

# Python藍牙連線教學

2019 spring, 電資工程入門設計與實作 final project

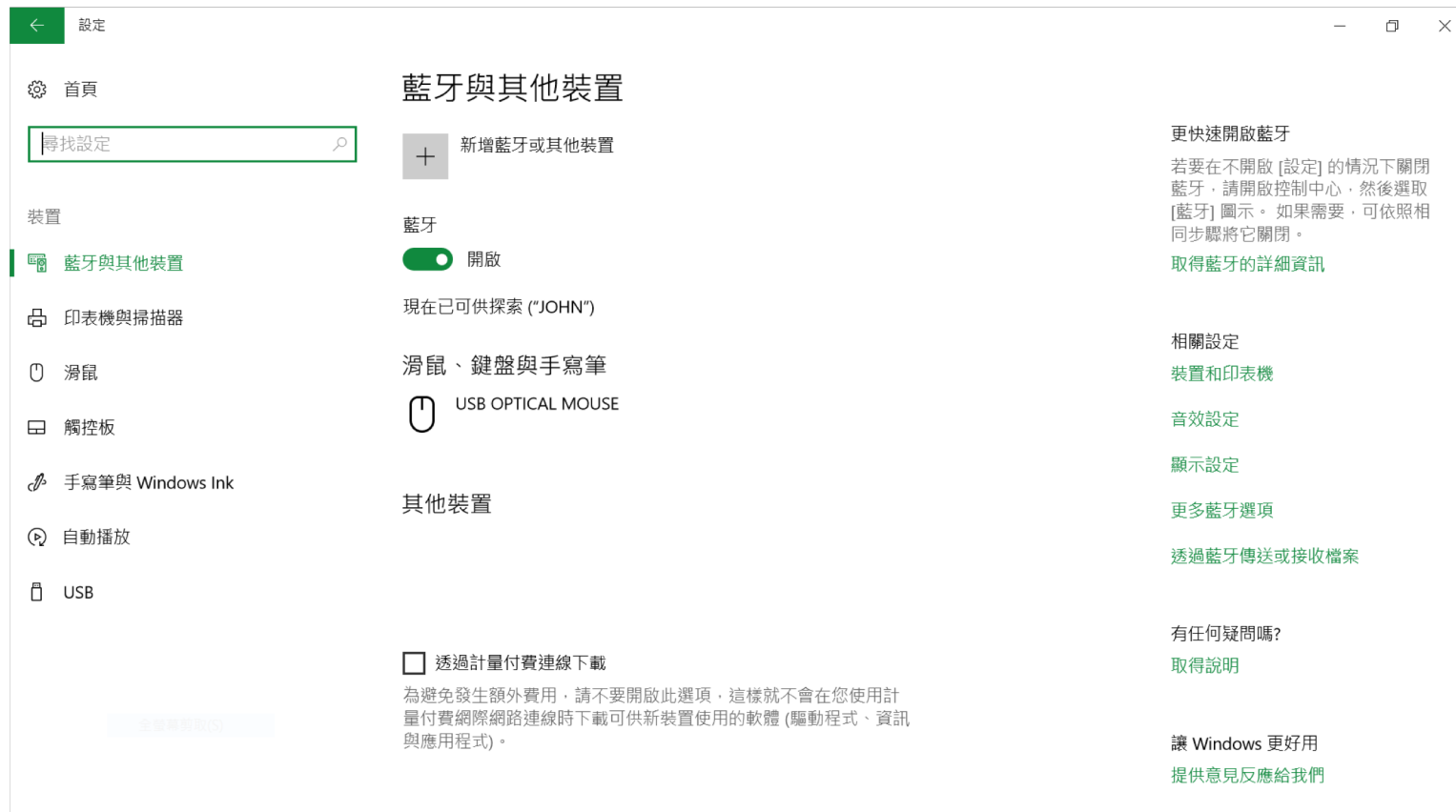
授課教師：李建模教授

課程助教：趙冠豪、陳界宇、周武堂

教案新增 mac 電腦藍芽連線：趙冠豪

# 設定電腦藍牙連線 --- 以WIN10為例

► 進入設定 -> 裝置 -> 藍牙與其他裝置



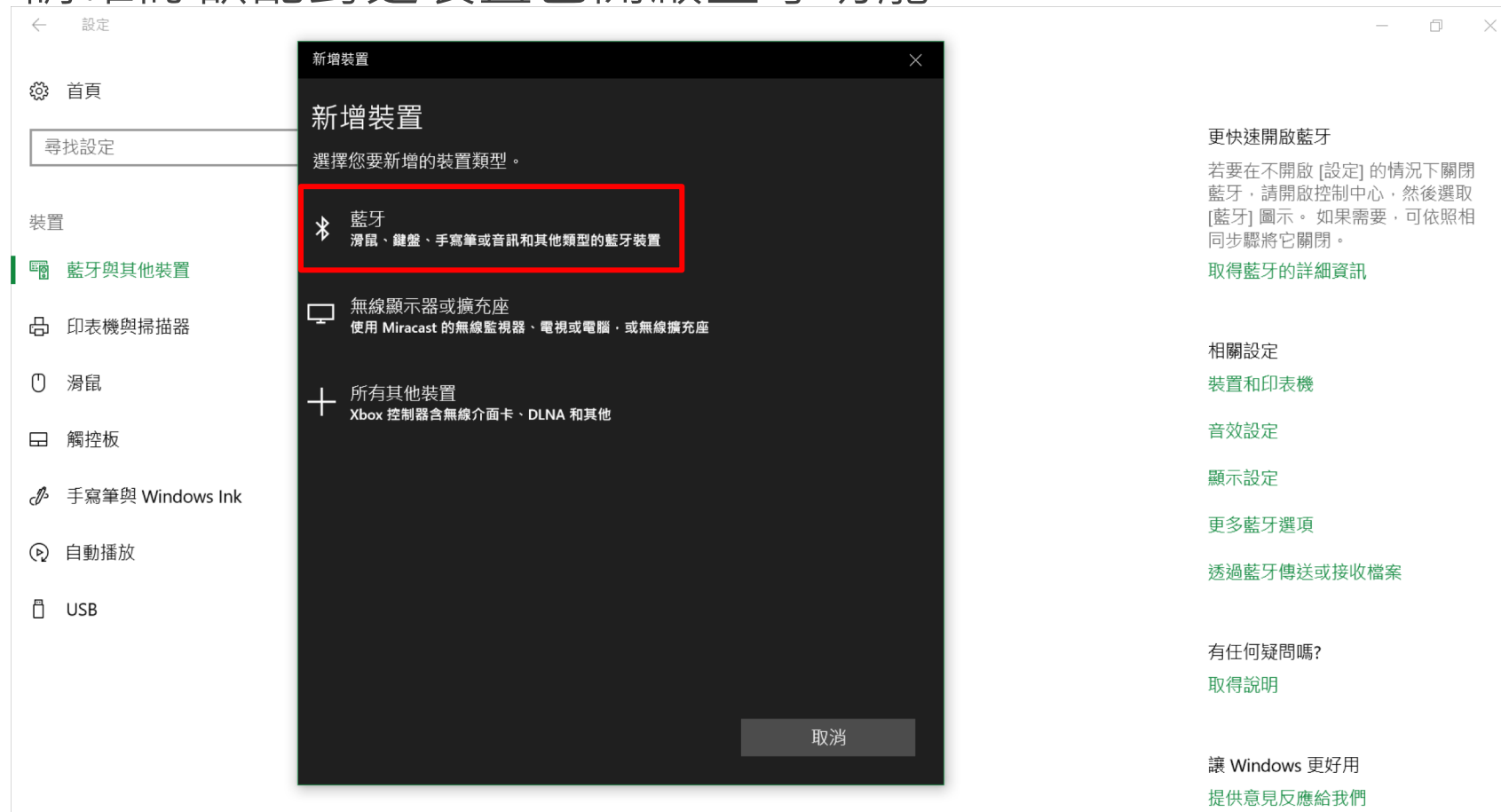
# 新增裝置並配對

## ► 開啟藍牙選項並點選 新增藍牙或其他裝置



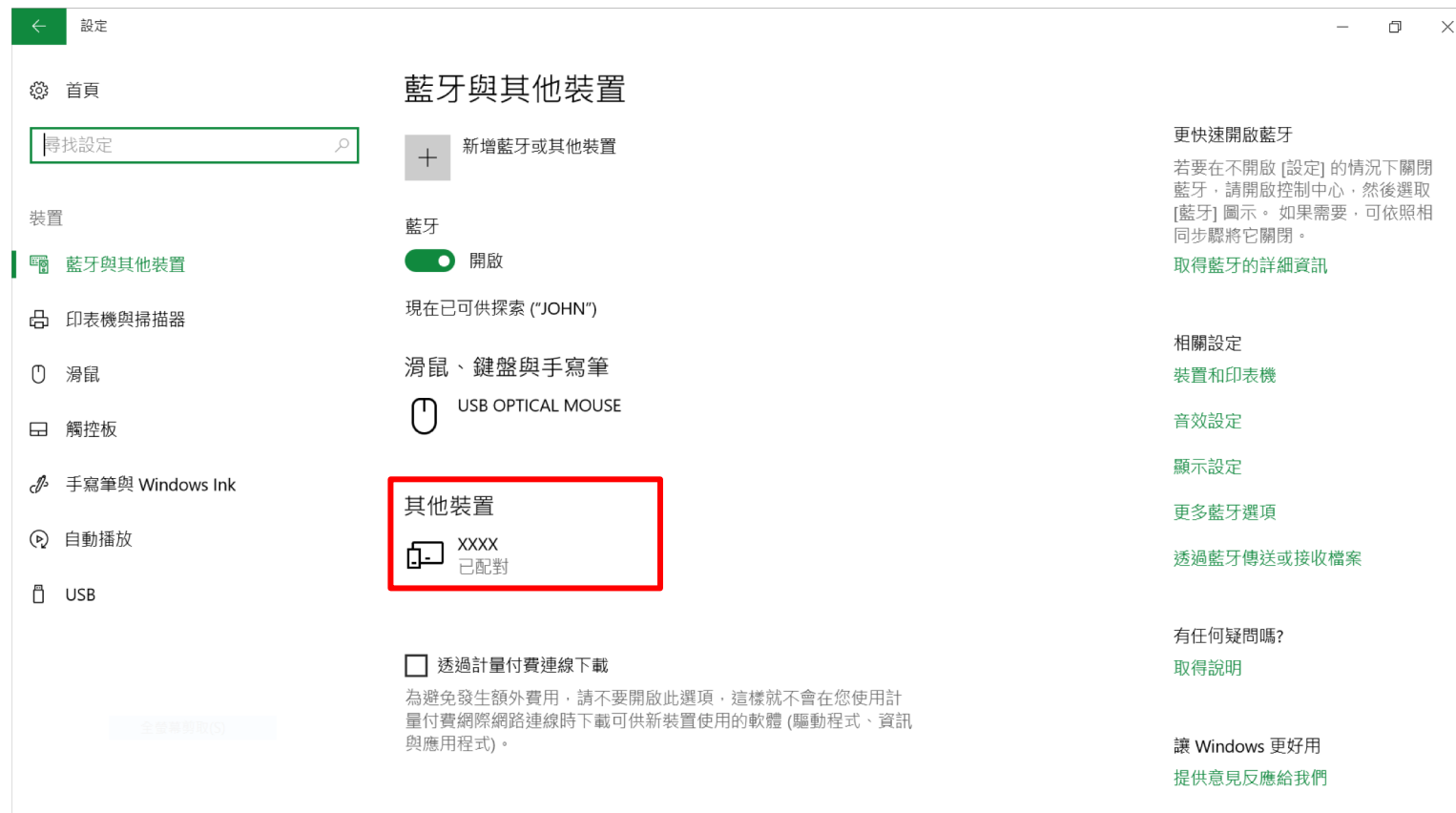
# 新增裝置並配對

- ▶ 選擇 藍牙，依照指示與裝置(藍牙模組)進行配對。
- ▶ 請確認欲配對之裝置已開啟藍牙功能



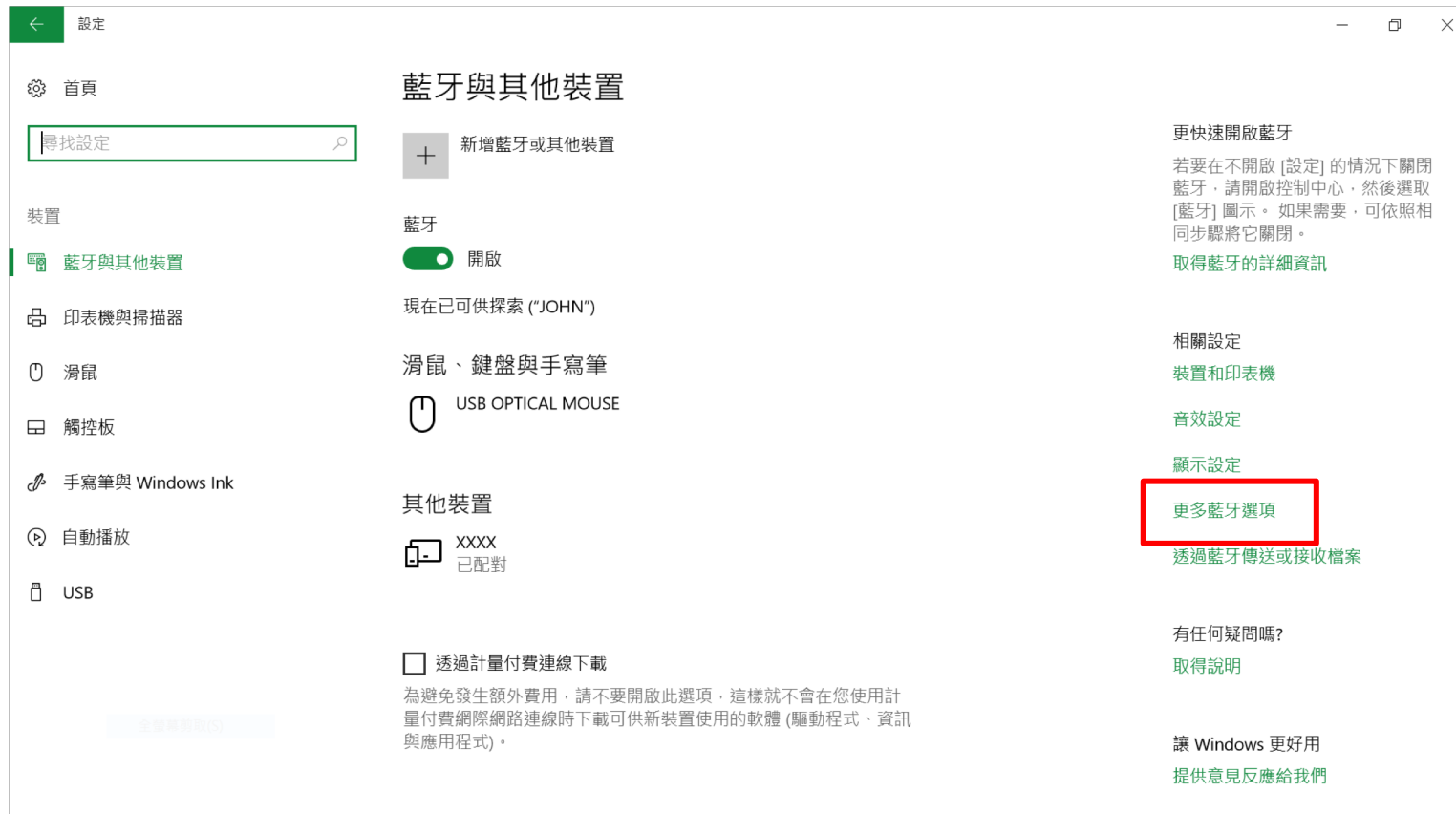
# 新增裝置並配對

## ▶ 配對完成



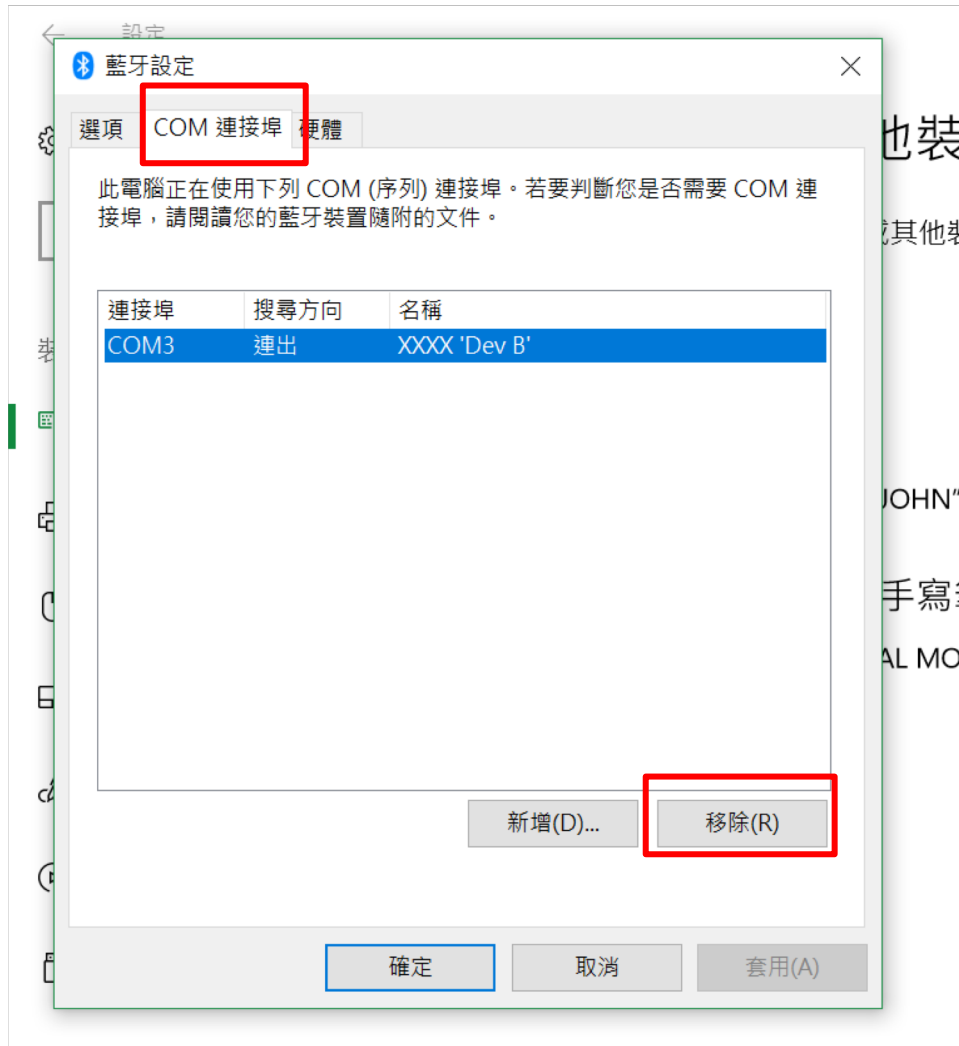
# 設定連接埠

## ► 點擊 更多藍牙選項



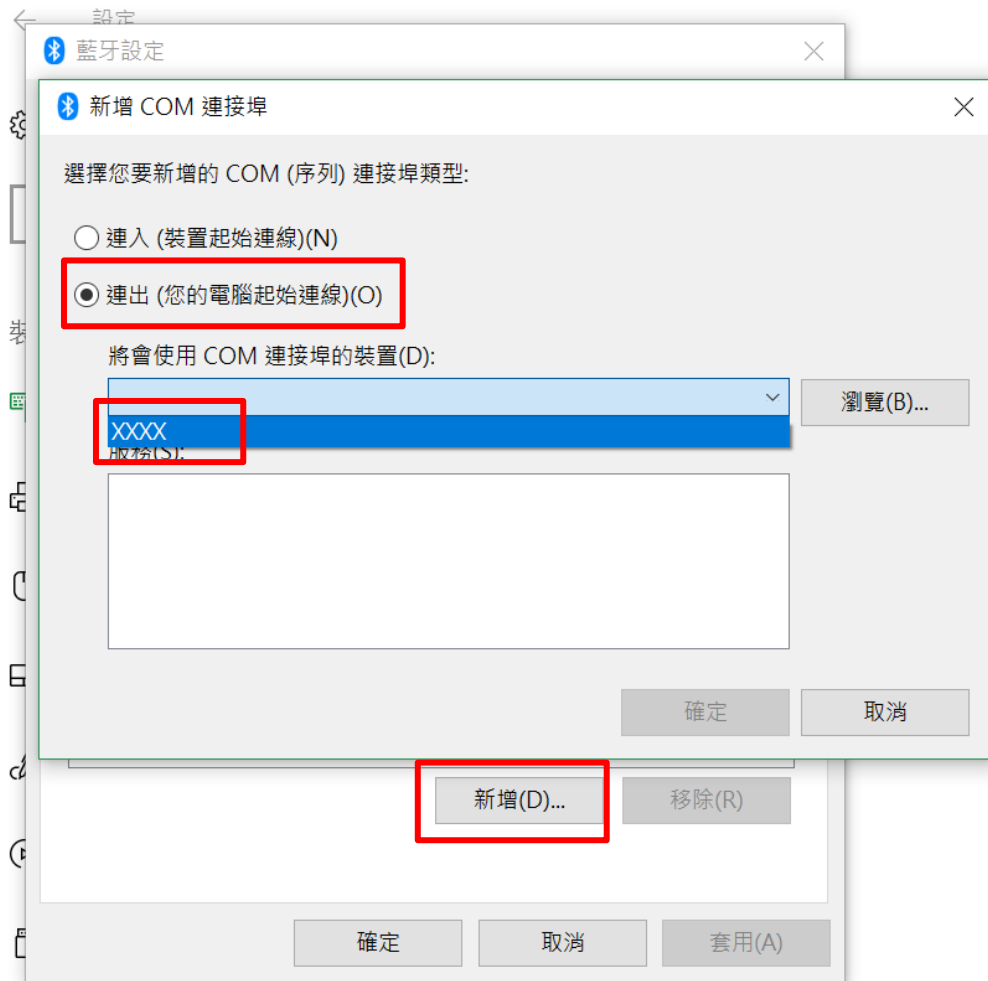
# 設定連接埠

## ► 進入 COM連接埠 並移除現有連接埠



# 設定連接埠

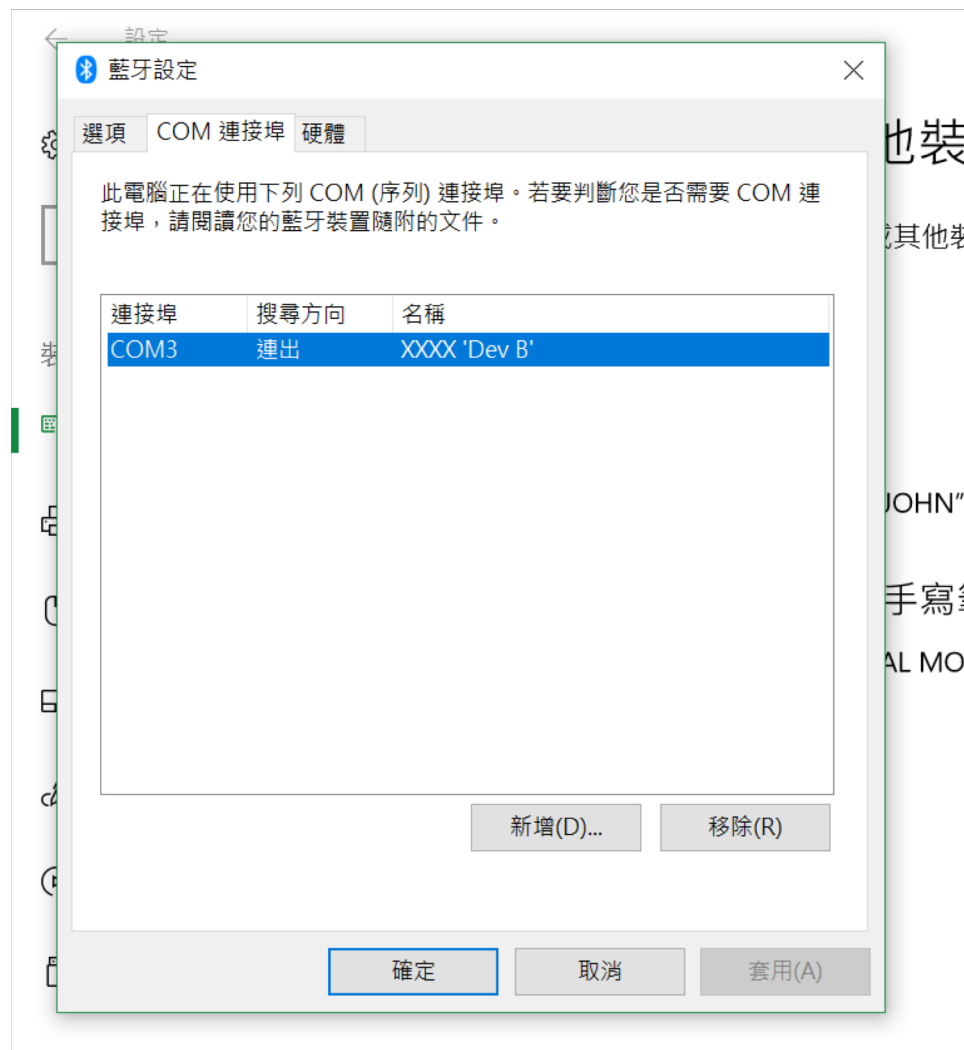
► 點選 新增 -> 連出 -> (裝置名)





# 設定連接埠

## ► 完成設定



# 設定電腦藍牙連線

## --- 以 macOS 為例

- ▶ 首先先用 mac 連接目標藍芽 (留下暫存檔)
- ▶ 預設密碼 : 1234



# 設定電腦藍牙連線

## --- 以 macOS 為例

- ▶ 進入 terminal ，輸入 command > “ls /dev/tty.\*”

```
[chaokuan-hao@Chaode-MacBook-Pro-6] - [~] - [⌵]
```

[日 12 16 - 12:23]

```
• ls /dev/tty.*
```

```
/dev/tty.Blueetooth-Incoming-Port /dev/tty.HC-05-SPPDev
```

- ▶ 跑Python code 時，輸入找到的 /dev/tty.HC-05-SPPDev
- ▶ 確認藍芽連線

# 確認藍芽模組與電腦連線

- ▶ 原本閃爍紅燈速度快，連線之後閃爍速度會變慢（約兩秒刪一次）
- ▶ 確認可以順利的利用電腦傳送指令，切換車子再 Remote mode 以及 Tracking mode.
- ▶ \*\*（報告補充）：可以幫他們 debug，先燒入我們寫好的程式，他們改完腳位，確認可以成功運行，再幫他們洗掉程式。

# Python 藍牙連線函式庫

## --- pySerial

► Documentation:

<https://pythonhosted.org/pyserial/>



# pySerial 安裝方式

- ▶ For Windows:
  - ▶ 終端機輸入 `pip install serial`
- ▶ For MacOS or Linux
  - ▶ 終端機輸入 `sudo pip install pyserial`
- ▶ 請確認系統已先安裝pip
- ▶ 若系統有區分python2與python3，請將pip改成pip3
- ▶ 或下載package並移至python路徑
- ▶ <https://pypi.python.org/pypi/pyserial>

# pySerial API

- ▶ 建立Serial物件 (建立連線)
- ▶ `ser = serial.Serial(port,baudrate,timeout, ...)`
  - ▶ `port: string`，連接埠名稱(ex: COM4、/dev/ttyUSB0)，預設為None
  - ▶ `baudrate: int`，鮑率，務必與所有裝置一致，預設為9600
  - ▶ `timeout: float`，讀取時間長度，預設為None

# pySerial API

- ▶ `read(size)`
  - ▶ 讀取input buffer 中的前size個byte，size預設為1。
  - ▶ Return type: bytes
- ▶ `readline()`
  - ▶ 讀取input buffer直到'\n'
  - ▶ 必須與timeout配合使用
  - ▶ Return type: bytes
- ▶ `write(data)`
  - ▶ 將data寫入連接埠
  - ▶ data的型別為bytes(或bytearray)



# pySerial API

- ▶ `in_waiting` 、 `out_waiting`
  - ▶ 回傳input/output buffer 中的 byte 數量
- ▶ `reset_input_buffer()` 、 `reset_output_buffer()`
  - ▶ 清除input/output buffer中的資料
- ▶ `is_open` 、 `is_close`
  - ▶ 檢查連線是否開啟或關閉

# pySerial API

► Sample code:

```
import serial
ser = serial.Serial("COM3",9600,timeout=2)
waiting = ser.in_waiting
if(waiting != 0):
    rv = [chr(c) for c in ser.read(waiting)]
    print("".join(rv))
    ser.reset_input_buffer()
```

# pySerial API

► Sample code:

```
import serial  
ser = serial.Serial("COM3",9600,timeout=2)  
rv = ser.readline().decode("utf-8")  
print(rv)  
ser.reset_input_buffer()
```

# pySerial API

► Sample code:

```
import serial  
ser = serial.Serial("COM3",9600,timeout=2)  
output = "A"  
send = "A".encode("utf-8")  
ser.write(send)
```