

Arduino 開發板在資料擷取之應用

專案說明與軟硬體準備



Project-Based Learning - 積木程式與智慧科技

叮噹和鄰近家鄉的好朋友A夢、鈴鈴和大雄在某大學工程學系就讀，看到媒體上經常報導未來智慧化科技將會廣泛運用在生活和工作中。

對新科技總是很好奇的他們，雖然在學校沒有相關課程教授智慧科技的新知和技術，聽聞目前不需太多電子電路基礎就可以快速上手的套件。

而且將要上研究所的他們也聽說國外有越來越多學術研究的科學家，將創客工具如 Arduino 和 3D 列印等帶入實驗室製作低成本實驗設備，於是希望能在大學畢業前準備相關技能。

Project-Based Learning - 積木程式與智慧科技

在這個動機下，他們上網找到 Arduino UNO 開發板與感測器，於是集資買回來一起自學。他們透過學校課程和網路上的相關自學資源來學習。

- ▶ 首先將 Arduino UNO 與溫溼度感應器 DTH11 完成接線。
- ▶ 並學習 pySerial 套件的語法。

為了發揮團隊最大效益，每一個實驗都採取全體成員一起集思廣益的討論方式，務求完全瞭解軟硬體安裝、設定與程式邏輯。

Project-Based Learning - 積木程式與智慧科技

如果實驗失敗

- ▶ 他們決定先仔細仔細逐一檢查程式與接線有無錯誤，發現錯誤就立即修正並再次測試。

如果檢查沒錯仍無法解決問題

- ▶ 就上網下問題相關的關鍵字查詢網路上是否有人分享類似問題的解法。

Project-Based Learning - 積木程式與智慧科技

為求投資與工作時間最大效益，他們決定以下幾個工作目標：

1. 電腦 - Arduino UNO 開發板 - 溫溼度感應器組裝與接線等。
2. 安裝測試 pySerial 與 Arduino 開發板連線與訊號擷取等 Python 程式撰寫。
3. Python 程式撰寫 / Arduino 開發板 / 溫溼度資訊擷取，結合視窗介面程式撰寫。
4. 除溫溼度環境監測，他們也希望增加 Python / Arduino + 地震感測資料擷取。

Do it yourself and save: Open-source revolution is driving down the cost of doing science



Michigan Tech's Joshua Pearce with **a second-generation, open-source, 3D printer** called a Mendel RepRap.

Do it yourself and save: Open-source revolution is driving down the cost of doing science

The machine is made up of parts available in any local hardware store, open-source electronics available online, and parts that it can make for itself--all the red, white and blue components.

Pearce has saved thousands of dollars by building his own lab equipment with this machine and others like it.

Credit: Sarah Bird/Michigan Tech

Read more at: <https://phys.org/news/2012-09-open-source-revolution-science.html#jCp>



學習規劃

- ▣ 課程使用的學習法與成果要求
 - ▣ 以問題導向學習法 (Problem-based Learning) 方式，引導學生小組自學完成本教材的學習。
 - ▣ 自學投影片在 PBL 第五個步驟提供學生小組參考與分工。
 - ▣ 學生小組必須提供 Word 檔 PBL 1 - 6 步驟記名發言的小組討論報告一份 (單行間距，14號字，標楷體和 Times New Roman)。

PBL-7 Steps*

1. 閱讀問題-仔細閱讀問題，跟組員討論確認彼此確實了解問題。
2. 確認先前知識與技術-跟組員分享問題相關的先前（已知）知識與技術。
3. 提出問題相關的假設與機制。
4. 列出解決問題需要自學的新知與新技術。
5. 公平分配工作並分頭學習新知與新技術。
6. 組員一起將習得的新知與新技術予以整合來解決問題。
7. 組員自評與互評、反思與回饋。

*: [The Tutor in PBL, McMaster University, Canada](#)



PART 01

軟體準備步驟



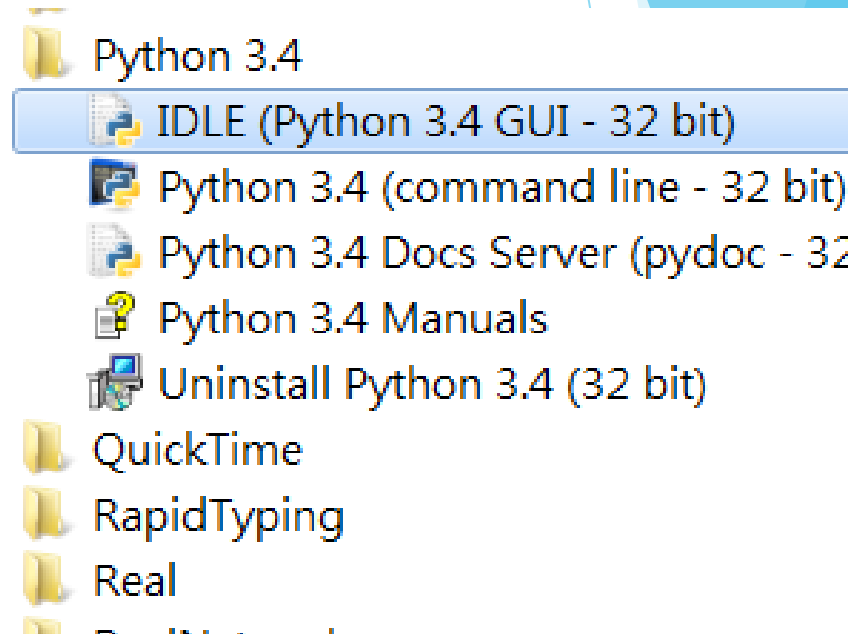
Python 版本

本教材是以

Python 3.4 來進行 Arduino UNO / DHT11 溫濕度感測的開發專案。

本專案安裝的 pySerial 模組也是針對 Python 3.4 版本。

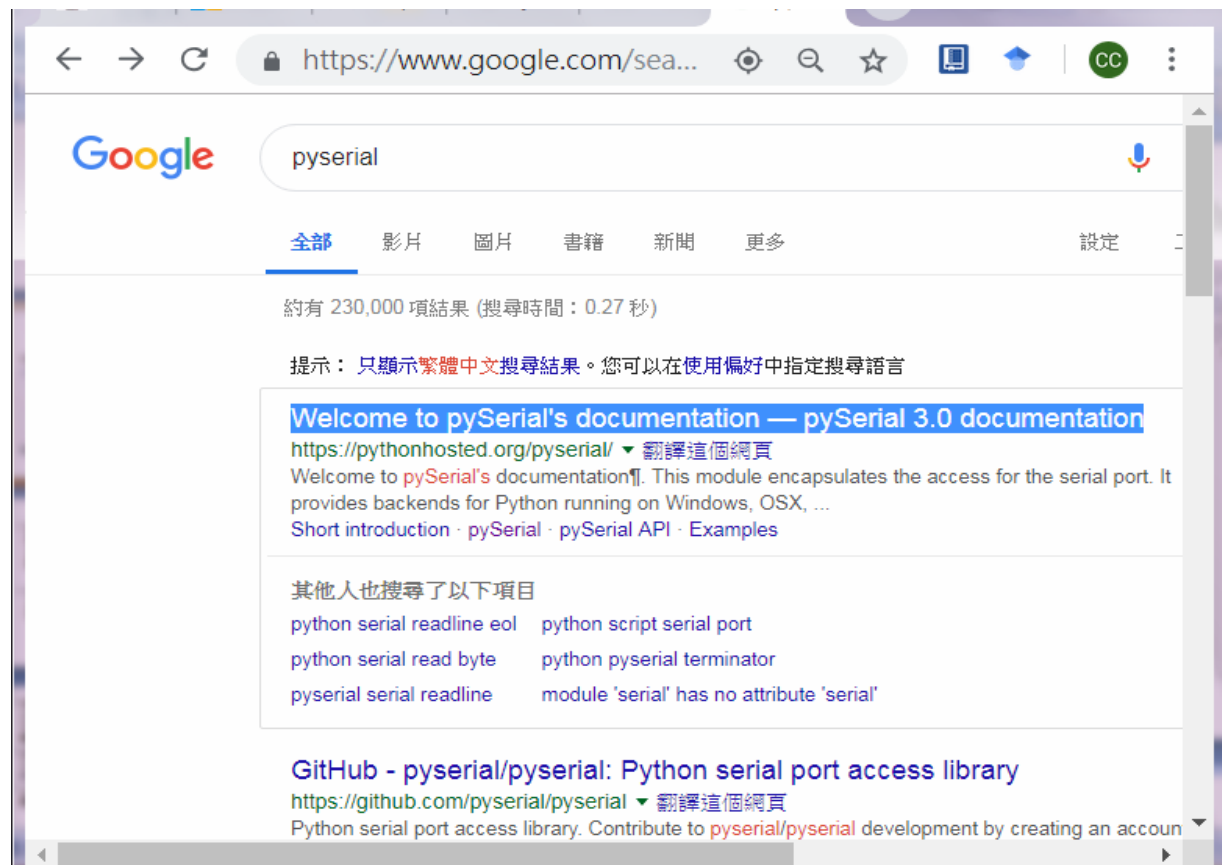
因此，建議使用本教材的初學者也安裝 Python 3.4 來學習，以免出現版本不同造成的諸多不相容問題。



[Python 3.4 官網下載連結](#)

pySerial: An Introduction

Google 搜尋引擎中鍵入 pyserial，出現下列搜尋結果



pySerial: An Introduction

選取第一個連結進入 pySerial 3.0 官方說明網頁

pySerial 3.0 documentation »

next | modules | index




Table Of Contents

Welcome to pySerial's documentation
Indices and tables

Next topic

pySerial

This Page

Show Source

Quick search

Enter search terms or a module, class or function name.

Welcome to pySerial's documentation

This module encapsulates the access for the serial port. It provides backends for [Python](#) running on Windows, OSX, Linux, BSD (possibly any POSIX compliant system) and IronPython. The module named "serial" automatically selects the appropriate backend.

Other pages (online)

- [project page on GitHub](#)
- [Download Page with releases](#)
- This page, when viewed online is at <https://pyserial.readthedocs.org/en/latest/> or <http://pythonhosted.org/pyserial/>.

Contents:

- pySerial
 - Overview
 - Features
 - Requirements
 - Installation
 - References
 - Older Versions
- Short introduction
 - Opening serial ports
 - Configuring ports later
 - Readline
 - Testing ports
- pySerial API
 - Classes
 - Exceptions

pySerial: An Introduction

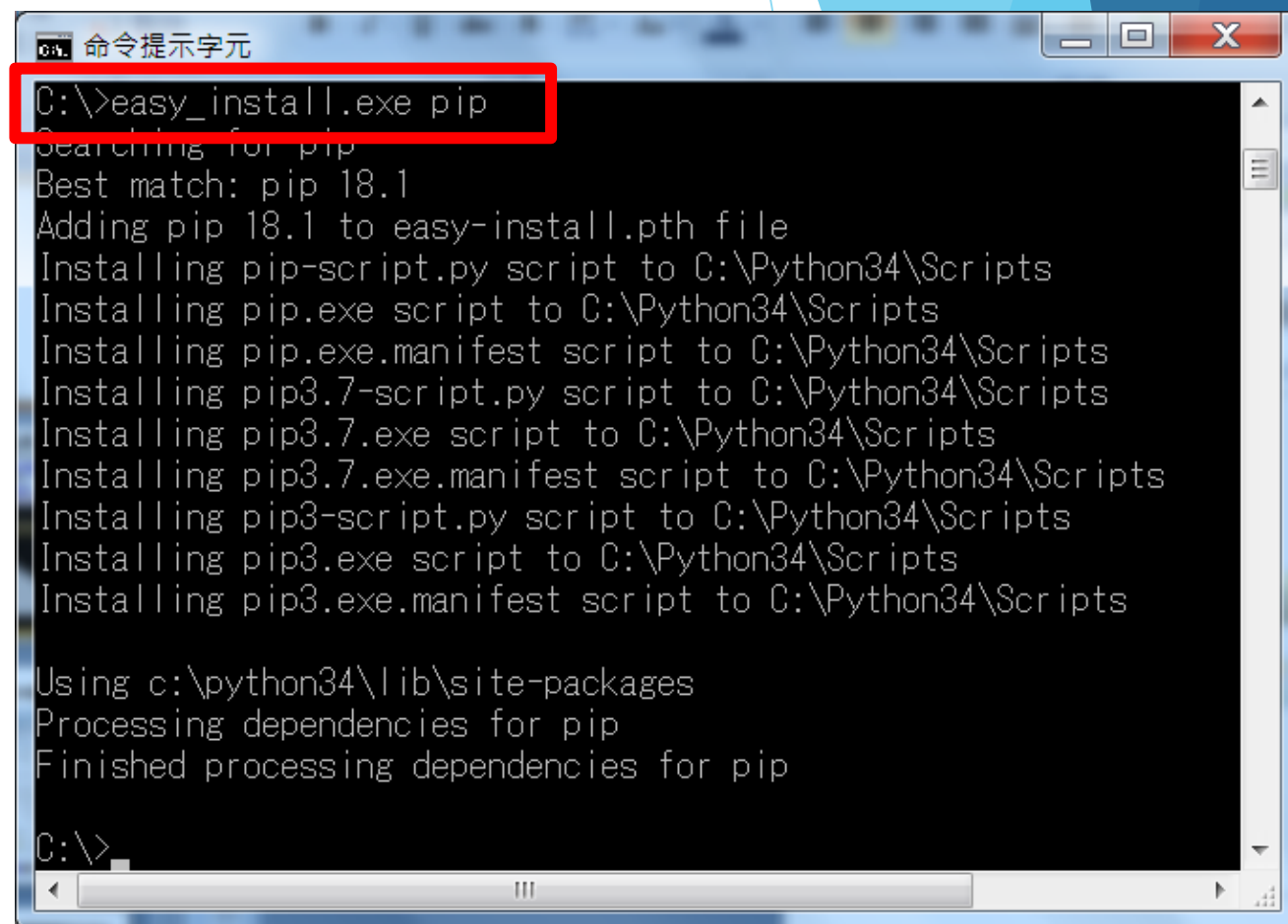
- ▶ 根據官網說明*，pySerial 是連接電腦序列埠 (COM Port) 的 Python 套件 (Package ，通常是一個檔案夾) 。
- ▶ 它提供 Python 在 Windows, Linux 等數種平台上執行，pySerial 實際安裝在電腦上的是名為 serial，內含許多模組 (module 或單一程式中有需要多次使用的一個到多個函式 [function] 或類別 [class]) 的檔案夾 (directory) 。
- ▶ Python 程式中，通常以 `import serial` 來匯入這個模組。

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

pySerial 模組安裝

- ▶ 安裝 pySerial 需要使用到管理 Python 套件或模組的 pip。
- ▶ 執行 pySerial 安裝前，建議先將 pip 更新到最新版。
- ▶ 利用 Python3.4 中 setuptools 中的 easy_install.exe 即可安裝並更新 pip (圖)。

```
C:\>easy_install.exe pip
```



The screenshot shows a Windows command prompt window titled "命令提示字元". The command `C:\>easy_install.exe pip` has been entered and is highlighted with a red rectangle. The output of the command is as follows:

```
C:\>easy_install.exe pip
Searching for pip
Best match: pip 18.1
Adding pip 18.1 to easy-install.pth file
Installing pip-script.py script to C:\Python34\Scripts
Installing pip.exe script to C:\Python34\Scripts
Installing pip.exe.manifest script to C:\Python34\Scripts
Installing pip3.7-script.py script to C:\Python34\Scripts
Installing pip3.7.exe script to C:\Python34\Scripts
Installing pip3.7.exe.manifest script to C:\Python34\Scripts
Installing pip3-script.py script to C:\Python34\Scripts
Installing pip3.exe script to C:\Python34\Scripts
Installing pip3.exe.manifest script to C:\Python34\Scripts

Using c:\python34\lib\site-packages
Processing dependencies for pip
Finished processing dependencies for pip

C:\>
```

pySerial 模組安裝

在命令提示字元視窗中鍵入

pip install pyserial

按 Enter 後，出現右圖訊息

首先會從

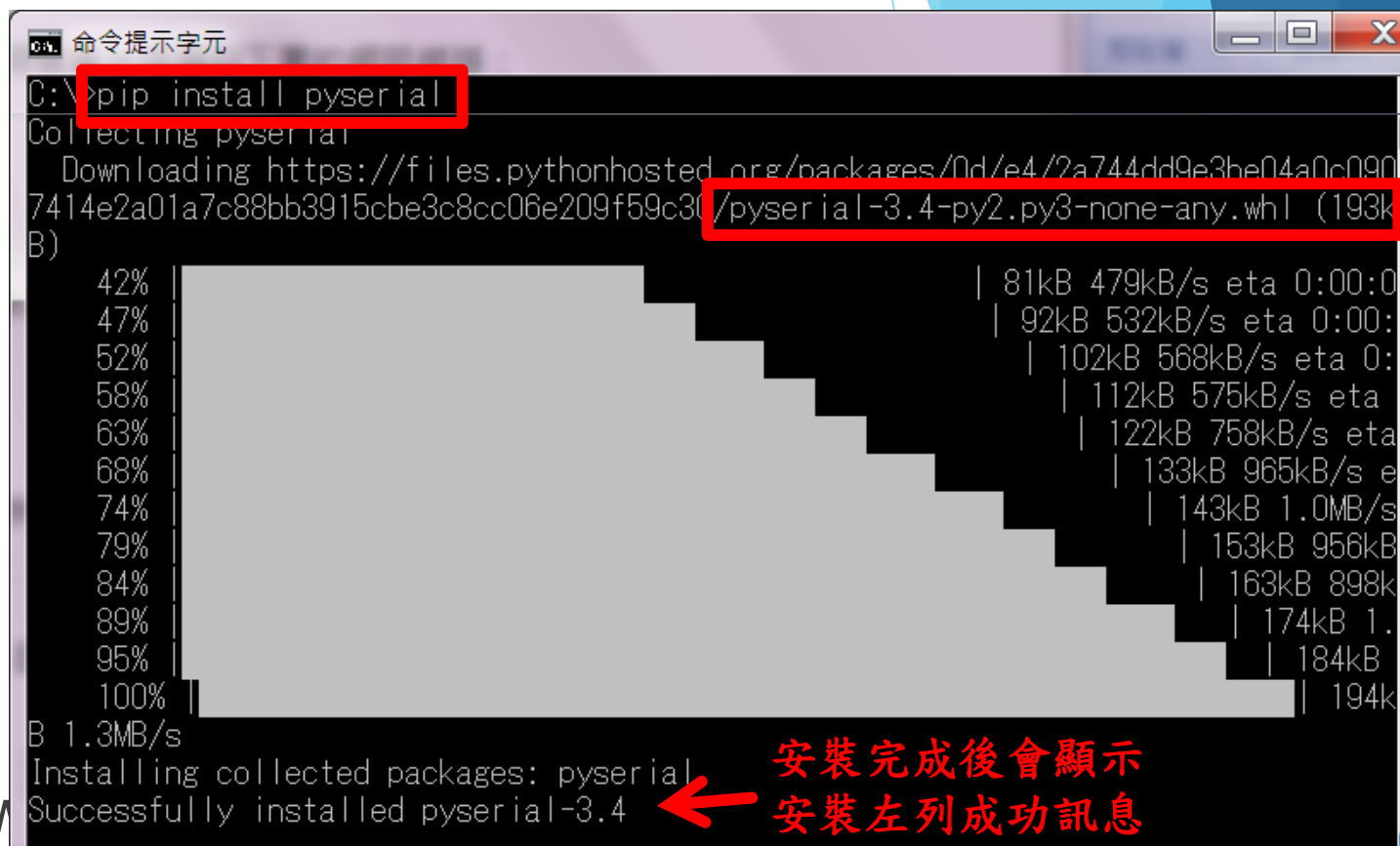
<https://files.pythonhosted.org>

網頁下載

pySerial-3.4-py2.py3-none-any.whl

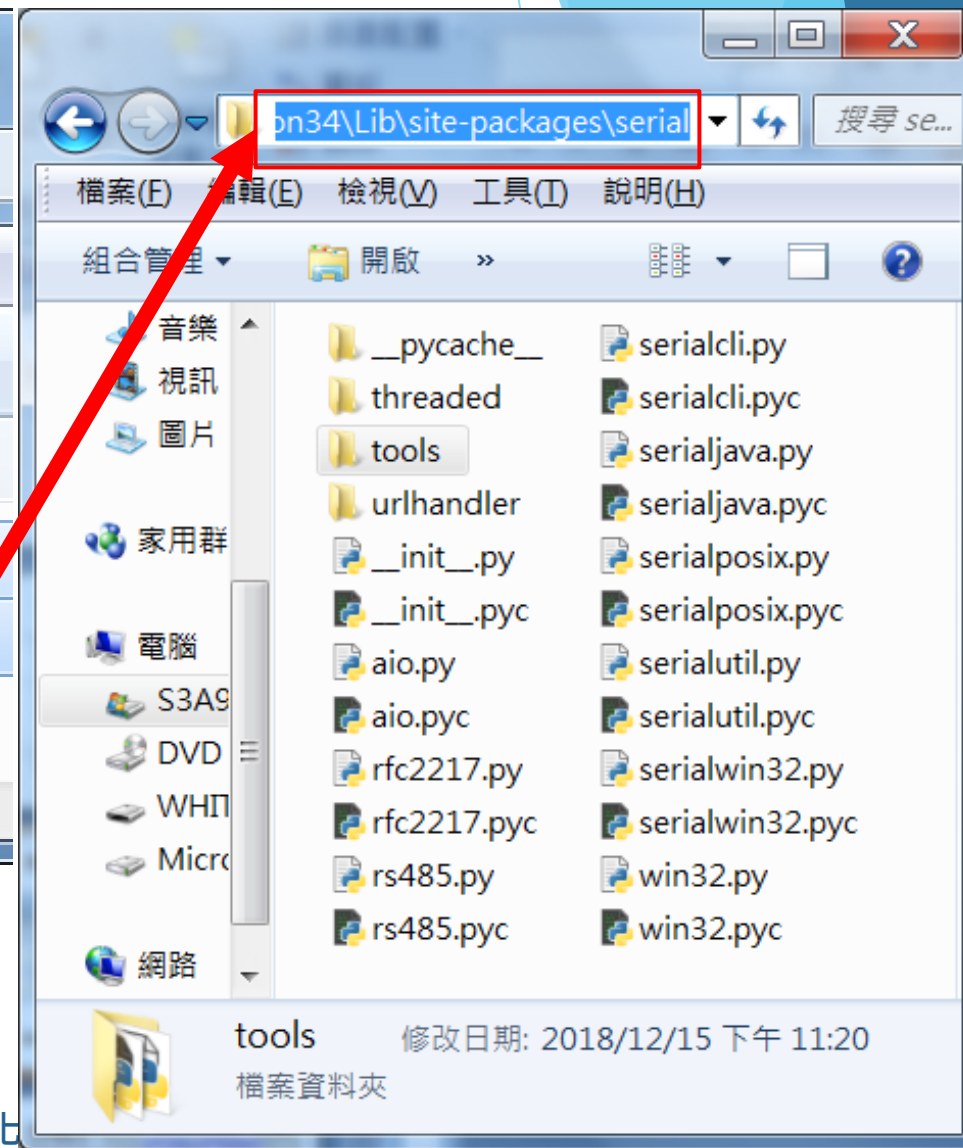
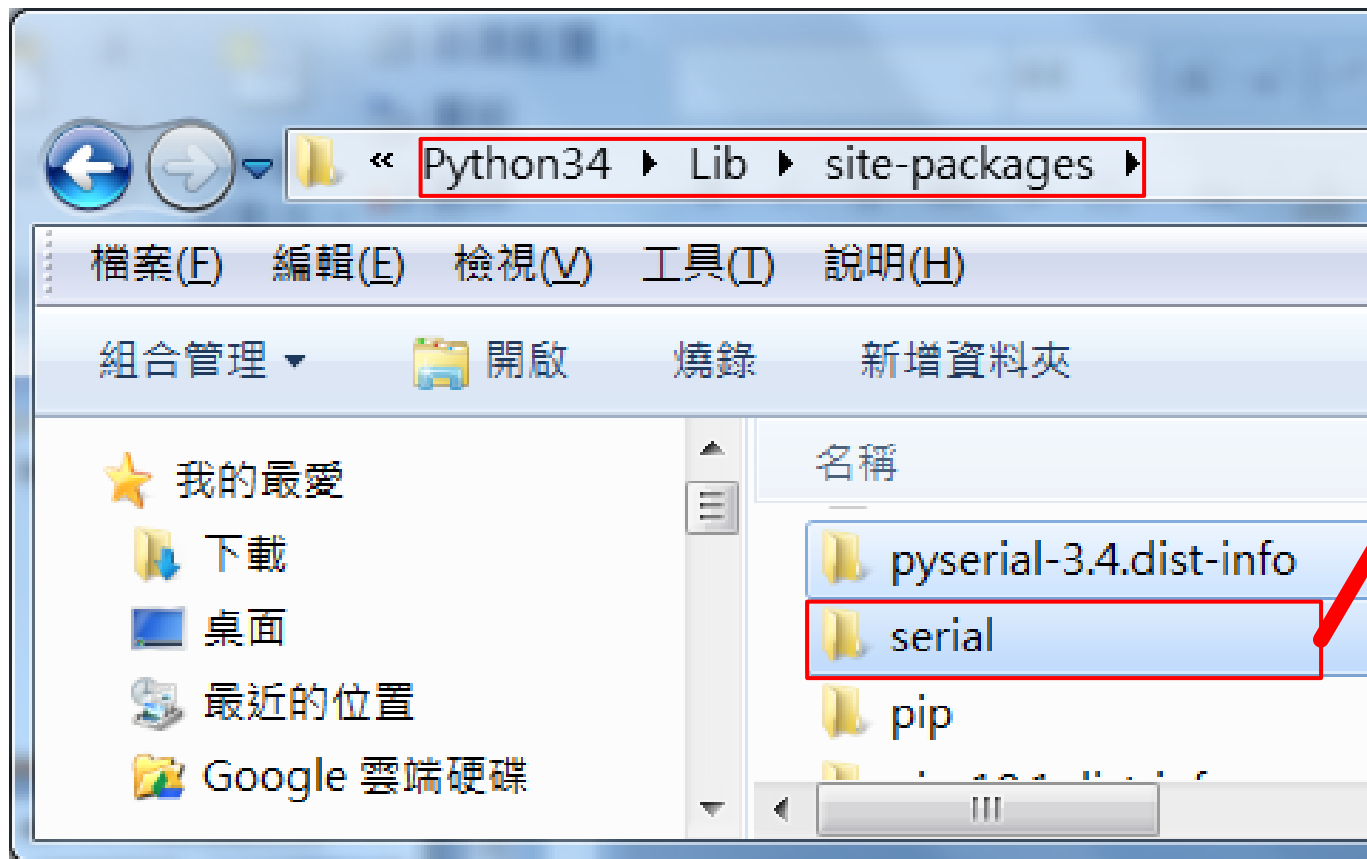
下載完成後接著自動安裝 pySerial。

(或可以直接到[官網下載](#))



```
命令提示字元
C:\>pip install pyserial
Collecting pyserial
  Downloading https://files.pythonhosted.org/packages/0d/e4/2a744dd9e3be04a0c0907414e2a01a7c88bb3915cbe3c8cc06e209f59c30/pyserial-3.4-py2.py3-none-any.whl (193kB)
    42% |#####| 81kB 479kB/s eta 0:00:0
    47% |#####| 92kB 532kB/s eta 0:00:0
    52% |#####| 102kB 568kB/s eta 0:00:0
    58% |#####| 112kB 575kB/s eta 0:00:0
    63% |#####| 122kB 758kB/s eta 0:00:0
    68% |#####| 133kB 965kB/s eta 0:00:0
    74% |#####| 143kB 1.0MB/s eta 0:00:0
    79% |#####| 153kB 956kB/s eta 0:00:0
    84% |#####| 163kB 898kB/s eta 0:00:0
    89% |#####| 174kB 1.0MB/s eta 0:00:0
    95% |#####| 184kB 1.0MB/s eta 0:00:0
    100% |#####| 194kB 1.3MB/s
Installing collected packages: pyserial
Successfully installed pyserial-3.4
```


pySerial 套件安裝後 serial 檔案夾位置



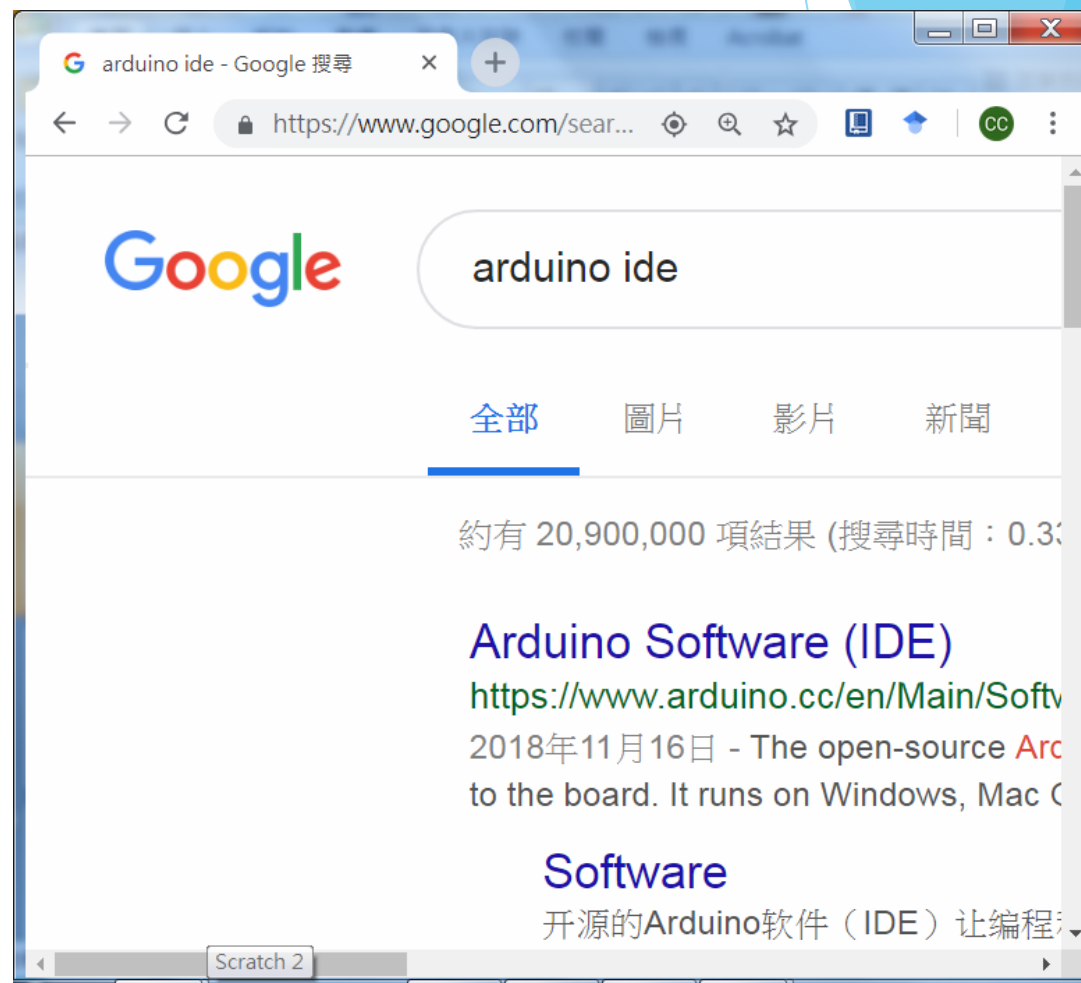
詳細的 pySerial API 說明可到[英文官網](#)閱讀

運用 Arduino IDE 設定開發板與序列埠的 COM Port

- ▶ Arduino 開發板有多種，Arduino UNO 只是其中一個基礎的開發板，不同的開發板有不同的規格。
- ▶ 利用 Arduino 官網提供的免費軟體來幫電腦辨識是哪一塊 Arduino 開發板並告訴電腦使用哪一個 COM Port 來連接，可以幫助初學者克服 COM Port 設定問題。

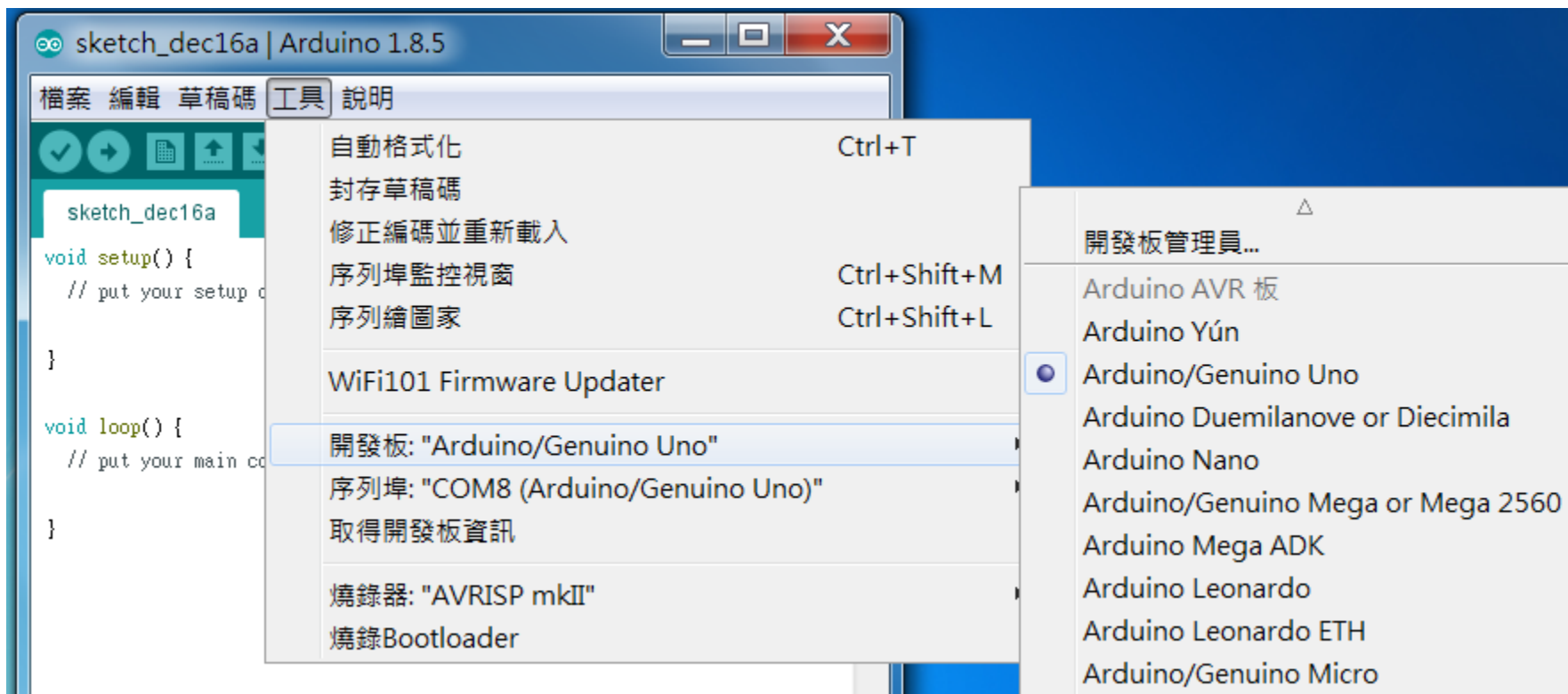
運用 Arduino IDE 設定開發板與序列埠的 COM Port

► Arduino IDE 官網免費下載



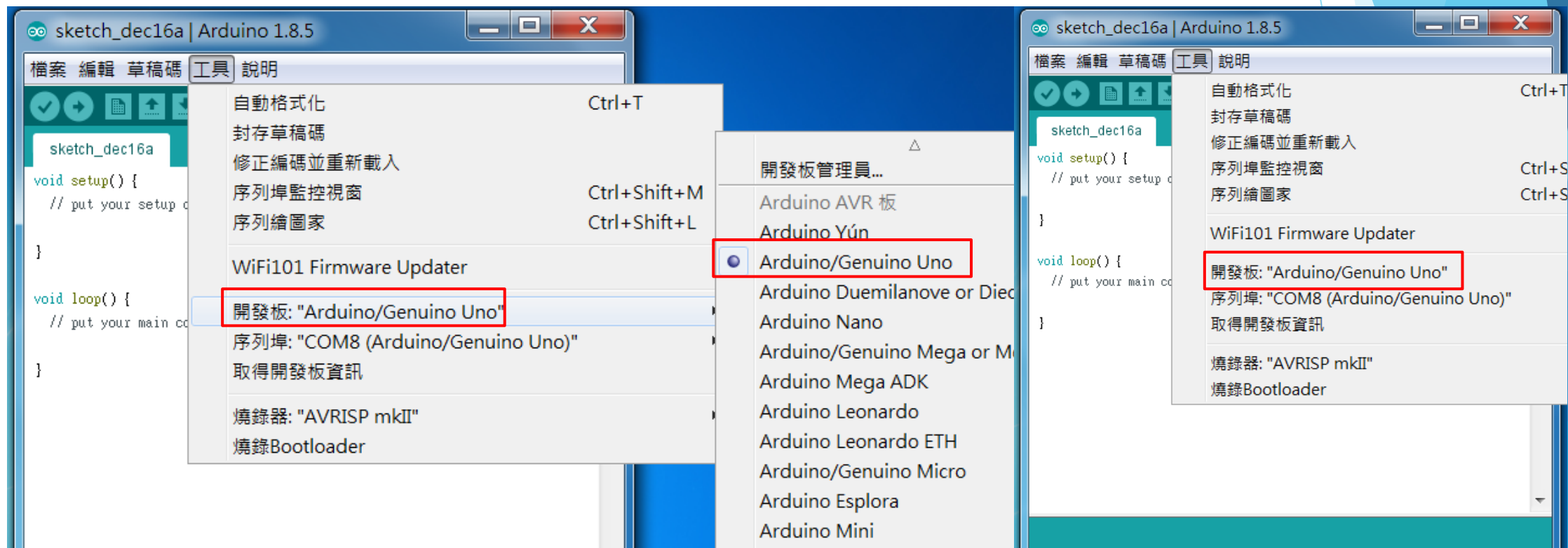
Arduino IDE 軟體安裝與設定

- ▶ 首次安裝 Arduino IDE 開啟後，在上面選單選擇工具，滑鼠下移到開發板，選擇 **Arduino / Genuino Uno**。



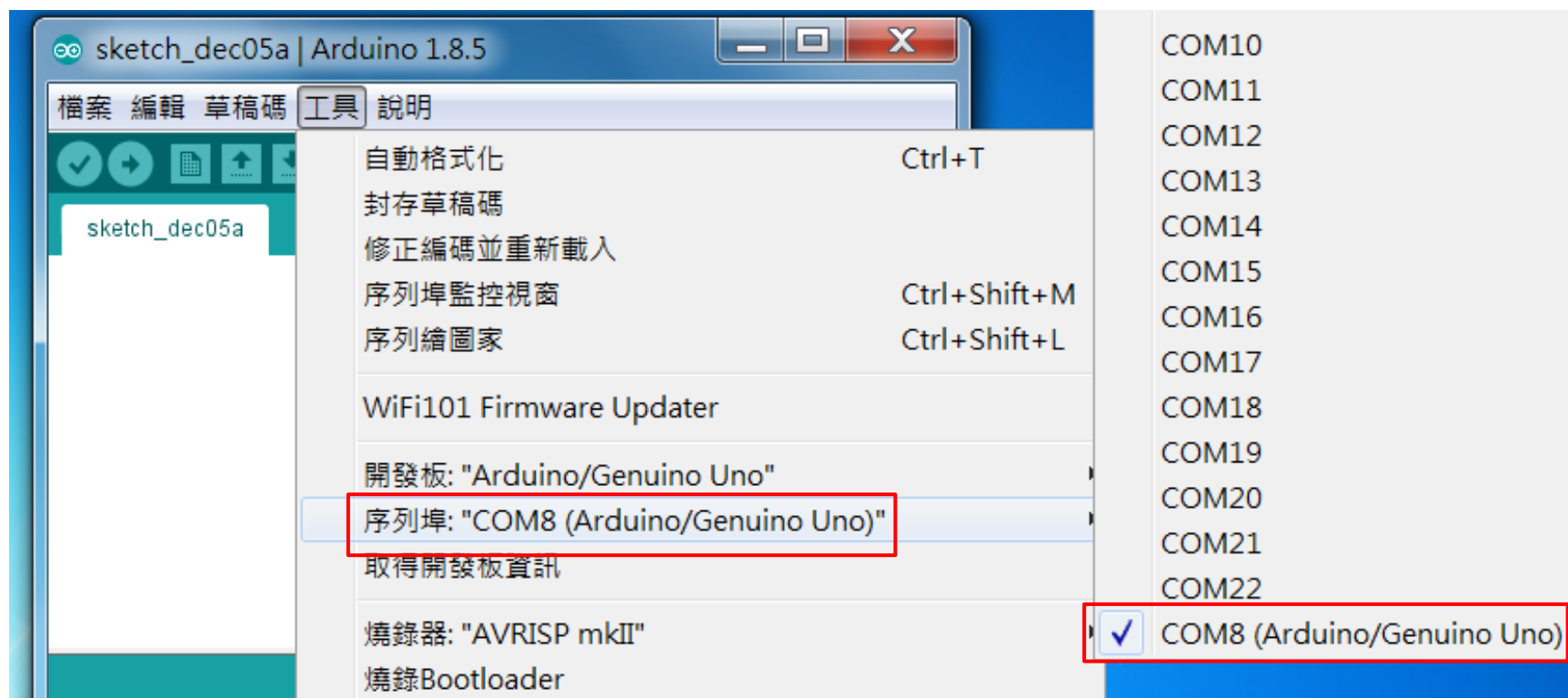
Arduino IDE 軟體安裝與設定

- 紅色框處開發板: 後面出現 **Arduino / Genuino Uno**



Arduino IDE 選單 / 工具 / 選取序列埠

- ▶ 選單選擇工具，滑鼠下移到序列埠，選擇有 (**Arduino / Genuino Uno**) 的 COM。



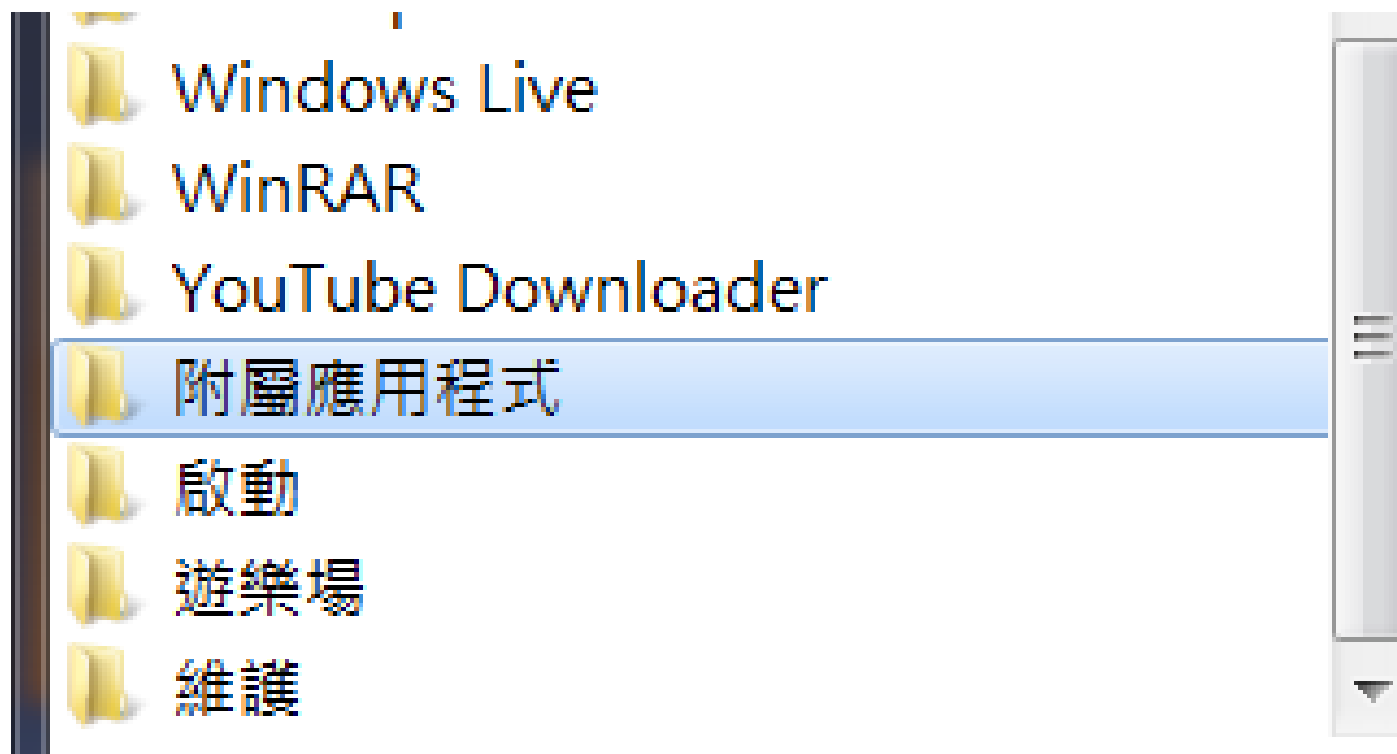
PART 02

硬體準備步驟



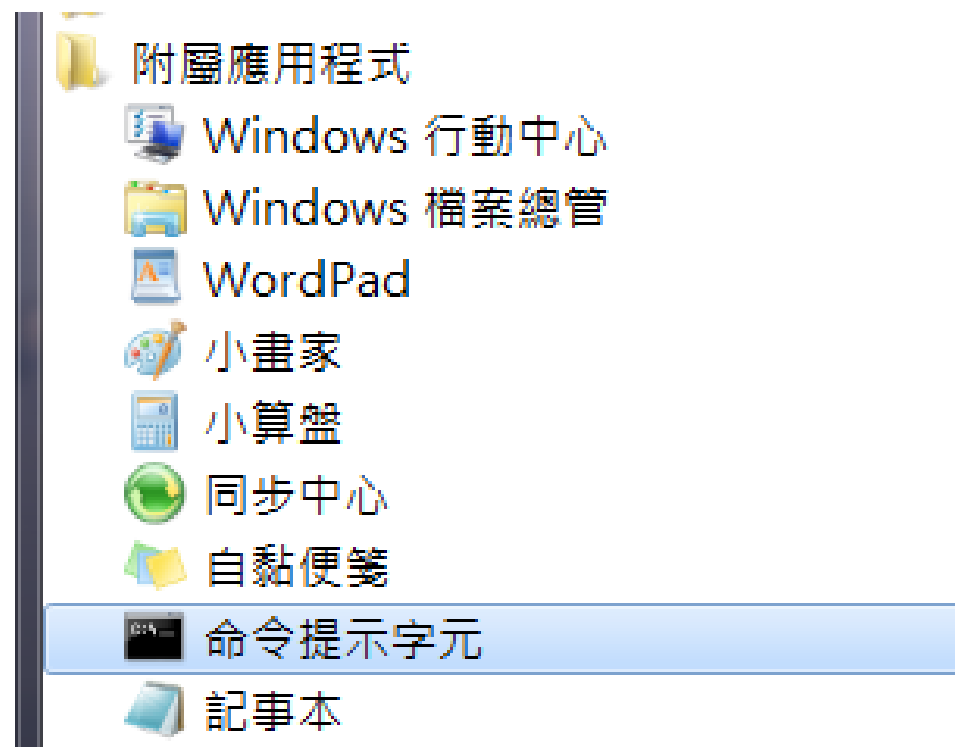
命令提示字元

- ▶ 按下 Windows 左下角 Start (開始) 鍵，螢幕左邊出現如右圖的選單，將滑鼠移到**附屬應用程式**上。



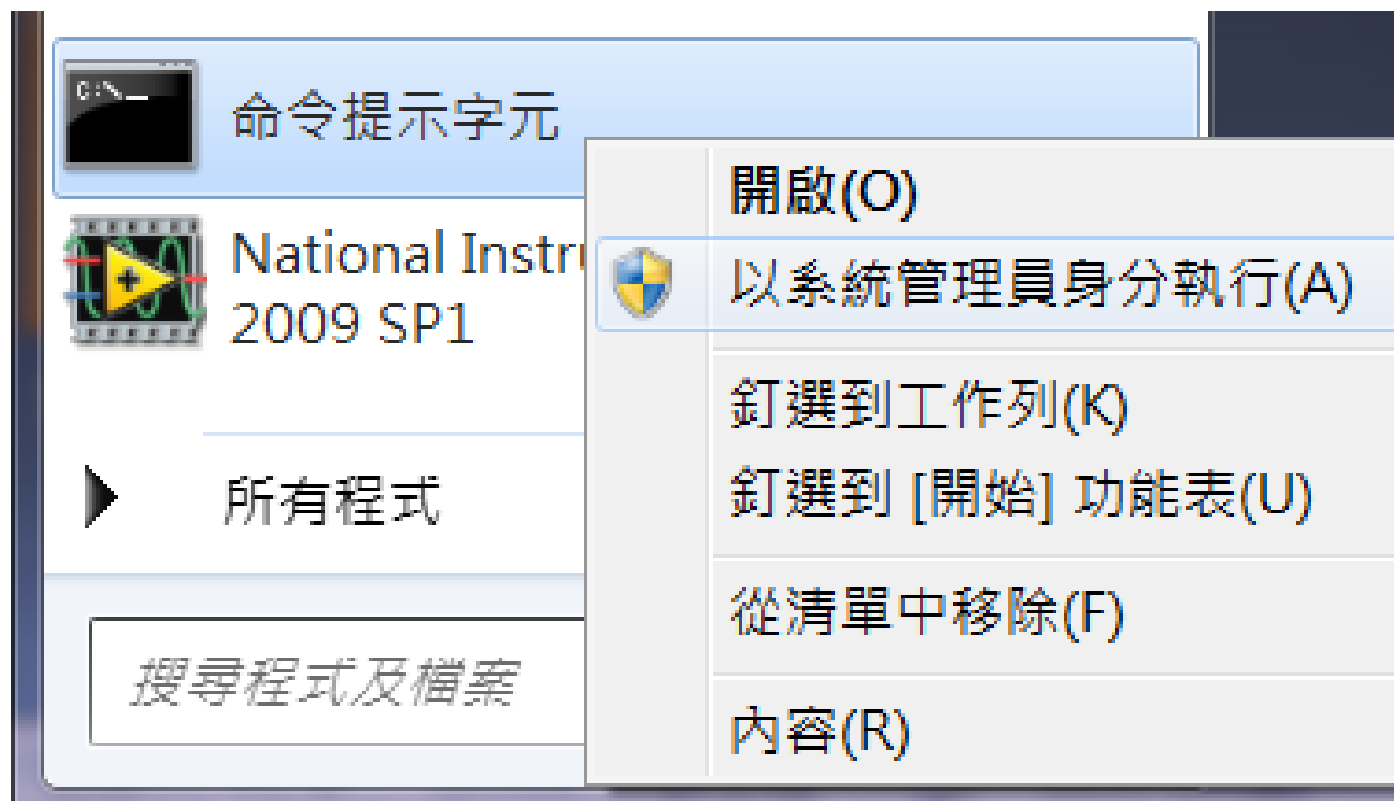
命令提示字元

- ▶ 用滑鼠點選**附屬應用程式**後，出現右下的選單，滑鼠移動到**命令提示字元**上。



命令提示字元

- ▶ 在命令提示字元上按右鍵後出現圖中右邊的選單，以滑鼠選取以系統管理員身分執行。



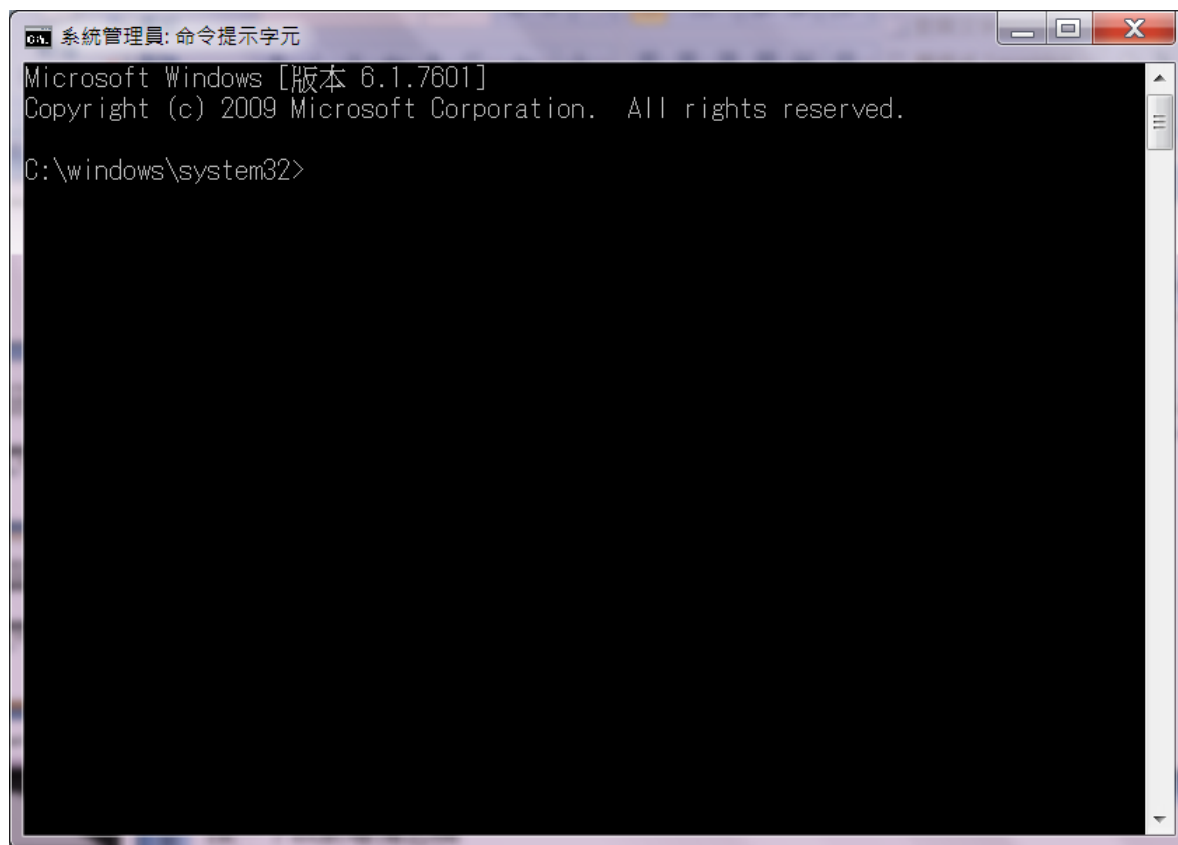
命令提示字元

- ▶ 螢幕上出現使用者帳戶控制視窗（圖），以滑鼠選取是（Y）按鍵。



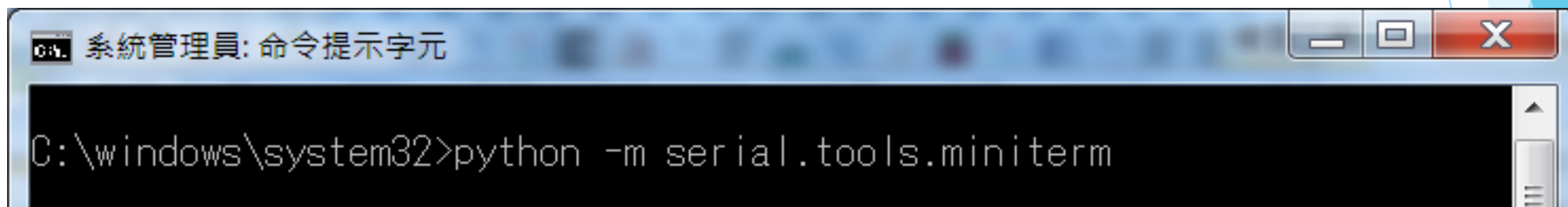
命令提示字元

- ▶ 出現系統管理員：命令提示字元視窗。



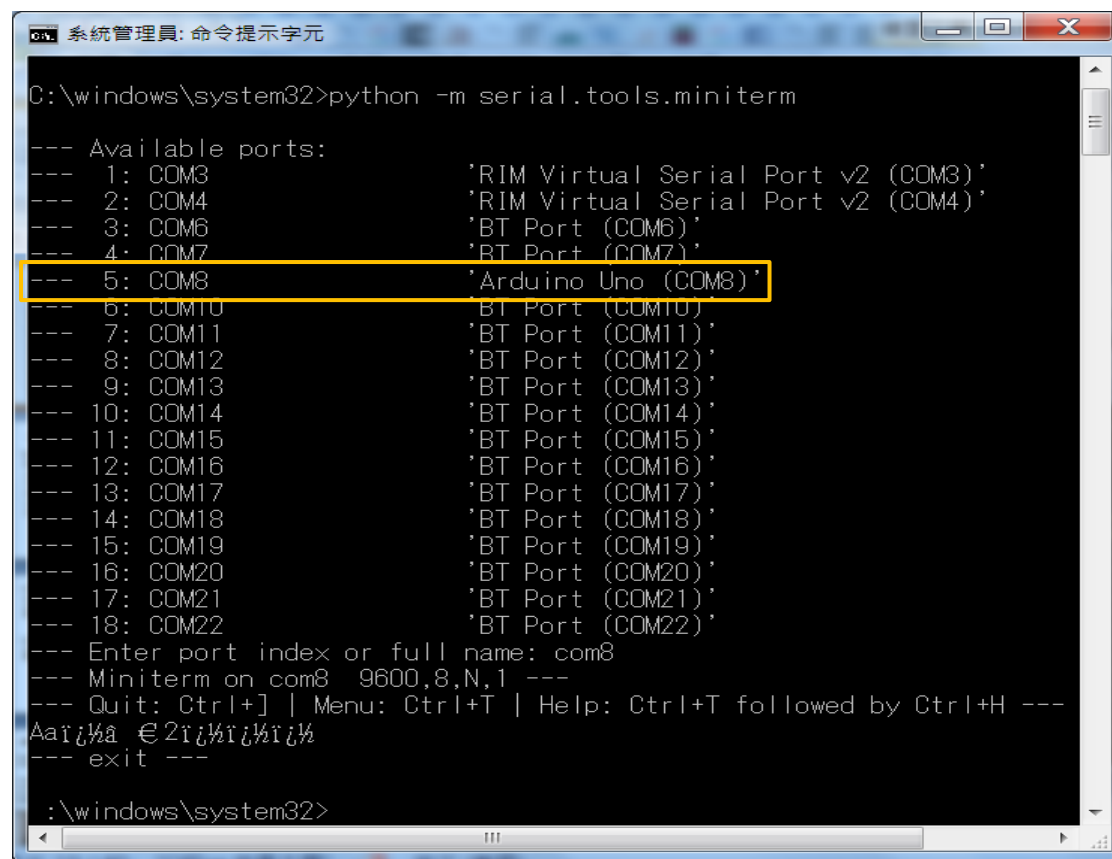
序列埠目前使用情形

- ▶ 在命令提示字元視窗 prompt 後鍵入
`python -m serial.tools.miniterm`



序列埠目前使用情形

► 出現下圖中各序列埠目前使用情形



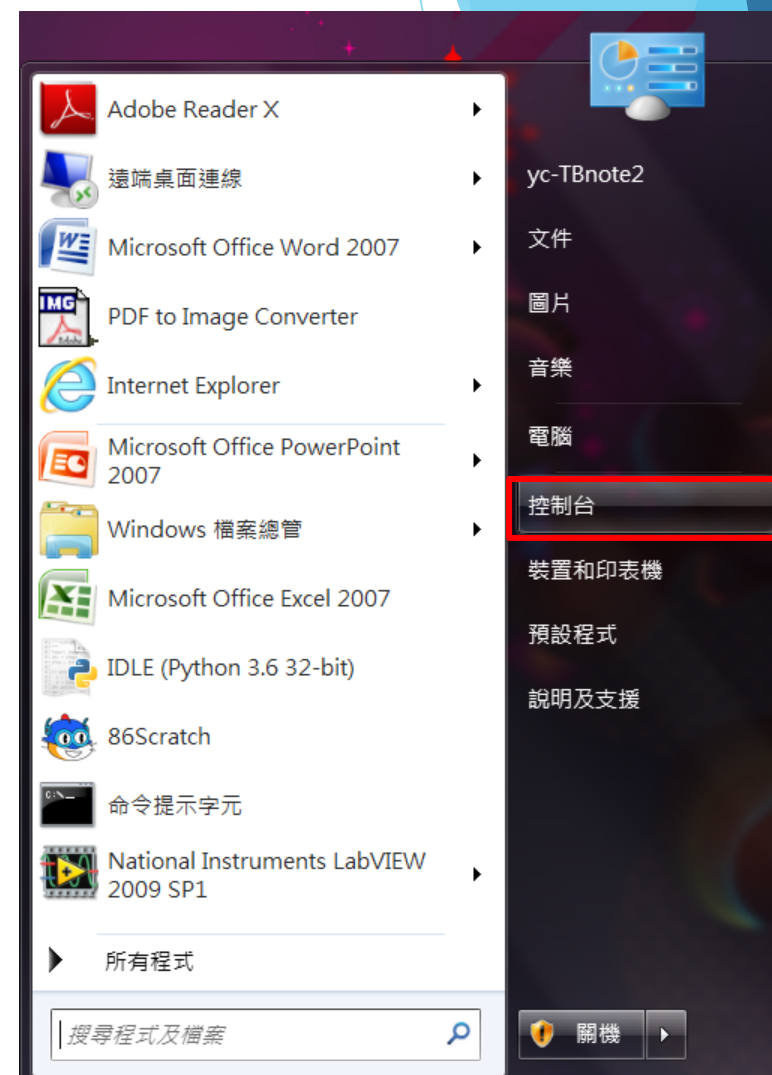
```
系統管理員: 命令提示字元
C:\windows\system32>python -m serial.tools.miniterm

--- Available ports:
--- 1: COM3          'RIM Virtual Serial Port v2 (COM3)'
--- 2: COM4          'RIM Virtual Serial Port v2 (COM4)'
--- 3: COM6          'BT Port (COM6)'
--- 4: COM7          'BT Port (COM7)'
--- 5: COM8          'Arduino Uno (COM8)'
--- 6: COM10         'BT Port (COM10)'
--- 7: COM11         'BT Port (COM11)'
--- 8: COM12         'BT Port (COM12)'
--- 9: COM13         'BT Port (COM13)'
--- 10: COM14        'BT Port (COM14)'
--- 11: COM15        'BT Port (COM15)'
--- 12: COM16        'BT Port (COM16)'
--- 13: COM17        'BT Port (COM17)'
--- 14: COM18        'BT Port (COM18)'
--- 15: COM19        'BT Port (COM19)'
--- 16: COM20        'BT Port (COM20)'
--- 17: COM21        'BT Port (COM21)'
--- 18: COM22        'BT Port (COM22)'
--- Enter port index or full name: com8
--- Miniterm on com8 9600,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
Aa;ikâ €2i;ik;ik;ik
--- exit ---

C:\windows\system32>
```

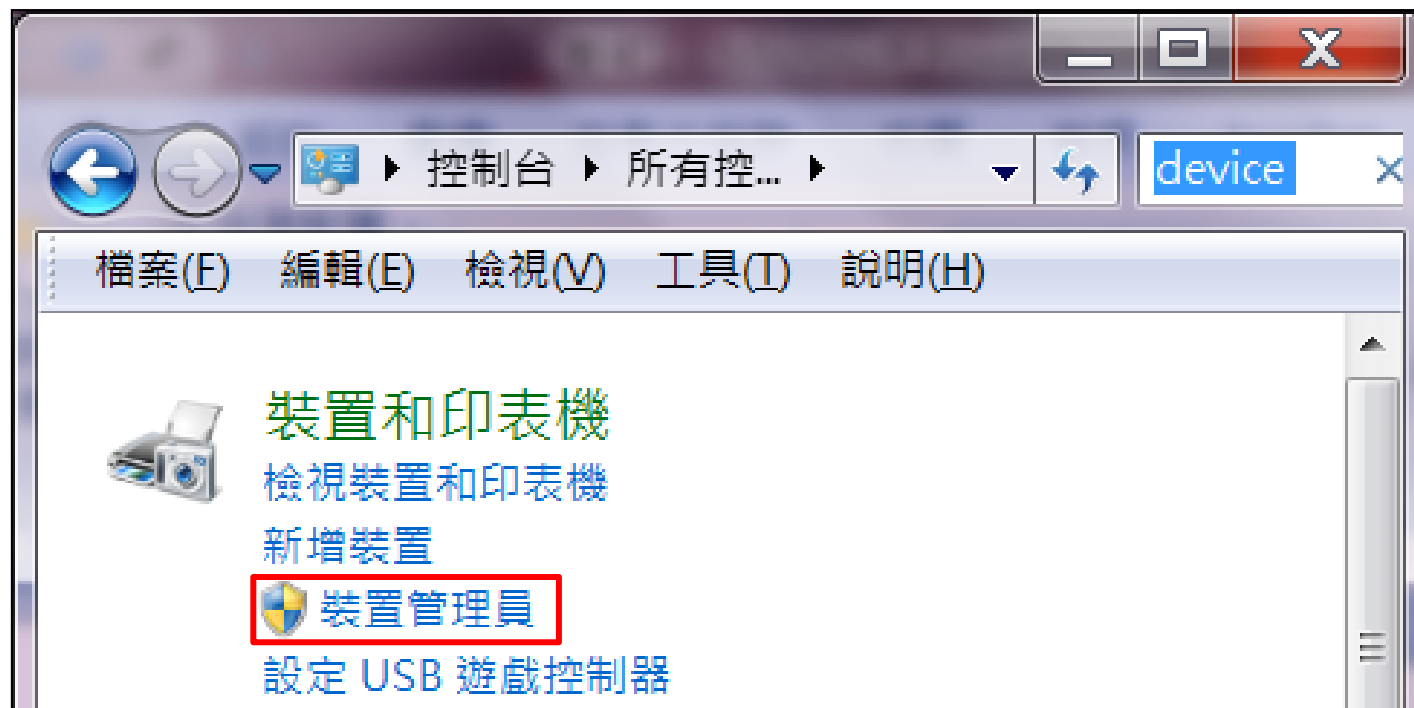
序列埠或連接埠 (Serial or COM Port)

- ▶ 按下 Windows 左下角 Start (開始) 鍵。
- ▶ 螢幕左邊出現如右圖的選單。
- ▶ 將滑鼠移到控制台上。
- ▶ 選取控制台。



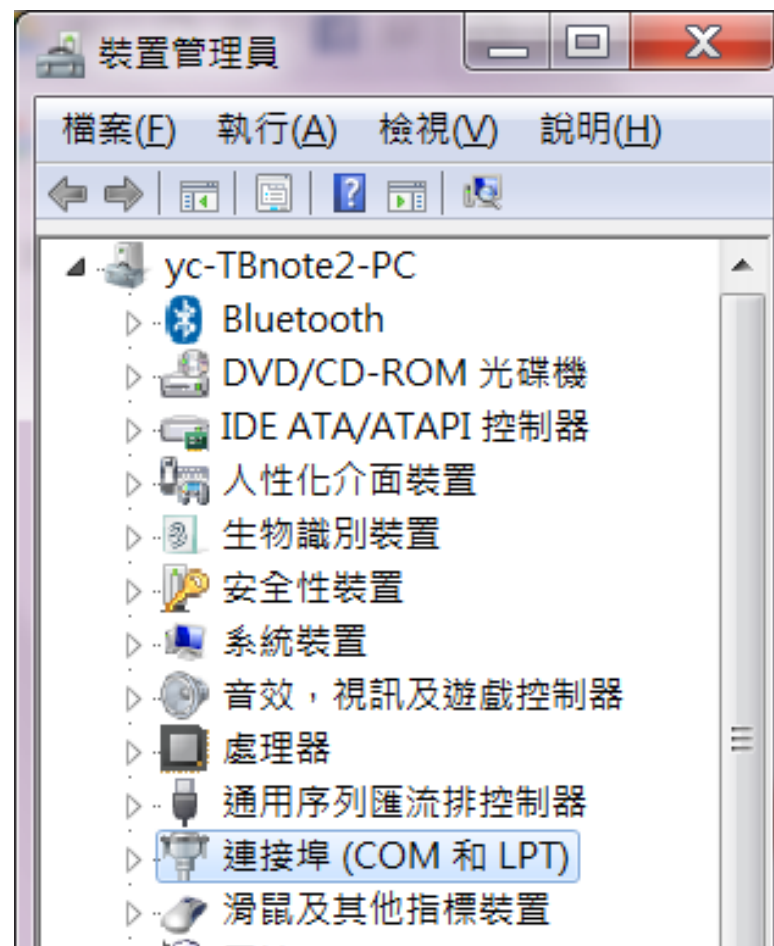
序列埠或連接埠 (Serial or COM Port)

- ▶ 右圖右上角鍵入 **device** 後，下方出現如圖的選單，將滑鼠移到**裝置管理員**上。



序列埠或連接埠 (Serial or COM Port)

- ▶ 點選裝置管理員後出現右圖
- ▶ 將滑鼠移到
- ▶ 連接埠 (**COM和LPT**)。

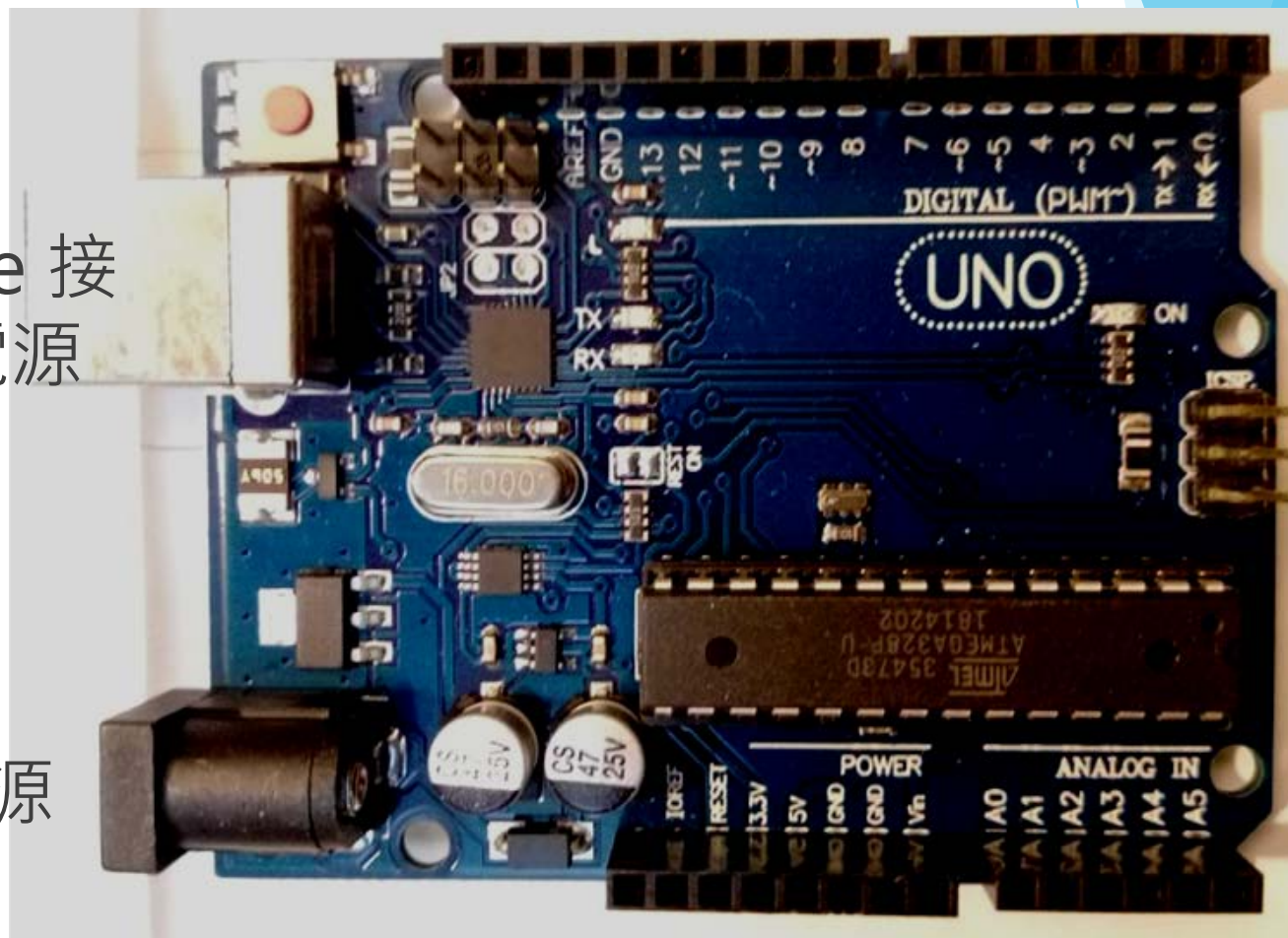


Arduino UNO board

- ▶ 電子材料行購買的平價 Arduino UNO 開發板如圖。

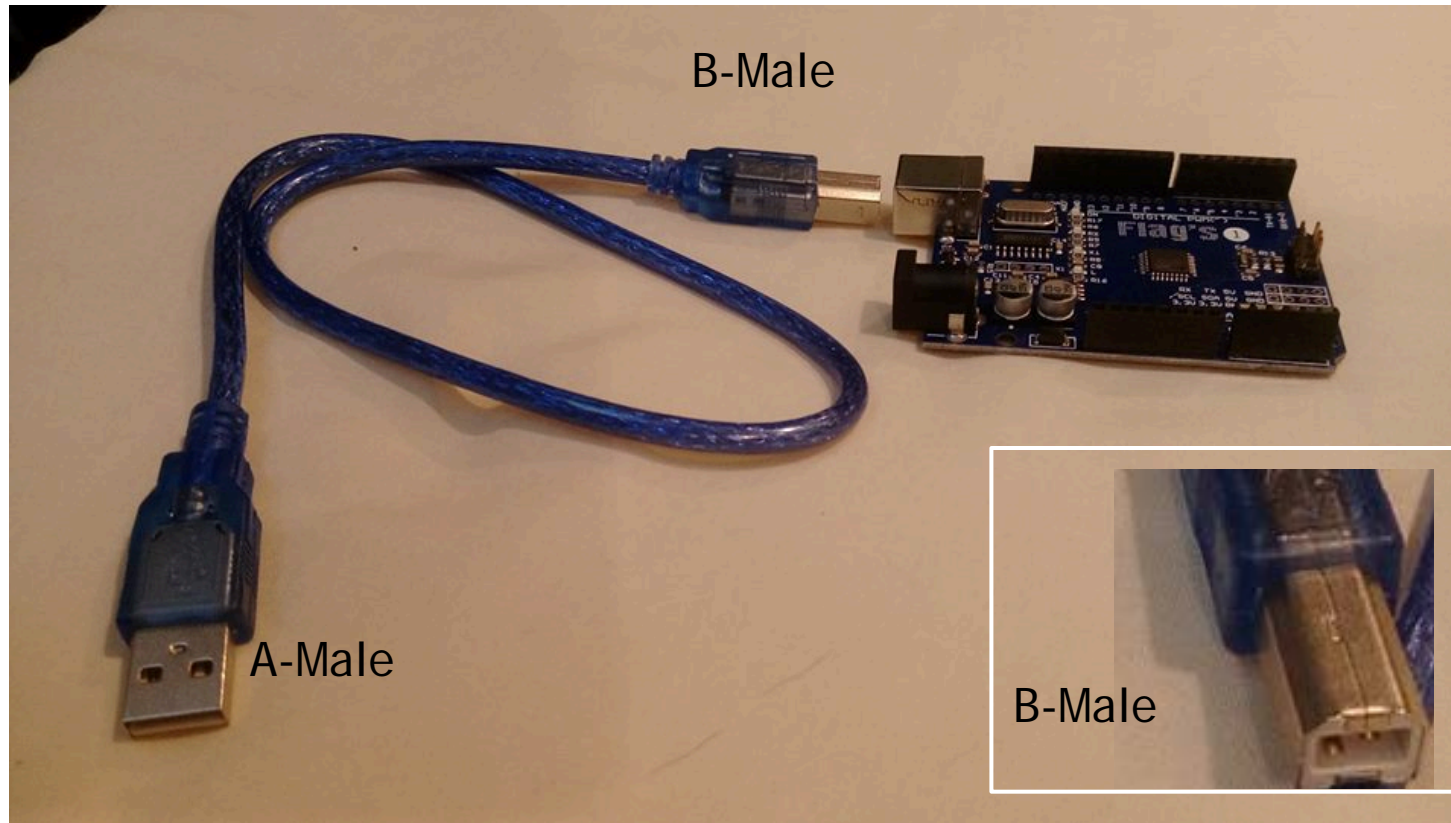
透過 USB 2.0 A / B Cable 接
電腦傳輸資訊或接行動電源

外接電源



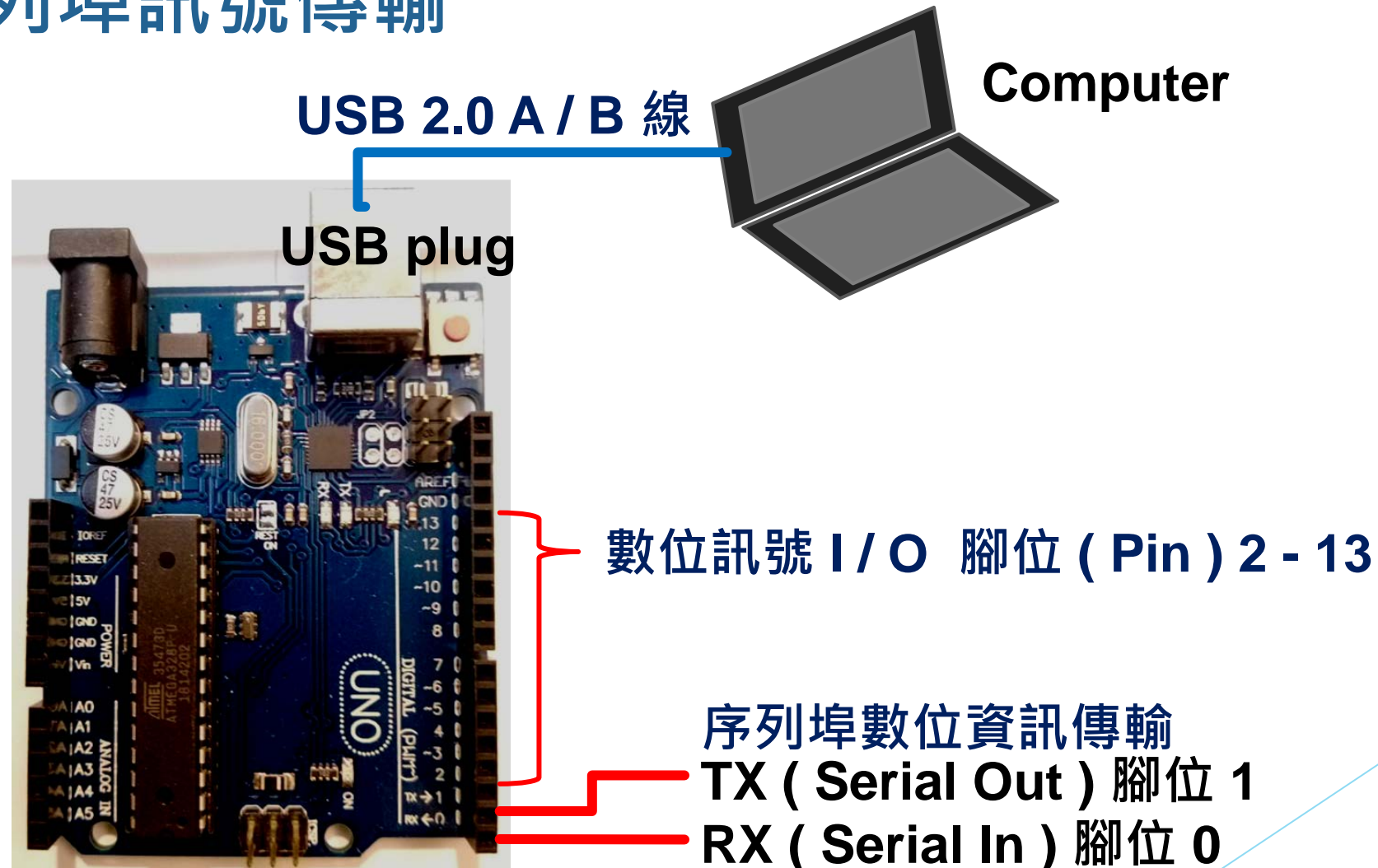
Arduino UNO board

- ▶ Arduino UNO 是最基礎的 Arduino 開發板。下圖左邊的藍色線是 A-Male to B-Male Cable。



Arduino to PC Serial Communication

序列埠訊號傳輸



USB 2.0 Cable Type A / B

- ▶ USB 是 Universal Serial Bus 的英文縮寫，中文譯名為通用序列匯流排（參考資料）。
- ▶ USB 是廣泛使用於連接電腦與外部裝置的序列埠（Serial Port）匯流的技術規範或標準之一（參考資料）。

Material design of Blue USB Cable (Look inside of the cable)

USB 2.0 Cable Type A / B

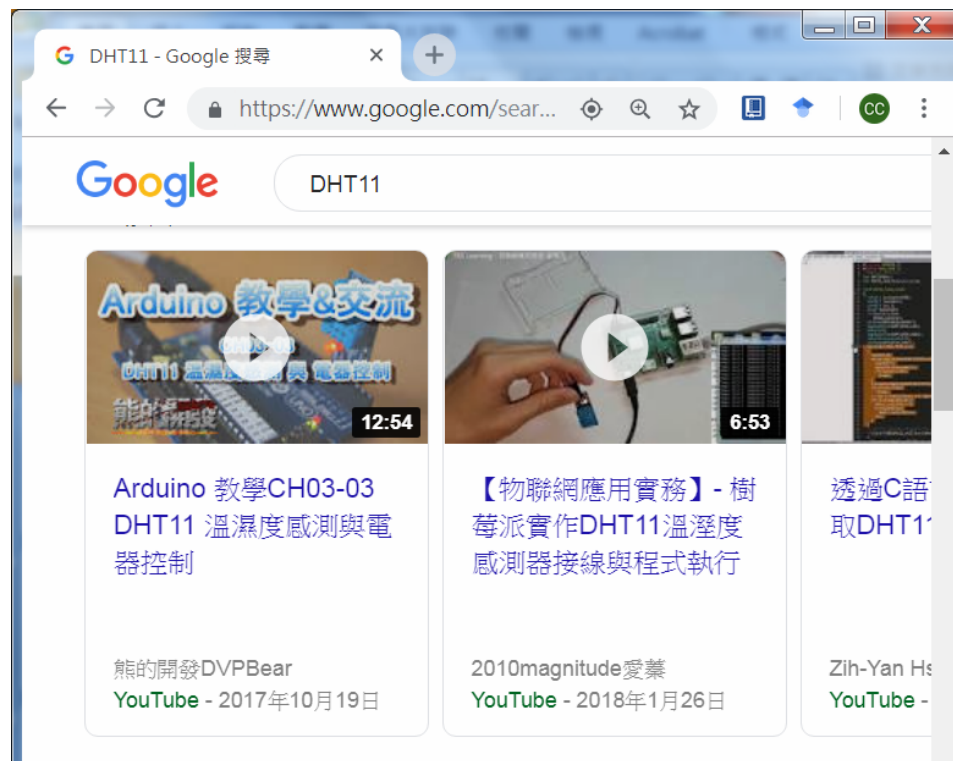
- ▶ USB 在資訊傳輸速度上，遠比傳統電腦用標準匯流排如 RS-232 等快許多 (參考資料概述)。

Material design of Blue USB Cable (Look inside of the cable)



DHT11 溫濕度感測器

Google 瀏覽器下輸入 DHT11 進行搜尋，可以找到許多相關的影片與網頁介紹這個感測器。



DHT11 溫濕度感測器

其中一個網頁根據 DHT11 的 Datasheet，介紹了 DHT11 的詳細規格，譬如，DHT11 是電阻感應濕度的高低，以 NTC 感應溫度，以數位方式輸出。



VCC: 3-5VDC 電源輸入

DATA: 數位訊號傳輸

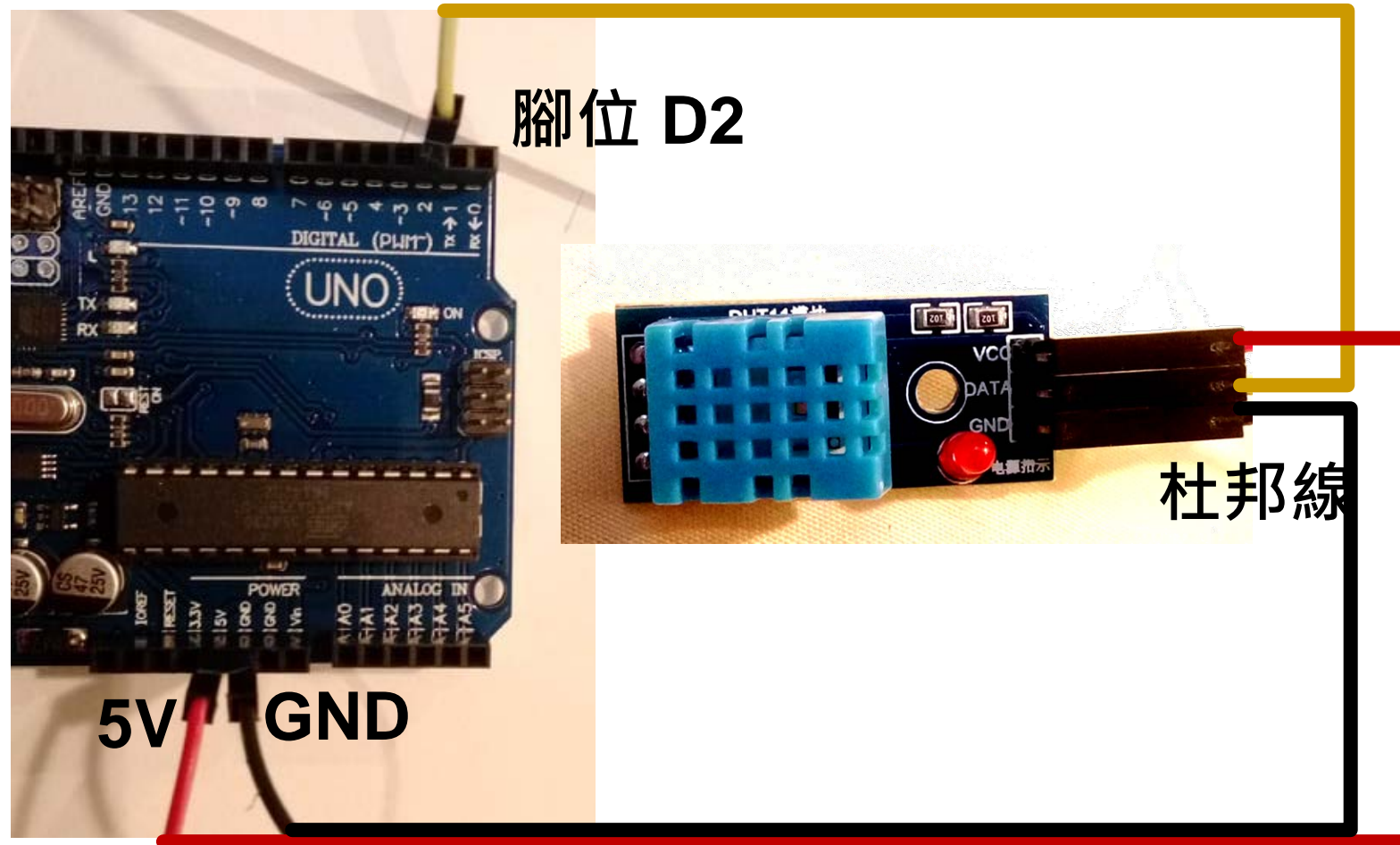
GND: 接地

RH Range: 20-90%

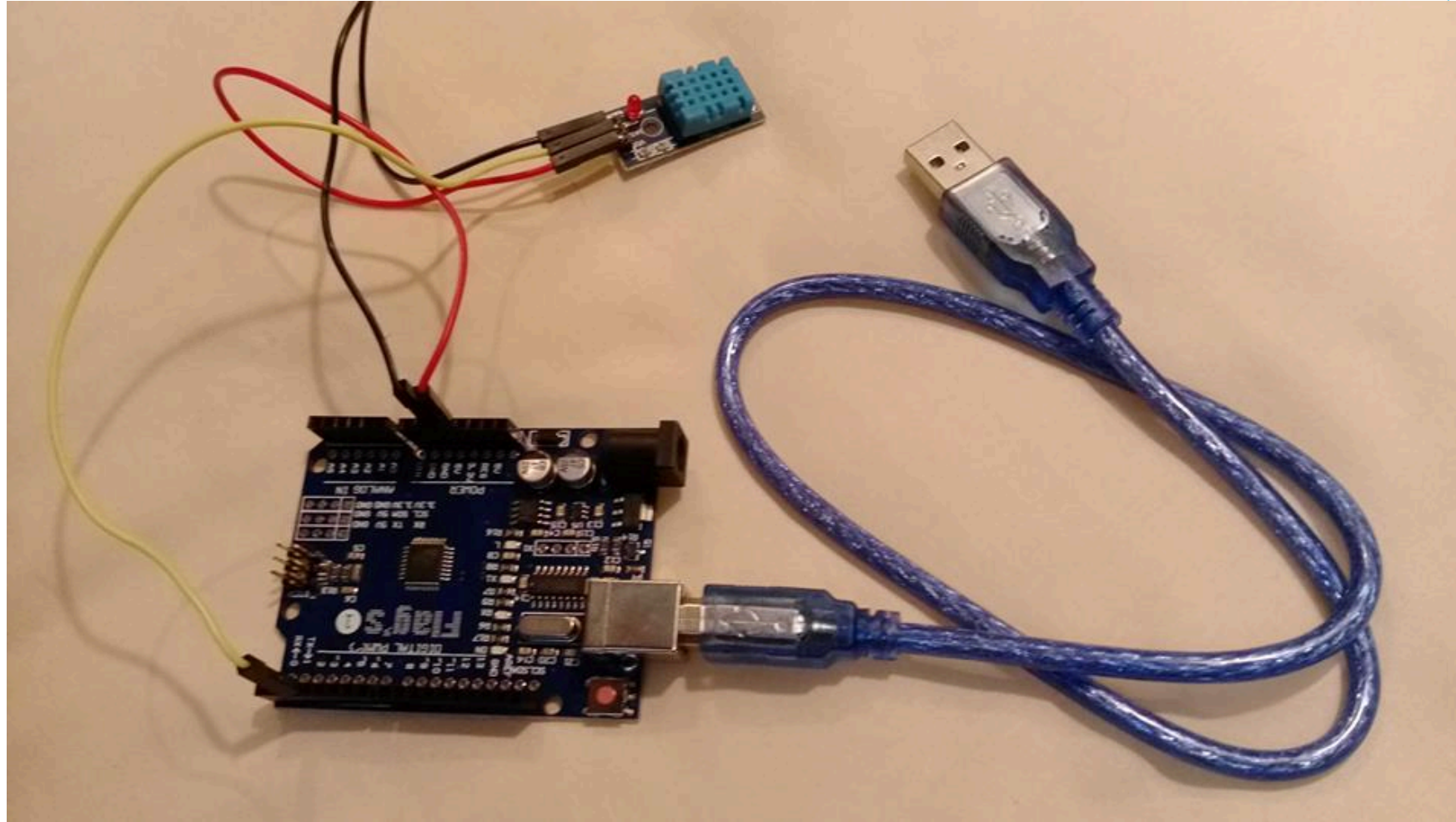
RH Accuracy: $\pm 5\%$ RH.

Temperature Range: 0-50 °C

Arduino UNO 與 DHT11 溫濕度感測器接線圖

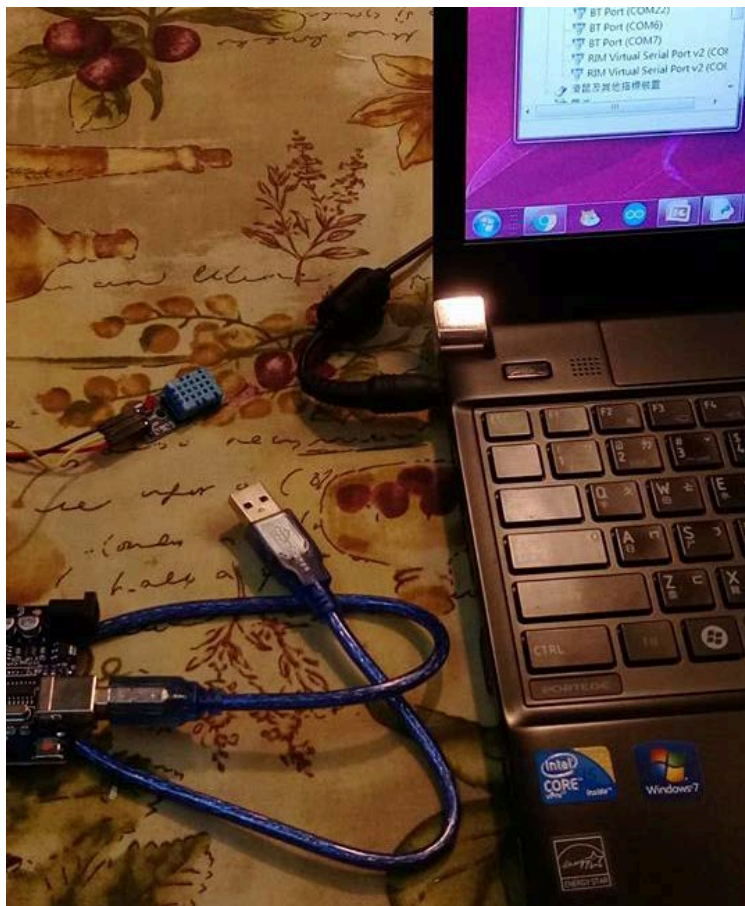


Arduino UNO + Temp / RH Sensor 接線完成圖



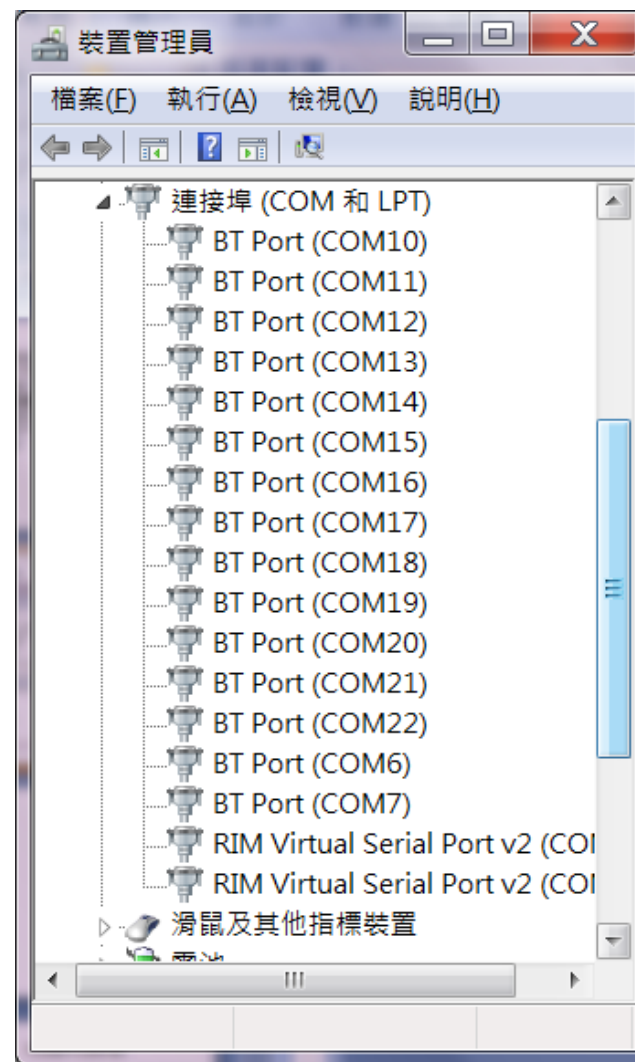
Arduino UNO cable USB 連接電腦 USB 槽

Arduino UNO cable USB 插頭尚未接到電腦 USB 槽中。



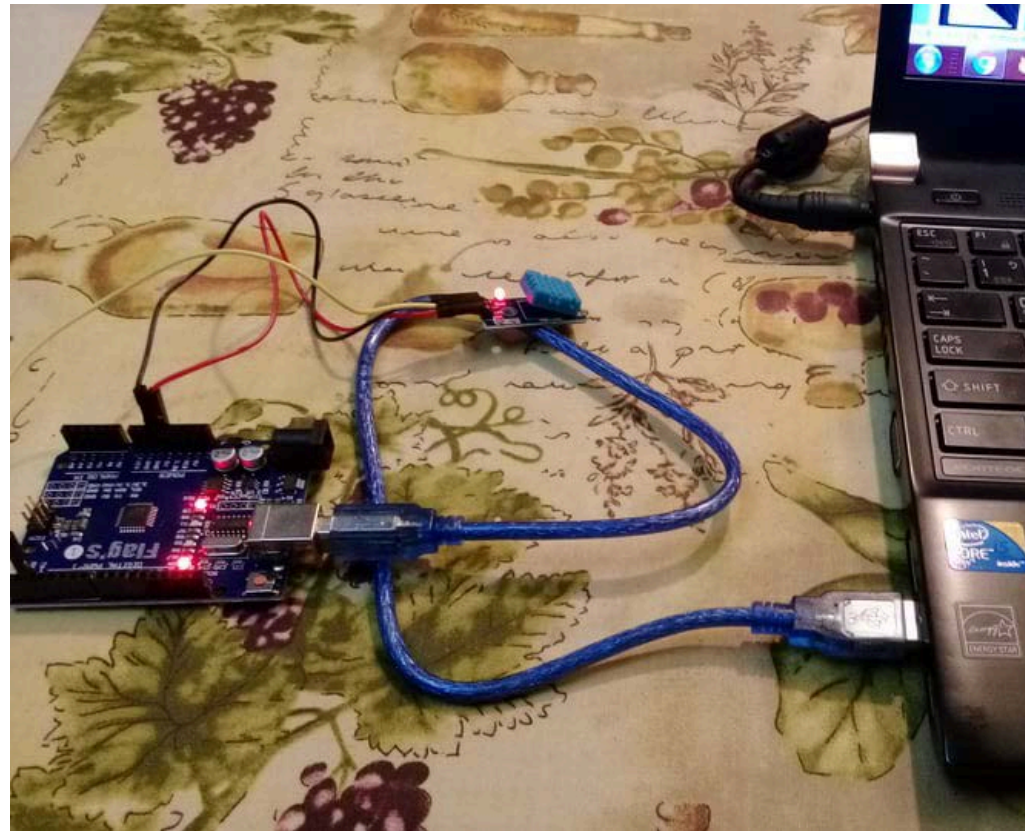
Arduino UNO cable USB 連接電腦 USB 槽

點選**連接埠**後出現 COM Ports 。



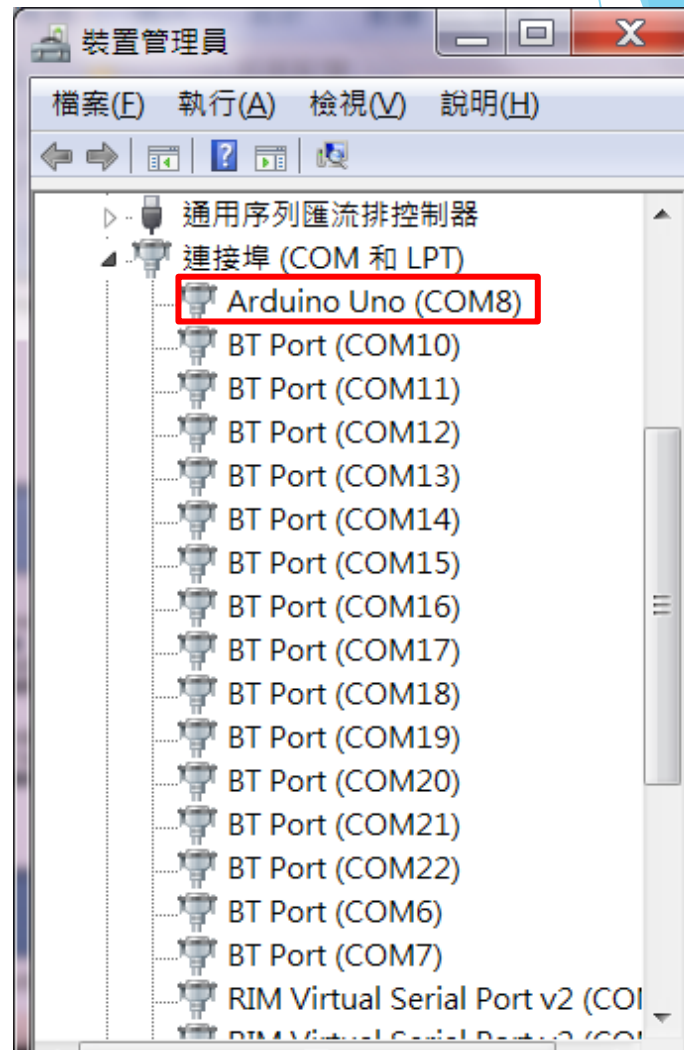
Arduino UNO cable USB 連接電腦 USB 槽

Arduino UNO cable USB 插進電腦 USB 槽中，
Arduino UNO 板和 DHT11 的 LED 燈都亮起。



Arduino UNO cable USB 連接電腦 USB 槽

裝置管理員自動重新掃描後，
連接埠出現 Arduino Uno (COM 8) 。



COM Port 設定

- ▶ Baud rate: 數位訊號傳送速率，每秒傳送多少位元

Baud rate (bits per second) 每秒位元數(B): 9600

1個資料位元 (5-8位元) Data bits 資料位元(D): 8

同位檢查位元 Parity 同位檢查(P): 無

停止位元 Stop bit 停止位元(S): 1

Flow control 流量控制(E): 無

進階(A)... 還原成預設值(R)

Bytes vs String in Python

	Bytes	String
可讀性	低 (給機器讀)	高 (給人類讀)
儲存在電腦內	可	不可

一個 Byte string 由單字 'b' 開頭，例如: `b'23C, 63%\r\n'`，
需要被解碼過後才可以讓人類讀懂。

而 String 則需要被編碼成 Byte string 來讓電腦存取。

Arduino 開發板在資料擷取之應用

pySerial 訊號處理



PART 03

pySerial 套件指令語法



pySerial 套件下程式指令和語法

► Python Shell 下測試常用 pySerial 指令與說明 - 開啟序列埠

```
import serial          #匯入 serial 模組
ser=serial.Serial()
#serial.Serial 括弧內可以設定要開啟的序列埠與傳輸方式
print(ser.name)        #ser.name 是開啟的序列埠
```

None #因為前面 serial.Serial() 括弧內是空的，因沒指定要開啟的序列埠，所以沒有開啟

```
ser=serial.Serial('COM8')
#serial.Serial 後的括弧的字串指定開啟 COM8
print(ser.name)      #再次螢幕輸出 ser.name
```

COM8 #這次 ser.name 是 COM8，表示 COM8 被開啟

pySerial 套件下程式指令和語法

► Python Shell 下測試常用 pySerial 指令與說明 - 讀值

```
ser.is_open    #ser.is_open 回傳 COM8 是否是開啟的布林值
```

```
True #ser.is_open 回傳 True，即 COM8 是開啟的
```

```
s=ser.readline()
```

```
#ser.readline() 回傳 COM8 有 \n 結尾的一行溫濕度資訊到變數 s 中
```

```
print(s)      #螢幕上輸出變數 s 儲存的溫濕度資訊一組
```

```
b'23C, 61%\r\n' #螢幕上輸出溫度23C, 濕度61%
```

pySerial 套件下程式指令和語法

► Python Shell 下測試常用 pySerial 指令與說明 - 資料轉譯

```
ser.readline()
```

```
b'22C, 64%\r\n'
```

從機器讀取的 Bytes 資料型態的一行 b 字母起首，有 \n 的資料

```
sbytes=ser.readline()
```

將這行 Bytes 資料指向新變數 sbytes 的記憶空間

```
sbytes
```

```
b'22C, 64%\r\n'
```

```
sbytes.decode('ASCII')
```

將這行 Bytes 資料轉譯成 ASCII 的字串資料型態

pySerial 套件下程式指令和語法

► Python Shell 下測試常用 pySerial 指令與說明 - 轉譯後處理

```
sbytes.decode('ASCII').strip()  
# 將轉譯的字串資料前、後空格和換行符號都刪去
```

```
'22C, 64%'
```

```
sstrs=sbytes.decode('ASCII').strip()  
sstrs
```

```
'22C, 64%' # 將轉譯前後空格都刪去的字串指向新變數sstrs記憶空間
```

```
sstrs.split(',')  
# 將sstrs的字串資料以split(',')  
將其內的溫度和濕度以list型態儲存
```

pySerial 套件下程式指令和語法

► Python Shell 下測試常用 pySerial 指令與說明 - list 取值

```
DTH=sstrs.split(',')  
#將 sstrs 字串資料以 split(',')  
將溫度和濕度以 list 新變數 DTH 儲存  
DTH=[0] #List DTH 第一個 item DTH[0] 是溫度
```

```
'22C'
```

```
DTH=[1] #List DTH 第二個 item DTH[1] 是濕度
```

```
'64%'
```

pySerial 套件下程式指令和語法

► Python Shell 下測試常用 pySerial 指令與說明 - 輸出格式

`DTH=[1].strip` #利用字串語法 `strip()` 將 `DTH[1]` 濕度前面的空格刪去

```
'64%'
```

```
'Temp=' + DTH[0][: -1] + chr(176) + 'C'
```

#利用 `chr(176)` 將 ° 符號加在溫度值和 `C` 中間

```
'Temp=22°C'
```

```
'RH=' + DTH=[1].strip
```

 #List `DTH` 第二個 item `DTH[1]` 是濕度

```
'RH=64%'
```

```
Time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
```

推 '2018-12-16 19:15:19' #利用 `time.strftime` 將讀取時間轉存成左邊輸出格式

Python Example 1

► 使用迴圈連續讀值、轉譯、處理、輸出

```
import serial
import time
ser=serial.Serial()
#ser.baudrate=9600
ser.port='COM5'
ser.open()
print("Serial Port opened:",ser.name)
for i in range(0,10):
    sbytes=ser.readline() #
    sstrs=sbytes.decode('ASCII').strip()
    DTH=sstrs.split(',')
    t=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
    print(str(i+1).rjust(2),t,'Temp='+DTH[0][:1]+chr(176)+'C', 'RH='+DTH[1].strip())
ser.close()
```

Python Example 1 執行結果

Serial port opened: COM8

1	2018-12-16	20:01:40	Temp=23°C	RH=64%
2	2018-12-16	20:01:41	Temp=23°C	RH=63%
3	2018-12-16	20:01:43	Temp=23°C	RH=64%
4	2018-12-16	20:01:45	Temp=23°C	RH=63%
5	2018-12-16	20:01:48	Temp=23°C	RH=63%
6	2018-12-16	20:01:50	Temp=23°C	RH=63%
7	2018-12-16	20:01:52	Temp=23°C	RH=63%
8	2018-12-16	20:01:54	Temp=23°C	RH=63%
9	2018-12-16	20:01:56	Temp=23°C	RH=63%
10	2018-12-16	20:01:58	Temp=23°C	RH=64%

Python Example 2

► 使用迴圈連續讀值、轉譯、處理、輸出 2

```
import serial          #匯入 serial
import time            #匯入 time
ser=serial.Serial('COM8') #因為是 Winsows 系統,需要直接指定 COM Port, \
                        #這台電腦開發板在裝置管理員中顯示是接到 COM8 , \
                        #開啟並連線序列埠且命名為 ser

if ser.is_open:
    print("Serial Port opened:",ser.name)    #螢幕上顯示使用的 COM Port
    for i in range(0,10):                    #for 迴圈, i = 0,1,...,9
        tstr=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
        #將溫濕度取得日期時間儲存到變數 tstr 中
        sbytes=ser.readline() #讀一個 '\n' 結尾的一行 'bytes' 資料型態的機器可讀資訊 \
                               #並將其儲存到變數 sbytes 中
        sstrs=sbytes.decode('ASCII').strip() #將變數 sbytes 儲存的機器可讀資訊利用 decode() 語法 \
        #將其轉成 ASCII 字串資料型態後刪去前後空格存到變數 sstrs 中
        DTH=sstrs.split(',') #將字串資料型態變數 sstrs 中的以逗點分隔的溫度與濕度 \
                               #利用 split() 分成兩個字串資料存到一個 list 變數 DTH 中
        print(str(i+1).rjust(2),tstr,'Temp='+DTH[0][: -1]+chr(176)+'C','RH='+DTH[1].strip()) \
        #將量測編號、日期時間、溫度、濕度一筆一筆由螢幕輸出 \
        #日期時間戳印 溫度 degree 符號 濕度
    ser.close()
```

在程式執行結束記得要關閉序列埠並連線

Python Example 2 執行結果

Serial port opened: COM8

1	2018-12-16	19:58:08	Temp=23°C	RH=64%
2	2018-12-16	19:58:10	Temp=23°C	RH=63%
3	2018-12-16	19:58:12	Temp=23°C	RH=64%
4	2018-12-16	19:58:14	Temp=23°C	RH=64%
5	2018-12-16	19:58:16	Temp=23°C	RH=64%
6	2018-12-16	19:58:18	Temp=23°C	RH=64%
7	2018-12-16	19:58:20	Temp=23°C	RH=64%
8	2018-12-16	19:58:22	Temp=23°C	RH=64%
9	2018-12-16	19:58:24	Temp=23°C	RH=64%
10	2018-12-16	19:58:26	Temp=23°C	RH=64%

Python Example 3

► 使用迴圈連續讀值、不轉譯直接輸出

```
import serial                                #匯入 serial 模組
ser=serial.Serial('COM8',9600)
# Winsows 系統使用 COM5 以波特率 9600 連線序列埠並命名為 ser
if ser.is_open:
    print(ser.name)                          # 螢幕上顯示使用的 COM Port
    for i in range(0,10):                    #for迴圈, i = 0,1,...,9
        s=str(ser.readline())
        # 讀一個 '\n' 結尾的一行並以字串 ( string ) 資料型態儲存到變數 s 中
        print(s[2:4]+'C',s[7:9]+'%')
        # 將變數 s 儲存的字串指標[2:4]的部分字串 ( 即溫度 ) 加上攝氏的 'C' 符號
        # 將變數 s 儲存的字串指標[7:9]的部分字串 ( 即濕度 ) 加上相對濕度的 '%' 符號
ser.close()
```

Python Example 3 執行結果

24C 55%

24C 55%

24C 55%

24C 55%

24C 56%

24C 56%

24C 55%

24C 56%

24C 56%

24C 56%

#i=0,1,2...,9 總共輸出10筆

Arduino 開發板在資料擷取之應用

matplotlib圖表繪製



matplotlib

- ▶ matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.

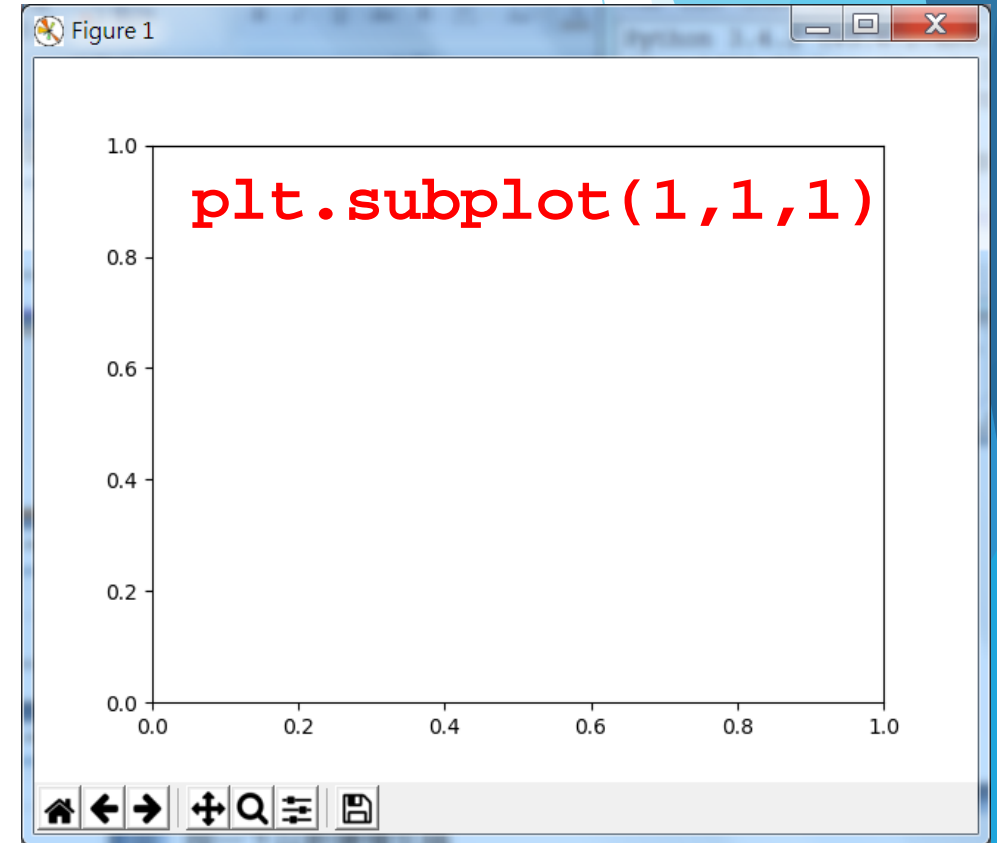


https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots.html

matplotlib library module

▶ 測試 matplotlib 模組之語法 - subplot 子畫面切割範例1

```
import matplotlib.pyplot as plt  
plt.subplot(1,1,1)  
plt.show()
```



matplotlib library module

► 測試 matplotlib 模組之語法 - subplot 函式

subplot() 函式用於子畫面切割可以將一個視窗切為多個子畫面一次顯示多張圖表方便比較或分析。

```
matplotlib.pyplot.subplots(  
    nrows=1, ncols=1,  
    sharex=False, sharey=False,  
    squeeze=True,  
    subplot_kw=None, gridspec_kw=None,  
    **fig_kw  
)
```

[\[source\]](#)

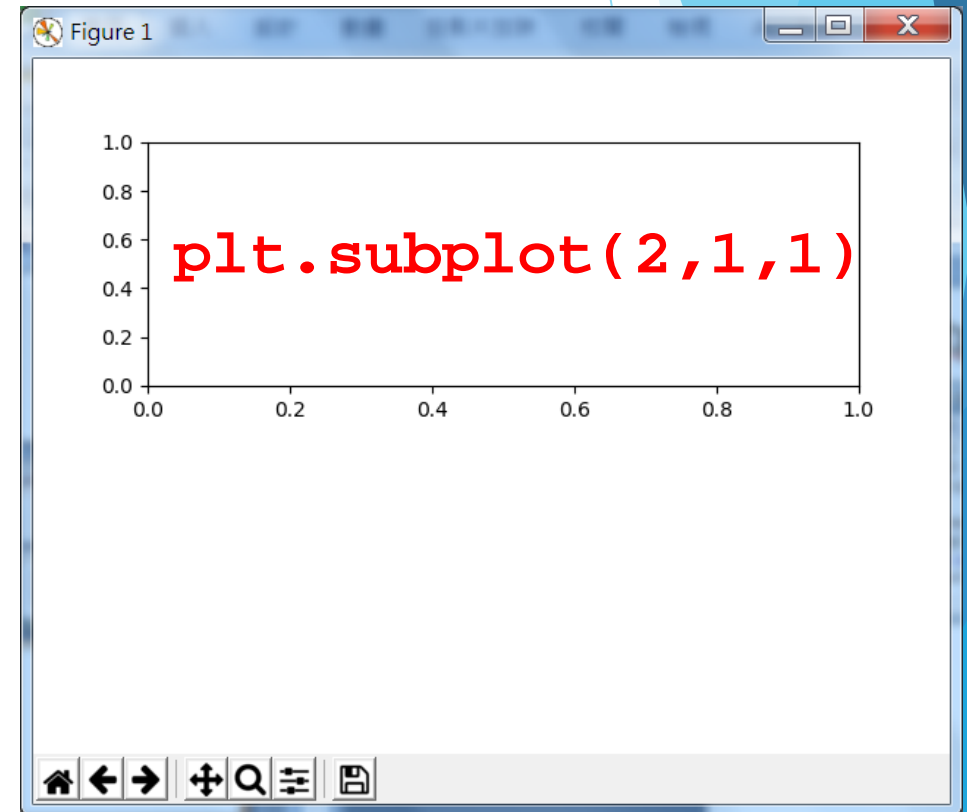
https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots.html



matplotlib library module

▶ 測試 matplotlib 模組之語法 - subplot 子畫面切割範例2

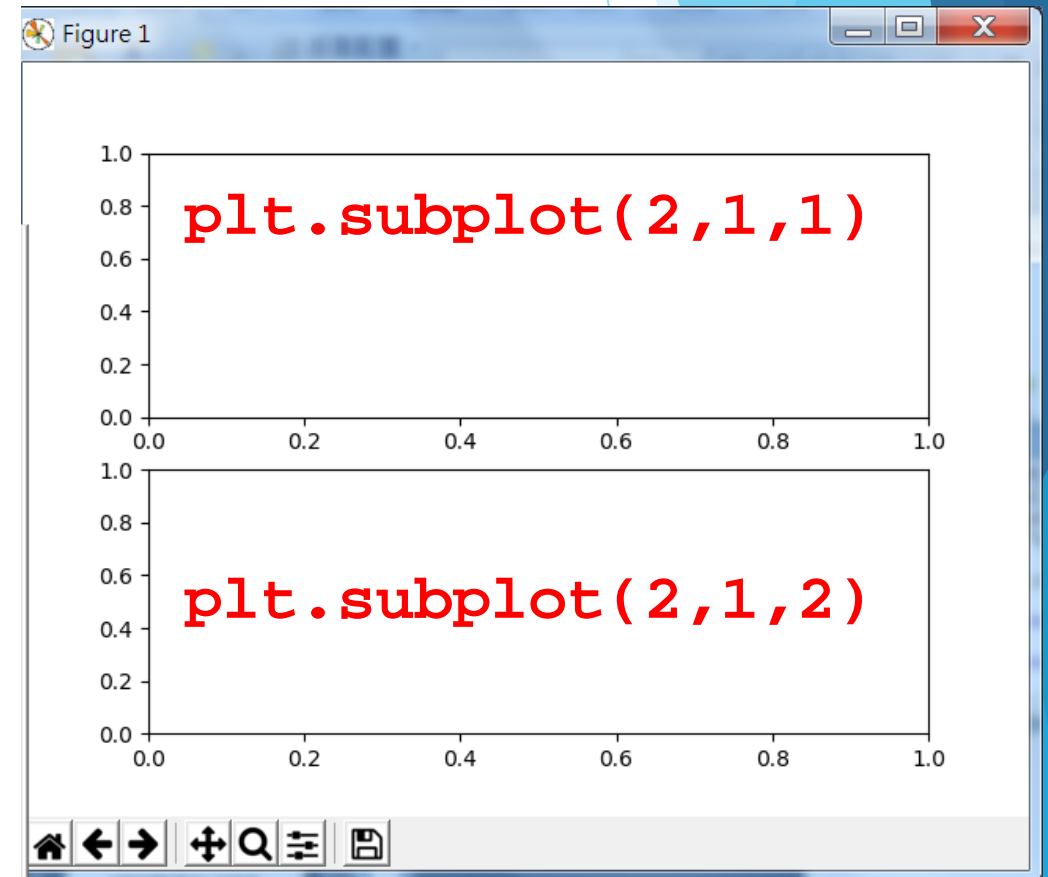
```
import matplotlib.pyplot as plt  
plt.subplot(2,1,1)  
plt.show()
```



matplotlib library module

▶ 測試 matplotlib 模組之語法 - subplot 子畫面切割範例3

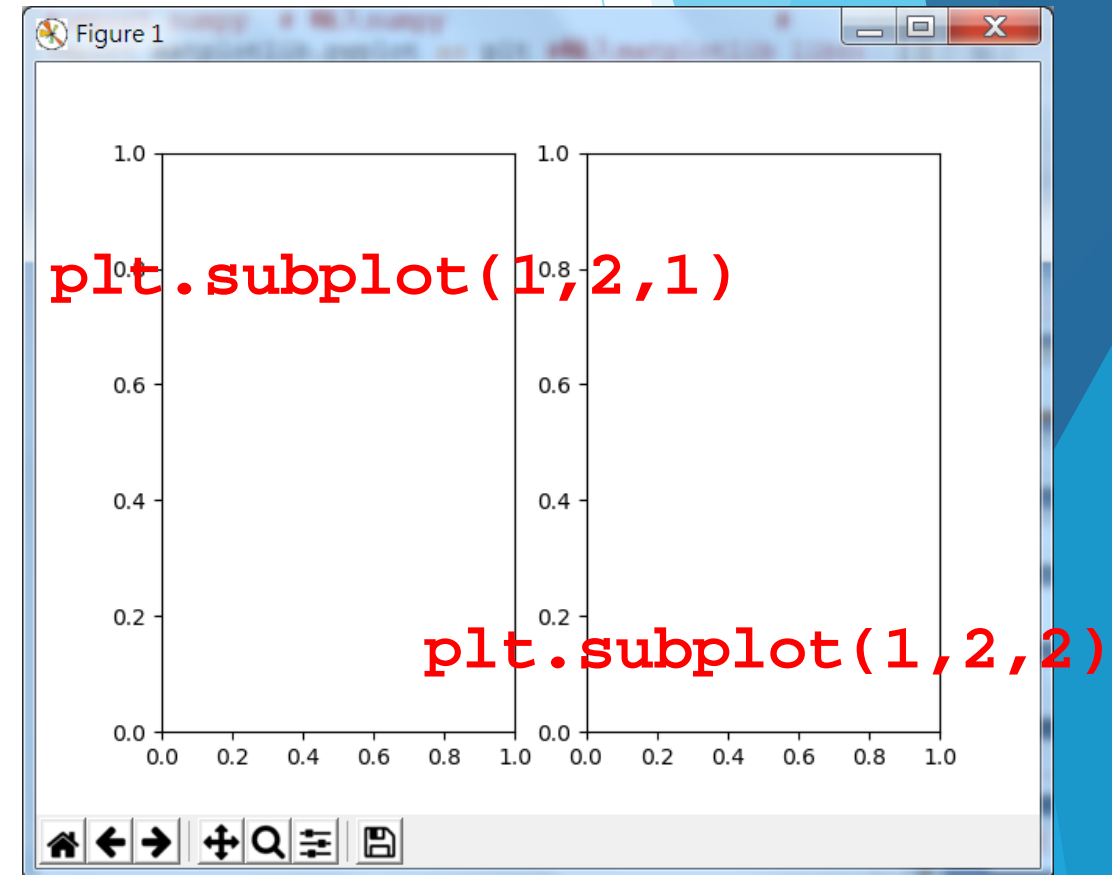
```
import matplotlib.pyplot as plt  
plt.subplot(2,1,1)  
plt.subplot(2,1,2)  
plt.show()
```



matplotlib library module

▶ 測試 matplotlib 模組之語法 - subplot 子畫面切割範例4

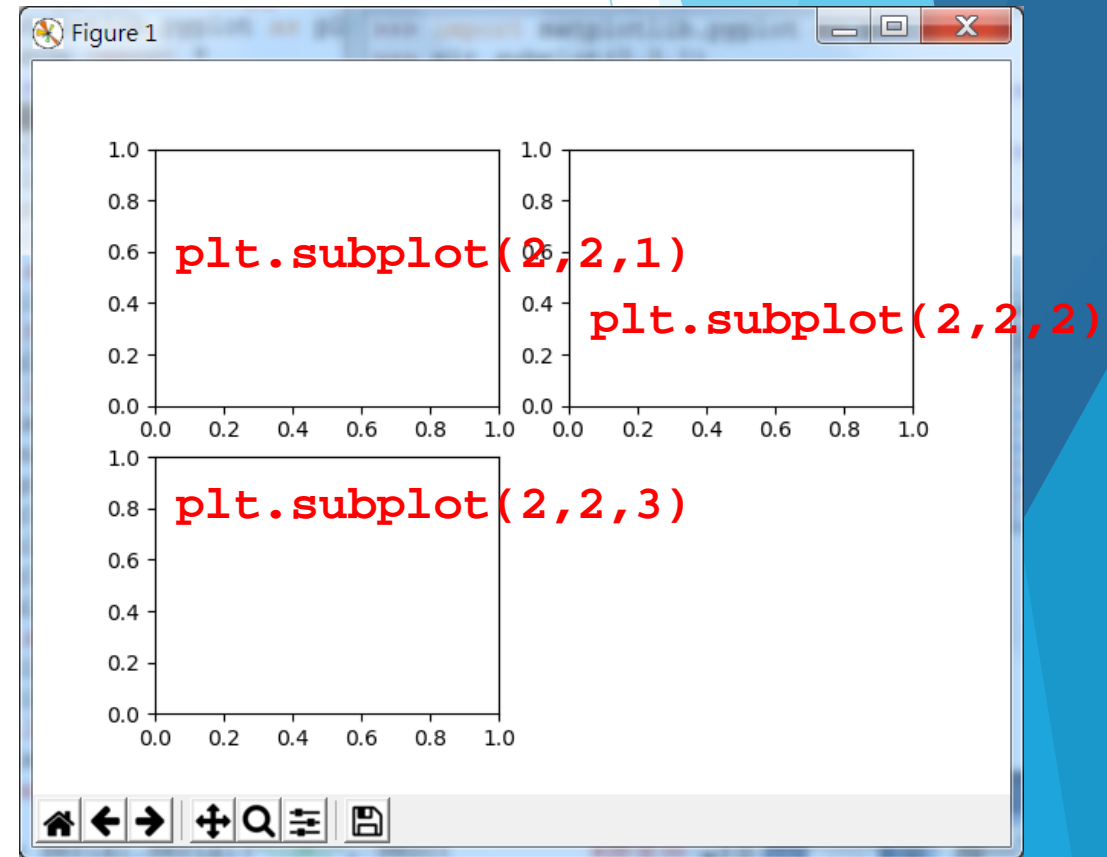
```
import matplotlib.pyplot as plt  
plt.subplot(1,2,1)  
plt.subplot(1,2,2)  
plt.show()
```



matplotlib library module

▶ 測試 matplotlib 模組之語法 - subplot 子畫面切割範例5

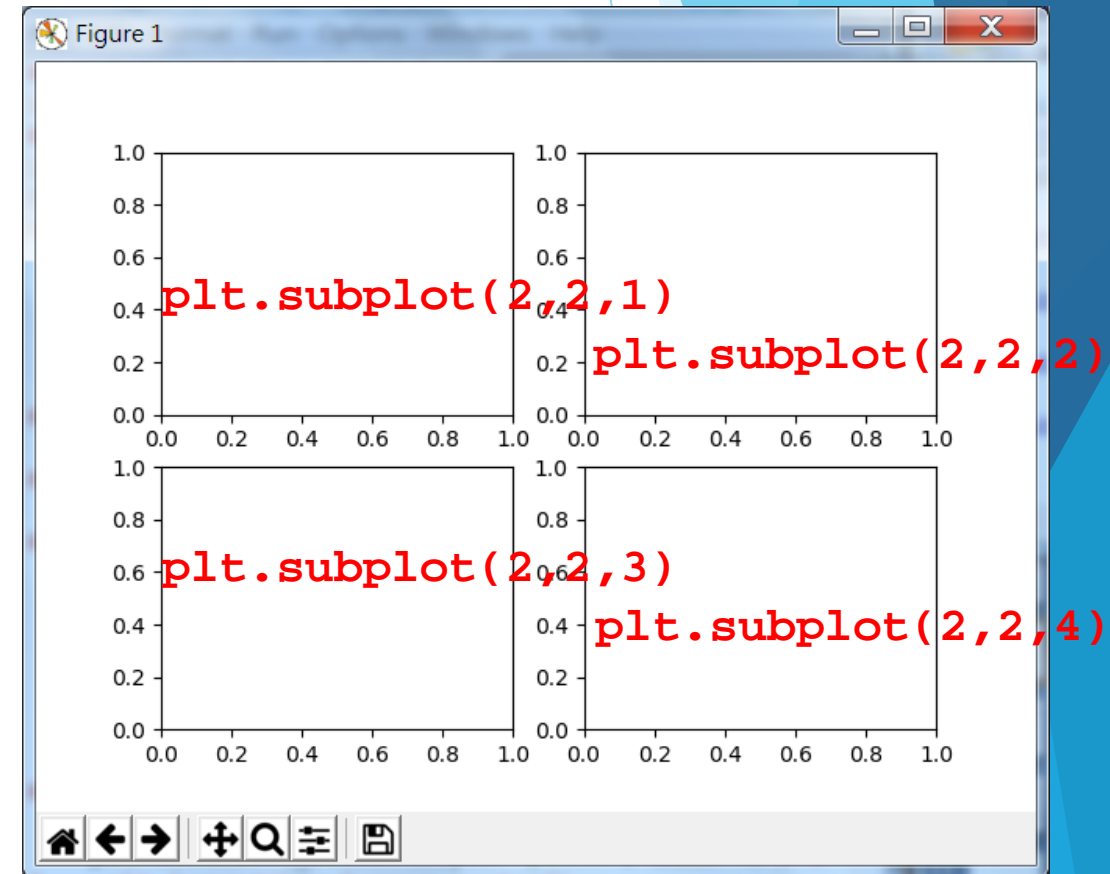
```
import matplotlib.pyplot as plt  
plt.subplot(2,2,1)  
plt.subplot(2,2,2)  
plt.subplot(2,2,3)  
plt.show()
```



matplotlib library module

▶ 測試 matplotlib 模組之語法 - subplot 子畫面切割範例6

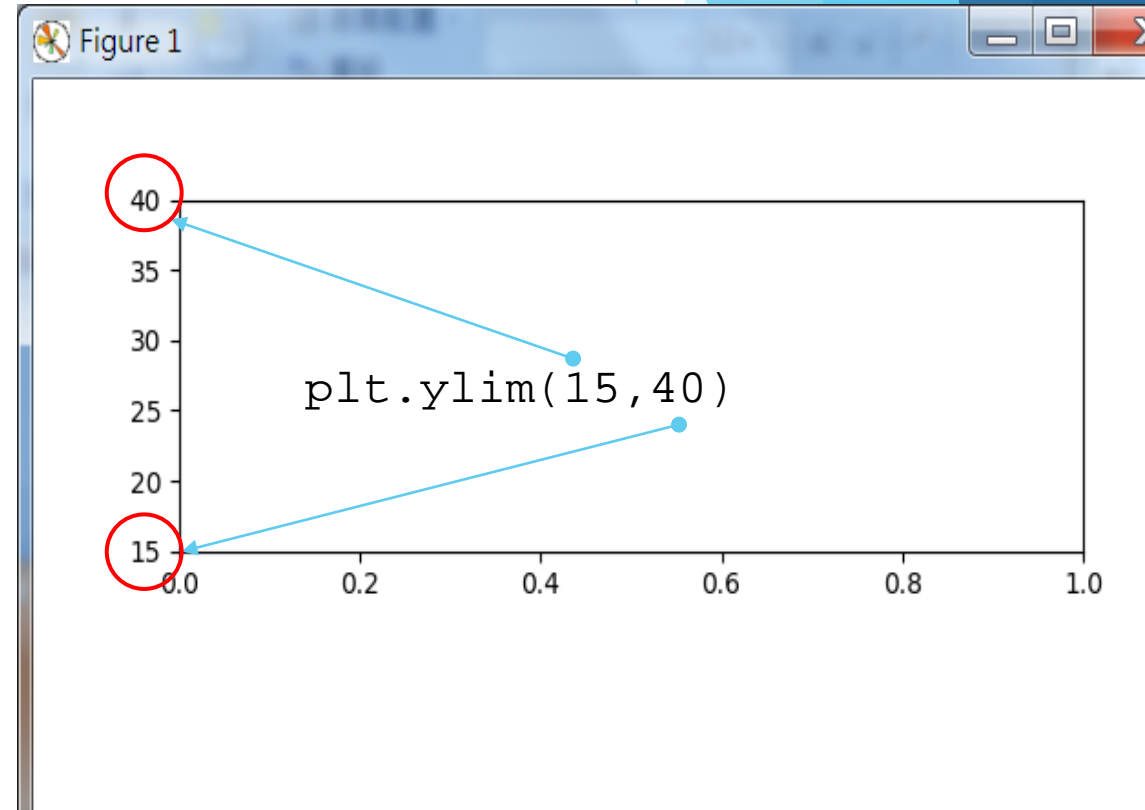
```
import matplotlib.pyplot as plt  
plt.subplot(2,2,1)  
plt.subplot(2,2,2)  
plt.subplot(2,2,3)  
plt.subplot(2,2,4)  
plt.show( )
```



matplotlib library module

▶ 測試 matplotlib 模組之語法 - ylim y軸上下限設定範例

```
import matplotlib.pyplot as plt  
plt.subplot(2,1,1)  
plt.ylim(15,40)  
plt.show()
```



matplotlib library module

▶ 測試 matplotlib 模組之語法 - title 函式

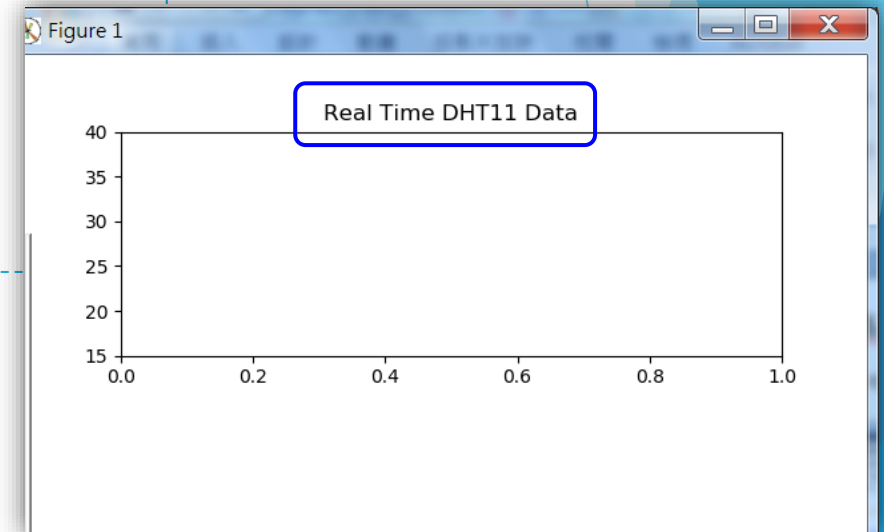
title 函式用來設定圖表上方的標題。

```
matplotlib.pyplot.title(  
    label, #(str)Text to use for the title  
    fontdict = None,  
    loc = {'center', 'left', 'right'},  
    pad = None,  
    **kwargs  
)
```

matplotlib library module

▶ 測試 matplotlib 模組之語法 - title 標題範例1

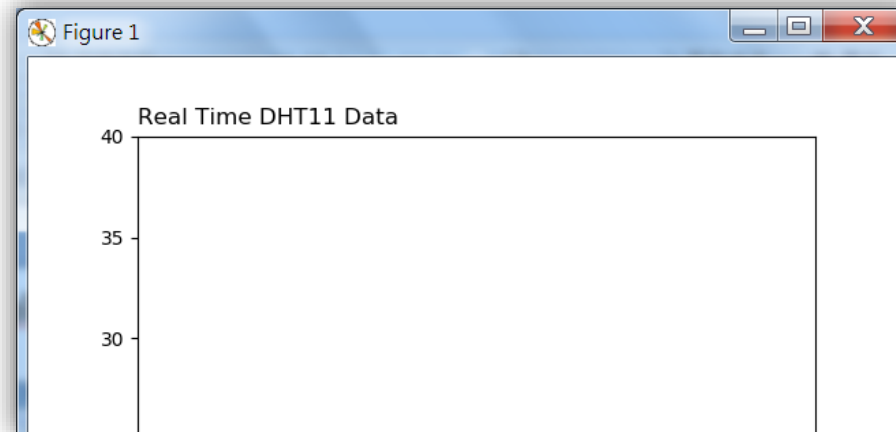
```
import matplotlib.pyplot as plt  
plt.subplot(2,1,1)  
plt.ylim(15,40)  
plt.title('Real Time DHT11 Data')  
#圖上方處加標題  
plt.show()
```



matplotlib library module

▶ 測試 matplotlib 模組之語法 - title 標題範例2

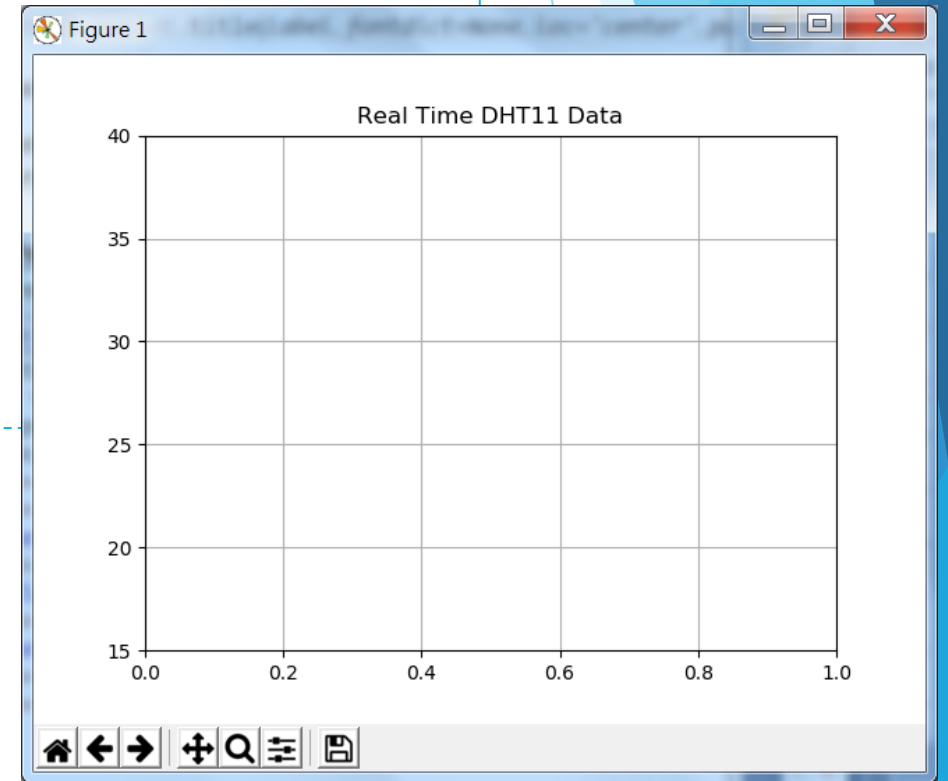
```
import matplotlib.pyplot as plt  
plt.subplot(1,1,1)  
plt.ylim(15,40)  
plt.title('Real Time DHT11 Data', loc='left')  
plt.show()
```



matplotlib library module

► 測試 matplotlib 模組之語法 - grid 顯示網格

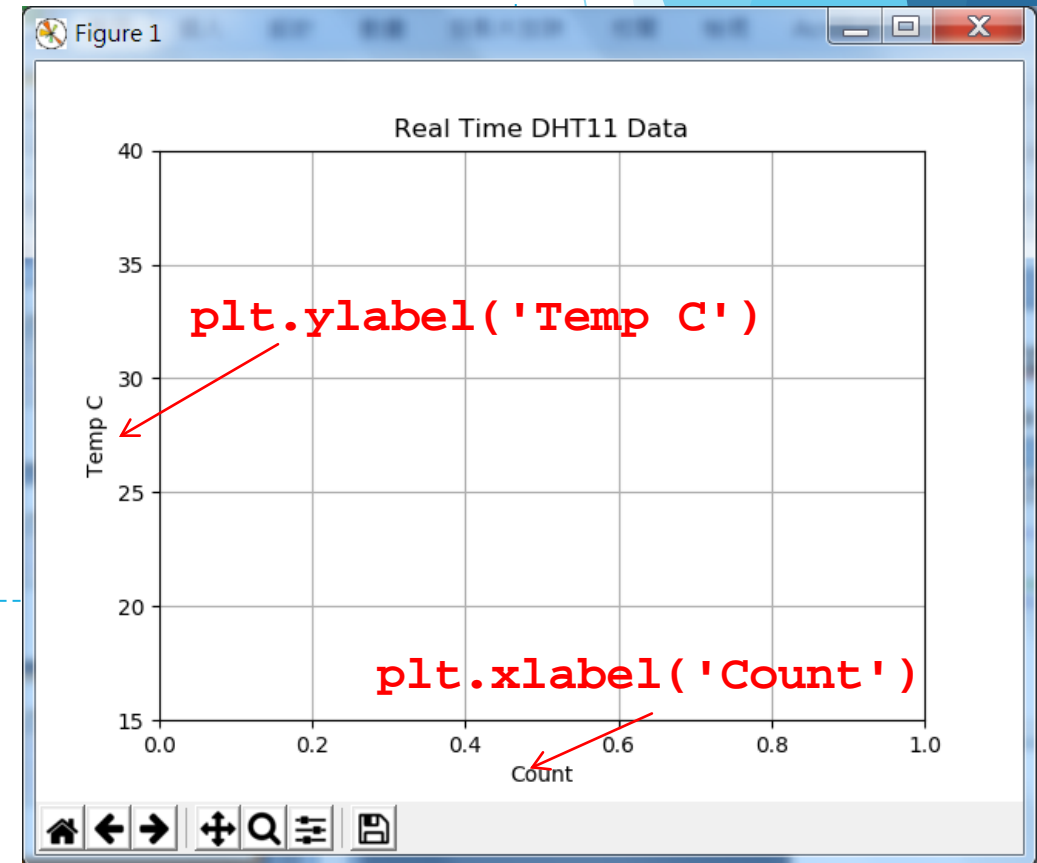
```
import matplotlib.pyplot as plt  
plt.subplot(1,1,1)  
plt.ylim(15,40)  
plt.title('Real Time DHT11 Data')  
plt.grid(True)  
plt.show()
```



matplotlib library module

► 測試 matplotlib 模組之語法 - label 顯示 xy 軸標題

```
import matplotlib.pyplot as plt
plt.subplot(1,1,1)
plt.ylim(15,40)
plt.title('Real Time DHT11 Data')
plt.grid(True)
plt.ylabel('Temp C')
plt.xlabel('Count')
plt.show()
```



主程式 - import

```
import serial # 輸入 serial Library # 序列埠
#import numpy # 輸入 numpy #
import matplotlib.pyplot as plt # 輸入 matplotlib library
from drawnow import *
```



主程式 - CreatePlot() 繪圖函式

```
def CreatePlot(): #Create a function that makes our desired plot
    plt.subplot(2,1,1) #設定上圖的高度、寬度
    plt.ylim(15,40) #設定y軸的上下限值
    plt.title('Real Time DHT11 Data') #上圖上方處加上標題
    plt.grid(True) #讓上圖中呈現灰色網格
    plt.ylabel('Temp C') #設定上圖y軸標題 ( label )
    plt.plot(tempC, 'b^-', label='Degree C') #將溫度數據畫出來
    plt.legend(loc='upper center') #畫出符號說明 ( legend )
    plt.subplot(2,1,2) #設定下圖的高度、寬度
    plt.grid(True)
    plt.ylim(15,100) #設定下圖軸上下限
    plt.plot(humidity, 'g*-', label='Humidity (%)') #將濕度結果畫出來
    plt.ylabel('Humidity (%)') #設定下圖y軸標題 ( label )
    plt.ticklabel_format(useOffset=False) #to stop autoscale y axis
    plt.legend(loc='upper center')
```

主程式 - 設定

```
#設定Serial port為Arduino板接到的com port同時設定Baudrate  
arduino = serial.Serial("COM5", 9600)  
#指定使用 matplotlib 的互動模式來繪製動態數據  
plt.ion()  
tempC = []  
humidity = []  
count=0
```

主程式 - 主迴圈 (從 Arduino 讀值後繪圖輸出)

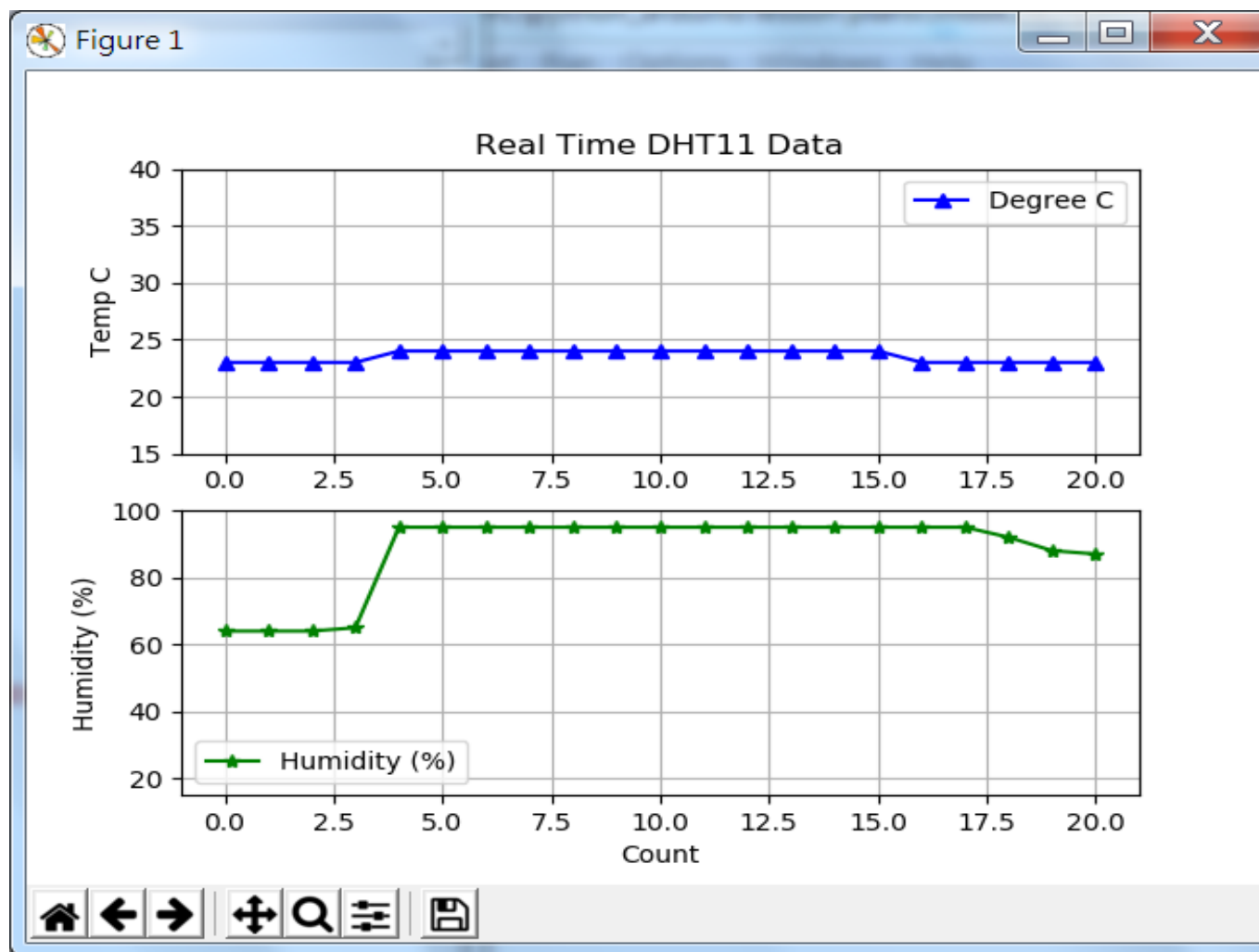
```
while True:
    while (arduino.inWaiting()==0):
        pass
    arduinoString = arduino.readline()
    sstrs=arduinoString.decode('ASCII').strip()
    dataArray = sstrs.split(',')
    # print(eval(dataArray[0][0:2]),eval(dataArray[1][1:3]))
    temp = float(eval(dataArray[0][0:2]))
    hum = float(eval(dataArray[1][1:3]))
    tempC.append(temp)
    humidity.append(hum)
    drawnow(CreatePlot)
    plt.pause(.000001)
    count=count+1
    if(count>20):
        tempC.pop(0)
        humidity.pop(0)
```

#While loop不停的循環
等待直到有數據
沒有數據就繼續等
[有數據] 從serial port讀取數據
將數據拆解成一個陣列(array)
將第一個部分字串轉換成數字並將其指向變數temp
將第二個部分字串轉換成數字並將其指向變數hum
藉append將每個temp數據以陣列方式存到變數tempC
藉append將每個hum數據以陣列方式存到變數humidity

假如數據多於20筆只取最後20筆
#pop out first element of tempC
#pop out first element of humidity



主程式 - 結果



matplotlib.axes.Axes.ticklabel_format

```
Axes.ticklabel_format(*, axis='both', style='', scilimits=None, useOffset=None, useLocale=None, useMathText=None)
```

Change the `ScalarFormatter` used by default for linear axes.

Optional keyword arguments:

Keyword	Description
<code>axis</code>	['x' 'y' 'both']
<code>style</code>	['sci' (or 'scientific') 'plain'] plain turns off scientific notation
<code>scilimits</code>	(m, n), pair of integers; if <code>style</code> is 'sci', scientific notation will be used for numbers outside the range 10^m to 10^n . Use (0,0) to include all numbers. Use (m,m) where $m \neq 0$ to fix the order of magnitude to 10^m .
<code>useOffset</code>	[bool offset]; if True, the offset will be calculated as needed; if False, no offset will be used; if a numeric offset is specified, it will be used.
<code>useLocale</code>	If True, format the number according to the current locale. This affects things such as the character used for the decimal separator. If False, use C-style (English) formatting. The default setting is controlled by the <code>axes.formatter.use_locale</code> rparam.
<code>useMathText</code>	If True, render the offset and scientific notation in <code>mathtext</code>

Only the major ticks are affected. If the method is called when the `ScalarFormatter` is not the `Formatter` being used, an `AttributeError` will be raised.

https://matplotlib.org/api/_as_gen/matplotlib.axes.Axes.ticklabel_format.html

Location String	Location Code
'best'	0
'upper right'	1
'upper left'	2
'lower left'	3
'lower right'	4
'right'	5
'center left'	6
'center right'	7
'lower center'	8
'upper center'	9
'center'	10

The 'best' option can be quite slow for plots with large amounts of data. Your plotting speed may benefit from providing a specific location.

Alternatively can be a 2-tuple giving x , y of the lower-left corner of the legend in axes coordinates (in which case `bbox_to_anchor` will be ignored).