



微算機實驗報告

Lab # 12

姓名：仇健安

系級：電機系

學號：111511239

上課時間：2025/05/27

一、實驗目的：

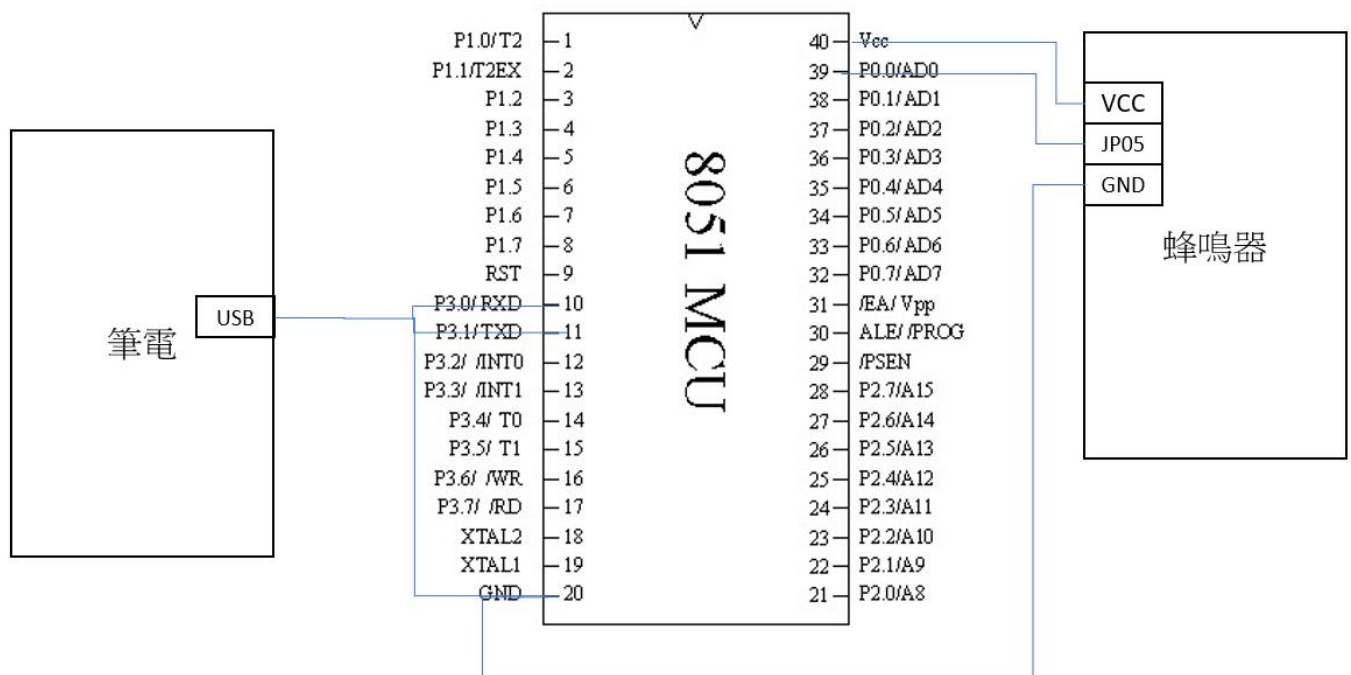
本實驗旨在學習 8051 串列通訊 (UART) 的基本應用，透過設定 Timer1 產生 Baud Rate，實現電腦與單晶片之間的資料傳輸。

內容分為兩部分：

1. 接收電腦傳入的大寫英文字母，轉換為小寫後回傳，熟悉串列輸入與輸出流程。
2. 接收數字字元，以 Timer0 控制 P0.0 產生對應頻率的方波，驅動蜂鳴器發出 Do~Si 音階，實現輸入控制音頻輸出。

本實驗結合串列通訊與音頻控制，強化對 Timer 與輸出裝置整合應用的理解。

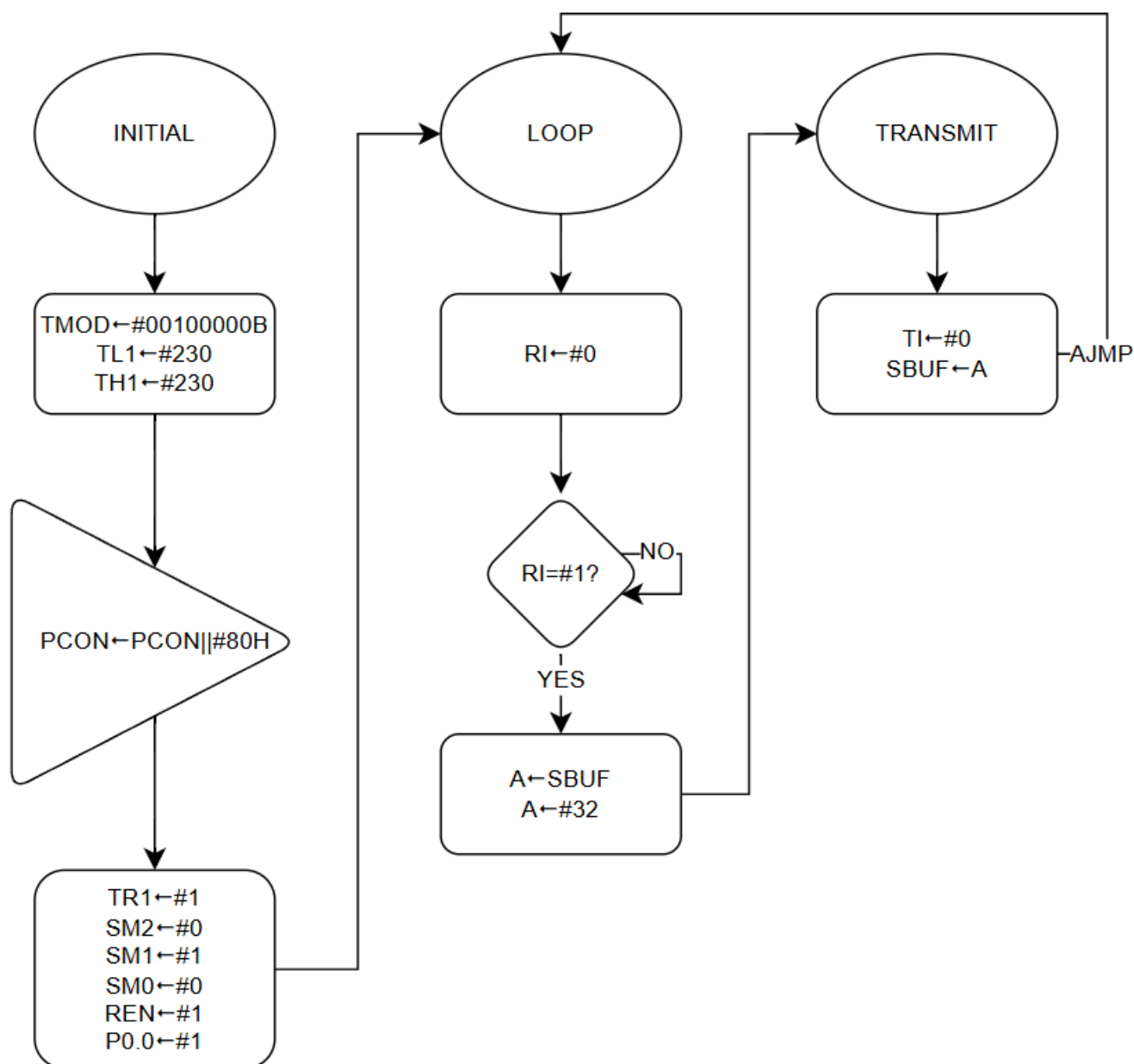
二、硬體架構：



功能	元件／模組	8051 接腳 / 接口	備註說明
串列通訊接收	USB-to-Serial 線	TX(綠) → P3.0 (RXD)	電腦傳送資料 → 單晶片接收
串列通訊傳送	USB-to-Serial 線	RX(白) → P3.1 (TXD)	單晶片回傳資料 → 電腦顯示
串列通訊接地	USB-to-Serial 線	GND ↔ GND	必須共地，否則無法穩定通訊
蜂鳴器輸出控制	被動蜂鳴器模組	P0.0	由程式透過 CPL 產生方波控制發聲
蜂鳴器電源正極	蜂鳴器模組	+5V	可接開發板 VCC(若蜂鳴器為有源也可用)
蜂鳴器電源接地	蜂鳴器模組	GND	接單晶片或模組共地

三、程式流程圖：

基本題：



INITIAL：初始化設定

設定 Timer1 為 Mode 2，自動重載產生 2400 bps 的 Baud Rate，並啟用串列埠 Mode 1 (8-bit UART)，準備好接收資料。

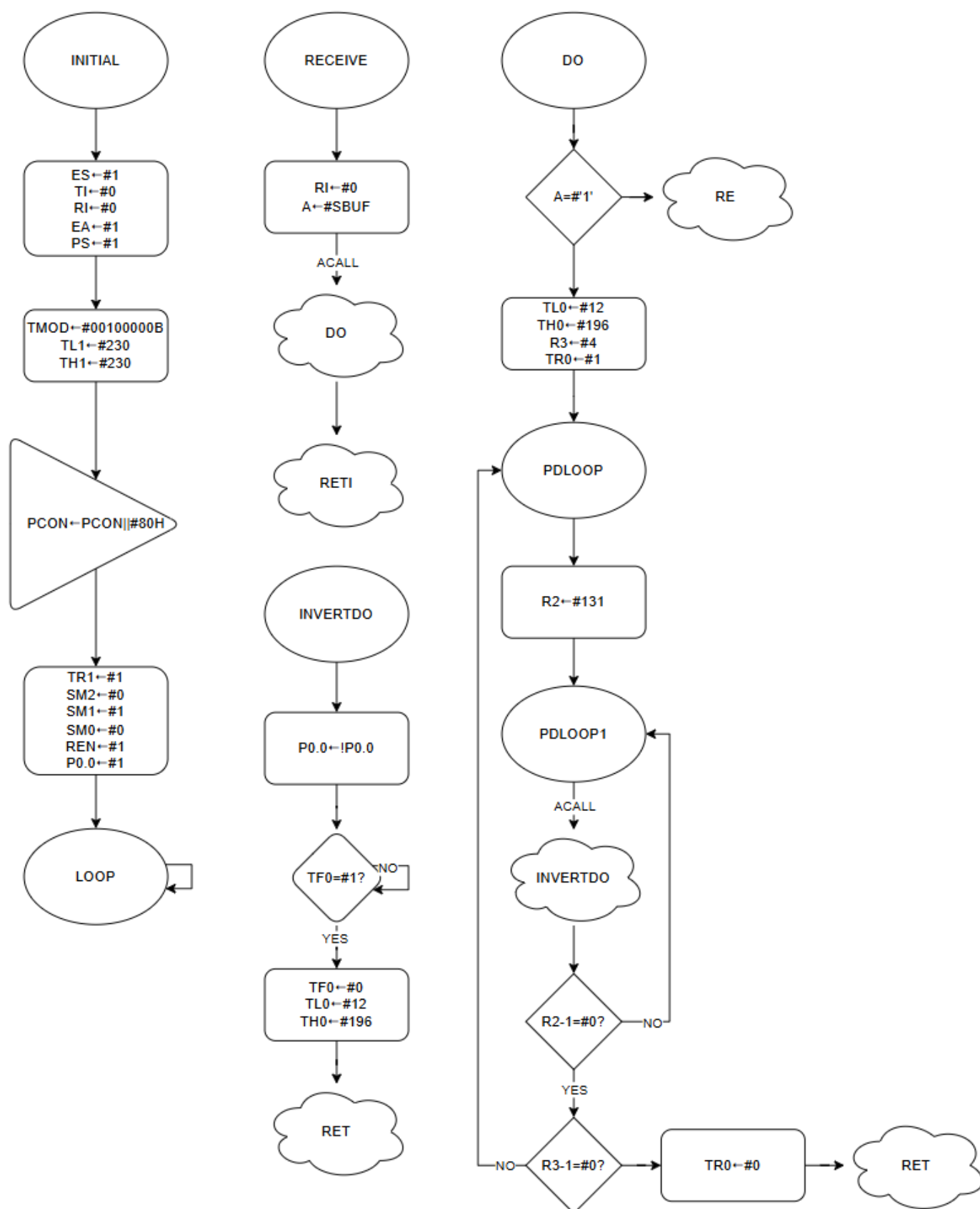
LOOP：接收並轉換資料

等待串列資料接收完成 (RI=1)，將收到的大寫字元從 SBUF 取出，加 32 轉換為小寫字元，準備回傳。

TRANSMIT：傳送小寫資料

清除傳送旗標，將轉換後的小寫字元寫入 SBUF，送出至電腦端，然後回到主迴圈等待下一筆輸入。

進階題:



INITIAL (初始化):

設定 Timer1 為 Mode 2，自動重載產生 2400 bps 串列通訊速率，並設定串列埠為 Mode 1 (8-bit UART)。同時開啟中斷與接收功能，進入主迴圈等待資料傳入。

RECEIVE (接收處理):

當 RI=1 表示接收到資料後，將 SBUF 中的字元讀入累加器 A，再呼叫 DO 子程式判斷並處理輸入內容。

DO (音符判斷與播放):

判斷收到的字元是否為 '1'，若是則設定 Timer0 對應 DO 音所需的 TH0、TL0 以及音長控制用的 R3，透過方波反轉控制蜂鳴器發聲。

RE、MI、FA、SO、LA、SI 的處理邏輯皆與 DO 相同，差異在於：

判斷條件為 A=#'2'~#'7'

對應的 Timer0 初值 (TL0/TH0) 不同，對應不同音高頻率

播放長度 (R3) 也根據方波週期不同略作調整

每個音符播放時，皆透過內外兩層迴圈控制 P0.0 的 CPL 翻轉次數，產生穩定的對應音階。

四、問題與討論：

(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？

1. Baud Rate 是指每秒傳輸的位元數，設定 Baud Rate 的目的是為了讓 傳送端與接收端以相同的速度進行通訊。若雙方 Baud Rate 不一致，資料就會無法正確對齊，造成誤判或漏讀。

2. 誤差的因素

- Timer 設定誤差：如 TH1/TL1 設定值不準確。
- SMOD 設定錯誤：若未開啟 SMOD=1，Baud Rate 無法加倍。
- 主時脈頻率誤差：單晶片的時鐘來源若有偏差，會導致計時不準。
- 傳送端與接收端設定不一致：兩端 Baud Rate 若未一致，會出現同步問題。

3. 可能造成的問題

- 資料位元對不準：造成字元錯誤、亂碼顯示。
- 接收漏位或溢位：接收緩衝區判斷錯誤，導致資料遺失。
- 無法通訊：誤差嚴重下會完全無法建立有效資料交換

五、程式碼與註解：

基本題：

ORG 0000H

AJMP INITIAL ; 開機後跳轉到 INITIAL 區段執行初始化

ORG 0050H ; 主程式碼從位址 0050H 開始放置

=====

; 初始化程式區段

=====

INITIAL:

MOV TMOD, #00100000B ; 設定 Timer1 為 Mode 2 (8-bit 自動重載), Timer0 無使用

MOV TL1, #230 ; Timer1 初值設定為 230 (TL1)

MOV TH1, #230 ; 自動重載值也設為 230, 搭配 SMOD=1 可產生 2400 bps

ORL PCON, #80H ; 設定 PCON.7 = SMOD = 1, 將 Baud Rate 加倍

SETB TR1 ; 啟動 Timer1, 使其開始計數, 提供串列通訊時基

CLR SM2 ; 關閉多機通訊功能 (僅在特殊多主通訊下使用)

SETB SM1 ; SM1 = 1, 與 SM0 組合為 Mode 1 (8-bit UART 可變速)

CLR SM0 ; 串列通訊設定為 Mode 1

SETB REN ; REN = 1, 啟用接收功能 (允許接收資料)

SETB P0.0 ; 預設蜂鳴器控制腳 P0.0 為高電位 (無聲), 可忽略

=====

; 主執行迴圈

=====

LOOP:

CLR RI ; 清除接收完成旗標 RI (Receive Interrupt)

JNB RI, \$; 等待 RI=1, 也就是收到一筆資料前停在這行

MOV A, SBUF ; 將接收到的資料從 SBUF 讀到暫存器 A

ADD A, #32 ; 將 ASCII 大寫轉小寫 (大寫字元 + 32 = 小寫字元)

=====

; 傳送轉換後的資料

=====

TRANSMIT:

CLR TI ; 清除傳送完成旗標 TI (Transmit Interrupt)

MOV SBUF, A ; 將處理後的小寫字元寫入 SBUF 傳送出去

AJMP LOOP ; 跳回主迴圈, 繼續等待下一筆輸入

END ; 程式結束

進階題:

```
;-----  
; 8051 蜂鳴器音階播放程式 (使用串列接收數字 1~7 播 Do~Si)  
;-----
```

```
ORG 0000H  
AJMP INITIAL          ; 程式進入點 → 跳到初始化區段 INITIAL
```

```
ORG 0023H  
AJMP RECEIVE          ; 串列中斷向量地址 (Interrupt 4) → 接收到資料跳至 RECEIVE
```

```
ORG 0050H              ; 主程式碼起始位址 (避免和中斷向量區衝突)
```

```
;-----  
; INITIAL: 初始化區段  
;-----
```

INITIAL:

```
SETB ES                ; ES = 1, 啟用串列中斷 (Serial Interrupt Enable)  
CLR TI                 ; 清除傳送中斷旗標 (防止進入不必要的中斷)  
CLR RI                 ; 清除接收中斷旗標  
SETB EA                ; EA = 1, 啟用總中斷 (Global Interrupt Enable)  
SETB PS                ; 設定串列中斷為高優先權 (Priority Serial)
```

```
; 設定 Timer1 為 Mode2 (8-bit 自動重載), 作為串列通訊用的 Baud Rate 時基  
MOV TMOD, #00100000B ; T1 = Mode2, T0 保留
```

```
; Timer1 計數初值與重載值 (230 = 0xE6), 搭配 SMOD=1 可達 2400 bps  
MOV TL1, #230  
MOV TH1, #230
```

```
ORL PCON, #80H         ; 設定 PCON.7 (SMOD) = 1 → Baud Rate 加倍  
SETB TR1               ; 啟動 Timer1 計數器 (供串列通訊時基使用)
```

```
; 設定串列通訊模式為 Mode 1: 8-bit UART (可變速率)  
CLR SM2                ; 關閉多機通訊模式  
SETB SM1                ; SM1 = 1  
CLR SM0                ; SM0 = 0 → Mode 1  
SETB REN                ; REN = 1, 啟用接收功能 (Receive Enable)
```

```
SETB P0.0              ; 預設蜂鳴器輸出腳 P0.0 為高電位 (靜音)
```

```

LOOP:
    JMP $                ; 空迴圈等待中斷（程式停在此等待接收中斷）

;-----
;RECEIVE：串列接收中斷服務程式
;-----
RECEIVE:
    CLR RI                ; 清除接收中斷旗標（RI）
    MOV A, SBUF            ; 將接收到的字元從 SBUF 複製到暫存器 A
    ACALL DO                ; 呼叫 DO 程式根據 A 的內容播放對應音階
    RETI                    ; 中斷結束，返回主程式或等待下一次中斷

;-----
;DO：接收 '1' 判斷並播放 Do 音（523Hz）
;-----
DO:
    CJNE A, #'1', RE        ; 若 A ≠ '1'，跳到 RE 檢查是否是 '2'
    MOV TL0, #12            ; 設定 Timer0 初值對應 Do 頻率（TL0）
    MOV TH0, #196          ; 設定 Timer0 初值對應 Do 頻率（TH0）

    MOV R3, #4              ; 外層迴圈次數（音符播放時長控制）
    SETB TR0                ; 啟動 Timer0

    DELAYDO:
        MOV R2, #131        ; 內層迴圈次數（決定單一方波持續次數）
    DELAYDO1:
        ACALL INVERTDO        ; 呼叫產生方波函式
        DJNZ R2, DELAYDO1      ; 內圈迴圈倒數
        DJNZ R3, DELAYDO        ; 外圈迴圈倒數（總音長控制）
        CLR TR0                ; 播放結束，關閉 Timer0
        RET

    INVERTDO:
        CPL P0.0              ; 翻轉蜂鳴器輸出腳 P0.0（產生方波）
        JNB TF0, $            ; 等待 Timer0 溢位（直到 TF0=1）
        CLR TF0                ; 清除溢位旗標
        MOV TL0, #12          ; 重設 Timer0 計數初值
        MOV TH0, #196
        RET

;-----

```

; RE : 播放 Re 音 (587Hz)

;-----

RE:

CJNE A, #'2', MI

MOV TL0, #28

MOV TH0, #202

MOV R3, #3

SETB TR0

DELAYRE:

MOV R2, #196

DELAYRE1:

ACALL INVERTRE

DJNZ R2, DELAYRE1

DJNZ R3, DELAYRE

CLR TR0

RET

INVERTRE:

CPL P0.0

JNB TF0, \$

CLR TF0

MOV TL0, #28

MOV TH0, #202

RET

;-----

; MI : 播放 Mi 音 (659Hz)

;-----

MI:

CJNE A, #'3', FA

MOV TL0, #21

MOV TH0, #208

MOV R3, #3

SETB TR0

DELAYMI:

MOV R2, #220

DELAYMI1:

ACALL INVERTMI

DJNZ R2, DELAYMI1


```
DJNZ R3, DELAYMI
CLR TR0
RET
```

INVERTMI:

```
CPL P0.0
JNB TF0, $
CLR TF0
MOV TL0, #21
MOV TH0, #208
RET
```

```
;-----
; FA : 播放 Fa 音 (698Hz)
;-----
```

FA:

```
CJNE A, #'4', SO
MOV TL0, #8
MOV TH0, #211
MOV R3, #3
SETB TR0
```

DELAYFA:

```
MOV R2, #233
```

DELAYFA1:

```
ACALL INVERTFA
DJNZ R2, DELAYFA1
DJNZ R3, DELAYFA
CLR TR0
RET
```

INVERTFA:

```
CPL P0.0
JNB TF0, $
CLR TF0
MOV TL0, #8
MOV TH0, #211
RET
```

```
;-----
; SO : 播放 So 音 (784Hz)
```

;-----

SO:

```
CJNE A, #'5', LA
MOV TL0, #5
MOV TH0, #216
MOV R3, #4
SETB TR0
```

DELAYSO:

```
MOV R2, #196
```

DELAYSO1:

```
ACALL INVERTSO
DJNZ R2, DELAYSO1
DJNZ R3, DELAYSO
CLR TR0
RET
```

INVERTSO:

```
CPL P0.0
JNB TF0, $
CLR TF0
MOV TL0, #5
MOV TH0, #216
RET
```

;-----

; LA : 播放 La 音 (880Hz)

;-----

LA:

```
CJNE A, #'6', SI
MOV TL0, #16
MOV TH0, #220
MOV R3, #4
SETB TR0
```

DELAYLA:

```
MOV R2, #220
```

DELAYLA1:

```
ACALL INVERTLA
DJNZ R2, DELAYLA1
DJNZ R3, DELAYLA
CLR TR0
```

RET

INVERTLA:

CPL P0.0

JNB TF0, \$

CLR TF0

MOV TL0, #16

MOV TH0, #220

RET

;-----

; SI：播放 Si 音（988Hz）

;-----

SI:

CJNE A, #'7', RESET

MOV TL0, #12

MOV TH0, #224

MOV R3, #4

SETB TR0

DELAISI:

MOV R2, #247

DELAISI1:

ACALL INVERTSI

DJNZ R2, DELAISI1

DJNZ R3, DELAISI

CLR TR0

RET

INVERTSI:

CPL P0.0

JNB TF0, \$

CLR TF0

MOV TL0, #12

MOV TH0, #224

RET

;-----

; RESET：未輸入 1~7 時不動作

;-----

RESET:

RET

；接收到非 1~7 的字元時，直接結束無動作

END

；程式結束

六、心得：

上課心得:

這次課程對串列通訊 (Serial Port) 的講解，使我對 8051 的 UART 傳輸模式有了更深入的理解，尤其是在學習 Baud Rate 設定、Timer1 與 SMOD 的配合下，能夠準確產生穩定的傳輸速率，這對於雙向通訊的正確性至關重要。

老師也強調了串列通訊的資料格式 (起始位元、資料位元、停止位元) 與錯誤同步的情況，幫助我建立了完整的串列通訊觀念，原本我只知道 SBUF 是傳輸暫存器，現在能夠實際掌握 RI、TI 等旗標的應用與判斷條件，是很有收穫的地方。

實驗心得:

在實作上，我運用 AccessPort 傳送字元到 8051，並透過中斷接收處理，成功實作了「大寫轉小寫回傳」與「傳送數字控制蜂鳴器發聲」兩個功能，尤其是在播放音階的部分，我結合之前學過的 Timer0 控制方波技巧，讓 Serial Port 成為輸入控制來源，實現了軟體與硬體整合應用。從接收字元、轉譯、播放音符這一系列流程中，我實際體會到通訊控制與外部裝置互動的全過程。

這次實驗不只是單純學會 UART 傳輸，更讓我理解到中斷機制與 Timer 配合的重要性。

Notes:

1. 內容字體大小為 12，間距為單行間距
2. 中文字字體為標楷體
3. 英文字和阿拉伯數字為 Times New Roman
4. 嚴禁抄襲，抄襲者以 0 分計算
5. 請於報告左上角附上照片
6. 每次實驗課繳交上次實驗結報