

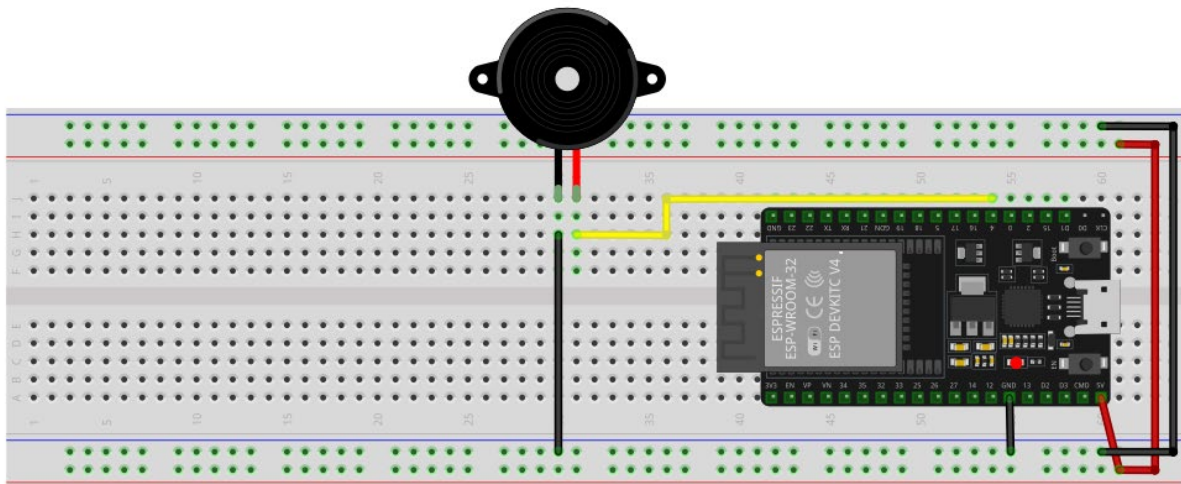
## 單元 2-1：聲音控制與輸出

### 學習目標：

- 練習讓小蜂鳴器發出聲音
- 製作一首歌曲
- 了解指令運用
  - a. `tone(sp,notes[i],1000);` //tone(接腳,聲音頻率,延遲時間)
  - b. `noTone(SP)`
  - c. 陣列 `notes[8]`
- 實際操作可變電阻(光敏電阻器)分壓電路連接

### 電路接線圖：

輸出接腳為 GPIO 4。



### 材料需求：

ESP32 開發板	X1
蜂鳴器(小喇叭)	X1
電阻 220Ω	X1
LED	X1

### 設備需求：

三用電表	X1
------	----

### 指令說明：

將隨著程式的編寫，逐步說明指令功能，請大家循序漸進學習。

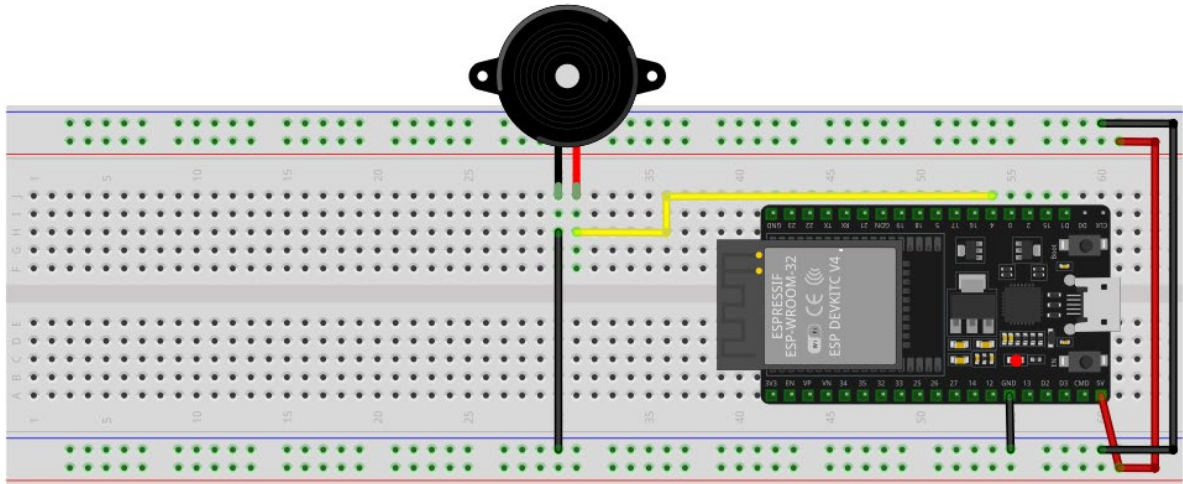
- a. `tone(sp,notes[i],1000);` //tone(接腳,聲音頻率,延遲時間)  
ex. `tone(4,500,1000);` GPIO4 接腳發出 500HZ 的聲音持續 1 秒
- b. `noTone(SP)` //不發出聲音

c. `notes[i]` 可將資料儲存在陣列變數

ex.`notes[8]={Do, Re, Mi, Fa, So, La, Si,Do2}; //音階頻率放入陣列`

`notes[0~7]`中

接線圖：輸出為 GPIO4



程式內容(一) `Speaker_tone`

```
int sp=4; //設定 sp=4 (GPIO4)
void setup() {
    pinMode(sp,OUTPUT); //設定 GPIO4 為喇叭輸出腳
}
void loop()
{
    tone(sp,500); //發出 500HZ 的音頻
    delay(500);
    tone(sp,1000); //發出 1000HZ 的音頻
    delay(500);
}
```

程式內容(二) `Speaker_do_re_mi`

```
#define Do 262
#define Re 294
#define Mi 329
#define Fa 349
```

```

#define So 392
#define La 440
#define Si 493
#define Do2 523
int sp=4;
int i;
int notes[8]={Do, Re, Mi, Fa, So, La, Si,Do2}; //音階頻率放入陣列 notes[0~7]中

void setup()
{

}

void loop()
{
  for(i=0;i<8;i++)
  {
    tone(sp,notes[i],1000); //tone(接腳,頻率,時間)
    noTone(sp);
    delay(50);
  }
}

```

---

整合練習：利用上述指令編寫小蜜蜂歌曲

### 小蜜蜂

1=C $\frac{4}{4}$  ♩=120

德國民謠

記譜：Sunghsi

(1)

| 5 3 3 - | 4 2 2 - | 1 2 3 4 | 5 5 5 - |

(5)

| 5 3 3 - | 4 2 2 - | 1 3 5 5 | 3 - 0 0 |

(9)

| 2 2 2 2 | 2 3 4 - | 3 3 3 3 | 3 4 5 - |

(13)

| 5 3 3 - | 4 2 2 - | 1 3 5 5 | 1 - 0 0 ||

校歌

中庸速度

C 調 4/4

5 5 3 4 5 | 6 5 4 2 3 | <sup>V</sup> 5 6 3 5 6 3 | 1̇ 5 3 5 6 - |  
龍崗之麓碧潭之旁茶香馥郁稻實金黃

1̇ 5 3 4 5 | 6 5 6·7 <sup>V</sup> 1̇ | 2̇ 1̇ 2̇ 6 | 3̇ 2̇ 1̇ 7 1̇ - |  
龍潭農工使命非常足兵足食國族斯光

1 2 3 1 | 5 6 7 5 2̇ - | <sup>V</sup> 3 4 5 3 | 1̇ 1̇ 1̇ 3 5 - |  
習勞崇儉作民榜樣敬業樂群事無怠荒

1̇ 1̇ 5 6 | 6 7 1̇ 2̇ 3̇ - | <sup>V</sup> 2̇ 2̇ 6 6 | 5 2̇ 3̇ 1̇ |  
龍潭建設康樂富強中華建設悠久無

1̇ - - 0 ||

疆

const int c = 261; //定義音階頻率

const int d = 294;

const int e = 329;

const int f = 349;

const int g = 391;

```

const int gS = 415;
const int a = 440;
const int aS = 466;
const int b = 494;
const int cH = 523;
const int cSH = 554;
const int dH = 587;
const int dSH = 622;
const int eH = 659;
const int fH = 698;
const int fSH = 740;
const int gH = 784;
const int gSH = 830;
const int aH = 880;

const int buzzerPin =4 ; //定義蜂鳴器輸出接腳 GPIO4
const int ledPin1 = 5; //定義 LED1 輸出接腳 D1
const int ledPin2 = 0; //定義 LED2 輸出接腳 D3

int counter = 0;

void setup()
{
  pinMode(buzzerPin, OUTPUT); //設定 buzzerPin 為數位輸出接腳
  pinMode(ledPin1, OUTPUT); //設定 ledPin1 為數位輸出接腳
  pinMode(ledPin2, OUTPUT); //設定 ledPin2 為數位輸出接腳
}

void loop()
{

  firstSection(); //演奏第一段 firstSection()副程式

  delay(650);

```

```
}
```

```
void beep(int note, int duration) //副程式 beep，音階 note 與時間 duration
```

```
{
```

```
tone(buzzerPin, note, duration); //輸出接腳、音階 note 與時間 duration
```

```
if(counter % 2 == 0) //判斷 counter/2 餘數為零，功能為改變音階切換燈號
```

```
{
```

```
digitalWrite(ledPin1, HIGH); //如果 餘數為零，ledPin1 亮、ledPin2 滅
```

```
delay(duration);
```

```
digitalWrite(ledPin1, LOW);
```

```
}else
```

```
{
```

```
digitalWrite(ledPin2, HIGH); //如果 餘數不為零，ledPin2 亮、ledPin1 滅
```

```
delay(duration);
```

```
digitalWrite(ledPin2, LOW);
```

```
}
```

```
noTone(buzzerPin); //停止輸出聲音 50mS
```

```
delay(50);
```

```
counter++; //counter 加一
```

```
}
```

```
void firstSection() //副程式 firstSection，第一段
```

```
{
```

```
beep(g, 500); //1
```

```
beep(g, 500);
```

```
beep(e, 250);
```

```
beep(f, 250);
```

```
beep(g, 500);
```

```
beep(a, 500);
```

```
beep(g, 500);
```

```
beep(f, 250);
```

```
beep(d, 250);  
beep(e, 500);
```

```
beep(g, 250);  
beep(a, 250);  
beep(e, 250);  
beep(g, 250);  
beep(a, 500);  
beep(e, 500);
```

```
beep(cH, 250);  
beep(g, 250);  
beep(e, 250);  
beep(g, 250);  
beep(a, 1000);
```

```
beep(cH, 500); //2  
beep(g, 500);  
beep(e, 250);  
beep(f, 250);  
beep(g, 500);
```

```
beep(a, 500);  
beep(g, 500);  
beep(a, 250);  
beep(b, 250);  
beep(cH, 500);
```

```
beep(dH, 500);  
beep(cH, 500);  
beep(dH, 500);  
beep(a, 500);
```

```
beep(eH, 250);  
beep(dH, 250);  
beep(cH, 250);  
beep(b, 250);  
beep(cH, 1000);
```

```
beep(c, 500); //3  
beep(d, 500);  
beep(e, 500);  
beep(c, 500);
```

```
beep(g, 250);  
beep(a, 250);  
beep(b, 250);  
beep(g, 250);  
beep(dH, 1000);
```

```
beep(e, 500);  
beep(f, 500);  
beep(g, 500);  
beep(e, 500);
```

```
beep(cH, 250);  
beep(cH, 250);  
beep(cH, 250);  
beep(e, 250);  
beep(g, 1000);
```

```
beep(cH, 750); //4  
beep(cH, 250);  
beep(g, 500);  
beep(a, 500);
```

```
beep(a, 250);  
beep(b, 250);  
beep(cH, 250);  
beep(dH, 250);  
beep(eH, 1000);
```

```
beep(dH, 750);  
beep(dH, 250);  
beep(a, 500);  
beep(a, 500);
```

```
beep(g, 500);  
beep(eH, 500);  
beep(dH, 750);  
beep(cH, 250);  
beep(cH, 1500);
```

```
}
```