



# 微算機實驗報告

姓名：洪巧芸

系級：資工 114

# Lab #11

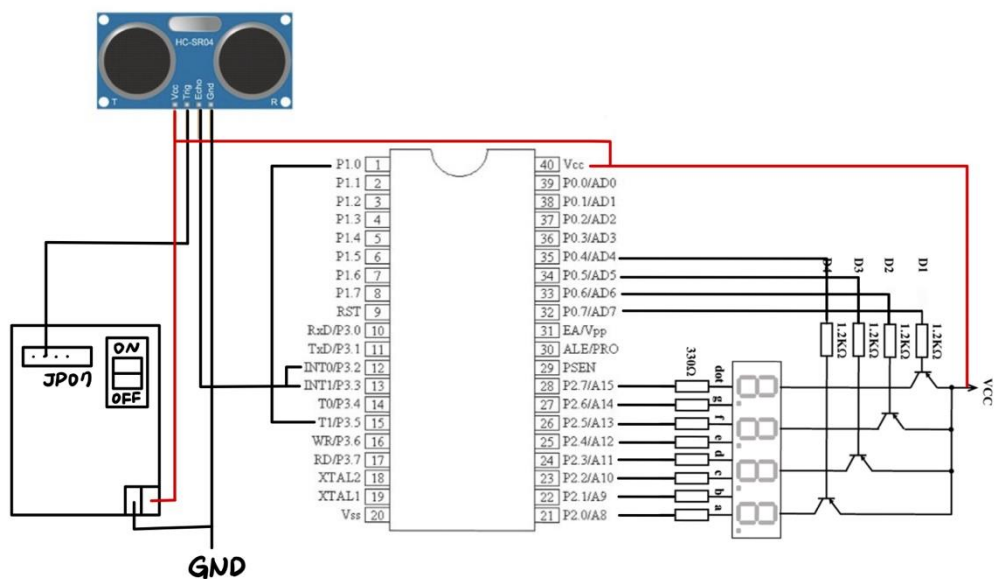
**學號：110550143**

**上課時間：2023/12/19**

### 一、實驗目的：

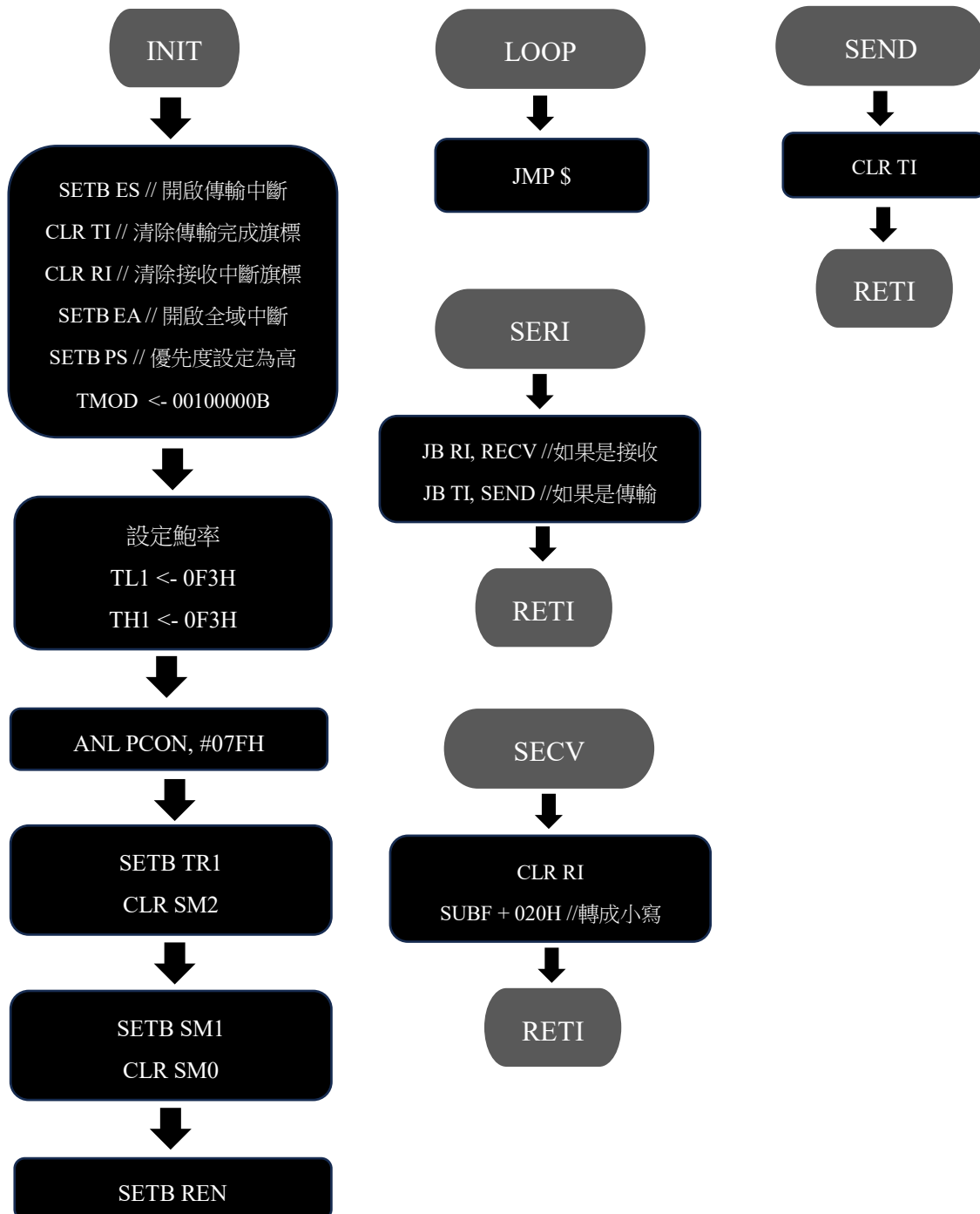
了解蜂鳴器產生八度音(八音度)的工作原理與控制方法，使得 8051 得以與其他裝置傳送接收訊息，同時認識 UART 串列通訊之使用，並以串列通訊的方式讓電腦控制 MCS-51 開發板，使其控制蜂鳴器發出不同頻率的聲音。

## 二、硬體架構：

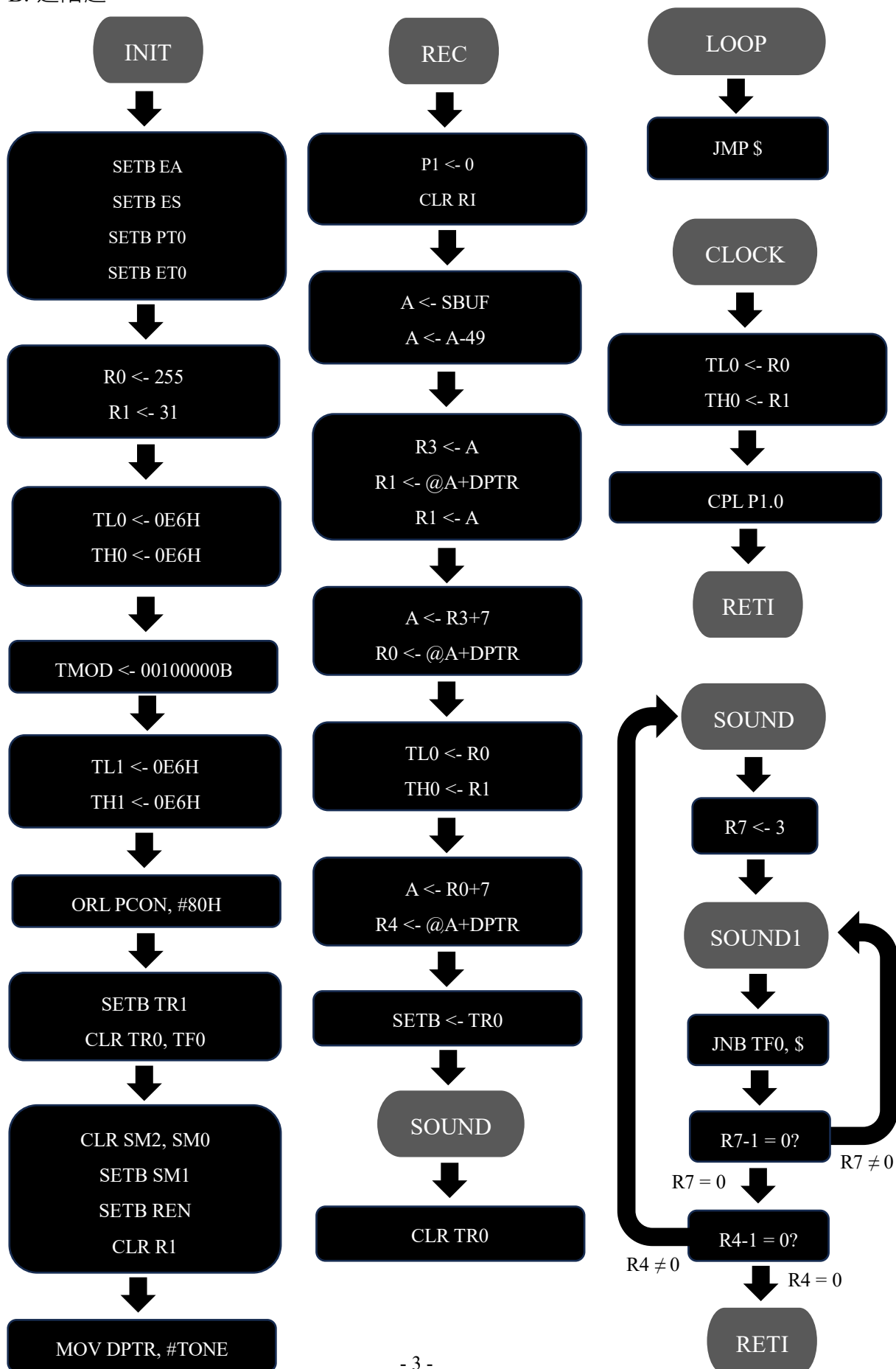


### 三、程式流程圖：

#### A. 基礎題



## B. 進階題



#### 四、問題與討論：

(1) 試根據你所設計的程式架構，解說如何實現蜂鳴器的長短音功能。

負責輸出聲音的部分在 SOUND 副程序中，而我在輸出時固定在輸出時都以 R7=3 開始，因此需要改變的變數為 R4，又 R4 是從 TONE (Table)的第三組數據 (7 個 7 個一組, i.e, HZ/4 這組) 所以只要改變這組數據就可以改變蜂鳴器輸出音的長短。

(2) 假設要更改音色，試問要如何更改？

改變波形：因為是這次的實驗是由 P1.0 腳位反轉實現簡單的方波進而輸出聲音，因此波形的改變可能有限。我們可以藉由先輸入到 RLC 電路之類的系統再輸進喇叭，藉此改變波形輸出不同的音色。此外，如過今天要改變的是輸出音高我們可以藉由調整 TIMER0 的頻率來改變 (ex. TONE 副程式中的 TH0, TL0、TIMER0 自身的初始化數據)。

(3) 在串列傳輸中，設定 Baud Rate 的目的為何？有那些因素會造成 Baud Rate 的誤差？這又會造成傳輸的資料發生什麼樣的問題？

- i. 設定 Baud Rate 的目的是為了定義數據傳輸的速率，讓接收端知道甚麼時候應該要採樣，同時確保兩者在相同的速率下發送和接收數據，讓通信線路的利用率最大化。
- ii. TIMER0 設定的誤差、振盪器的精度、溫度和電源變化（環境溫度變化和電源電壓波動可能影響電路的性能）、信號衰減（在長距離傳

輸中，信號可能會衰減 )、傳輸介質的質量 ( 通信線路的質量，如線纜的類型和長度 )。

- iii. 數據錯誤 ( 波特率的偏差可能導致數據位的解讀錯誤，因為接收方可能會在錯誤的時間讀取位值 )、幀同步失效 ( 在幀同步的通信協議中，波特率的偏差可能導致同步失效，從而導致數據幀的錯誤解析 )、通信效率下降 ( 由於需要重傳錯誤的數據，波特率誤差可能導致整體通信效率下降 )

## 五、程式碼與註解：

### A. 基本題

```
1  ORG    0000H
2  LJMP   INIT
3  ORG    0023H // 傳輸中斷
4  LJMP   SERI
5  ORG    0050H
6
7  INIT:
8      SETB    ES // 開啟傳輸中斷
9      CLR    TI // 清除傳輸完成旗標
10     CLR    RI // 清除接收中斷旗標
11
12     SETB    EA // 開啟全域中斷
13     SETB    PS // 優先度設定為高
14     MOV    TMOD,#00100000B // 設定 TIMER1 為 AUTORELOAD 模式
15
16     // 設定速率為 2400
17     MOV    TL1,#0F3H
18     MOV    TH1,#0F3H
19     ANL    PCON,#07FH // 設定 SMOD=1
20
21     SETB    TR1 // 啟動 Timer1 開始計時
22     CLR    SM2 // SERIAL PORT MODE 1 (傳輸模式為 MD1)
23
24     SETB    SM1 // 設定 SERIAL PORT MODE 1
25     CLR    SM0
26     SETB    REN // 啟用自動接收
27
28 // 死循環
29 LOOP:
30     JMP    $
31
32 SERI:
33     JB     RI,RECV // 若接收中斷觸發
34     JB     TI,SEND // 若傳輸中斷觸發
35     RETI
36
37 RECV:
38     CLR    RI // 清除接收中斷旗標
39     MOV    A,SBUF // 讀取接收的資料
40     ADD    A,#020H // 轉換為小寫字元
41     MOV    SBUF,A // 送出資料
42     RETI // 回傳中斷
43
44 SEND:
45     CLR    TI // 清除傳輸中斷旗標
46     RETI // 回傳中斷
47
48     END
```

## B. 進階題

```
1  ORG 0000H
2  AJMP INIT
3  ORG 000BH          //IMER0 中斷
4  AJMP Clock
5  ORG 0023H          //串口接收中斷
6  AJMP REC
7  ORG 0050H
8
9  INIT:
10     SETB EA  //全域中斷
11     SETB ES  //串口中斷
12     SETB PT0  //優先度設定
13     SETB ET0  //開啟TIMER0中斷
14
15     MOV R0,#255
16     MOV R1,#31
17     MOV TL0, #0E6H  //TIMER0 低位初始值
18     MOV TH0, #0E6H  //TIMER0 高位初始值
19
20     /*TIMER1 -> AUTORELOAD
21     |  TIMER0 -> Mode 0*/
22     MOV TMOD,#00100000B
23
24     //鮑率 2400
25     MOV TL1, #0E6H
26     MOV TH1, #0E6H
27     ORL PCON, #80H  //SMOD=1
28
29     //Timer設定
30     SETB TR1 //啟動 Timer1 開始計時
31     CLR TR0 //Timer0 不計時
32     CLR TF0 //清除 Timer0 中斷旗標
33
34     //Port設定
35     CLR SM2 //串口傳輸模式 Mode 1
36     CLR SM0
37     SETB SM1 //SERIAL PORT MODE1
38     SETB REN //啟動自動接收
39     CLR RI  //清除接收中斷旗標
40
41     MOV DPTR,#TONE //設定音符表的指針
42
43  REC:
44     //清除輸出, 接收中斷旗標
45     MOV P1,#0
46     CLR RI
47
48     MOV A,SBUF //讀取串口接收的數據
49     SUBB A,#49 //ASCII -> NUM
```

```

51      //Table存的時候是7個一組 有3組 (HIGH, LOW, 音色)
52      // HIGH
53      MOV R3,A      //備份數字(一開始串口接收的數據
54      MOVC A,@A+DPTR
55      MOV R1,A      //複製到R1
56
57      //LOW
58      MOV A,R3      //恢復備份(一開始串口接收的數據
59      ADD A,#7      //+7換成下一組 (LOW)
60      MOVC A,@A+DPTR
61      MOV R0,A      //複製到R0
62
63      //設定TH0和TL0的值
64      MOV TL0,R0
65      MOV TH0,R1
66
67      //音色
68      //計算播放的次數
69      MOV A,R0      //恢復備份
70      ADD A,#7      //+7換成下一組 (音色)
71      MOVC A,@A+DPTR //從音符表中取執行次數的值
72      MOV R4,A      //複製到R4
73
74      SETB TR0      //啟動TIMER0開始運行
75      ACALL SOUND   //播放音符
76      CLR TR0      //停止TIMER0

```

```

78      //無限循環
79      LOOP:
80      |      SJMP $
81
82      //MODE0計時中斷處理程序
83      Clock:
84      |      MOV TL0,R0 //設定TIMER0的低位值
85      |      MOV TH0,R1 //設定TIMER0的高位值
86      |      CPL P1.0   //反向輸出方波信號
87
88      |      RETI       //返回中斷
89
90      SOUND:
91      |      MOV R7,#3
92      SOUND1:
93      |      JNB TF0,$   //等待Timer0中斷
94      |      DJNZ R7,SOUND1
95      |      DJNZ R4,SOUND
96      |      RETI       //返回中斷
97
98      TONE:
99      |      //TH0
100     |      DB 226,229,232,233,236,238,240
101     |      //TL0
102     |      DB 4,12,9,20,2,8,6
103     |      //HZ/4 -> 音色
104     |      DB 131,147,165,175,196,220,247
105
106     END

```



## 六、心得：

### 1. 對於上課內容的心得感想：

這週的課程因為發燒所以沒有到課，因此錯過了老師對於程式所要使用到的基本知識講解，導致在後面實作時遇到了許多困難。

### 2. 對於實驗內容的心得感想：

這次的實驗相比之前的我覺得有很大程度的提升，除了增加了新的「鮑率」概念，輸出也從原本的在微算機連接端口上改變影響其他連接的模組，變成與電腦鍵盤配合。從電腦鍵盤輸入，再在電腦上輸出或是從其他模組輸出，有點類似再拿 FPGA 板寫 Verilog 的感覺。