# 微處理機 Lab6 結報

0413335 郭逸琳 0416039 李佳燕

## 實驗主題與訓練目的

透過這次實驗學會如何使用4x4 Keypad Scanning，並能透過按鍵在七段式顯示器上顯示相對應的數字。此實驗有兩個版本要實作，第一個版本按按鍵時要顯示的相對應數字如下圖:

x x x x

8 9 x x

4 5 6 7

0 1 2 3

第二個版本按按鍵時要顯示的相對應數字如下圖:

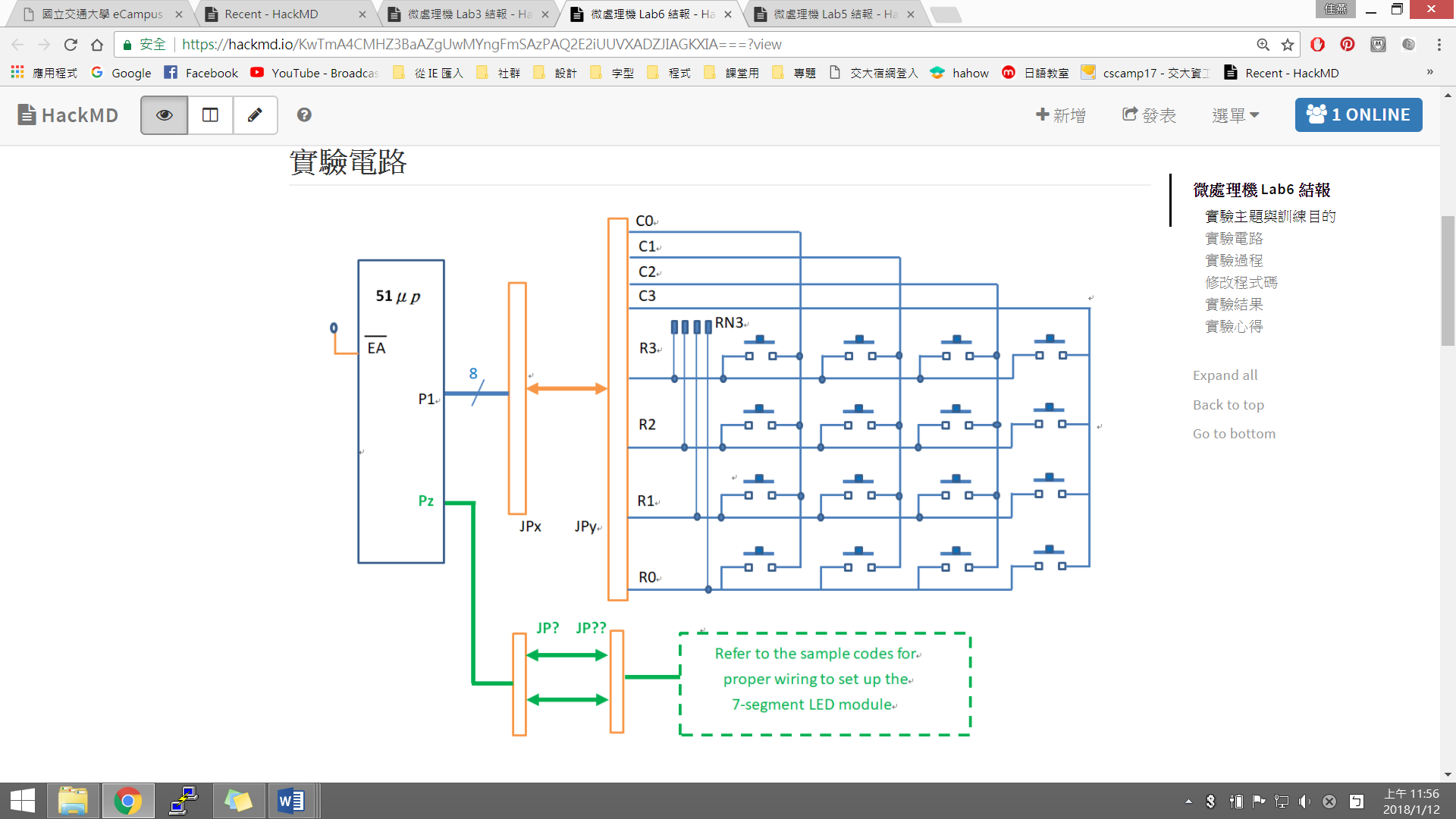
12 13 14 15

08 09 10 11

04 05 06 07

00 01 02 03

## 實驗電路



## 實驗過程

在uvision裡貼上spec裡的程式碼並遵照以下方法修改程式。按按鍵時會把值傳入P1裡面，然後P1會把值給A，A經過一些轉換後會進入keycoding的function，然後在keycoding的funciton中會再經過轉換使A成為我們想要的值(如果是 vision2 的話會把想要的值拆成十位樹跟個位數再分別給A跟B)，最後再傳入P2顯示答案。

vision1的時候P2可以只接七段式顯示器的上排，只要能在某一個位數顯示想要的數字即可(用一個pin腳的線接到電路中預設是0的位置上就好了)。但因為vision2中要分別顯示兩個位數的東西，需要控制哪一個位置要顯示哪一個數字，所以需要將P2拆成兩個4pin的線，其中一條接到七段式顯示器的上排，另外一條接到七段式顯示器的下排。

## 修改程式碼

### VERSION1

keycoding裡:

anl #0FH 改成->anl A, #0FH

jnb A.2, cont 改成->jnb ACC.2, cont

display裡:

oal A, #0E0H 改成->orl A, #0E0H

### VERSION2

除了VERSION1要改的地方要改之外

col0裡:

要加上

orl A, #0FH

swap A

display裡:

push PSW

push A 改成->push ACC

push F0H ; push B?? 改成->push 0F0H

mov A, R7

mov B, #10

div A, B 改成->div AB

anl A, #0FH

orl A, #0D0H

mov P2, A

call delay

mov A, B

anl A, #0FH

orl A, #0E0H

mov P2, A

call delay

pop F0H 改成->pop 0F0H

pop A 改成->pop ACC

pop PSW

ret

## 實驗結果

### VERSION1

x x x x

8 9 x x

4 5 6 7

0 1 2 3

### VERSION2

12 13 14 15

08 09 10 11

04 05 06 07

00 01 02 03

## 實驗心得

一開始修改好code可以編譯之後, 跑出來的結果是row跟column顛倒,(跟spec上面要求的相反)

但老師上面有寫註解：

P1: 0-3 output for column scanning

P1: 4-7 input for row reading

可以發現, 只要P1的0-3bit接到keypad的4-7bit;

P1的4-7bit接到keypad的0-3bit就可以把row跟column顛倒回來, 達成spec要的結果。

在改version2的程式碼的時候我們那時一直沒有找到bug，導致我們一直跑出錯誤的結果，後來發現其實bug沒有我們想像中的那個難找，只要稍微比對col0跟col1就可發現錯的地方，如果我們再細心一點就可以更有效率地找到bug了。

; =================================

; version 1 key-code display on one 7-seg

; digit

; x x x x

; 8 9 x x

; 4 5 6 7

; 0 1 2 3

; Xs: non-decimal 7-seg patterns

; =================================

; port 1 for keypad scanning

; P1: 0-3 output for column scanning

; P1: 4-7 input for row reading

; port 2 for using one digit of 7-seg LED

org 0

mov sp, #50H

mov P2, #0FFH

col0: mov R6, #0

mov P1, #0FEH

mov A, P1

orl A, #0FH

swap A

cpl A

jnz keycoding

col1: mov R6, #1

mov P1, #0FDH

mov A, P1

orl A, #0FH

swap A

cpl A

jnz keycoding

col2: mov R6, #2

mov P1, #0FBH

mov A, P1

orl A, #0FH

swap A

cpl A

jnz keycoding

col3: mov R6, #3

mov P1, #0F7H

mov A, P1

orl A, #0FH

swap A

cpl A

jz col0

keycoding: ; A: 0-9,A,B,C,D,E F

rr A

anl A, #0FH

jnb ACC.2, cont

mov A, #3

cont: jz cont2

mov R7, A

clr A

cont1: add A, #4

djnz R7, cont1

cont2: add A, R6

display: ; when A>9 ???

orl A, #0E0H

mov P2, A

jmp col0

end

; =============================

; =================================

; version 2 key-code display on two 7-seg

; digits

; 12 13 14 15

; 08 09 10 11

; 04 05 06 07

; 00 01 02 03

; =================================

; port 1 for keypad scanning

; P1: 0-3 output for column scanning

; P1: 4-7 input for row reading

; port 2 for using two digits of 7-seg LED

org 0

mov sp, #50H

mov R7, #15

mov P2, #0FFH

col0: mov R6, #0

mov P1, #0FEH

mov A, P1

orl A, #0FH

swap A

cpl A

jnz keycoding

call display ; ==??==

col1: mov R6, #1

mov P1, #0FDH

mov A, P1

orl A, #0FH

swap A

cpl A

jnz keycoding

call display ; ==??==

col2: mov R6, #2

mov P1, #0FBH

mov A, P1

orl A, #0FH

swap A

cpl A

jnz keycoding

call display ; ==??==

col3: mov R6, #3

mov P1, #0F7H

mov A, P1

orl A, #0FH

swap A

cpl A

jz col0

keycoding: ; A: 0-9,A,B,C,D,E F

rr A

anl A, #0FH

jnb ACC.2, cont

mov A, #3

cont: jz cont2

mov R7, A

clr A

cont1: add A, #4

djnz R7, cont1

cont2: add A, R6

mov R7, A

call display ; ==??==

jmp col0

display:

push PSW

push ACC

push 0F0H ; push B??

mov A, R7

mov B, #10

div AB

anl A, #0FH

orl A, #0D0H

mov P2, A

call delay

mov A, B

anl A, #0FH

orl A, #0E0H

mov P2, A

call delay

pop 0F0H

pop ACC

pop PSW

ret

delay: push 2

push 3

mov R2, #20

xxx: mov R3, #250

djnz r3, $

djnz r2, xxx

pop 3

pop 2

ret

end