



低功耗 藍牙 BLE

我是誰

- 吳奇峯 (Wright)
- 20年半導體電子
 - 硬體工程師
 - 韌體工程師
 - 產品 PM
- 就喜歡搞東搞西
- 甚麼都學
 - 感應器
 - 組合 / C / Python



Wi - Fi

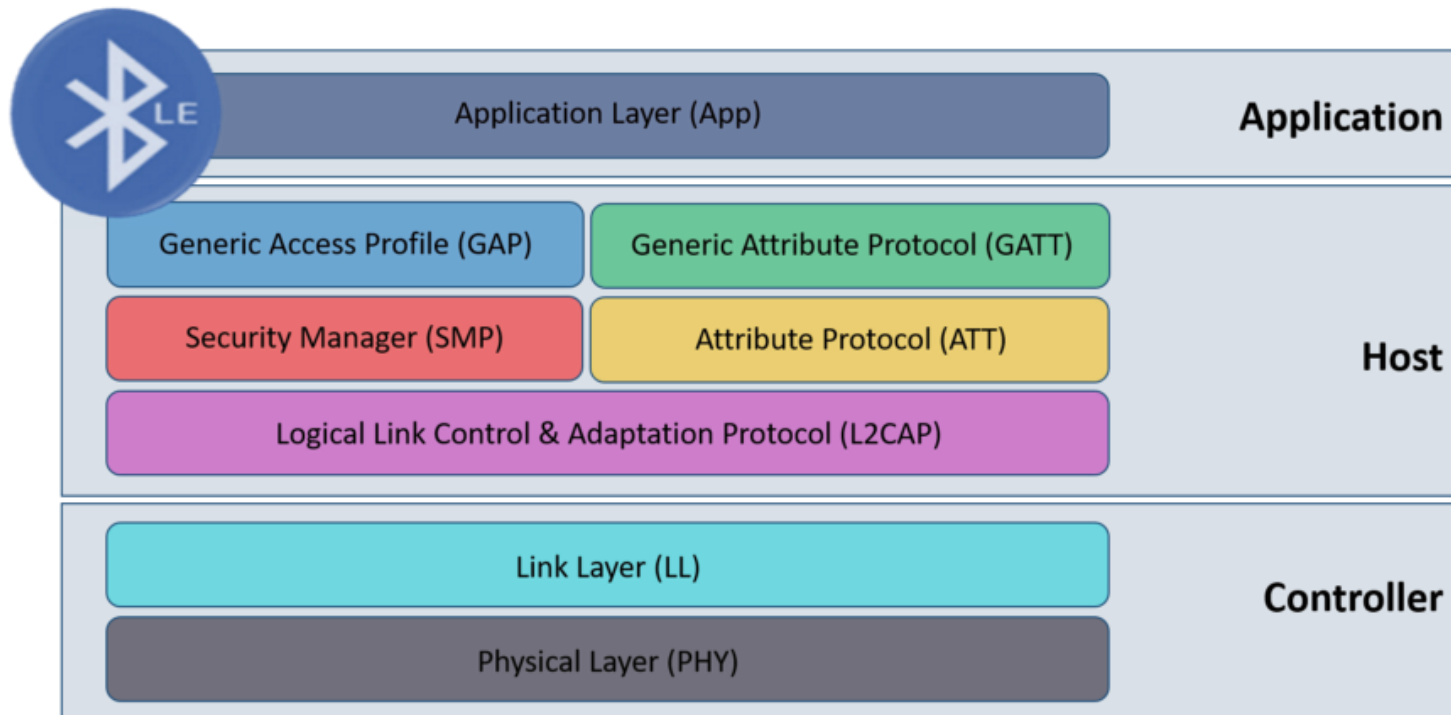
SSID : IoT Service Hub

Password : 05076416

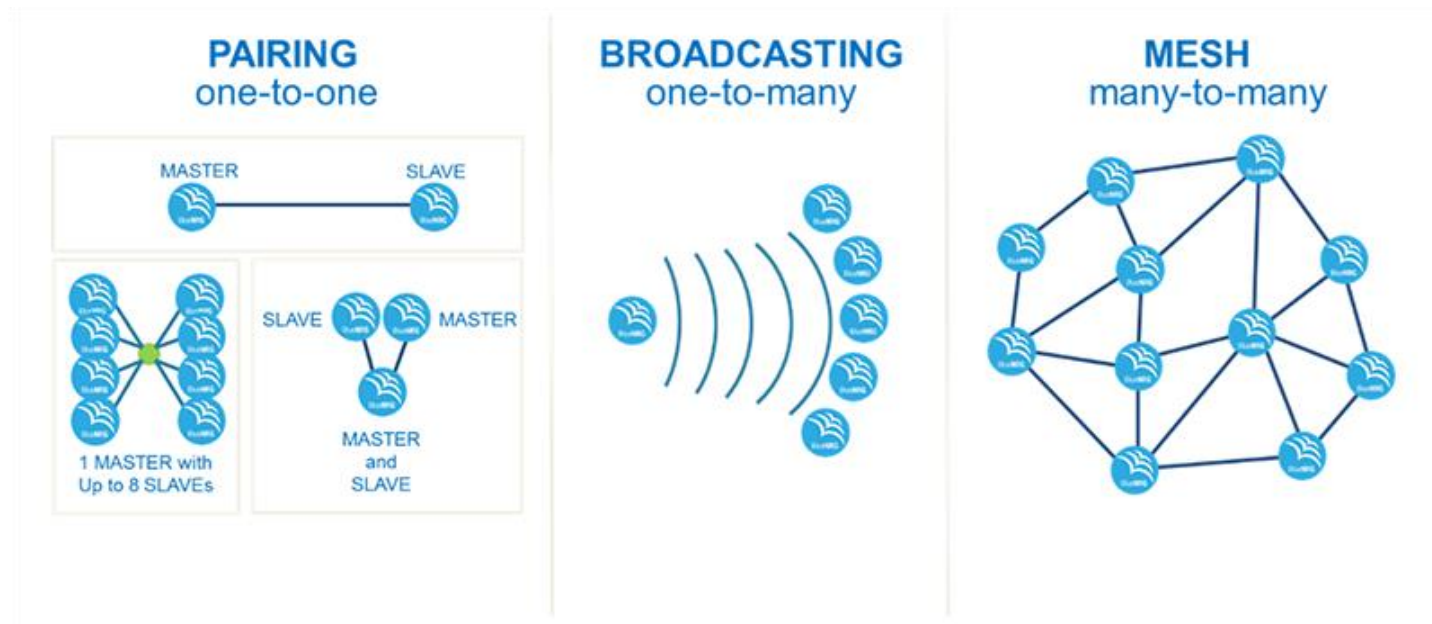
藍牙技術變遷歷史

	藍牙版本	發布時間	最大傳輸速度	傳輸距離
	藍牙5.3	2021	48 Mbit/s	300公尺
	藍牙5.2	2020	48 Mbit/s	300公尺
	藍牙5.1	2019	48 Mbit/s	300公尺
	藍牙5.0	2016	48 Mbit/s	300公尺
	藍牙4.2	2014	24 Mbit/s	50公尺
	藍牙4.1	2013	24 Mbit/s	50公尺
BLE →	藍牙4.0	2010	24 Mbit/s	50公尺
	藍牙3.0+HS	2009	24 Mbit/s	10公尺
	藍牙2.1+EDR	2007	3 Mbit/s	10公尺
	藍牙2.0+EDR	2004	2.1 Mbit/s	10公尺
	藍牙1.2	2003	1 Mbit/s	10公尺
	藍牙1.1	2002	810 Kbit/s	10公尺
	藍牙1.0	1998	723.1 Kbit/s	10公尺

BLE protocol Stack

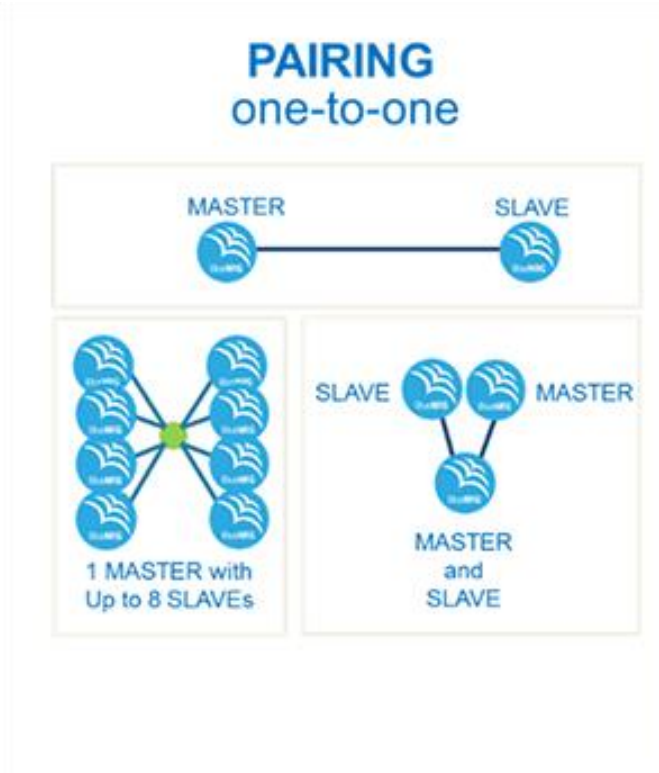


藍牙低功耗網路拓撲



One to One

- GATT



RL62M Module



UART (TX,RX)
Buad rate =115200 bps



RL62M01 介紹

- BLE 5.0 GATT Profile (與 BT 2.0 SPP Profile 不同)
 - GATT AT Command 透傳模式
 - 支援 Master 或 Slave Mode
- 低功耗
 - power down mode 450nA
 - TX 8.4mA
 - Rx 6.8mA
- 2Mbps 高速傳輸模式
- OTA 韌體更新
- 通過 Radio 認證 FCC, CE, TELEC Japan,
- 通過 安全認證 IEC/EN 62368-1
- 1.05mm(W) x 17mm(L) x 2.3mm(H)



RL62M01 GATT AT Command 功能

- Command Mode \leftrightarrow Data Mode
- Server (從機 Slave) / Master (主機 Client)
- 自訂義 廣播封包 (應用 NFC ，傳給多主機)
- 調整TX 功率 0db ~ 8db
- 主機 模式 – 掃描 與 連線 / 斷線
- 從機 模式 – 主動斷線
- 資料自動分包
- 多種 UART BaudRate :
1200/2400/9600/38400/57600/115200/230400/..

如何使用 RL62M01 Module

- <https://www.richlink-tech.com>

Data Sheet

RL62M01A Data Sheet

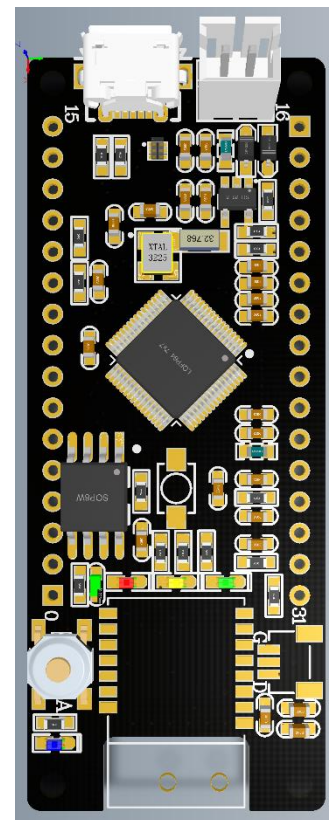
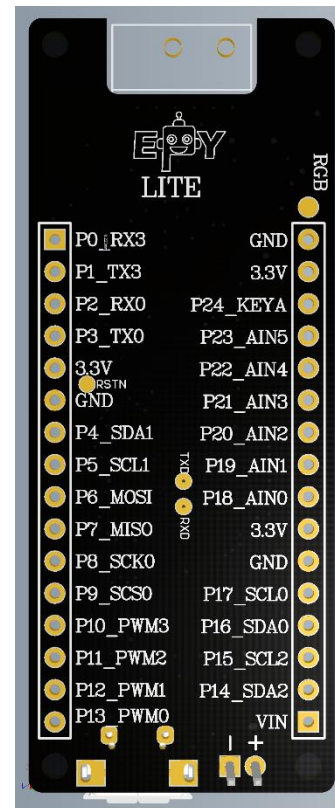
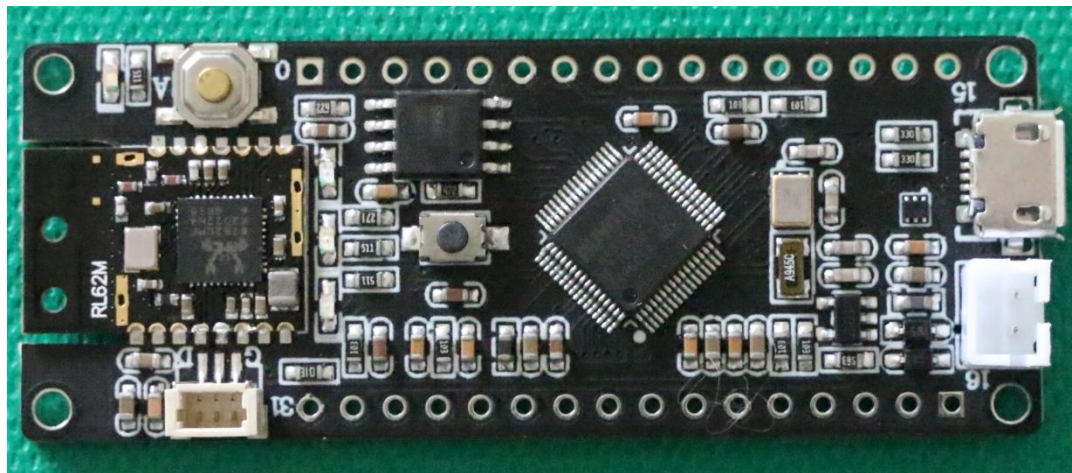
RL62M02A Data Sheet

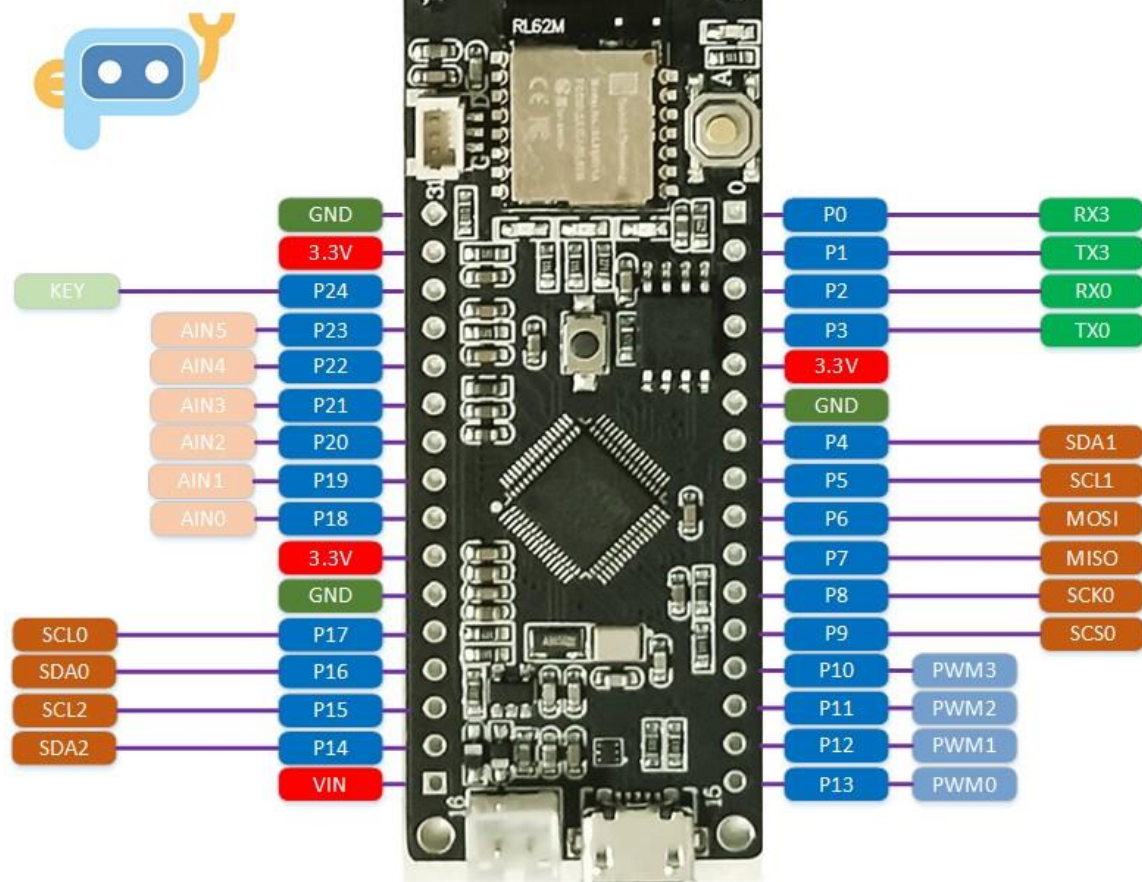
Programming Guide

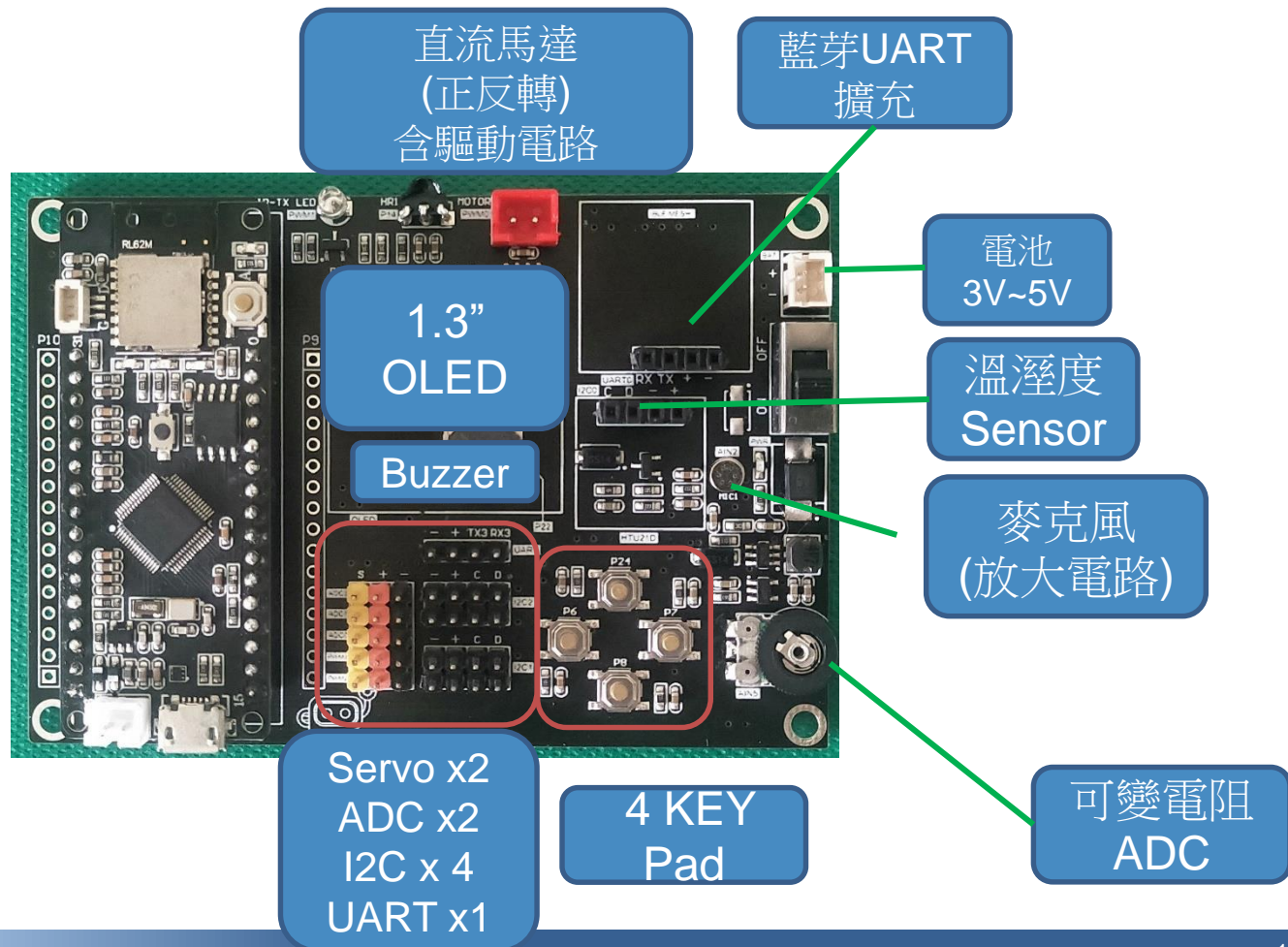
RL62M01A Programming Guide

RL62M02A Programming Guide

ePy Lite







十分鐘 馬上會用 BLE

- 提供 Micro Python 下的 UART RL62M 程式庫 (Class)
 - RL62M Copy ePy Lite 內置 Flash 即可
- RL62M Library API (5 個)
 - 匯入 RL62M Library (import RL62M)
 - 設定腳色 (主機，從機) -- (BLE = RL62M.GATT(uart_port,role=" ")
 - 掃描連線 -- BLE.ScanConnect (mac= ' ', name_head=' ')
 - 傳輸資料 -- BLE.SendData ('傳輸的資料')
 - 接收資料 -- data = BLE.RecvData()
 - 斷線 -- BLE.disconnect()

```

from machine import UART, delay
import RL62M

uart = UART(1, 115200, timeout=200, read_buf_len=512)

BLE = RL62M.GATT(uart, role='CENTER')
BLE.ScanConnect(mac='7002000008B6')
BLE.ScanConnect(name_header = 'EPY_') # wait 5sec

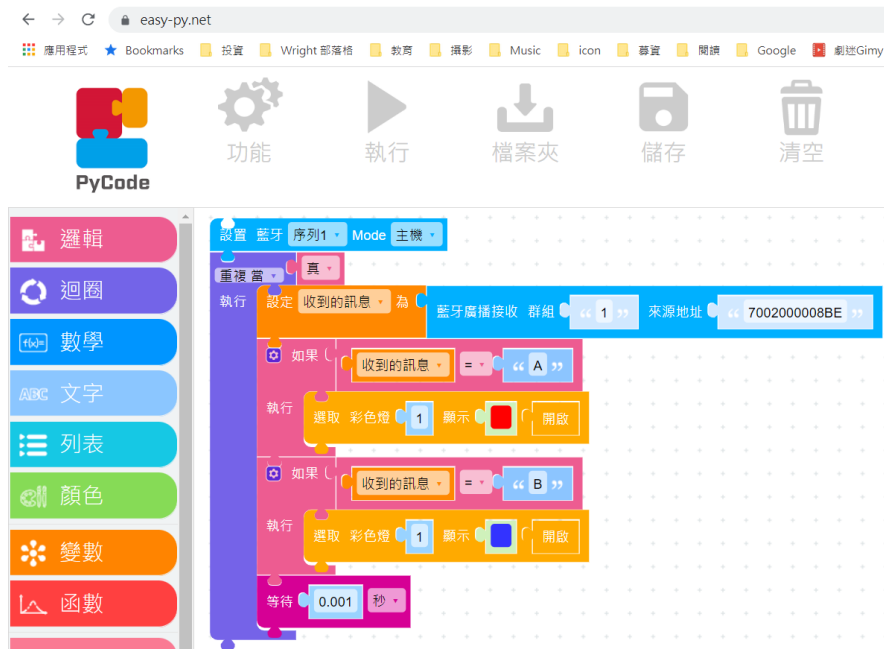
while True: # wait be connected
    BLE.RecvData()
    if BLE.state == 'CONNECTED':
        break
    else :
        delay(100)

while True: # send / recv data
    m = BLE.RecvData()
    if BLE.state == 'DISCONNECTED':
        break
    else :
        BLE.SendData('ABC')
        delay(100)

```

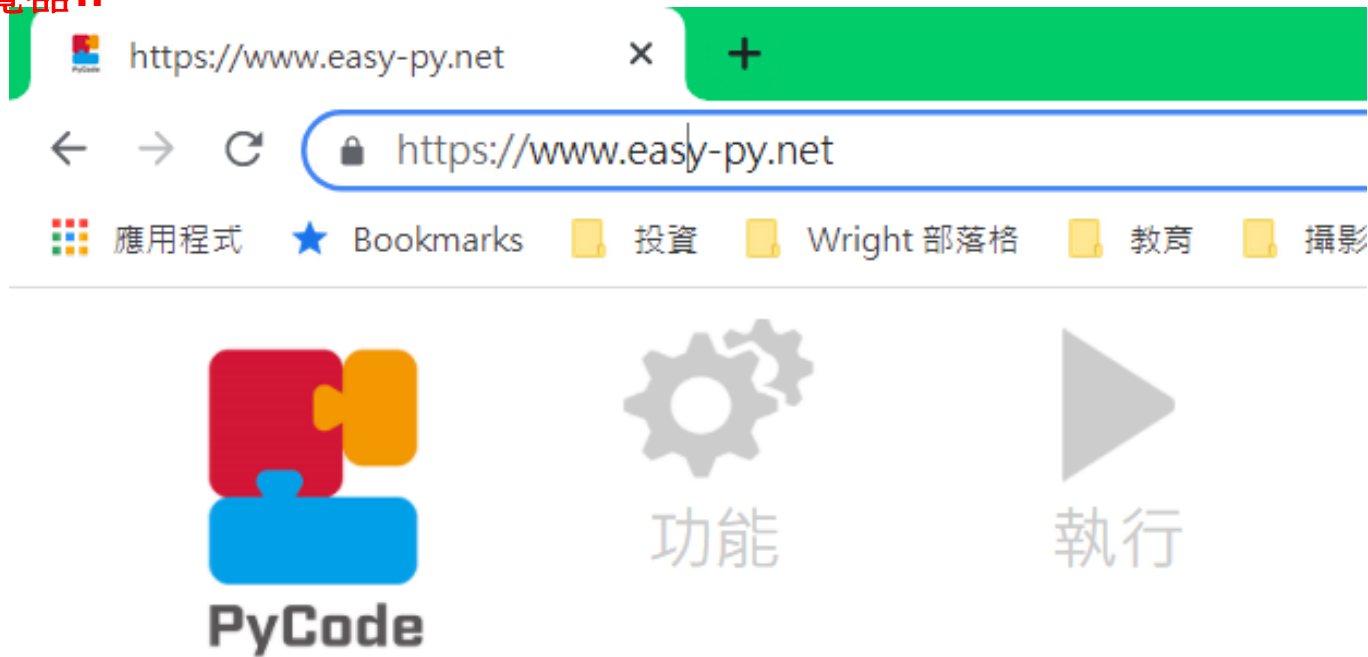

PyCode 怎麼用

- <https://www.easy-py.net>



線上 PyCode

使用 **Chrome** 瀏覽器 !! 使用 **Chrome** 瀏覽器 !! 使用 **Chrome** 瀏覽器!!



```
1 from machine import LED
```



功能



執行



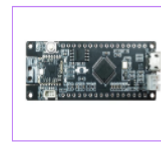
檔案夾



儲存



清空



ePy-Lite小拍



邏輯



迴圈



數學



文字



列表



顏色



變數



函數



主機板



按鍵

重複 10 次

執行

使用 i 從範圍 1 到 5 每隔 1

執行

選取 彩色燈 i 顯示 開啟

等待 1 秒

關閉 彩色燈

```
1 from machine import LED
2 import utime
3 from machine import RTC
4
5 i = None
6 ledRgb = None
7 rtc = None
8
9
10 ledRgb = LED(LED.RGB)
11 rtc = RTC()
12 for count in range(10):
13     for i in range(1, 6):
14         ledRgb.rgb_write(int(i),255,0,0)
15         utime.sleep( 1 )
16         ledRgb.off()
17
```

清除

狀態 : Web USB

EPY: soft reset

MicroPython v1.10 on 2021-07-23; EPy-Lite v1.7-10-gda4d258

>>> OK>OK

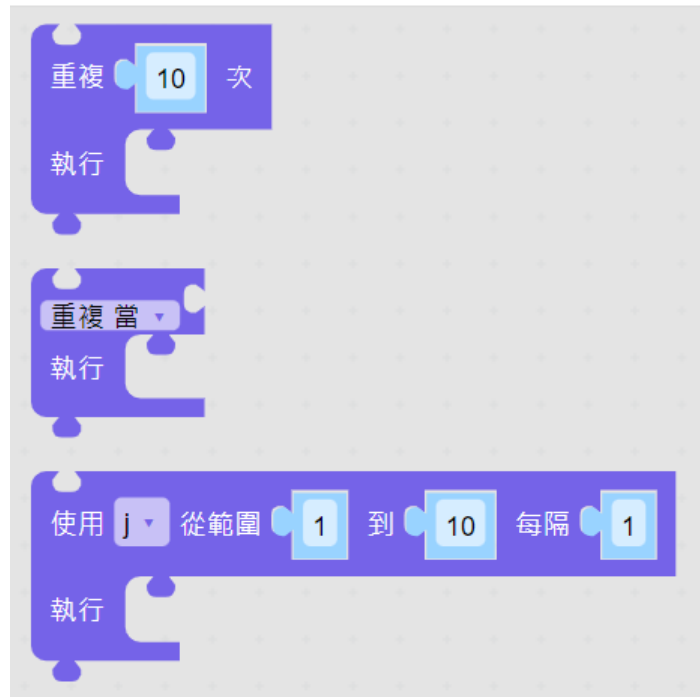
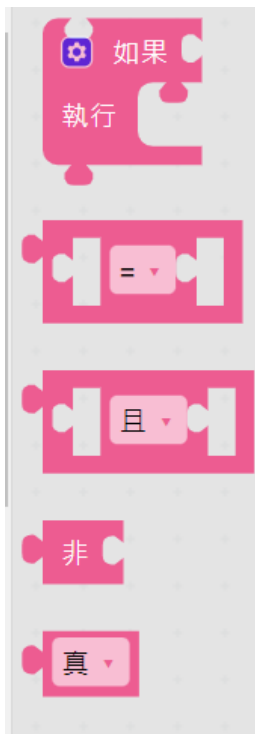
EPY: soft reset

raw REPL; CTRL-B to exit

>OK>



基礎邏輯



練習



主機板



LED燈

設置 燈R ▼ 為 開 ▼

設置 燈R ▼ 為 開 ▼

- ✓ 燈R
- 燈Y
- 燈G

設置 燈R ▼ 為 開 ▼

設置 燈Y ▼ 為 開 ▼

設置 燈G ▼ 為 開 ▼

等待 1 秒 ▼

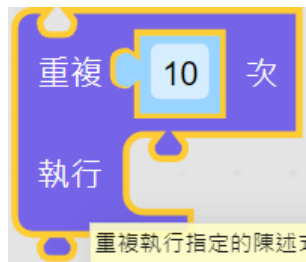
設置 燈R ▼ 為 關 ▼

設置 燈Y ▼ 為 關 ▼

設置 燈G ▼ 為 關 ▼

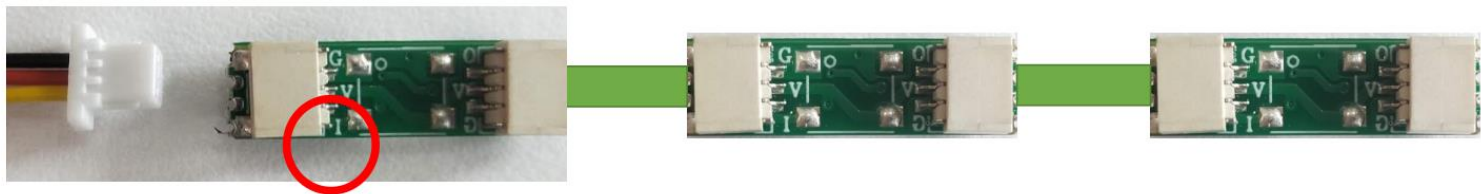
等待 1 秒 ▼

迴圈



RGB LED

- 注意連接方法



接 ePy Lite



主機板



LED燈

選取 彩色燈

1

顯示



開啟

關閉

重複

10

次

執行

選取 彩色燈

1

顯示



開啟

等待

1

秒

選取 彩色燈

1

顯示

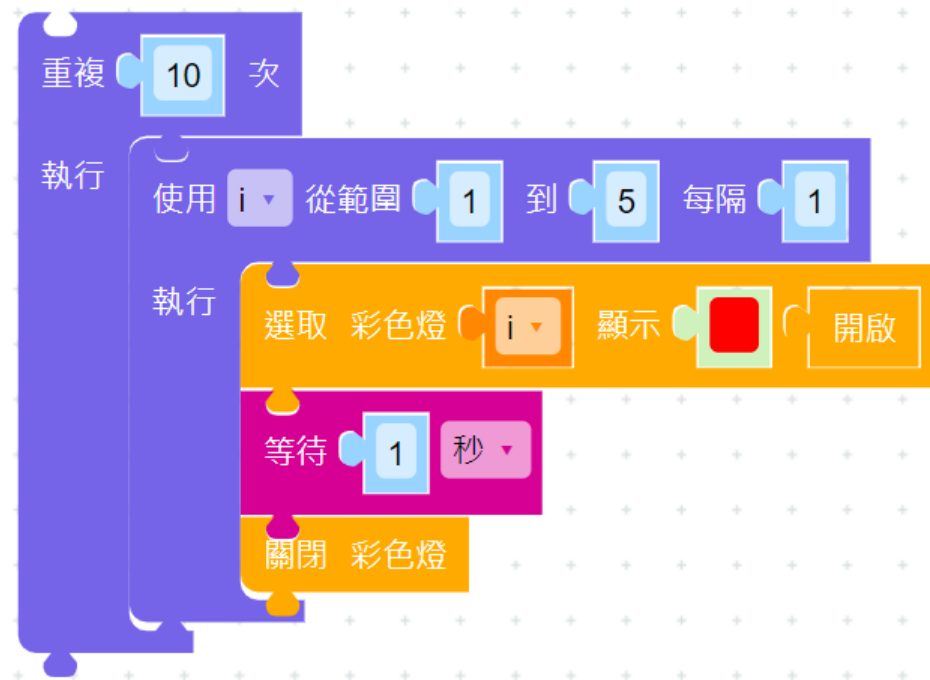


關閉

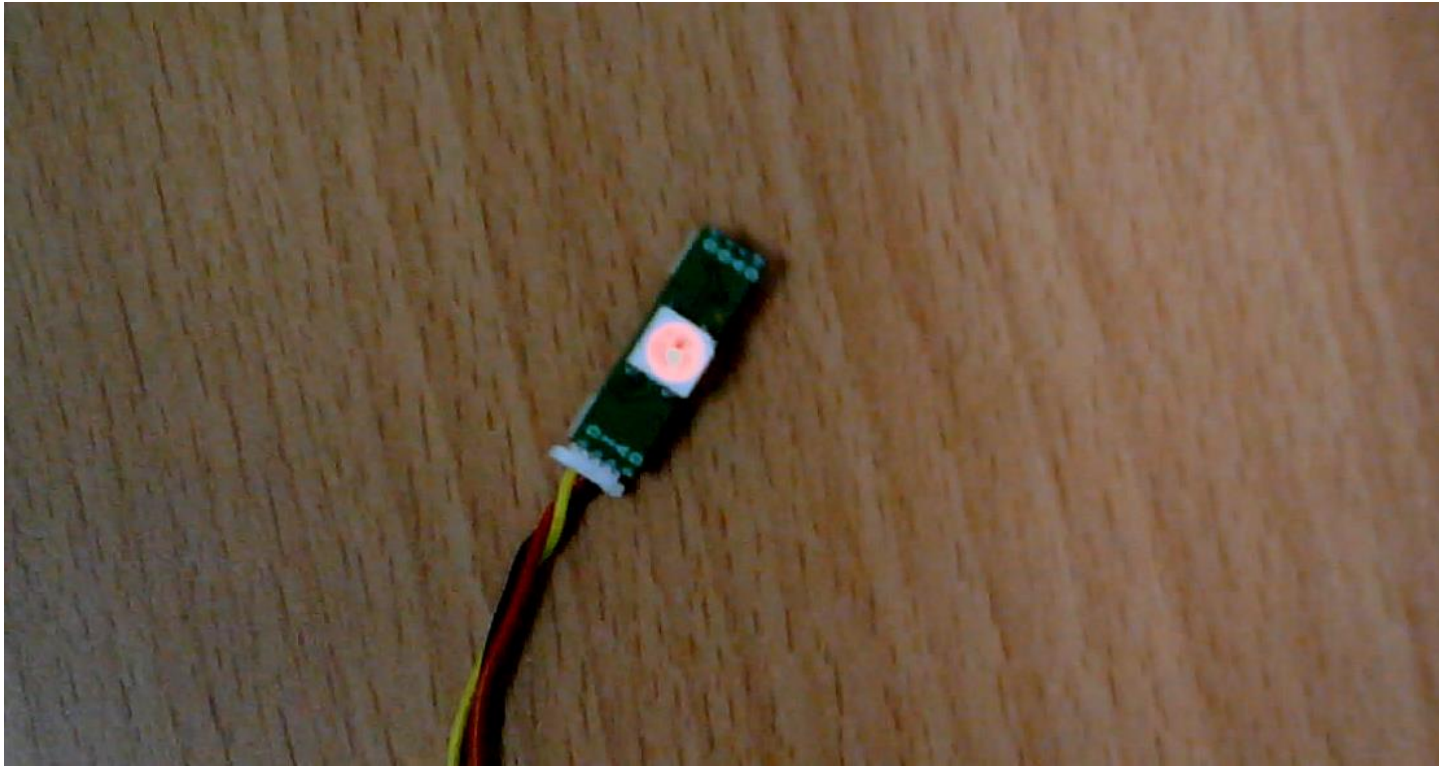
等待

1

秒



呼吸燈 怎麼做？



重複 10 次

執行

使用 紅色 從範圍 0 到 255 每隔 10

執行

選取 彩色燈 1 顯示 顏色 紅 紅色 開啟

綠 0

藍 0

等待 0.1 秒

使用 紅色 從範圍 255 到 0 每隔 -10

執行

選取 彩色燈 1 顯示 顏色 紅 紅色 開啟

綠 0

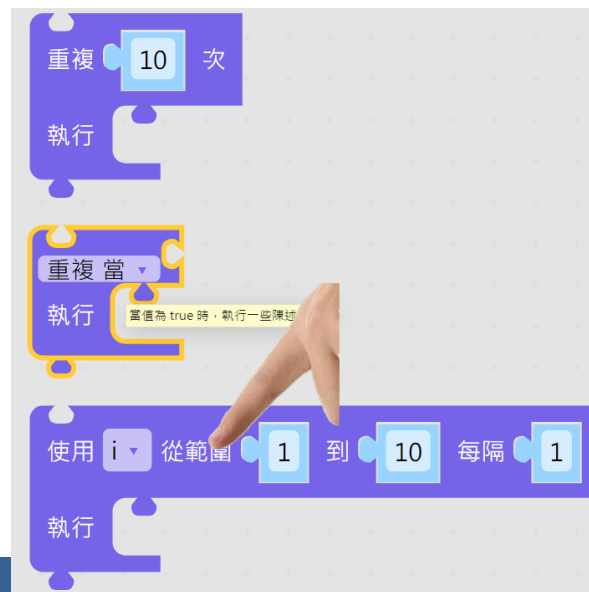
藍 0

等待 0.1 秒

顏色積木在哪？



變數迴圈





新變數名稱：

紅色

確定 取消

This dialog box prompts for a new variable name. The text '紅色' (Red) is entered in the input field. There are '確定' (Confirm) and '取消' (Cancel) buttons at the bottom.



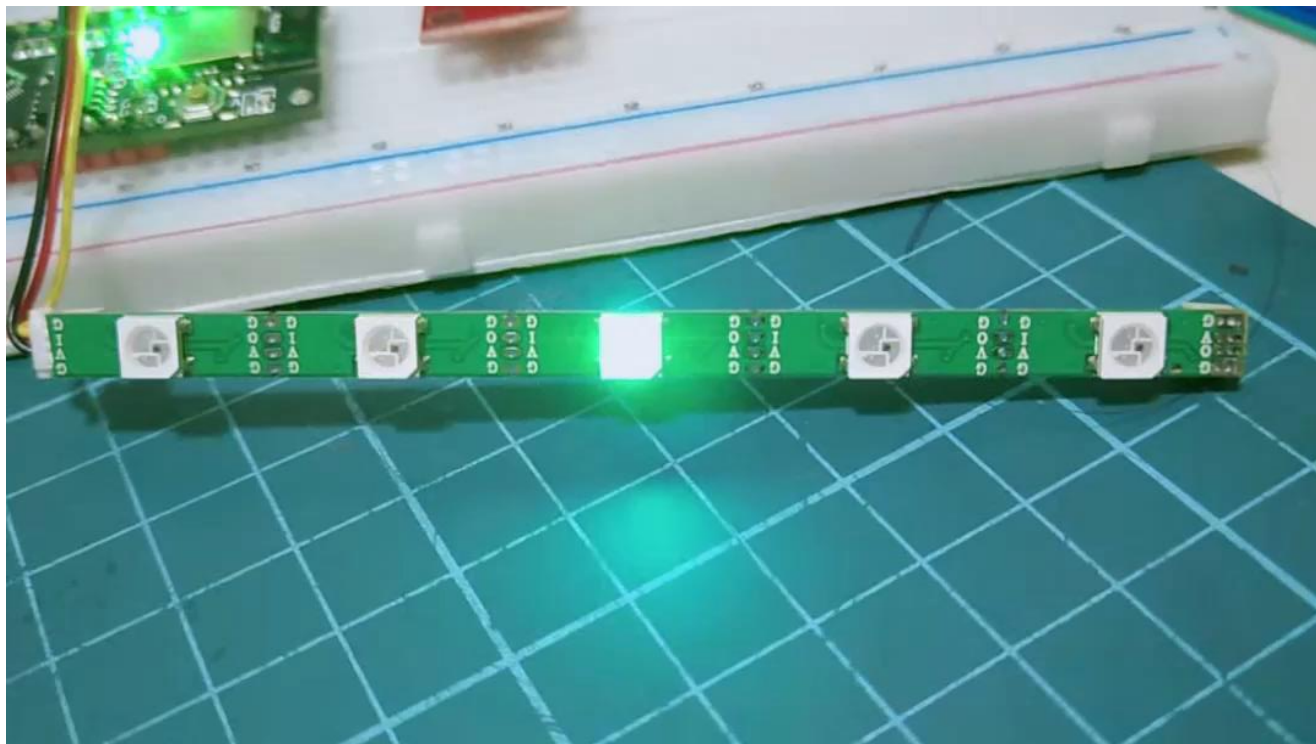


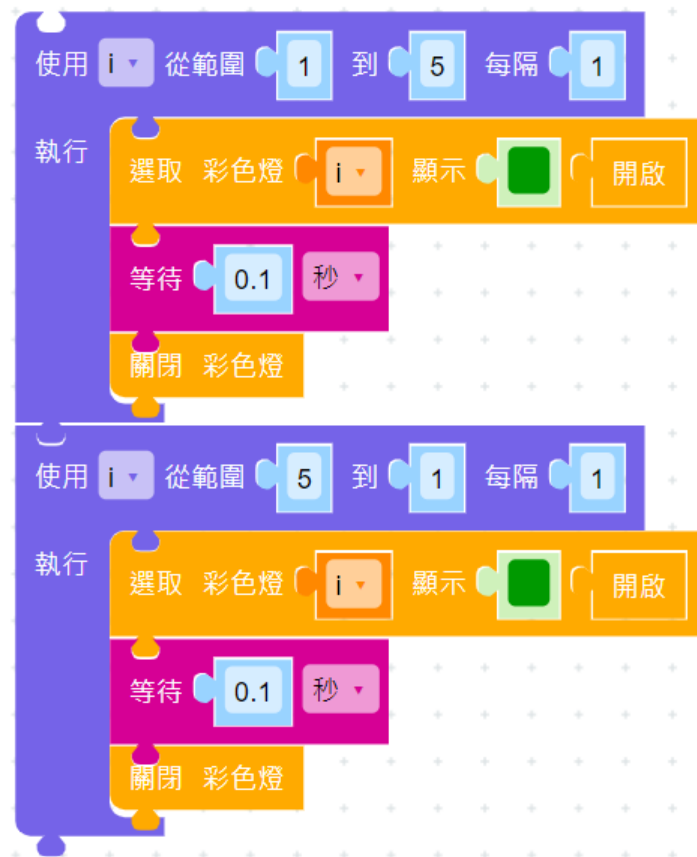






一起來做看看





Blockly \leftrightarrow Python

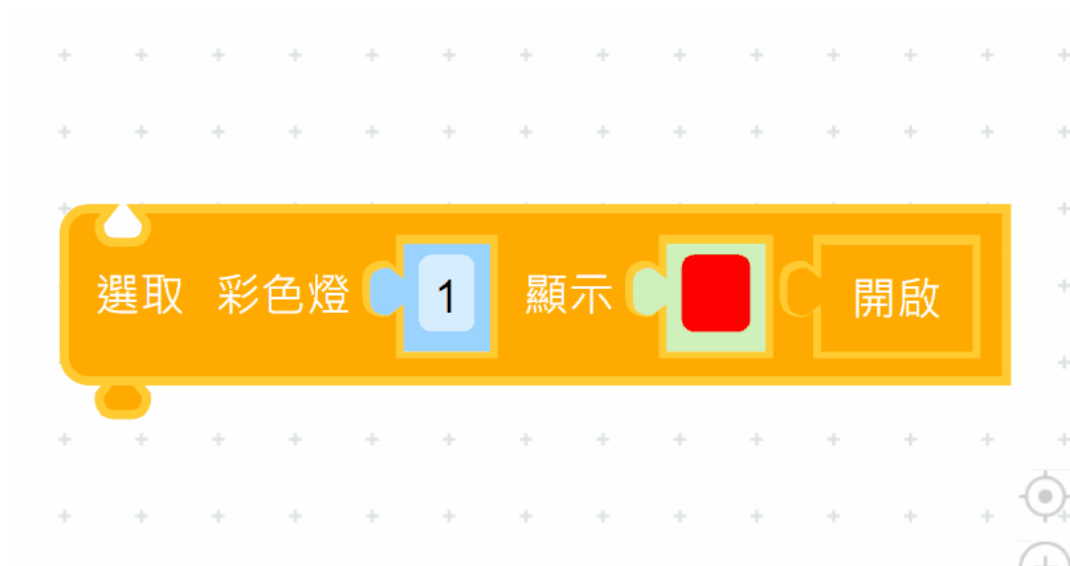


```
from machine import LED
import utime
from machine import RTC
```

```
i = None
ledRgb = None
rtc = None
```

```
ledRgb = LED(LED.RGB)
rtc = RTC()
for i in range(1, 11):
    if i > 5:
        ledRgb.rgb_write(int(1),255,0,0)
    elif i < 3:
        ledRgb.rgb_write(int(3),255,153,0)
    else:
        ledRgb.rgb_write(int(5),51,255,51)
    utime.sleep( 1 )
    ledRgb.off()
```

立即變化



```
1 from machine import LED
2
3 ledRgb = None
4
5
6 ledRgb = LED(LED.RGB)
7 ledRgb.rgb_write(int(1),255,0,0)
8
```

立即修改



PyCode



功能



執行



檔案夾



儲存



清空



- 邏輯
- 迴圈
- 數學
- 文字
- 列表
- 顏色
- 變數



清除
狀態

File
<mo

立即測試



PyCode



功能



執行

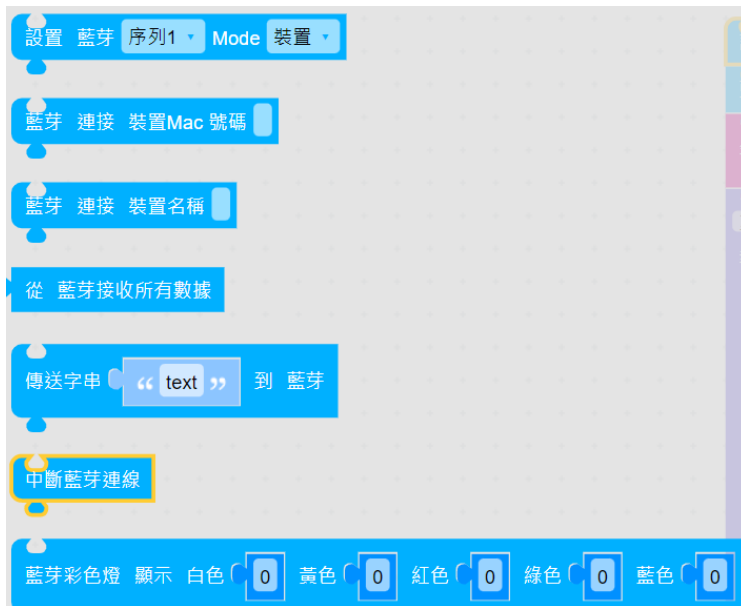


檔案夾

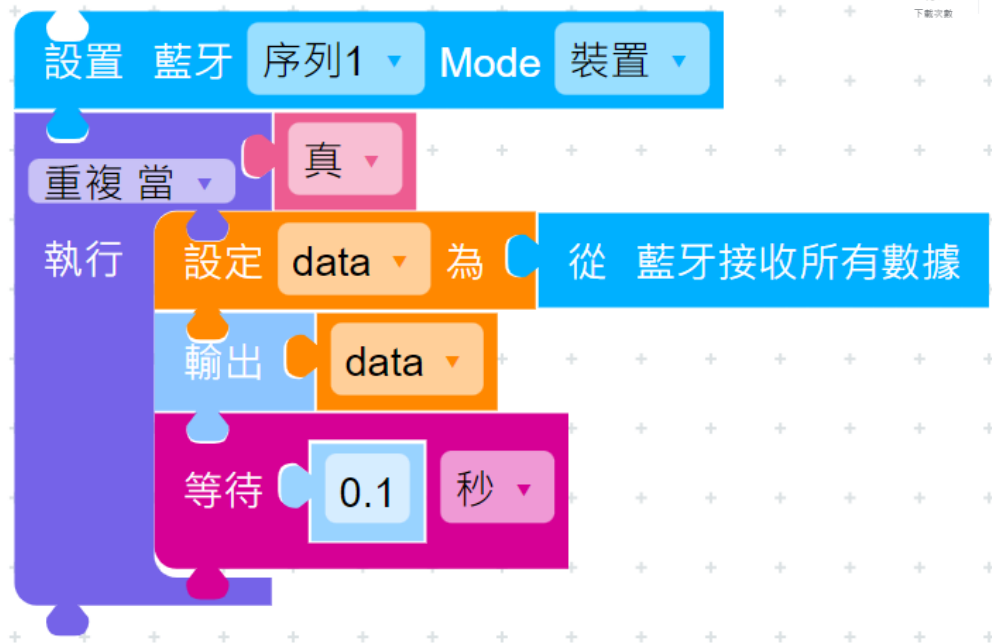
```
1 from machine import LED
2
3 i = None
4 ledRgb = None
5
6
7 ledRgb = LED(LED.RGB)
8 for i in range(1, 30):
9     ledRgb.rgb_write(int(i+1),255,204,204)
10
```

藍芽使用

- 應用下 出現積木



GATT Server (裝置)



遊戲 應用程式 電影 圖書



RL62M BLE Tools

Richlink Technology Co., Ltd.

10+

下載次數



3 歲以上

RL62M BLE APP



BLE NAME

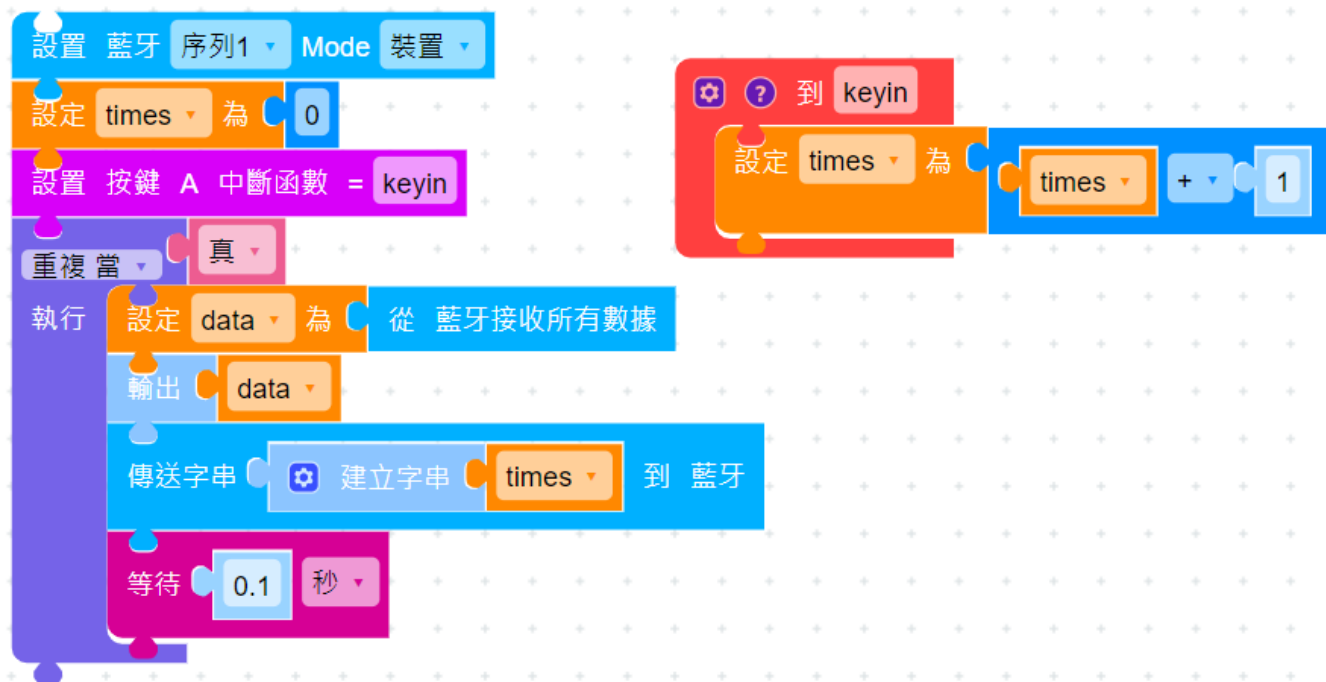
Input Send Data



\n

Send

傳送 接收



兩台 Epy Lite 傳資料

裝置

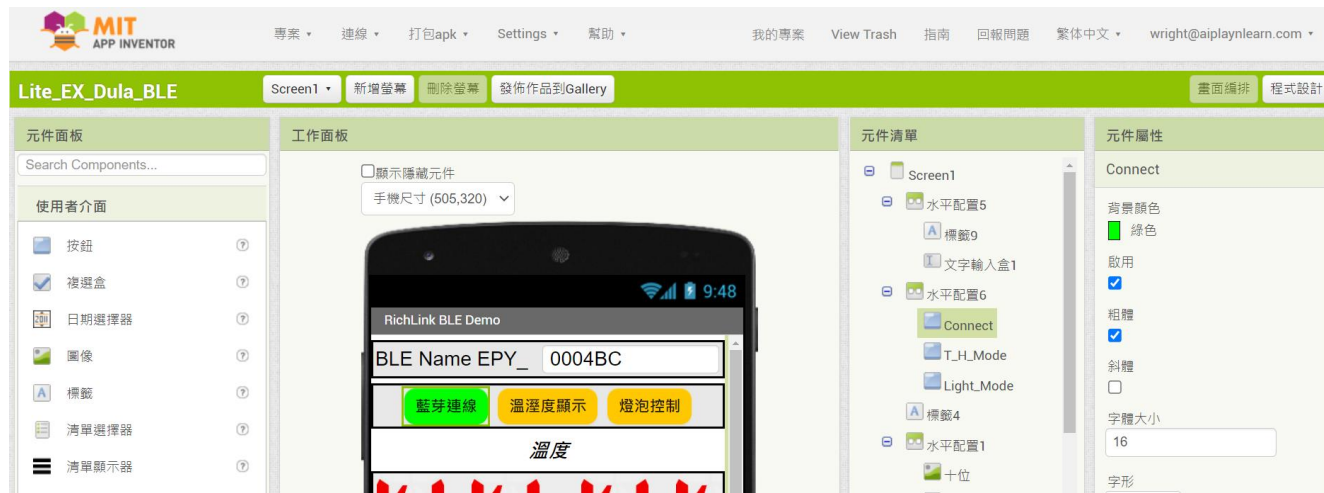


主機



手機 APP Inventor 2

Blockly 圖形介面 易學 手機 平板 上雲

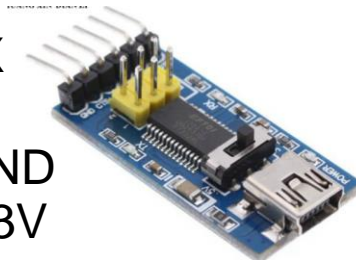


RL62M01A User Guide

Hardware Setup



TX → RX
RX → TX
GND → GND
3.3V → 3.3V



USB

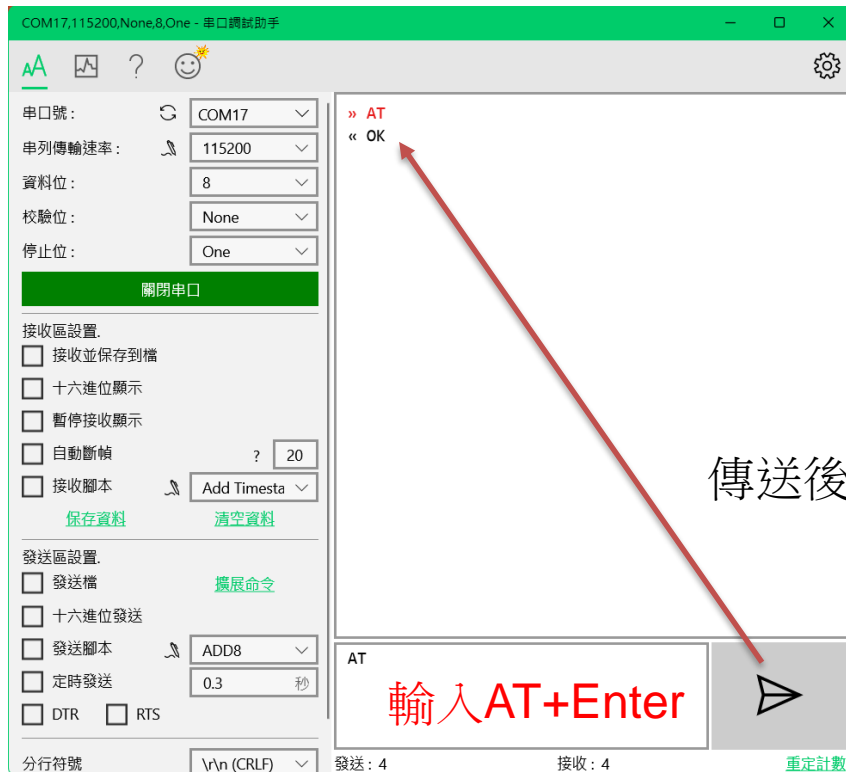


FT232 USB to UART



Win10 Microsoft Store
Serial Debug

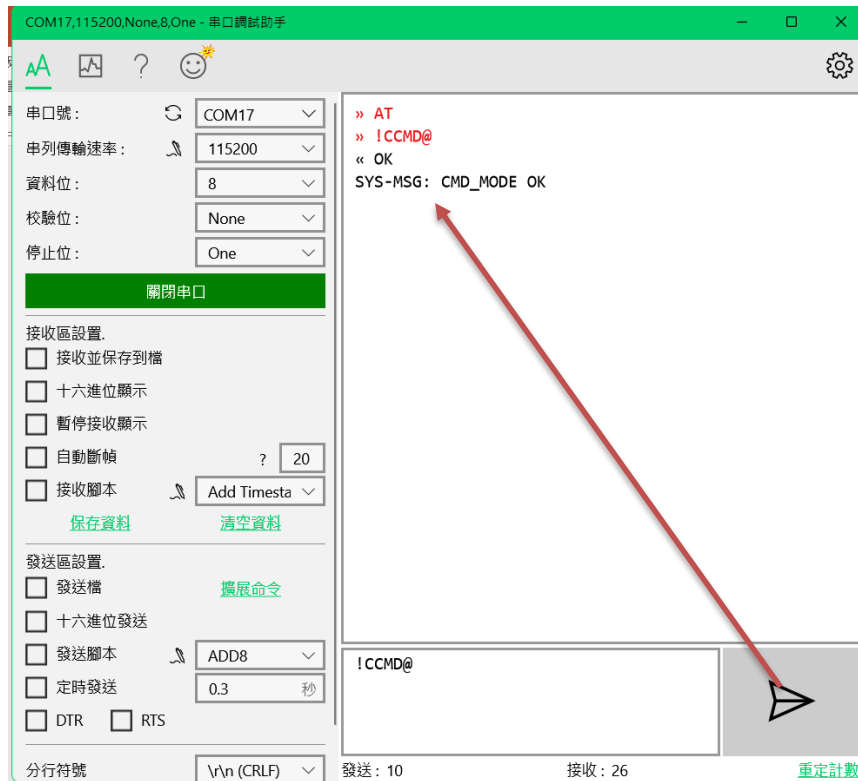
PC Connect RL62M01



傳送後 RL62M 會回 OK

如沒有回 OK, RL62M在 DATA Mode
可以先切換回 Command Mode

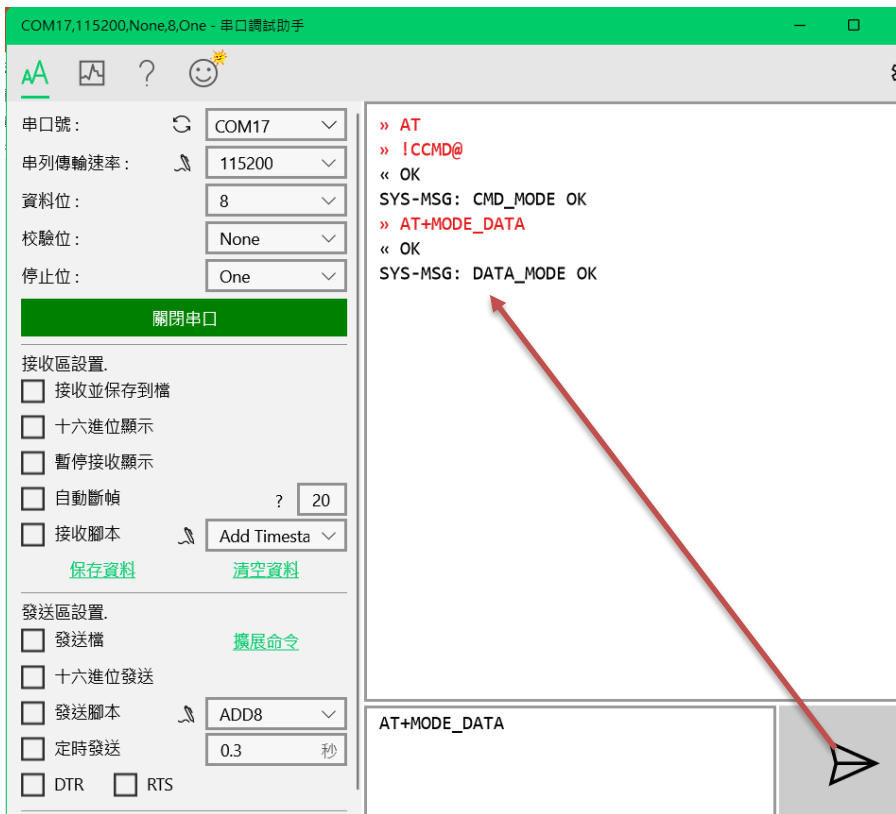
Data Mode to Command Mode



RL62M 會回應系統訊息
SYS-MSG: CMD_MODE OK

!CCMD@
Note: 不需要 Enter

Command Mode to Data Mode



RL62M 會回應系統訊息
SYS-MSG: DATA_MODE OK

AT+MODE_DATA +Enter

Android GATT AT 透傳 測試工具



Google Play

遊戲

應用程式

電影

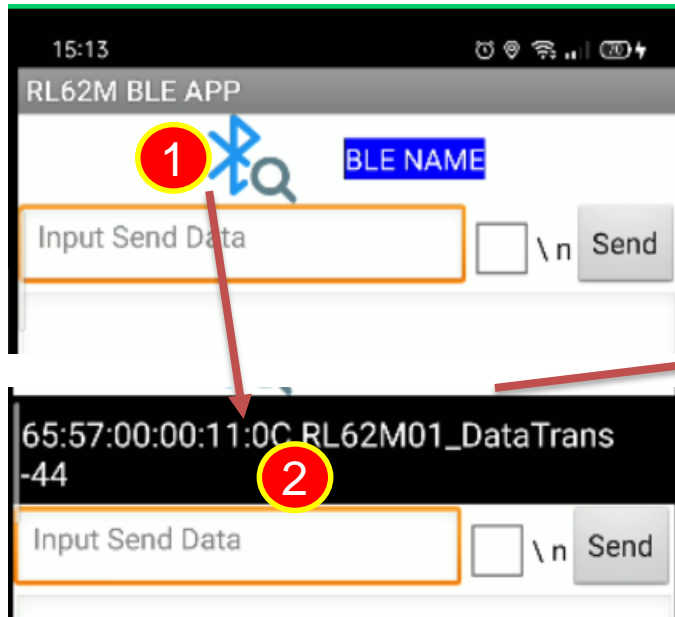
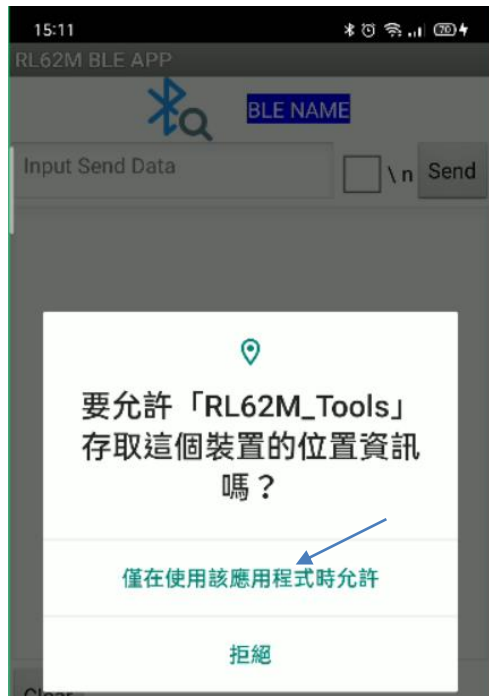


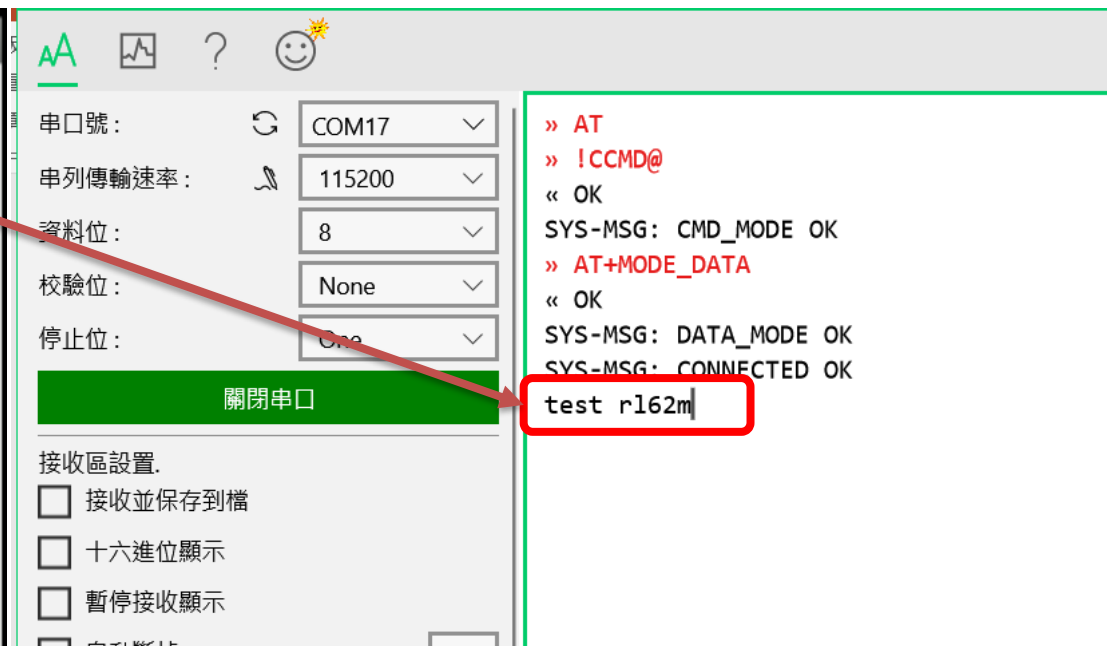
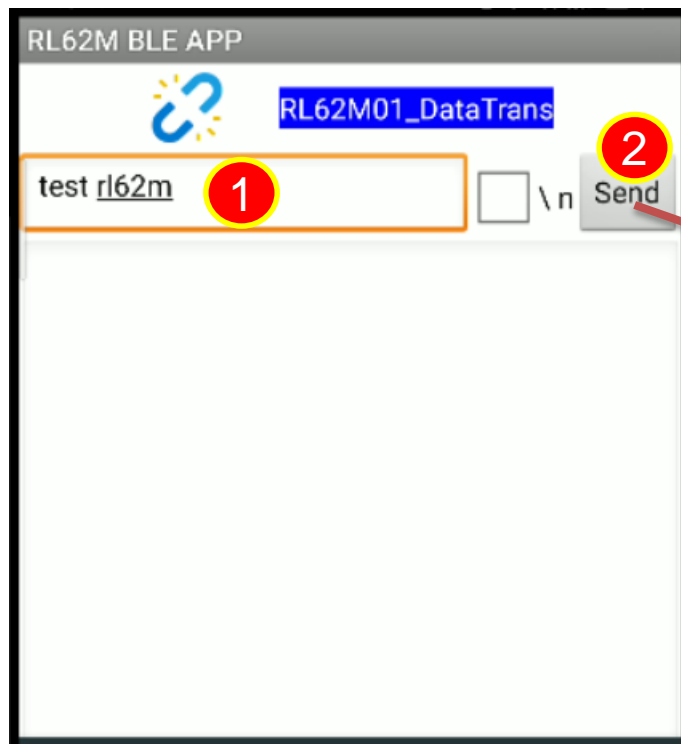
RL62M BLE Tools

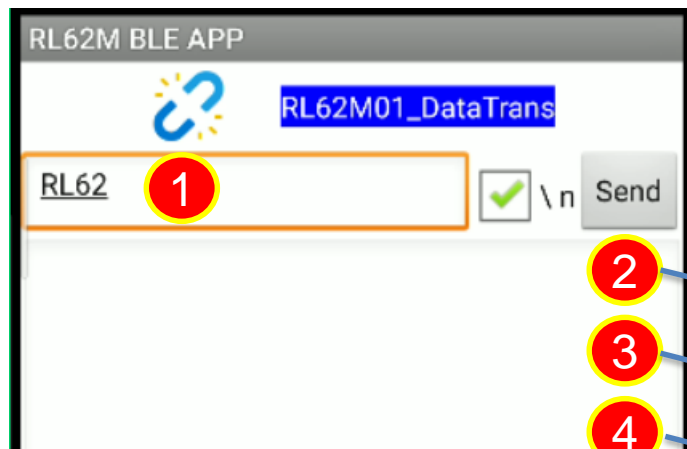
Richlink Technology Co., Ltd.

AT 透傳模式 (接收)

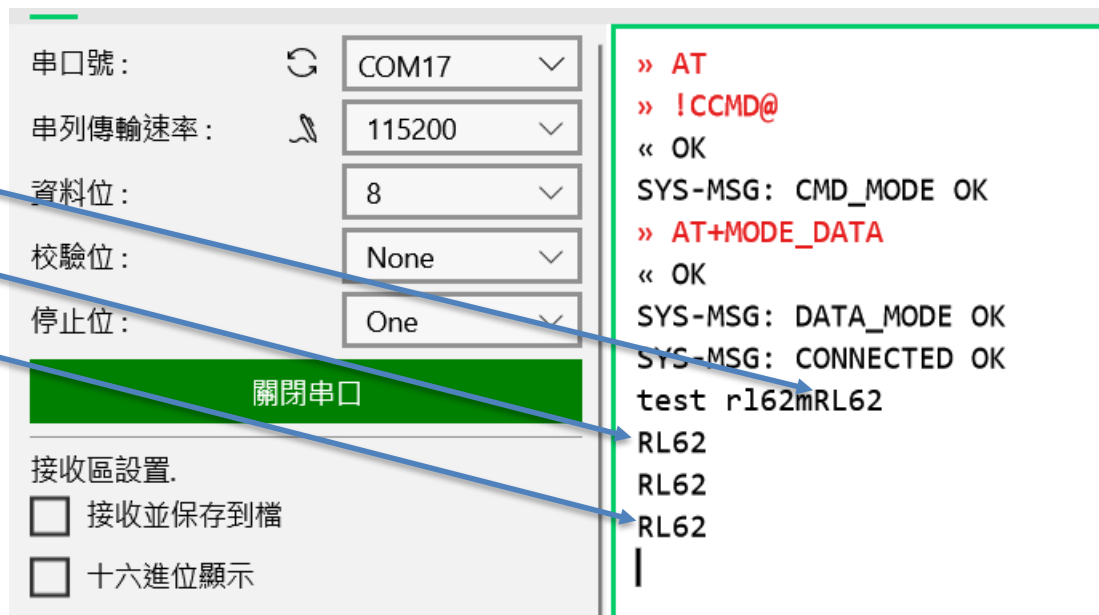
開啟 Android APP “RL62M BLE Tools”







多次傳送
包含換行



RL62M 傳送

確認使用
AT+MODE_DATA
切換到 DATA Mode

The image shows two overlapping windows. The left window is the 'RL62M BLE APP' on a mobile device. It has a status bar at the top showing '15:28' and various icons. Below the title bar, there's a blue icon and a label 'RL62M01_DataTrans'. A text input field contains 'RL62' and is highlighted with an orange border. To its right is a green checkmark icon and a 'Send' button. Below the input field, the text 'Hello !' is visible. At the bottom, there's a 'Clear' button and a status message: 'Mac Address:65:57:00:00:11:0C free ICON by https://icons8.com'.

The right window is a '串口測試助手' (Serial Port Test Assistant) application. It has a title bar 'COM17,115200,None,8,One - 串口測試助手'. The interface includes several dropdown menus for configuration: '串口号:' (COM17), '串列傳輸速率:' (115200), '資料位:' (8), '校驗位:' (None), and '停止位:' (One). There are buttons for '關閉串口' (Close Serial Port), '接收區設置' (Receive Area Settings), and '發送區設置' (Transmit Area Settings). The '接收區設置' section has checkboxes for '接收並保存到檔', '十六進位顯示', '暫停接收顯示', '自動斷幀', and '接收腳本', along with a 'Add Timesta' button. The '發送區設置' section has checkboxes for '發送檔', '十六進位發送', '發送腳本', '定時發送', 'DTR', and 'RTS', along with a 'ADD8' dropdown and a '0.3' timer. A red arrow points from the 'Hello !' text in the mobile app to the 'Hello !' text in the serial terminal's output area. The serial terminal's output area shows a series of AT commands and responses: '» AT', '» !CCMD@', '« OK', 'SYS-MSG: CMD_MODE OK', '» AT+MODE_DATA', '« OK', 'SYS-MSG: DATA_MODE OK', 'SYS-MSG: CONNECTED OK', 'test rl62mRL62', 'RL62', 'RL62', 'RL62', and '» Hello !'. At the bottom right, there's a text input field with 'Hello !' and a '輸入傳送字串' (Enter transmission string) label, and a large right-pointing arrow button.

注意事項

- AT command 最後都需要有 Enter (\r\n)
- RL62M 只有在Data mode 才可送出資料
- 兩個 mode 都可以 接收資料
- CMD/DATA Mode 設定後，下次開機不需要再設定(會記住)

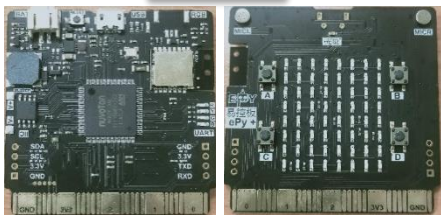
沅鉅科技

智慧藍芽產品

ePy Lite



ePy +



BLE / MESH 彩色球泡燈



BLE MESH 嵌式 插座



多功能 BLE MESH Sensor



BLE MESH 插座



BLE MESH 電燈開關

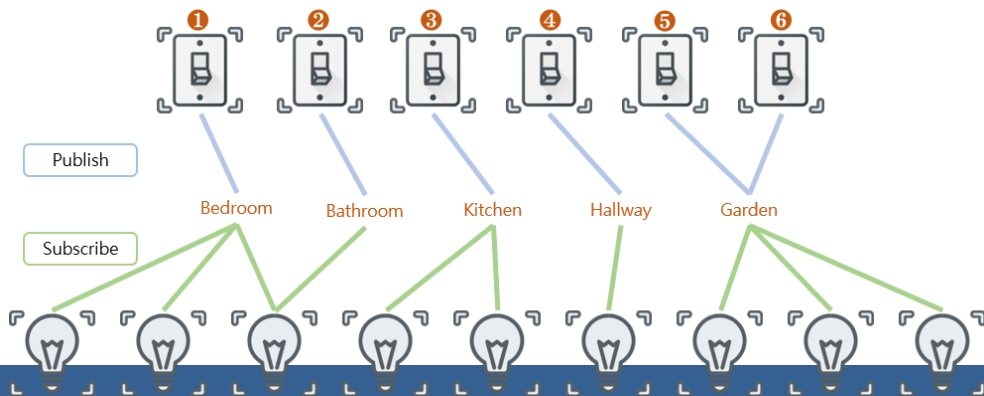
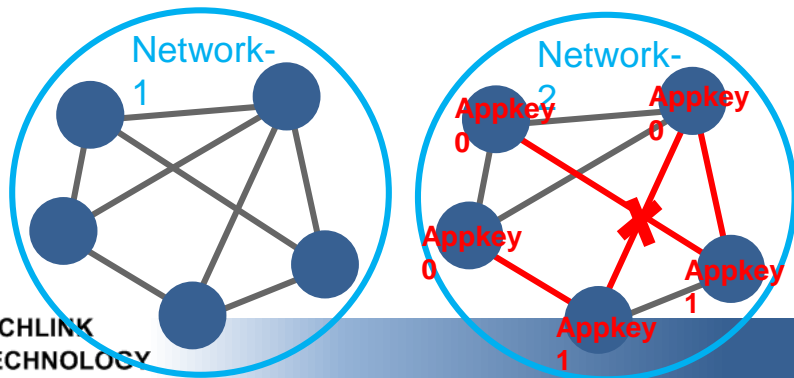


萬用藍芽紅外遙控

介紹

BLE Mesh 網路技術 - (1/2)

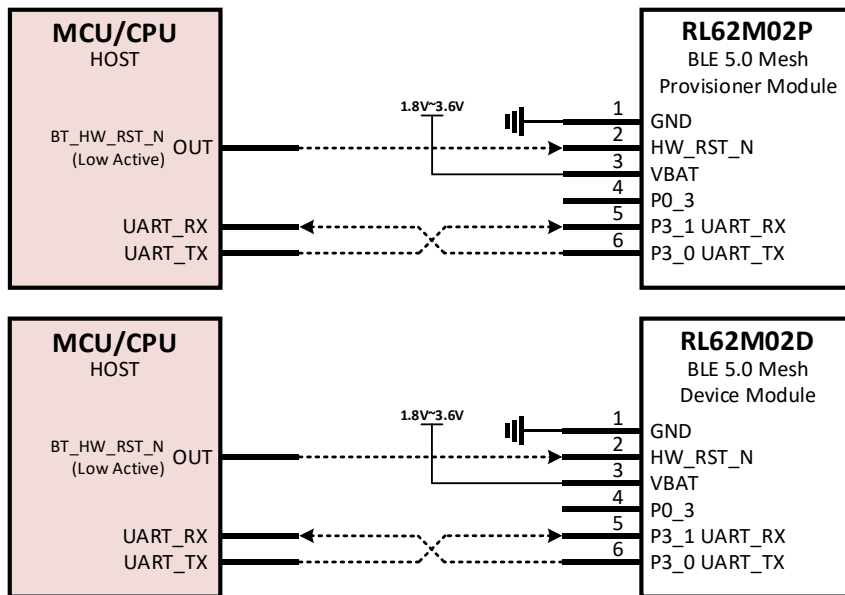
1. 多對多(many-to-many)設備通訊
2. 基於廣播(Broadcast-based)方式通訊
3. 設備中繼功能(Relay)
4. BLE Mesh網路訊息 – 發佈/訂閱(publish/subscribe)
5. 安全性 – NetKey, Appkey



Mesh ATCMD 介紹 (1/4)

- 提供自定義**ATCMD Mesh**指令
- 透過**MCU**控制**Mesh**網路
- 通訊介面： **UART**
- 提供兩種模組**FW**
- 模組角色：

- Provisioner Role: Mesh網路管理者



問卷

- <https://forms.gle/iT4w6ziewr1GHNZv6>



RL62M MESH User Guide

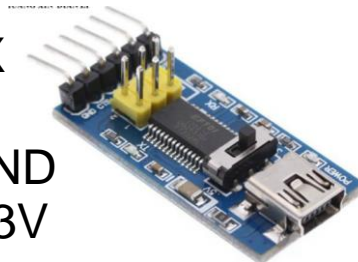
Hardware Setup



mesh device

TX → RX
RX → TX
GND → GND
3.3V → 3.3V

FT232 USB to UART



USB



Win10 Microsoft Store
Serial Debug



mesh provisioner

TX → RX
RX → TX
GND → GND
3.3V → 3.3V

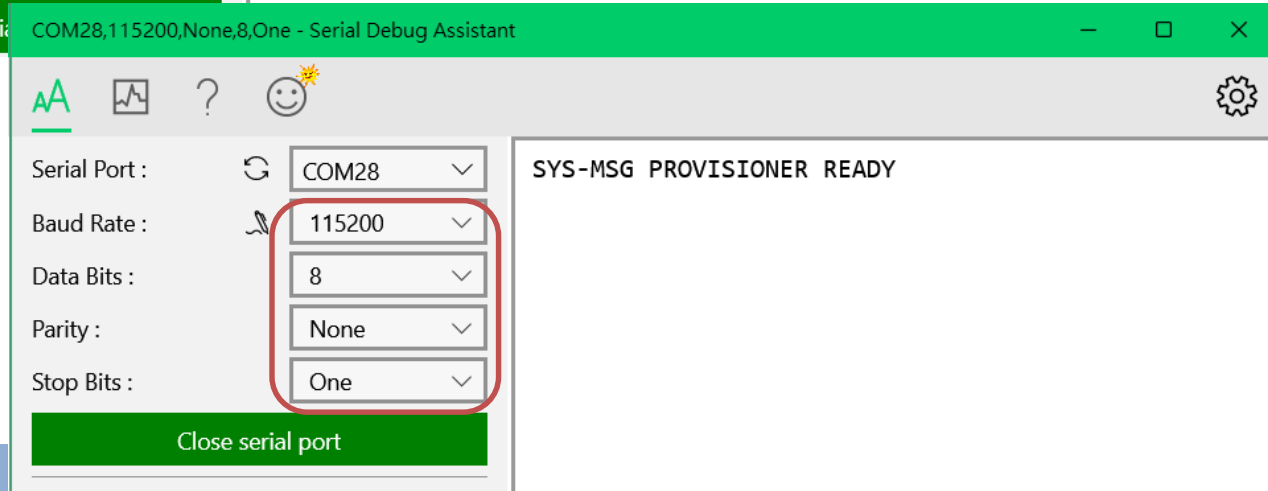
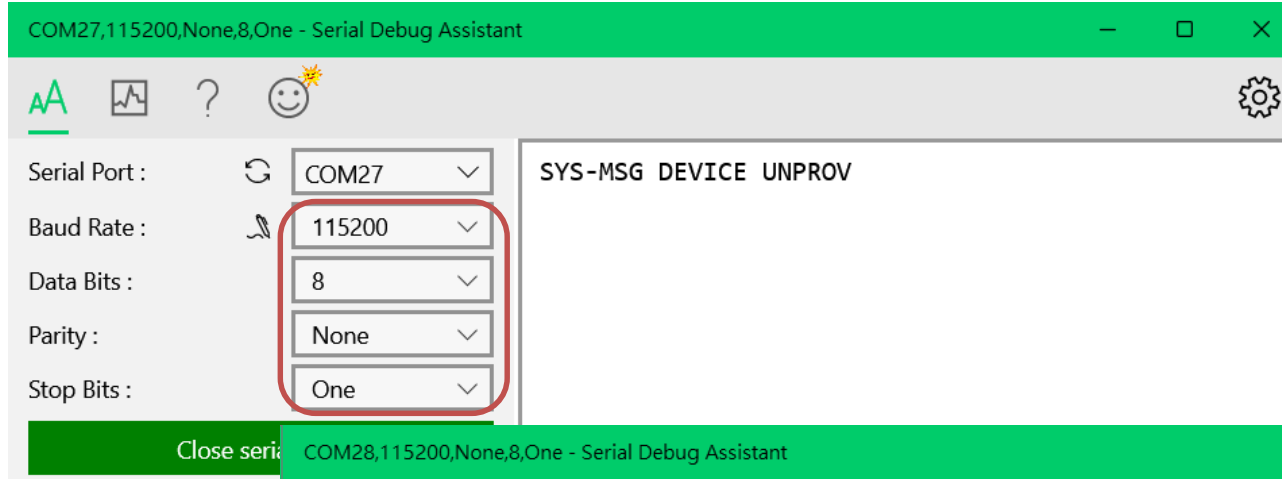
FT232 USB to UART



USB







PC Connect MESH Module





Rest Module

COM27,115200,None,8,One - Serial Debug Assistant



DEVICE

Serial Port :  COM27 ▼


Baud Rate :  115200 ▼

Data Bits : 8 ▼

Parity : None ▼

Stop Bits : One ▼





Close serial port

Receiving settings.
☐ Receive and save to file
☐ HEX display
☐ Pause receiving display
☐ Auto break frame ? 20
☐ Receive scripts  Add Timesta ▼


SYS-MSG DEVICE UNPROV


unprovision
未被綁定

COM28,115200,None,8,One - Serial Debug Assistant



PROVISIONER

Serial Port :  COM28 ▼


Baud Rate :  115200 ▼

Data Bits : 8 ▼

Parity : None ▼

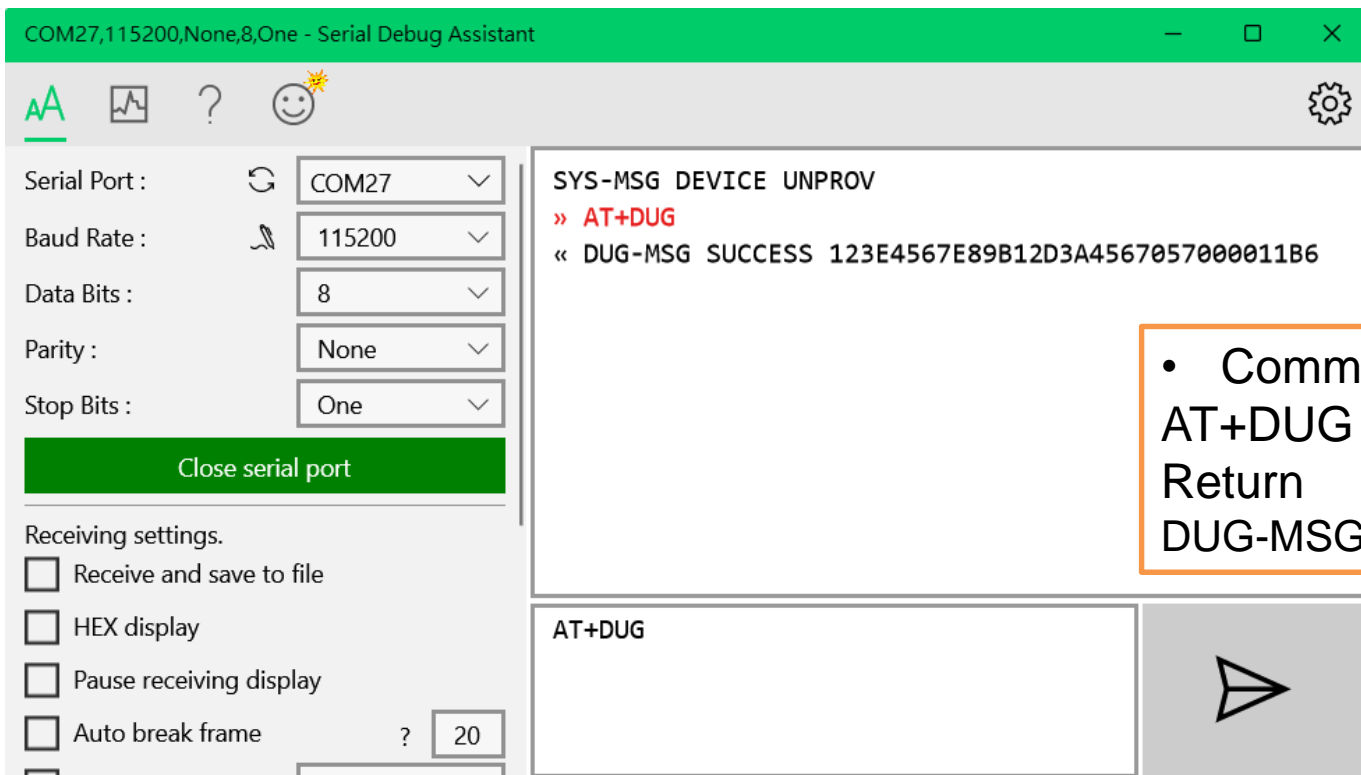
Stop Bits : One ▼

Close serial port

Receiving settings.
☐ Receive and save to file
☐ HEX display
☐ Pause receiving display
☐ Auto break frame ? 20
☐ Receive scripts  Add Timesta ▼

SYS-MSG PROVISIONER READY

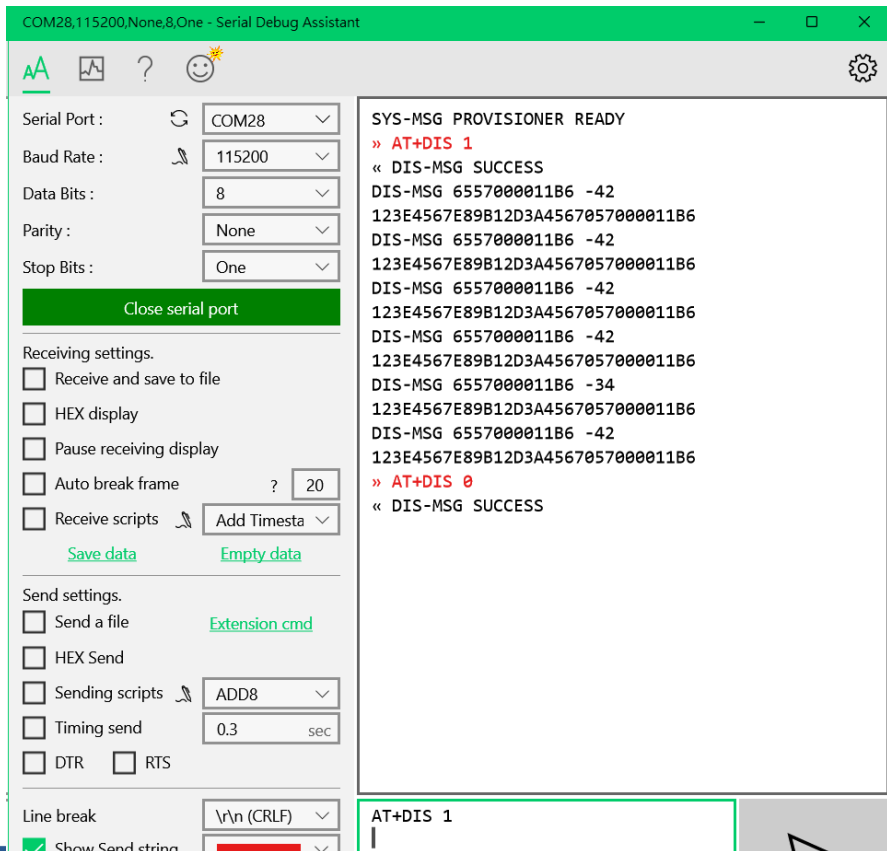
Device Get Self UUID



- Command
AT+DUG `\r\n` → Get Device UUID
Return
DUG-MSG SUCCESS <128bitUUID>

綁定裝置

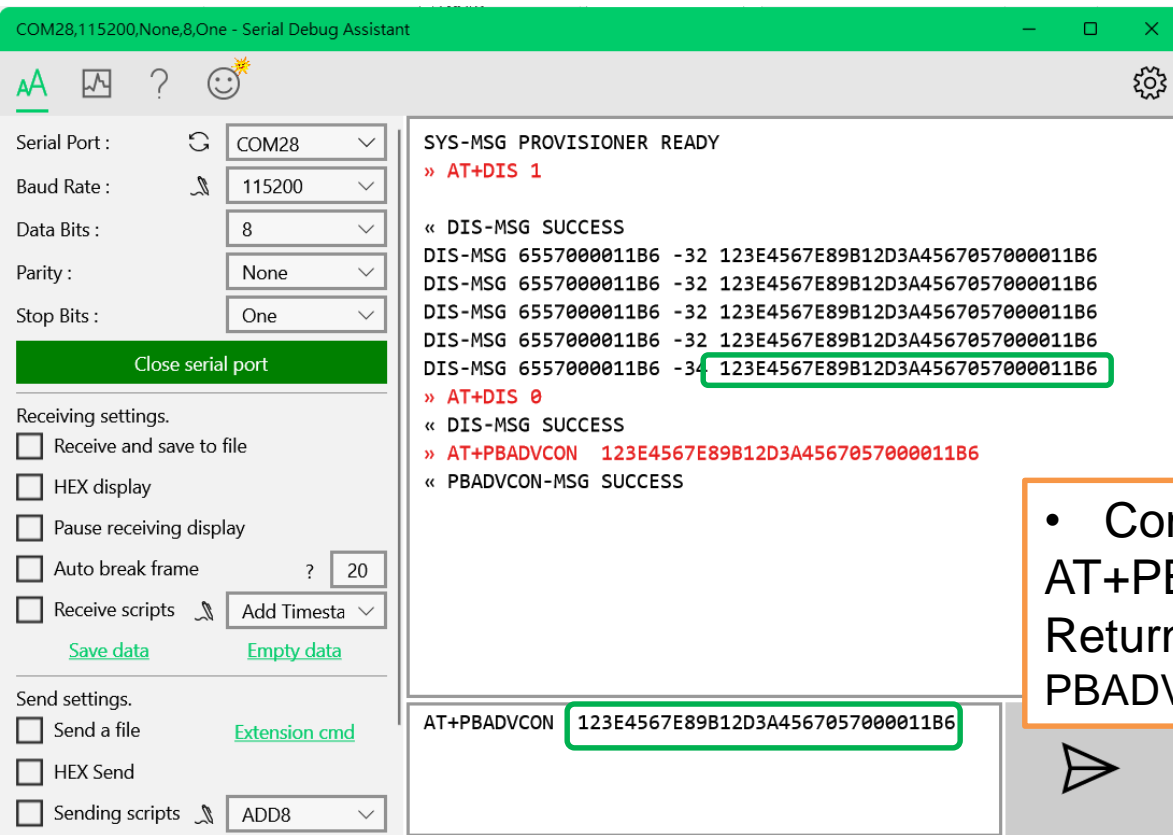
Provisioner Scanning Device



- Command
`AT+DIS 1\r\n` → **Start Scanning**
Return
`DIS-MSG <device MacAddr> <RSSI> <UUID>`

- Command
`AT+DIS 0\r\n` → **Stop Scanning**
Return
`DIS-MSG SUCCESS`

開啟配對廣播通道



- Command
AT+PBADVCON <Device UUID>\r\n
Return
PBADVCON-MSG SUCCESS <128bitUUID>

綁定裝置

DEVICE

```
SYS-MSG DEVICE UNPROV
» AT+DUG
« DUG-MSG SUCCESS 123E4567E89B12D3A4567057000011B6
PROV-MSG SUCCESS 0x0100
```

PROVISIONER

```
SYS-MSG PROVISIONER READY
» AT+DIS 1
« DIS-MSG SUCCESS
DIS-MSG 6557000011B6 -32
123E4567E89B12D3A4567057000011B6
DIS-MSG 6557000011B6 -28
123E4567E89B12D3A4567057000011B6
» AT+DIS 0
« DIS-MSG SUCCESS
» AT+PBADVCON 123E4567E89B12D3A4567057000011B6
« PBADVCON-MSG SUCCESS
» AT+PROV 1
« PROV-MSG SUCCESS 0x0100
```

AT+PROV 1

Send : 8 Receive : 98 Reset count

Send : 77 Receive : 226 Reset cc

- Command
AT+PROV 1\r\n
- Return
PROV-MSG SUCCESS <Device ID>

分配裝置密鑰

COM28

115200

8

None

One

al port

file

lay

?

20

Add Timesta

Empty data

SYS-MSG PROVISIONER READY

» AT+DIS 1

« DIS-MSG SUCCESS

DIS-MSG 6557000011B6 -32

123E4567E89B12D3A4567057000011B6

DIS-MSG 6557000011B6 -28

123E4567E89B12D3A4567057000011B6

» AT+DIS 0

« DIS-MSG SUCCESS

» AT+PBADVCON 123E4567E89B12D3A4567057000011B6

« PBADVCON-MSG SUCCESS

» AT+PROV 1

« PROV-MSG SUCCESS 0x0100

» AT+AKA 0x100 0 0

« AKA-MSG SUCCESS

AT+AKA 0x100 0 0

• Command - 分配app key(0) 到 mesh net (0)
AT+AKA <Device ID> <app key index> <net key index>\r\n
Return
AKA-MSG SUCCESS

綁定裝置節點 Model app Key

COM28,115200,None,8,One - Serial Debug Assistant

Serial Port : COM28
Baud Rate : 115200
Data Bits : 8
Parity : None
Stop Bits : One

Close serial port

Receiving settings.
☐ Receive and save to file
☐ HEX display

```
» AT+MAKB 0x100 0 0x0004005D 0
« MAKB-MSG SUCCESS
```

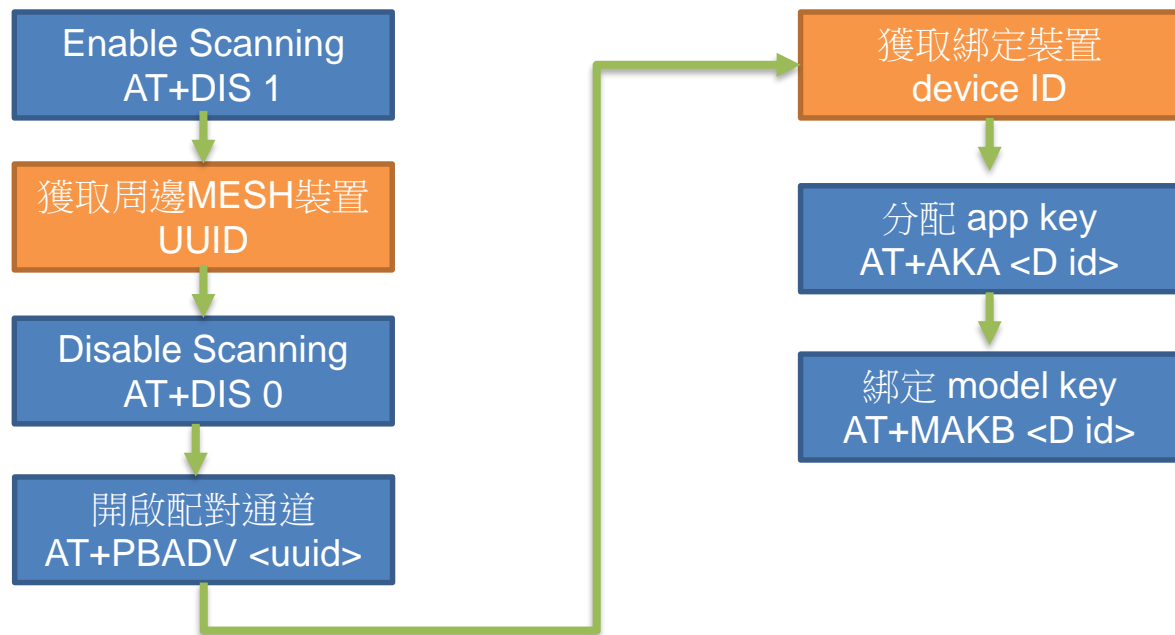
• Command – 綁定app key(0) 到 model (0)
AT+MAKB <Device ID> <app key index> <model>\r\n
Return
AKA-MSG SUCCESS

```
AT+MAKB 0x100 0 0x0004005D 0
```

3.2. Device - Mesh Model list

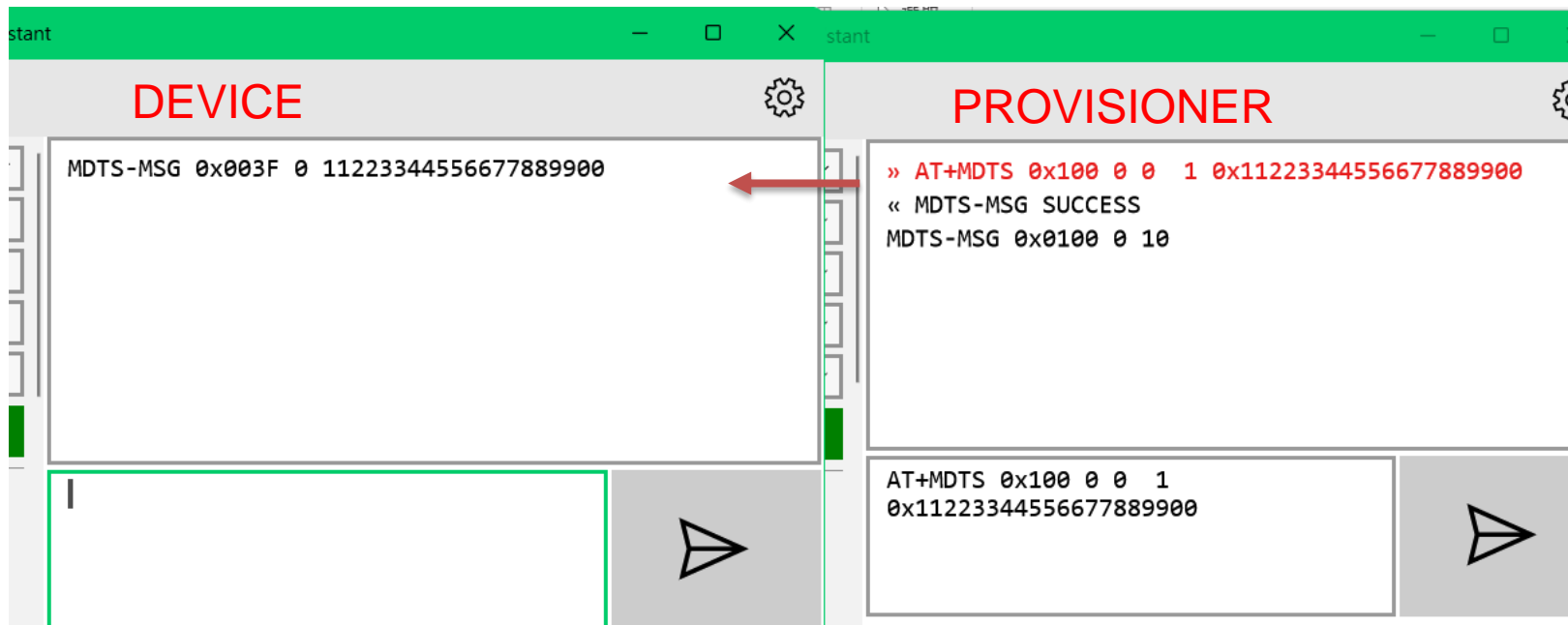
Model Index	Model ID	Model Name	Model Define
0	0x0000005D	Ping Client	Vendor Model
1	0x0004005D	Data Trans Server	Vendor Model
2	0x1000FFFF	Generic On/Off Client	SIG Model

綁定流程



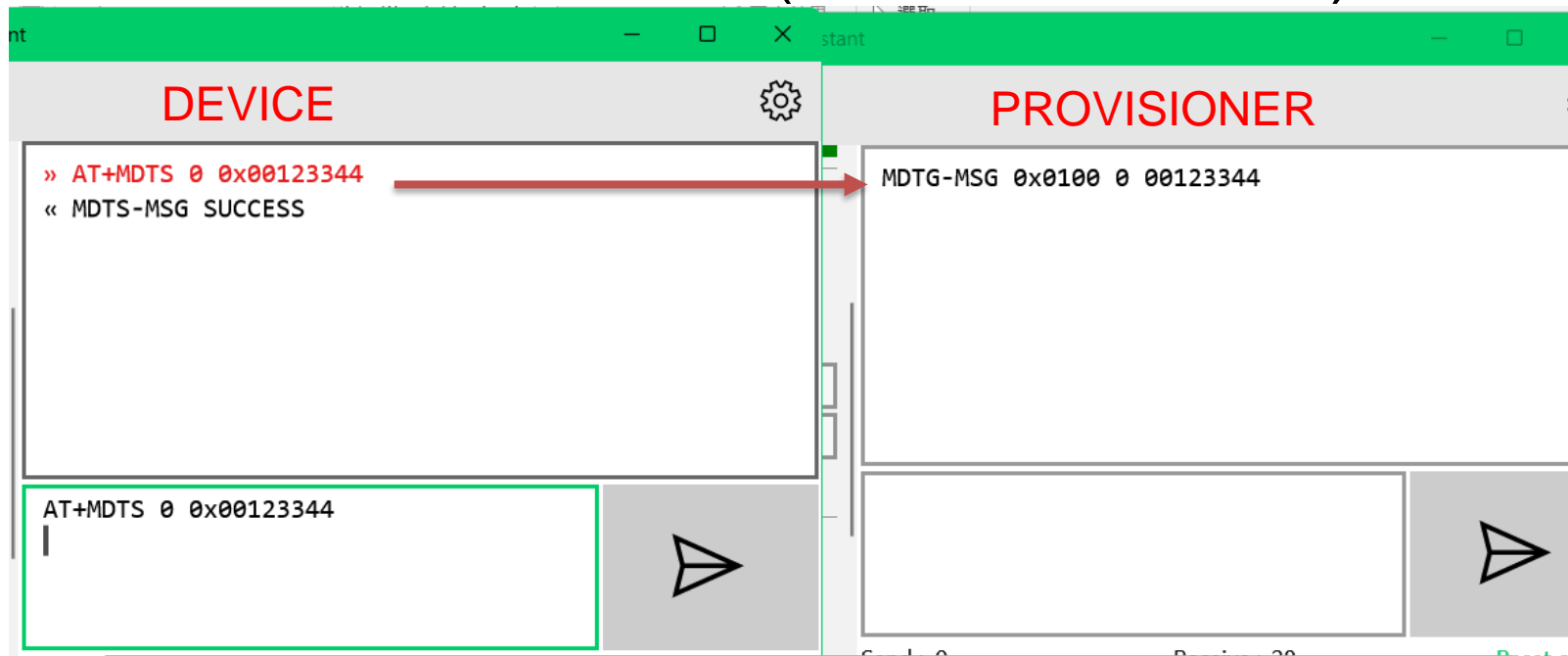
收送資料

PROVISIONER to Device (Datatrans Model)



- Command – DataTrans Model Send
`AT+MDTS [Device ID] [element_index] [app_key_idx] [ack] [data(1~20bytes)]r\n`
Return
`MDTS-MSG <Device ID> <element> <send byte>`

Device to PROVISIONER (Datatrans Model)



- Command – DataTrans Model Send
`AT+MDTS <element_index> <data(1~20bytes)>\r\n`
Return
`MDTS-MSG SUCCESS`

MESH AT Command

Mesh ATCMD 介紹 (2/4)

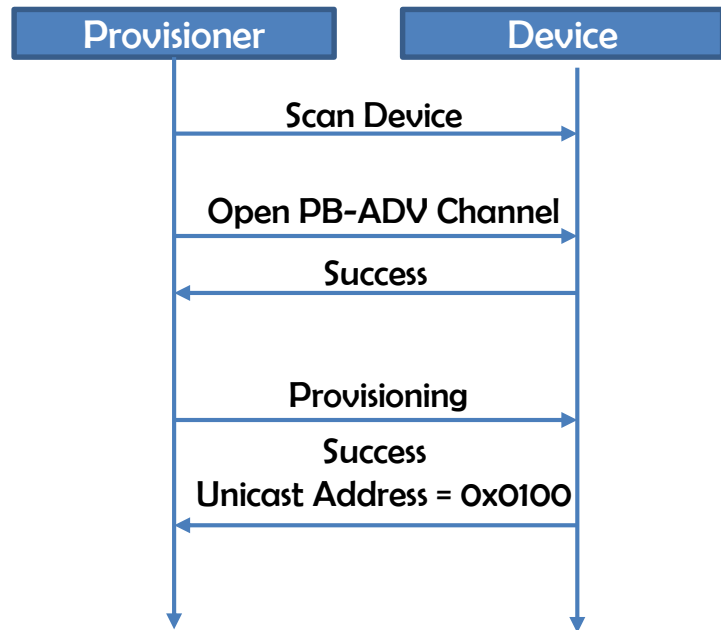
• Provisioner 指令

a. `<< AT+DIS 1\r\n`
`>> DIS-MSG SUCCESS\r\n`
`>> DIS-MSG 655600000152 -48 123E4567E89B12D3A456655600000152\r\n`
`>> DIS-MSG 655600000152 -48 123E4567E89B12D3A456655600000153\r\n`
`>> DIS-MSG 655600000152 -48 123E4567E89B12D3A456655600000151\r\n`
`<< AT+DIS 0\r\n`
`>> DIS-MSG SUCCESS \r\n`

b. 開啟Mesh PB-ADV通道(AT+PBADVCON [DEV UUID])

`<< AT+PBADVCON 123E4567E89B12D3A456655600000151\r\n`
`>> PBADVCON-MSG SUCCESS\r\n`

c. `<< AT+PBADVCON 123E4567E89B12D3A456655600000152\r\n`
`>> PBADVCON-MSG SUCCESS\r\n`
`<< AT+PROV\r\n`
`>> PROV-MSG SUCCESS 0x0100\r\n`



Mesh ATCMD 介紹 (3/4)

• Provisioner 指令

a. 設置節點AppKey並綁至NetKey上(AT+AKA [dst] [app_key_index])

```
<< AT+AKA 0x100 0 0\r\n<<  
>> AKA-MSG SUCCESS\r\n<>
```

b. 綁定Model的Appkey (AT+MAKB [dst] [element_index] [model_id])

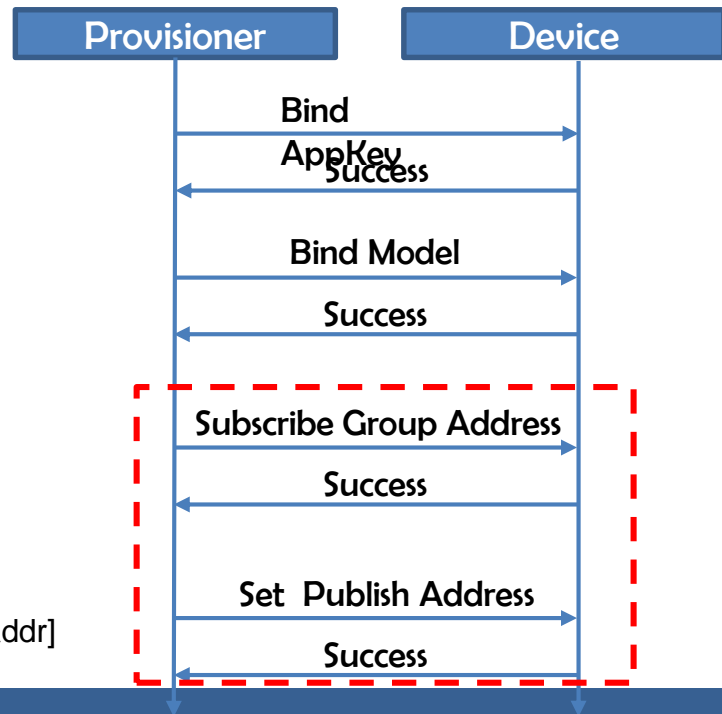
```
<< AT+MAKB 0x100 0 0x1000ffff 0\r\n<<  
>> MAKB-MSG SUCCESS\r\n<>
```

c. 新增節點訂閱群組位址 (AT+MSAA [dst] [element_index] [model_id])

```
<< AT+MSAA 0x100 0 0x1000ffff 0xc000\r\n<<  
>> MSAA-MSG SUCCESS\r\n<>
```

d. 新增節點Publish位址 (AT+MPAS [dst] [element_idx] [model_id] [publish_addr])

```
<< AT+MPAS 0x100 0 0x1000ffff 0x101 0\r\n<<  
>> MPAS-MSG SUCCESS\r\n<>
```



Mesh ATCMD 介紹 (4/4)

• Provisioner 指令

SIG Model: Generic ON/OFF Model Server ID = 0x1000ffff

Vendor Model: Datatrans Model Server ID = 0x0004005d

a. ~~設置SIG Model – Generic on/off model 狀態~~ (AT+GOOS [dst] [on/off] [ack] [app_key_idx] [steps])

```
[res]<< AT+GOOS 0x100 1 1 0 0 0 0\r\n<<  
>> GOOS-MSG SUCCESS\r\n<<  
>> GOOG-MSG 0x0100 0 1\r\n<<
```

b. << AT+GOOG 0x100 0 0\r\n<< model 狀態 (AT+GOOG [dst] [element_index] [app_key_idx])

```
>> GOOG-MSG SUCCESS\r\n<<  
>> GOOG-MSG 0x0100 0 1\r\n<<
```

c. ~~設置Vendor Model – Datatrans model 狀態~~ (AT+MDTS [dst] [element_index] [app_key_idx] [ack])

```
[da]<< AT+MDTS 0x100 0 0 1 0x1122335566778899\r\n<<  
>> MDTS-MSG SUCCESS\r\n<<  
>> MDTS-MSG 0x0100 0 8\r\n<<
```

d. ~~讀取Vendor Model – Datatrans model 狀態~~ (AT+MDTG [dst] [element_index] [app_key_idx])

```
[res]<< AT+MDTG 0x100 0 0 3\r\n<<  
>> MDTG-MSG SUCCESS\r\n<<  
>> MDTG-MSG 0x0100 0 112233\r\n<<
```

群組(Group)

- 單一節點(Node)可訂閱單個或多個Group address (0xC000 ~ 0xffff)

Example:

- AT+{cmd} 0xC001 {data}
- AT+{cmd} 0xC003 {data}

Group-1 (Address:
0xC001)

Group-3 (Address:
0xC003)

Group-2 (Address:
0xC002)

