**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 # 2

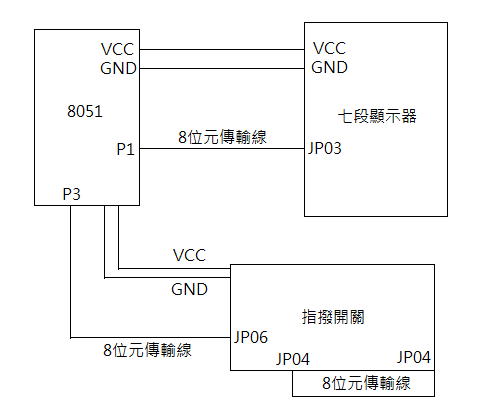
**ㄧ、實驗目的：**

但請以自己對本次實驗的認知來加以說明。

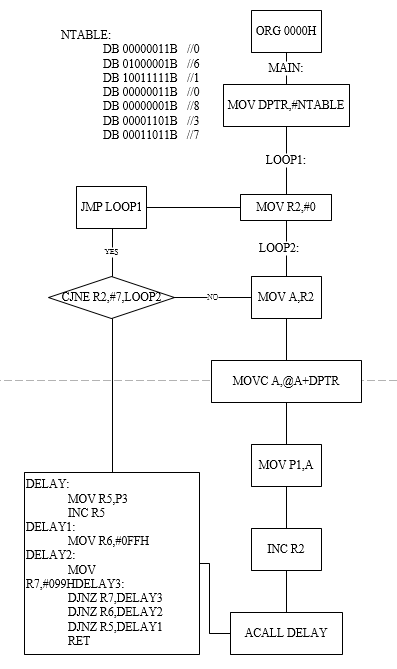
了解七段顯示器電路結構以及相關的控制方式。

**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：**



**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。

當重複次數大於Table所存取的個數的時候，會顯示亂碼，這時候需要檢查程式中是否有引導到正確位置，否則會讀取錯誤的值輸出。

(1)

DB 00010001B //A

DB 11000001B //b

DB 01100011B //C

DB 10000101B //D

DB 01100001B //E

DB 01110001B //F

(2)

注意讀取記憶體的位置，內部和外部記憶體會有不同的指令，內部讀取要注意記憶體讀取的大小不可以大於8 bit，內建記憶體大小不可以大於4k，由0000~0FFF。內部記憶體有分為兩區，分別是0000~007F、0080~00FF，存取時要注意。

**五、心得：**

Note:

1.對於實驗內容的心得感想

2.對本課程的建議。

我認為這次的實驗相對有趣，除了可以經由自己測試的結果調整程式的Table來達到想要的結果，還能銜接上次的DELAY迴圈，讓我在嘗試新指令的同時，也能達到複習的功效。進階題則是相同的部分，讓人感覺輕鬆許多。雖然是第二次的實驗，但是在打主語的時候仍然常常不小心打出C++的語法，這點需要多注意，導致編譯的時候常常會有ERROR，希望未來能更熟練的熟悉主語的寫作方式，更快更流暢的達成目標。

**六、程式碼與註解：**

**基礎題：**

ORG 0000H //從0000位置開始

JMP MAIN //跳到MAIN執行

ORG 0050H //從0050位置開始

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

MAIN:

MOV DPTR,#NTABLE //把NTABLE移動到DPTR儲存

LOOP1:

MOV R2,#0 //將R0歸零

LOOP2:

MOV A,R2 //將R2移動到A

MOVC A,@A+DPTR //將A加上DPTR(將A+目前的R2移動到A)

MOV P1,A //移動A到P1給七段顯示器讀取

INC R2 //R2++

ACALL DELAY //CALL DELAY的迴圈

CJNE R2,#10,LOOP2 //重複LOOP2共10次

JMP LOOP1 //回到LOOP1歸零R2

NTABLE:

DB 00000011B //0

DB 10011111B //1

DB 00100101B //2

DB 00001101B //3

DB 10011001B //4

DB 01001001B //5

DB 01000001B //6

DB 00011011B //7

DB 00000001B //8

DB 00001001B //9

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

DELAY:

MOV R5,#0FFH //設定R5=255

DELAY1:

MOV R6,#0FFH //設定R6=255

DELAY2:

MOV R7,#0FH //設定R7=15

DELAY3:

DJNZ R7,DELAY3 //將R7減1後回到DELAY3，0則繼續執行下行

DJNZ R6,DELAY2 //將R6減1後回到DELAY2，0則繼續執行下行

DJNZ R5,DELAY1 //將R5減1後回到DELAY1，0則繼續執行下行

RET //retrun

END //結束

**進階題：**

ORG 0000H //從0000位置開始

JMP MAIN //跳到MAIN執行

ORG 0050H //從0050位置開始

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

MAIN:

MOV DPTR,#NTABLE //把NTABLE移動到DPTR儲存

LOOP1:

MOV R2,#0 //將R0歸零

LOOP2:

MOV A,R2 //將R2移動到A

MOVC A,@A+DPTR //將A加上DPTR(將A+目前的R2移動到A)

MOV P1,A //移動A到P1給七段顯示器讀取

INC R2 //R2++

ACALL DELAY //CALL DELAY的迴圈

CJNE R2,#7,LOOP2 //重複LOOP2共7次

JMP LOOP1 //回到LOOP1歸零R2

NTABLE:

DB 00000011B //0

DB 01000001B //6

DB 10011111B //1

DB 00000011B //0

DB 00000001B //8

DB 00001101B //3

DB 00011011B //7

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

DELAY:

MOV R5,P3 //讀取P3到R5

INC R5 //R5++

DELAY1:

MOV R6,#0FFH //設定R6=255

DELAY2:

MOV R7,#099H //設定R7=153

DELAY3:

DJNZ R7,DELAY3 //將R7減1後回到DELAY3，0則繼續執行下行

DJNZ R6,DELAY2 //將R6減1後回到DELAY2，0則繼續執行下行

DJNZ R5,DELAY1 //將R5減1後回到DELAY1，0則繼續執行下行

RET //return

END //結束