**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 # 7

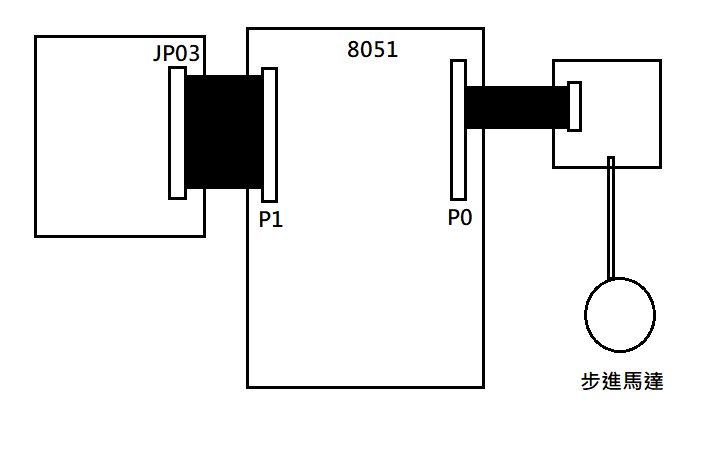
**ㄧ、實驗目的：**

但請以自己對本次實驗的認知來加以說明。

了解步進馬達的控制方式與規格，學習如何控制馬達速率。複習鍵盤的使用方式，熟悉如何選擇鍵盤上單一按鍵而非一排固定格式。Delay時間的熟悉與應用。

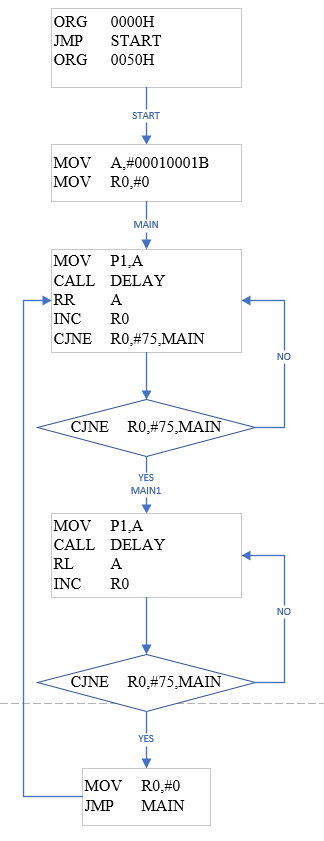
**二、硬體架構：**

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

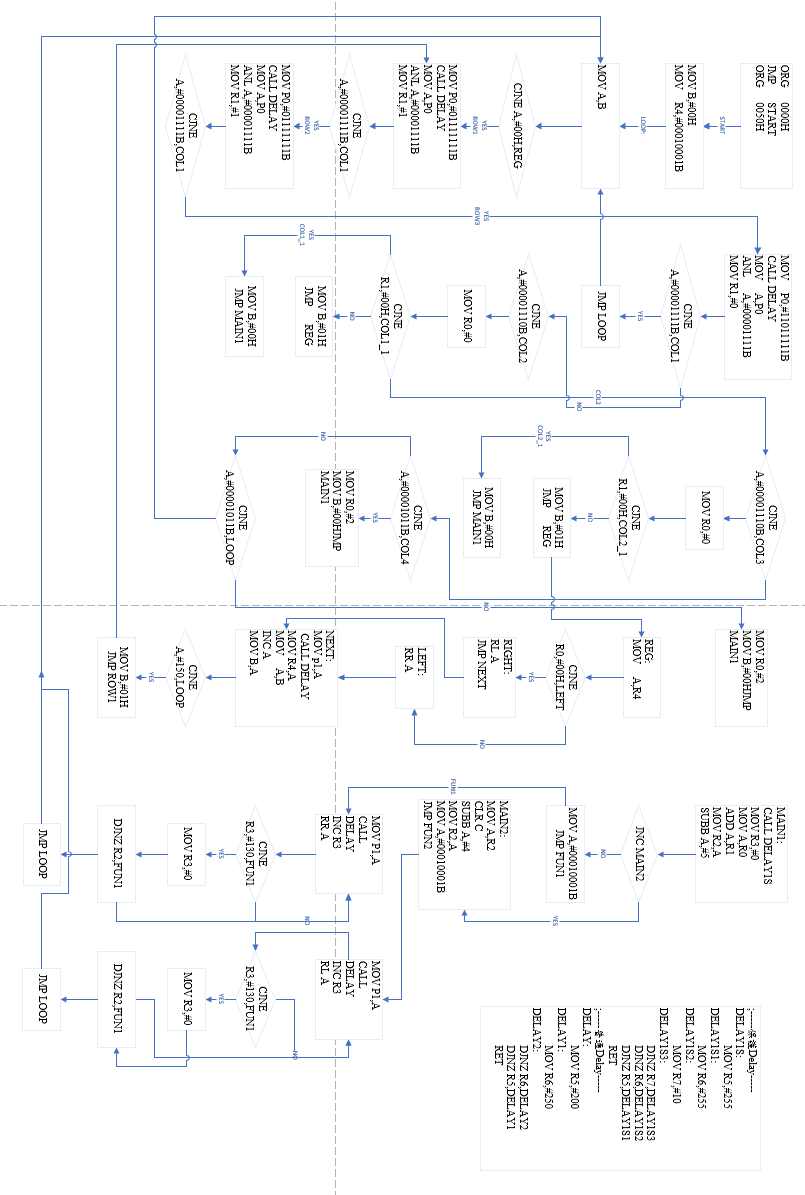


**三、程式流程圖：**

**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。

(1)當步進馬達被轉動時，單板上F112的四顆LED燈會亮起，分別對應A, A’, B, Bi，但是由於是在快速轉動，其delay時間間隔極短，所以看起來像是四顆全亮。

(2)若單板不接電LED燈在手動旋轉馬達的情況下，會造成磁場改變，進而產生電場，電場會產生電位差形成電流，讓單板上的LED燈亮起。

(3)如果沒有加delay的話，磁場的改變速度過快，只有一瞬間而已，馬達的機械在尚未旋轉前，磁場就已經改變到下一個方向，會造成馬達的旋轉無法跟上磁場，最後使得馬達停留而不動。

(4) 在實驗的時候，若是delay時間過長或過短都會造成旋轉不順，其delay的時間必須善加控制，若是delay時間過短，則只會造成馬達持續發燙而沒有動作。然而在讓馬達持續旋轉的測試中，若delay值過大，會造成馬達按鍵重複掃描的次數相對下降，但是若delay值過小，馬達則會重複呼叫delay次數過多，造成馬達走走停停。實驗的時候曾經遇過馬達只會往單一方向旋轉的問題，經檢查發現是其中一個方向少了delay，可見delay的重要性。

**五、心得：**

Note:

1.對於實驗內容的心得感想

2.對本課程的建議。

我覺得這幾次的實驗主要著重於delay的控制，雖然和之前的鍵盤實驗有點像，但是持續旋轉的功能又與按鍵的按下輸入不盡相同，讓實驗在複習舊的技巧時，也能學習新的知識。先前對於馬達的操控並不是很了解，了解的部分只是機械式固定轉速的馬達，經過本次實驗熟悉了馬達如何運用電腦運算來達到控制的效果，也比以往的實驗多了新鮮感。鍵盤在實驗時也有不同的看法，如何不看一整排或一整列，而是精細的操作鍵盤上的其中一個按鍵，這讓我在複習舊的技巧上有很大的幫助。

**六、程式碼與註解：**

**基礎題：**

ORG 0000H

JMP START

ORG 0050H

START:

MOV A,#00010001B //設定旋轉參數

MOV R0,#0 //將R0設為一個度數參考

MAIN:

MOV P1,A //輸出P1

CALL DELAY //Delay讓馬達可以旋轉

RR A //輪轉A讓參數方向不同

INC R0 //R0++

CJNE R0,#75,MAIN //R0到75則繼續執行

MOV R0,#0

MAIN1:

MOV P1,A //增大R0的值改變角度

CALL DELAY

RL A

INC R0

CJNE R0,#150,MAIN1

MOV R0,#0

JMP MAIN

DELAY:

MOV R6,#200

DELAY1:

MOV R7,#250

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

RET

END

**進階題：**

ORG 0000H

JMP START

ORG 0050H

START:

MOV B,#00H //初始化B作為持續旋轉的計數器

MOV R4,#00010001B //初始化R4當作紀錄持續選轉變數

LOOP:

MOV A,B

CJNE A,#00H,REG //查看B是不是0

ROW1:

MOV P0,#01111111B //查看ROW按鍵

CALL DELAY

MOV A,P0

ANL A,#00001111B

MOV R1,#1 //將R1設為1當作轉的角度的參考

CJNE A,#00001111B,COL1

ROW2:

MOV P0,#10111111B

CALL DELAY

MOV A,P0

ANL A,#00001111B

MOV R1,#5 //將R1設為5當作轉的角度的參考

CJNE A,#00001111B,COL1

ROW3:

MOV P0,#11011111B

CALL DELAY

MOV A,P0

ANL A,#00001111B

MOV R1,#0 //將R1設為0當作持續旋轉的訊號

CJNE A,#00001111B,COL1

JMP LOOP //沒有數字被按下

COL1:

CJNE A,#00001110B,COL2 //查看COL按鍵

MOV R0,#0

CJNE R1,#00H,COL1\_1 //查看要不要持續旋轉

MOV B,#01H //counter開始記數

JMP REG //跳到持續旋轉的函數

COL1\_1: //不用持續旋轉

MOV B,#00H //將counter歸零

JMP MAIN1 //跳到顯示

COL2:

CJNE A,#00001101B,COL3 //同上

MOV R0,#1

CJNE R1,#00H,COL2\_1

MOV B,#01H

JMP REG

COL2\_1:

MOV B,#00H

JMP MAIN1

COL3:

CJNE A,#00001011B,COL4

MOV R0,#2

MOV B,#00H //不可能有此按鍵排列，忽略偵測

JMP MAIN1

COL4:

CJNE A,#00000111B,LOOP //沒有數字被按下

MOV R0,#3

MOV B,#00H //不可能有此按鍵排列，忽略偵測

JMP MAIN1

REG:

MOV A,R4 //載入旋轉方向變數

CJNE R0,#00H,LEFT //查看旋轉方向

RIGHT:

RL A

JMP NEXT

LEFT:

RR A

NEXT:

MOV p1,A //顯示

CALL DELAY

MOV R4,A //更新R4供下次使用

MOV A,B //更新counter

INC A

MOV B,A

CJNE A,#150,LOOP //若重複150次旋轉偵測一次按鍵

MOV B,#01H //重置counter

JMP ROW1

MAIN1:

CALL DELAY1S //呼叫Delay保護按鍵

MOV R3,#0 //將R3設為0作為計數

MOV A,R0 //整理R0+R1

ADD A,R1

MOV R2,A

SUBB A,#5 //查看方向

JNC MAIN2

MOV A,#00010001B //初始化旋轉參數

JMP FUN1

MAIN2:

MOV A,R2

CLR C

SUBB A,#4

MOV R2,A

MOV A,#00010001B

JMP FUN2

FUN1:

MOV P1,A //旋轉

CALL DELAY //延遲讓機械跟上

INC R3 //R3計數++

RR A //更新旋轉參數

CJNE R3,#130,FUN1 //130轉

MOV R3,#0 //更新R3給下一層迴圈

DJNZ R2 ,FUN1 //依照得到的數大小來決定角度

JMP LOOP //回到迴圈

FUN2:

MOV P1,A

CALL DELAY

INC R3

RL A

CJNE R3,#130,FUN2

MOV R3,#0

DJNZ R2,FUN2

JMP LOOP

;-----保護Delay-----

DELAY1S:

MOV R5,#255

DELAY1S1:

MOV R6,#255

DELAY1S2:

MOV R7,#10

DELAY1S3:

DJNZ R7,DELAY1S3

DJNZ R6,DELAY1S2

DJNZ R5,DELAY1S1

RET

;-----普通Delay-----

DELAY:

MOV R5,#200

DELAY1:

MOV R6,#250

DELAY2:

DJNZ R6,DELAY2

DJNZ R5,DELAY1

RET

END