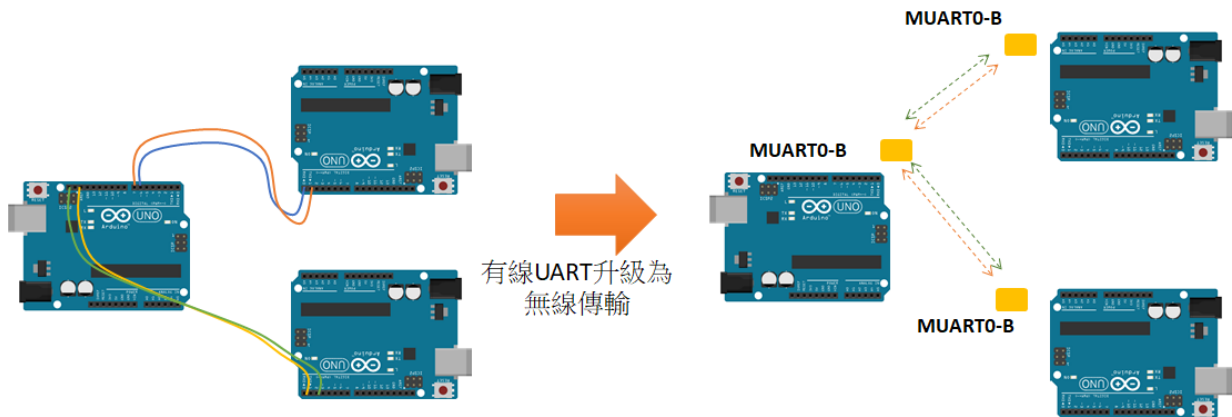


MUART0-B

解放 UART

立即無痛升級無線傳輸



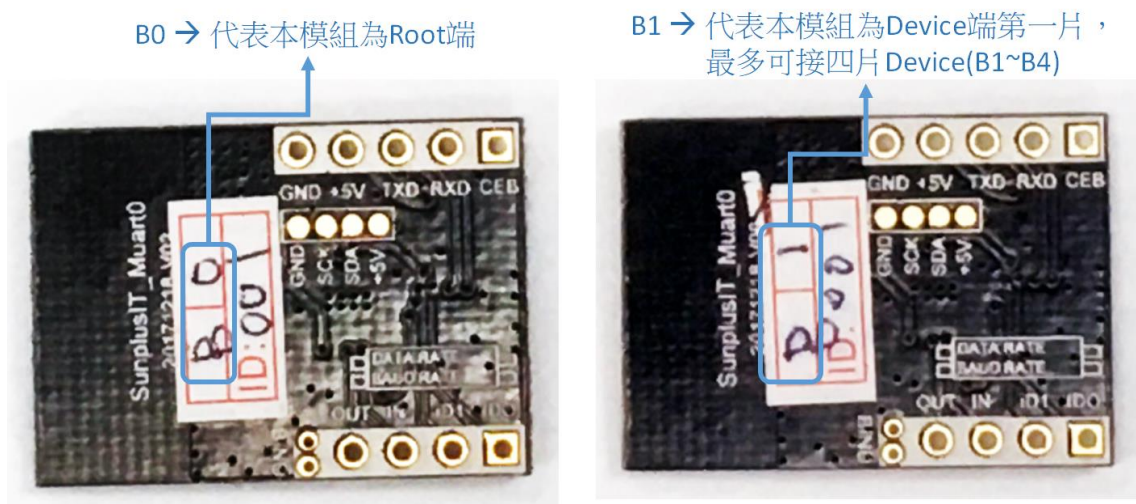
目錄

模組外觀與尺寸.....	2
模組特性.....	2
Pin 腳定義	3
如何使用	3
選擇要連接的 Device 編號.....	3
傳送資訊.....	4
與 Arduino 搭配使用	4
Root 端傳送端程式範例：	5
RX 接收端程式範例：	5
執行.....	6
與 Raspberry Pi 樹莓派搭配使用.....	6
程式範例：	7
直接對接感測器.....	7
使用 IO Ports	8

MUART0-B 無線 Uart 傳輸模組是一款簡單易用的模組，它能將有線的 UART 立即無痛升級為無線 UAR 傳輸，此外還提供一組 I/O pin，讓您不需要額外 coding 及硬體及就有可相互遙控的 IO 開關來使用。

模組外觀與尺寸

MUART0-S-P 模組包含 Root 端（編號 B0）一片，以及最多四片的 Device 端（編號 B1~B4），兩者外觀相同，但可由背面的標籤來辨識 Root 或 Device（如下圖）。

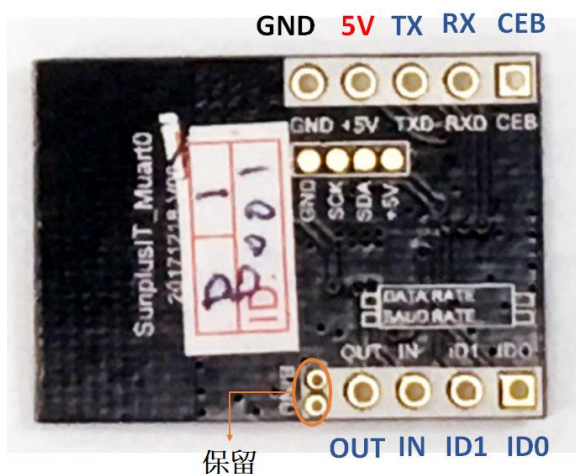


模組特性

1. 操作電壓：3.3~5.5V
2. **RF** 頻率：2400MHz~2480MHz。
3. 耗電量：傳送約 24mA@+5dBm，接收約 23mA。
4. 發射功率：+5dBm
5. 傳輸速率：250Kbps
6. 傳輸距離：空曠處約 80~100m
7. **Baud rate**：9,600bps
8. 可支援 1 對 1 或 1 對多（最多四個）的傳輸。

Pin 腳定義

由左而右，由上至下：



GND → Ground

+5V → 5V 電壓輸入

TX → 對應到開發板 Uart 的 RX

RX → 對應到開發板 Uart 的 TX

CEB → CEB pin 腳需接地(GND)模組才會通電運作，可作為省電控制功能使用。

OUT → IO Port 的輸出 pin (On/Off 輸出)

IN → IO Port 的輸入 pin (On/Off 接收)

ID1, ID0 → 透過此兩 pin 腳的 HIGH/LOW 組合來選擇與那一片 device 連接。

如何使用

凡是支援 UART 通訊介面的各類開發板及 MCU 皆可直接使用本模組，不需要安裝額外的 driver 或 API 程式。

選擇要連接的 Device 編號

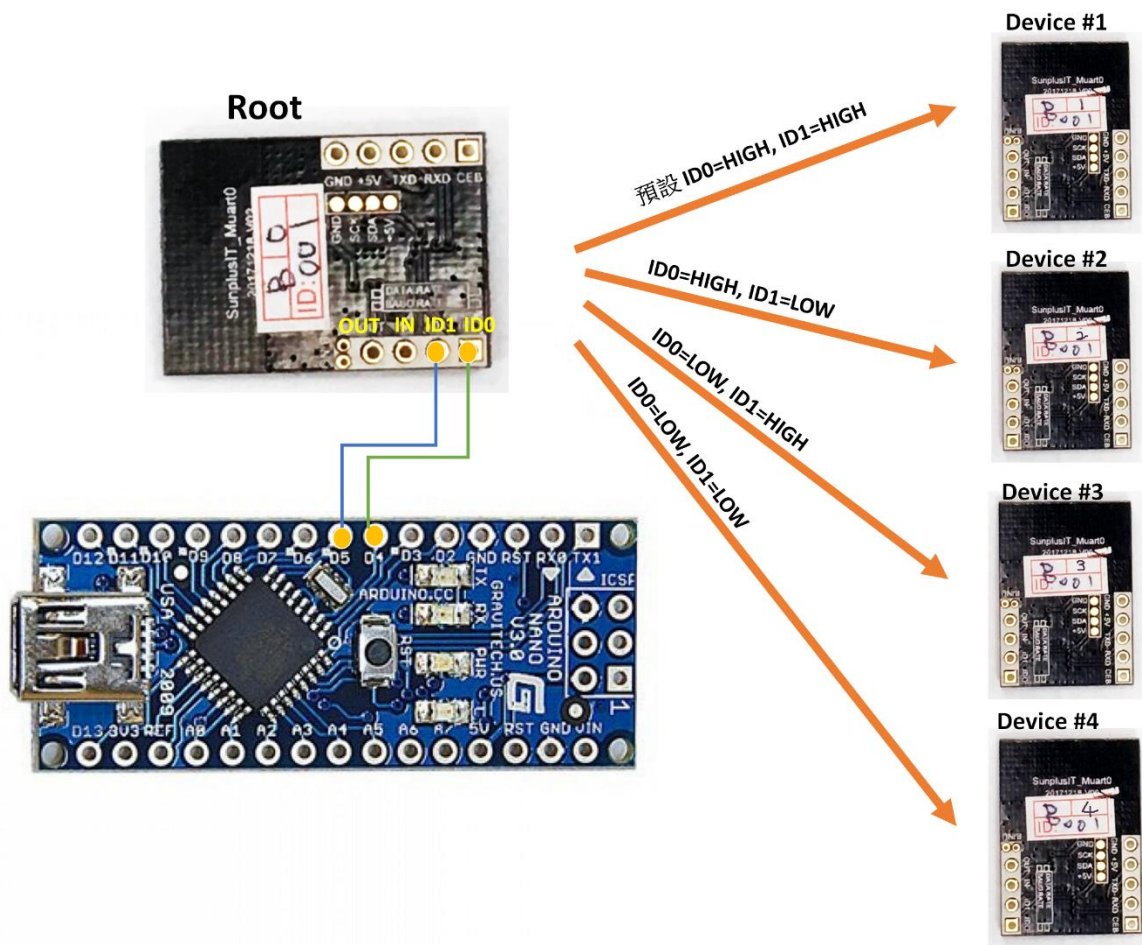
傳統有線 TTL 為 1 對 1 方式傳輸，本 MUART0-B 無線 Uart 傳輸模組額外支援 1 對多模式，預設 Root 端 (P0) 通電後與 Device (P1) 連接，如果您有其它編號的 Device (P2~P4)，可在 Root 端針對 ID0, ID1 pin 送出不同的 HIGH/LOW 組合，來選擇欲連接的 device 編號，Root 端的 ID0, ID1 編號選擇組合請參考下方表格。

	Device 1 (B1)	Device 2 (B2)	Device 3 (B3)	Device 4 (B4)
ID0 pin	HIGH	HIGH	LOW	LOW
ID1 pin	HIGH	LOW	HIGH	LOW

ID0, ID1 pin 預設為 HIGH，若接到 GND 則為 LOW

您可以參考上方的對應表格，手動將 GND 線路接到 ID0 或 ID1，讓 Root 端固定與指定 Device 相連接，也可以透過開發板的 pin 腳位，透過程式送出 LOW/HIGH 訊號來動態指定要連接的 Device。

例如下圖，Arduino Nano 透過 D4 與 D5 pin 來選擇要連接的 Device。



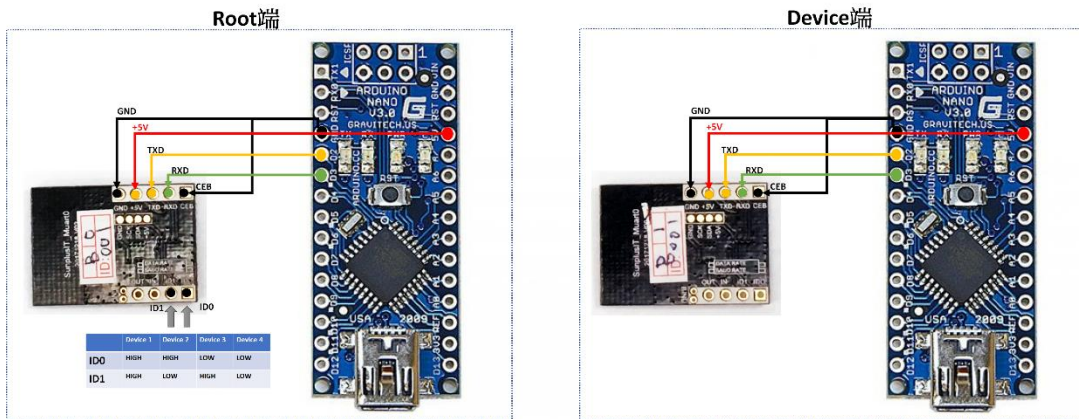
傳送資訊

在您確定連接的 **device** 對象後，便可以開始傳送資訊了。以下分別以 **Android**、樹莓派兩種開發板以及感測器直接對接的三種方式為例來說明：

與 **Arduino** 搭配使用

除了直接使用 **Arduino** 的 **hardware TX/RX ports** 之外，本模組也支援 **software serial**，因此能以軟體模擬 **Uart** 方式來使用以避免佔用實體的 **UART** 介面。例如下方範例為透過 **software serial**，將 **Arduino Nano** 的 2, 3 pin 腳分別模擬為 **Uart RX** 與 **TX** 相互傳送及接收訊息，其它型號的 **Arduino** 板子也是同樣的接法。

如果要透過程式連接指定的 **device**，可以使用 **Arduino** 的指令 **digitalWrite** 針對 **ID0, 1** 兩個腳位輸出 **LOW** 或 **HIGH**，來達到動態連接到指定的 **device**。



Root 端傳送端程式範例：

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); // RX, TX

void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
}

void loop() {
  mySerial.print("0123456789");
  Serial.println("0123456789");
  delay(1000);
}
```

RX 接收端程式範例：

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(2, 3); // RX, TX

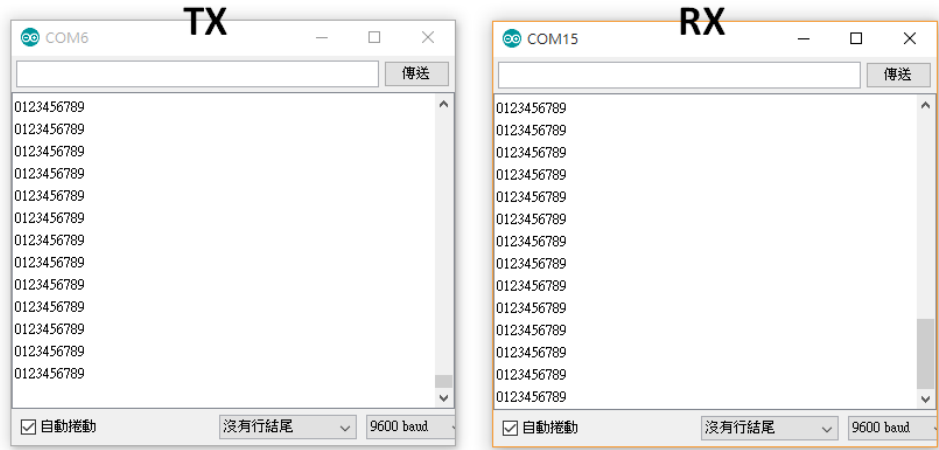
void setup() {
  Serial.begin(9600);
  mySerial.begin(9600);
}

void loop() { // run over and over

  if (mySerial.available()) {
    Serial.println("");
    while (mySerial.available()) {
      Serial.print(char(mySerial.read()));
    }
  }
}
```

```
}  
  
delay(1000);  
  
}
```

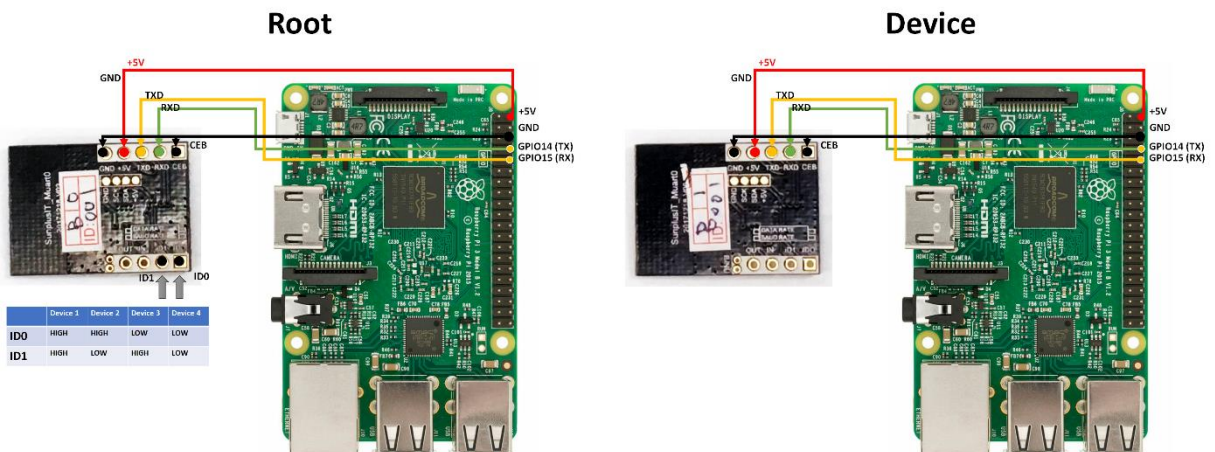
執行



與 Raspberry Pi 樹莓派搭配使用

在樹莓派上使用本模組也是相當容易的！將本 RF 模組各腳位接到樹莓派上相對應的 RX/TX 腳位即可，您就可以直接對 RX/TX 腳位進行讀取寫入，使用方式就如同傳統的 UART。

下圖為樹莓派與 MUART0-B Root 端模組的接法，Device 端的接法也相同，但 ID 與 ID1 pin 腳則不需要。



程式範例：

傳送與接收端皆適用

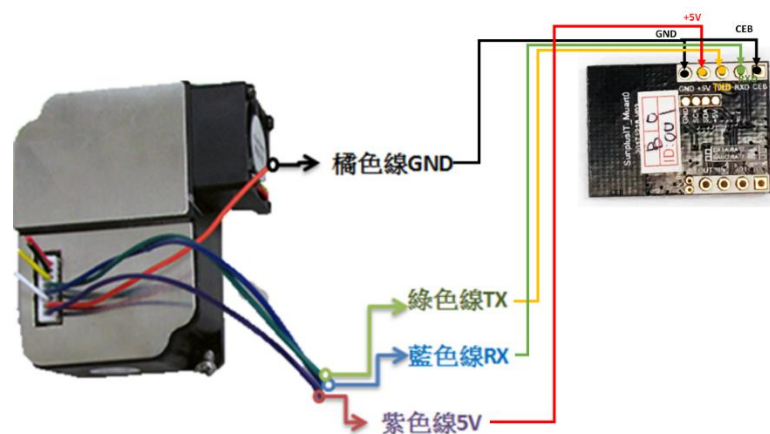
```
#!/usr/bin/env python
import time
import serial

ser = serial.Serial(
    port='/dev/ttyS0',
    baudrate = 9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=1
)
counter=0

while 1:
    #讀取 UART 資料
    x=ser.readline()
    print x
    #寫入資料到 UART
    ser.write('Write counter: %d \n'%(counter))
```

直接對接感測器

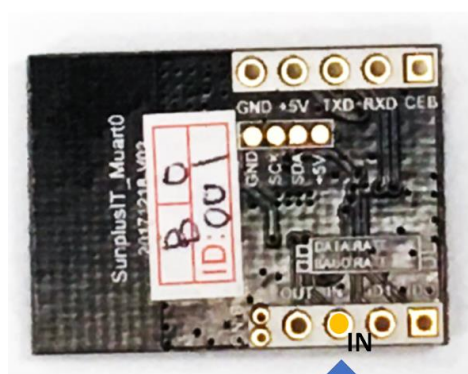
如果您的感測器支援 UART 介面而且 Baud rate 是 9,600，那麼您可以直接將它接到本模組，就能迅速無痛升級成無線功能的感測器了。下方以 G3 PM2.5 感測器為例，參考如下接法，



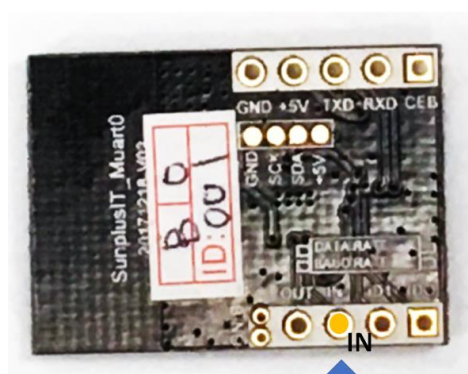
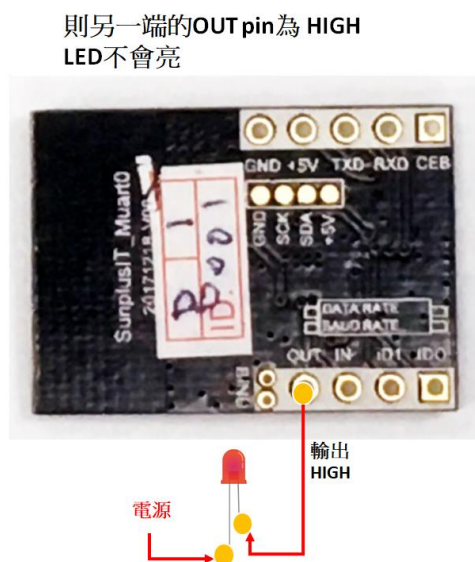
接著再請您準備一張開發板（**Arduino** 或樹莓派皆可），就可以像一般的 **UART** 方式來讀取到 **G3** 所傳來的 **PM2.5** 數據，恭喜你，該 **G3** 已升級為具有無線傳輸功能的 **PM2.5** 感測模組。

使用 IO Ports

MUART0-B 模組提供一組 **IO port**，它讓您可以透過無線來傳送 **on/off** 指令，且這組 **IO Port** 並不受限於模組的傳送或接收端，兩端皆可相互控制。您只要在任何一端的 **IN port** 改變電壓，就會同步改變另一端 **Out port** 的輸出電壓。請參考如下的使用範例，說明如何使用 **IO Port** 來遙控開關 **led** 燈泡。



當一端的 **IN** 腳位空接或電位為 **HIGH** 時



當一端的 **IN** 腳位接到 **GND** 或輸入 **LOW**

