

ARDUINO藍芽連線

範例與函式庫下載

- <https://github.com/silencecork/AndroidBluetoothWorkshop2014>

ArduinoLEDController/ino/bt_led.ino

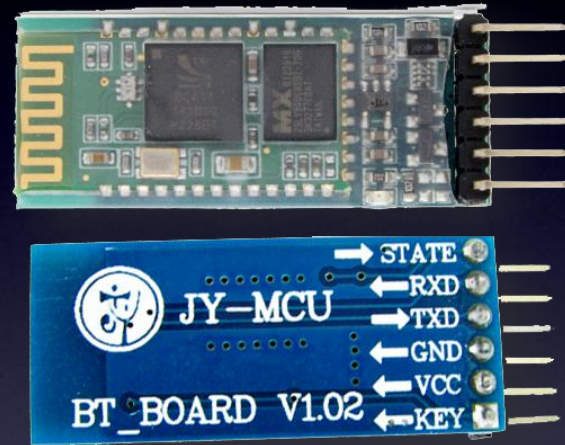
藍芽開關ARDUINO連接之LED

範例說明

- 以手機的藍芽程式控制Arduino Leonardo上的LED開關

藍芽HC-05

- 很普遍的藍芽晶片
- RXD為接收針腳
- TXD為傳送針腳
- GND接地線
- VCC電源
 - VCC33 – 3.3v
 - VCC50 – 5.0v
- KEY 寫入高電壓表示讓晶片進入AT Command模式

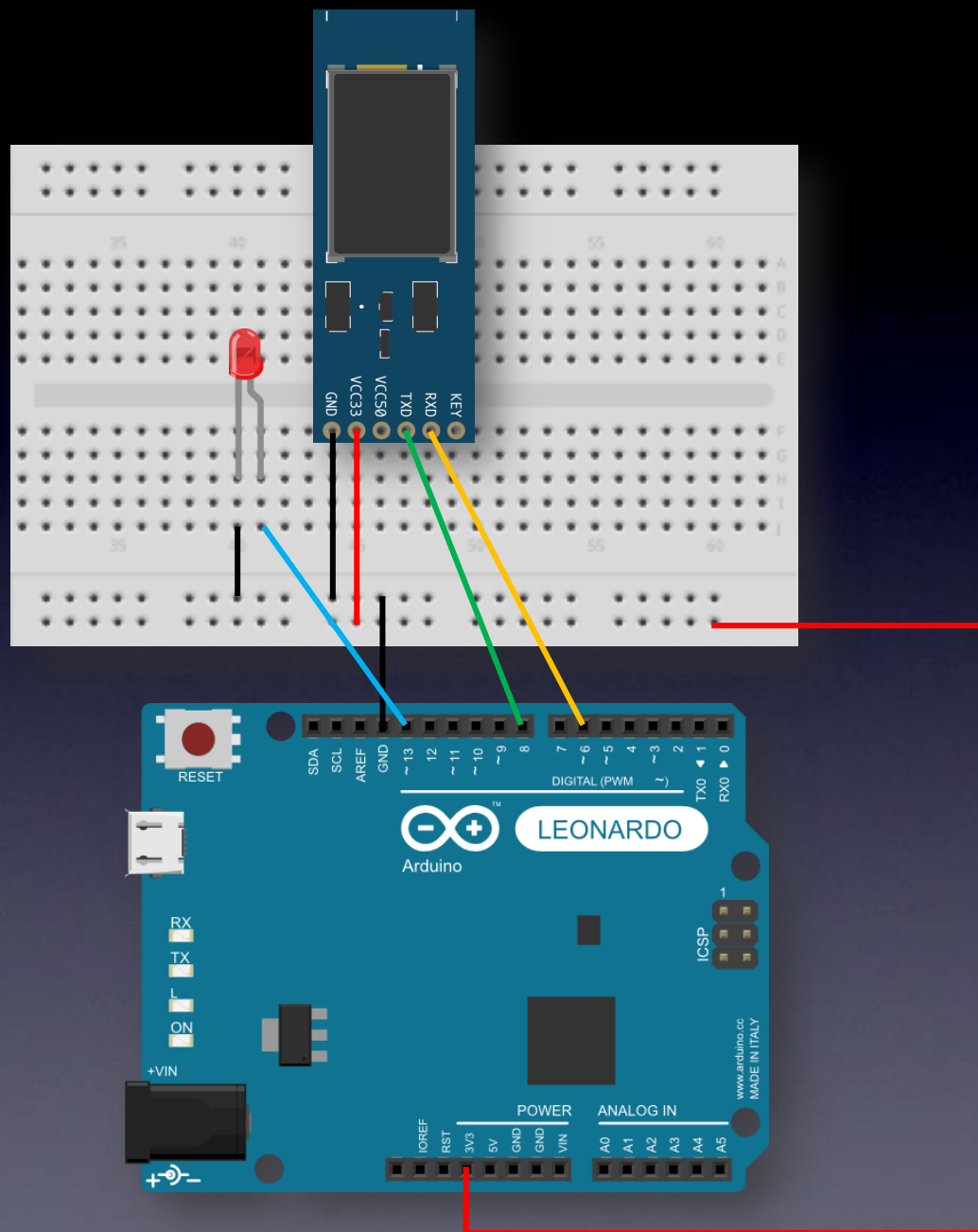


LED

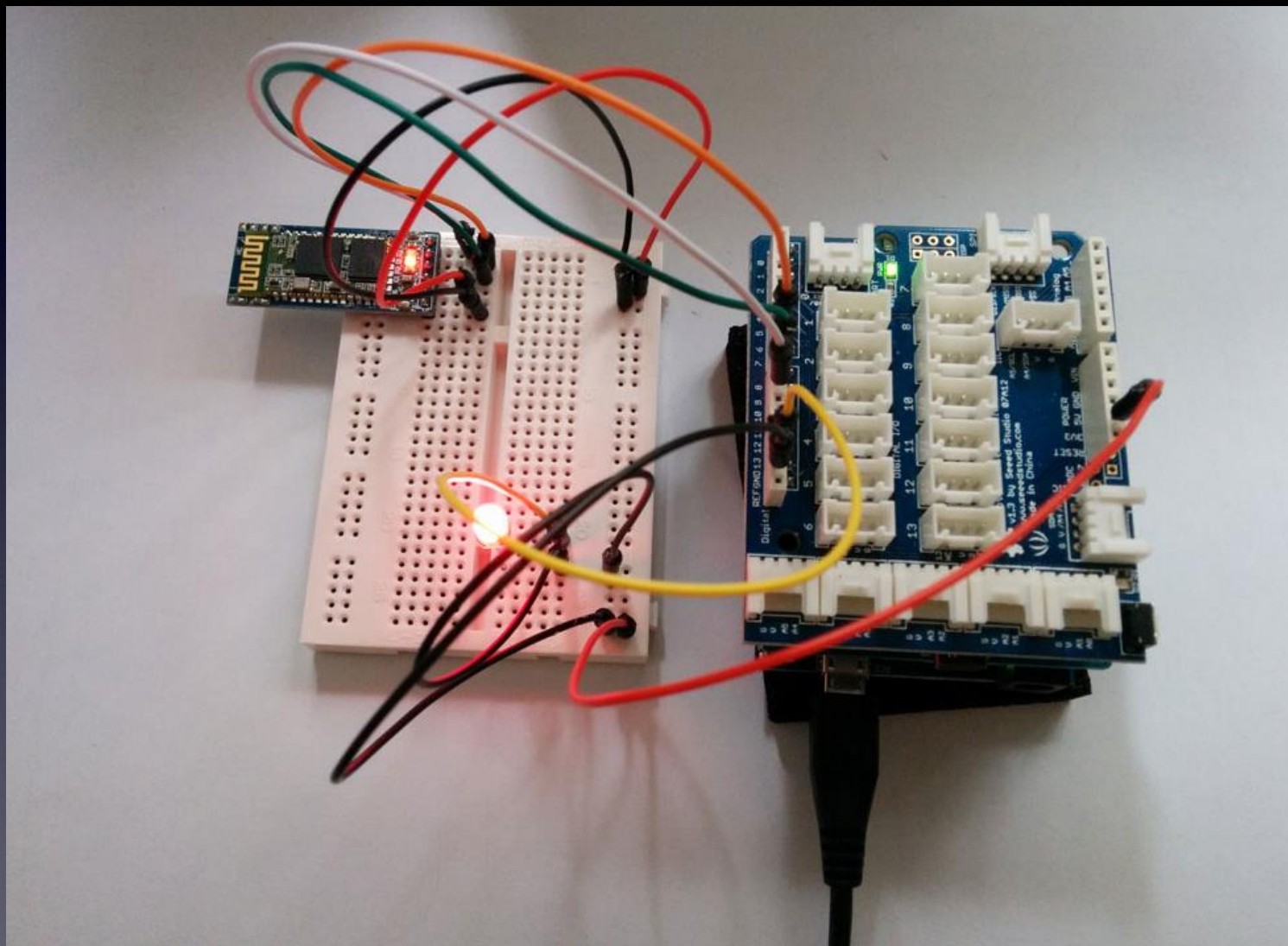
- 一般常見的3mm LED
- 長的針腳接Arduino任一數位輸出孔
- 短的針腳接接地線



電路圖



實機圖



ARDUINO程式碼

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial blueToothSerial(RX, TX);
```

- 建立軟體的Serial，負責與藍芽晶片連結
- RX必須要填藍芽晶片TXD所連結的針腳
- TX必須要填藍芽晶片RXD所連結的針腳

ARDUINO程式碼

```
void setup() {  
    pinMode(LED, OUTPUT);  
    blueToothSerial.begin(38400);  
}
```

- setup()
 - 為Arduino定義，程式啟動時會先呼叫一次的function
 - 可以用來處理硬體初始化的動作
- pinMode()
 - 指定數位輸出針腳的作用
 - 參數1：針腳編號
 - 參數2：INPUT或OUTPUT

ARDUINO程式碼

```
void setup() {  
    pinMode(LED, OUTPUT);  
    blueToothSerial.begin(38400);  
}
```

- 啟動與藍芽連結的Serial
 - 參數為baud rate，大部分預設是38400
 - 需要參閱晶片的說明文件
 - 也可以使用AT Command: AT+UART?\r\n查詢

ARDUINO程式碼

```
void loop() {  
    if(blueToothSerial.available()){  
        .....  
    }  
    delay(500);  
}
```

- loop()
 - 為Arduino定義的程式執行區，只要當程式執行，loop將不停被呼叫
- bluetoothSerial.available()
 - 當藍芽連接後有傳送資料，available就會是true
- delay() 讓程式暫停一下子，參數是暫停的微秒

ARDUINO程式碼

```
if(blueToothSerial.available()){  
    char c = blueToothSerial.read();  
    if (c == '1') {  
        digitalWrite(LED, HIGH);  
    }  
    else {  
        digitalWrite(LED, LOW);  
    }  
}
```

- read()
 - 讀取出藍芽所接收到的內容，讀取出來的皆是char

ARDUINO程式碼

```
if(blueToothSerial.available()){  
    char c = blueToothSerial.read();  
    if (c == '1') {  
        digitalWrite(LED, HIGH);  
    }  
    else {  
        digitalWrite(LED, LOW);  
    }  
}
```

- digitalWrite()
 - 針對數位針腳的控制
 - 參數1：針腳編號
 - 參數2：**HIGH**就會輸出電壓、**LOW**就會停止輸出電壓

ArduinoLEDController

ANDROID端送出控制訊號

ANDROID送出控制

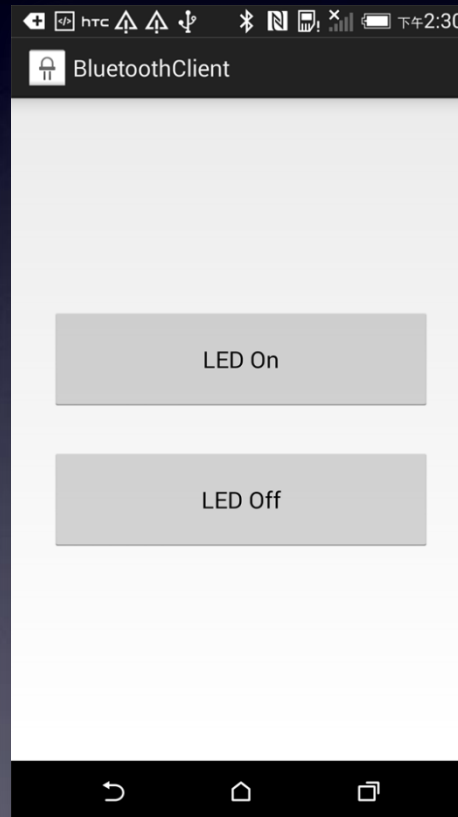
- 這部分延續上一章「基本藍芽連線」，只是介面稍作修改

ANDROID送出控制

- 本專案有兩個Activity
- ClientConnectionActivity
 - 負責與Arduino裝置連線送出控制代碼0或1
- DiscoveryActivity
 - 負責找尋附近的裝置
 - HC-05藍芽晶片基本上搜尋都是開啟的
配對的預設密碼為1234

ANDROID送出控制

- 介面更改為兩個大型按鈕，分別是ON與OFF



ANDROID送出控制

- 重點提示
- 使用BluetoothConnectionHelper.createClient()
 - UUID要使用00001101-0000-1000-8000-00805F9B34FB
- 使用OnBluetoothMessageListener接收訊息
 - 只需呼叫BluetoothConnectionHelper的setMessageReceiver()
- 在送出訊息的部分，使用BluetoothConnectionHelper.sendMessage()送出字串0或1即可

ArduinoSensorReceiver/ino/bt_lm35.ino

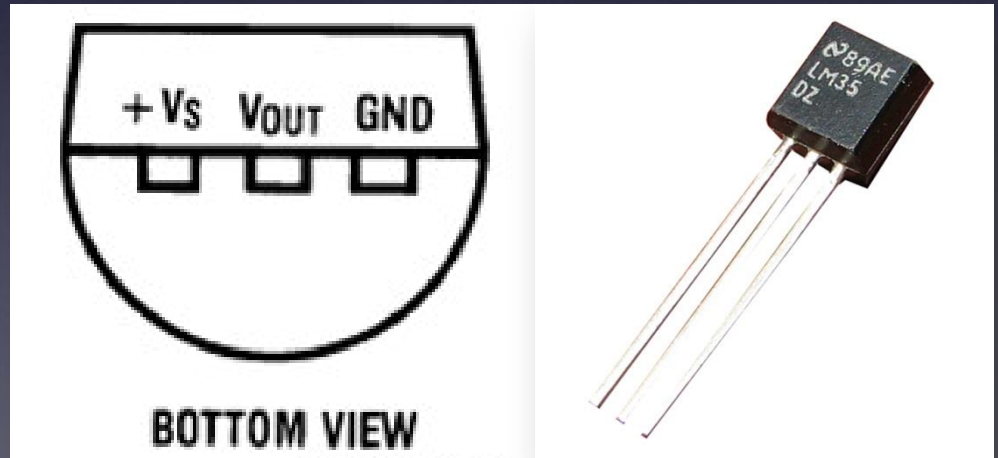
ARDUINO連接藍芽與溫度傳感器

範例說明

- 藉由Arduino Leonardo與LM35溫度感測器
- 藍芽HC-05負責將溫度感測的結果傳送給目標裝置

LM35溫度感測器

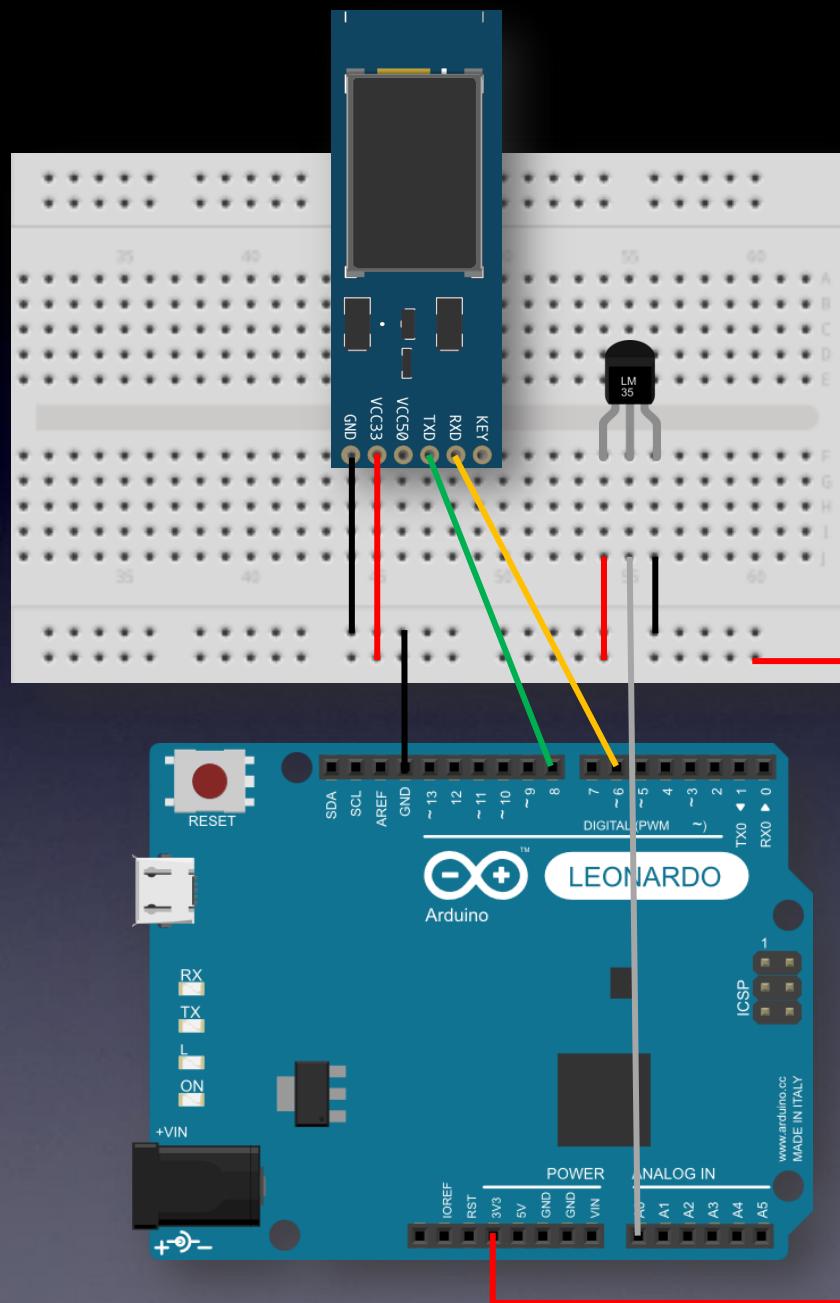
- LM35是一款滿準確的溫度感測裝置
- 感測範圍 0° - 100°
- 接受電壓3.3v
- VS電源
- Vout資料傳出針腳
- GND接地線



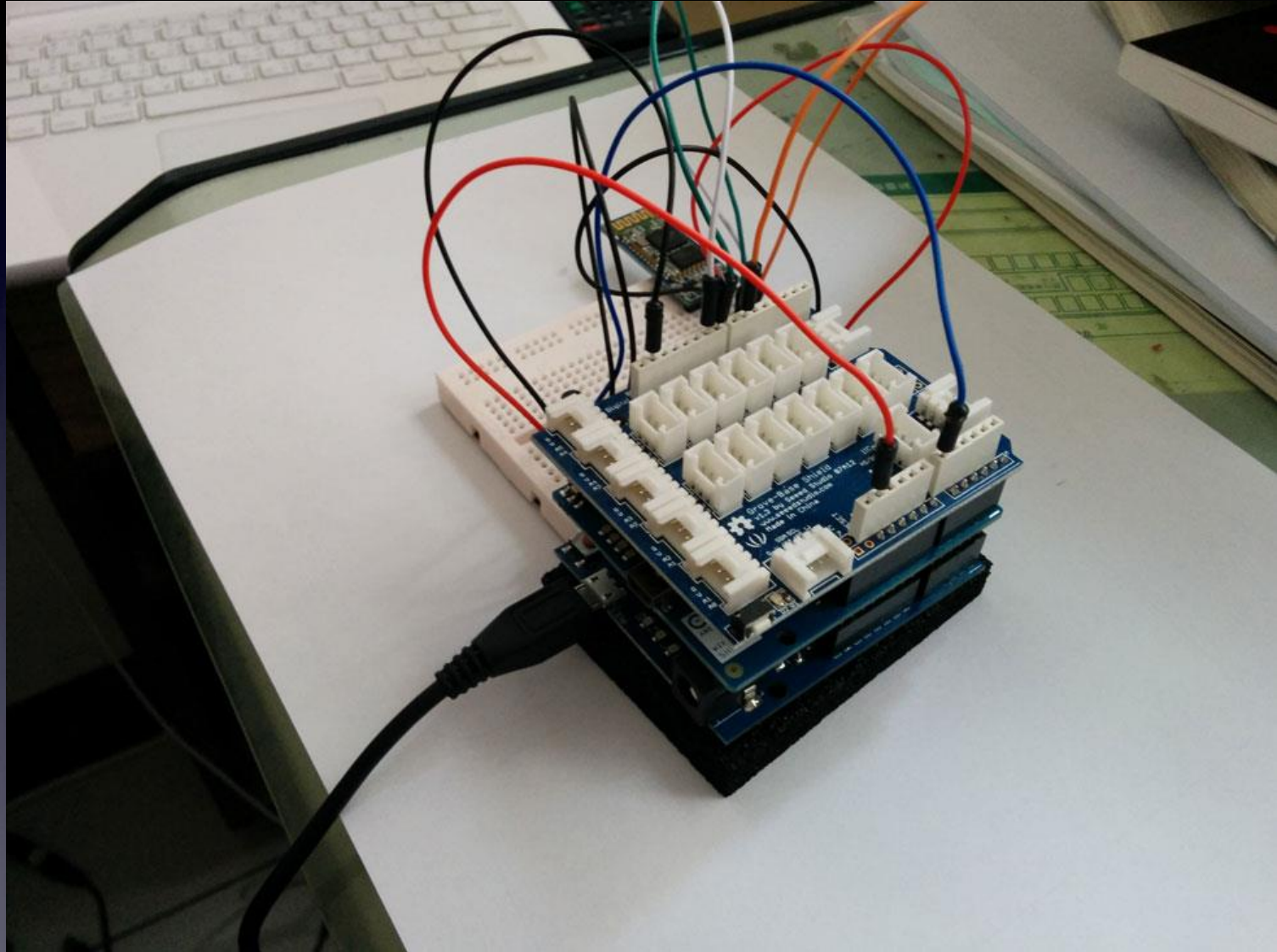
LM35溫度感測器

- 溫度每升高一度電壓輸出提高10mv
- 經由Arduino可以藉由固定公式換算目前攝氏
 - $(5.0 * \text{讀取到的LM35值} * 100.0) / 1024.0$

電路圖



實機圖



ARDUINO程式碼

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial blueToothSerial(RX, TX);
```

- 建立軟體的Serial，負責與藍芽晶片連結
- RX必須要填藍芽晶片TXD所連結的針腳
- TX必須要填藍芽晶片RXD所連結的針腳

ARDUINO程式碼

```
void setup() {  
    blueToothSerial.begin(38400);  
}
```

- setup()
 - 為Arduino定義，程式啟動時會先呼叫一次的function
 - 可以用來處理硬體初始化的動作
- 啟動與藍芽連結的Serial
 - 參數為baud rate，大部分預設是38400
 - 需要參閱晶片的說明文件
 - 也可以使用AT Command: AT+UART?\r\n查詢

ARDUINO程式碼

```
void loop() {  
    if(blueToothSerial.available()){  
        .....  
    }  
    delay(500);  
}
```

- loop()
 - 為Arduino定義的程式執行區，只要當程式執行，loop將不停被呼叫
- bluetoothSerial.available()
 - 當藍芽連接後有傳送資料，available就會是true
- delay() 讓程式暫停一下子，參數是暫停的微秒

ARDUINO程式碼

```
if(blueToothSerial.available()){  
    blueToothSerial.read();  
    int val = analogRead(SENSOR);  
    int dat=(5.0 * val * 100.0)/1024.0;  
    blueToothSerial.print(String(dat, DEC));  
    blueToothSerial.println("C");  
}
```

- `bluetoothSerial.read()`
 - 讀取出傳送來的內容
- `analogRead()`
 - 讀取類比port的值，這裡是讀取LM35晶片的值
 - 參數帶入port number

ARDUINO程式碼

```
if(blueToothSerial.available()){  
    blueToothSerial.read();  
    int val = analogRead(SENSOR);  
    int dat=(5.0 * val * 100.0)/1024.0;  
    blueToothSerial.print(String(dat, DEC));  
    blueToothSerial.println("C");  
}
```

- 這是將LM35傳感器傳出的值轉換成攝氏的公式
 - $(5.0 * \text{讀取到的LM35值} * 100.0) / 1024.0$

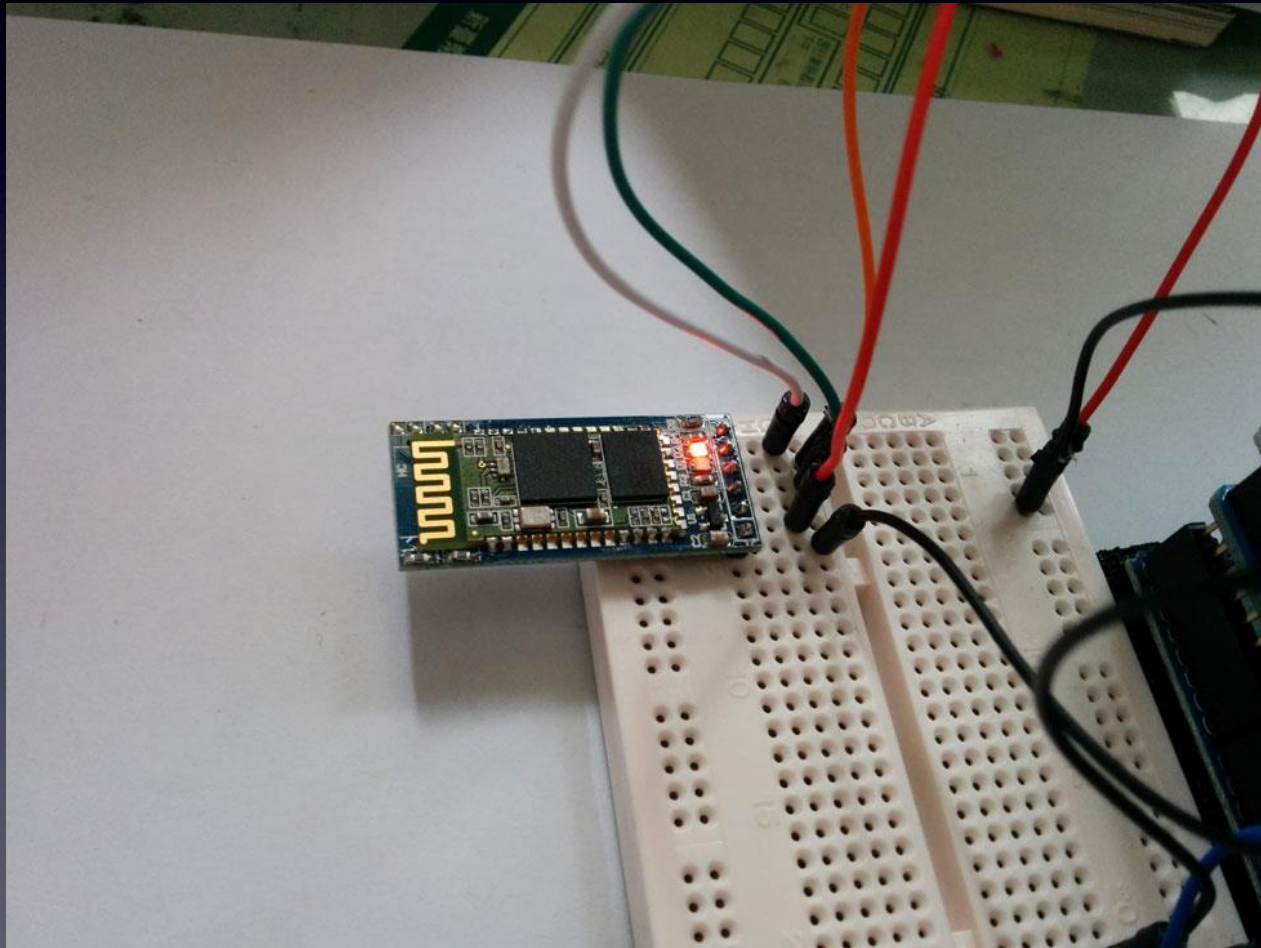
ARDUINO程式碼

```
if(blueToothSerial.available()){  
    blueToothSerial.read();  
    int val = analogRead(SENSOR);  
    int dat=(5.0 * val * 100.0)/1024.0;  
    blueToothSerial.print(String(dat, DEC));  
    blueToothSerial.println("C");  
}
```

- String(dat, DEC)
 - 將數字轉成十進位的字串
- bluetoothSerial.print(), println()
 - 將指定內容藉由藍芽Serial傳送出去

ARDUINO程式執行

- 藍芽等待連線時，會不停地閃爍LED紅燈



ArduinoSensorReceiver

ANDROID端接收SENSOR資料

ANDROID接收資料

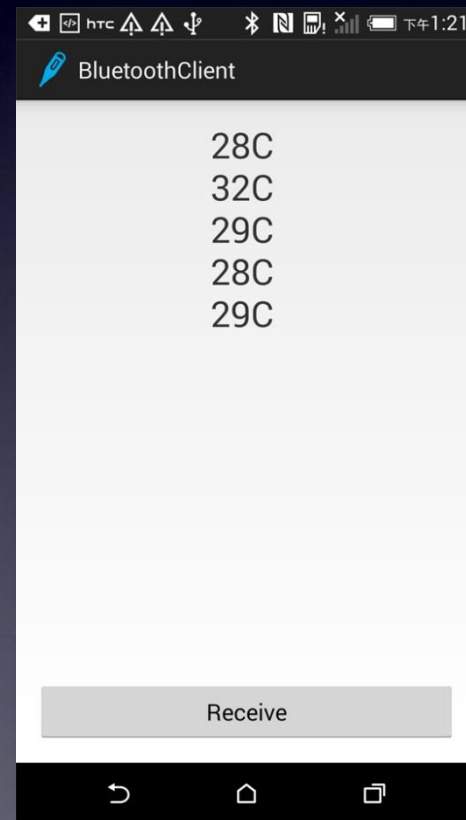
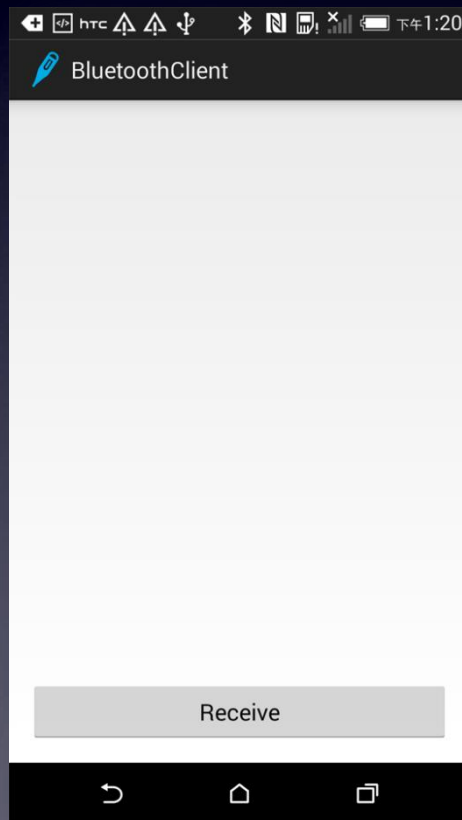
- 這部分延續上一章「基本藍芽連線」，只是介面稍作修改

ANDROID接收資料

- 本專案有兩個Activity
- ClientConnectionActivity
 - 負責與Arduino裝置連線
- DiscoveryActivity
 - 負責找尋附近的裝置
 - HC-05藍芽晶片基本上搜尋都是開啟的
配對的預設密碼為1234

ANDROID接收資料

- 本畫面改為按下Receive，就會向Arduino傳送一個字元，而Arduino就會回傳溫度



ANDROID接收資料

- 重點提示
- 使用BluetoothConnectionHelper.createClient()
 - UUID要使用00001101-0000-1000-8000-00805F9B34FB
- 使用OnBluetoothMessageListener接收訊息
 - 只需呼叫BluetoothConnectionHelper的setMessageReceiver()

更多資料

- 切換HC-05至AT Command模式
 - <http://www.techbitar.com/modify-the-hc-05-bluetooth-module-defaults-using-at-commands.html>