



微算機實驗報告

Lab #06

姓名：洪巧芸

系級：資工 114

學號：110550143

上課時間：2023/10/31

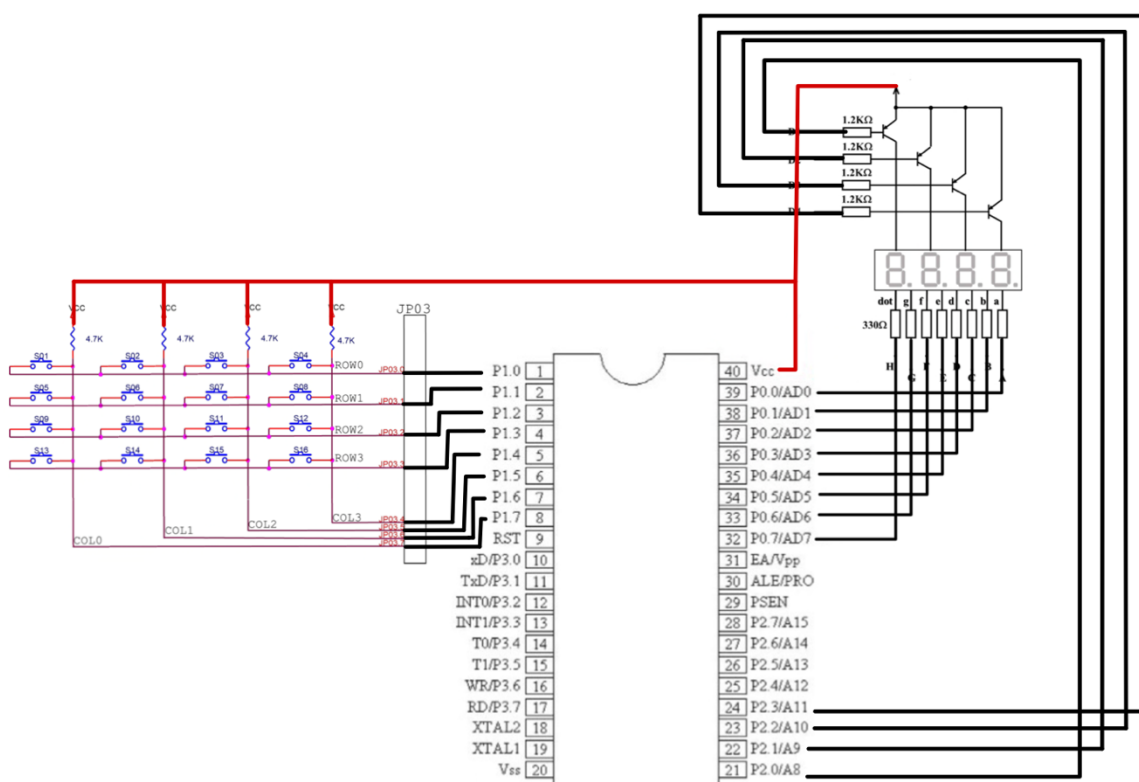
一、實驗目的：

了解鍵盤之電路架構及其工作原理，並學習其驅動方法。運用 01 代表導通與否去判定按鍵是否被按下。並在回傳給 8051 後再輸出給七段顯示器，以題目要求的方式顯示(e.g. 10 進位或是 16 進位)。

二、硬體架構：

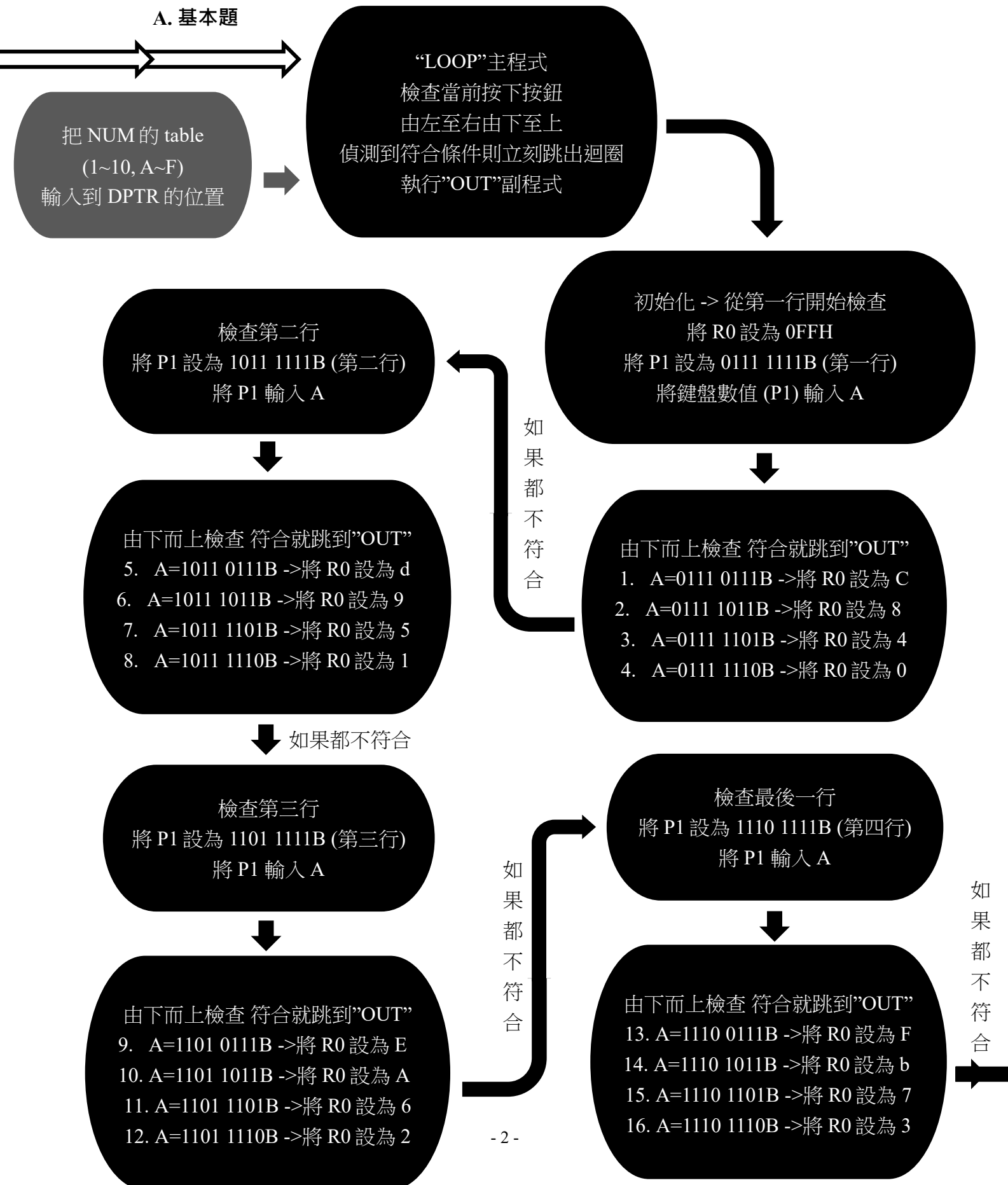
P0 連接七段顯示器要顯示的數字(A~H)，P1 連接鍵盤按鈕(JP03)，

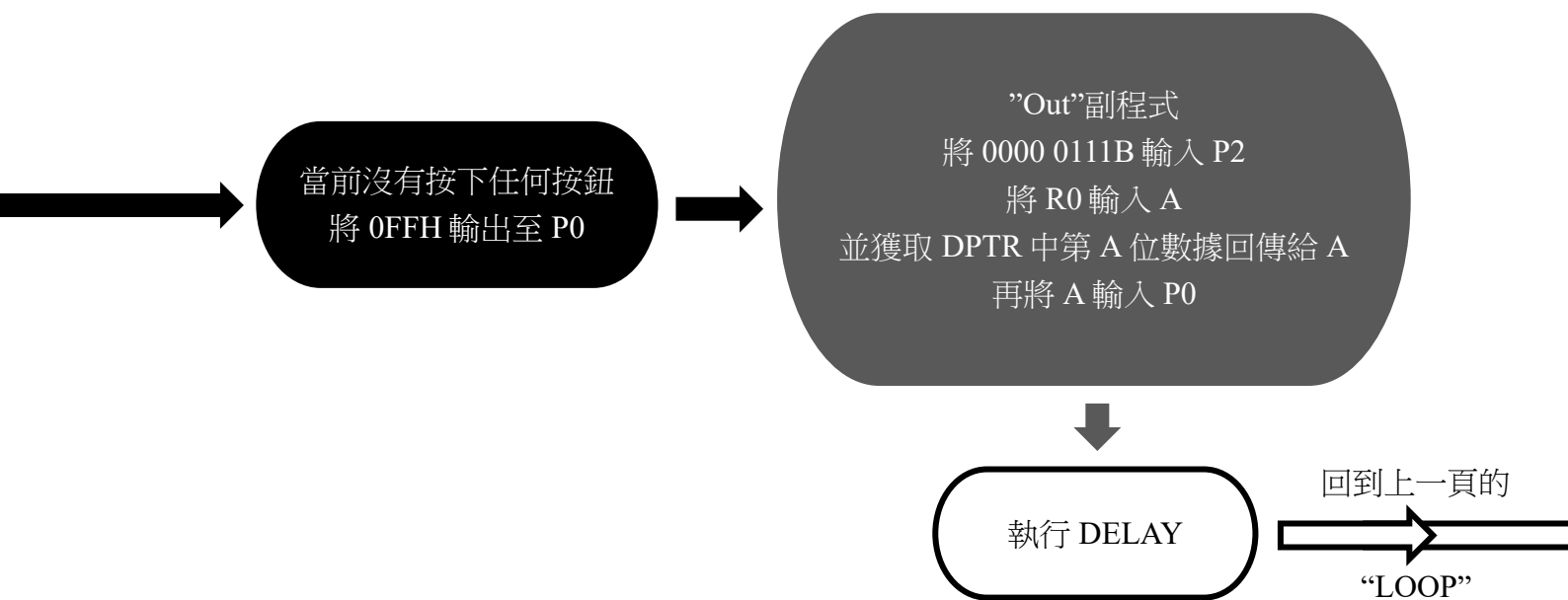
P2 連接七段顯示器的控制燈亮(D1~D4)。



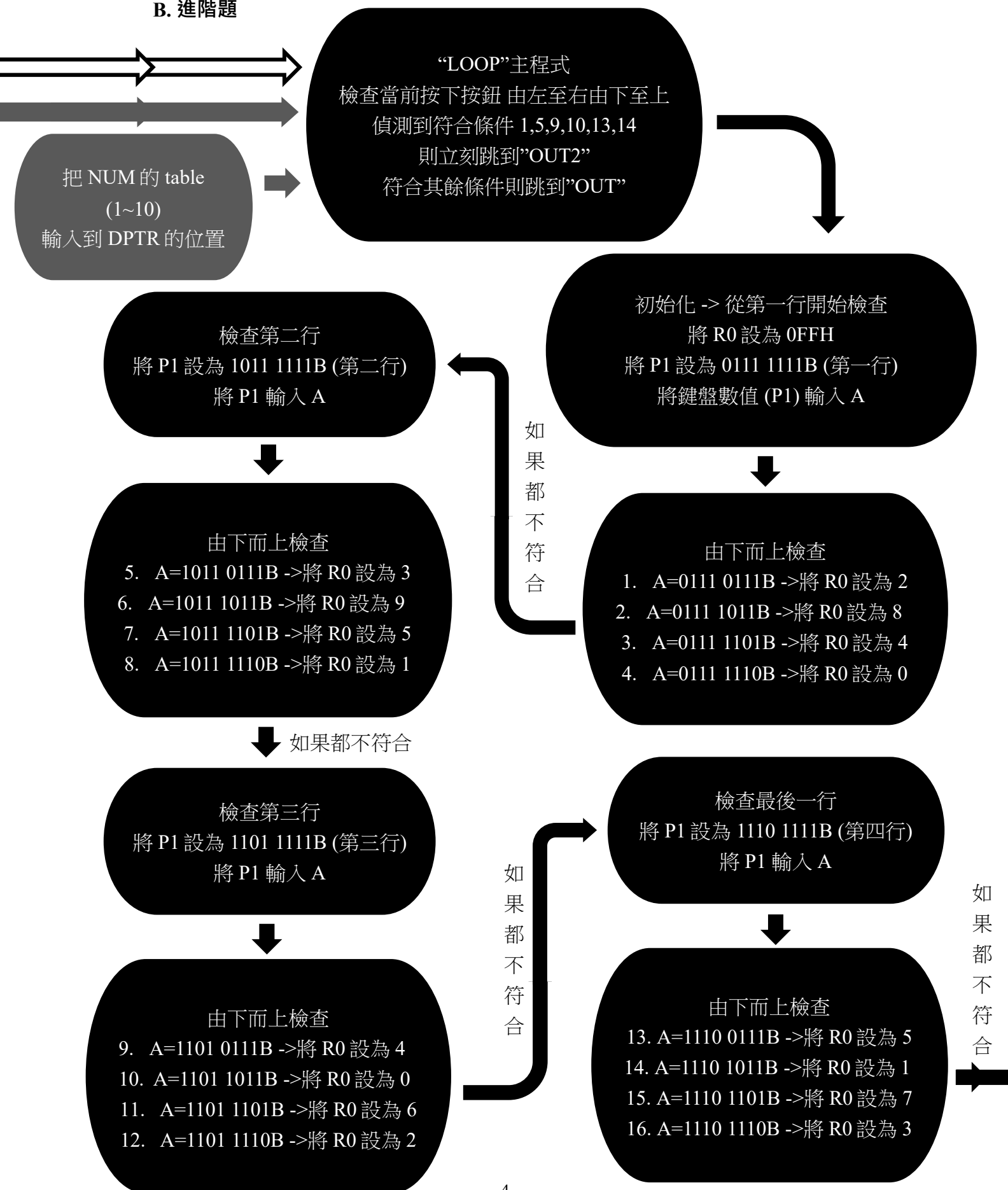
三、程式流程圖：

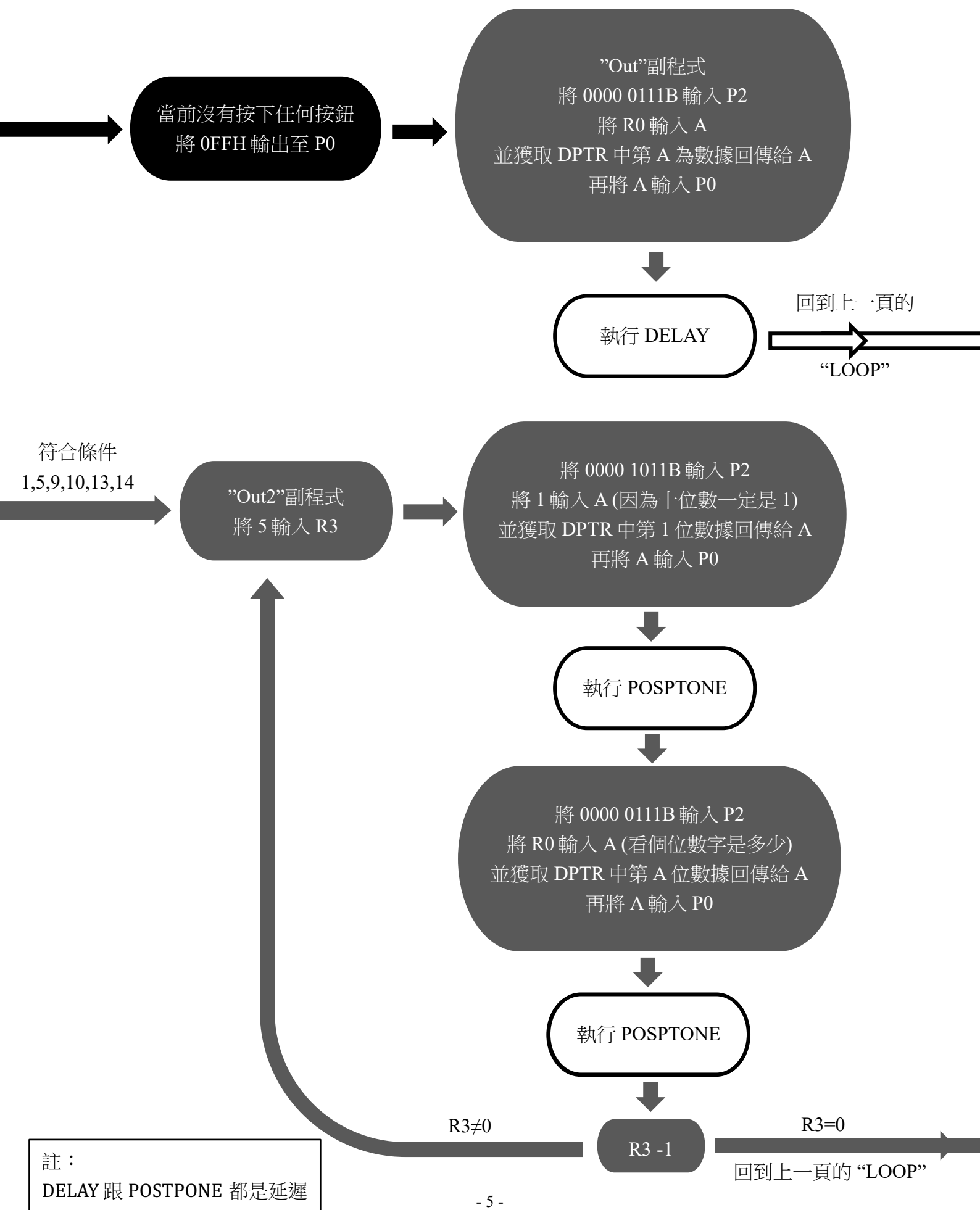
A. 基本題





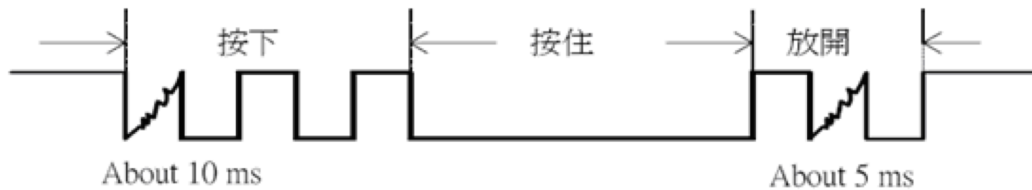
B. 進階題





四、問題與討論：

(1) 一般開關在按下之後，必然有機械振動使接點開(open)、閉(close)多次才穩定觸合，如下圖 3 所示為開關彈跳波形，如果產生以下波形時，應如何消除彈跳？請就軟體面(程式)詳細說明你解決開關彈跳的方法。



因為已經知道按下與放開時開關彈跳所需要的時間(10ms)，因此我們可以在 8051 接收到鍵盤訊號後加入一個時間大於彈跳時間的 Delay 副程式進行延遲，然後再重新掃描，找到被按下的按鍵位置。可以設一個暫存器記錄每次接收到的按鍵訊號，並在重新掃描之後比較當前信號與先前信號是否有不同，如果不同的話便代表有按鍵的狀態改變 (原本按下的放開了或是原本放開的按鍵被按下了)，藉此消除彈跳並接收鍵盤訊號。

(2) 在實際使用上，我們會賦予鍵盤每一個按鍵相對應的定義，而在應用上可能會需要同時按兩個按鍵來啟動某一個功能，例如：通過 CTRL+V 來複製資料，請問我們該如何以掃描讀取資料的方式為基礎，來讀取兩個按鍵？請詳細說明你對該問題的解析及相對應的解決方法，不需要附程式碼。

原本的程式碼是從左下角依序向上向右檢查，如果當前檢查到的按鈕有被按下去的話就跳到輸出。如果現在要一次讀取兩個按鍵的話我覺得可以在每次

的迴圈中，檢查到按下的按鈕先做紀錄，但先不要輸出，然後繼續檢查，直到檢查完全部 16 個按按鈕後，再分為只按一個按鈕的情況或按兩個按鈕的情況分別進行輸出。如果情況更複雜的話按兩個按鈕的情況要再進一步判斷它們的組合在進行相應的輸出。

五、程式碼與註解：

A、基本題

```
1  ORG 0000H
2  AJMP MAIN
3  ORG 0050H
4
5  MAIN:
6      MOV     DPTR, #NUM
7
8      //目標：從左下開始檢查 先往上再往右
9      //第一行 12->8->4->0
10     LOOP:
11         MOV R0, #0FFH //255
12         //P1只看前4位 0在哪裡就是哪行
13         MOV  P1, #01111111B //從第一行開始
14         MOV  A, P1 //看現在要輸出的數字是多少
15
16         //12
17         CJNE A, #01110111B, NEXT_8 //從最下面一列開始檢查
18         MOV  R0, #12 //個位數是2
19         AJMP OUT //十位數是1
20     NEXT_8:
21         CJNE A, #01111011B, NEXT_4 //換到倒數第二列
22         MOV  R0, #8
23         AJMP OUT
24     NEXT_4:
25         CJNE A, #01111101B, NEXT_0 //換到倒數第三列
26         MOV  R0, #4
27         AJMP OUT
28     NEXT_0:
29         CJNE A, #01111110B, NEXT_13 //換到最上面一列
30         MOV  R0, #0
31         AJMP OUT
32
33     //第二行 13->9->5->1
34     NEXT_13:
35         MOV  P1, #10111111B //換到第二行
36         MOV  A, P1
37         CJNE A, #10110111B, NEXT_9 //從最下面一列開始檢查
38         MOV  R0, #13 //E
39         AJMP OUT
40     NEXT_9:
41         CJNE A, #10111011B, NEXT_5 //換到倒數第二列
42         MOV  R0, #9
43         AJMP OUT
```



```

44 NEXT_5:
45     CJNE     A, #10111101B, NEXT_1 //換到倒數第三列
46     MOV      R0, #5
47     AJMP     OUT
48 NEXT_1:
49     CJNE     A, #1011110B, NEXT_14 //換到最上面一列
50     MOV      R0, #1
51     AJMP     OUT
52
53 //第三行 14->10->6->2
54 NEXT_14:
55     MOV      P1, #11011111B //換到第三行
56     MOV      A, P1
57     CJNE     A, #11010111B, NEXT_10 //從最下面一列開始檢查
58     MOV      R0, #14
59     AJMP     OUT
60 NEXT_10:
61     CJNE     A, #11011011B, NEXT_6 //換到倒數第二列
62     MOV      R0, #10
63     AJMP     OUT
64 NEXT_6:
65     CJNE     A, #11011101B, NEXT_2 //換到倒數第三列
66     MOV      R0, #6
67     AJMP     OUT
68 NEXT_2:
69     CJNE     A, #11011110B, NEXT_15 //換到最上面一列
70     MOV      R0, #2
71     AJMP     OUT
72

```

```

73 //第四行 15->11->7->3
74 NEXT_15:
75     MOV      P1, #11101111B //換到第四行
76     MOV      A, P1
77     CJNE     A, #11100111B, NEXT_11 //從最下面一列開始檢查
78     MOV      R0, #15
79     AJMP     OUT
80 NEXT_11:
81     CJNE     A, #11101011B, NEXT_7 //換到倒數第二列
82     MOV      R0, #11
83     AJMP     OUT
84 NEXT_7:
85     CJNE     A, #11101101B, NEXT_3 //換到倒數第三列
86     MOV      R0, #7
87     AJMP     OUT
88 NEXT_3:
89     CJNE     A, #11101110B, NEXT_N0 //換到最上面一列
90     MOV      R0, #3
91     AJMP     OUT
92

```

```

93 //都沒有按按鍵
94 NEXT_NO:
95     MOV     P0, #0FFH    //no
96
97 //數字顯示
98 OUT:
99     MOV     P2, #00000111B
100    MOV     A, R0
101    MOVC    A, @A+DPTR
102    MOV     P0, A
103    ACALL   DELAY
104    AJMP    LOOP
105
106 DELAY:
107     MOV     R5, #0FFH
108 DELAY1:
109     MOV     R6, #0FFH
110 DELAY2:
111     DJNZ    R6, DELAY2
112     DJNZ    R5, DELAY1
113     RET
114

```

```

115 NUM:
116     DB 1000000B //0
117     DB 11111001B //1
118     DB 10100100B //2
119     DB 10110000B //3
120     DB 10011001B //4
121     DB 10010010B //5
122     DB 10000010B //6
123     DB 11111000B //7
124     DB 10000000B //8
125     DB 10011000B //9
126
127     DB 10001000B //A
128     DB 10000011B //b
129     DB 11000110B //C
130     DB 10100001B //d
131     DB 10000110B //E
132     DB 10001110B //F
133 END

```

B、進階題

```

1  ORG    0000H
2  AJMP   MAIN
3  ORG    0050H
4
5  MAIN:
6      MOV    DPTR, #NUM
7
8  //目標：從左下開始檢查 先往上再往右
9  //第一行 12->8->4->0
10 LOOP:
11      MOV    R0, #0FFH
12      //P1只看前4位 0在哪裡就是哪行
13      MOV    P1, #01111111B //從第一行開始
14      MOV    A, P1 //看現在要輸出的數字是多少
15
16      //12
17      CJNE   A, #01110111B, NEXT_8 //從最下面一列開始檢查
18      MOV    R0, #2 //個位數是2
19      AJMP   OUT2 //十位數是1
20 NEXT_8:
21      CJNE   A, #01111011B, NEXT_4 //換到倒數第二列
22      MOV    R0, #8
23      AJMP   OUT
24 NEXT_4:
25      CJNE   A, #01111101B, NEXT_0 //換到倒數第三列
26      MOV    R0, #4
27      AJMP   OUT
28 NEXT_0:
29      CJNE   A, #01111110B, NEXT_13 //換到最上面一列
30      MOV    R0, #0
31      AJMP   OUT
32
33 //第二行 13->9->5->1
34 NEXT_13:
35      MOV    P1, #10111111B //換到第二行
36      MOV    A, P1
37      CJNE   A, #10110111B, NEXT_9 //從最下面一列開始檢查
38      MOV    R0, #3 //個位數是3
39      AJMP   OUT2 //十位數是1
40 NEXT_9:
41      CJNE   A, #10111011B, NEXT_5 //換到倒數第二列
42      MOV    R0, #9
43      AJMP   OUT

```

```

44 NEXT_5:
45     CJNE    A, #10111101B, NEXT_1 //換到倒數第三列
46     MOV     R0, #5
47     AJMP    OUT
48 NEXT_1:
49     CJNE    A, #10111110B, NEXT_14 //換到最上面一列
50     MOV     R0, #1
51     AJMP    OUT
52
53 //第三行 14->10->6->2
54 NEXT_14:
55     MOV     P1, #11011111B //換到第三行
56     MOV     A, P1
57     CJNE    A, #11010111B, NEXT_10 //從最下面一列開始檢查
58     MOV     R0, #4
59     AJMP    OUT2
60 NEXT_10:
61     CJNE    A, #11011011B, NEXT_6 //換到倒數第二列
62     MOV     R0, #0
63     AJMP    OUT2
64 NEXT_6:
65     CJNE    A, #11011101B, NEXT_2 //換到倒數第三列
66     MOV     R0, #6
67     AJMP    OUT
68 NEXT_2:
69     CJNE    A, #11011110B, NEXT_15 //換到最上面一列
70     MOV     R0, #2
71     AJMP    OUT
72

```

```

73 //第四行 15->11->7->3
74 NEXT_15:
75     MOV     P1, #11101111B //換到第四行
76     MOV     A, P1
77     CJNE    A, #11100111B, NEXT_11 //從最下面一列開始檢查
78     MOV     R0, #5
79     AJMP    OUT2
80 NEXT_11:
81     CJNE    A, #11101011B, NEXT_7 //換到倒數第二列
82     MOV     R0, #1
83     AJMP    OUT2
84 NEXT_7:
85     CJNE    A, #11101101B, NEXT_3 //換到倒數第三列
86     MOV     R0, #7
87     AJMP    OUT

```

```

88 NEXT_3:
89     CJNE    A, #11101110B, NEXT_N0 //換到最上面一列
90     MOV     R0, #3
91     AJMP    OUT
92
93     //都沒有按按鍵
94 NEXT_N0:
95     MOV     P0, #0FFH
96
97     //0-9數字顯示
98 OUT:
99     MOV     P2, #00000111B
100    MOV     A, R0
101    MOVC    A, @A+DPTR
102    MOV     P0, A
103    ACALL   DELAY
104    AJMP    LOOP
105
106    //兩位數數字顯示
107 OUT2:
108    MOV     R3, #5
109
110 L1:
111     //十位數
112     MOV     P2, #00001011B
113     MOV     A, #01 //因為十位數一定是1 所以直接把1輸入至A
114     MOVC    A, @A+DPTR
115     MOV     P0, A
116     ACALL   POSTPONE
117     //個位數
118     MOV     P2, #00000111B
119     MOV     A, R0 //看現在按的按鈕的個位數是多少
120     MOVC    A, @A+DPTR
121     MOV     P0, A
122     ACALL   POSTPONE
123     DJNZ    R3, L1
124     AJMP    LOOP
125
126     /*因為印二位數時會需要讓肉眼同時看見兩個七段顯示器
127     | 因此用到兩個延遲讓肉眼產生視覺暫留*/
128 DELAY:
129     MOV     R5, #0FFH
130 DELAY1:
131     MOV     R6, #0FFH
132 DELAY2:
133     DJNZ    R6, DELAY2
134     DJNZ    R5, DELAY1
135     RET

```

```

136 POSTPONE:
137     MOV     R5, #0FFH
138 POSTPONE1:
139     DJNZ    R5, POSTPONE1
140     RET
141
142 //七段顯示器的數字表
143 NUM:
144     DB 1000000B; //0
145     DB 11111001B; //1
146     DB 10100100B; //2
147     DB 10110000B; //3
148     DB 10011001B; //4
149     DB 10010010B; //5
150     DB 10000010B; //6
151     DB 11111000B; //7
152     DB 10000000B; //8
153     DB 10011000B; //9
154 END

```

六、心得：

1. 對於上課內容的心得感想：

以往對鍵盤的認知就是上面有幾個按鈕就需要有幾個控制器去實作它，沒有想過可以結合之前在邏設學到的概念，分成導通跟不導通，考慮直排與橫列的相互組合，如此一來只需要用兩個控制項便可以實作出鍵盤的邏輯，而在這次的實驗中因為鍵盤較小，所以可以將兩個合併連接到同一個 Port，就只需要用一個暫存器就可以表示鍵盤按鈕導通位置。

2. 對於實驗內容的心得感想：

這次實驗的基礎題跟進階題沒有之前的差別這麼大，主要的思考邏輯都一樣，只是在進階題的部分因為要讓肉眼可以一次看到兩個七段顯示器亮起來，所以需要加入視覺暫留的考慮。換句話說也就是只要把第二題實做出來第一題也可以用第二題的程式碼實現（只需要更改輸出的部分）。