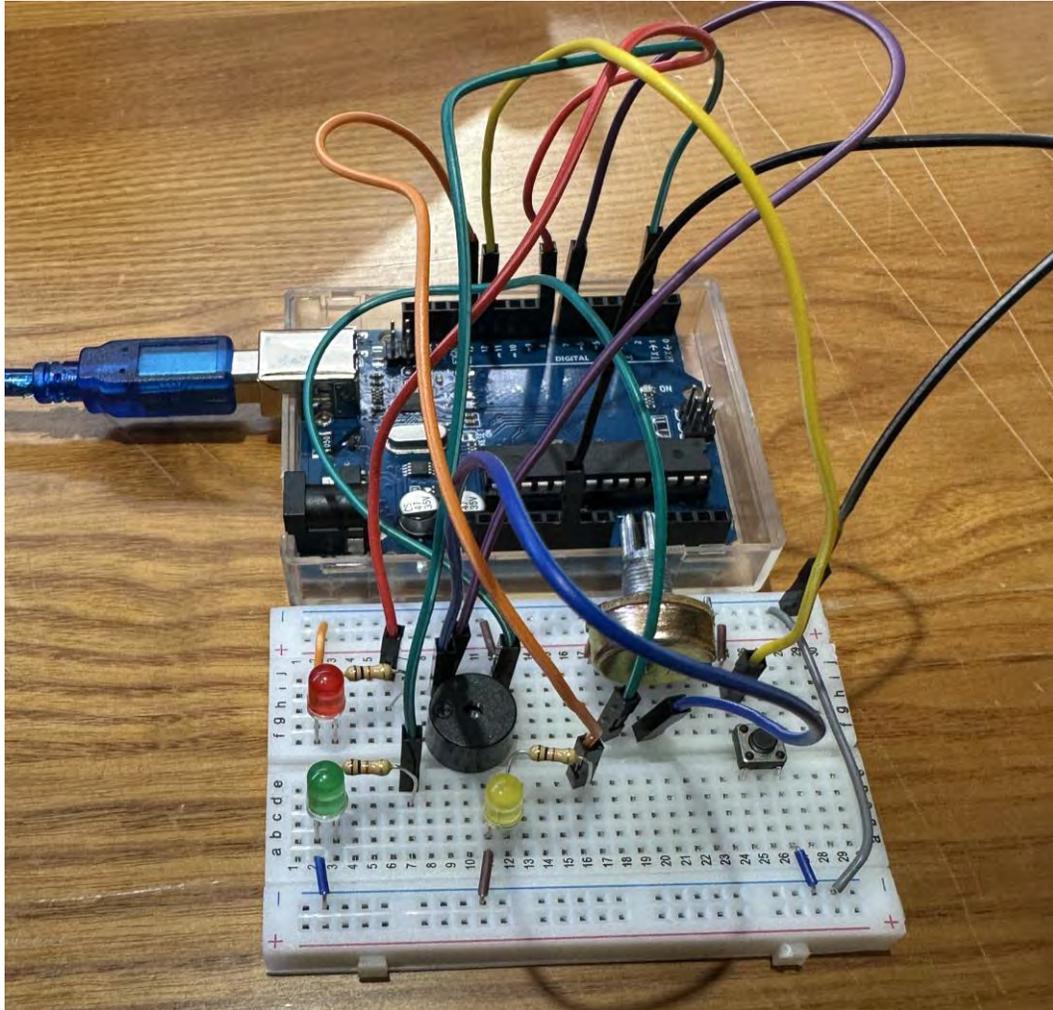


智慧居家監控實習



作品名稱: 音樂播放器

控制二乙 05 黃郁善

目錄

壹、摘要	4
貳、研究動機	4
參、研究方法	5
一、硬體裝置	5
二、相關軟體	11
三、軟體研究過程	12
肆、研究結果	17
一、硬體結構	17
二、成果展示	17
陸、結論	18
一、心得.....	18
二、未來改進方向.....	19

三、總結.....19

捌、參考資料 20

【音樂播放器】

壹、摘要

利用 OneButton 程式庫，透過單一按鈕開關控制蜂鳴器播放音樂，並以 LED 顯示播放狀態。實現待機時音樂停止且綠燈常亮、單擊按鈕開始或停止音樂播放、長按按鈕回到待機狀態，以及音樂播放時紅燈閃爍、暫停時黃燈閃爍的功能。使用 Arduino UNO R3 完成實體電路接線，達成預期目標。

貳、研究動機

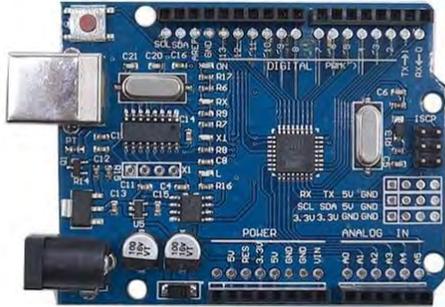
在嵌入式系統開發中，常需要透過簡單的硬體介面實現多樣的控制功能,且按鈕作為最基本的輸入設備，被廣泛應用於各種控制系統中。 。本次實習的動機在於學習如何利用 OneButton 程式庫，透過單一按鈕實現多種功能，並結合蜂鳴器與 LED 指示燈，提升系統與使用者的互動性。

參、研究方法

(一)硬體裝置

(1.) Arduino UNO R3

Arduino UNO R3 是一款易用型開源 (Open Source) 控制器，使用 ATmega16U2 單片機，與 Arduino IDE 完全兼容，可簡單地與感測器及各式各樣的電子元件連接，為學習 Arduino 編程最入門的版本。在音樂播放器系統中, Arduino UNO R3 扮演著關鍵的中樞角色。負責處理所有輸入與輸出設備的互動，確保系統能夠正確響應按鈕指令並執行對應功能。



Arduino UNO R3

(2.)電磁式有源蜂鳴器

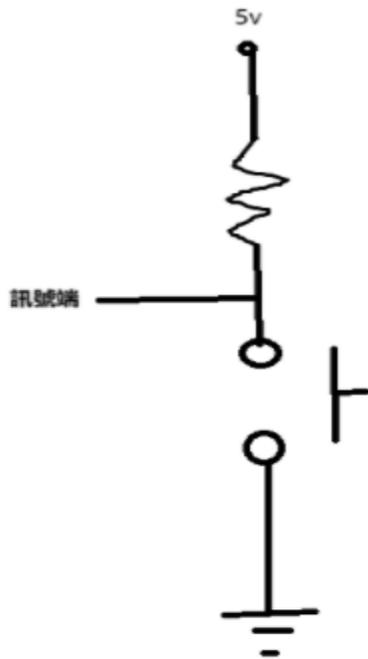
電磁式有源蜂鳴器內部已含驅動線路，只要提供直流電壓，正極與負極不要接錯，蜂鳴器就會發出聲響，透過 `tone()` 函數產生不同頻率的聲音訊號，用於音樂播放。具有體積較小、重量輕、音質清晰透徹、耐溫度高、等優點。



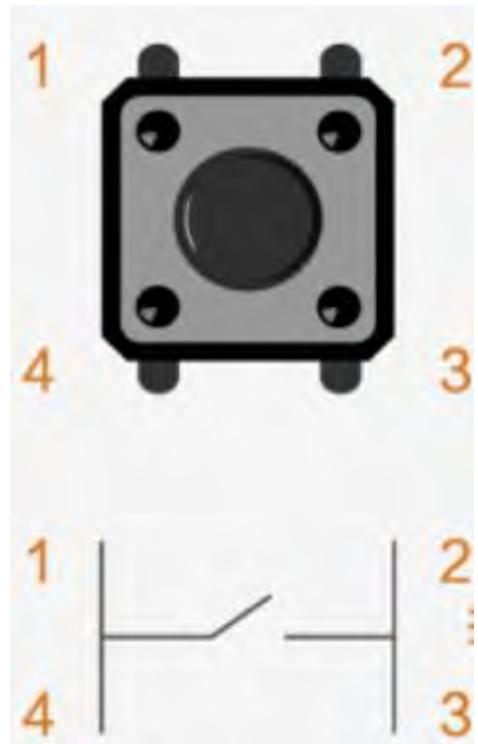
電磁式有源蜂鳴器

(3.)按鈕開關

提供使用者輸入控制, 可用於開關 LED、啟動蜂鳴器、切換模式等功能。



低態動作的按鈕開關接線時 使用對角才能達到按鈕的功能



(4.) 發光二極體

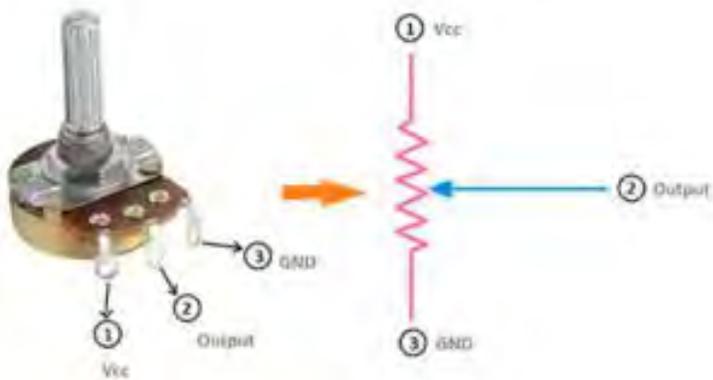
作為視覺指示燈，顯示當前系統狀態,使蜂鳴器的狀態可視化。



紅黃綠發光二極體

(5.)可變電阻

將可變電阻與蜂鳴器並聯,控制蜂鳴器上的電流,進而控制蜂鳴器的音量大小。



將蜂鳴器正負端分別接在相鄰的兩隻腳,以調整電阻大小。

(6.)LED 限流電阻

保護 LED ,避免過大電流損壞元件,通常使用 $220\Omega\sim 1K\Omega$ 確保 LED 在安全電流範圍內運行 。



1KΩ電阻

(二)相關軟體



Arduino IDE2.3.4

Arduino IDE 是一個簡潔且功能強大的程式設計環境,支援多種操作系統 (Windows、macOS 和 Linux)。它使用簡化的 C/C++ 語言,讓使用者能輕鬆撰寫、編譯和上傳程式到開發板,大大的降低了一般人開發的門檻。

(三)軟體研究過程

•功能說明

- 1.開機後,進到待機狀態,音樂停止播放、綠燈恆亮。
- 2.每按一下按鈕開關,可開始或暫停播放音樂。
- 3.長按按鈕開關,回到待機狀態,音樂停止播放、綠燈恆亮。
- 4.當音樂正在播放時,紅燈會閃爍;當音樂暫停播放時,紅燈停止閃爍、黃燈閃爍;當音樂結束播放,則回到待機狀態。

(1.) 定義音符頻率

```
buzzer11.ino
1 #define B3 247
2 // 中音區 (C4 - B4) (包含標準 C4)
3 #define C4 262
4 #define C#4 277 // C#4 / D#4
5 #define D4 294
6 #define D#4 311 // D#4 / E#4
7 #define E4 330
8 #define F4 349
9 #define F#4 370 // F#4 / G#4
10 #define G4 392
11 #define G#4 415 // G#4 / A#4
12 #define A4 440
13 #define A#4 466 // A#4 / B#4
14 #define B4 494
15
16 // 高音區 (C5 - B5)
17 #define C5 523
18 #define C#5 554 // C#5 / D#5
19 #define D5 587
20 #define D#5 622 // D#5 / E#5
21 #define E5 659
22 #define F5 698
23 #define F#5 740 // F#5 / G#5
24 #define G5 784
25 #define G#5 831 // G#5 / A#5
26 #define A5 880
27 #define A#5 932 // A#5 / B#5
28 #define B5 988
29 #define rest 0 // 休止符
```

(2.) 音符序列和音符播放時間

```
31 int melody[] = { G4, F#4, E4, F#4, E4, E4, E5, rest, G#4, F#4, rest, E4, rest, E5, rest, G#4, F#4, E4, rest, F#4, E4, E4, F#4, E4, E4, G#4, F#4, rest, C#5, C#5, C5, C#5, D#4, G#4, C#5, D#5, //1
32 E5, G#4, F#4, rest, E5, rest, F#5, F#5, E5, F#5, G#5, F#5, E5, E5, rest, G#5, F#5, E5, F#5, E5, B5, rest, C5, C#5, rest, rest, //2
33 C#4, F#4, G#4, C5, G4, F#4, E4, E5, C#5, E4, F#5, C5, G#4, C#5, B4, G#4, rest, F#4, rest, F#4, rest, G#4, rest, C#5, C#5, C5, B4, G#4, rest, G#5, //3
34 B4, C#5, B4, G#4, G4, F#4, E4, E5, C#5, E4, F#5, C5, B4, C#5, B4, G#4, rest, A4, rest, A4, rest, G#4, rest, C#4, E4, E5, E4, F#4, G#4, //4
35 C#5, C#4, B3, C#4, rest, C#4, rest, B3, C#4, D#4, E4, D#4, C#4, B3, rest, C#4, C#4, C#4, C#4, D#4, E4, F#4, E4, F#4, E4, G#4, rest, //5
36 C#5, C#4, B3, C#4, rest, C#4, rest, B3, C#4, D#4, E4, D#4, C#4, B3, rest, C#4, C#4, C#4, C#4, D#4, E4, F#4, F#4, E4, F#4, E4, G#4, rest, G#4, //6
37 G4, G#4, G4, G#4, F#4, C#4, rest, G#4, G4, G#4, G4, G#4, F#4, C#4, rest, F#4, F#4, F#4, E4, F#4, E4, F#4, E4, F#4, E4, F#4, B4, G#4, G#4, rest, G#4, //7
38 G4, G#4, G4, G#4, F#4, C#4, rest, G#4, G4, G#4, G4, G#4, F#4, C#4, rest, F#4, F#4, F#4, rest, E4, rest, F#4, E4, F#4, E4, F#4, E4, F#4, B4, B4, G#4, G#4, A4, //8
39 G#4, rest, G#4, G#4, B4, G#4, rest, rest, G#4, G#4, B4, G#4, rest, rest, G#4, G#4, G#4, F#4, E4, F#4, E4, E4, F4, G4, G4, G#4, rest, B4, rest, //9
40 C#5 };
41 int lengths[] = { 0.5, 0.25, 0.25, 0.25, 0.25, 1, 0.5, 0.5, 0.75, 0.25, 0.5, 0.5, 0.5, 0.25, 0.25, 0.5, 0.5, 0.75, 0.25, 0.5, 1.5, 0.25, 0.25, 0.5, 0.25, 0.25, 0.75, 0.25, 0.5, 0.75, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, //1
42 1, 0.5, 0.5, 0.5, 0.5, 0.5, 0.75, 0.25, 0.25, 0.25, 1, 0.5, 0.5, 0.5, 0.5, 1, 0.25, 0.25, 0.25, 0.75, 0.5, 0.5, 0.5, 0.5, 1, //2
43 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, //3
44 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, //4
45 1, 1, 0.75, 0.5, 0.5, 0.5, 0.5, 0.3, 0.3, 0.3, 0.25, 0.25, 0.25, 0.25, 0.5, 0.3, 0.3, 0.3, 0.25, 0.25, 0.25, 0.5, 0.25, 0.25, 0.25, 1, //5
46 1, 1, 0.75, 0.25, 0.5, 0.5, 0.5, 0.3, 0.3, 0.3, 0.25, 0.25, 0.25, 0.25, 0.5, 0.3, 0.3, 0.3, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 1, 0.5, 0.5, //6
47 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.5, 0.5, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.5, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, //7
48 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.5, 0.5, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.5, 0.25, 0.25, 0.175, 0.175, 0.175, 0.175, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 1, 1, //8
49 0.5, 0.5, 0.5, 0.25, 0.25, 1, 0.5, 1, 0.5, 0.25, 0.25, 1, 0.5, 0.5, 0.5, 0.3, 0.3, 0.3, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.5, 0.5, 0.5, //9
50 100 };
51
```

(3.) 參數和函式庫設定

```

52 #define pb 12 // 定義按鈕引腳
53 #define redled 8// 定義led引腳
54 #define greenled 2 // 定義led引腳
55 #define yellowled 13 // 定義led引腳
56 #define buzzer 7 // 定義蜂鳴器引腳
57 int reallt=0;
58 int t = 150;//每個音符的基準時間
59 int i = 0;// 當前播放的音符索引
60 int ledt = 500;//led閃爍間隔時間
61
62 unsigned long mspMillis = 0; // 上一次切換音符的時間
63 unsigned long redledpMillis = 0; //上一次led亮的時間
64 unsigned long yellowledpMillis = 0; //上一次led亮的時間
65 bool redstate = HIGH; //led狀態 HIGH:led ON , LOW:led OFF
66 bool yellowstate = HIGH; //led狀態 HIGH:led ON , LOW:led OFF
67 bool isPlaying = false; // 是否正在播放音符
68 #include <OneButton.h> // 引用OneButton函式庫
69
70 OneButton button(pb, true); // 建立OneButton物件，名稱為button，按鈕為低態動作
71 boolean startPlay = 0; // 0：停止播放，1：開始播放
72

```

(4.)宣告接腳和開機狀態

```

74 void setup() {
75     pinMode(buzzer, OUTPUT);// 宣告蜂鳴器接腳為輸出
76     pinMode(redled, OUTPUT);// 宣告led接腳為輸出
77     pinMode(yellowled, OUTPUT);// 宣告led接腳為輸出
78     pinMode(greenled, OUTPUT);// 宣告led接腳為輸出
79     digitalWrite(greenled, HIGH);//開機時綠燈恆亮
80     button.attachClick(singleClick); // 單擊
81     button.attachLongPressStart(longPressStart); // 長按開始
82 }

```

(5.)播放音樂

```

89   unsigned long currentMillis = millis(); // 取得當前時間
90   if (startPlay == 1) // 開始播放
91   {
92       redblink();//紅燈閃爍
93       digitalWrite(yellowled,LOW);//黃燈熄滅
94       digitalWrite(greenled,LOW);//綠燈熄滅
95       if (i < sizeof(melody) / sizeof(int)) // 檢查是否還有音符未播放
96       {
97           realt = t * lengths[i]; // 當前音符的持續時間
98           Serial.println("start");
99           if (!isPlaying) // 若處於"未播放"狀態，則開始播放當前音符
100          {
101              tone(buzzer, melody[i], realt); // 發出聲音
102              mspMillis = currentMillis; // 記錄開始播放的時間
103              isPlaying = true; // 標記為正在播放
104          } else if (currentMillis - mspMillis >= t) {
105              // 當前音符播放完畢，切換到下一個音符
106              //noTone(pin); // 停止發聲
107              i++; // 切換到下一個音符
108              isPlaying = false; // 標記為未播放
109          }
110      }

```

(6.)音樂播放完畢

```

111   else // 曲目播放完畢
112   {
113       startPlay = 0; // 停止播放
114       i = 0; // 重置音符索引
115       mspMillis = currentMillis; // 重置時間
116   }
117

```

(7.)暫停及結束時的亮燈狀態

```

else if (startPlay == 0 && i!=0) //播放途中按下暫停
{
    noTone(buzzer);// 停止發聲
    digitalWrite(redled, LOW);//紅燈熄滅
    digitalWrite(greenled, LOW);//綠燈熄滅
    yellowblink();//黃燈閃爍
}
else if (startPlay == 0 ) //播放結束或長按復歸
{
    noTone(buzzer);// 停止發聲
    digitalWrite(redled, LOW);//紅燈熄滅
    digitalWrite(yellowled, LOW);//黃燈熄滅
    digitalWrite(greenled, HIGH);//綠燈恆亮
}

```

(8.)One Button 函式庫副程式

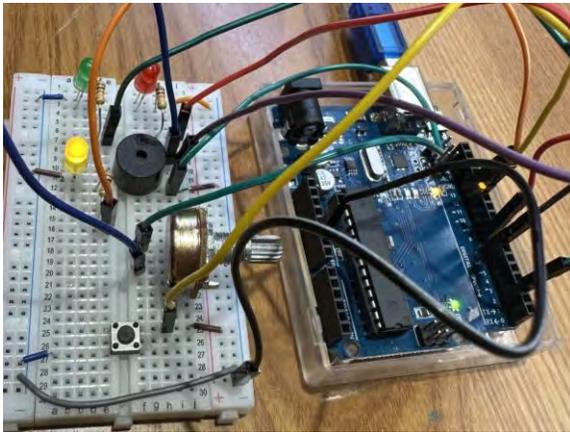
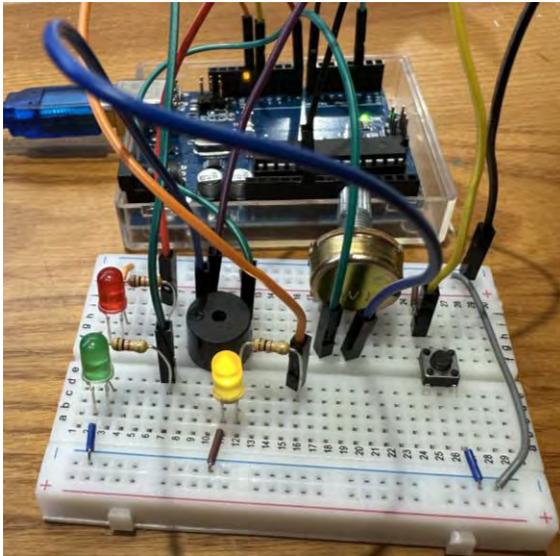
```
136 //單擊按鈕
137 void singleClick() {
138     Serial.println("Button Single Click.");
139     startPlay = !startPlay; // 開始播放或停止播放
140 }
141
142 // 長按開始
143 void longPressStart() {
144     Serial.println("Button Long Press Start.");
145     startPlay = 0; // 停止播放
146     i = 0;
147 }
```

(9.)紅,黃燈閃爍副程式

```
148 void redblink()//紅燈閃爍
149 {
150     unsigned long redledcurrentMillis = millis();// 取得當前時間
151     if (redledcurrentMillis - redledpMillis >= ledt) // 若達到預設計時的時間
152     {
153         // 每經過ledt的時間要做的事
154         redstate = !redstate; // 改變LED狀態
155         digitalWrite(redled, redstate); // 紅燈輸出狀態
156
157         redledpMillis = redledcurrentMillis; // 更新前一次的millis()時間
158     }
159 }
160 void yellowblink()//黃燈閃爍
161 {
162     unsigned long yellowledcurrentMillis = millis();// 取得當前時間
163     if (yellowledcurrentMillis - yellowledpMillis >= ledt) // 若達到預設計時的時間
164     {
165         // 每經過ledt的時間要做的事
166         yellowstate = !yellowstate; // 改變LED狀態
167         digitalWrite(yellowled, yellowstate); // 黃燈輸出狀態
168
169         yellowledpMillis = yellowledcurrentMillis; // 更新前一次的millis()時間
170     }
171 }
172
```

肆、研究結果

(一) 硬體結構



(二) 成果展示

<https://youtu.be/dpThZiVRZQY>

陸、結論

(一)心得

這次實習讓我學到了 Arduino IDE 的基本程式撰寫和電路的連接，並且透過 OneButton 程式庫，使用單一按鈕開關來控制音樂播放、LED 顯示與蜂鳴器發聲。

過程中,我遇到 LED 指示燈須同步顯示的挑戰,LED 需要根據音樂的狀態變化來閃爍或恆亮，因此程式中必須使用 非同步計時 millis() 來控制 LED，而不是 delay()，以確保音樂播放不中斷,且在撰寫程式時要將蜂鳴器的時間和 LED 的時間分開看,才能避免程式的錯誤。

除此之外，這次實作也讓我體驗到電路和程式的關聯,一開始寫程式時，會發現有些程式邏輯看似合理，但當電路實際運行時，卻可能因為 延遲時間、輸出衝突或接線錯誤而發生問題,因此，反覆測試與修改程式是必須的。

(二)未來改進方向

雖然這次的音樂播放器已經能夠正常運作，但仍有一些可以改進的地方：

- 1.支援多個按鈕控制：目前只使用了一顆按鈕，未來可以增加 音量調整鈕和切換曲目鈕，讓系統更加人性化。
- 2.LCD 液晶顯示器：：可以新增 LCD 液晶顯示器顯示當前播放狀態，例如音樂名稱、播放時間等資訊，提升使用者體驗。
- 3.使用 DFPlayer Mini：目前蜂鳴器的音樂是透過 `tone()` 產生單音調旋律，未來可以改用 DFPlayer Mini 播放 MP3 檔案，讓音質更豐富。

(三)總結

這次實習不僅讓我學會了 使用 Arduino 控制蜂鳴器與 LED，還讓我體驗到程式和電路設計的樂趣與挑戰。從程式設計到電路連接，每一步都要求精準,因此更需要耐心和決心。整體來說，這次實驗讓我更有信心應對未來的專題製作，也讓我對 Arduino 的應用有了更深的理解。

捌、參考資料

[蜂鳴器、tone\(\)進階練習：結合按鈕開關、蜂鳴器、LED 指示燈的音樂播放器 – 漢之民也](#)

[Arduino 快速入門-1 什麼是 Arduino? | 偉恩電子](#)

[《Arduino 入門》第二篇：開始使用 Arduino IDE 寫程式](#)

[Arduino 按鈕控制 LED | 瘋狂創客](#)

[製造蜂鳴器的專家,壓電式蜂鳴器,警報器,號角型警報器,電磁式警報器,自激式蜂鳴器,他激式蜂鳴器,音片,陶瓷音片,迷你喇叭,數位麥克風,有源電磁式蜂鳴器 無源電磁式蜂鳴器,光鍵股份有限公司 專業製造商,數千種產品 歡迎蒞臨指教!](#)

[《Arduino 入門》第六篇：有源蜂鳴器、無源蜂鳴器](#)

[關於 Arduino uno r3 開發板的介紹 – 台灣物聯科技 TaiwanIOT Studio](#)

[認識 Arduino UNO R3 - 台灣智能感測科技](#)

[03.電機與電子群 - Google 雲端硬碟](#)

