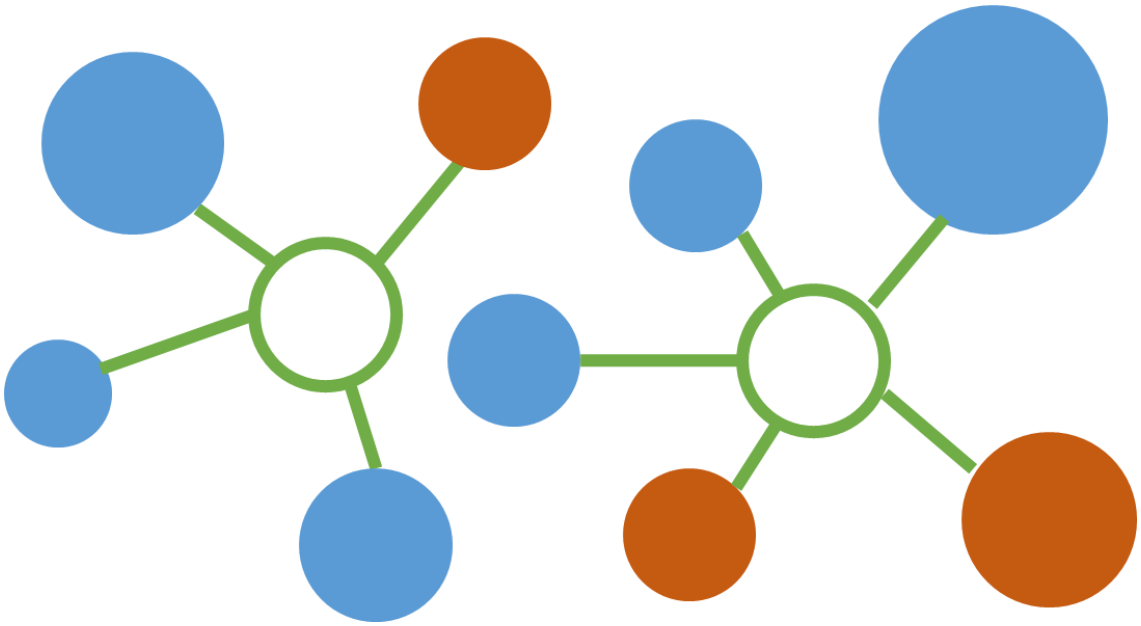
MUART0-PP-1-N

SunplusIT 一對多無線 Uart 模組

無限擴展您的想像力



目錄

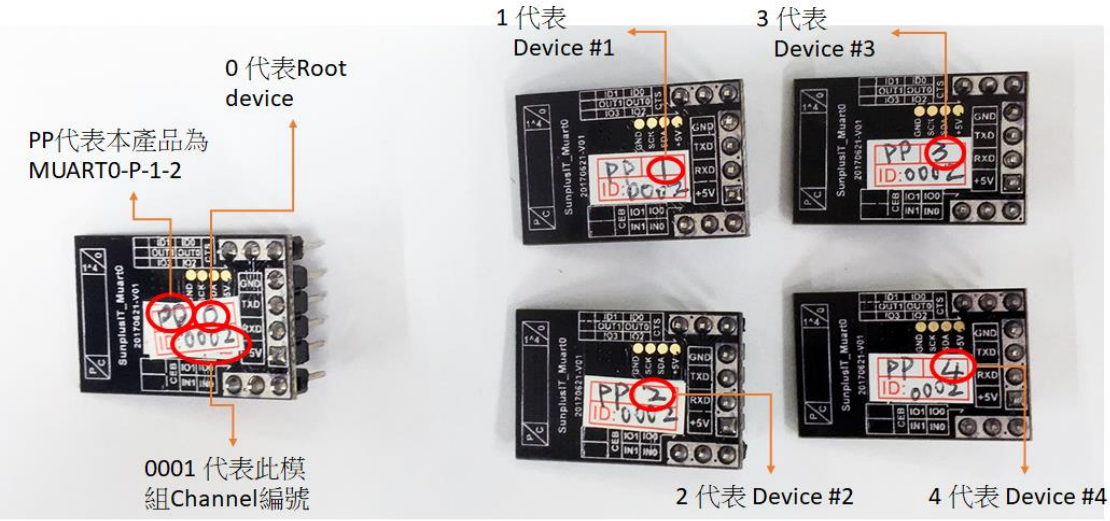
MUART0-PP-1-N 系列為 SunplusIT 最新推出的一對多無線 Uart 傳輸模組，支援最多高達 8 個 UART 介面，它能將傳統 1 對 1 的 UART 連接埠輕易升級至最多 1 對 8 的無線 UART 傳輸，此外還支援單向的廣播模式，以及二組可相互溝通的 I/O pin，讓您在開發各種裝置功能時，更加的得心應手。

模組外觀

MUART0-PP-1-N 支援最多 8 組 devices，每個模組的外觀相同，但可由背面的標籤來識別 Root、Device 及所屬編號，以下以 1:4 及 1:8 為例說明。

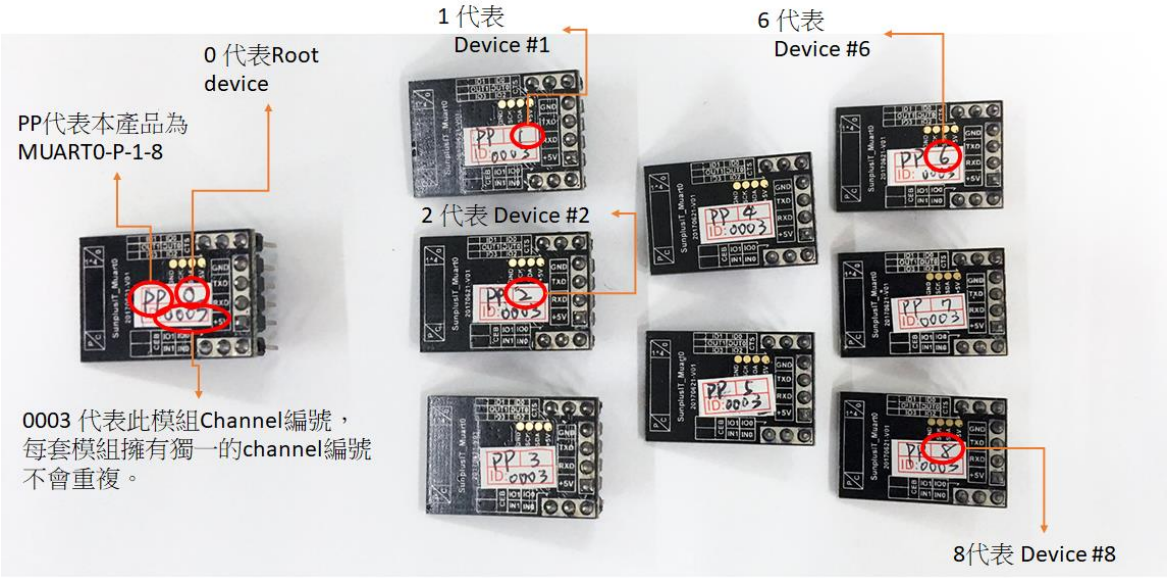
一對四 1:4

1:4 表示 root 可連接四個 UART device，共有五個 IC 模組，分別為 Root 端（編號 PP0）一片以及 Device 端（編號 PP1~PP4）四片。



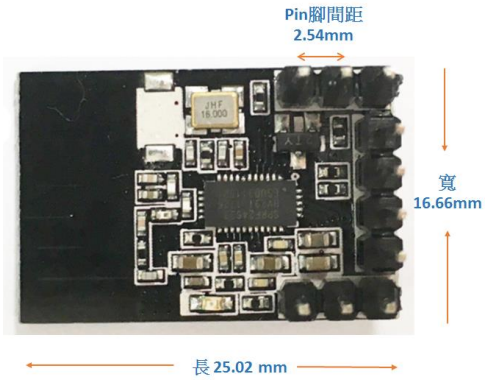
一對八 1:8

1:4 表示 Root 可連接八個 UART device，共有九個 IC 模組，分為 Root 端（編號 PP0）一片以及 Device 端（編號 PP1~PP8）八片。

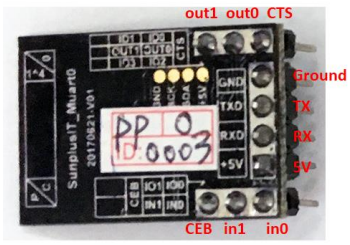
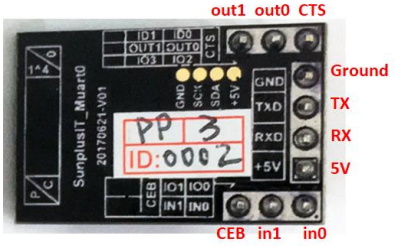


模組尺寸

MUART0-PP-1-N 系列模組尺寸為 25.02x16.66(mm)，pin 腳間距為標準的 2.54mm。



PIN 腳功能及動作

Root	Device
	

CTS	CTS 腳位的用途為廣播模式的切換，Root 端可透過接地來選擇是否 enable 廣播模式。 **CTS 未接** →指定連接模式，root device 可指定要與那一個 device 相連，此模式為雙向傳輸，root 與 device 皆能相互對傳資料。 **CTS 接地**→開啟廣播模式，在此模式下，所有的 device 皆會收到相同的訊息，此廣播模式為單向，root 與 device 分別僅能傳送及接收。	CTS 腳位目前在 Device IC 沒有功能。
CEB	接到 Ground 模組才會運作。 您可利用此 pin 透過程式來 enable 及 disable 模組的運作。	
IN1/IN2 及 OUT1/OUT2	IN pin 空接 →另一端 IC 相對應的 OUT pin 輸出高電位（3~3.3V）。 IN pin 接地 →另一端 IC 相對應的 OUT pin 輸出低電位（0~0.2V）。 此 I/O 功能於**指定連接**或**廣播**兩種模式下皆適用。 **廣播模式**：所有的 device 會同步收到 OUT 訊息。 **指定連接模式**：只有連接的 device 會收到 OUT 訊息。	
TX/RX	**TX** → 對應到 Uart/Serial 介面的 RX **RX** → 對應到 Uart/Serial 介面的 TX	

模組特性及規格

特性

MUART0-PP-1-N 系列模組支援的 baud rate 為 9,600，過電後，若 root 端的 CTS pin 未接地則預設會與 Device #1 自動連線，若 root 端的 CTS pin 腳接地，則 root 端會以廣播方式將 Uart 或 I /O port 資訊傳送至所有的 devices。

規格

1. **每套皆有兩種模組**：分別為 Root 及 Device，Root device 固定一個，device 數量視型號而定。
2. **操作電壓**：3.3~5.5V

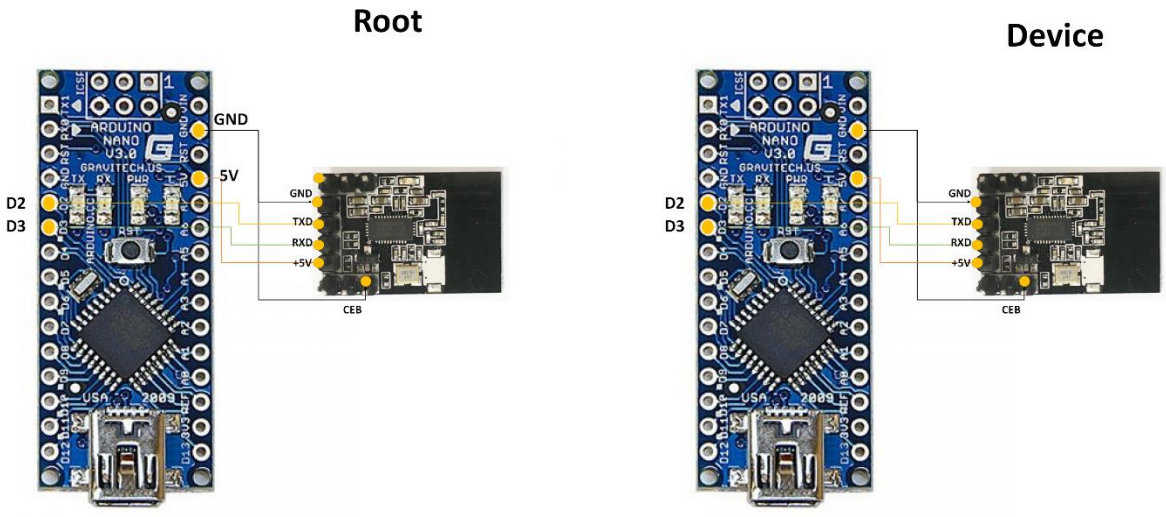
3. **RF 頻率**：2400MHz~2480MHz。
4. **耗電量**：傳送約 24mA@+5dBm，接收約 23mA。
5. **發射功率**：+5dBm
6. **傳輸速率**：250Kbps
7. **傳輸距離**：空曠處約 80~100m
8. **Baud rate**：9,600bps
9. **傳輸功能**：可選擇**指定連接模式**或**廣播模式**。

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。下面以 Android 開發板為例作說明，圖示使用的是 Nano，但其它型號的 Arduino 板子也適用。

指定連接模式（模組的 **CTS pin** 腳不接）

此模式下，Root 端可透過發送指令來選擇與那一個 device 連接。線路接法如下：



Arduino Nano/UNO	**MUART0-PP-1-N 系列模組**
GND	GND
D2	TXD
D3	RXD
5V	5V+
GND	CEB

Arduino 程式範例

下面的程式示範 MUART0-PP-1-N 模組的 root 端如何傳送字串"Root-01234"到 device #3，並接收 device 傳回的訊息。注意使用的 RX/TX 對應 pin 腳並沒有限制為 D2/D3，您可以使用其它的腳位。

Root 端

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

byte cmdTarget[4] = { 0x55, 0x01, 0x03, 0X03 };   // Device #3 的 ID
byte cmdStandby[4] = { 0x55, 0x02, 0x00, 0X0d };
String msg = "Root-01234";
boolean lineBreak = 0;

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

void loop() {
  Serial.println("Connect to device #3");
  mySerial.write(cmdStandby, sizeof(cmdStandby));
  delay(20);
  mySerial.write(cmdTarget, sizeof(cmdTarget));
  delay(20);

  mySerial.print(msg);
  Serial.println("Send: " + msg);
  delay(50);

  //接收來自 device 的訊息
  mySerial.listen();
  String rcvmsg = "";
  while (mySerial.available() > 0) {
    byte inByte = mySerial.read();
    Serial.print(char(inByte));
    rcvmsg += char(inByte);
    lineBreak = 1;
  }
  if (lineBreak == 1) {
    Serial.println();
    lineBreak = 0;
    Serial.println("Received from device: " + rcvmsg);
  }
  delay(1000);
}
```

各編號 **device** 的 **16bits ID** 如下：

Device	*ID (16bits)*
Device #1	{ 0x55, 0x01, 0x01, 0X0d }
Device #2	{ 0x55, 0x01, 0x02, 0X02 }
Device #3	{ 0x55, 0x01, 0x03, 0X03 }
Device #4	{ 0x55, 0x01, 0x04, 0X00 }
Device #5	{ 0x55, 0x01, 0x05, 0X01 }
Device #6	{ 0x55, 0x01, 0x06, 0X06 }
Device #7	{ 0x55, 0x01, 0x07, 0X07 }
Device #8	{ 0x55, 0x01, 0x08, 0X04 }

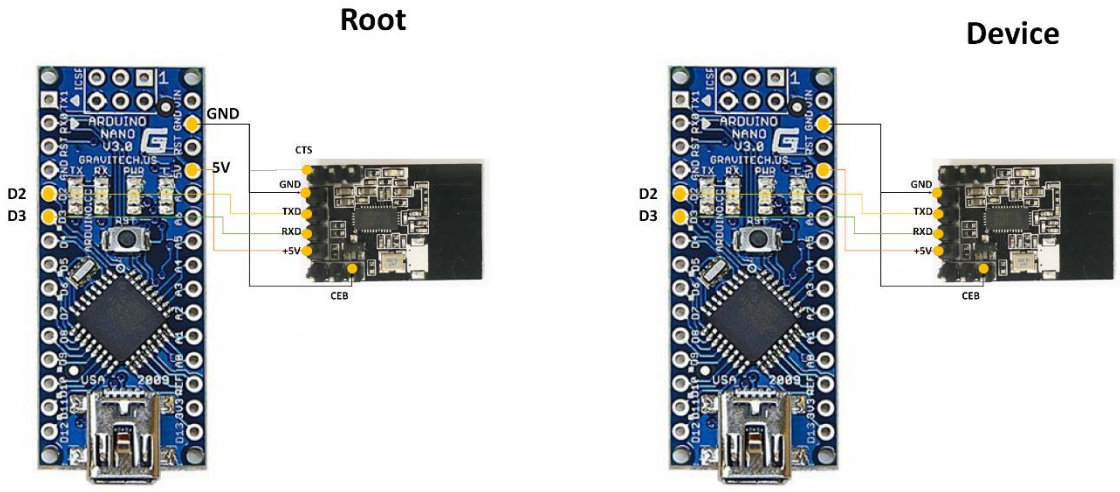
Device 端

```
#include <SoftwareSerial.h>
SoftwareSerial myRX(2, 3); // RX, TX
boolean lineBreak = 0;
String msg = "abcdefghij";

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

void loop() {
  myRX.print(msg);
  Serial.print("Send to Root: ");
  Serial.println(msg);

  myRX.listen();
  String rcvmsg = "";

  while (myRX.available() > 0) {
    byte inByte = myRX.read();
    //Serial.print(char(inByte));
    rcvmsg += char(inByte);
    lineBreak = 1;
  }

  if (lineBreak == 1) {
    Serial.println("Received from Root: " + rcvmsg);
    Serial.println();
    lineBreak = 0;
  }
```

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

廣播模式（模組的 CTS pin 接到 Ground）

此模式下，Root 所發送的訊息將送至所有的 device，Root 端不需指定 device ID。線路接法類似，但 root 端的 CTS 需接地：



Arduino Nano/UNO	MUART0-PP-1-N 系列模組
GND	GND
D2	TXD
D3	RXD
5V	5V+
GND	CEB
CTS	CTS（Root 端）

Arduino 程式範例

在下方的示範程式中，您會注意到，雖然 Root 端與 Device 端分別進行了發送及接收動作，但 console 中 Root 端只有送出而沒有接收到資料，而 Device 端有接收到 Root 端的資訊但送出的訊息卻沒有被 Root 端所接收，這是因為在廣播模式下僅支援單向的傳輸，即 Root → Device。

Root 端

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

String msg = "Root-01234";
```

```
boolean lineBreak = 0;

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

void loop() {
  Serial.println("Send: " + msg);
  mySerial.print(msg);
  delay(50);

  //接收來自 device 的訊息
  mySerial.listen();
  String rcvmsg = "";
  while (mySerial.available() > 0) {
    byte inByte = mySerial.read();
    Serial.print(char(inByte));
    rcvmsg += char(inByte);
    lineBreak = 1;
  }
  if (lineBreak == 1) {
    Serial.println();
    lineBreak = 0;
    Serial.println("Received from device: " + rcvmsg);
  }
  delay(1000);
}
```

Device 端

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

String msg = "Device-01234";
boolean lineBreak = 0;

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

void loop() {
  Serial.println("Send: " + msg);
  mySerial.print(msg);
  delay(50);

  //接收來自 root 的訊息
```

```
  mySerial.listen();
  String rcvmsg = "";
  while (mySerial.available() > 0) {
    byte inByte = mySerial.read();
    rcvmsg += char(inByte);
    lineBreak = 1;
  }
  if (lineBreak == 1) {
    Serial.println();
    lineBreak = 0;
    Serial.println("Received from root: " + rcvmsg);
  }
  delay(1000);
}
```