**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 #8

**ㄧ、實驗目的：**

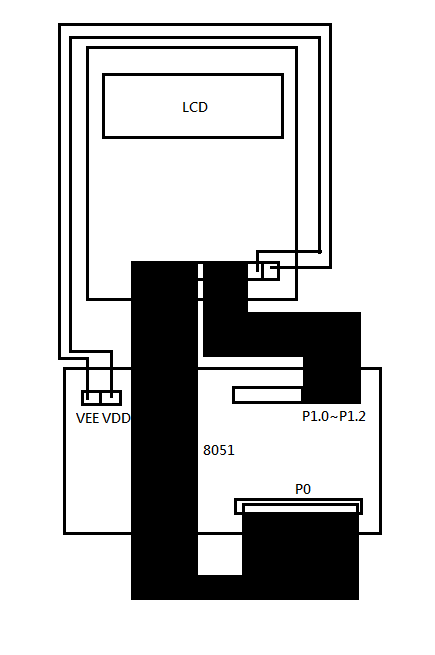
但請以自己對本次實驗的認知來加以說明。

1. 了解LCD的工作方式，與如何自行將字型寫入LCD。
2. Delay時間的運用與操作。

**二、硬體架構：**

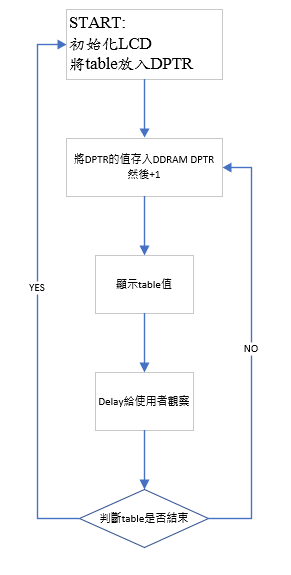
請儘量自己畫圖不管是手繪或者是以電腦繪圖(Word or PoworPoint)。

1. LCD實習版
2. 1bit接線條、2bit接線條、3bit接線條、8bit接線條
3. 8051 ICE Adapter

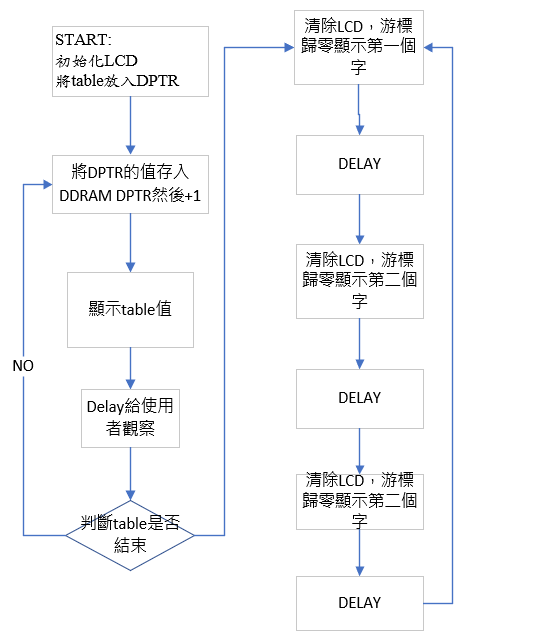


**三、程式流程圖：**

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)使用LCD時，LCD有需要不同的等待時間，然而又不能過長或過短導致程式閒置時間過長，延遲時需要等待輸入、等待LCD計算等等，才能繼續傳下一筆資料，LCD也需要處理時間，否則會在LCD處理完之前就將下一筆動作輸入，導致LCD同時接收太多資料而錯誤。

(2)AC是用來指定要存取的DD RAM和CG RAM的地址。當資料讀取之後，AC會自動+1或-1，游標會在輸入完一個字型後自動往後，另外LCD換行的原因可能是因為LCD沒有換行的指令，所以會自動換行。

(3)在寫程式的時候，一度以為LCD沒有正常顯示，所以浪費很多時間，但是經過拔除pin3後調整，發現顯示的文字是黑底白字，於是重新接上pin3，利用麵包版和1k可變電阻調整明暗，不過結果是反而只能將文字越條越暗導致顯示不清楚。推測可能是與LCD廠牌有關，或者LCD本身一開始就已經故障，導致顯示是黑底白字以至於不清楚。

**五、心得：**

Note:

1.對於實驗內容的心得感想

2.對本課程的建議。

這周的LCM實驗，使用前須要做各種設定與測試，並且在接線上感覺較為複雜，在傳送資料時，用的都是同一條傳輸線，必須利用其他接腳控制寫入，不是非常方便，但是確實也讓接線簡化不少，避免更複雜的接線出現。想必這是以後工程上常會見到的狀況，確實在實驗上得花上不少心力。除此之外，虛擬指令的使用，也是這次實驗的一個重點，如何正確的學會使用，也是讓我覺得在主語上有進步的地方。

**六、程式碼與註解：**

基礎題：

ORG 0000H

JMP START

ORG 0050H

START: ;事前設定

MOV DPTR,#TABLE ;載入表格

MOV R5,#40 ;R5紀錄delay

CALL DELAY

MOV A,#00111011B ;八位元存取、二列、5x7點矩陣

CALL COMMAND

MOV R5,#2

CALL DELAY

MOV A,#00001111B ;顯示器ON游標閃爍

CALL COMMAND

MOV R5,#2

CALL DELAY

MOV A,#00000001B ;清除螢幕

CALL COMMAND

MOV R5,#40

CALL DELAY

MOV A,#10000000B ;設定DD RAM位置為0

CALL COMMAND

MOV R5,#2

CALL DELAY

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

AGAIN: ;顯示

CLR A

MOVC A,@A+DPTR

INC DPTR

JZ AGAIN1

CALL SENDDATA

JMP AGAIN

AGAIN1:

JMP START ;顯示完回到start

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

COMMAND:

MOV P0,A

MOV P1,#00000100B ;E\_R/W\_RS 100 ->INSTRUCTION REGISTER

MOV P1,#00000000B ;E\_R/W\_RS 000 ->DISABLE

MOV R5,#2

CALL DELAY

RET

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

SENDDATA:

MOV P0,A

MOV P1,#00000101B ;E\_R/W\_RS 101 ->WRITE TO DD-RAM

MOV P1,#00000001B ;E\_R/W\_RS 001 ->DISABLE

MOV P1,#00000000B ;E\_R/W\_RS 000 ->DISABLE

MOV R5,#2

CALL DELAY

RET

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

DELAY:

MOV R6,#240

DELAY1:

MOV R7,#255

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

DJNZ R5,DELAY

RET

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

TABLE:

DB "0610837",0

END

進階題：

ORG 0000H

JMP START

ORG 0050H

START:

MOV R0,#10000000B

MOV R5,#0

CALL DELAY

CALL DELAY

MOV A,#00111011B ;初始設定，2列、5\*7、8位元存取

CALL COMMAND ;把A送到instruction register

MOV A,#00001110B ;Display ON,Cursor ON,Blink OFF

CALL COMMAND

MOV A,#00000110B ;設定寫資料進入LCD時, 顯示器不要動, 游標向右移一格

CALL COMMAND

MOV A,#00000001B ;清除全螢幕

CALL COMMAND

MOV R5,#0 ;延遲等清除LCD命令處理完畢

CALL DELAY

CALL DELAY

MOV A,#01000000B ;設定CG RAM位址

CALL COMMAND

CALL SETFONT

MOV A,#10000000B ;設定DD RAM位址=0

CALL COMMAND

LOOP:

MOV A,#00000000B ;顯示第一個字

CALL SENDDATA

MOV R5,#100

CALL DDELAY

MOV A,#00000001B ;清除LCD

CALL COMMAND

MOV R5,#10

CALL DDELAY

MOV A,R0 ;游標移到第一行開頭

CALL COMMAND

MOV A,#00000001B ;顯示第二個字

CALL SENDDATA

MOV R5,#100

CALL DDELAY

MOV A,#00000001B ;清除LCD

CALL COMMAND

MOV R5,#10

CALL DDELAY

MOV A,R0 ;游標移到第一行開頭

CALL COMMAND

MOV A,#00000010B ;顯示第三個字

CALL SENDDATA

MOV R5,#100

CALL DDELAY

MOV A,#00000001B ;清除LCD

CALL COMMAND

JMP LOOP

COMMAND:

MOV P1,#00000100B

MOV P0,A

MOV R7,#50 ;延遲等訊號穩定

DJNZ R7,$

MOV P1,#00000000B ;關掉溝通

MOV R7,#60 ;延遲等訊號處裡完畢

DJNZ R7,$

RET

SENDDATA:

MOV P1,#00000101B

MOV P0,A

MOV R7,#50 ;延遲等訊號穩定

DJNZ R7,$

CLR P1.2 ;讓Enable接腳產生負緣訊號

MOV R7,#250 ;延遲等訊號處裡完畢

DJNZ R7,$

RET

SETFONT: ;造字副程式

MOV DPTR,#FONT ;指標指到字型表開頭

MOV R6,#8\*3 ;一個字8筆資料, n個字共 8 x n 筆資料

NEXT:

CLR A ;清除累加器內資料

MOVC A,@A+DPTR ;讀取字型資料

CALL SENDDATA

INC DPTR ;指標內容加1, 指到下一行字型

DJNZ R6,NEXT

RET

FONT:

DB 00100B ;第一個字的5x7字型

DB 01010B

DB 00100B

DB 11111B

DB 10101B

DB 00100B

DB 01010B

DB 00000B

DB 00100B ;第二個字的5x7字型

DB 01010B

DB 10100B

DB 11111B

DB 00101B

DB 00100B

DB 01010B

DB 00000B

DB 00100B ;第三個字的5x7字型

DB 01010B

DB 00101B

DB 11111B

DB 10100B

DB 00100B

DB 01010B

DB 00000B

DDELAY: ;長延遲

MOV R6,#100

DDELAY1:

MOV R7,#100

DDELAY2:

DJNZ R7,DDELAY2

DJNZ R6,DDELAY1

DJNZ R5,DDELAY

RET

DELAY: ;短延遲

MOV R6,#2

DELAY1:

MOV R7,#5

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

DJNZ R5,DELAY

RET

END