MUARTO-P-1-2

雙管齊下 同時無線連接兩個 UART 裝置



目錄

2
5

MUARTO-P-1-2 無線 Uart 傳輸模組是 MUARTO-S-1-1 的升級版,它能將傳統1對1的UART 連接埠輕易的升級為1對2 無線 UART 傳輸,此外還提供二組 I/O pin,讓您不需要透過昂貴的 多 Uart 埠開發板,就能同時連接兩組 Uart 裝置,以及使用可遠端操控的 IO 開關。

模組外觀

MUARTO-P-1-2 模組外觀與 MUARTO-S-1-1 相同,三個一組,包含 Device 端(編號 P1) 兩片及 Root 端(編號 P0)一片,雖然其外觀皆相同,但可由背面的標籤來識別(P0 為 Root, P1/P2 為 Device #1/#2)。



模組尺寸

MUARTO-P-1-2 模組尺寸為 25.02x16.66(mm), pin 腳間距為標準的 2.54mm。



2

PIN 腳功能及動作

	Root	Device #1/#2
	P0 - 代表本模组为Root湖 outlout0 CTs outlout0 CTs outlout0 CTs outlout0 CTs outlout0 CTs outlout0 CTs outlout0 CTs outlout0 CTs forund Tx Rx sv sv cEB in1 in0	P1 → 代表本横組為Device #1端 P2 → 代表本横組為Device #2端 outl outl CTS outl outl CTS From United TS CEB in1 in0
CTS	CTS 腳位在 Root 端可透過接地來選擇 與那個 device 連接。	CTS 腳位目前在 Device IC 沒有功能。
	CTS 未接 → Device #1 CTS 接地 → Device #2	
CEB	需接到 Ground,IC 才能運作。	
IN1/IN2 及	IN pin 空接 →另一端 IC 相對應的	OUT pin 輸出高電位(3~3.3V)。
OUT1/OUT2	IN pin 接地 →另一端 IC 相對應的	OUT pin 輸出低電位(0~0.2V)。
TX/RX	TX → 對應到 Uar	t/Serial 介面的 RX
	RX → 對應到 Uar	t/Serial 介面的 TX

模組特性及規格

MUARTO-P-1-2 模組 baud rate 為 9,600,過電後,Root 端預設會先與 Device #1 自動連線。

- 1. 每套三個模組:分別為 Root、Device #1, Device #2
- 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

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。以下分別以 Android、樹莓派兩種開發板為例來說明:

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 以軟體模擬 Uart 方式來使用,以避免佔用實體的 UART 介面。下方範例為透過 software serial,將 Arduino Nano 的 D2、D3 pin 腳模擬為 Serial RX 與 TX,D13 腳位(不限 D13, 您亦可定義其它的腳位)接到 MUARTO-P-1-2 模組的 CTS pin,接下來我們便可透過程式來 控制 D13 輸出高電位 HIGH 或低電位 LOW 來選擇欲連接的 Device 編號(LOW→Device #1 / HIGH→Device #2)。除了本範例使用的 Nano,其它型號的 Arduino 板子也適用。



Arduino Nano/UNO	MUART0-P-1-2 模組
GND	GND
02	TXD
03	RXD
5V	5V+
GND	CEB
CTS	CTS (Root 端透過 CTS 輔入訊號來決定 與那個 device 連接。 HIGH → Device #1 LOW → Device #2)

Arduino 程式範例:

下面的程式示範 MUARTO-P-1-2 模組的 root 端如何傳送字串"Root-01234"到指定的 device #1 或 device #2,並接收來自目前連接 device 傳回的訊息。(使用的 Arduino RX/TX 及 CTS 對應 pin 腳並沒有限制為 D13,您可以更改為其它的腳位。)

```
Root 端:每隔五秒切換一次(連到 device #1 或 device #2)
```





Device #1, #2 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.listen();
  String rcvmsg = "";
  Serial.print("Send to Root: ");
  myRX.print(msg);
  Serial.println(msg);
```

```
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);
```



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,如下圖,您只要將 MUARTO-P-1-2 模組各腳位接 到樹莓派上對應的腳位,即可如傳統的 UART 使用方式一樣操作,而且還能透過 GPIO 17 來選 擇要連到那個 device。(GPIO14 及 GPIO15 為樹莓派的 hardware TX/RX ports,無法改變。 GPIO17 為我們自行定義 for device 選擇的 CTS 腳位,腳位可自行更改)。

樹莓派程式範例

Root 端:每隔二秒切换一次(GPIO17 依序的輸出 LOW 與 HIGH,分別連到 device #1 及 device #2)



程式範例(Root 端):

#!/usr/bin/env python import time

```
import serial
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
pinDevice = 17 #使用此腳位來選擇 device
ser = serial.Serial(
    port='/dev/ttyS0',
    baudrate = 9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=1
GPIO.setup(pinDevice, GPIO.OUT)
counter=0
while 1:
    if(counter \% 2 == 0):
        GPIO.output(pinDevice, 1)
    else:
        GPIO.output(pinDevice, 0)
    time.sleep(0.5)
    x=ser.readline()
    print (x)
    ser.write('Write counter: %d \n'%(counter))
    time.sleep(2)
```



#!/usr/bin/env python import time import serial import RPi.GPIO as GPIO GPIO.setmode(GPIO.BCM)

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

```
GPIO.setup(pinDevice, GPIO.OUT)
counter=0
while 1:
#續取資料
x=ser.readline()
print (x)
#寫入資料
ser.write('Write counter: %d \n'%(counter))
time.sleep(2)
```