於上課內容的心得感想：

這次的課程用到了點矩陣 LED，在基礎題的部分要延續上次 Lab 中的延遲概念，讓人眼產生視覺暫留，才可以一次看到由好幾個 LED 燈組合而成的文字，對我而言比上次的實驗更為複雜一點。相反地，進階題因為一次只要顯示一個點，所以就不用考慮視覺暫留，只要考慮燈亮位置的行列改變而已。

2. 對於實驗內容的心得感想：

實驗到現在我發現有幾個指令特別常在我的程式碼中使用到，其中 CJNE 跟 XCH 最讓我印象深刻。因為之前在寫 Lab 的時候常常很想念平時寫 C++ 的 if else，想要在 8051 的組合語言中使用，因此最近幾次 Lab 的時候發現 CJNE 時就個外開心。此外一開始在用 XCH 的時候會不小心用成 MOV 的寫法，在要掉換回來的時候把兩個暫存器的位置交換，程式就會出現錯誤。