HC-05藍牙模組Master Slave配對

V2020.06.07

目的︰

HC-05藍牙模組配對，Master Slave之間互相傳送接收資料。

材料清單︰

|  |  |
| --- | --- |
| 材料 | 數量 |
| Arduino開發板 | 2 |
| USB訊號線 (A公對B公) | 1 |
| HC-05藍牙模組 | 2 |
| 按鈕開關 | 2 |
| 10KΩ電阻 | 2 |
| 杜邦線 | 一些 |

流程︰

1. HC-05藍牙模組設定Slave，並找出ADDR資料，並紀錄備用
2. HC-05藍牙模組設定Master，並指定要連線的Slave ADDR
3. 簡易硬體規畫
4. 簡易設計Arduino程式
5. 測試Master Slave配對
6. HC-05藍牙模組設定Slave，並顯示ADDR資料，紀錄備用
   1. 藍牙模組，連接Arduino開發板

藍牙模組Key準備接Arduino的 5V，線拉好備用，不接上。

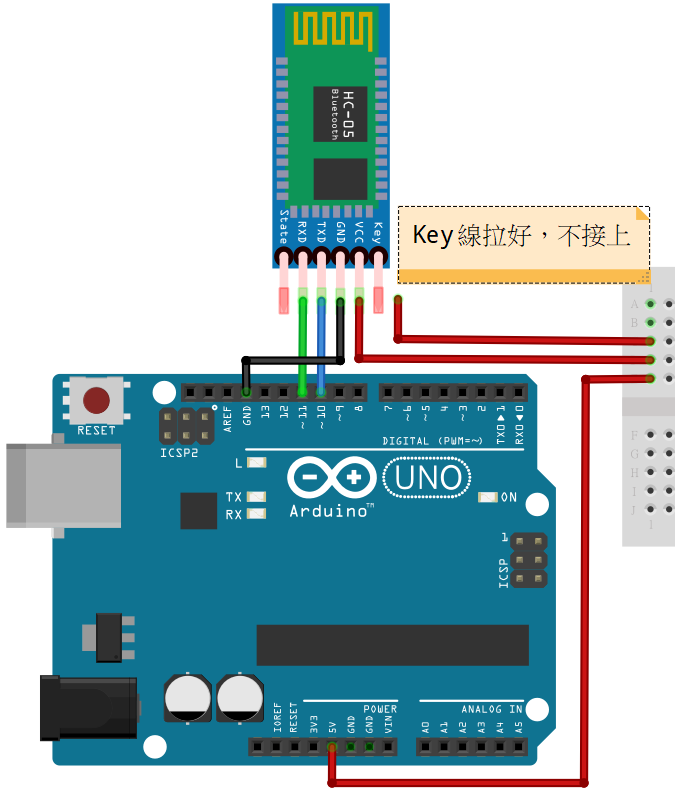
藍牙模組VCC接Arduino的 5V，(有些用 3.3V，模組背板有標註，或查原廠文件)

藍牙模組GND接Arduino的 GND

藍牙模組TX接Arduino的 D10 (軟體RX)

藍牙模組RX接Arduino的 D11 (軟體TX)

藍牙模組State 不接 (此案例用不到)



電腦USB接Arduino開發板，上傳AT mode程式 BT\_AT\_HC-05.ino

以下AT mode程式是由網路下載 (使用SoftwareSerial)，小弟精簡過︰

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11); // RX, TX

void setup()

{

Serial.begin(9600);

Serial.println("Enter AT commands:");

BTSerial.begin(38400); // HC-05 default speed in AT command mode

}

void loop()

{

// Keep reading from HC-05 and send to Arduino Serial Monitor

if (BTSerial.available())

Serial.write(BTSerial.read());

// Keep reading from Arduino Serial Monitor and send to HC-05

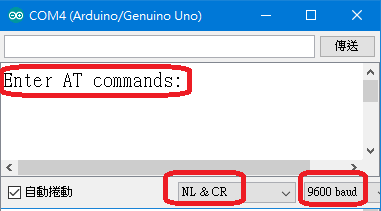
if (Serial.available())

BTSerial.write(Serial.read());

}

* 1. 進入AT mode設定Slave

打開Serial Monitor(序列埠監控視窗): 設定NL & CR，9600 baud



看到 Enter AT commands: 表示程式已準備好進入AT mode，

此時將藍牙模組Key接上5V使藍牙模組進入AT mode，

如果沒看到 Enter AT commands: 關閉Serial Monitor再重開，等待數秒。

再沒看到，將Arduino斷電後拆下Key pin，重新接上USB，打開Serial Monitor，

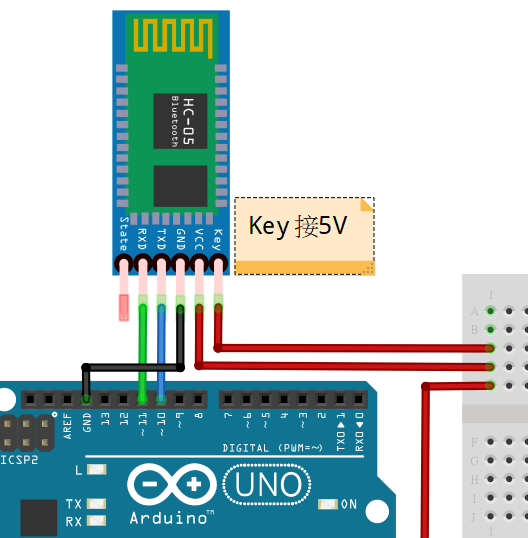
再將Key pin接上5V使藍牙模組進入AT mode

註︰目前HC-05藍牙模組有分Key pin和EN pin的不同種類

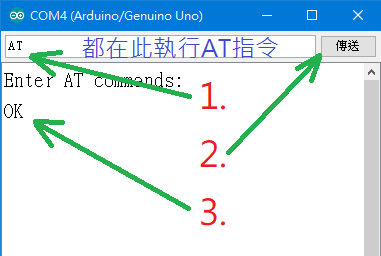
有EN pin 的HC-05藍牙模組，進入 AT mode 的方式是先接 EN pin ，先不接VCC pin，

待 Arduino開發板，程式，Serial Monitor準備好後，再接上VCC pin

設定完成後都要取下Key pin和EN pin，之後重新啟動（VCC重送電）藍牙模組才能使用



輸入大寫 AT 傳送後如果看到回應OK，就已正確進入AT mode



執行AT指令如果成功只會回應OK

Slave設定指令如下︰

AT+UART=38400,0,0

AT+CMODE=0

AT+ROLE=0

查詢指令如下，確認設定是否正確

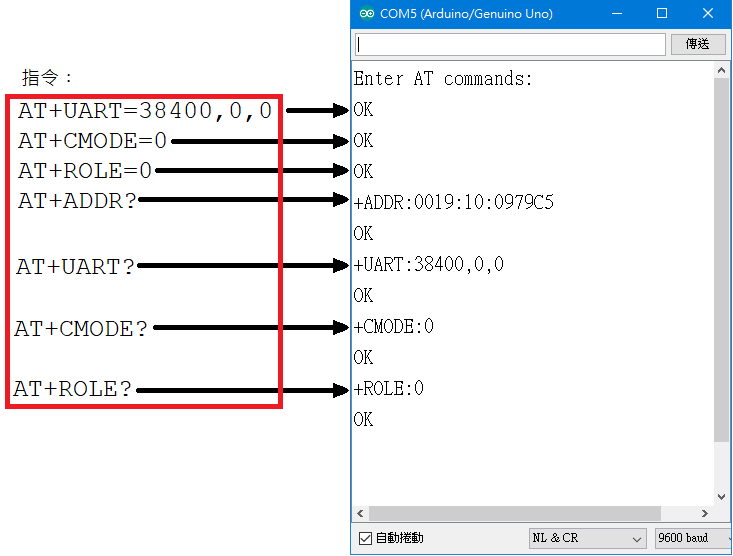
AT+UART?

AT+CMODE?

AT+ROLE?

AT+ADDR?

AT+ADDR?的回應如︰ +ADDR:0019:10:0979C5請紀錄0019:10:0979C5



設定完成後Key pin保持空接，就是不接線了，

才能進入一般使用時的communication mode

1. HC-05藍牙模組設定Master，並指定要連線的Slave ADDR

2-1. 取下已設定好的Slave藍牙模組，接上另一個藍牙模組設為Master，接線方式如1-1

2-2. 程式已上傳，不用再重新上傳。

2-3. 如1-3正確進入AT mode後，設定指令Master的指令如下︰

AT+UART=38400,0,0

AT+CMODE=0

AT+ROLE=1

AT+BIND=0019,10,0979C5

AT+BIND=0019,10,0979C5是由之前Slave查詢的+ADDR:0019:10:0979C5修改，

例如︰0019:10:0979C5 改為 0019,10,0979C5 ( : 改為 , )

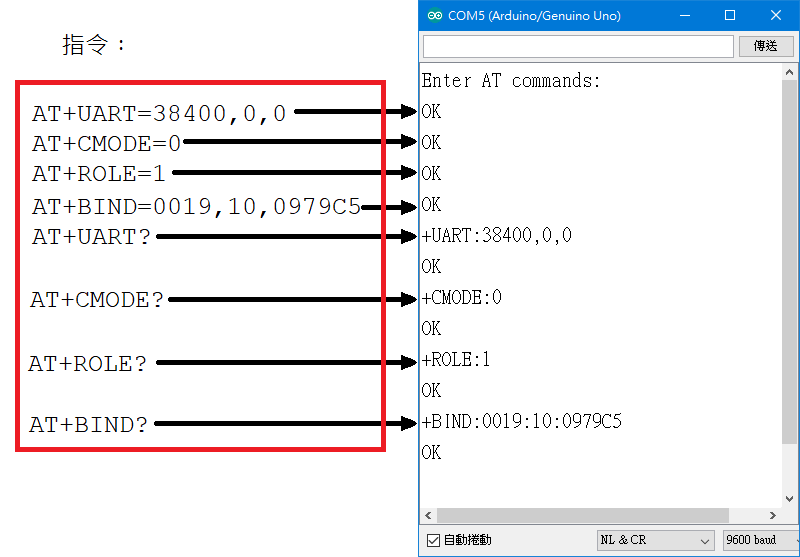
查詢指令如下，確認設定是否正確

AT+UART?

AT+CMODE?

AT+ROLE?

AT+BIND?



設定完成後Key pin保持空接，就是不接線了，

才能進入一般使用時的communication mode

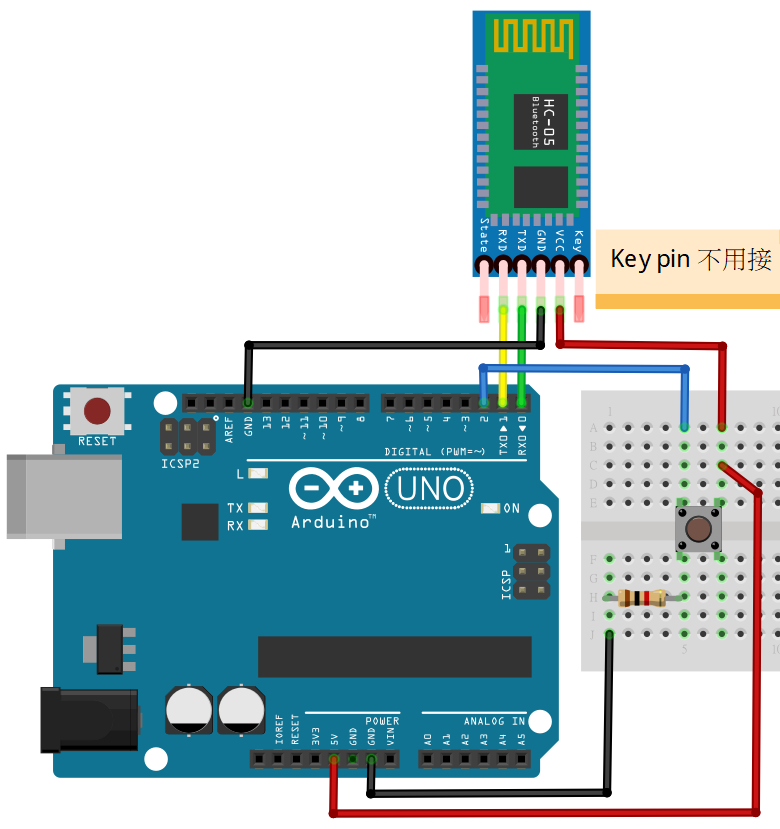
關於AT command 請看

HC-03/05 Embedded Bluetooth Serial Communication Module AT command set

附檔︰HC-05\_AT\_command.pdf

1. 簡易硬體規畫

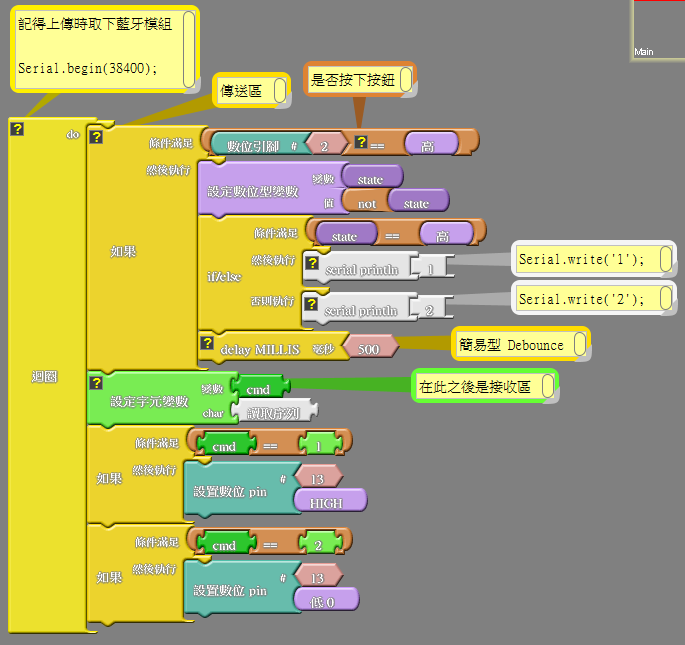
Master, Slave硬體規畫相同



1. 簡易設計Arduino程式

設計目標︰Master, Slave 使用同一支程式，用最簡單的方式驗證

ArduBlock 簡易設計如下︰ ( BT\_Master\_Slave.abp )



先取下藍牙模組RX, TX再上傳程式，ArduBlock上傳後，

在Arduino程式碼修改紅色部份如下，並二次上傳 ( BT\_Master\_Slave.ino )

bool \_ABVAR\_1\_state= false ;

char \_ABVAR\_2\_cmd = ' ' ;

void setup()

{

pinMode( 2 , INPUT);

Serial.begin(38400);

pinMode( 13 , OUTPUT);

}

void loop()

{

if (( ( digitalRead(2) ) == ( HIGH ) ))

{

\_ABVAR\_1\_state = !( \_ABVAR\_1\_state ) ;

if (( ( \_ABVAR\_1\_state ) == ( HIGH ) ))

{

Serial.write('1');

}

else

{

Serial.write('1');

}

delay( 500 );

}

\_ABVAR\_2\_cmd = Serial.read();

if (( ( \_ABVAR\_2\_cmd ) == ('1') ))

{

digitalWrite( 13 , HIGH );

}

if (( ( \_ABVAR\_2\_cmd ) == ('2') ))

{

digitalWrite( 13 , LOW );

}

}

上傳後就可以將 藍牙模組RX, TX接回

因為硬體規畫都相同，所以第二個 Arduino 開發板也用同一程式，

記得先取下藍牙模組RX, TX再上傳程式

1. 測試Master Slave配對

如果有I/O 擴展板，依其方式連接，先取下藍牙模組RX, TX再上傳程式

以上AT command設定，硬體接線，上傳軟體 如果沒有問題，

兩組Arduino供電後，藍牙模組 LED一直快閃，

Master, Slave 自動連線後，2~3秒快閃2下循環，表示連線成功

待Arduino完成開機，正常執行程式後，

按下Master按鈕開關，Slave Arduino板上的D13 LED亮，再按一下按鈕開關LED滅，

按Slave按鈕開關Master Arduino板上的D13 LED亮，再按一下按鈕開關LED滅

以上演示了Arduino藍牙Master, Slave無線通訊互傳資料簡單的方式，（雙向）

會傳一個資訊，多傳幾個應該沒什麼問題，

再套上其他感測器，多層次的程式，就可以有多功能

參考資料︰

珩宇老師的生活科技網站

<https://livingtech.education/2018/06/03/arduino%e5%af%a6%e7%94%a8-hc-05-at-mode%e4%b8%bb%e5%be%9e%e9%85%8d%e5%b0%8d/>

雄

<https://gsyan888.blogspot.com/2015/03/arduino-hc-05-master-and-slave.html>

超圖解系列圖書

<https://swf.com.tw/?p=693>

<https://swf.com.tw/?p=705>

<https://swf.com.tw/?p=712>

<https://swf.com.tw/?p=750>

<https://swf.com.tw/?p=759>

後記︰

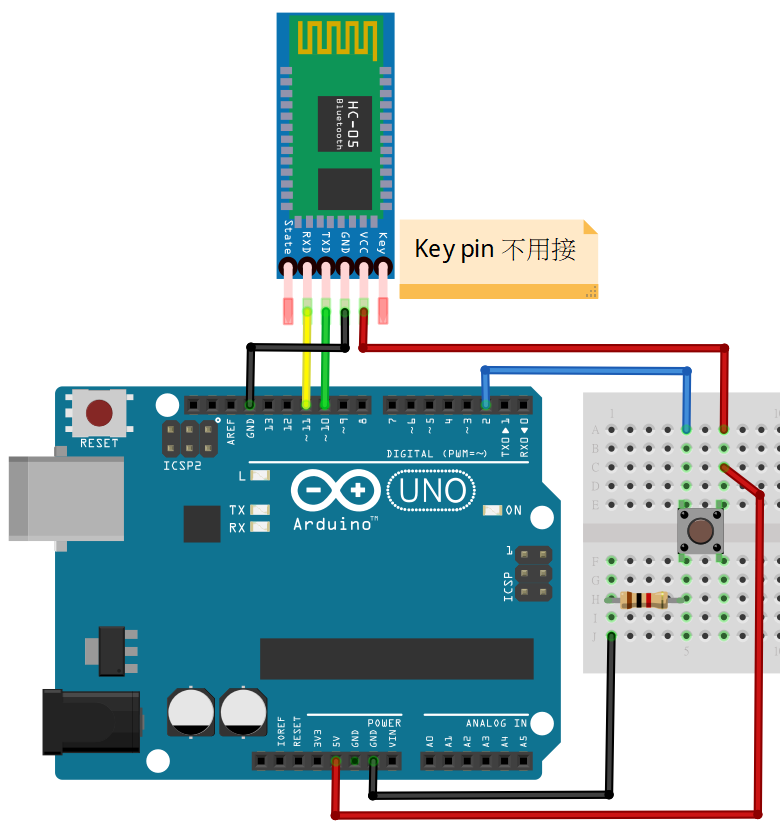
開發程式時要經常上傳程式測試，一直要拆模組，接模組，拆模組，接模組……..實在麻煩，所以用SoftwareSerial函式庫，可以不用拆裝模組，但是要佔用2個digital pin，在此用了D10，D11做軟體 RX, TX

Serial Monitor，上傳程式與Arduino通訊用了RX (D0)， TX(D1)

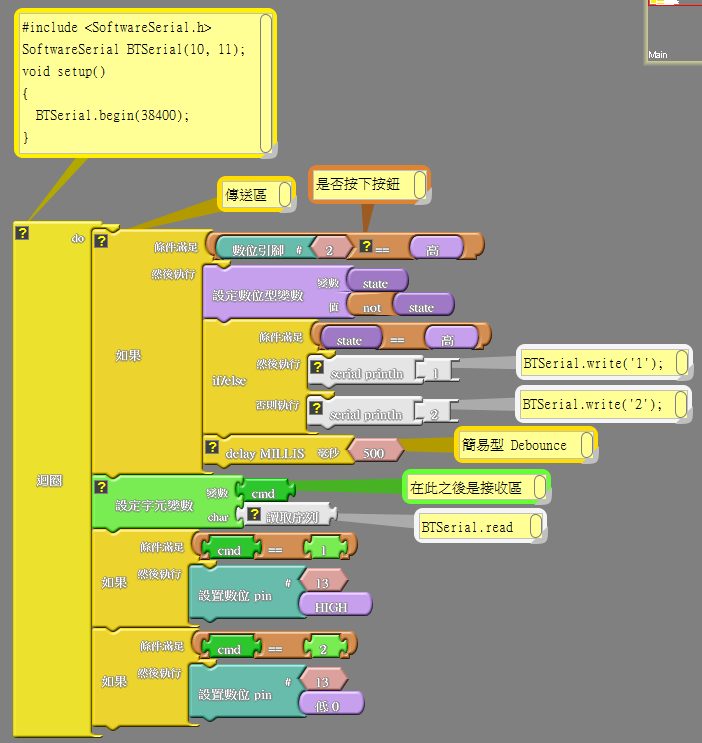
以SoftwareSerial函式庫，定義Arduino D10為軟體RX, D11為軟體TX，

給藍牙模組與Arduino通訊

硬體接線︰



ArduBlock程式如下︰其實和之前程式相同 ( BT\_Master\_Slave\_SoftwareSerial.abp )



ArduBlock上傳後，在Arduino程式碼修改部份和上面不同，

新增及修改紅色部份如下，並二次上傳 ( BT\_Master\_Slave\_SoftwareSerial.ino )

#include <SoftwareSerial.h>

SoftwareSerial BTSerial(10, 11);

bool \_ABVAR\_1\_state= false ;

char \_ABVAR\_2\_cmd = ' ' ;

void setup()

{

pinMode( 2 , INPUT);

Serial.begin(9600);

pinMode( 13 , OUTPUT);

BTSerial.begin(38400);

}

void loop()

{

if (( ( digitalRead(2) ) == ( HIGH ) ))

{

\_ABVAR\_1\_state = !( \_ABVAR\_1\_state ) ;

if (( ( \_ABVAR\_1\_state ) == ( HIGH ) ))

{

BTSerial.write('1');

}

else

{

BTSerial.write('2');

}

delay( 500 );

}

\_ABVAR\_2\_cmd = BTSerial.read();

if (( ( \_ABVAR\_2\_cmd ) == ('1') ))

{

digitalWrite( 13 , HIGH );

}

if (( ( \_ABVAR\_2\_cmd ) == ('2') ))

{

digitalWrite( 13 , LOW );

}

}

因為硬體規畫都相同，所以第二個 Arduino 開發板也用同一程式，直接上傳就好。

不同的是不用再拆接藍牙模組 RX, TX

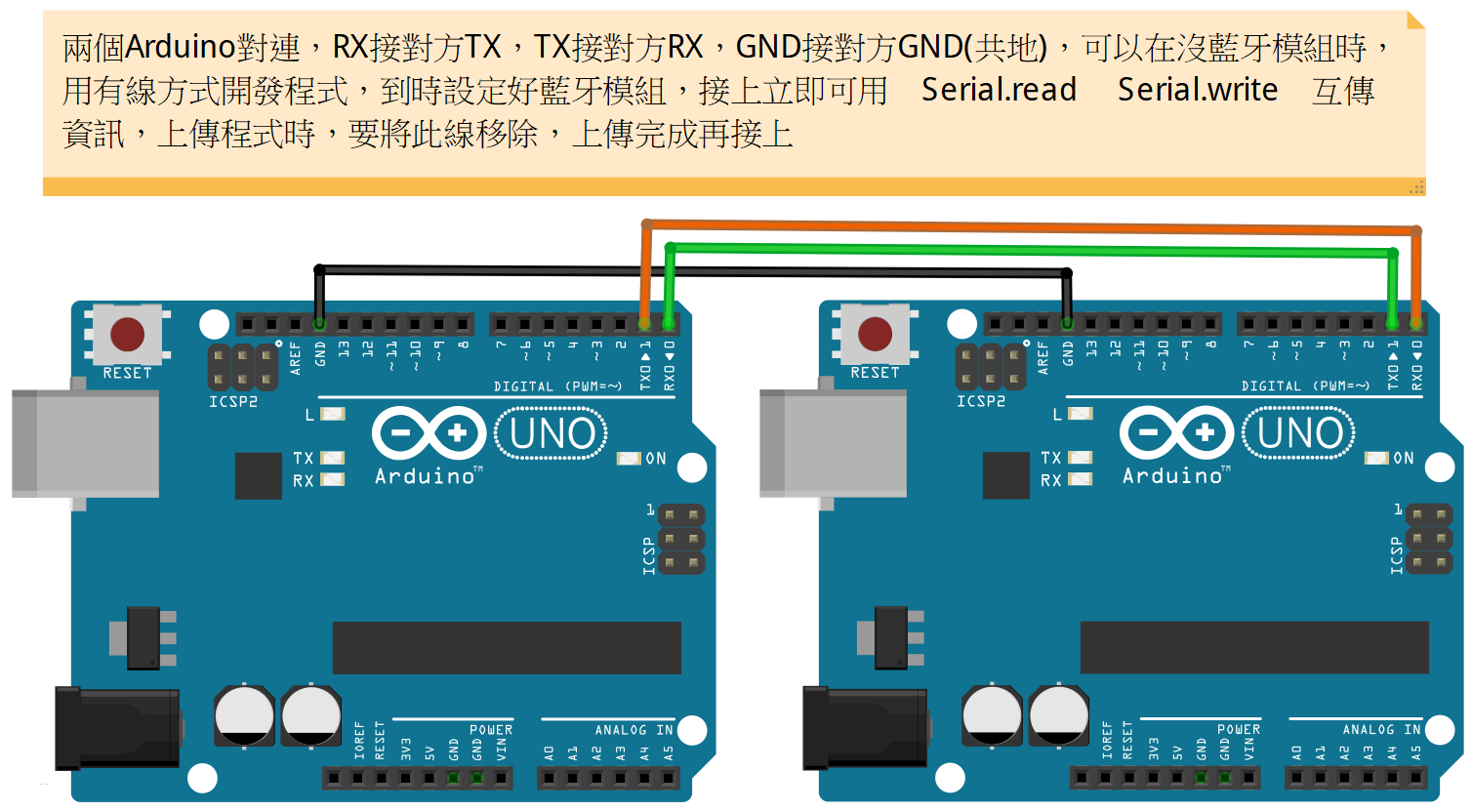
註1︰ 可修改程式定義RX/TX腳位，並接好藍牙模組對應Arduino腳位（我試過D7,D8正常）

註2︰ 有些藍牙模組VCC使用3.3V，經測5V在設定階段正常，長久使用時建議改接3.3V，

有些藍牙模組VCC使用5V

再後記︰

假如我暫時沒有藍牙模組可用，也想體驗一下，是可行的！



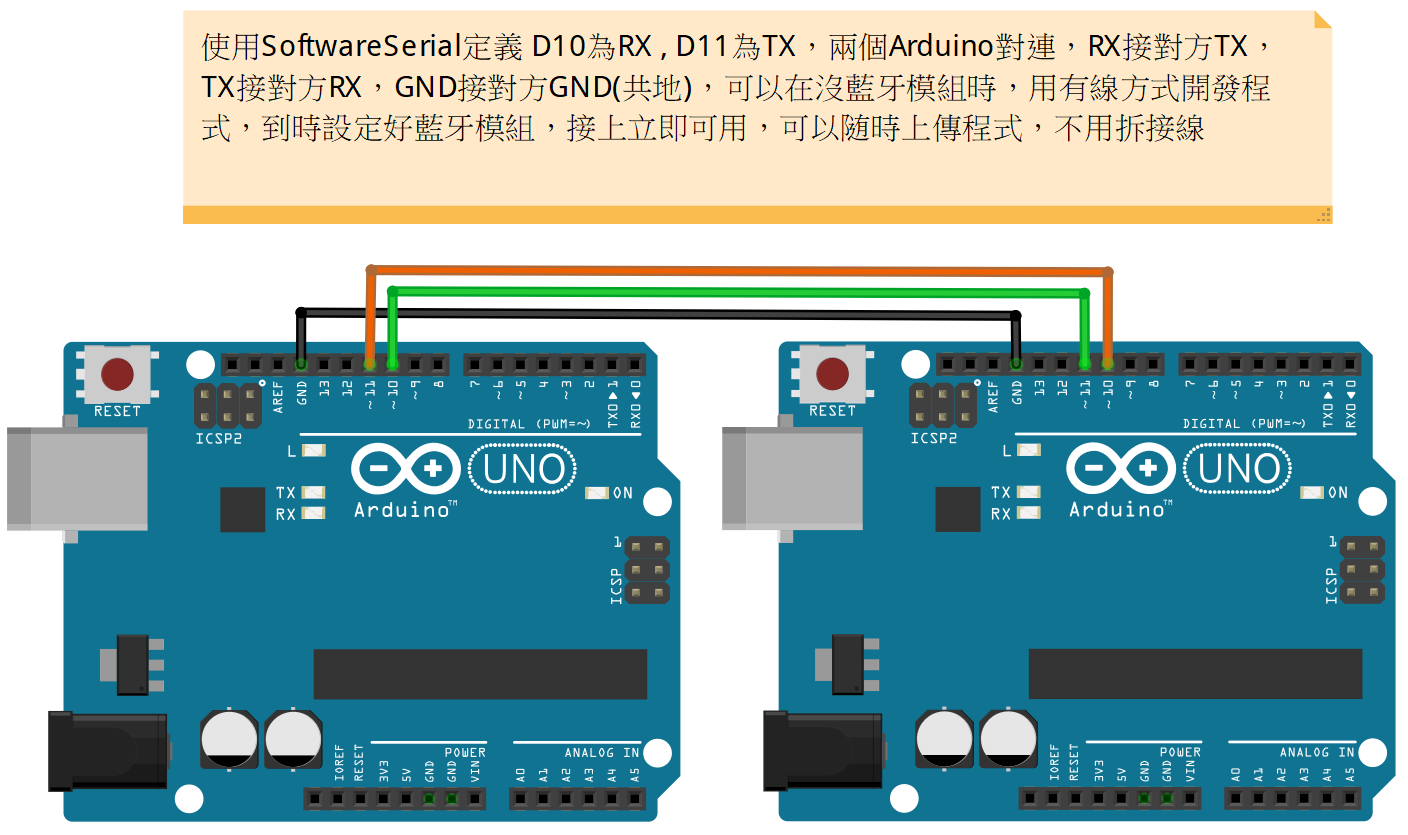
上圖只有畫出取代 HC-05藍牙模組部份，按鈕開關感測器和之前一樣不變

一樣上傳程式時要取下 RX, TX

兩組Arduino程式和之前一樣 ( BT\_Master\_Slave.ino )

再再後記︰

假如我暫時沒有藍牙模組可用，想體驗一下，不想上傳程式時一直拆接線，也是可行的！



也是只畫出取代 HC-05藍牙模組部份，按鈕開關感測器和之前一樣不變

程式和”後記”使用SoftwareSerial一樣 ( BT\_Master\_Slave\_SoftwareSerial.ino )