**微算機實驗報告**



姓名：滕瑋安

學號：0510743

系級：電機09

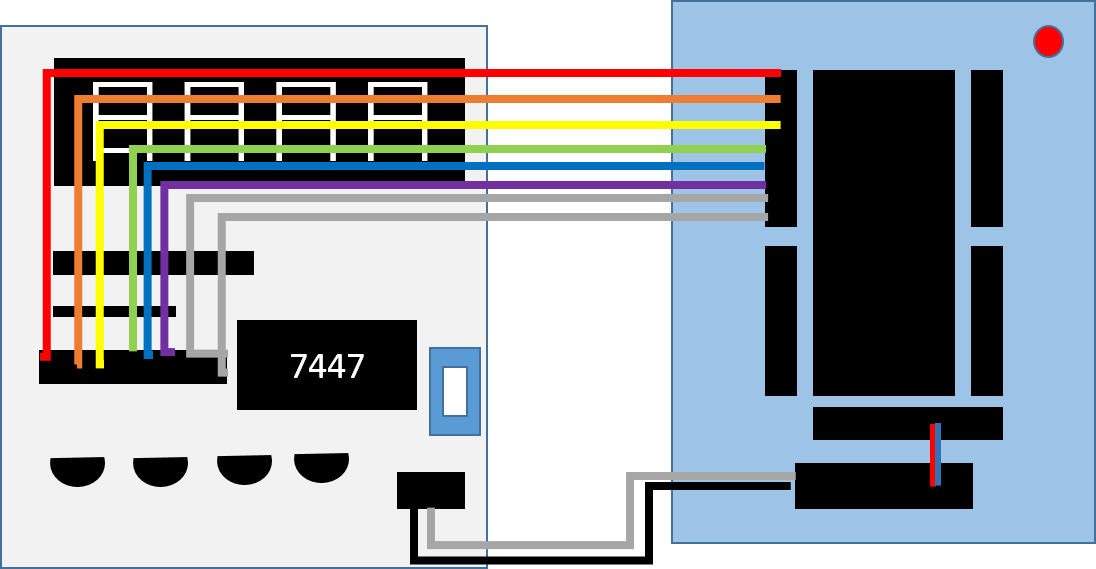
上課日期：10/17

Lab #4

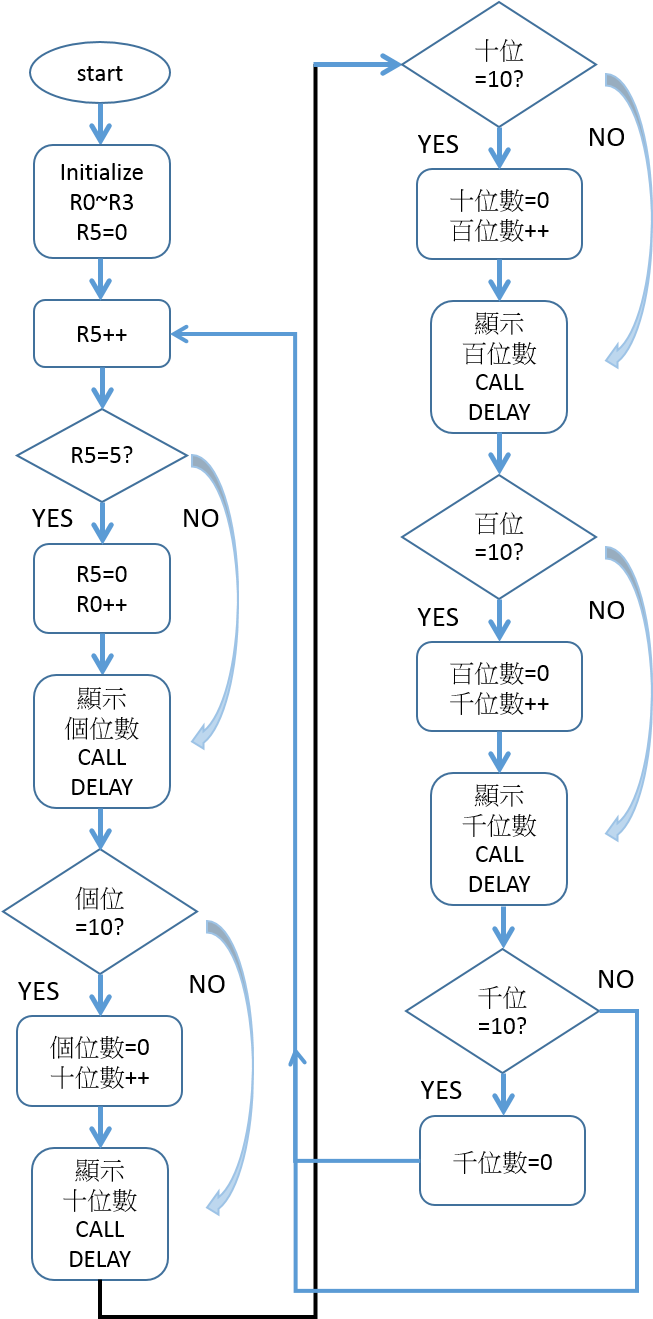
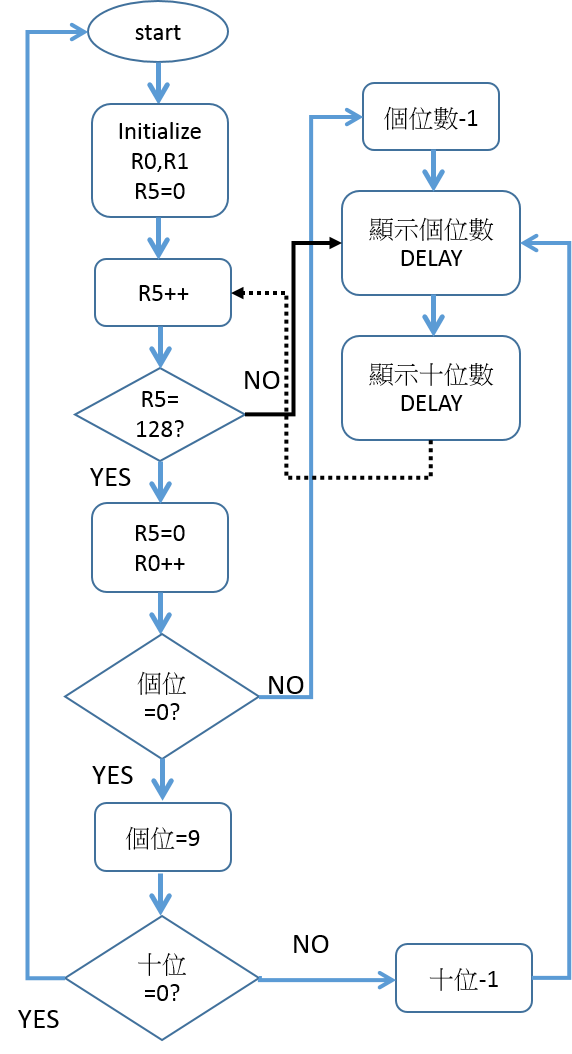
**ㄧ、實驗目的：**

學習運用視覺暫留的方法將數字印在四個七段顯示器上。

**二、硬體架構：**

****

**三、程式流程圖：**

****

**四、問題與討論：**

**若要使用一個開關來控制計數器 run 或 stop，要如何設計？**

把開關接到P0，四個七段顯示器接在P1。

在每次掃描印出各個位數的程序後面，加上判斷P0腳位的指令，若開關是開的就將掃描計數器往回加/減1，不是的話就繼續印。

**五、程式碼與註解：**

第一題：

ORG 0000H

JMP START

ORG 50H

START:

MOV R0,#70H ;01110000B X X X N

MOV R1,#0B9H ;10110000B X X N X

MOV R2,#0D9H ;11010000B X N X X

MOV R3,#0E9H ;11100000B N X X X

MOV R5,#00H ;掃描次數(5次

MAIN1:

INC R5

CJNE R5,#5,MAIN2 ;if R5!=5 JMP MAIN2

MOV R5,#00H ;掃描五次後 R5=0

INC R0 ;個位數++

MAIN2:

MOV P1,R0 ;顯示個位數

CALL DELAY

CJNE R0,#7AH,SHOW1 ;個位數不是10, 就顯示十位數

INC R1 ;個位數=10就進位到十位數

MOV R0,#70H ;01110000B ;導通個位數 個位數回到0

SHOW1:

MOV P1,R1 ;顯示十位數

CALL DELAY

CJNE R1,#0BAH,SHOW2 ;十位數不是10, 顯示十位數((上數

INC R2 ;十位數進位到百位數

MOV R1,#0B0H

SHOW2:

MOV P1,R2

CALL DELAY

CJNE R2,#0DAH,SHOW3 ;百位數是10的話跳到千位

INC R3 ;++千位

MOV R2,#0D0H

SHOW3:

MOV P1,R3

CALL DELAY

CJNE R3,#0EAH,MAIN1 ;不為10就回去從頭開始掃描

MOV R3,#0E0H

JMP MAIN1

DELAY:

MOV R6,#0AFH

DELAY1:

MOV R7,#0FH

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

RET

END

第二題:

ORG 0000H

JMP START

ORG 50H

START:

MOV R0,#70H ;01110000B X X X N

MOV R1,#0B6H ;10110000B X X N X

MOV R5,#00H

MAIN1:

INC R5

CJNE R5,#0FFH,MAIN2 ;if R5!=FF JMP MAIN2

MOV R5,#00H ;掃描FF次後 R5=0

CJNE R0,#70H,OZ ;如果個位數不是0就跳到OZ

MOV R0,#79H ;如果個位數是0就變成9

CJNE R1,#0B0H,TENZ ;如果十位數不是0就跳到TENZ

JMP START ;如果十位數是0 也就是數到都是0，就回到初始設定

TENZ:

;十位數!=0

DEC R1

JMP MAIN2

OZ:

;個位數!=0

DEC R0

MAIN2:

MOV P1,R0 ;顯示個位數

CALL DELAY

SHOW1:

MOV P1,R1 ;顯示十位數

CALL DELAY

JMP MAIN1

DELAY:

MOV R6,#0FFH

DELAY1:

MOV R7,#0FH

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

RET

END

**六、心得：**

這次的範例程式在看完之後還是霧煞煞的，畢竟前幾次的實驗都比較簡單，這次開始就加入了一些比較新穎的概念，尤其是在「掃描」的部分。最開始照著範例程式打下來，打完了個位數以後，我就先把範例遮住，自己試著打出十位數到千位數的邏輯，在發現成功了以後覺得滿開心的，有種又增加了一門知識的感覺。

然而在第二題的下數計時器發生了判斷式的小問題。在確認程式的確可以下數之後，發現10的整數位數竟然都只會跑一次的DELAY，前前後後重寫好幾次的程式，一次比一次精準，直到倒數第二個的時候，發現只差60不會顯示，從59數完又回到59，卻閃爍的很嚴重。當下的自己只覺得或許先把60印出來就好了，在寫了一段印出60的程式以後，卻只有60閃的最久，閃完以後才從59開始下數。為了這個問題，我花了約兩個小時去尋找問題的根源。最後再重新跑一次邏輯時，發現59~1的下數閃爍情形嚴重，是因為我把 DEC R0 放錯地方，導致每個數字只會經歷一次的掃描就更改成下個數字。發現這個萬惡的“小bug”之後，把它放回判斷式前面，再修改了一次邏輯以後終於正確了！

然而這次並沒有時間挑戰BONUS，我打算回去自己寫寫看，並與已經成功的同學討論程式的邏輯運作，找出一個最有系統的寫法。

而雖然我自己本來也沒有多厲害，但剛好這學期從別系轉來一個朋友，她有時也會找我幫忙看程式，所以我也會再確認自己邏輯正確以後，幫她看一下程式的邏輯問題。

希望以後我可以早點打完自己的程式，這樣就有比較多時間去思考BONUS，也可以快點幫助朋友憶起導通邏輯。

而這次發現的一個小問題：

ORG 00H下一個是JMP的指令，然而之後的程式都不會回到JMP後面的ORG 50H，那為什麼要把ORG 50H放在那裏？

原因是因為主譯器在瀏覽了整個程式以後，並不是像C語言一樣照著順序讀下來；程式還是會在50H開始放。