

微算機實驗報告

系級: 資工 114

姓名:洪巧芸

Lab #06

學號:110550143

上課時間:2023/10/31

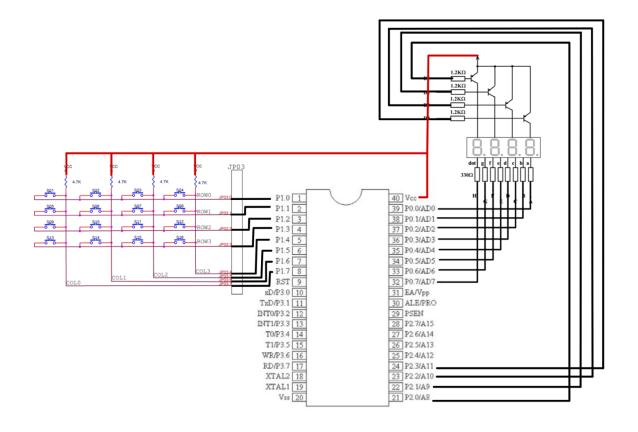
一、實驗目的:

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)。



A. 基本題

把 NUM 的 table (1~10, A~F) 輸入到 DPTR 的位置 "LOOP"主程式 檢查當前按下按鈕 由左至右由下至上 偵測到符合條件則立刻跳出迴圈 執行"OUT"副程式

如果都不

符

合

初始化 -> 從第一行開始檢查 將 R0 設為 0FFH 將 P1 設為 0111 1111B (第一行) 將鍵盤數值 (P1) 輸入 A

檢查第二行 將 P1 設為 1011 1111B (第二行) 將 P1 輸入 A



由下而上檢查符合就跳到"OUT"

- 5. A=1011 0111B ->將 R0 設為 d
- 6. A=1011 1011B ->將 R0 設為 9
- 7. A=1011 1101B ->將 R0 設為 5
- 8. A=1011 1110B ->將 R0 設為 1

將 P1 設為 0111 11111 將鍵盤數值 (P1) [



- 1. A=0111 0111B ->將 R0 設為 C
- 2. A=0111 1011B ->將 R0 設為 8
- 3. A=0111 1101B ->將 R0 設為 4
- 4. A=0111 1110B ->將 R0 設為 0

■ 如果都不符合

檢查第三行 將 P1 設為 1101 1111B (第三行) 將 P1 輸入 A



由下而上檢查 符合就跳到"OUT" 9. A=1101 0111B ->將 R0 設為 E 10. A=1101 1011B ->將 R0 設為 A 11. A=1101 1101B ->將 R0 設為 6 12. A=1101 1110B ->將 R0 設為 2

如

果

都

不

符

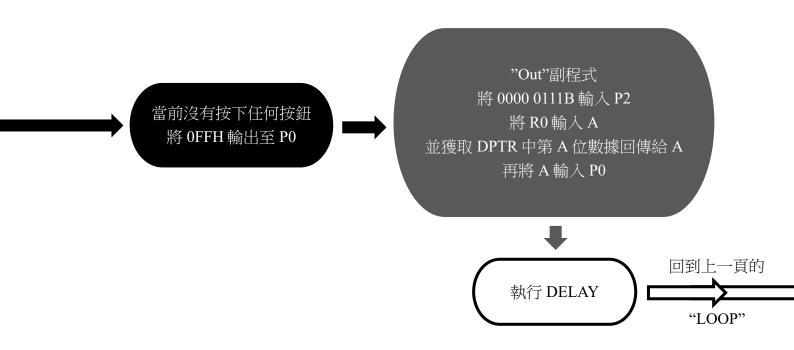
合

檢查最後一行 將 P1 設為 1110 1111B (第四行) 將 P1 輸入 A



由下而上檢查 符合就跳到"OUT" 13. A=1110 0111B ->將 R0 設為 F 14. A=1110 1011B ->將 R0 設為 b 15. A=1110 1101B ->將 R0 設為 7 16. A=1110 1110B ->將 R0 設為 3

- 2 -





把 NUM 的 table $(1\sim10)$ 輸入到 DPTR 的位置

"LOOP"主程式 檢查當前按下按鈕 由左至右由下至上 偵測到符合條件 1,5,9,10,13,14 則立刻跳到"OUT2" 符合其餘條件則跳到"OUT"

> 如 果 都 不

符

合



檢查第二行 將 P1 設為 1011 1111B (第二行) 將 P1 輸入 A



由下而上檢查

- 5. A=1011 0111B ->將 R0 設為 3
- 6. A=1011 1011B ->將 R0 設為 9
- 7. A=1011 1101B ->將 R0 設為 5
- 8. A=1011 1110B ->將 R0 設為 1

初始化 -> 從第一行開始檢查 將 R0 設為 0FFH 將 P1 設為 0111 1111B (第一行) 將鍵盤數值 (P1) 輸入 A



由下而上檢查

- 1. A=0111 0111B ->將 R0 設為 2
- 2. A=0111 1011B ->將 R0 設為 8
- 3. A=0111 1101B ->將 R0 設為 4
- 4. A=0111 1110B ->將 R0 設為 0

如果都不符合

檢查第三行 將 P1 設為 1101 1111B (第三行) 將 P1 輸入 A



由下而上檢查

- 9. A=1101 0111B ->將 R0 設為 4
- 10. A=1101 1011B ->將 R0 設為 0
- 11. A=1101 1101B ->將 R0 設為 6
- 12. A=1101 1110B ->將 R0 設為 2

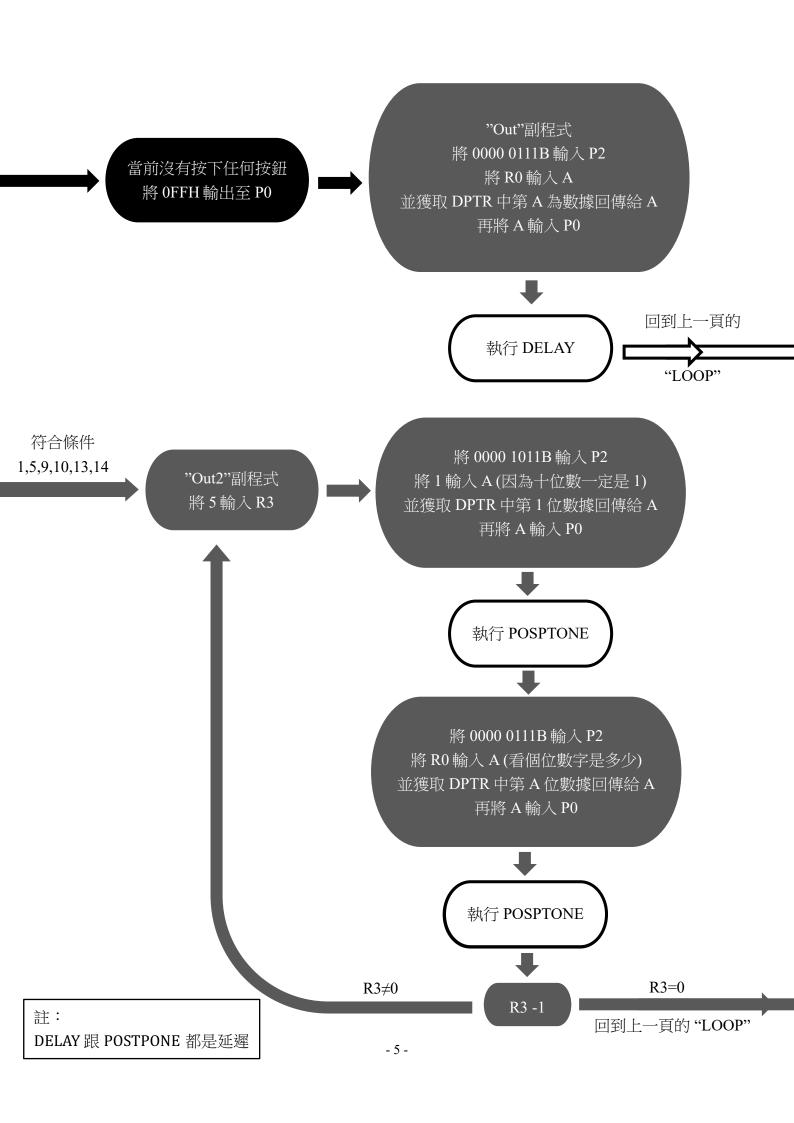
如 果 都 不 符

檢查最後一行 將 P1 設為 1110 1111B (第四行) 將 P1 輸入 A



由下而上檢查

- 13. A=1110 0111B ->將 R0 設為 5
- 14. A=1110 1011B ->將 R0 設為 1
- 15. A=1110 1101B ->將 R0 設為 7
- 16. A=1110 1110B ->將 R0 設為 3



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。



因爲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

```
ORG 0000H
 2
     AJMP MAIN
     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
                A, P1 //看現在要輸出的數字是多少
        MOV
15
16
         //12
17
         CJNE
                A, #01110111B, NEXT_8 //從最下面一列開始檢查
18
                R0, #12 //個位數是2
        MOV
19
         AJMP
                0UT //十位數是1
20
     NEXT_8:
21
         CJNE A, #01111011B, NEXT_4 //換到倒數第二列
        MOV R0, #8
22
         AJMP OUT
23
24
     NEXT_4:
25
         CJNE A, #01111101B, NEXT_0 //換到倒數第三列
26
         MOV R0, #4
         AJMP OUT
27
28
     NEXT_0:
29
         CJNE A, #01111110B, NEXT_13 //換到最上面一列
30
        MOV R0, #0
         AJMP OUT
31
32
     //第二行 13->9->5->1
33
34
     NEXT_13:
35
         MOV
                P1, #10111111B //換到第二行
36
         MOV
                A, P1
                A, #10110111B, NEXT_9 //從最下面一列開始檢查
37
         CJNE
                R0, #13 //E
38
         MOV
39
         AJMP
                0UT
40
     NEXT_9:
                 A, #10111011B, NEXT_5 //換到倒數第二列
41
         CJNE
```

MOV

AJMP

42 43 R0, #9

0UT

```
44
     NEXT_5:
45
         CJNE
                 A,#10111101B, NEXT_1 //換到倒數第三列
46
         MOV
                 R0,#5
47
         AJMP
                 0UT
48
     NEXT_1:
                 A, #10111110B, NEXT_14 //換到最上面一列
49
         CJNE
50
         MOV
                 R0,#1
51
         AJMP
                 0UT
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
                 A, #11010111B, NEXT_10 //從最下面一列開始檢查
         CJNE
58
         MOV
                 R0, #14
59
         AJMP
                 0UT
     NEXT_10:
60
61
         CJNE
                 A, #11011011B, NEXT_6 //換到倒數第二列
62
         MOV
                 R0, #10
63
                 0UT
         AJMP
64
     NEXT_6:
65
                 A, #11011101B, NEXT_2 //換到倒數第三列
         CJNE
66
         MOV
                 RØ, #6
67
         AJMP
                 0UT
68
     NEXT_2:
69
                 A, #11011110B, NEXT_15 //換到最上面一列
         CJNE
70
         MOV
                 R0,#2
71
                 0UT
         AJMP
72
     //第四行 15->11->7->3
74
     NEXT_15:
75
         MOV
                 P1, #11101111B //換到第四行
76
         MOV
                 A, P1
77
                 A, #11100111B, NEXT_11 //從最下面一列開始檢查
         CJNE
                 R0, #15
78
         MOV
79
         AJMP
                 OUT
80
     NEXT_11:
81
                 A, #11101011B, NEXT_7 //換到倒數第二列
         CJNE
82
         MOV
                 R0, #11
83
         AJMP
                 0UT
     NEXT_7:
84
85
         CJNE
                 A, #11101101B, NEXT_3 //換到倒數第三列
                 R0, #7
86
         MOV
87
         AJMP
                 0UT
88
     NEXT_3:
89
         CJNE
                 A,#11101110B, NEXT_NO //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
                   L00P
105
106
      DELAY:
107
          MOV
                   R5, #0FFH
108
       DELAY1:
109
          MOV
                   R6, #0FFH
110
      DELAY2:
111
           DJNZ
                   R6, DELAY2
                   R5, DELAY1
112
           DJNZ
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
           DB 11000110B //C
129
130
           DB 10100001B //d
131
           DB 10000110B //E
132
           DB 10001110B //F
133
      END
```

B、進階題

```
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
     //第一行 12->8->4->0
 9
10
     LOOP:
11
                     R0, #0FFH
             MOV
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
                     0UT2 //十位數是1
20
     NEXT_8:
21
             CJNE
                     A, #01111011B, NEXT_4 //換到倒數第二列
22
                     R0, #8
             MOV
23
             AJMP
                     0UT
24
     NEXT_4:
25
             CJNE
                     A, #01111101B, NEXT_0 //換到倒數第三列
26
                     R0, #4
             MOV
27
                     0UT
             AJMP
28
     NEXT_0:
29
                     A, #01111110B, NEXT_13 //換到最上面一列
             CJNE
30
             MOV
                     R0, #0
31
             AJMP
                     0UT
32
     //第二行 13->9->5->1
33
34
     NEXT_13:
35
                   P1, #10111111B //換到第二行
            MOV
36
            MOV
37
            CJNE
                   A, #10110111B, NEXT_9 //從最下面一列開始檢查
38
                   R0, #3 //個位數是3
            MOV
39
                   OUT2 //十位數是1
            AJMP
40
     NEXT_9:
41
            CJNE
                   A, #10111011B, NEXT_5 //換到倒數第二列
42
            MOV
                   R0, #9
                   0UT
43
            AJMP
```

```
44
     NEXT_5:
                     A, #10111101B, NEXT_1 //換到倒數第三列
45
             CJNE
46
             MOV
                     R0, #5
47
             AJMP
                     0UT
48
     NEXT_1:
                     A, #10111110B, NEXT_14 //換到最上面一列
49
             CJNE
                     R0, #1
50
             MOV
             AJMP
                     0UT
51
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
                     A, #11010111B, NEXT_10 //從最下面一列開始檢查
             CJNE
58
             MOV
                     R0, #4
59
             AJMP
                     0UT2
60
     NEXT_10:
61
             CJNE
                     A, #11011011B, NEXT_6 //換到倒數第二列
62
             MOV
                     R0, #0
63
             AJMP
                     0UT2
64
     NEXT_6:
65
                     A, #11011101B, NEXT_2 //換到倒數第三列
             CJNE
66
             MOV
                     R0, #6
                     0UT
67
             AJMP
68
     NEXT_2:
69
             CJNE
                     A, #11011110B, NEXT_15 //換到最上面一列
70
                     R0, #2
             MOV
                     0UT
71
             AJMP
72
```

```
//第四行 15->11->7->3
73
74
     NEXT_15:
75
                     P1, #11101111B //換到第四行
             MOV
76
             MOV
                     A, P1
77
                     A,#11100111B, NEXT_11 //從最下面一列開始檢查
             CJNE
78
             MOV
                     R0,#5
79
             AJMP
                     0UT2
80
     NEXT_11:
81
                     A, #11101011B, NEXT_7 //換到倒數第二列
             CJNE
82
             MOV
                     R0, #1
83
             AJMP
                     0UT2
84
     NEXT_7:
85
             CJNE
                     A, #11101101B, NEXT_3 //換到倒數第三列
                     R0, #7
86
             MOV
87
             AJMP
                     0UT
```

```
NEXT_3:
 88
 89
                     A, #11101110B, NEXT_NO //換到最上面一列
              CJNE
 90
              MOV
                     R0, #3
                     0UT
 91
              AJMP
 92
 93
      //都沒有按按鍵
 94
      NEXT_NO:
 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
                     L00P
105
106
      //兩位數數字顯示
107
      0UT2:
108
              MOV
                     R3, #5
109
110
      L1:
111
              //十位數
112
                      P2, #00001011B
              MOV
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
                      A, @A+DPTR
              MOVC
121
              MOV
                      P0, A
122
              ACALL
                      POSTPONE
123
              DJNZ
                     R3, L1
124
              AJMP
                      L00P
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
      DELAY2:
132
133
                     R6, DELAY2
              DJNZ
134
              DJNZ
                      R5, DELAY1
135
              RET
```

```
136
       POSTPONE:
137
               MOV
                       R5, #0FFH
138
       POSTPONE1:
                       R5, POSTPONE1
139
               DJNZ
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 1000000B; //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)。