

第七組微算機系統文件

作品名稱：復古機台音樂遊戲

作者：聶怡安、黃芊、黃律瑛、陳培恩

一、系統功能與原理說明：

我們設計的作品為音樂遊戲機台，當使用者按下開機鍵、系統會隨機對應到一首歌，同時馬達指針轉到對應的難度，接著使用者根據 LED 燈的位置按下對應琴鍵，蜂鳴器發出對應音符，隨著 LED 燈的變化使用者就能彈奏出完整曲目。

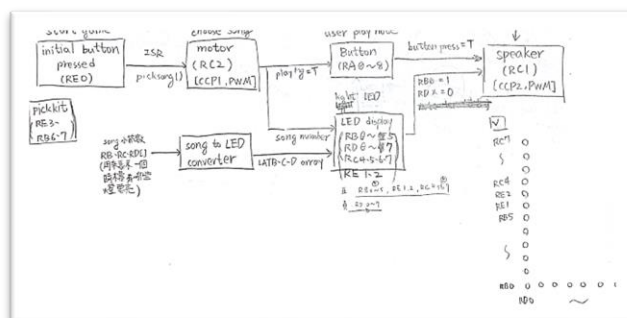
使用到 interrupt、CCP 模組、programed I/O 等多項原理，詳見第三點。

二、系統使用環境及對象：

使用 MPLAB xc8 編譯環境、pic18f4520 pickit4，使用對象無特殊限制，所有喜歡音樂、對音遊有興趣的人都可以體驗。

三、系統完整架構圖、流程圖、電路設計

程式一開始會進行 CCP 相關 register、燈泡輸出相關腳位的設定、interrupt 相關 register 設定 (控制 Timer)，接著進入 while 迴圈等待使用者按下開機鍵 (使用 polled I/O)，使用者按下開機鍵後隨即進入 picksong() 函數進行隨機選歌，選完歌之後會根據選到的歌進行音頻、節奏設定 (reinitCCP1()函數、Timer1_Initial()函數) 因為每一首歌的音域與不同，會影響蜂鳴器和 LED 燈的輸出。設定完成後，程式等在一個 while 迴圈，等待 timer interrupt 發生，interrupt 發生後進入 ISR 改變燈號，並呼叫 sing()函數判斷是否需要發出聲音，在 sing() 函數中，若該排燈號有亮且使用者按下琴鍵，蜂鳴器就會發出聲音，同時使用 getnote()函數取得每一首的音符資訊，進而更新 LED 燈泡亮的圖案，待歌曲完全結束會等在一個 while 迴圈直到使用者再次按下開機鍵進行下一輪遊戲。



四、實際組員之分工項目

聶怡安：負責蜂鳴器音頻的設計與設定

黃芊：負責按鈕的電路、程式設計

黃律瑛：負責 LED 燈的電路、程式設計

陳培恩：負責馬達電路、程式設計

此為初步分工，後來我們遇到問題大部分都是大家一起討論、解決，因為我們是一天一 bug

五、遇到的困難及如何解決

1、led 所需的 port 太多

--> 改用點矩陣，但太小 --> 用 led 燈自製點矩陣

2、Interrupt 數量不夠偵測 8 個 button，若接在同一個 port 會出問題

--> 改用 polled I/O 來偵測 8 顆 button

3、Speaker 和 motor 都要用 PWM，但不能共用 CCP

--> speaker 改用 CCP1、motor 改用 CCP2

4、按下開始後，馬達有時不會轉

--> 先做完 pick song 再將 playing = 1

5、led 無法直接用點矩陣對應的 port 來呈現我們要的亮法(有些多餘的燈會同步亮)

--> 用視覺暫留的方式去呈現

6、speaker 在發出聲音的間隔之間會有雜訊

--> 在每次 ISR 時將 PWM 方波變超小，使 speaker 安靜下來

7、speaker 在發出聲音的同時，馬達會跟著轉動(理論上要停止)

--> 將馬達使用到的 CCP2 關掉

8、pic18 記憶體不足(RAM 只有 1536 bytes, 但我們一首歌就有 4000 多 bytes)

--> 原本的設定是每個燈亮的圖(每個瞬間)都要有 48 個值，後來改成用 function 去偵測音符並直接自行換算成 LATX 的 port，如此一來就只需要紀錄一首歌的音符及節拍即可，共壓縮了 48 倍的長度

9、led 在掃描的過程中，有時不該出現的燈被亮起

--> 多加一行空白值使 led 在掃的時候不會因為掃太快而影響到其他的燈號

10、在製作外殼時，led 和 button 的距離太遠，杜邦線無法負荷

--> 去買更長的杜邦線 XD

11、馬達在選歌結束的最後會自動轉到一個神奇的地方

--> 將之後 speaker 跟馬達共用的 PR2 值固定，使馬達不會因為 PR2 改變而影響角度