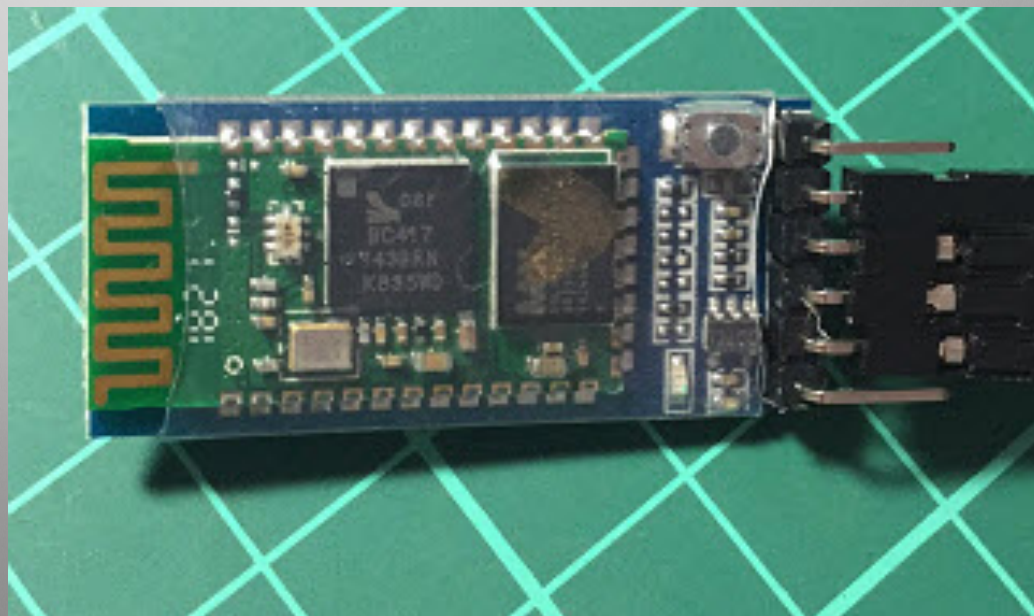
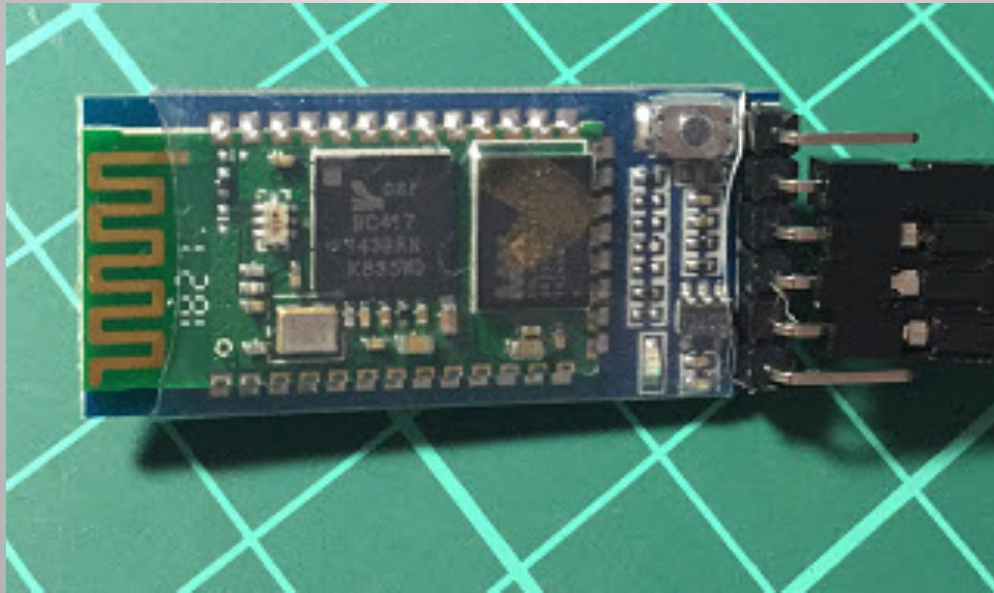


# HC-05 藍芽模組





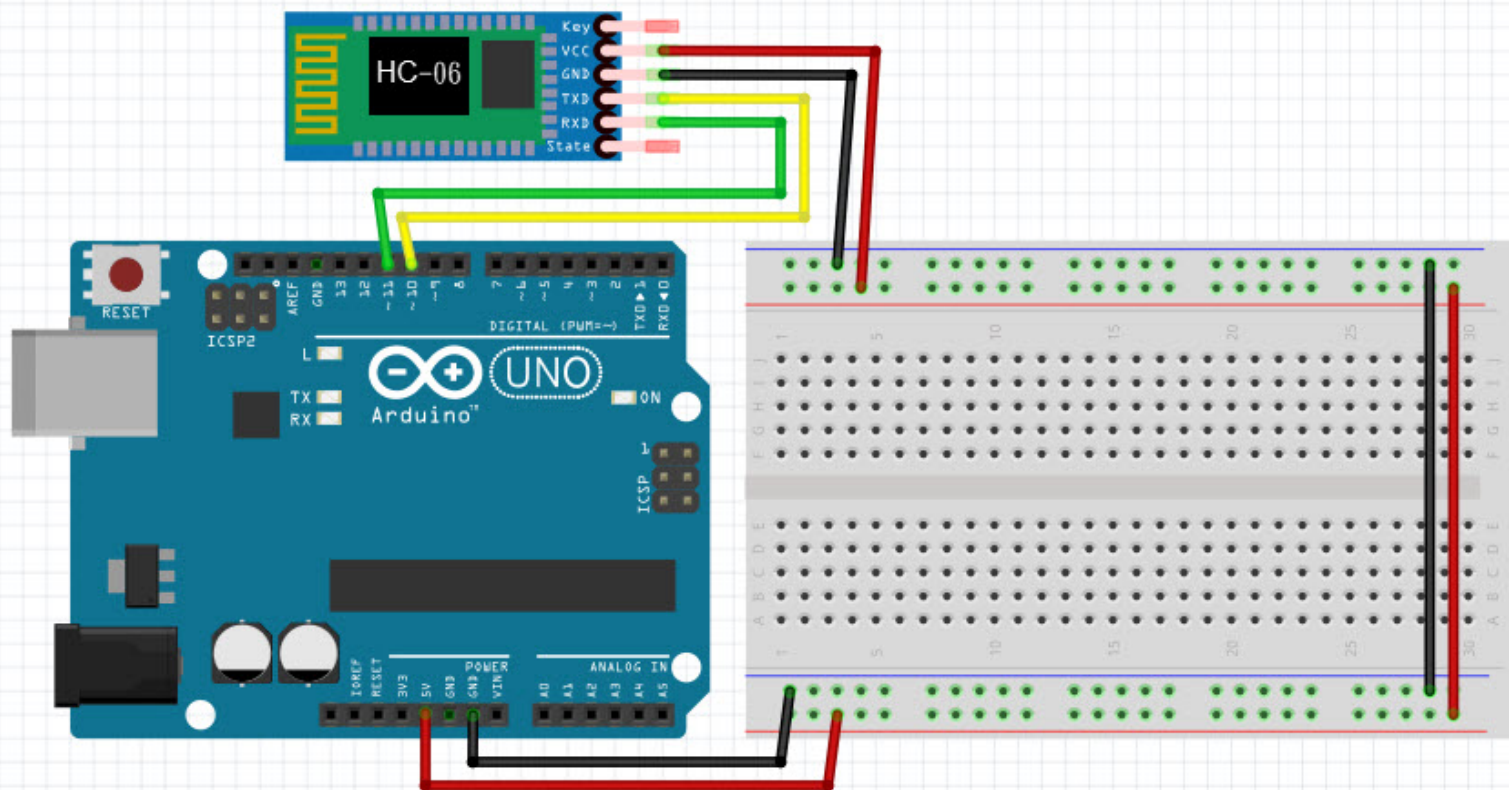


# HC-05

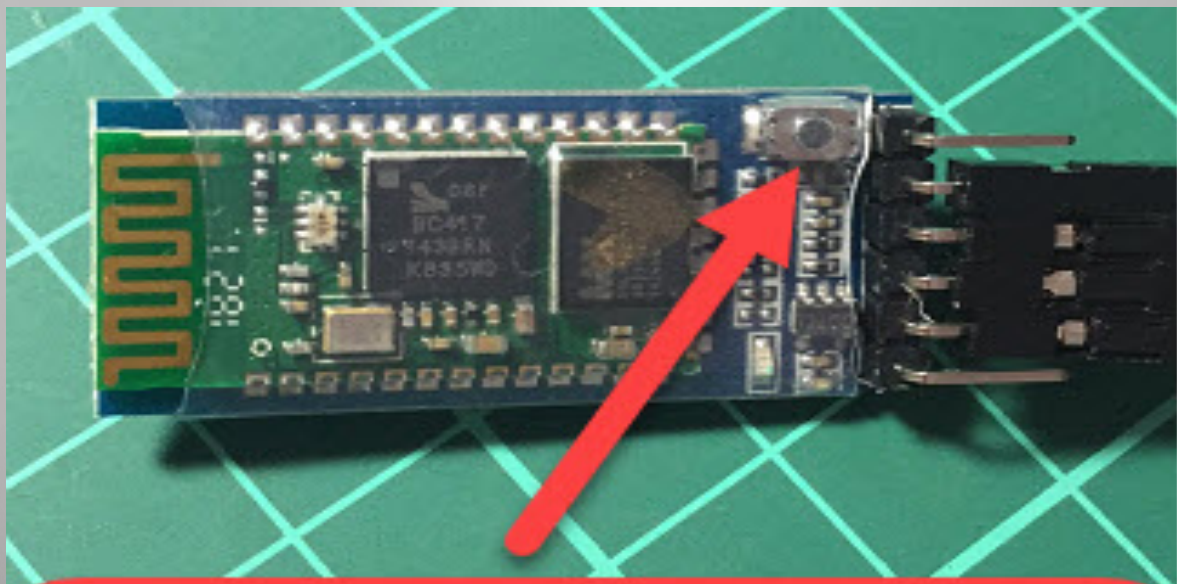
- 腳位(右至左)  
5V GND TX RX
- 密碼: 1234
- 波特率: 38400



# HC-05連線測試



進入AT模式(LED 二秒閃一次)



按著它再接上5V  
燈號會變成慢閃

# ARDUINO 程式碼

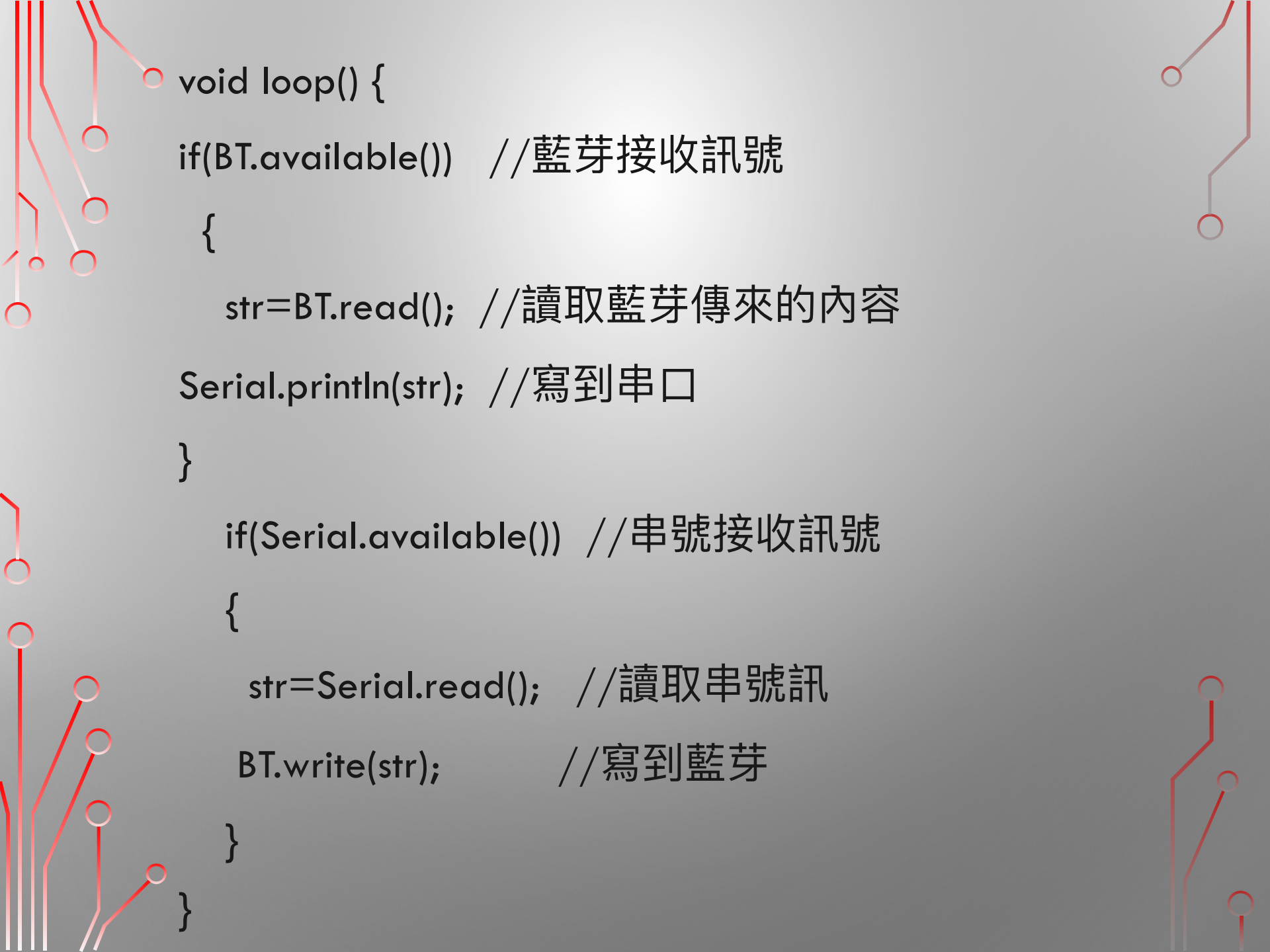
```
#include <SoftwareSerial.h> //引用程式庫
```

```
SoftwareSerial BT(10,11); //接入數位接腳(RX TX)
```

```
int str=0;
```

```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(38400);    //波特率38400  
    while(!Serial)        //等待串口  
    {  
        ;  
    }  
    Serial.println("BT START!!");  
    BT.begin(38400);    //藍芽串口波特率  
    38400  
}
```

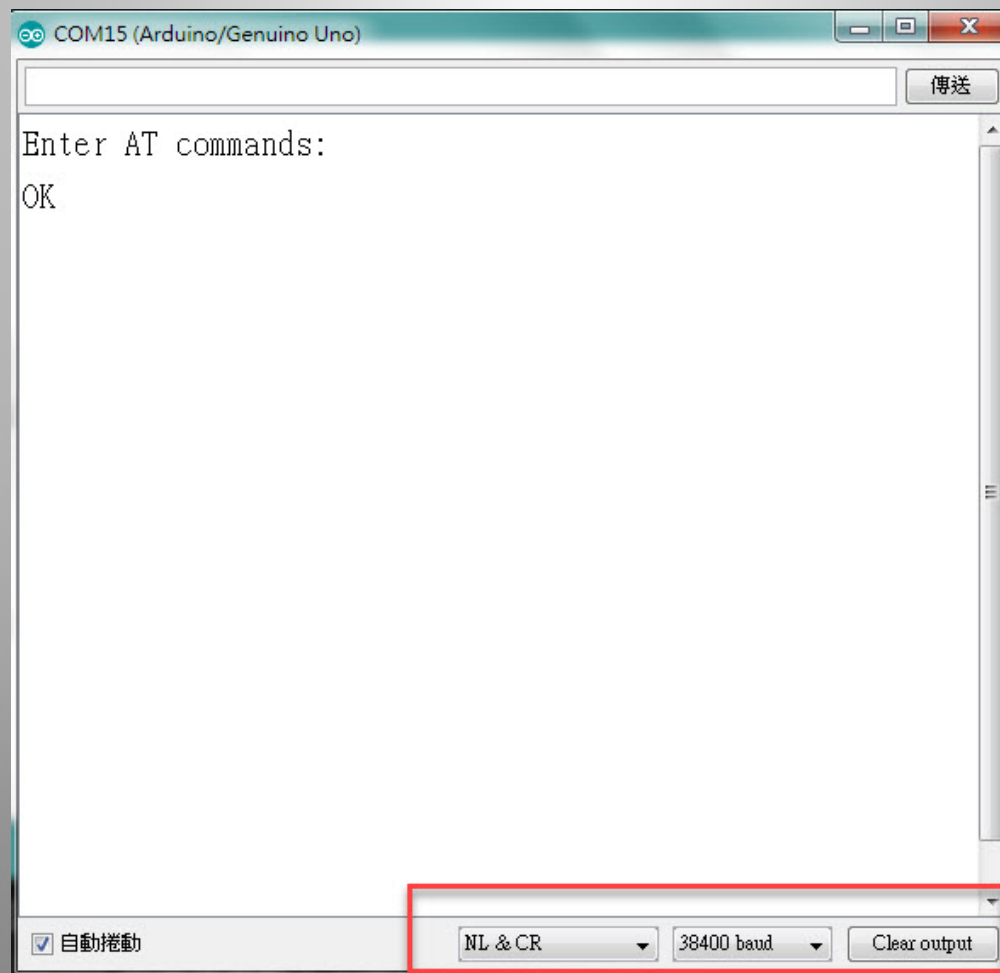




```
void loop() {  
    if(BT.available()) //藍芽接收訊號  
    {  
        str=BT.read(); //讀取藍芽傳來的內容  
        Serial.println(str); //寫到串口  
    }  
  
    if(Serial.available()) //串號接收訊號  
    {  
        str=Serial.read(); //讀取串號訊  
        BT.write(str);      //寫到藍芽  
    }  
}
```



# 串口設定



# HC 05 AT指令

- 查詢藍芽版本:AT+VERSION?
- 查詢藍芽名稱：AT+NAME?
- 設定藍芽名稱：AT+NAME=yourname
- 查詢藍芽密碼：AT+PSWD? (預設1234)
- 查詢藍芽MAC:AT+ADDR?
- 藍芽重置:AT+RESET
- 波特率：AT+UART



# 完成設定

- 設定完成後重新上電回到待機狀態

LED快閃



# APP INVENTOR 程式方塊

- 選擇模芽

當 清單選擇器1 ▾ .準備選擇

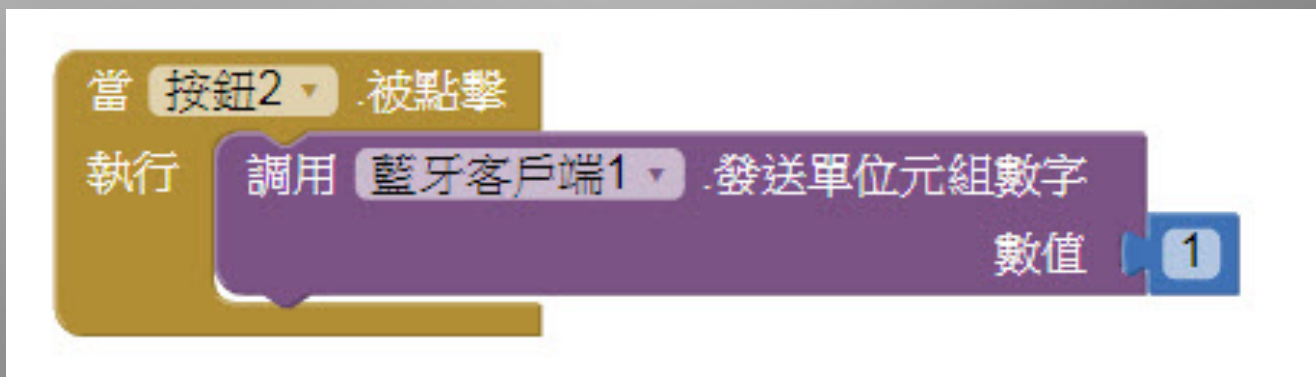
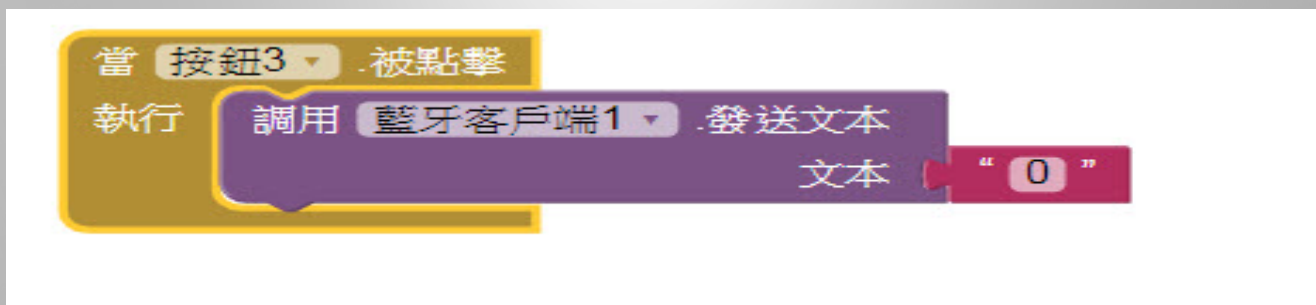
執行 設置 清單選擇器1 ▾ .元素 ▾ 為 藍牙客戶端1 ▾ .地址及名稱 ▾

當 清單選擇器1 ▾ .選擇完成

執行 如果 調用 藍牙客戶端1 ▾ .連接  
地址 清單選擇器1 ▾ .選中項 ▾  
則 設置 清單選擇器1 ▾ .背景顏色 ▾ 為



# 發送文字或數字



# 斷開連線

當 按鈕1 ▾ 被點擊

執行

調用 藍牙客戶端1 ▾ .斷開連接

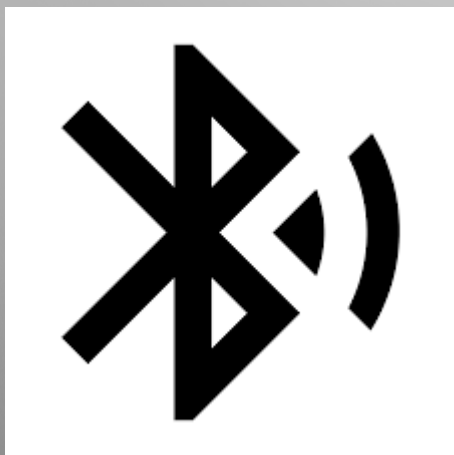
設置 清單選擇器1 ▾ . 背景顏色 ▾ 為



# 用手機控制LED開關



# 利用圖形按鍵美化APP



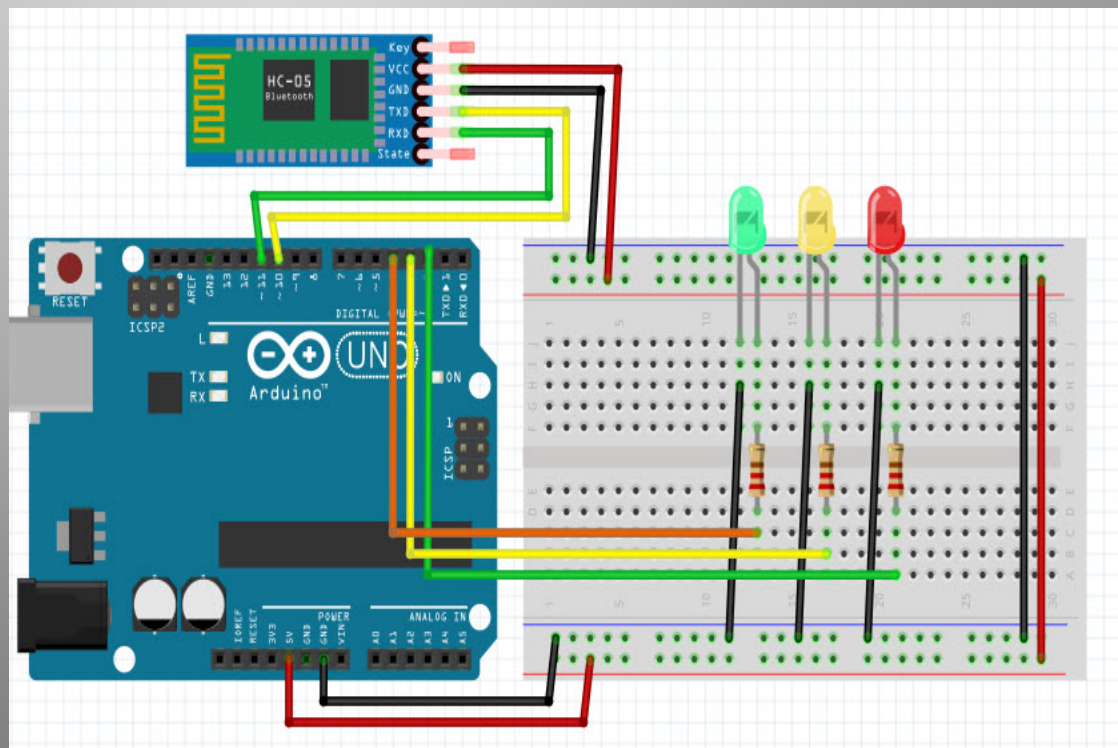


# 用手機控制三個LED燈開

- 使用手機控制三個不同LED燈開關一共六個按鍵

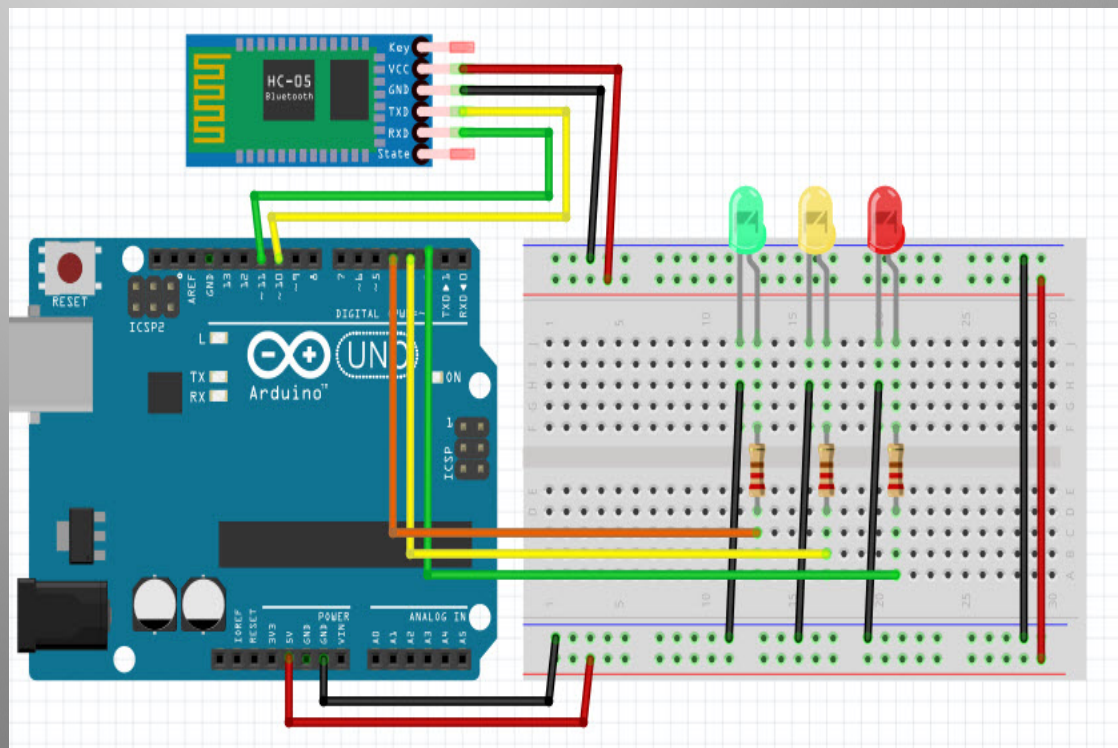
紅 綠 黃

各自獨立開關



# 用手機控制三個LED燈開

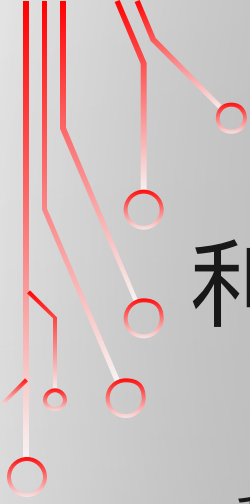
- 使用手機控制三個不同LED燈開關使用三個按鍵  
紅 綠 黃  
各自獨立開關






# 感測器--計時器元件

- ARDUINO 在固定時間發送訊息給ARDUINO
- 



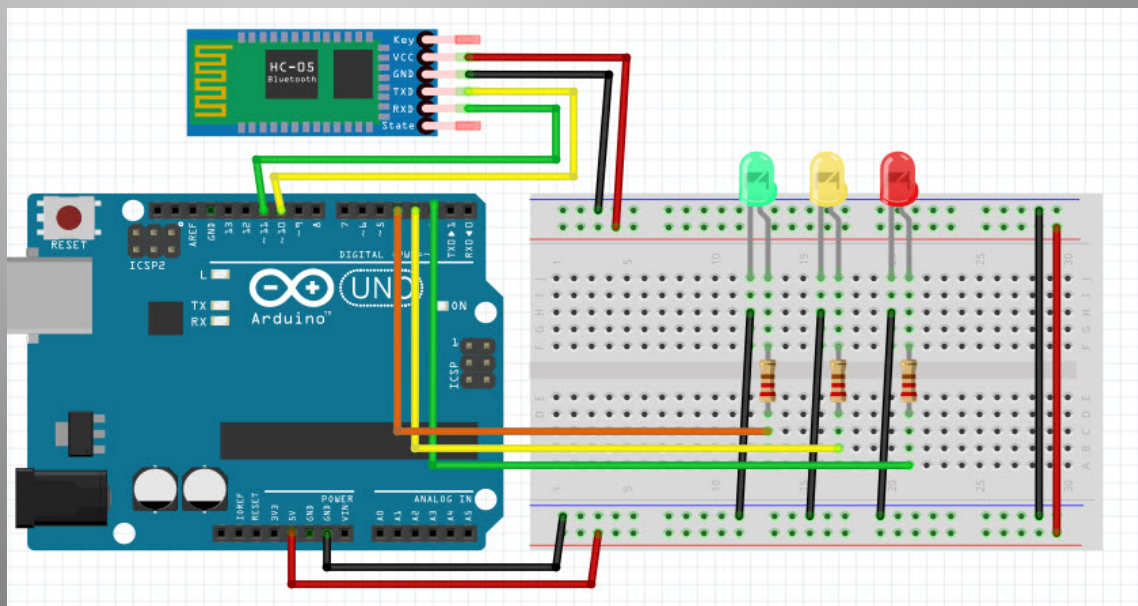
# 利用滑動條控制LED亮度

- 利用滑動條控制單一LED燈 半秒更新一次  
拉動滑動條控制明亮度
- 



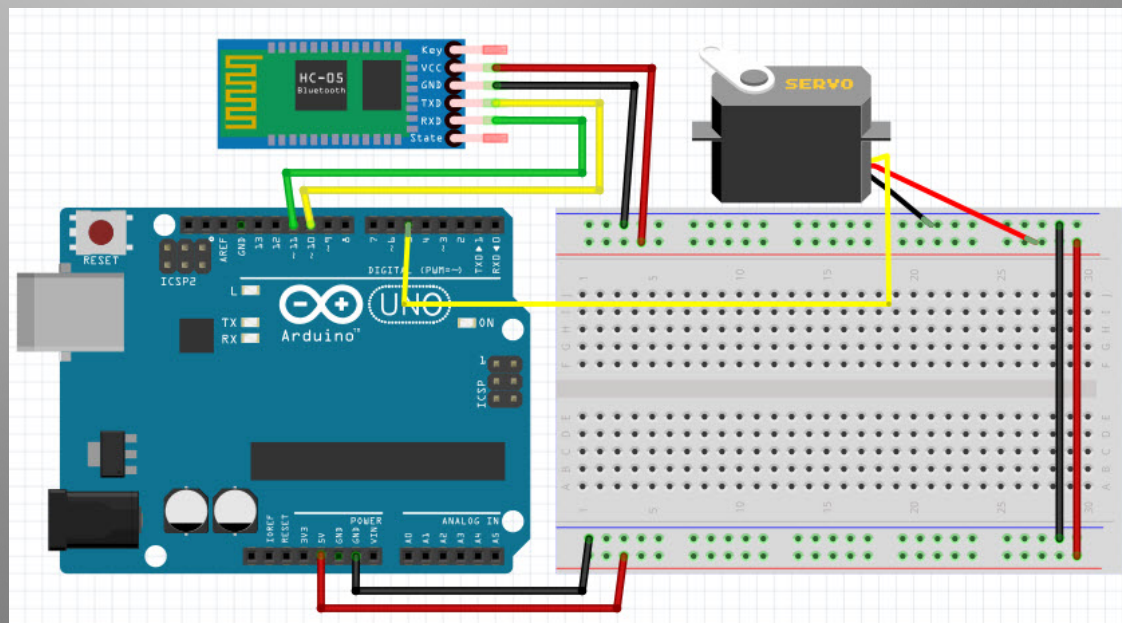
# 利用滑動條控制LED亮度


- 利用滑動條控制三個LED燈  
利用字元L 及套用PWM概念  
0~100 紅  
150~250 黃  
300~400 綠



# 利用按鍵控制伺服馬達

- 作5個按鍵控制馬達轉向  
0 45 90 135 180
- 利用滑動條控制馬達位置 半秒更新一次角度



The background features a light gray gradient with decorative circuit-like lines in red and white. These lines are concentrated in the corners, forming a frame around the central text. The lines consist of straight segments and small circles, resembling a printed circuit board (PCB) layout.

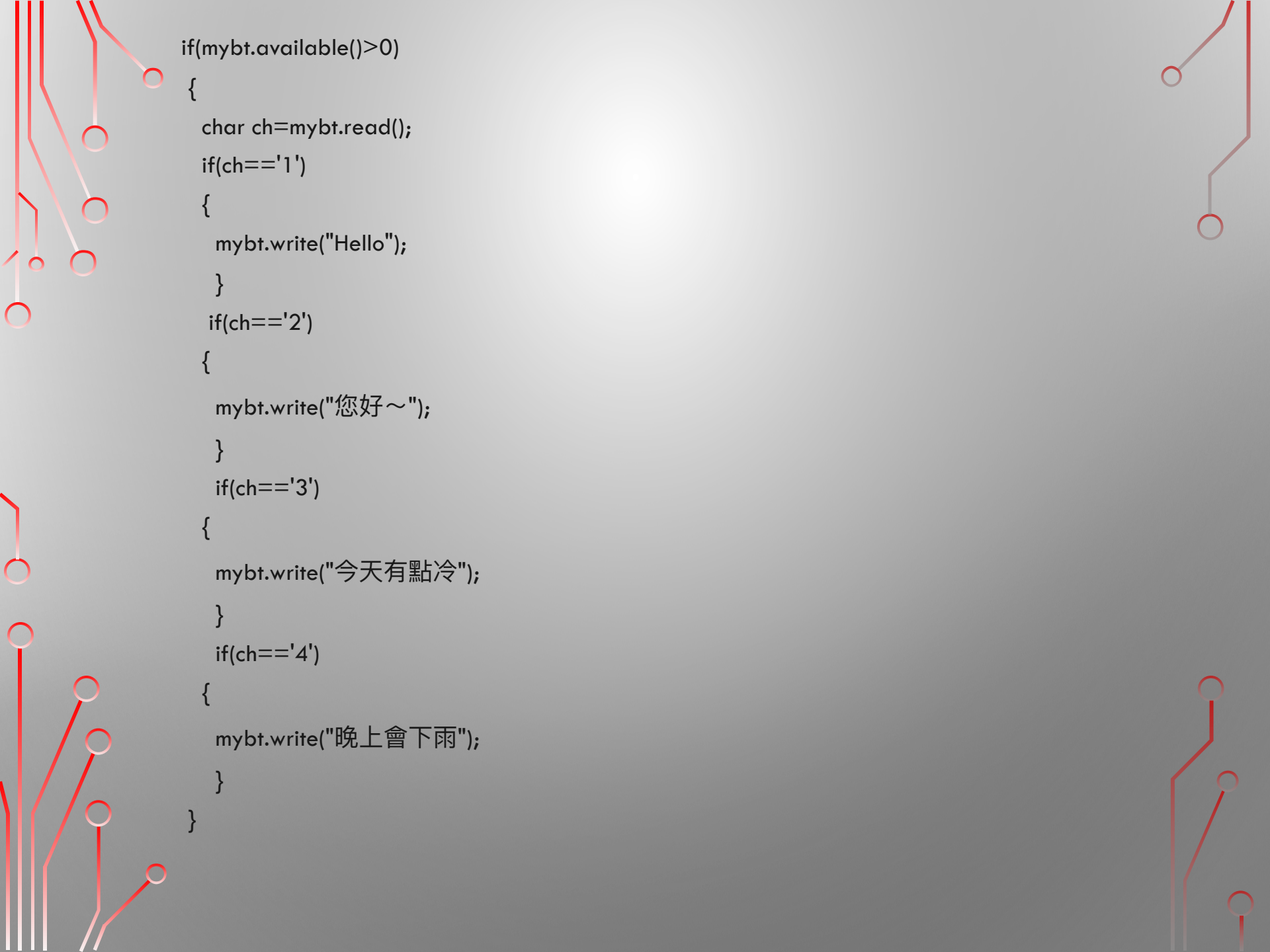
# ARDUINO 發送到ANDROID 訊息

# 讀取一個字元回覆一個字串

- 手機發送訊息給ARDUINO

- 1 返回Hello
- 2 返回您好～～
- 3 返回今天有點冷
- 4 返回晚上會下雨
- 5.....





```
if(mybt.available()>0)
{
    char ch=mybt.read();
    if(ch=='1')
    {
        mybt.write("Hello");
    }
    if(ch=='2')
    {
        mybt.write("您好~");
    }
    if(ch=='3')
    {
        mybt.write("今天有點冷");
    }
    if(ch=='4')
    {
        mybt.write("晚上會下雨");
    }
}
```

當 按鈕2 ▾ .被點選

執行

如果

文字輸入盒1 ▾ . 文字 ▾ ≠ " " ▾

則

呼叫 藍牙客戶端1 ▾ .發送文字

文字 文字輸入盒1 ▾ . 文字 ▾

當 計時器2 ▾ .計時

執行

如果

呼叫 藍牙客戶端1 ▾ .取得接收位元組數 ≠ 0

則

設 標籤1 ▾ . 文字 ▾ 為 呼叫 藍牙客戶端1 ▾ .接收文字

位元組數 呼叫 藍牙客戶端1 ▾ .取得接收位元組數

利用計時器每0.5秒讀回一次

# 讀取一組字串回覆同一組字串

- 手機發送字串給ARDUINO 同時ARDUINO反回同一個字串給手機
- 利用String 及readStrig()
- 注意發送字串要用print(字串變數)



```
if(mybt.available()>0)
```

```
{
```

```
    String str=mybt.readString();
```

```
    mybt.print(str);
```

```
}
```



# ARDUINO 發送數值到ANDROID

- ARDUINO 數值時以字元方式傳送  
所以一個INT類型發送時需要轉換成字元

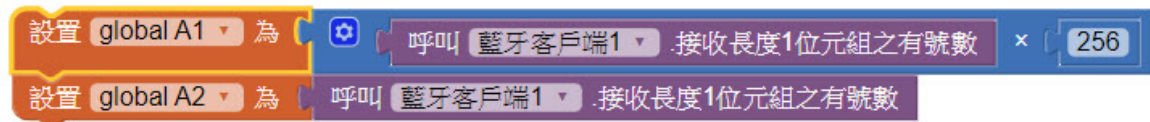
$\text{int}/256$

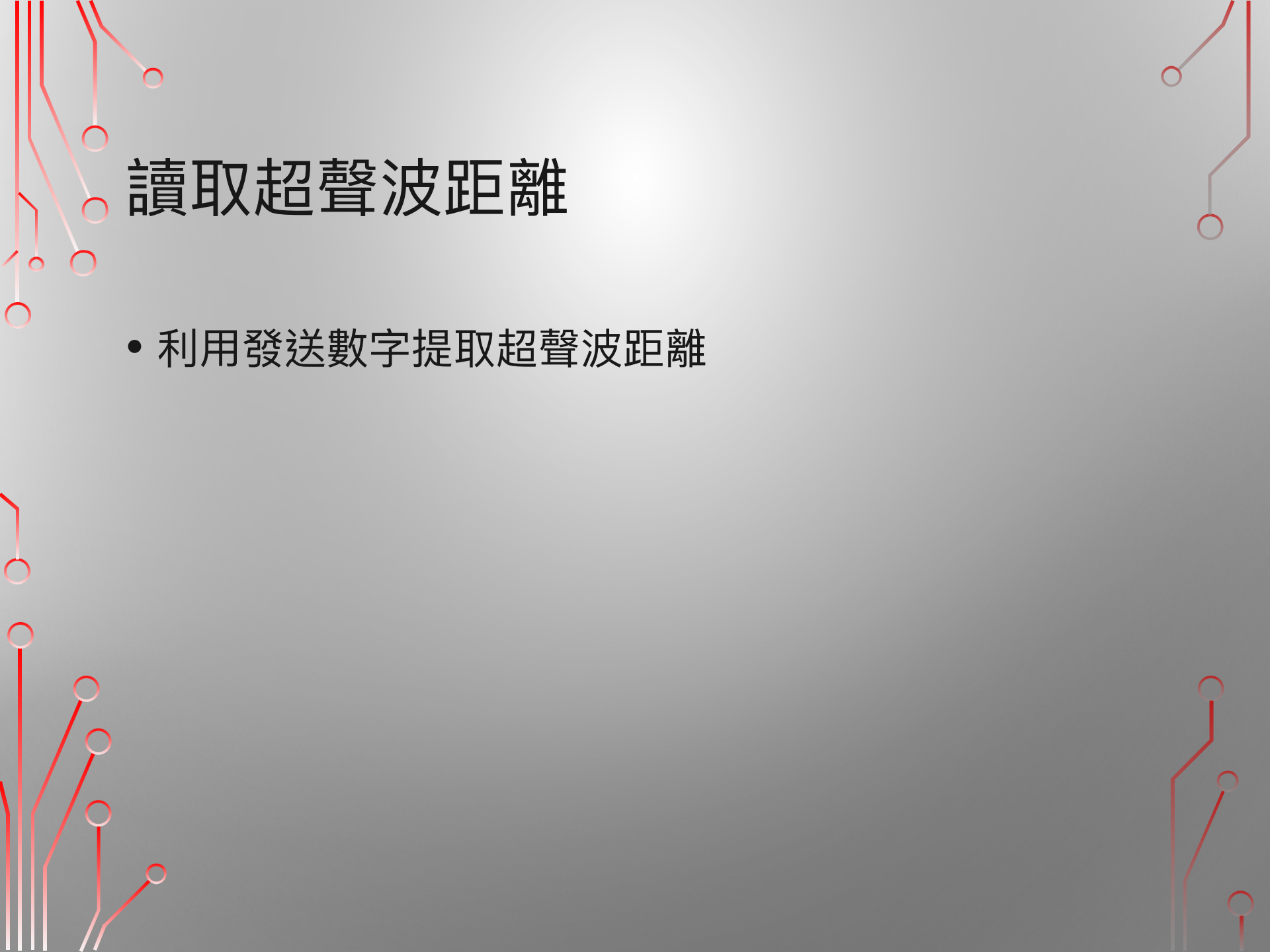
$\text{int}\%256$

byte b[2];

b[0]= int/256;

b[1]= int%256 ;





# 讀取超聲波距離

- 利用發送數字提取超聲波距離

# 超聲波距離計算

```
float dis()
{
    digitalWrite(trig, LOW);
    delayMicroseconds(2);
    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    return pulseIn(echo, HIGH) / 58.8;
    // 1/0.034*2 CM/uS
}
```

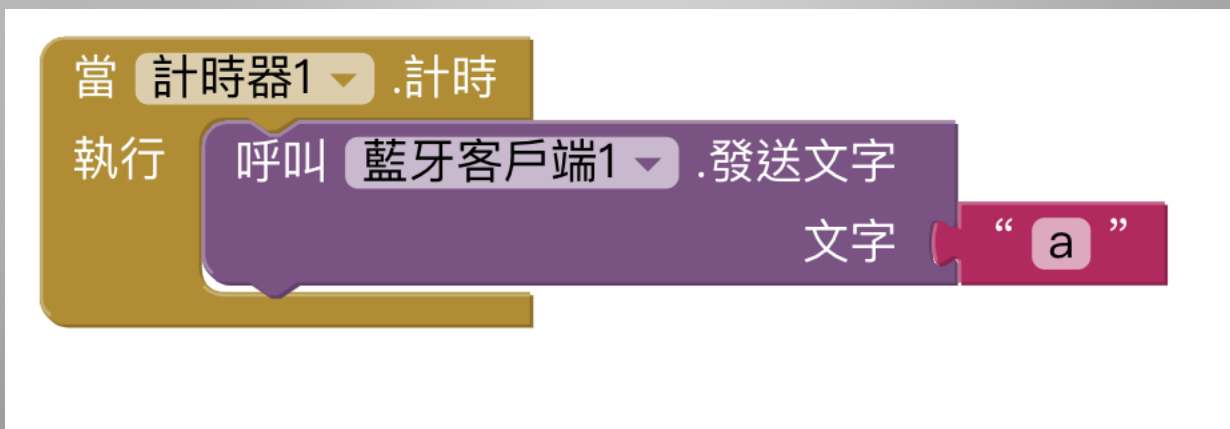
# HC05接收手機傳來的訊號

```
if (mybt.available() > 0)
{
    int value = dis()*100;    //讀取距離
    x[2];
    x[0] = value / 256;    //數值轉換成字元取256為一個數組
    x[1] = value % 256;    //把最後的餘數傳過去
    char ch = mybt.read(); //讀取是否傳來a這個字元
    Serial.println(ch);
    if (ch == 'a')
    {
        for (int i = 0; i < 2; i++)
        {
            mybt.write(x[i]);
        }
    }
}
```



# APPINVENTOR 計時器接收

- 計時器在連線後定時發送a字元取得返回訊號



# 讀取ARDUINO發送來的訊號

如果有訊號接收位元會大於0  
分別取出第一及第二個位元內容

當 計時器1 ▾ .計時

執行

呼叫 藍牙客戶端1 ▾ .發送文字

文字 “a”

如果

呼叫 藍牙客戶端1 ▾ .取得接收位元組數

>

0

則

設置 global X1 ▾ 為

為

呼叫 藍牙客戶端1 ▾ .接收長度1位元組之有號數

×

256

設置 global X2 ▾ 為

為

呼叫 藍牙客戶端1 ▾ .接收長度1位元組之有號數

# 判斷第二個數值

- 由於Arduino 的 8 bit 整數範圍是 -127 ~ 128 的緣故如果第二位數小於零則加256



# 輸出到標籤

當 計時器1 .計時

執行

呼叫 藍牙客戶端1 .發送文字

文字

“ a ”

如果 呼叫 藍牙客戶端1 .取得接收位元組數 > 0

則 設置 global X1 為 呼叫 藍牙客戶端1 .接收長度1位元組之有號數 × 256

設置 global X2 為 呼叫 藍牙客戶端1 .接收長度1位元組之有號數

如果 取 global X2 < 0

則 設置 global X2 為 取 global X2 + 256

設 標籤1 . 文字 為 取 global X1 + 取 global X2 / 100



# ARDUINO發送二筆類比訊號

//讀取光敏及可變電阻的類比數值

X = analogRead(A0);

Y = analogRead(A1);

byte[6];

byte[0] = 'a' ;

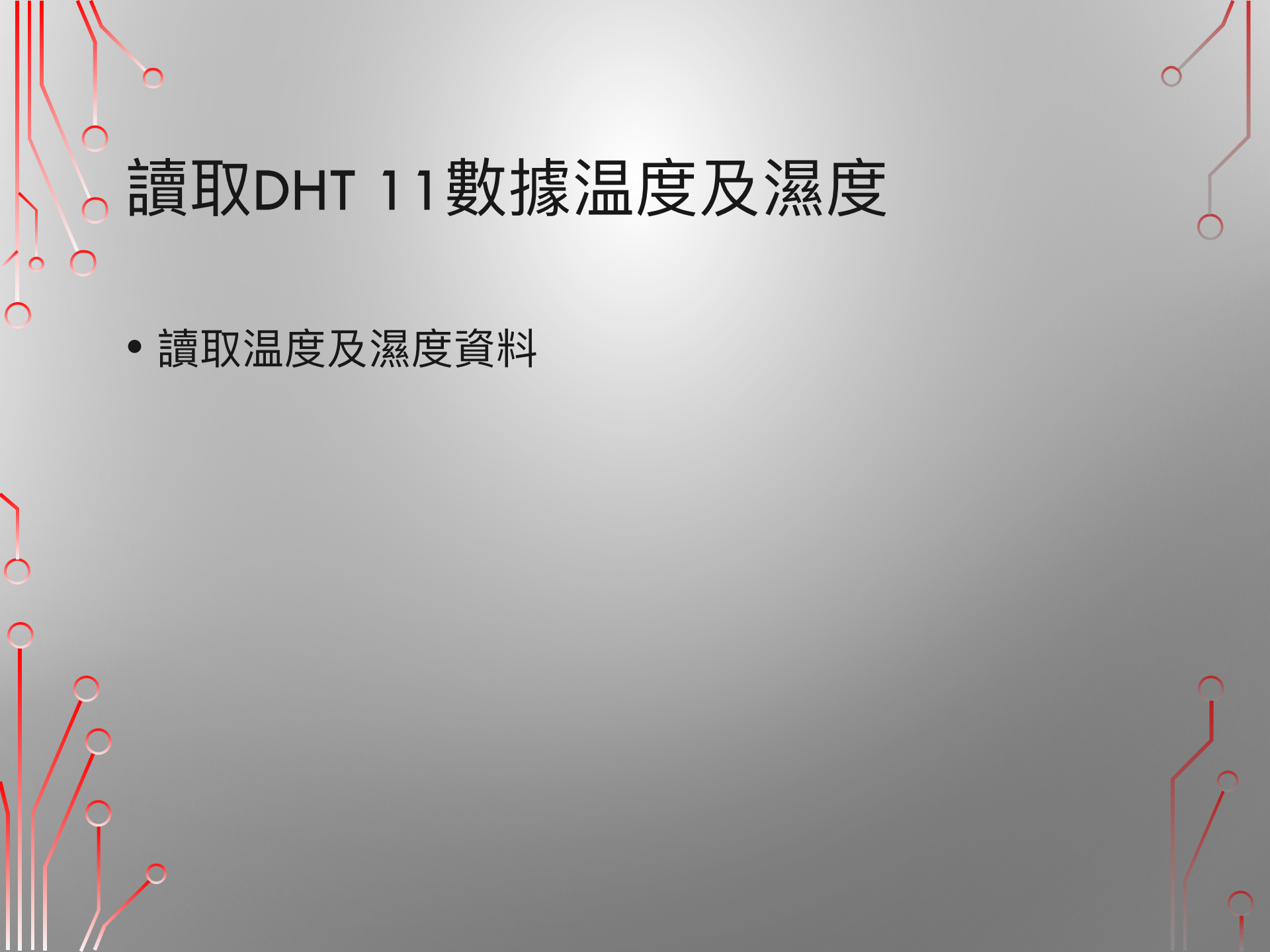
byte[1] = X/256;

byte[2] = X%256;

byte[3] = 'b' ;

byte[4] = Y/256;

byte[5] = Y%256;



# 讀取DHT 11 數據溫度及濕度

- 讀取溫度及濕度資料

