

LinkIt One 實作體驗課程

國立交通大學 電子工程學系

Outline

- ▶ **Linkit One 簡介**
- ▶ 藍芽基本介紹
- ▶ 蜂鳴器介紹
- ▶ 電路圖
- ▶ **Mission1：旋律產生器(程式碼)**
- ▶ **Mission2：旋律產生器 + 藍芽(程式碼)**
- ▶ **References**

LinkIt One 簡介

- LinkIt ONE是台灣 聯發科技 (MediaTek) 所開發的產品，主要是針對物聯網和穿戴式裝置而設計的微控制器開發平台。



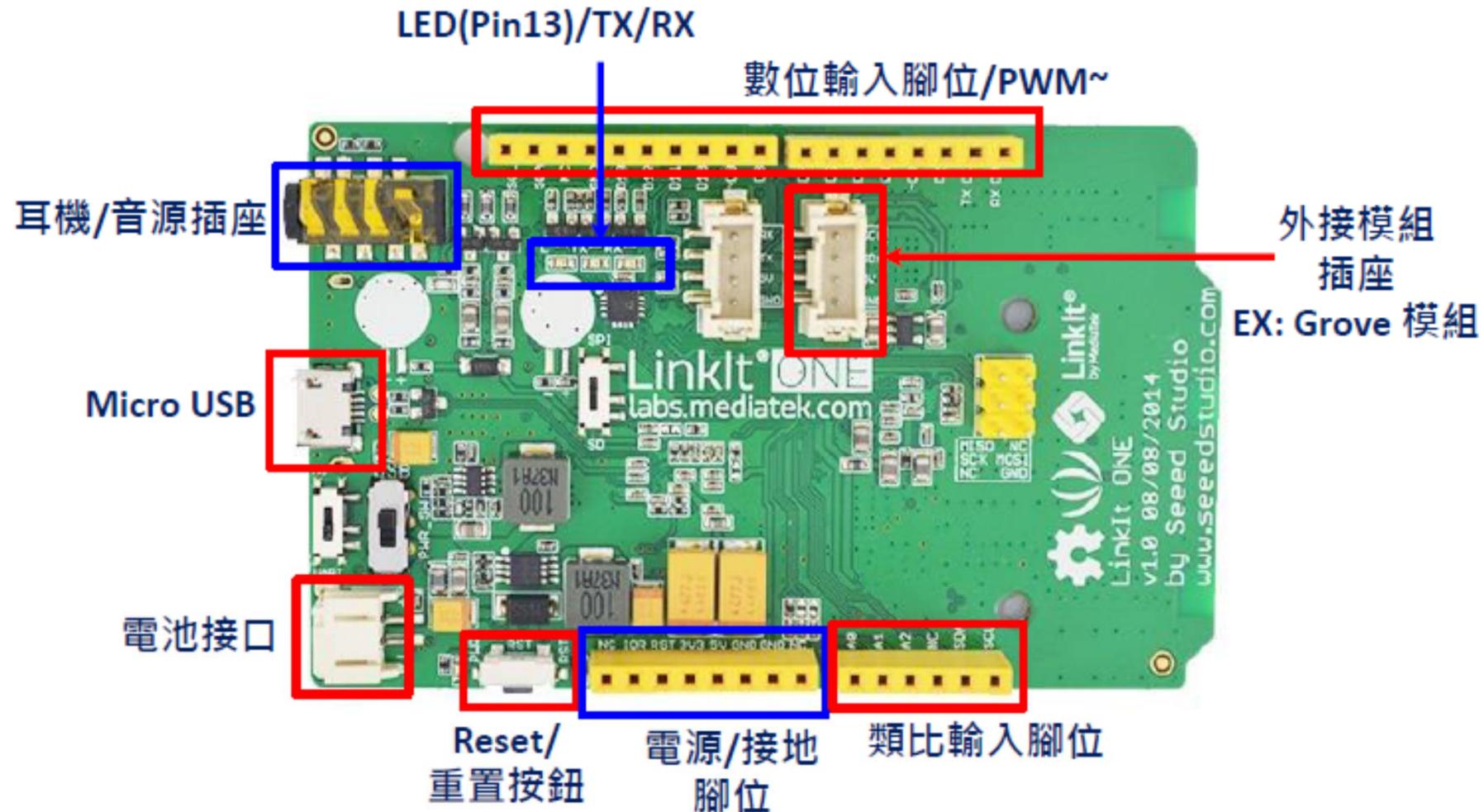
LinkIt™ ONE development
platform



特色：

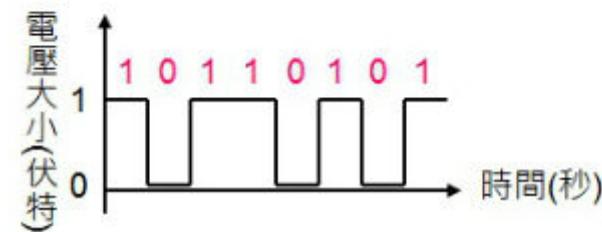
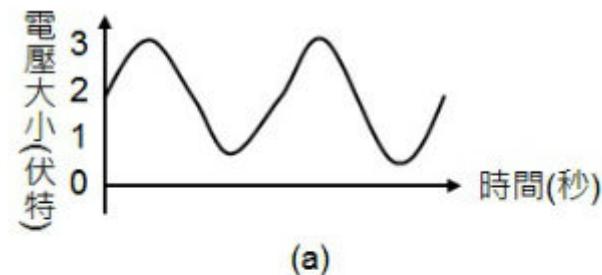
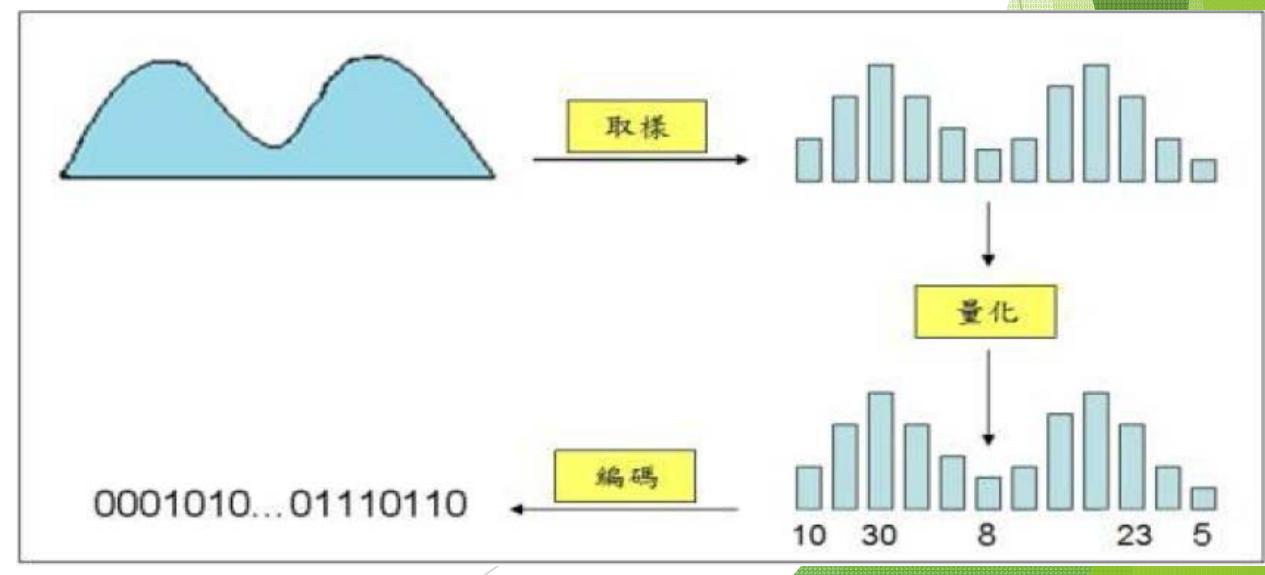
- ▶ 軟體層級與Arduino相容，因此大部分Arduino的資源皆可使用。
- ▶ 腳位配置與Arduino UNO類似，包含數位IO、類比 IO、PWM、I2C、SPI、UART與電源供應。
- ▶ 完整的通訊與多媒體功能，支援GSM、GPRS、Bluetooth EDR/BLE、SD記憶卡以及MP3/AAC音效，另外還有Wi-Fi與GPS（需另外加裝天線）。
- ▶ 開發板硬體完全開放分享，包含電路圖與腳位定義等。

Linkit One 介紹(正面)



數位訊號 與 類比訊號

Linkit One
如何讀取類比訊號？



Linkit One 介紹(背面)



藍芽基本介紹

■ 藍牙 (Bluetooth) , 是一種無線個人區域網路 (WPAN) , 最初由易利信 (Ericsson) 於1994年提出 , 後來由藍牙技術聯盟訂定技術標準 。

■ 特色 :

提供全球統一適用的、短距離的無線通訊能力

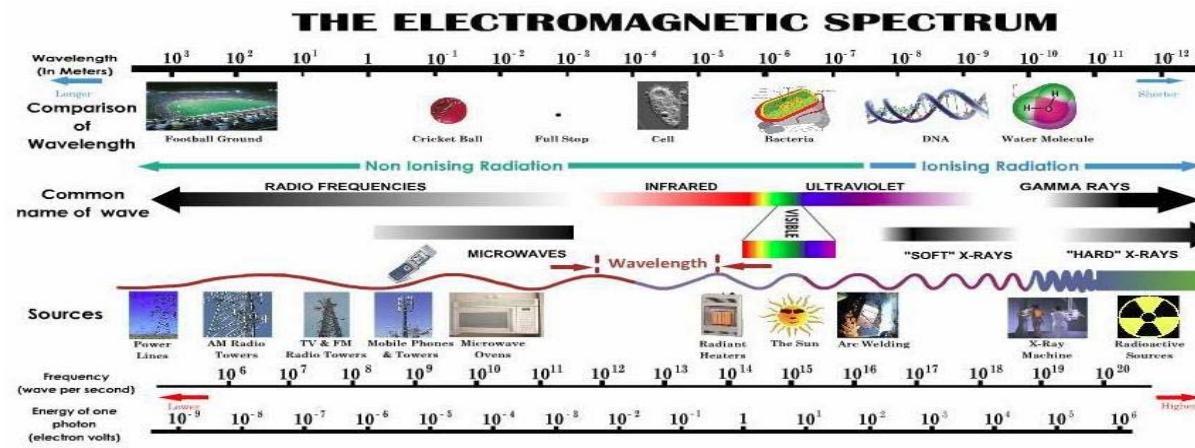
操作在2.4 GHz 頻帶，幾乎全球都毋須執照
功率低(電池運作)

主從式架構(建立連線：配對)

一般可傳送範圍(~10m)

支援多種的應用

資料、音訊、圖畫...等



Mission 1:

控制蜂鳴器發聲，讓蜂鳴器發出中音 Do(523Hz), Re (587Hz), Mi(659Hz), Fa(698Hz), So(784Hz), La(880Hz), Si(988Hz) 到高音 Do (1047Hz) 這八個不同音階的聲音。八個音分別對應到8顆LED。

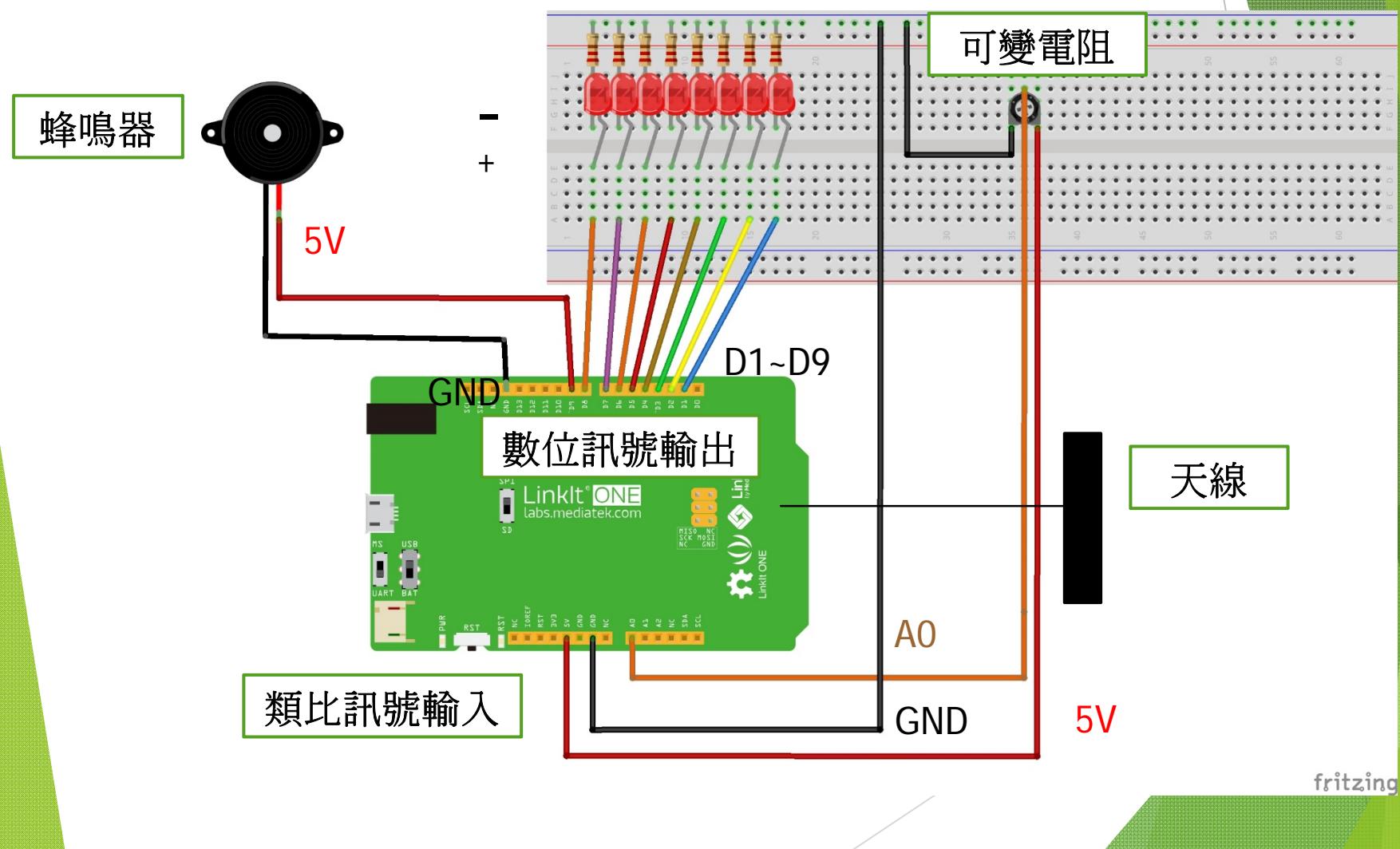
1. 將完成的程式碼燒到Linkit One 中。
2. 將想播放的 音階 儲存在 melody 陣列中。
3. 將各個音階的 節拍 儲存在 beat 陣列中。
4. 利用 可變電阻 調整 樂曲的快慢。

Mission 2:

修改Arduino 程式碼後，將 Linkit One 與 手機進行 藍芽配對。
再透過 APP 進行播放控制。

1. 將完成的程式碼燒到Linkit One 中。
2. 手機設定步驟： 設定 > 安全性 > 未知的來源 (打勾)
3. 安裝 藍芽控制App 於 Android手機中，開啟藍芽裝置 並進行配對。
4. 點選 Connect，找到並選取 對應的 藍芽裝置
5. 點選 Open 開始播放
6. 透過滑桿 改變聲音快慢
7. 點選 Close 停止播放
8. 點選 Disconnect 取消連線

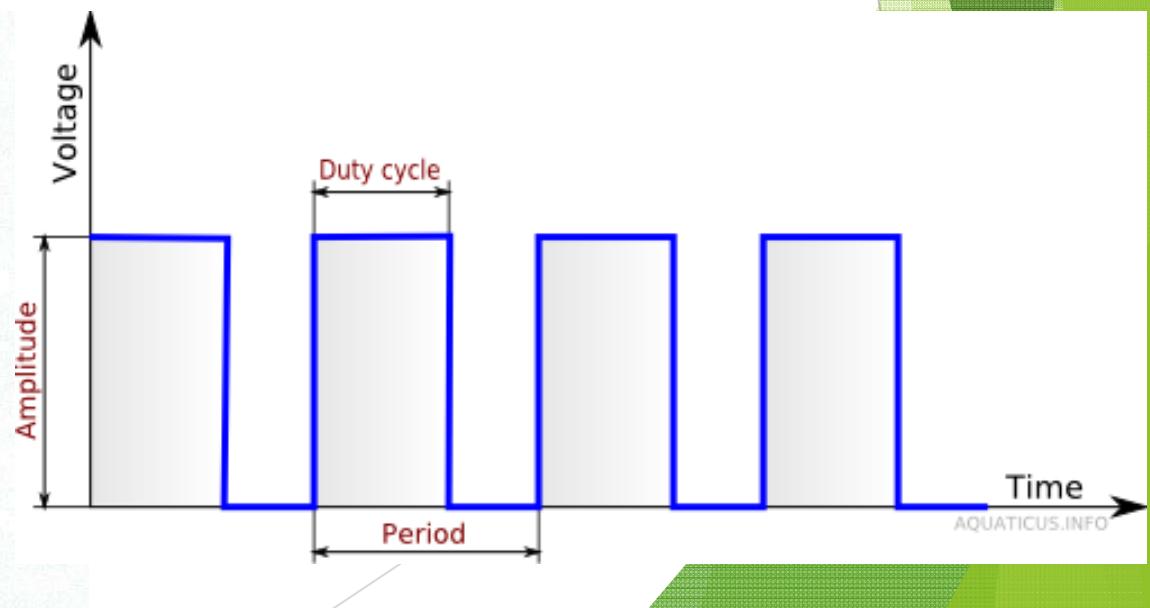
電路圖



蜂鳴器

■ 蜂鳴器

蜂鳴器接通電源後（1.5~15V直流工作電壓），裡面的多諧振蕩器起振，可以輸出1.5~2.5kHz的音頻信號。蜂鳴器發聲的原理，其實只是利用PWM產生音頻，驅動蜂鳴器，讓空氣產生振動，便能發出聲音。只要適當地改變振動頻率，就可以產生不同音階的聲音。例如，送出523Hz的脈波可以產生中音Do，587Hz的脈波可以產生中音Re，659Hz可以產生中音Mi。蜂鳴器腳位有正負方向，接電路時要特別注意。若是蜂鳴器發出的聲音太小，可以將上面的貼紙稍微撕開。



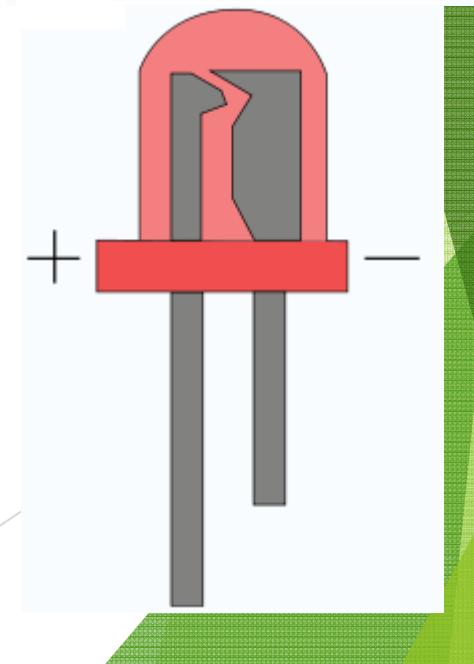
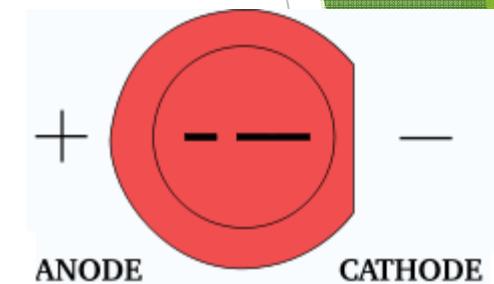
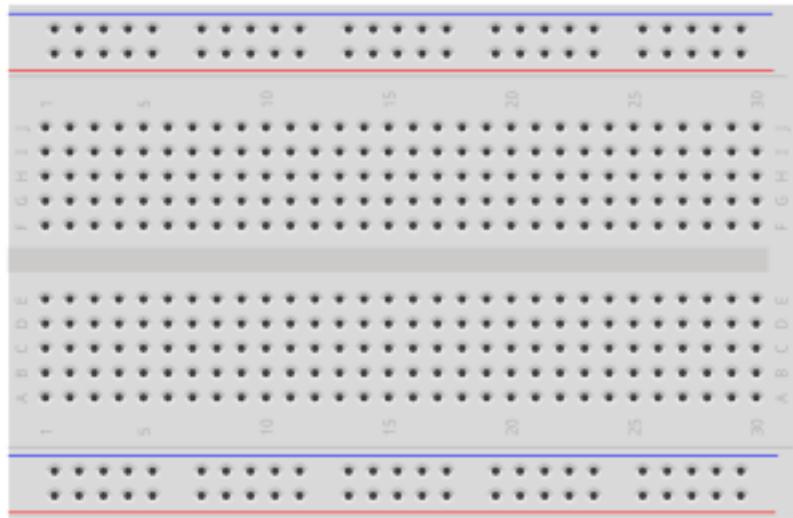
補充說明

■ 麵包板 Breadboard

麵包板可以方便將電路連接起來做初步驗證。要注意麵包板上面孔洞之間的關係，下圖紅線部分表示那排孔洞之間是相連的。

■ LED

LED叫做發光二極體，給他的兩隻腳適當的電壓差就會發光，要注意長腳要接高電位，短腳要接低電位，通常還會串接一顆電阻來調整他的亮度，沒有接電阻或是長短腳接反有可能會讓LED燒掉。



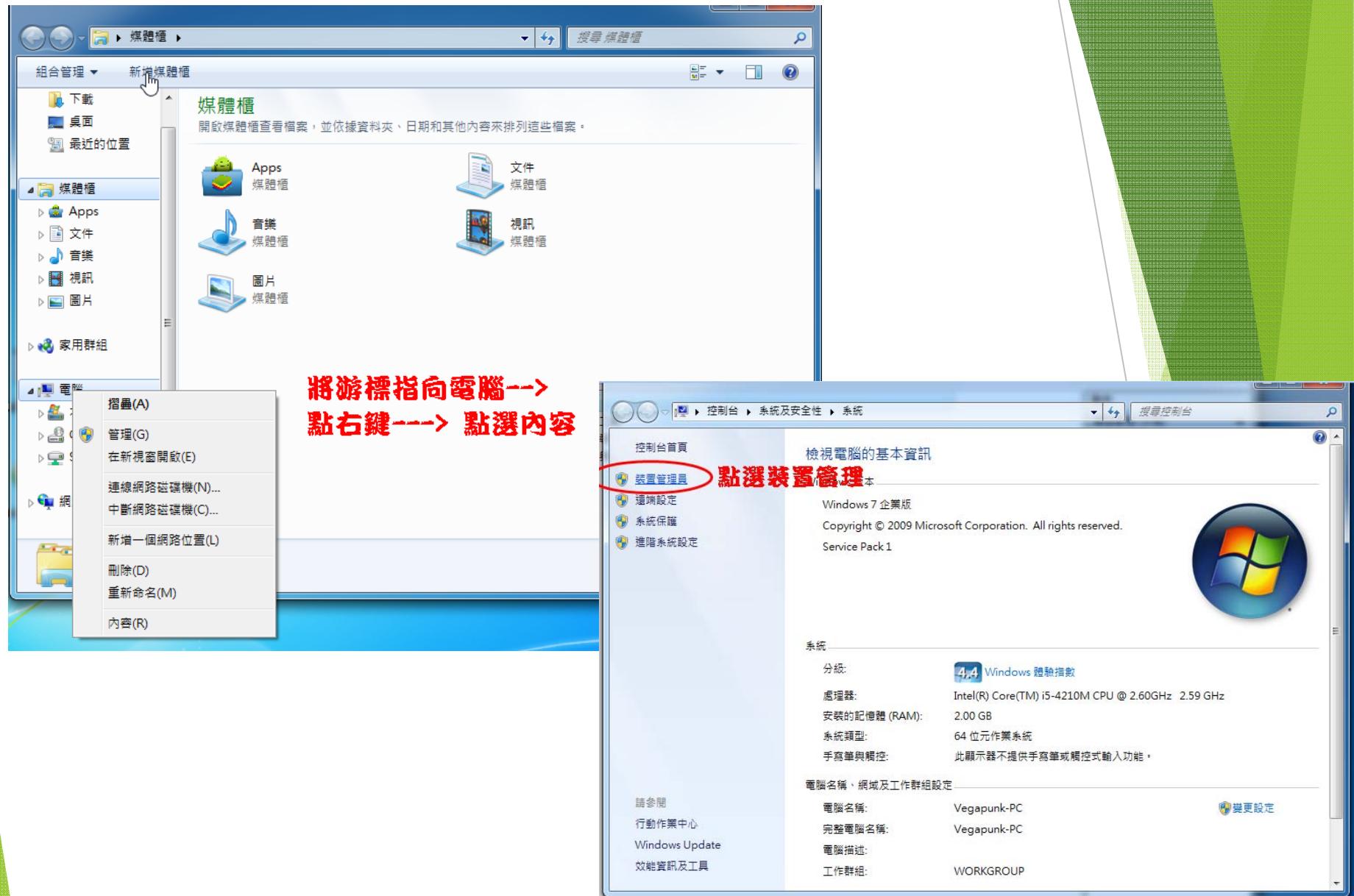
可變電阻

► **可變電阻**一種具有三個端子,其中有兩個固定接點與一個滑動接點,可經由滑動而**改變滑動端與兩個固定端間電阻值**,屬於被動元件,使用時可形成不同的分壓比率,改變滑動點的電位。最常見的用途是各式音響聲源設備裡的音量控制或電子設備裡的各式準位與功率等的控制,也可以做為位置或角度的傳感器,或者是作為鎢絲燈泡調光器或電熱絲功率調節器的控制元件等

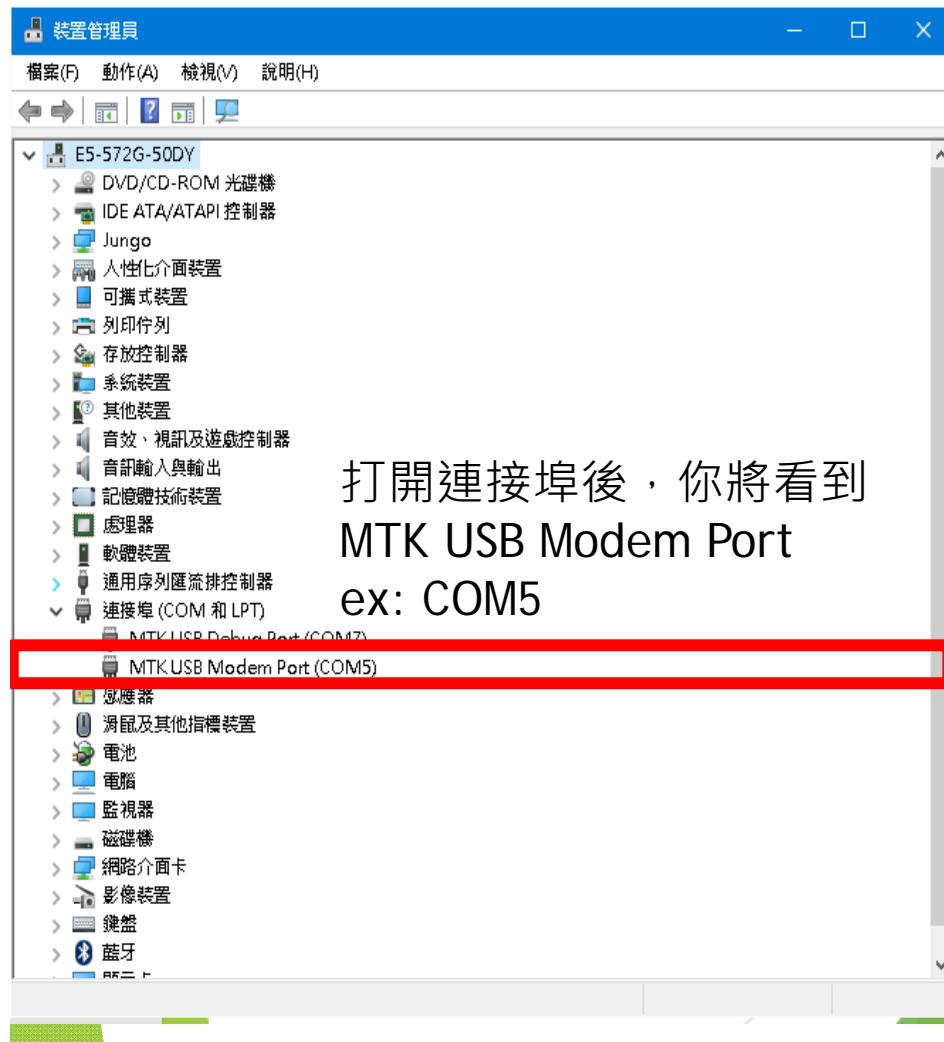


燒錄程式前的設置

確認序列埠位址



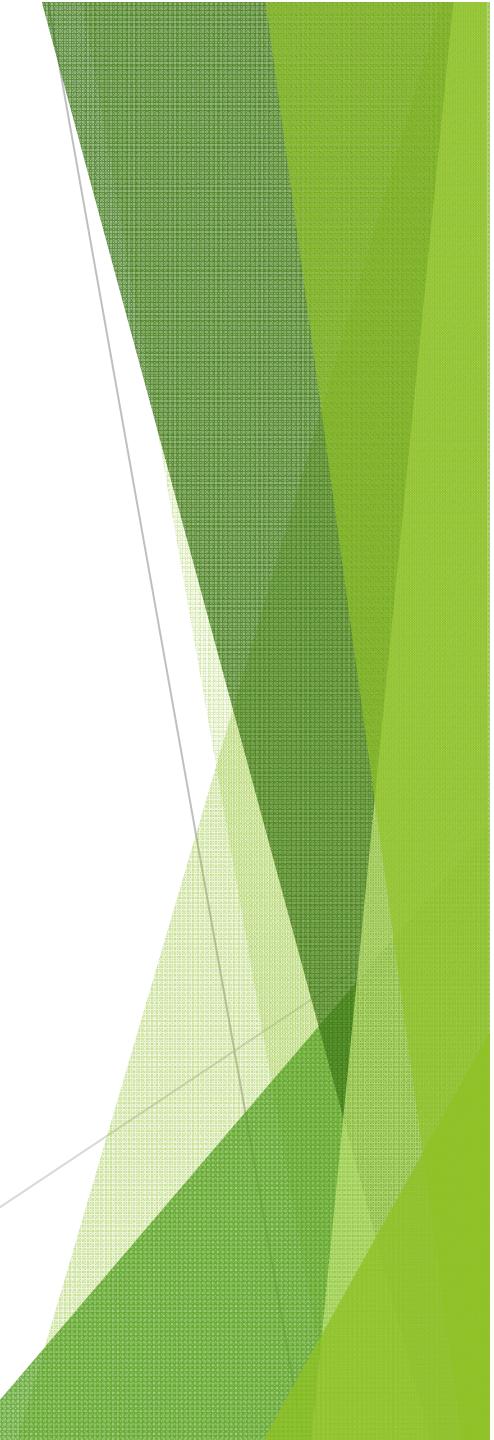
設定序列埠

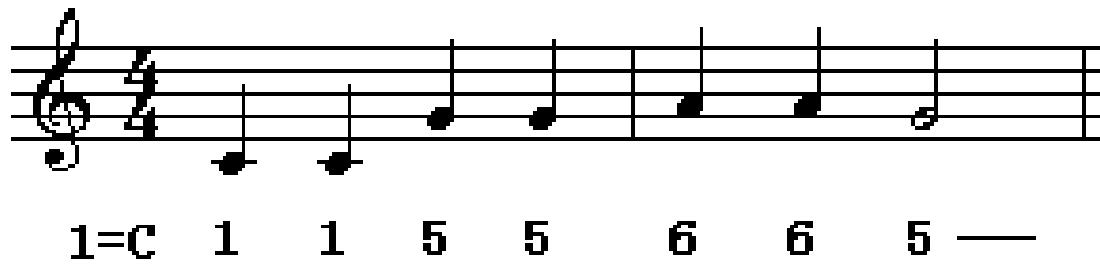


回到Arduino > 工具 > 序列埠 > 選取 對應的位址
(ex: COM5)



Mission 1





```
int melody = {1, 1, 5, 5, 6, 6, 5, -1};
```

```
float beat = {1, 1, 1, 1, 1, 1, 2};
```

Mission1：旋律產生器(程式碼)

```
//melody存每個音的音調，1~8分依序是Do, Re, ..., 高音Do, 0是休止符，最後用-1來表示結束
int melody[] = {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, -1};

//beat存每個音有幾拍
float beat[] = {1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2,
                 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2,
                 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2};

//frequency各個音對應的頻率
const int frequency[] = {1, 523, 587, 659, 698, 784, 880, 988, 1047};
const int speaker = 9; // 蜂鳴器接到D9腳位，speaker即代表D9腳位
int bpm;

void setup() {
    for (int i = 1; i <= 9; ++i) // 將連接到8個LED和蜂鳴器的D1~D9腳位設為輸出
        pinMode(i, OUTPUT);
    pinMode(A0, INPUT); // 將用來控制速度的A0腳位設為輸入
}
```

Mission1：旋律產生器(程式碼)

```
void loop() {
    for (int i = 0; melody[i] != -1; i++) {
        bpm = analogRead(A0) * 210 / 1024 + 30;           // bpm的範圍 30 ~ 240
        digitalWrite(melody[i], HIGH);                     // 讓對應的LED亮
        // 讓蜂鳴器發出聲音，每個音持續beat[i](拍) * (60000 / bpm)(毫秒/拍)毫秒
        playTone(frequency[melody[i]], (beat[i] * 60000 / bpm)-30);
        digitalWrite(melody[i], LOW);                      // 換下個音前先把上一個LED關掉
        delay(30);                                         // 延遲0.03秒以區隔兩個音
    }

    delay(2000);                                       // 放完整首歌休息兩秒
}

void playTone(int freq, int duration) { // 輸出頻率為freq的方波並持續輸出duration毫秒
    int halfPeriod = (1.0 / freq) * 1000000 / 2;
    for (long i = 0; i < duration * 1000L; i += halfPeriod * 2) {
        digitalWrite(speaker, HIGH);
        delayMicroseconds(halfPeriod);

        digitalWrite(speaker, LOW);
        delayMicroseconds(halfPeriod);
    }
}
```

上傳 與 啟動



燒錄：點選上傳

(Mission 1 的程式碼 上傳完成 後 會自動起動)



```
Linkit_BTLED_150912 | Arduino 1.6.5
檔案 編輯 草稿碼 工具 說明
Linkit_BTLED_150912
void setup() {
    for(int i=1; i<=9; ++i)
        pinMode( i, OUTPUT); // 設定連接到speaker的D8數位腳位為輸出
    Serial.begin(115200);
    while (!Serial);
    LBTServer.begin((uint8_t*)serverName);
    Serial.printf("[%s] is waiting for any client...",serverName);

    while(!LBTServer.accept(serverWaitTime)){
        Serial.print(" ... ");
    }
    Serial.println("\n====Connected====");
}

void loop() {
    char BTread;
    int speed=0;
    if(LBTServer.available()){
        BTread = LBTServer.read();
    }

    if(BTread == 'T')
    {
        Serial.println("Open");
    }
}
```

Mission 2

手機設定與安裝：

設定 > 安全性 > 未知的來源 (打勾)

設定完成後 將 NCTU_AaPaTo.apk 上傳至手機中進行安裝



Mission2：旋律產生器 + 藍芽(程式碼) (1/3)

```
#include <LBT.h>
#include <LBTSERVER.h>
char serverName[] = "My_LinkitOne";// 這裡是幫自己的LinkitOne 命名,可以取自己的英文名子加上生日
// ex: char serverName[] = "DaMingWan0127"

int melody[] = {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, -1};
float beat[] = {1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2,
1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2};

const int serverWaitTime = 5;
const int frequency[] = {1, 523, 587, 659, 698, 784, 880, 988, 1047};
const int speaker = 9;
char BTread = 0;
char bpm = 60;

void setup() {
    for (int i = 1; i <= 9; ++i)
        pinMode(i, OUTPUT);
    Serial.begin(9600);
    LBTSERVER.begin((uint8_t*)serverName);
}
```

Mission2：旋律產生器 + 藍芽(程式碼) (2/3)

```
void loop() {
    if (!LBTServer.connected()) {
        Serial.println("==Disconnected!===");
        Serial.printf("[%s] is waiting for any client...\n", serverName);
        while (!LBTServer.accept(serverWaitTime))
            Serial.println(" ... ");
        Serial.println("==Connected!===");
        BTread = 0;
    }

    for (int i = 0; melody[i] != -1; i++) {
        while (LBTServer.available() ) {
            BTread = LBTServer.read();
            if (BTread == 1)
                Serial.println("Open");
            else if (BTread == 0)
                Serial.println("Close");
            else {
                bpm = BTread;
                Serial.printf("bpm = %d\n", bpm);
            }
        } // while 迴圈 末括弧
        if (BTread == 0)
            break;

        digitalWrite(melody[i], HIGH);
        playTone(frequency[melody[i]], (beat[i] * 60000 / bpm)-30);
        digitalWrite(melody[i], LOW);
        delay(30);
    } // for 迴圈 末括弧
```

Mission2：旋律產生器 + 藍芽(程式碼) (3/3)

```
if (BTread)
    Serial.println("End of the song");
delay(2000);
} // loop() 未括弧

void playTone(int freq, int duration) {
    int halfPeriod = (1.0 / freq) * 1000000 / 2;

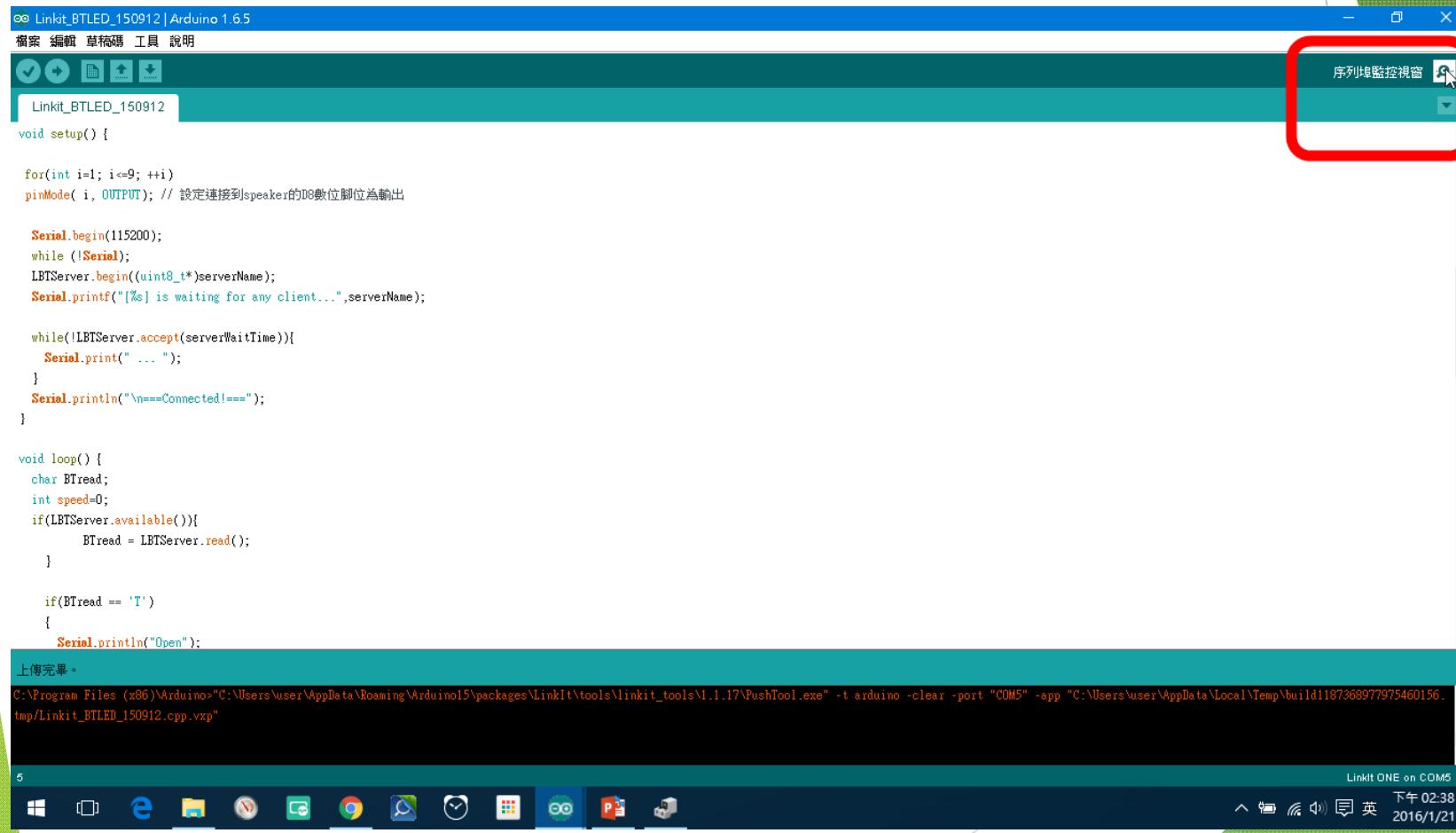
    for (long i = 0; i < duration * 1000L; i += halfPeriod * 2) {
        digitalWrite(speaker, HIGH);
        delayMicroseconds(halfPeriod);

        digitalWrite(speaker, LOW);
        delayMicroseconds(halfPeriod);
    }
}
```

上傳 與 啟動



上傳完成後 點選序列埠監視視窗



The screenshot shows the Arduino IDE interface. The title bar reads "Linkit_BTLED_150912 | Arduino 1.6.5". The menu bar includes "檔案" (File), "編輯" (Edit), "草稿碼" (Sketch), "工具" (Tools), and "說明" (Help). A toolbar with various icons is visible above the code area. The code editor contains the following sketch:

```
void setup() {
    for(int i=1; i<=9; ++i)
        pinMode( i, OUTPUT); // 設定連接到speaker的D0數位腳位為輸出
    Serial.begin(115200);
    while (!Serial);
    LBTServer.begin((uint8_t*)serverName);
    Serial.printf("[%s] is waiting for any client...",serverName);

    while(!LBTServer.accept(serverWaitTime)){
        Serial.print(" ... ");
    }
    Serial.println("\n====Connected!===");
}

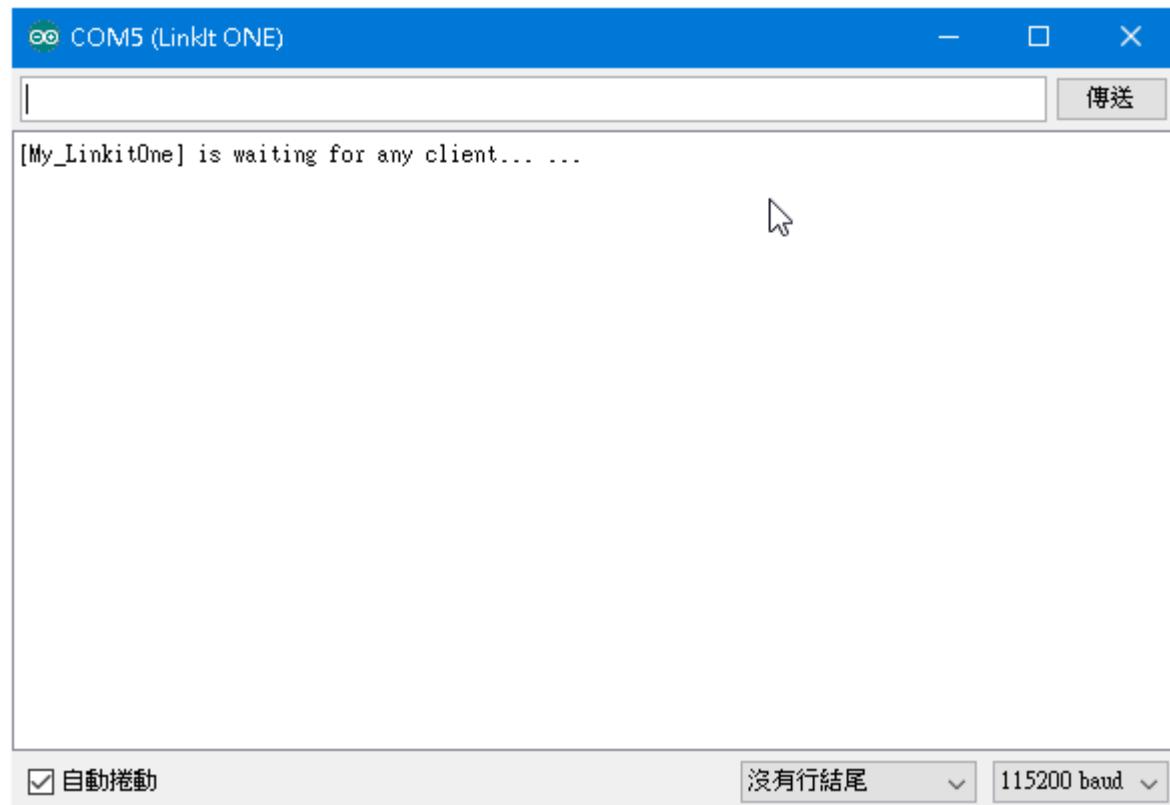
void loop() {
    char BTread;
    int speed=0;
    if(LBTServer.available()){
        BTread = LBTServer.read();
    }

    if(BTread == 'T')
    {
        Serial.println("Open");
    }
}
```

The serial monitor window at the bottom displays the message "上传完毕。" (Upload complete.) followed by the command: C:\Program Files (x86)\Arduino>C:\Users\user\AppData\Roaming\Arduino15\packages\LinkIt\tools\linkit_tools\1.1.17\PushTool.exe -t arduino -clear -port "COM5" -app "C:\Users\user\AppData\Local\Temp\build1187368977975460156.tmp\Linkit_BTLED_150912.cpp.vxp"

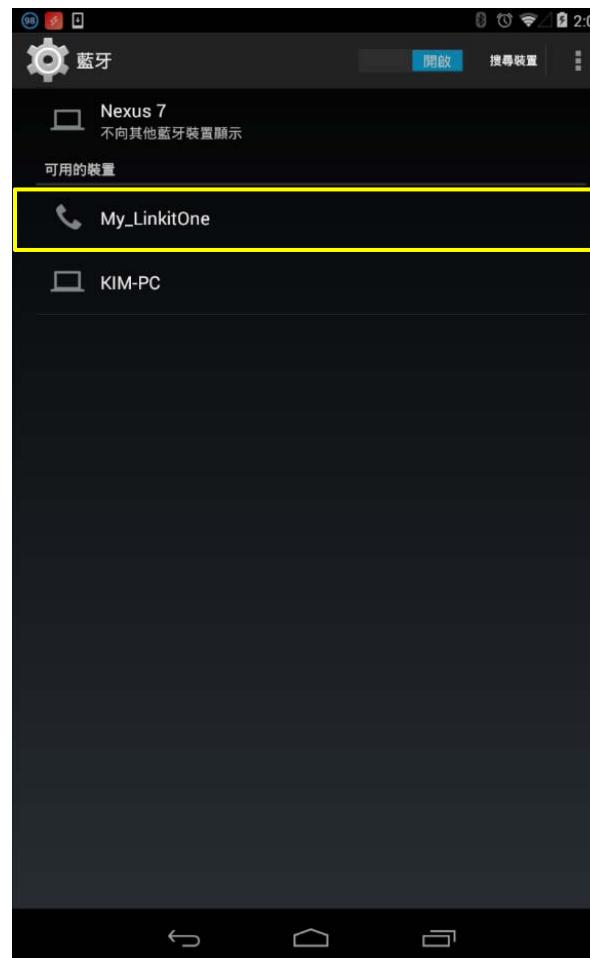
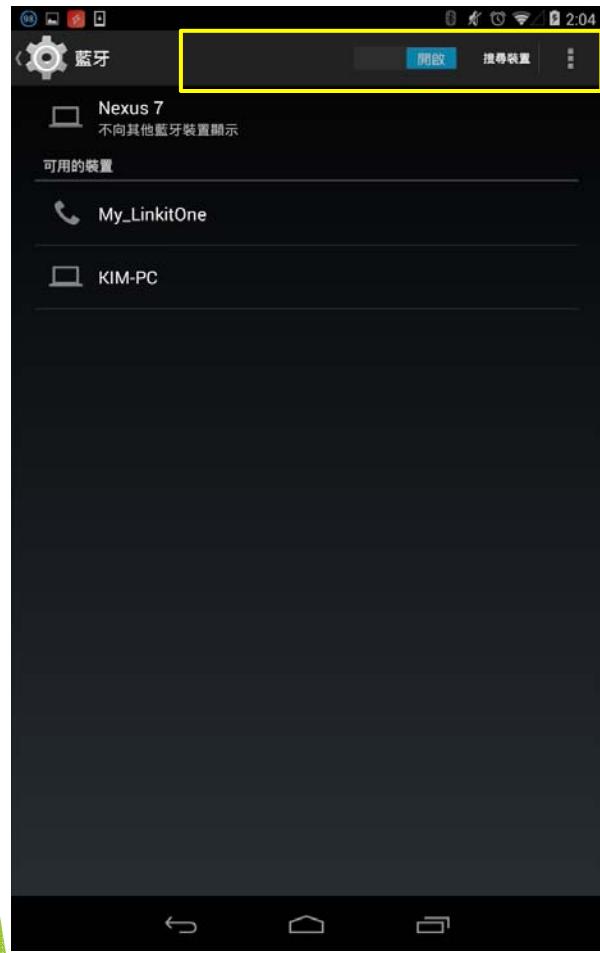
The taskbar at the bottom shows the Windows Start button, a file icon, the Microsoft Edge browser, Google Chrome, the File Explorer, the Task View icon, the Microsoft Store, the Microsoft Word icon, the Microsoft Powerpoint icon, the Microsoft Excel icon, and the Microsoft OneDrive icon. The system tray shows the date and time as "下午 02:38 2016/1/21" and the language as "英".

啟動完成



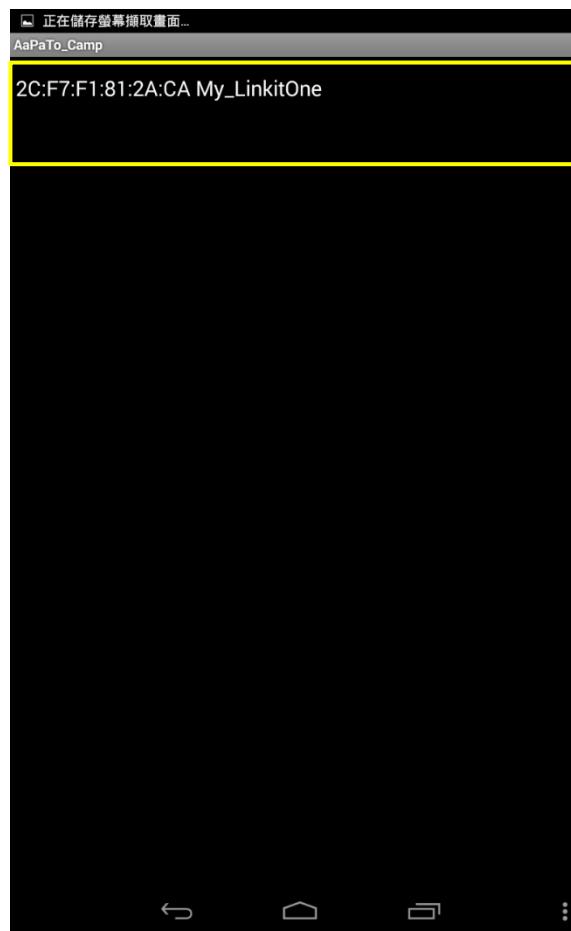
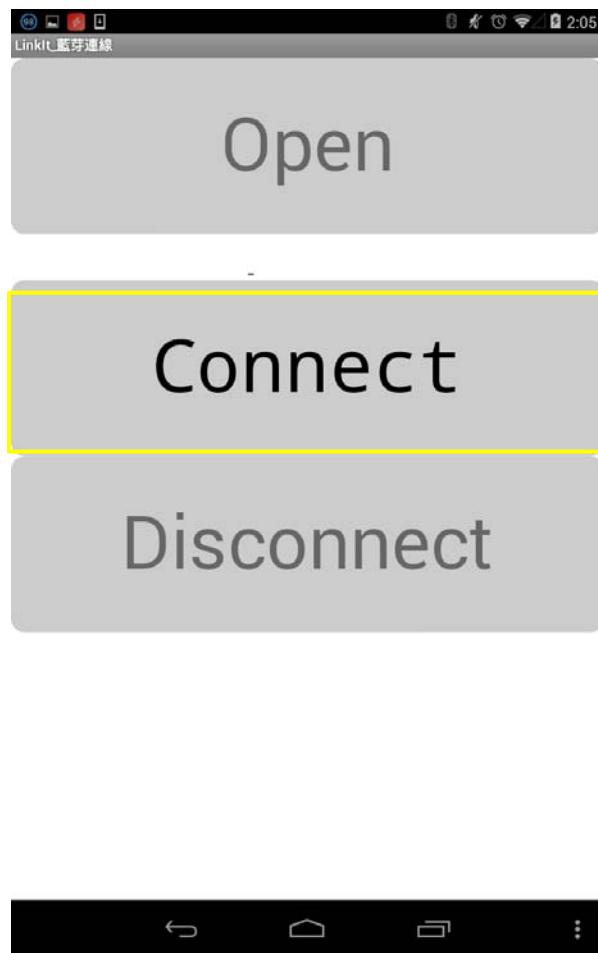
藍芽配對：

設定 > 藍芽 > 開啟 > 搜尋裝置 >
點選自己命名的LinkitOne 裝置 進行配對



啟動APP:

Connect > 點選自己命名的LinkitOne 裝置 > Open



Supplement:

■ 音頻

這邊列出了鋼琴88個鍵盤的音階，你可以自行參考使用。

B0	31	C1	33	CS1	35	D1	37	DS1	39
E1	41	F1	44	FS1	46	G1	49	GS1	52
A1	55	AS1	58	B1	62	C2	65	CS2	69
D2	73	DS2	78	E2	82	F2	87	FS2	93
G2	98	GS2	104	A2	110	AS2	117	B2	123
C3	131	CS3	139	D3	147	DS3	156	E3	165
F3	175	FS3	185	G3	196	GS3	208	A3	220
AS3	233	B3	247	C4	262	CS4	277	D4	294
DS4	311	E4	330	F4	349	FS4	370	G4	392
GS4	415	A4	440	AS4	466	B4	494	C5	523
CS5	554	D5	587	DS5	622	E5	659	F5	698
FS5	740	G5	784	GS5	831	A5	880	AS5	932
B5	988	C6	1047	CS6	1109	D6	1175	DS6	1245
E6	1319	F6	1397	FS6	1480	G6	1568	GS6	1661
A6	1760	AS6	1865	B6	1976	C7	2093	CS7	2217
D7	2349	DS7	2489	E7	2637	F7	2794	FS7	2960
G7	3136	GS7	3322	A7	3520	AS7	3729	B7	3951
C8	4186	CS8	4435	D8	4699	DS8	4978		

References

- **Arduino Website**
[\(http://arduino.cc/en/\)](http://arduino.cc/en/)
- **Arduino Language Reference**
[\(http://arduino.cc/en/Reference/HomePage\)](http://arduino.cc/en/Reference/HomePage)
- **Cooper Maa Arduino 筆記**
[\(http://coopermaa2nd.blogspot.tw/2010/12/arduino-lab6.html\)](http://coopermaa2nd.blogspot.tw/2010/12/arduino-lab6.html)
- **交大電子工程系 LinkitOne 課程講義 by Alpha Juan**
[\(http://www.alab.ee.nctu.edu.tw/wpmu/elab/linkit-one\)](http://www.alab.ee.nctu.edu.tw/wpmu/elab/linkit-one)