**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 # 10

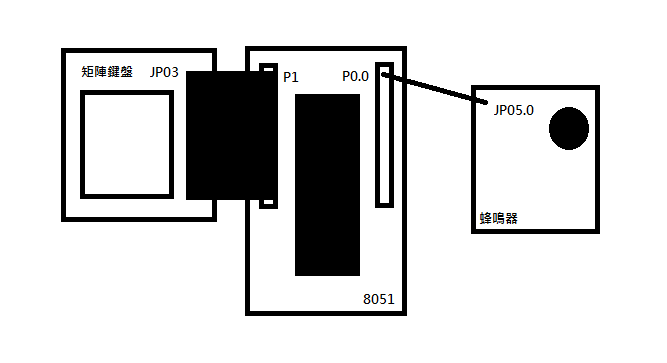
**ㄧ、實驗目的：**

但請以自己對本次實驗的認知來加以說明。

了解Timer/ Counter工作的原理以及控制的過程和結果。嘗試使用蜂鳴器，學習使用的方式和工作原理。Delay時間在Timer的運用。

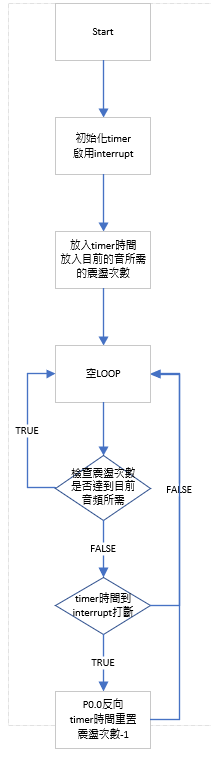
**二、硬體架構：**

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

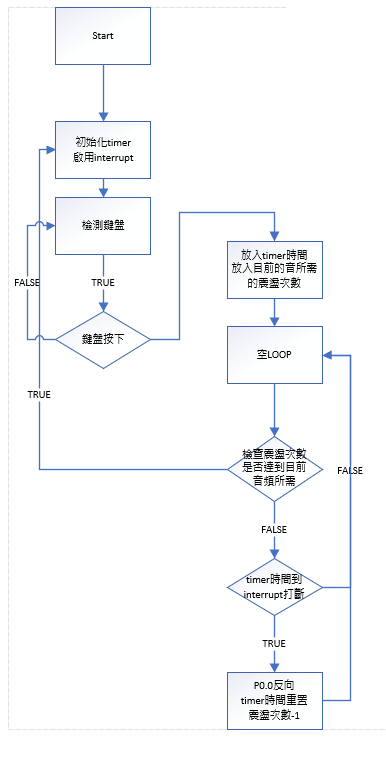


**三、程式流程圖：**

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。

只要控制蜂鳴器的震盪次數就可以，當震盪次數增多時，其音的延遲也會跟著變長。但是在使用的過程中，當震動次數>256，便超過8051可以儲存的範圍，因此將小於256的最大震盪次數設為長音，再以除2的方式當作短音。還有音如果中間沒有設定delay，便會導致每個音連在一起，難以辨別樂曲，若要辨別樂曲則必須加入適當的delay，讓使用者可以清楚地聽到節奏。

**五、心得：**

Note:

1.對於實驗內容的心得感想

2.對本課程的建議。

我覺得這次的程式，主要難的點是如何使用Timer當作delay，這是新的式來控制delay，感覺只是一個8051但是delay的方式卻有許多種，可見delay在電腦控制上的重要性，不僅是要精準的掌控，還必須顧及效率和多工。這幾次的程式下來，雖然8051不像C++一樣有很多地方需要歸零，但是8051卻也有需要初始化設定的地方，一開始覺得8051的機械語和C++差異甚大，不過現在看來也是有很多相似處。

**六、程式碼與註解：**

基本題：

ORG 0000H

AJMP MAIN

ORG 0BH

AJMP TIMER

ORG 0050H

MAIN:

MOV P0,#0H

MOV TMOD,#00000001B ;設定timer mode1

MOV TH0,#11110001B ;給timer register第一個音

MOV TL0,#00011000B

SETB ET0 ;啟用timer0 interrupt

CLR TF0 ;重置interrupt flag

SETB EA ;啟用interrupt

SETB TR0 ;設定edge trigger

START:

MOV R3,#0H ;暫存用

MOV R4,#13 ;一共七個音

SOUND:

MOV DPTR,#TABLE

MOV A,R3

MOVC A,@A+DPTR

MOV R0,A ;放入high 8bit

INC R3

MOV A,R3

MOVC A,@A+DPTR

MOV R1,A ;放入low 8bit

INC R3

MOV A,R3

MOVC A,@A+DPTR ;放入震盪次數

MOV R2,A

INC R3

ACALL LOOP ;進入空loop

DJNZ R4,SOUND

CALL DELAY

JMP START

LOOP:

CJNE R2,#0H,LOOP ;interrupt R2 次

RET

TIMER:

CPL P0.0 ;P0.0震盪

MOV TH0,R0 ;重新放入timer時間

MOV TL0,R1

DEC R2 ;震盪一次 R2--

RETI

TABLE: ;timer時間 high 8bit;timer時間 low 8bit;震盪次數

DB 11110110B,00001001B,196;SOL

DB 11110100B,00101010B,165;MI

DB 11110100B,00101010B,82;MI

DB 11110100B,11010111B,175;FA

DB 11110010B,10110111B,147;RE

DB 11110010B,10110111B,73;RE

DB 11110001B,00011000B,131;DO

DB 11110010B,10110111B,147;RE

DB 11110100B,00101010B,165;MI

DB 11110100B,11010111B,175;FA

DB 11110110B,00001001B,196;SOL

DB 11110110B,00001001B,196;SOL

DB 11110110B,00001001B,98;SOL

DELAY:

MOV R6,#255

DELAY1:

MOV R7,#255

DELAY2:

DJNZ R7,DELAY2

DJNZ R6,DELAY1

RET

END

進階題：

ORG 0000H

AJMP MAIN

ORG 0BH

AJMP TIMER

ORG 0050H

MAIN:

MOV P0,#0H

MOV TMOD,#00000001B ;設定timer mode1

MOV TH0,#0F1H ;給timer register第一個音

MOV TL0,#18H

SETB ET0 ;啟用timer0 interrupt

CLR TF0 ;重置interrupt flag

SETB EA ;啟用interrupt

SETB TR0;

START:

ROW1:

MOV P1,#7FH ;掃描第一個row(S01~S04)

CALL DELAY

MOV A,P1

ANL A,#0FH ;較高的4Bit清為0

MOV R5,#0

CJNE A,#0FH,COL1 ;看(s01~S04)有沒有按下鍵 有則跳至col1

ROW2: ;沒有則往下一列掃描

MOV P1,#0BFH

CALL DELAY

MOV A,P1

ANL A,#0FH

MOV R5,#12

CJNE A,#0FH,COL1 ;看(s05~S08)有沒有按下鍵 有則跳至col1

ROW3: ;沒有則往下一列掃描

MOV P1,#0DFH

CALL DELAY

MOV A,P1

ANL A,#0FH

MOV R5,#24

CJNE A,#0FH,COL1 ;看(s09~S12)有沒有按下鍵 有則跳至col1

ROW4: ;沒有則往下一列掃描

MOV P1,#0EFH

CALL DELAY

MOV A,P1

ANL A,#0FH

MOV R5,#36

CJNE A,#0FH,COL1 ;看(s13~S16)有沒有按下鍵 有則跳至col1

JMP ROW1 ;沒有跳回 則ROW1重新掃描

COL1:

CJNE A,#0EH,COL2 ;IF A<>0EH,JMP COL2

MOV R4,#0 ;A=0EH,R0=0

JMP SHOW

COL2:

CJNE A,#0DH,COL3 ;IF A<>0DH, JMP COL3

MOV R4,#3 ;A=0DH,R0=1

JMP SHOW

COL3:

CJNE A,#0BH,COL4 ;IF A<>0BH ,JMP COL4

MOV R4,#6 ;A=0BH ,R0=2

JMP SHOW

COL4:

CJNE A,#07H,ROW1 ;if a<>07H,重新掃描

MOV R4,#9 ;A=07H 則R0=3

SHOW:

SETB TR0

MOV A,R5

ADD A,R4

MOV R3,A ;儲存鍵盤按到第幾個音

SOUND:

MOV DPTR,#TABLE

MOV A,R3

MOVC A,@A+DPTR

MOV R0,A ;放入high 8bit

INC R3

MOV A,R3

MOVC A,@A+DPTR

MOV R1,A ;放入low 8bit

INC R3

MOV A,R3

MOVC A,@A+DPTR

MOV R2,A ;放入震盪次數

INC R3

ACALL LOOP ;進入空loop

CLR TR0 ;結束interrupt 重新偵測鍵盤

JMP START

LOOP:

CJNE R2,#0H,LOOP ;interrupt R2 次

RET

TIMER:

CPL P0.0 ;P0.0震盪

MOV TH0,R0 ;重新放入timer時間s

MOV TL0,R1

DEC R2 ;震盪一次 R2--

RETI

TABLE: ;timer時間 high 8bit;timer時間 low 8bit;震盪次數

DB 0F1H,018H,131

DB 0F2H,070H,144

DB 0F2H,0B7H,147

DB 0F3H,0DFH,161

DB 0F4H,02AH,165

DB 0F4H,0D7H,175

DB 0F5H,72H,185

DB 0F6H,09H,196

DB 0F6H,9DH,208

DB 0F7H,20H,220

DB 0F7H,9FH,233

DB 0F8H,18H,247

DELAY:

MOV R6,#80

DELAY2:

MOV R7,#100

DELAY3:

DJNZ R7,DELAY3

DJNZ R6,DELAY2

RET

END