**微算機實驗報告格式**

**姓姓名：陳達軒**

**學號：0610837**

**上課時間：**

**2018 Fall 2EF**



Lab # 1

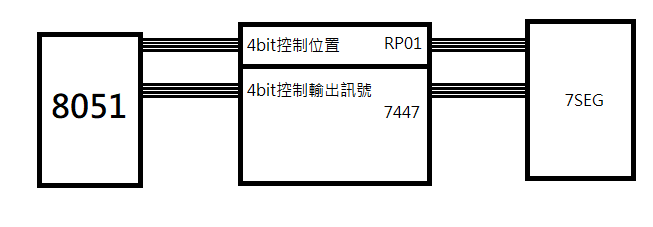
**ㄧ、實驗目的：**

但請以自己對本次實驗的認知來加以說明。

學習如何一次使用4顆七段顯示器，和如何將一筆資料同時做多種用途（控制、傳輸資料）。

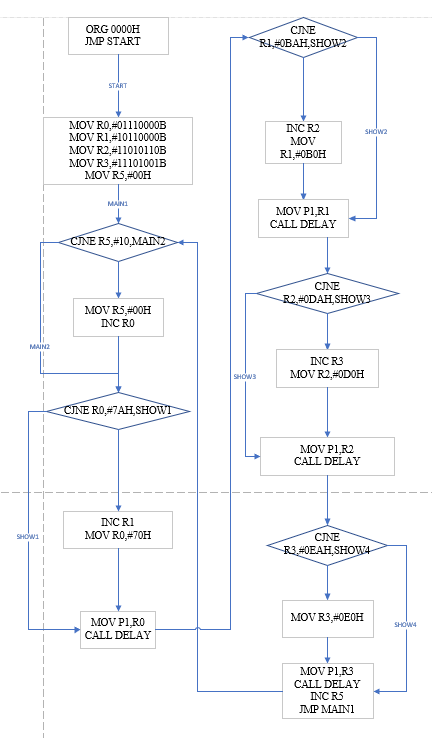
**二、硬體架構：**

請儘量自己畫圖不管是手繪或者是以電腦繪圖(Word or PoworPoint)。

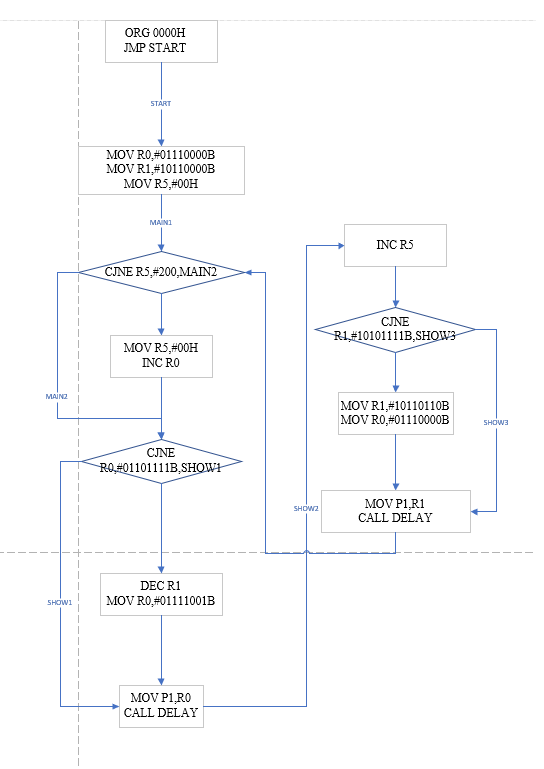


**三、程式流程圖：**

基礎題



進階題



**四、問題與討論：**

Note:

1.回答助教在實驗講義中所提的問題。

2.自己發現的疑問以及如何解決的方法。

我認為可以在迴圈中多寫入一個迴圈，這個迴圈會每次讀取開關的值放到，如果不是0的話，就重複執行這個步驟，不會跳出迴圈，也導致delay的過程會卡在這個迴圈中，導致delay無法結束，相等於stop。如果打開開關，使得讀到的值變成0的話，那麼這個迴圈基本上就無視。但是使用這個主語時，須注意delay因為每次多了讀取和比較的動作，花的時間會因此延長，須考量到誤差值。

**五、心得：**

Note:

1.對於實驗內容的心得感想

2.對本課程的建議。

這次的實驗課，必須利用之前都沒用過的方式，也就是在更新數字的圖中，為了能讓兩個以上的顯示器顯示正常，不可以逐項做運算，要用一個counter控制，先讓兩個以上運算好的值輪迴顯示，以達到同時顯示的方式，之後再做數據更新。這次的程式讓我更了解了主語的奧妙，感覺主語雖然運算非常不方便，但是卻也因此讓程式的設計上有了新的思維，讓我們在實驗之於，也能有所長進。我認為這次的實驗將對於以往都更加有挑戰和樂趣，非常期待下一次的實驗。

**六、程式碼與註解：**

基礎題

ORG 0000H //從0000H開始

JMP START //跳到START

ORG 0050H //從0050H開始

;-------------------------

START:

MOV R0,#01110000B //將R0存成01110000

MOV R1,#10110000B //將R1存成10110000

MOV R2,#11010110B //將R2存成11010110

MOV R3,#11101001B //將R3存成11101001

MOV R5,#00H //使得R5當作counter，並歸 0

MAIN1:

CJNE R5,#10,MAIN2 //如果R5不等於10 跳main2

MOV R5,#00H //使R5歸零，重新控制速度

INC R0 //R0++計算進位

MAIN2:

CJNE R0,#7AH,SHOW1 //如果R0不等於10 跳show1忽略計算進位

INC R1 //R1++

MOV R0,#70H //移動01110000到R0，視同R0歸零

SHOW1:

MOV P1,R0 //移動R0到P1

CALL DELAY //顯示延遲

CJNE R1,#0BAH,SHOW2 //如果R不等於10 跳show2

INC R2 //R2++計算進位

MOV R1,#0B0H //移動10110000到R1，視同R1歸零

SHOW2:

MOV P1,R1 //移動R1到P1

CALL DELAY //顯示延遲

CJNE R2,#0DAH,SHOW3 //如果R2不等於10 跳show3

INC R3 //R3++

MOV R2,#0D0H //移動11010000到R2，視同R2歸零

SHOW3:

MOV P1,R2 //移動R2到P1

CALL DELAY //顯示延遲

CJNE R3,#0EAH,SHOW4 //如果R3不等於10 跳show4

MOV R3,#0E0H //移動11100000到R3，視同R3歸零

SHOW4:

MOV P1,R3 //移動R3到P1

CALL DELAY //呼叫DELAY

INC R5 //R5++，讓counter+1

JMP MAIN1 //跳到MAIN1

DELAY:

MOV R6,#20

DELAY1:

MOV R7,#255

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

RET

END

進階題

ORG 0000H //從0000H開始

JMP START //跳到START

ORG 0050H //從0050H開始

START:

MOV R0,#01110000B //將01110000移動到R0

MOV R1,#10110000B //將10110000移動到R1

MOV R5,#00H //將R5當作counter並歸 0

MAIN1:

CJNE R5,#200,MAIN2 //如果R5不等於200跳main2控制整體速度

MOV R5,#00H //重複時會重置Counter

DEC R0 //個位數-1

MAIN2:

CJNE R0,#01101111B,SHOW1 //查看個位數若不為-1跳至SHOW1

DEC R1 //十位數-1，由於上面個位數不足扣除

MOV R0,#01111001B //個位數重新設為9

SHOW1:

MOV P1,R0 //顯示個位數

CALL DELAY //顯示延遲

SHOW2:

INC R5 //Counter加1

CJNE R1,#10101111B,SHOW3 //十位數若不為-1跳至SHOW3

MOV R1,#10110110B //重置十位數為6

MOV R0,#01110000B //重置個位數為0

SHOW3:

MOV P1,R1 //顯示十位數

CALL DELAY //顯示延遲

JMP MAIN1 //跳回MAIN1

DELAY:

MOV R6,#120

DELAY1:

MOV R7,#62

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

RET

END