

# 1

## 音樂盒與電子琴

1 使用類別程式設計

2 音樂盒

3 8鍵電子琴

講師 張傑帆 Chang, Jie-Fan

聲音的高低與聲音的大小，可透過控制頻率與通過的電壓來實現，讓我們使用Arduino來製作一簡單的音樂盒與電子琴吧！

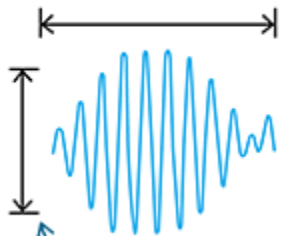
# 發音體和聲音

電子裝置常見的發音體為揚聲器（喇叭，speaker）和蜂鳴器（piezo transducer）



聲音是由震動產生, 其震動的頻率稱為**音頻**，音頻的範圍介於20Hz~200KHz之間，普通人可聽見聲音的頻率範圍約為20Hz~20KHz。

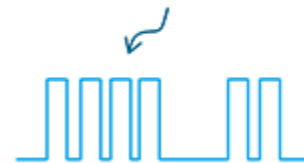
震動頻率越高，聲音越高亢，反之越低沉。



振幅越大，音量也越大。



從Arduino數位腳輸出的訊號，是震幅固定的方波，音質稱不上優美。



# 函式說明

`tone()` 函式的功能是輸出特定頻率的方波至指定數位接腳。`tone()` 函式的格式如下，有三個參數必須設定，`pin` 參數指定聲音輸出的數位接腳，輸出可連接至壓電式蜂鳴器或其他揚聲器。`frequency` 參數設定聲音輸出頻率，單位為赫芝 (Hz)。`duration` 參數設定聲音持續時間，單位為毫秒 (ms)。如果沒有指定 `duration` 參數，則必須使用 `noTone()` 函式關閉聲音。Arduino 板一次只能輸出一種音調，這是因為 `tone()` 函式使用計時器 (`timer2`) 來產生音調，如果您想輸出聲音至另一數位接腳，您必須先使用 `noTone()` 函式將目前數位接腳的輸出音調關閉，否則將沒有任何效果。

格式： `tone(pin, frequency)`

---

`tone(pin, frequency, duration)`

---

範例： `tone(2, 1000, 500)`

//數位接腳 2 輸出 1000Hz 方波，時間 0.5 秒。

# 函式說明

## noTone( ) 函式

noTone( ) 函式的功能是關閉輸出。noTone( ) 函式的格式如下，有一個參數必須

設定，pin 參數指定輸出數位接腳。

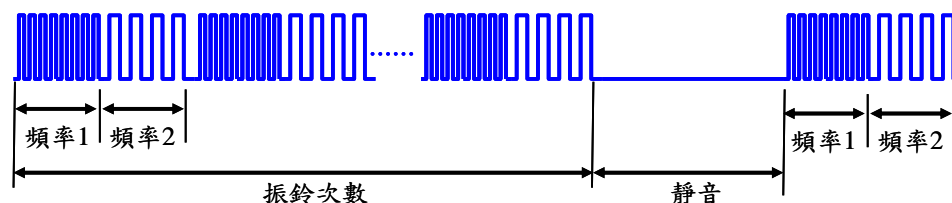
格式： noTone(pin)

範例： noTone(2); //關閉數位接腳 2 的輸出音調。

## 電話聲實習

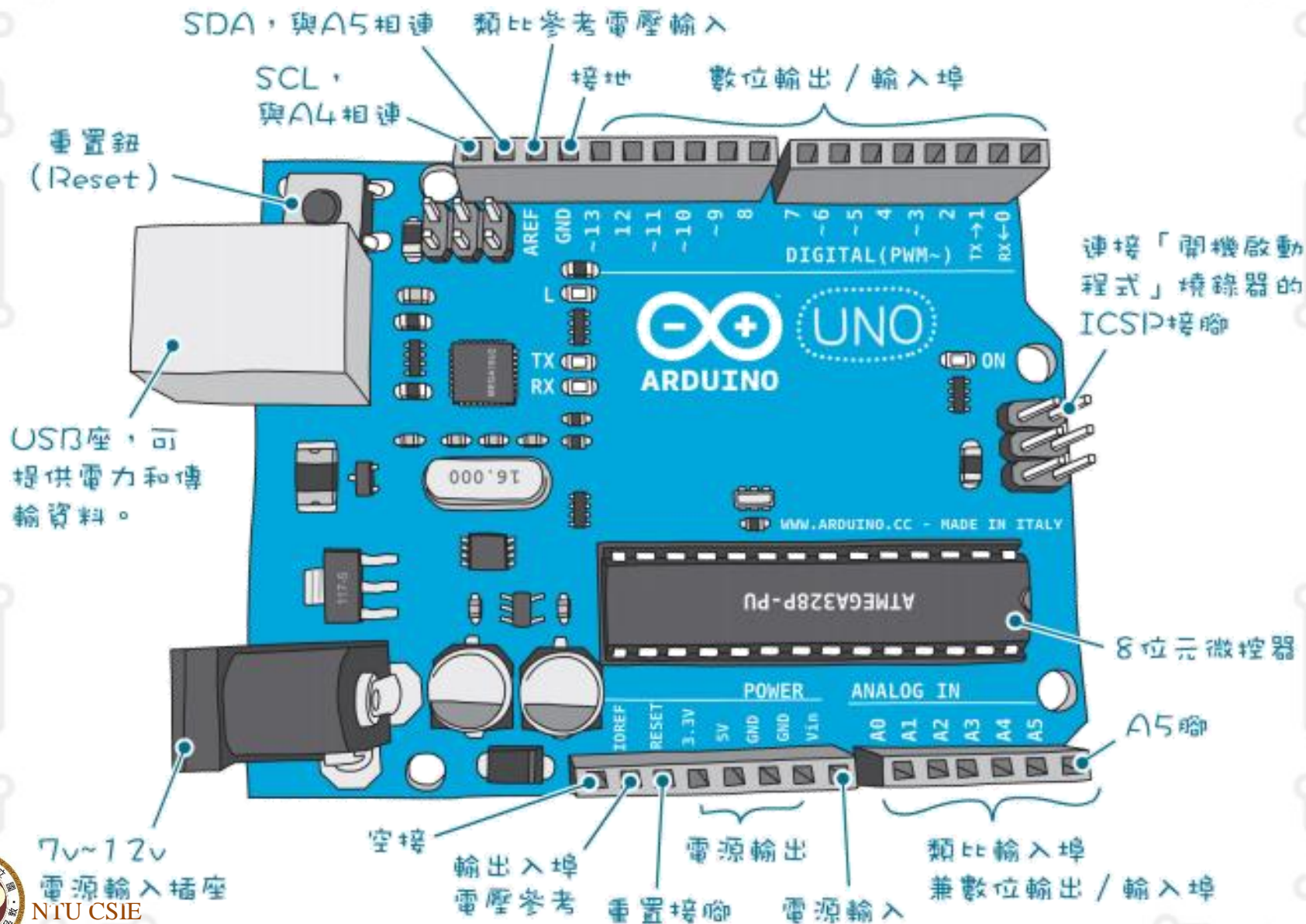
### □ 功能說明：

利用 Arduino 板輸出電話聲音。如圖 11-2 所示電話聲波形是由兩種不同頻率的信號組合成一次振鈴 (ringer)，經過多次振鈴後，再靜音一段時間，重覆不斷。

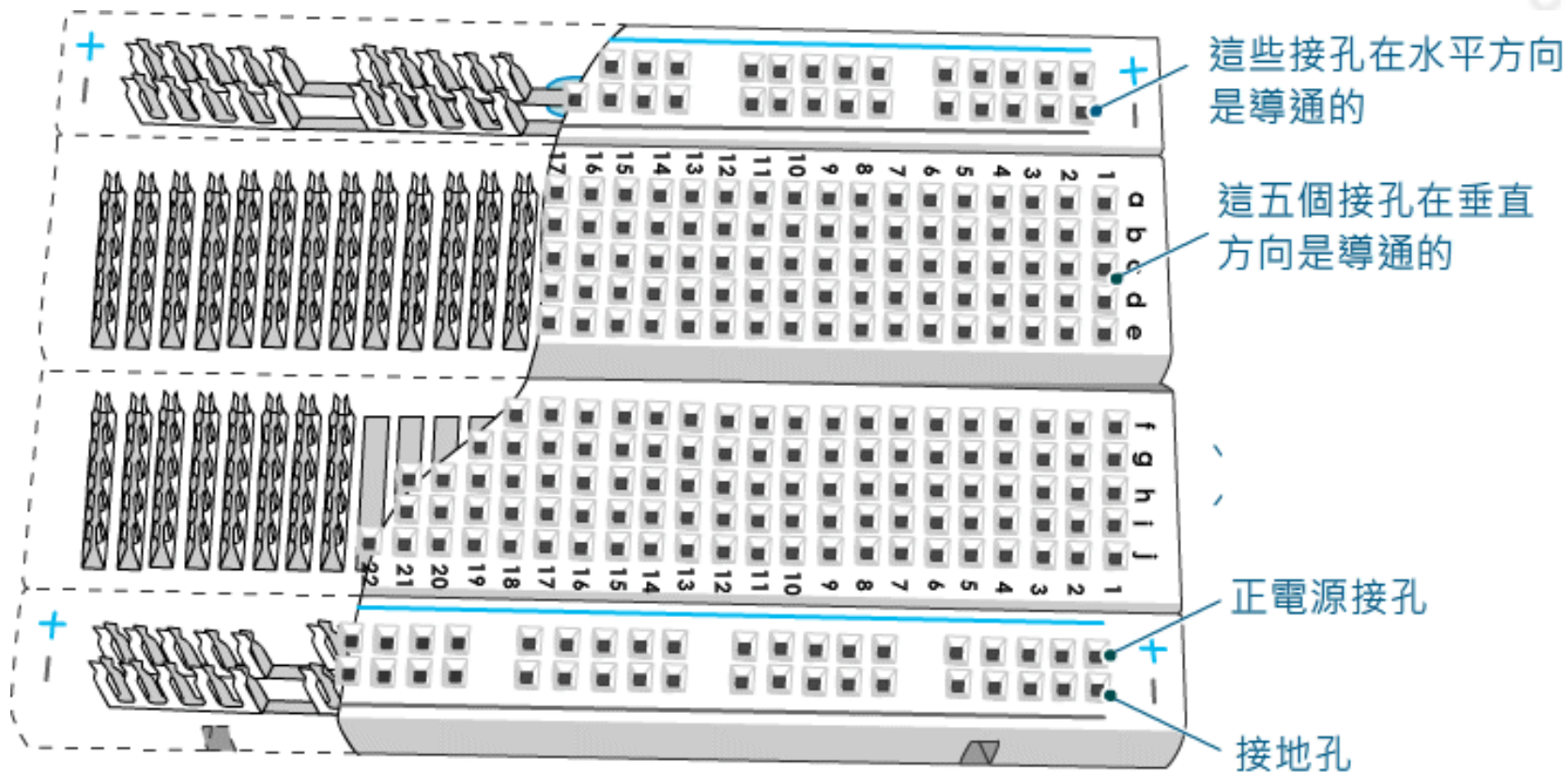




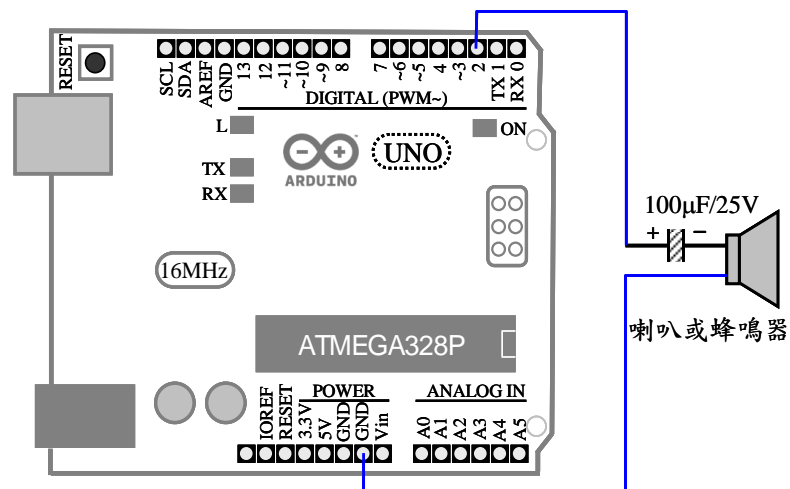
# Arduino的接腳



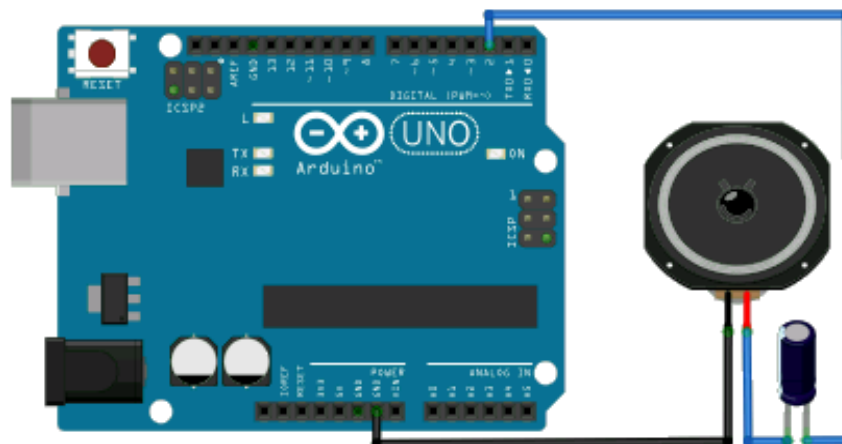
# 麵包板



□ 電路圖及麵包板接線圖：



電話聲實習電路圖

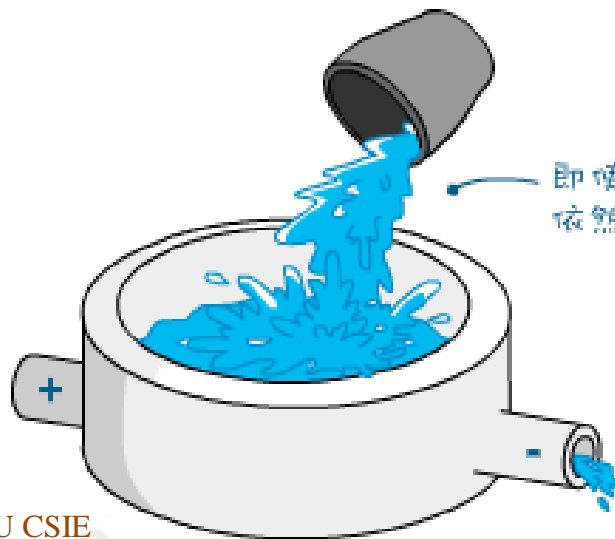
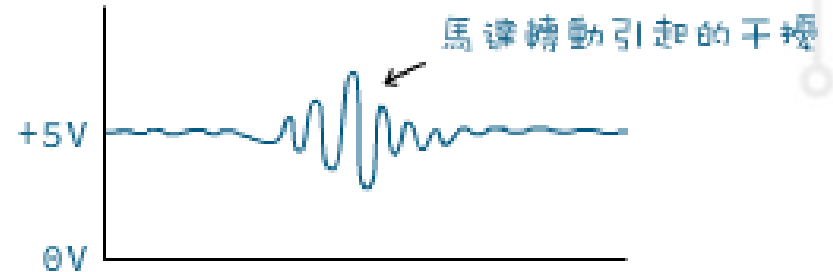


電話聲實習麵包板接線圖

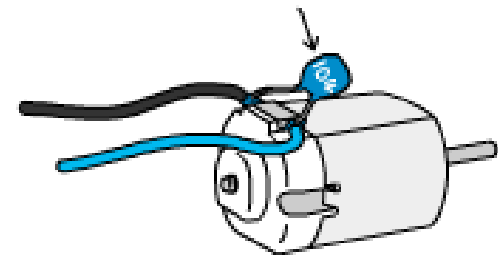
# 電容

電容器就是電的容器，簡稱電容，單位是法拉（Farad，簡寫成F），代表電容所能儲存的電荷容量。

在積體電路和馬達的電源接腳，經常可以發現電容，用於吸收電源瞬間變化的雜訊。

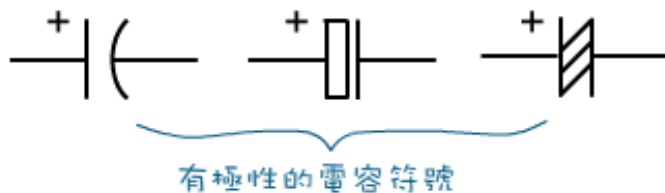
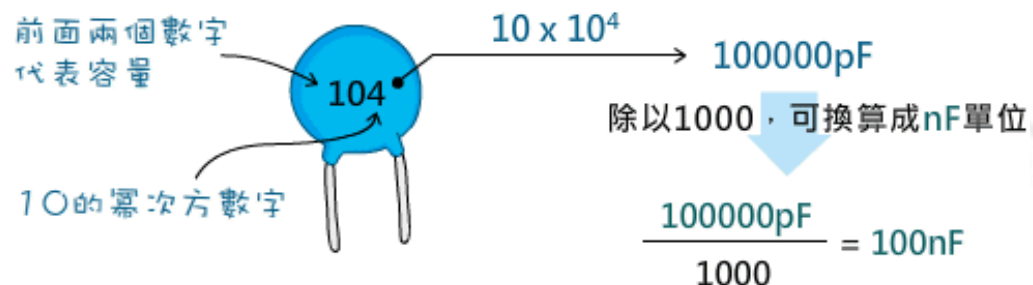


焊接在馬達電源的電容，  
可避免雜訊干擾處理器。





# 電容



# 實作練習



□ 程式： I101.ino

```
const int speaker=2;           //數位接腳 2 連接至喇叭。
void setup()
{
}
void loop()
{
    for(int i=0;i<10;i++)      //振鈴 10 次。
    {
        tone(speaker,1000);    //輸出 1000Hz 音調。
        delay(50);             //延遲 50ms。
        tone(speaker,500);     //輸出 500Hz 音調。
        delay(50);             //延遲 50ms。
    }
    noTone(speaker);           //靜音 2 秒。
    delay(2000);
}
```



## 練習

1. 設計 Arduino 程式，輸出由低至高的警車聲音。 Hint. 500~1000 step=10, delay 20
2. 設計 Arduino 程式，輸出嗶!嗶!嗶!警報聲音。 Hint. 500 delay 200, stop delay 200

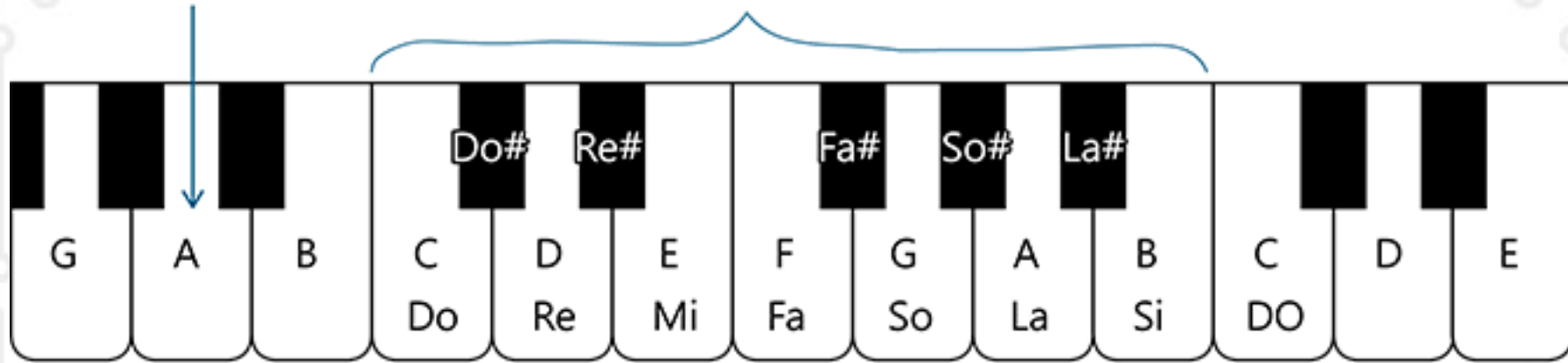
# 音高與節拍（一）

聲音的頻率（音頻）高低稱為音高（pitch），鋼琴鍵盤就是依照聲音頻率的高低階級（音階）順序來排列。

A, B, C, D, ... 是「音名」

Do, Re, Mi, ... 是「唱名」

每段音階分成12個半音，其中每個音的頻率是前一個的1.05946倍（即： $\sqrt[12]{2}$  或  $2^{\frac{1}{12}}$ ）。



標準音（A音）  
440Hz

中音Do 523Hz  
中音Re 587Hz

中音La  
880Hz

高音DO  
1046Hz

$$523\text{Hz} \times 1.05946^2 \approx 587\text{Hz}$$

$$523\text{Hz} \times 1.05946^9 \approx 880\text{Hz}$$

每段音階的頻率比為兩倍，即： $2^1$



# 音高與節拍（二）

	0	1	2	3	4	5	6	7	8
				位於鍵盤中間的中央C音 (Do)				88鍵樂器的最高音	
<b>C</b>	16	33	65	131	262	523	1046	2093	4186
<b>C#</b>	17	35	69	139	277	554	1109	2217	4435
<b>D</b>	18	37	73	147	294	587	1175	2349	4699
<b>D#</b>	19	39	78	156	311	622	1245	2489	4978
<b>E</b>	21	41	82	165	330	659	1319	2637	5274
<b>F</b>	22	44	87	175	349	698	1397	2794	5588
<b>F#</b>	23	46	93	185	370	740	1480	2960	5920
<b>G</b>	25	49	98	196	392	784	1568	3136	6272
<b>G#</b>	26	52	104	208	415	831	1661	3322	6645
<b>A</b>	28	55	110	220	440	880	1760	3520	7040
<b>A#</b>	29	58	117	233	466	932	1864	3729	7459
<b>B</b>	31	62	123	247	493	988	1976	3951	7902

88鍵樂器的最低音

標準音（用於調校樂器，有些採442Hz）



# 音高與節拍 (三)

除了音高，構成旋律的另一個要素是節拍 (beat)，它決定了各個音的快慢速度。

$\text{♩} = 200$  ← 代表一分鐘內有200個二分音符，亦即，每拍佔1/200分鐘。

659 (E5) 659 659 523 (C5) 659 784 (G5)

代表一小節有兩拍以二分音符為一拍

四分休止符

原本是F音，加上升號，代表「黑鍵」的F#音。

四分音符

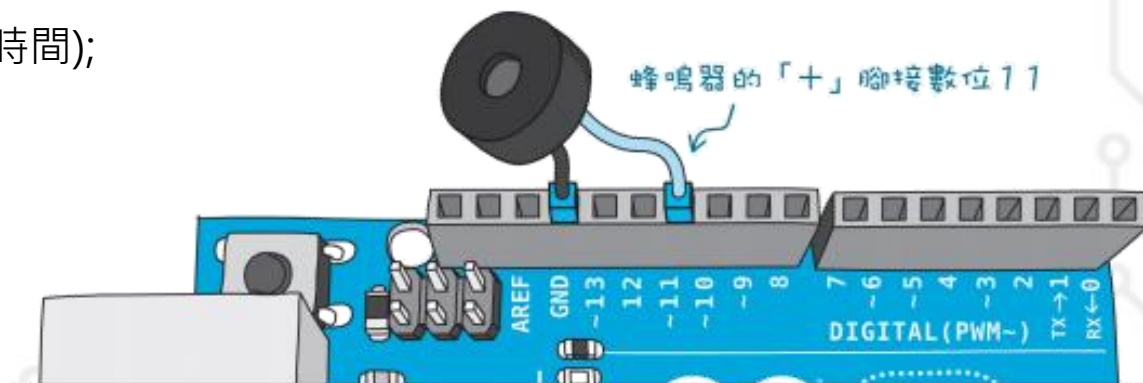
二分休止符

Arduino編輯器內建"Tone" (音調) 程式庫，可以輸出指定頻率的聲音和持續時間。

`tone`(輸出腳位, 頻率, 持續時間);

或：

`tone`(輸出腳位, 頻率);



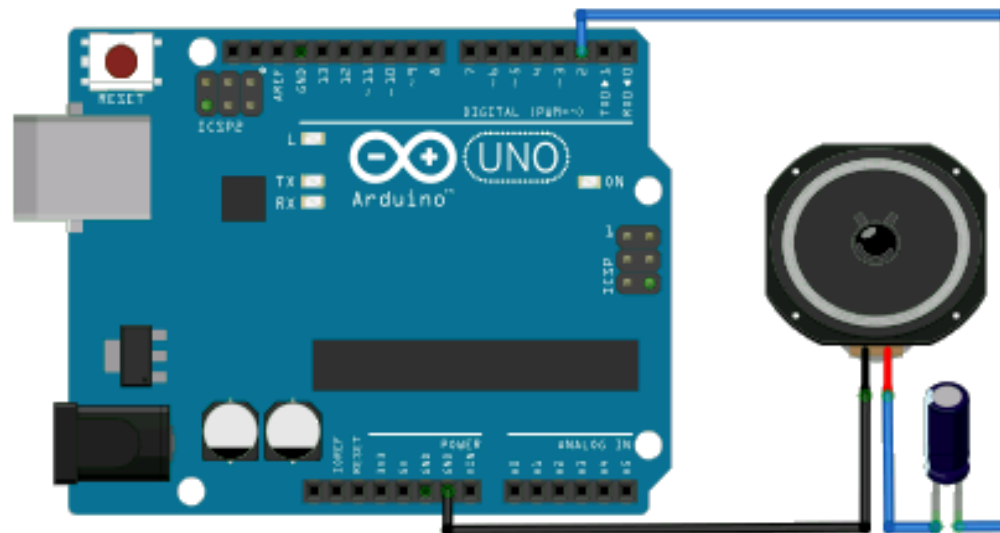
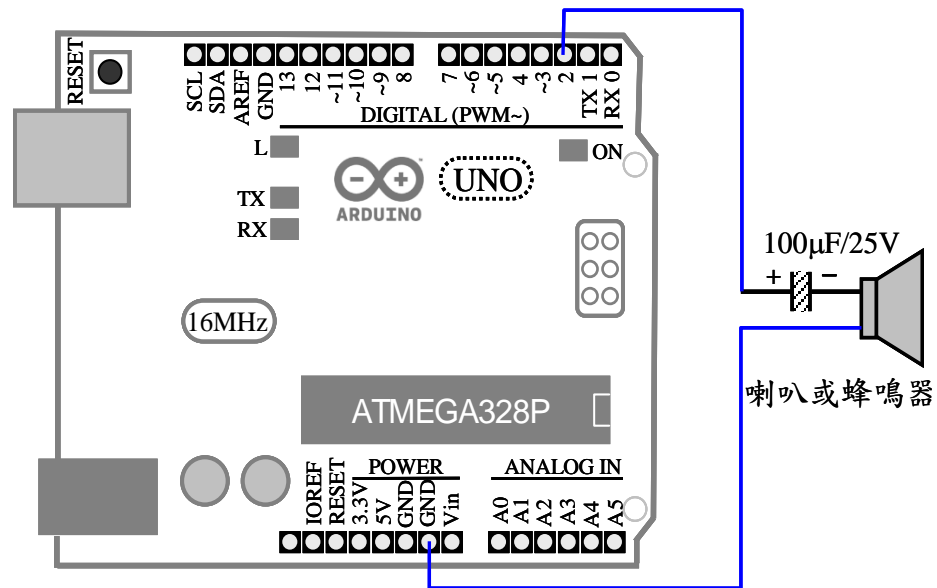


# 實作練習

表 11-1 C 調音符表

音階	n	1	2	3	4	5	6	7	8	9	10	11	12
	音符	C (Do)	C# (Do# )	D (Re)	D# (Re# )	E (Mi)	F (Fa)	F# (Fa# )	G (So)	G# (So# )	A (La)	A# (La# )	B (Si)
低音	頻率 (Hz)	262	277	294	311	330	349	370	392	415	440	466	494
中音	頻率 (Hz)	523	554	587	622	659	698	740	784	831	880	932	988
高音	頻率 (Hz)	1046	1109	1175	1245	1318	1397	1480	1568	1661	1760	1865	1976

# 實作練習



# 實作練習

□ 程式： I102.ino

```
const int speaker=2;                //數位接腳 2 連接至喇叭。
const int toneTable[8]={523,587,659,694,784,880,988,1046}; //音符。
void setup()
{
}
void loop()
{
  for(int i=0;i<8;i++)              //輸出 C、D、E、...等 8 個音符。
  {
    tone(speaker,toneTable[i]);      //輸出音符。
    delay(500);                      //延遲 0.5 秒。
  }
  noTone(speaker);                  //關閉聲音。
}
```



## 練習

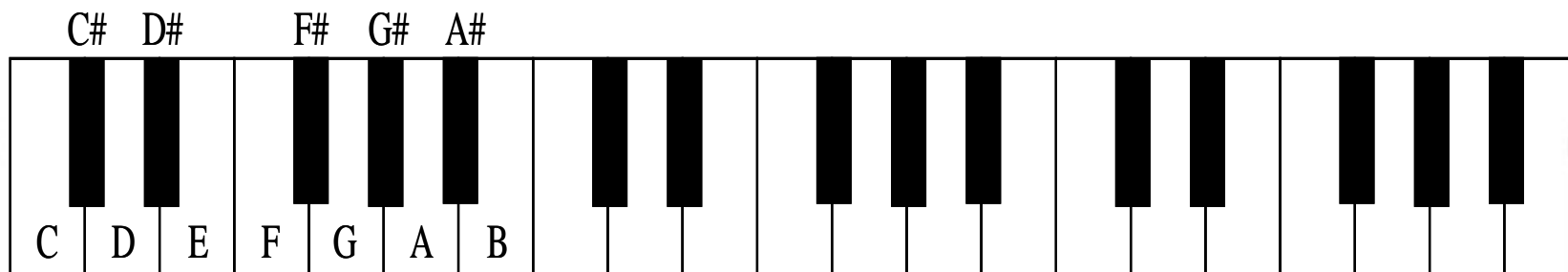
1 · 設計 Arduino 程式，依序播放低音階 C、B、A、G、F、E、D 及中音階 C 等 8 個音符。

# 實作練習

## 電子琴實習

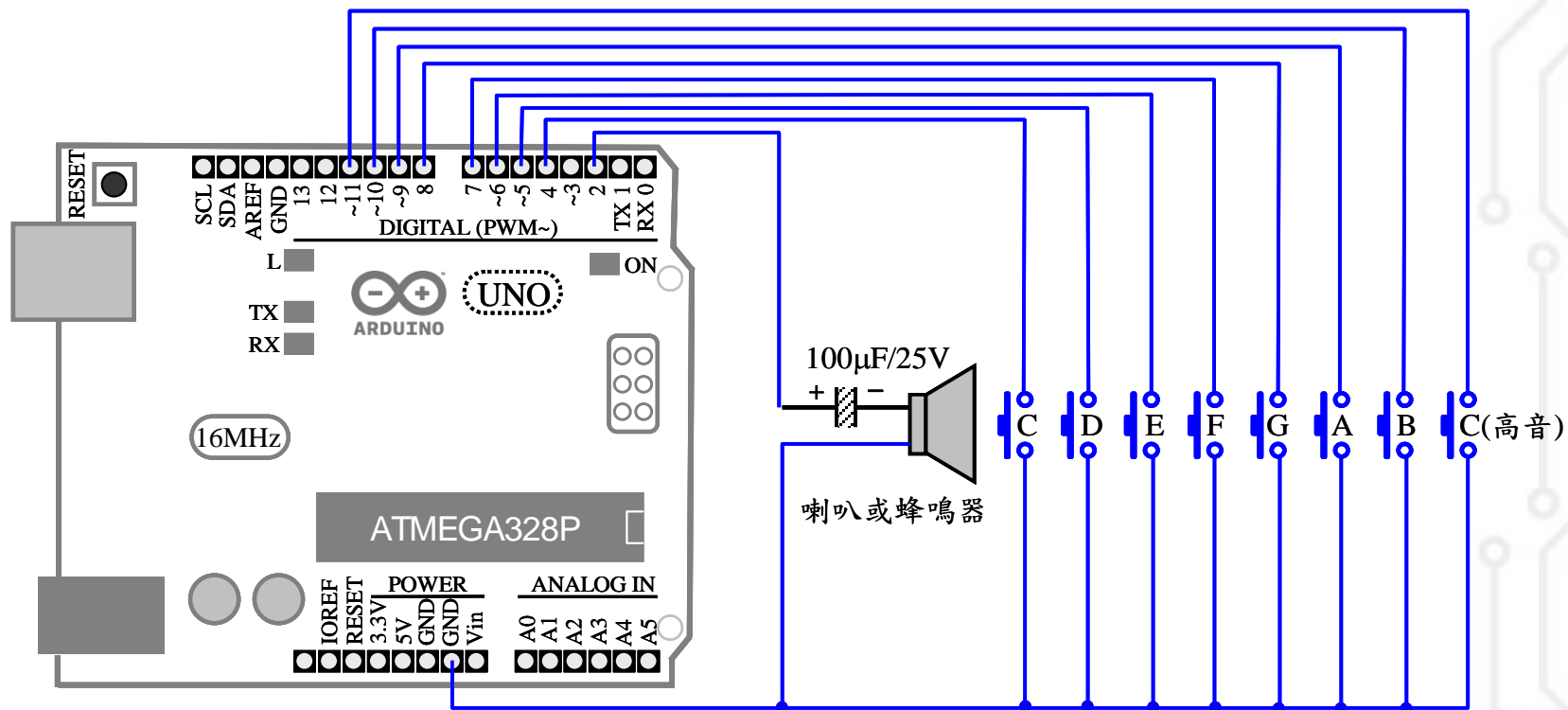
### □ 功能說明：

利用 Arduino 板模擬 8 鍵電子琴。如圖 11-5 所示琴鍵音符，本例使用 8 個 TACK 按鍵開關來模擬琴鍵中音 C、D、E、F、G、A、B 及高音 C 等 8 個音符。



# 實作練習

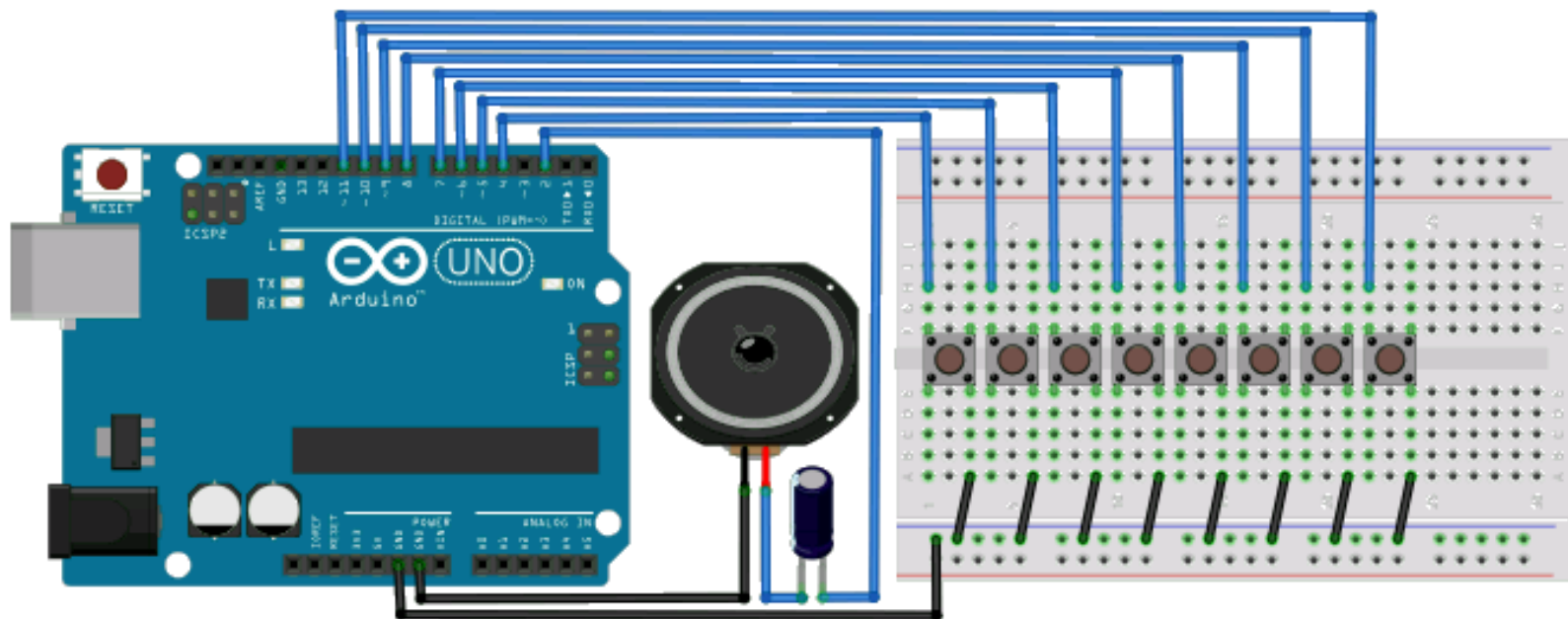
- 電路圖及麵包板接線圖：



電子琴實習電路圖



# 實作練習



電子琴實習麵包板接線圖

# 實作練習

□ 程式： ch11-3.ino

```
const int speaker=2; //數位接腳 2 連接至喇叭。
const int sw[8]={4,5,6,7,8,9,10,11}; //數位接腳 4~11 連接至琴鍵。
const int frequency[8]={523,587,659,694,784,880,988,1046}; //音符頻率表。
int i; //迴圈索引。
int val; //琴鍵狀態。
void setup()
{
    for(i=0;i<8;i++) //設定數位接腳 4~11 為輸入模式。
        pinMode(sw[i],INPUT_PULLUP);
}
void loop()
{
    for(i=0;i<8;i++)
    {
        val=digitalRead(sw[i]); //讀取琴鍵狀態。
        if(val==0) //按下琴鍵?
            tone(speaker,frequency[i],100); //播放所按下琴鍵的音符。
    }
}
```

# 實作練習



## 練習

1. 設計 Arduino 程式，模擬 8 鍵電子琴功能。使用 8 個 TACK 按鍵開關來模擬琴鍵低音階 C、D、E、F、G、A、B 及中音階 C 等 8 個音符。
2. 試將按鍵加到10個成 C、D、E、F、G、A、B、C、D、E



# 實作練習

## 播放旋律實習

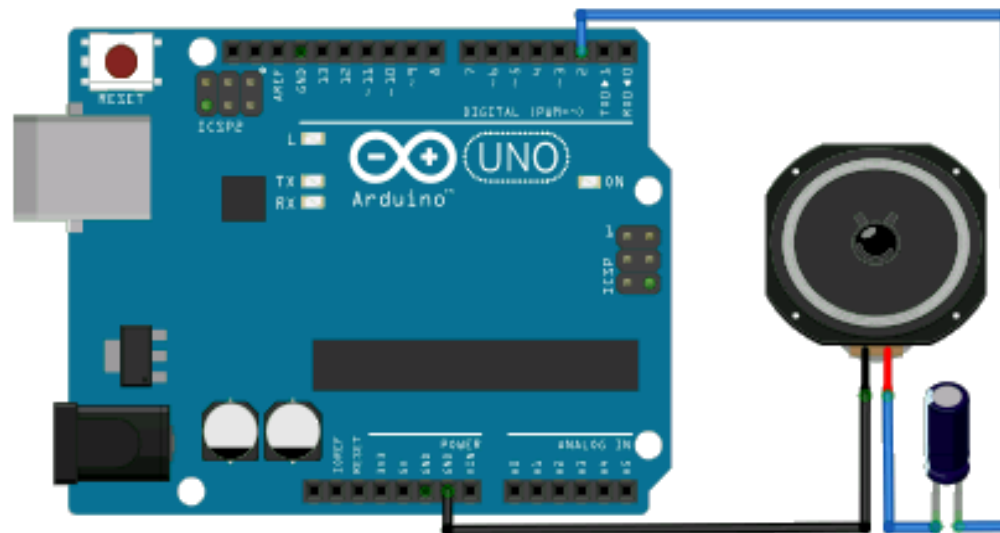
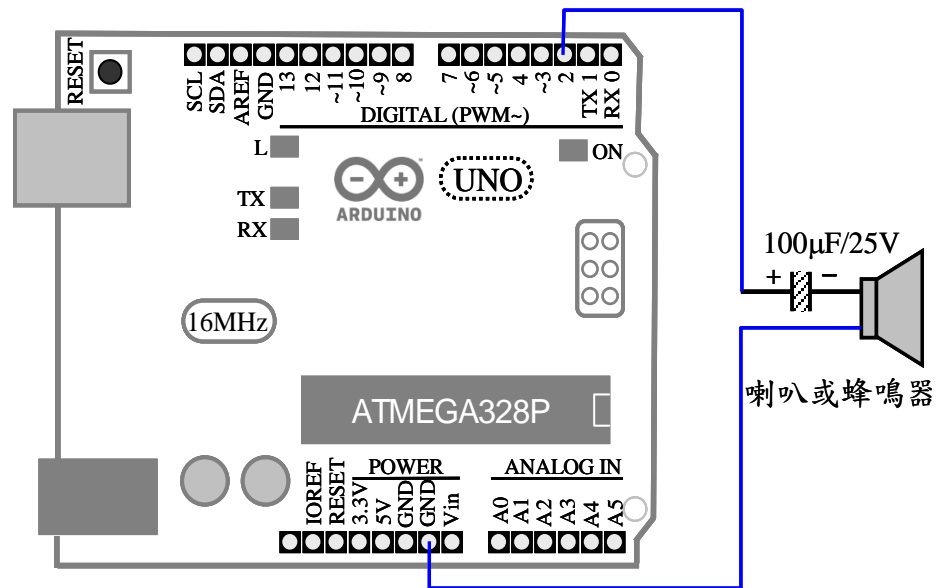
### □ 功能說明：

利用 Arduino 板播放鋼琴入門音樂—小蜜蜂 ( Little Bee )。如表 11-2 所示小蜜蜂簡譜，數字 1~5 代表音符 C、D、E、F、G。每一段有 4 小節，每小節有 4 拍，如果演奏的速度是每分鐘 180 拍，則每拍的時間是  $60/180$  秒= $60000/180$  毫秒。

表 11-2 小蜜蜂簡譜

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

# 實作練習





## 實作練習

□ 程式： I104.ino

```
const int speaker=2; //數位接腳2 連接至喇叭。
```

```
unsigned int frequency[7]={523,587,659,694,784,880,988}; //音符頻率。
```

```
char toneName[]="CDEFGAB"; //音符表。
```

```
char beeTone []="GEEFDDCDEFGGGEEFDDCEGGEDDDDDDEFEEEEFFGGEEFDDCEGGC";
```

```
byte beeBeat[]={1,1,2,1,1,2,1,1,1,1,1,1,2,
```

1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 4,

1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2,

$$1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 4\};$$

```
const int beeLen=sizeof(beeTone); //小蜜蜂音符總數。
```

```
unsigned long tempo=180; //每分鐘180拍。
```

```
int i,j;
```

```
void setup()
```

 $\{ \}$

# 實作練習

```
void loop()  
{  
    for(i=0;i<beeLen;i++)                //播放小蜜蜂樂曲。  
        playTone(beeTone[i],beeBeat[i]);  
    delay(3000);                          //間隔 3 秒重覆播放。  
}  
void playTone(char toneNo,byte beatNo)    //播放音符函式。  
{  
    unsigned long duration=beatNo*60000/tempo; //計算每拍時間（毫秒）。  
    for(j=0;j<7;j++)  
    {  
        if(toneNo==toneName[j])          //查音符表。  
        {  
            tone(speaker,frequency[j]);    //播放音符。  
            delay(duration);               //此音符的節拍。  
            noTone(speaker);               //關閉聲音。  
        }  
    }  
}
```

# 實作練習



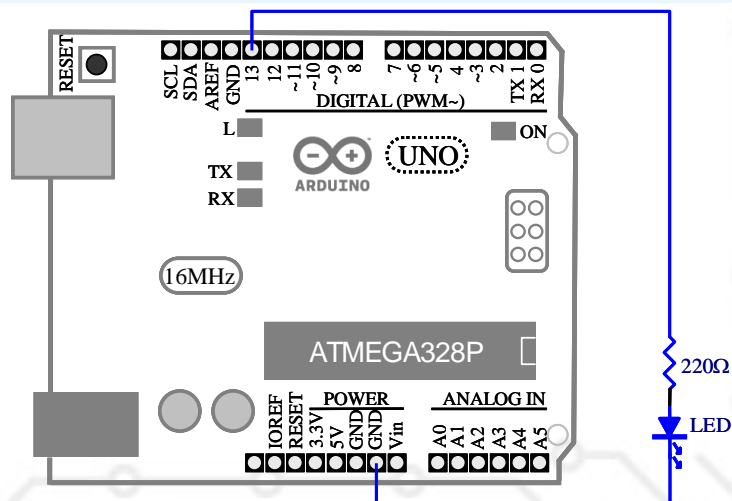
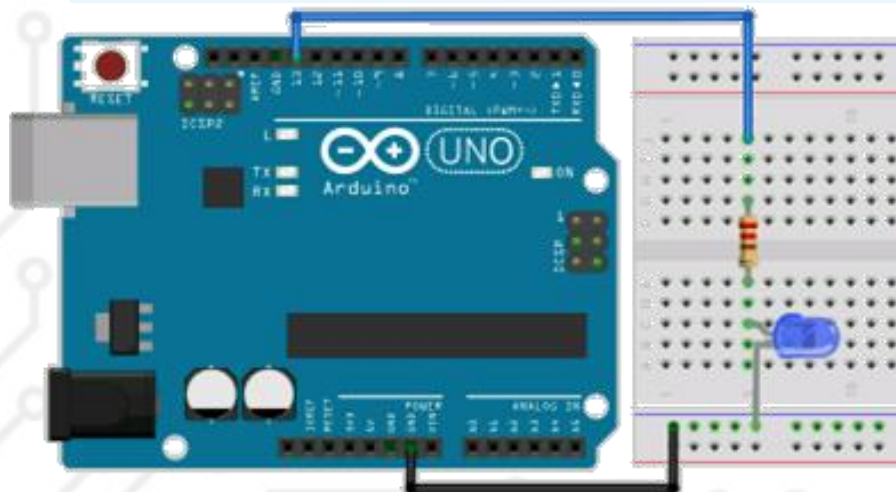
## 練習

1. 設計 Arduino 程式，利用 Arduino 板播放鋼琴入門音樂—小星星（Little Star）。小星星簡譜如表 11-3 所示，每一段有 4 小節，每小節有 4 拍，演奏速度是每分鐘 120 拍。


表 11-1 小星星簡譜

1	1	5	5	6	6	5	-	4	4	3	3	2	2	1	-
5	5	4	4	3	3	2	-	5	5	4	4	3	3	2	-
1	1	5	5	6	6	5	-	4	4	3	3	2	2	1	-


2. 接續上題，使用 6 個 LED 顯示 C、D、E、F、G、A 等 6 個目前正在播放的音符。




# 電阻的色環(補充)



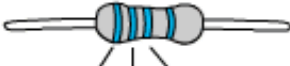
數字1	數字2	乘數	誤差率
黑 0	黑 0	黑 0	金 $\pm 10\%$
棕 1	棕 1	棕 1	銀 $\pm 5\%$
紅 2	紅 2	紅 2	棕 $\pm 1\%$
橙 3	橙 3	橙 3	紅 $\pm 2\%$
黃 4	黃 4	黃 4	綠 $\pm 0.5\%$
綠 5	綠 5	綠 5	藍 $\pm 0.25\%$
藍 6	藍 6	藍 6	紫 $\pm 0.1\%$
紫 7	紫 7	紫 7	
灰 8	灰 8	灰 8	
白 9	白 9	白 9	
		金 -1	
		銀 -2	




棕 黑 紅  
 $(10) \times 10^2 = 1000\Omega$   
 (1K $\Omega$ )



紅 紅 橙  
 $(22) \times 10^3 = 22000\Omega$   
 (22K $\Omega$ )



黃 紫 黑  
 $(47) \times 10^0 = 47\Omega$



橙 紅 棕  
 $(32) \times 10^1 = 320\Omega$

## digitalWrite( ) 函式 (補充資料)

Arduino 的 digitalWrite( ) 函式功能是在設定數位接腳的狀態，函式的第一個參數 pin 是定義數位接腳編號，第二個參數 value 是設定接腳的狀態，有兩種狀態：一為高態（HIGH），另一為低態（LOW）。如果所要設定的數位接腳已經由 pinMode( ) 函式設定為輸出模式，則高態電壓為 5V (或 3.3V)，低態電壓為 0V。

格式： digitalWrite(pin,value)

範例： pinMode(13,OUTPUT); //設定數位接腳 13 為輸出模式。

digitalWrite(13,HIGH); //設定數位接腳 13 輸出高態電壓。

## analogWrite( ) 函式

analogWrite( ) 函式功用是輸出脈波調變信號 (Pulse Width Modulation，簡記 PWM) 至指定接腳，頻率大約是 500Hz，可以用來在控制 LED 的亮度或是直流馬達的轉速，不需要先使用 pinMode( ) 函式去設定指定接腳為輸出模式。pin 參數設定 PWM 信號輸出腳，大多數的 Arduino 板使用 3、5、6、9、10 和 11 等接腳輸出 PWM 信號

格式： analogWrite(pin,value)

範例： analogWrite(5,127); //輸出工作週期為 50% 的 PWM 信號至接腳 5



# 實作練習

## 音樂盒實習

### □ 功能說明：

利用 Arduino 板播放兩首鋼琴入門音樂—小蜜蜂 (Little Bee) 及小星星 (Little Star)。本例使用 1 個 TACK 按鍵開關 SW，開機時預設為靜音，每按一次按鍵開關 SW，可以切換小蜜蜂 (Little Bee)、小星星 (Little Star) 及靜音等三種選擇，演奏速度為每分鐘 180 拍。

# 實作練習

□ 程式：🌐 I105.ino

```
const int speaker=2;           //數位接腳 2 連接至喇叭。
const int sw=4;                 //數位接腳 4 連接至 SW 按鍵。
const int debounce=20;         //除彈跳 20ms。
char toneName[]="CDEFGAB";     //音符表。
unsigned int frequency[7]={523,587,659,694,784,880,988}; //頻率表。
char beeTone[]="GEEFDDCDEFGGGGEEFDDCEGGEDDDDDFEFEFEFEFGGEEFDDCEGGC";
char starTone[]="CCGGAAGFFEEDDCGGFFEEDDGGFFEEDDCCGGAAGFFEEDDC";
byte beeBeat[]={1,1,2,1,1,2,1,1,1,1,1,2,      //小蜜蜂節拍。
                 1,1,2,1,1,2,1,1,1,1,4,
                 1,1,1,1,1,1,2,1,1,1,1,1,2,
                 1,1,2,1,1,2,1,1,1,1,4};
byte starBeat[]={1,1,1,1,1,1,2,1,1,1,1,1,2,    //小星星節拍。
                  1,1,1,1,1,1,2,1,1,1,1,1,2,
                  1,1,1,1,1,1,2,1,1,1,1,1,2};
unsigned long tempo=180;        //每分鐘 180 拍。
const int beeLen=sizeof(beeTone); //小蜜蜂音符總數。
const int starLen=sizeof(starTone); //小星星音符總數。
int len=0;                      //尚未演奏的剩餘音符。
int num;                        //音符指標。
int keyVal=0;                   //鍵值。
void setup()
{  pinMode(sw, INPUT_PULLUP);    //設定數位接腳 4 為輸入模式。
}
```

# 實作練習

```
void loop()  
{  
    if(digitalRead(sw)==0) //按下按鍵?  
    {  
        delay(debounce); //消除按鍵機械彈跳。  
        while(digitalRead(sw)==0) //按下按鍵?  
        ; //按鍵已放開?  
        keyVal++; //鍵值加 1。  
        if(keyVal>2) //鍵值大於 2?  
            keyVal=0; //清除鍵值。  
        num=0; //從第一個音符開始播放。  
        if(keyVal==1) //鍵值為 1?  
            len=beeLen; //取出「小蜜蜂」音符總數。  
        else if(keyVal==2) //鍵值為 2?  
            len=starLen; //取出「小星星」音符總數。  
    }  
    if(keyVal==1 && len>0) //鍵值為 1 且 len>0?  
    {    playTone(beeTone[num],beeBeat[num]); //播放一個音符。  
        num++; //下一個音符。  
        len--; //長度減 1。  
    }  
    else if(keyVal==2 && len>0) //鍵值為 2 且 len>0?  
    {    playTone(starTone[num],starBeat[num]); //播放一個音符。  
        num++; //下一個音符。  
        len--; //長度減 1。  
    }  
}
```

# 實作練習

```
void playTone(char toneNo,byte beatNo)           //播放音符函式。
{
    unsigned long duration=beatNo*60000/tempo;    //計算節拍時間(單位:毫秒)。
    int i;
    for(i=0;i<7;i++)
    {
        if(toneNo==toneName[i])                  //查音符表。
        {
            tone(speaker,frequency[i]);           //播放音符。
            delay(duration);                       //音符發音長度(節拍)。
            noTone(speaker);                       //關閉聲音。
        }
    }
}
```



## 練習

1. 設計 Arduino 程式，使用 1 個 TACK 按鍵開關 SW，控制播放兩首鋼琴入門音樂—小蜜蜂（Little Bee）及小星星（Little Star），同時使用 LED 顯示播放的音符。
2. 設計 Arduino 程式，使用 1 個 TACK 按鍵開關 SW，控制播放四首音樂，同時使用 LED 顯示播放的音符。

# 回家作業

- 使用七個按鍵開關實作一電子琴
- 有C、D、E、F、G、A、B七個音階
- 並利用另外三個按鍵開關切換成 低/中/高音模式

Hint:

```
const int toneTable[7][5]={  
    { 66, 131, 262, 523, 1046}, // C Do  
    { 74, 147, 294, 587, 1175}, // D Re  
    { 83, 165, 330, 659, 1318}, // E Mi  
    { 88, 175, 349, 698, 1397}, // F Fa  
    { 98, 196, 392, 784, 1568}, // G So  
    {110, 220, 440, 880, 1760}, // A La  
    {124, 247, 494, 988, 1976} // B Si  
};
```

※下次上課會用到LCD模組，回家可先焊接好

